

A Steiner Tree Based Approach For The Efficient Support Of Multipoint Communications In A Multi-Domain Context

Lunde Chen^{*†}, Slim Abdellatif^{*†}, Thierry Gayraud^{*‡}, Pascal Berthou^{*‡}

^{*} CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France

[†] Univ de Toulouse, INSA, LAAS, F-31400 Toulouse, France

[‡] Univ de Toulouse, UPS, LAAS, F-31400 Toulouse, France

Abstract—This work proposes an approach based on Steiner trees to efficiently support multipoint communications in a multi-domain context, where each domain exposes a synthetic and aggregated view of its network. The approach that we propose is based on two pillars: The adoption of a topology aggregation of each domain’s network as a Steiner tree and the use of a shortest path heuristic for the calculation of these aggregated networks as well as the global Steiner tree. An extensive experimental study on random and real network topologies shows the gains made by our approach in terms of both accuracy and computational complexity.

Index Terms—Multipoint communication, Steiner tree, topology aggregation, multi-domain network

I. INTRODUCTION

Network level multipoint communications are useful for many applications and network operations as it reduces the overhead of maintaining multiple one-to-one communications. Establishing a multipoint communication is often modelled as a Steiner tree that is a minimal cost tree connecting the nodes (or edges) involved in the multipoint communication [1]. For example, in wireless ad-hoc network [2], a Steiner tree based multicast delivery structure is proposed to reduce the number of forwarding nodes and hence the number of redundant packets and collisions. In [3], a Steiner tree is used to minimize the energy consumption of broadcast transmissions. Also, in Vehicular Ad-hoc Networks [4], a delay constrained minimum Steiner tree is used as a routing strategy for the dissemination of warning messages. In data centres, [5] proposes using multiple edge-disjoint Steiner trees to data delivery acceleration as well as to handle network failures.

With the emergence of Software Defined Network (SDN), a better perspective of managing multipoint communications emerges. In such cases, Steiner tree based techniques can always be used. [6] proposes constructing bandwidth efficient Steiner trees for multicast in a mobile network and efficient tree morphing that can adapt to multicast group changes in real time. [7] proposes to use k-Steiner tree for group multicast of large traffic. Finally, [8] proposes using Steiner tree to minimize the size of the routing tree for data centre multicast.

In this paper, we consider establishing multipoint communications in a multi-domain context, which consists in establishing a routing tree that spans multiple domains. In this particular context, in practice [9] [10], the network operator

that operates a network domain hides the detailed topology information and simply discloses and advertises to other domains an abstracted view of its domain. Clearly, the abstracted view has an impact on the convergence time of the algorithm in charge of computing the global routing tree as well as the optimality of the obtained tree (the extra cost with respect to the optimal solution computed on the detailed view of each domain).

Conventionally, full mesh is the most used topology aggregation technique, which consists in aggregating the topology with links connecting all pairs of border nodes. It is used as a general-purpose aggregation technique, finds widespread usage in multi-domain networks. Recent works regarding multi-domain SDN use full mesh as well, as presented in [11] [12]. However, this conventional topology aggregation technique targets inter-domain one-to-one communication. Applying full mesh as the topology aggregation technique, as we will present in this paper, is not suitable for multi-domain multipoint communications, the main pitfalls being a non-negligible loss of accuracy and a significant computation complexity. Therefore, an efficient and effective topology aggregation targeted for multipoint communications should be proposed.

The approach proposed in this paper to efficiently support multipoint communications in a multi-domain context relies on the construction of an approximate global Steiner tree with the following characteristics. First, our approach adopts a Steiner tree based topology aggregation to abstract each domain’s topology. The approximate global Steiner tree is hence derived assuming a Steiner tree based abstraction of each domain. We also adopt a shortest path heuristic to compute both the Steiner tree abstraction of each domain and the global Steiner tree. We show through experimentation that this heuristic is robust, flexible and resilient. We also show that the Steiner tree abstraction is significantly more effective than the full mesh abstraction, for both randomly generated topologies and real network topologies. Also, the Steiner tree abstraction requires less computational time with less spacial complexity.

This paper is organized as follows. First, section II presents the system model and introduces the notations used in this paper. Section III defines the shortest path heuristic (SPH) that we adopt in this paper for the construction of the Steiner tree abstraction of each network domain as well as for the

computation of the global Steiner tree. Sections IV and V respectively describe our proposed approach by presenting the benefits of using SPH for topology aggregation and its performance analysis. Section VI concludes the paper and presents the perspectives to this work.

II. SYSTEM MODEL AND NOTATIONS

We consider a multi-domain network composed of multiple network domains interconnected from their border nodes with inter-domain links. The physical network of each domain $d \in D$ of the network is modelled as an undirected graph $G_d^p = (N_d, A_d)$ with non-negative edge costs. We denote as $B_d \subset N_d$ the set of border nodes of domain d . The set of inter-domain links is denoted as A_{edge} , they are also labeled with non-negative link costs. The global physical network is noted as G^p , which is the union of all nodes and arcs in $d \in D$, as well as inter-domain links. Therefore, $G^p = (\bigcup_{d \in D} N_d, \bigcup_{d \in D} A_d \cup A_{edge})$. This network has a theoretical existence since in practice network domains only reveal an aggregated vision of their network. G^p is used in this paper to compute the optimal (minimum cost) global Steiner tree T^p spanning a set of node Z for performance comparison reasons. Figure 1 presents the G^p of a network with four domains a, b, c and d (link costs are omitted for clarity reasons). It also shows the minimum cost global Steiner tree $T^p(Z)$ that connects six end nodes Z (called target nodes) spread over the four domains. Its total cost is denoted as $L^p(Z)$.

The aggregated network disclosed by a network domain d is denoted as $G_d^a = (N_d^a, A_d^a)$, N_d^a (called terminal nodes) is typically the set of domain d 's border nodes B_d , but could also include some internal nodes (for instance, nodes that are within N-hop to all other nodes, etc.). The links belonging to A_d^a are typically logical, each is established on one or many data paths (often multi-hop). Figure 2 presents two possible examples of aggregated networks that could be exposed by a network domain whose physical topology is given in Figure 2-A (the green nodes are the border nodes). In Figure 2-B, a full-mesh topology between all its border nodes is exposed by the domain. In Figure 2-C, a tree-based abstraction including the border nodes but also some internal nodes (the two dark nodes in the figure) is considered. Clearly, it is the prerogative of each domain to choose the nodes (edge and potentially internal) to include in the disclosed abstraction. It is also the role of the domain to choose the appropriate topology to expose. Finally, it is also the role of the domain to choose to which disclosed node (i.e. $\in Z$) an end node participating to a multipoint communication should be attached (one obvious way is to attach to the nearest disclosed node).

We denote as G^a the global undirected graph obtained from the network abstractions exposed by the different domains. The Steiner tree computed for a set of nodes Z over G^a , is denoted as $T^a(Z)$ with a corresponding cost $L^a(Z)$. For illustration purpose, Figure 3 shows the minimum cost Steiner tree computed for the red nodes on G^a assuming two aggregation techniques (full-mesh and a tree-based technique).

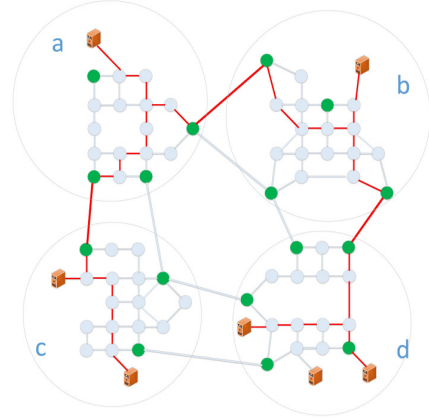


Fig. 1: A multi-domain network and the minimum cost global Steiner tree

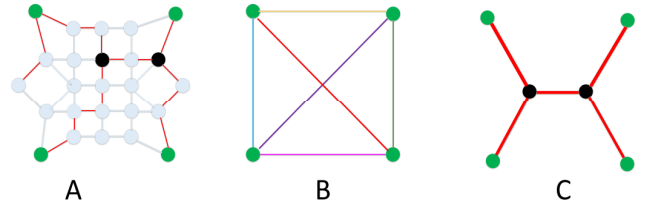


Fig. 2: Topology aggregation in full mesh and in Steiner tree

III. SHORTEST PATH HEURISTIC FOR STEINER TREE CONSTRUCTION

Let's consider a graph $G = (N, A)$, each link $e \in A$ labelled with a cost $c(e)$, with $n = |N|$ and $m = |A|$, a Steiner tree $T = (N_T, A_T)$ is defined as the tree that spans $Z \subset N$ (called Steiner nodes, $q = |Z|$) with minimum total cost $\sum_{e \in A_T} c(e)$.

Finding the optimal Steiner tree is well known to be NP-hard [13]. Here in this paper, we propose using the heuristic presented in Algorithm 1. We will show in the next section that this heuristic is well adapted for our problematic.

The algorithm starts with a random node from Z and chooses it as the Steiner tree T . At each iteration, T grows by the shortest path from T to a next element of Z . When all elements of Z are reached, the algorithm ends.

It is to be noted that the Step 2 in Algorithm 1 is essentially

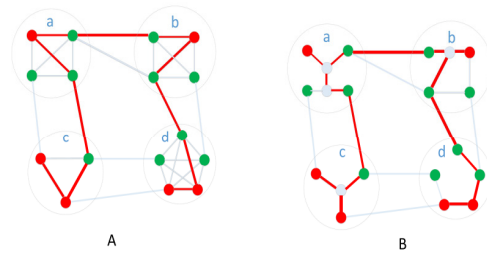


Fig. 3: Minimum cost Steiner trees computed for the target nodes (red nodes) with two aggregation techniques: (A) full mesh and (B) Steiner tree

the Dijkstra’s algorithm, the difference being that the starting point of Dijkstra’s algorithm is replaced by the tree T here.

In our implementation, we use the Fibonacci Heap instead of a priority queue or a heap, which has the specificity that the insert and decrease key operations take constant $O(1)$. This brings down the complexity of Step 2 to $O(n \log n + m)$. Since the Step 2 is repeated $q - 1$ times, the complexity of the whole algorithm is $O(qn \log n + qm)$.

Algorithm 1 Shorest Path Heuristic for Constructing Steiner Tree

- 1: Step 1: Initialization. $T = (z_0, \emptyset)$, with $z_0 \in Z$ being a random element of Z .
 - 2: Step 2: Choice of a shortest path. Choose a node z of Z that is the most in proximity to T . Add to T all the edges of the path from z to T .
 - a. Consider the tree T as a super-node and of zero cost. Add its neighbours to the set of nodes to be visited.
 - b. Repeat:
 - Choose the unvisited node with minimum cost. Add its unvisited neighbours to the set of nodes to be visited (as in Dijkstra’s algorithm).
 - Until we reach a node z of Z .
 - c. Add to T all the edges of the path from z to the tree T (the new super-node).
 - 3: Step 3: End of iteration. Repeat Step 2 as long as all elements in Z are not connected.
-

The heuristic approximates the Steiner tree within ratio of 2, which means that the constructed tree is at most 2 times the optimal solution, expressed as:

$$\frac{c(T_{SPH})}{c(T_{exact})} \leq 2(1 - \frac{1}{q})$$

IV. SPH FOR STEINER TREE BASED APPROACH FOR EFFICIENT SUPPORT OF MULTIPOINT COMMUNICATION IN A MULTI-DOMAIN CONTEXT

The approach proposed in this paper promotes the use of a Steiner tree network domain abstraction. We also proposes using the shortest path heuristic (SPH) of Section III to construct this topology abstraction as well as to construct the Steiner tree that spans nodes belonging to different domains (See Algorithm 2). Next, through experimentation, we show that SPH presents multiple advantages for the support of multi-point communication in a multi-domain context.

Algorithm 2 Constructing multi-domain multicast tree from topology aggregation

- 1: For each domain G_d^p , construct an abstract network view G_d^a using SPH (for Steiner tree aggregation) or using Floyd–Warshall algorithm (for full mesh aggregation). In each abstracted network view, the terminal nodes are preserved. Those domains with border links form G^a .
 - 2: Use the SPH to construct the new Steiner tree spanning Z over G^a .
-

The performance evaluation is based on the networkx graph generator [14]. Two types of graphs are formulated, namely gnp (also known as an Erdos-Renyi graph or a binomial graph) and grid2d (2-dimensional grid graph) graphs.

Some of the conducted experiments consider the case of a single domain. In that case, the number of nodes composing the domain is set to 100. The others consider a network with 20 domains, each consisting of 36 nodes. The domains are connected via 80 inter-domain links. For each experiment, the graphs are generated repeatedly at least 1000 times. The number of nodes being fixed, links are generated randomly following the gnp and grid-2d with an associated cost that is varied from 1 to 20. The performance results are averaged on all generated graphs.

A. Choice of the starting point

As the algorithm of SPH starts from a random node from Z , the resulted Steiner tree constructed could be seriously impacted by the choice of the initial node. It’s important to make sure that the choice of the starting point has limited impact on the performance of the heuristic.

We conduct a set of experiments, by changing the starting point of the SPH each time and compare the total cost of the constructed Steiner tree. More precisely, for each graph, we vary the number of Steiner nodes in each domain and also the starting point (there is as many different choices of starting point as there are Steiner nodes), and calculate the resulting coefficient of variance c_v of the total costs, defined as the ratio of the standard deviation σ to the mean μ .

The results for the single-domain case and the multi-domain case are respectively plotted in Figure 4 and Figure 5. We observe that whatever the number of Steiner nodes in the graph, the coefficient of variance of the Steiner trees constructed from different starting points is always around 0.01% and never surpasses 0.02%, which is quite insignificant and hence negligible. This observation is quite useful as this leads to the effectiveness of starting from a random starting node to construct the Steiner tree with SPH.

B. Constructing Steiner tree from multiple starting nodes

The very limited impact of choosing the starting node with SPH leads to the consideration of having multiple starting nodes, instead of one, so that for really large graphs and in a multi-domain context, the calculation could be distributed on multiple entities. Therefore, we investigate the possibility of extending the heuristic of shortest path to start from multiple starting nodes. With simple modifications, we have the algorithm shown in Algorithm 3.

We vary the number of starting nodes $\in [2, 16]$ for the single-domain graphs. As is shown in Figure 6, for G_d , constructing a Steiner tree from multiple starting points has negligible impact on the total cost of the tree. On average, we observe a variation within the scope of 0.05% in the total cost of the constructed Steiner tree in comparison to the tree constructed from a single starting node. We observe the

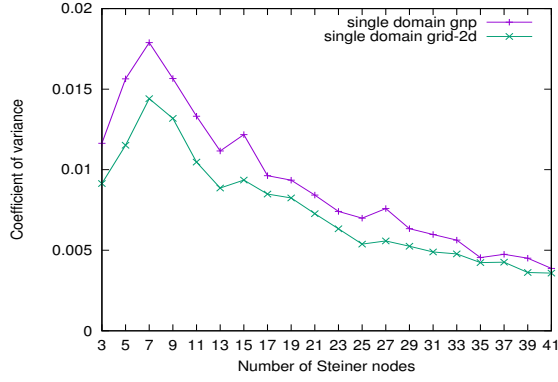


Fig. 4: Coefficient of variance for total costs calculated with SPH from different starting points (single-domain case)

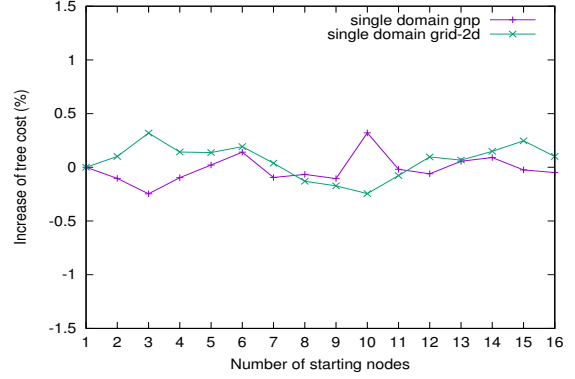


Fig. 6: SPH from multiple starting points: increase of total costs

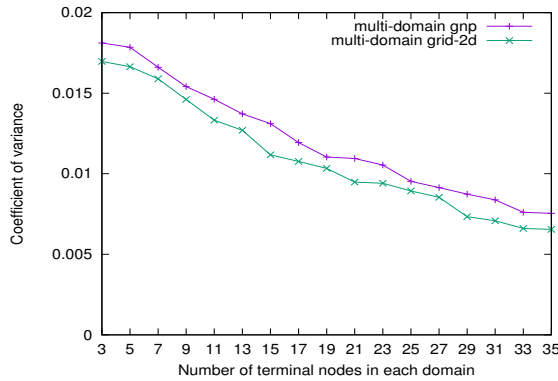


Fig. 5: Coefficient of variance for total costs calculated with SPH from different starting points (multi-domain case)

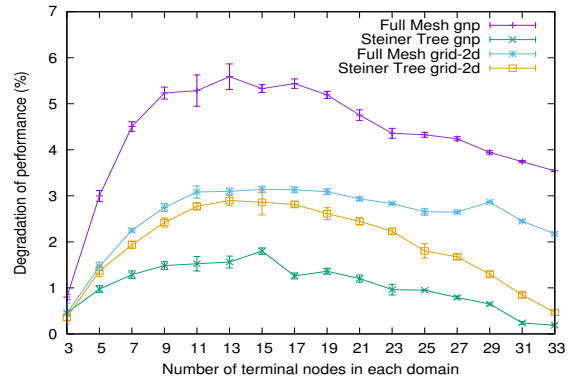


Fig. 7: Full mesh v.s. Steiner tree as topology aggregation for simulated topologies: performance degradation

same result for larger graphs (with number of nodes exceeding 100,000).

Therefore, while constructing a Steiner tree from multiple starting points could be used to bring down greatly the computation time (with paralleled computing), this is not in the sacrifice of other performance indicators.

C. Addition of new Steiner nodes

Our approach is able to handle quite efficiently the case where new nodes join the multipoint communication. Indeed, the addition of a new Steiner node can be easily handled with the SPH heuristics as explained in Algorithm 4. As the experiments from the previous section show, the impact of starting from multiple nodes have negligible impact on the overall cost.

D. Recovery of Steiner tree and Re-aggregation

On the advent of a link or node failure, two recovery alternatives can be adopted:

- either we can reconstruct a new complete Steiner tree with SPH following Algorithm 1.
- or, consider the two broken parts of the Steiner tree as two super-nodes, and use the SPH to connect them (Algorithm 5).

Algorithm 3 SPH with multiple starting point

- 1: Step 1: Initialization. $T = (Z, \emptyset)$
 - 2: Step 2: Choose a set of $s \subset Z$ as the starting nodes (super-nodes). For each starting point, make them zero-potential.
 - 3: Step 3: For each super-node s_i :
 1. increment its potential.
 2. If one node $z \in Z$ is reached:
 - a. add to s_i z as well as all the nodes in the path from s_i to z .
 - b. make s_i zero potential.
 3. Or if one node $s' \in s_j$ is reached:
 - a. add to s_i s' as well as all the nodes in the path from s_i to s' .
 - b. Union s_i and s_j as one super-node.
 - c. make s_i zero potential.
 - 4: Step 4: End of iteration. Repeat Step 3 as long as T is not connected.
-

Algorithm 4 Addition of a new Steiner node

- 1: Use the SPH presented in Algorithm 1 (Step 2) to reach out, until the newly to-be-added multicast node is reached.
-

In comparison to the former method, the second one requires less computation time ($O(N \lg(N))$ v.s. $O(qn \log(n) + qm)$), and have less impact on the recomputed Steiner tree, but it implies an increase in the total cost. Our experiments show that the increase remains under 0.04%. This is why we adopt the second method which is typically used as the recovery mechanism by a domain each time a node failure or link failure occurs.

Algorithm 5 SPH Recovery of broken link & node

- 1: Step 1. When there is a broken link & node, the Steiner tree constructed is broken into two parts. Identify the one of two broken part s_{b1} as a super-node with zero potential and the other part as s_{b2} .
 - 2: Step 2. Using the Algorithm 1 (Step 2) to incrementally increase the potential of the super-node s_{b1} , till the moment when the other part s_{b2} is reached.
-

V. PERFORMANCE COMPARISON WITH FULL MESH AGGREGATION

We have shown that SPH can effectively support multipoint communications in a multi-domain context. However, one of the most important evaluation metrics for an aggregation technique is the degradation of performance, which is the difference ratio between the total cost of the Steiner tree constructed with the global view and that constructed with the abstracted view, defined as $100 * (L^a(Z) - L^p(Z)) / L^p(Z)$.

In fact, as an aggregated topology hides the details about the intra-domain connectivity, inter-domain links with the aggregated topologies often don't lead to the optimal solution. And this holds true both for multi-domain one-to-one communications and multi-domain multipoint communications.

The intuition behind using a Steiner tree as topology aggregation is that a Steiner tree is the minimum spanning tree covering a set of target nodes. When all of those target nodes are included in the computed global Steiner tree, the total cost of the Steiner tree is less than the one obtained with a full mesh aggregation. However, in most cases, only some of those target nodes are covered by the computed Steiner tree. In the opposite extreme case where only two edge nodes are included in the Steiner tree, the full mesh is better. For this reason, we assess the performance degradation and the computation time of each aggregation technique as a function of the number of target nodes.

A. Simulated topologies

For each G_d^p , we have the choice to aggregate the topology of each domain in Steiner tree or in full mesh. We vary the number of Steiner nodes in each domain and compare the average degradation of performance (as well as its standard variance) and time complexity in constructing minimum-weight multicast trees, with the two techniques of topology aggregation, as presented in Algorithm 2.

Performance degradation: The result for performance degradation is shown in Figure 7, which varies around 1%

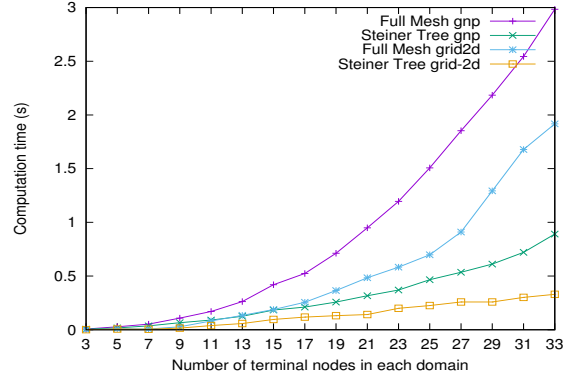


Fig. 8: Full mesh v.s. Steiner tree as topology aggregation for simulated topologies: computation time

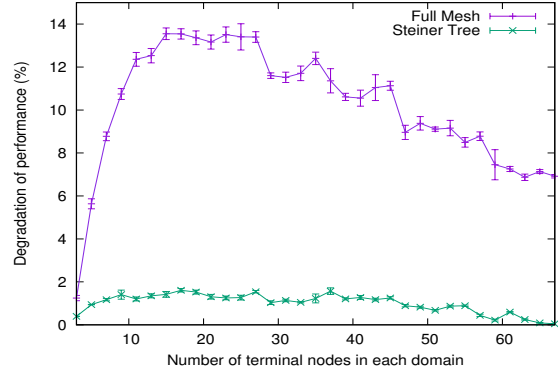


Fig. 9: Full mesh v.s. Steiner tree for topology aggregation, with each domain derived from ESNET

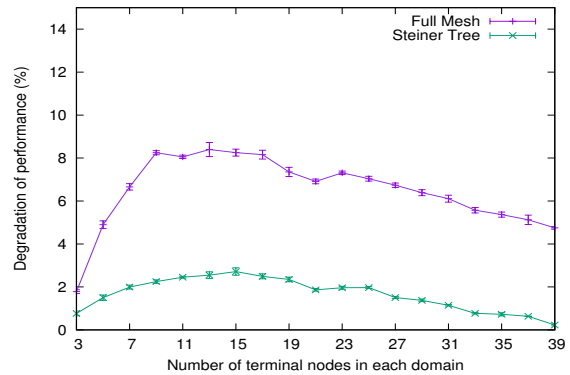


Fig. 10: Full mesh v.s. Steiner tree for topology aggregation, with each domain derived from GEANT

(gnp) to 3% (grid2d) with the Steiner tree as topology aggregation. When network domains are aggregated in full mesh, the performance degradation increases from 3% (grid-2d) to 5% (gnp).

Time complexity: In fact, the time complexity for building the Steiner tree abstraction is $O(qn \log(n) + qm)$ and the space complexity of the Steiner tree abstraction is $O(q)$. We can see clearly that the time complexity of building multicast tree grows linearly as the number of Steiner nodes of each domain increases.

With full mesh as topology aggregation, the computation complexity increases quadratically as the number of terminal nodes for each domain increases. In fact, the time complexity for building full mesh abstraction is $O(q^3)$ and the space complexity of the full mesh abstraction is $O(q^2)$. With Steiner tree as topology aggregation, however, the increase is in a quasi-linear manner. The result is shown in Figure 8.

B. Real topologies

We also measure the performance based on real topologies, with every G_d^p being ESNET [15] or GEANT [16]. As we can see in Figure 9 and Figure 10, for both topologies, the Steiner tree always has better results than full mesh based topology aggregation. Our experiments with the topology of INTERNET2 [17] also confirm this result.

As a conclusion, it's safe to say that Steiner tree is a better candidate than full mesh based topology aggregation, with the aim of constructing an approximate minimum-cost multi-domain multipoint communication trees.

As a complementary result, from our experiments, the Steiner tree based aggregation is always better than full mesh when we target multi-domain broadcast. However, when we target point-to-point communications, full mesh is better than the Steiner tree.

VI. CONCLUSIONS

We have proposed in this paper a Steiner tree based approach to efficiently support multipoint communications in a multi-domain context. Our approach advocates the use of a Steiner tree topology aggregation of domains' networks in replacement of the commonly used full mesh topology aggregation. We also propose to use the SPH heuristic for building the topology aggregations as well as the global Steiner tree that supports the multipoint communication. Through experimentations, we have shown that our algorithms are efficient in terms of computation time and induced total cost and exhibit some interesting properties that make them scalable and cope with the dynamicity of both the network and the multipoint communication. The main perspective to this work is to adapt the proposed algorithms to address multipoint communications with Quality of Service requirements.

ACKNOWLEDGMENT

This work was partially funded by the French National Research Agency (ANR) and the French Defense Agency (DGA) under the project ANR DGA ADN (ANR-13-ASTR-

0024) and by European Union's Horizon 2020 research and innovation programme under the ENDEAVOUR project (grant agreement 644960).

REFERENCES

- [1] M. R. Garey, R. L. Graham, and D. S. Johnson, "The complexity of computing Steiner minimal trees," *SIAM Journal on Applied Mathematics*, vol. 32, 1977, pp. 835–859, .
- [2] W. K. G. Seah, "Distributed Steiner-Like Multicast Path Setup for Mesh-based Multicast Routing in Ad Hoc Networks," *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)*, Taichung, 2006, pp. 192-197.
- [3] Deying Li, Xiaohua Jia and Hai Liu, "Energy efficient broadcast routing in static ad hoc wireless networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 2, pp. 144-151, April-June 2004.
- [4] A. Sebastian, M. Tang, Y. Feng and M. Looi, "A Multicast Routing Scheme for Efficient Safety Message Dissemination in VANET," *2010 IEEE Wireless Communication and Networking Conference*, Sydney, Australia, 2010, pp. 1-6.
- [5] J. Cao et al., "Datacast: A Scalable and Efficient Reliable Group Data Delivery Service for Data Centers," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 2632-2645, December 2013.
- [6] S. Xu, C. Wu and Z. Li, "Software Defined Mobile Multicast," *2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems*, Dallas, TX, 2015, pp. 208-216.
- [7] W. Cui and C. Qian, "Scalable and Load-Balanced Data Center Multicast," *2015 IEEE Global Communications Conference (GLOBECOM)*, San Diego, CA, 2015, pp. 1-6.
- [8] A. Iyer, P. Kumar and V. Mann, "Avalanche: Data center Multicast using software defined networking," *2014 Sixth International Conference on Communication Systems and Networks (COMSNETS)*, Bangalore, 2014, pp. 1-8.
- [9] Sung-eok Jeon, "Topology aggregation: Merged-star method for multiple non-isomorphic topology subgraphs". *Computer communications* 29, 11 (July 2006), 1959-1962
- [10] Uludag, Suleyman, et al. "Comparative analysis of topology aggregation techniques and approaches for the scalability of QoS routing." *Technical Report TR05-010*, DePaul University (2005).
- [11] Egilmez, Hilmi E., Seyhan Civanlar, and A. Murat Tekalp. "A distributed QoS routing architecture for scalable video streaming over multi-domain OpenFlow networks." *Image Processing (ICIP), 2012 19th IEEE International Conference on. IEEE*, 2012.
- [12] Zhao, Laiping, et al. "Traffic engineering in hierarchical SDN control plane." *Quality of Service (IWQoS), 2015 IEEE 23rd International Symposium on. IEEE*, 2015.
- [13] R. M. Karp, "On the computational complexity of combinatorial problems," *Networks*, vol. 5, pp. 45–68, 1975. *2015 IEEE 8th International Conference on Cloud Computing*, New York City, 2015, pp. 829-835.
- [14] "NetworkX Graph Generators,"[Online]. Available: <http://networkx.github.io/documentation/networkx-1.7/reference/generators.html>
- [15] "ESNET", [Online]. Available: www.es.net/engineering-services/the-network/network-maps/historical-network-maps
- [16] "GEANT", [Online]. Available: www.geant.org/Networks/Pan-European_network/Pages/GEANT_topology_map.aspx
- [17] "Internet2 Network Infrastructure Topology", [Online]. Available: www.internet2.edu/media/medialibrary/2013/07/31/Internet2-Network-Infrastructure-Topology.pdf