# Dynamic Compact Multicast Routing on Power-Law Graphs

Pedro Pedroso[1], Dimitri Papadimitriou[2], Davide Careglio[1]
[1] Universitat Politècnica de Catalunya, Barcelona, Spain
Email: {ppedroso,careglio}@ac.upc.edu
[2] Alcatel-Lucent Bell, Antwerp, Belgium
Email: dimitri.papadimitriou@alcatel-lucent.com

*Abstract*—**Compact routing schemes address the fundamental tradeoff between the memory space required to store the routing table entries and the length of the routing paths that these schemes produce. This paper introduces a compact routing scheme that allows the distribution of traffic from any source to any set of leaf nodes along a multicast routing path that defines a distribution tree. By means of the proposed scheme, a multicast distribution tree dynamically evolves according to the arrival of leaf-initiated join/leave requests. To evaluate the performance we consider the following metrics: the stretch of the produced routing paths, the size and the number of routing table entries, and the communication cost. The results obtained by simulation on synthetic power law graphs (modeling the Internet topology) show that our scheme can successfully handle leaf-initiated dynamic setup of multicast distribution trees. Two reference multicast routing schemes (the Shortest Path Tree and the Steiner Tree algorithm) are used to compare the performance of the proposed scheme. While increasing the communication cost compared to the Shortest Path Tree, the proposed scheme achieves considerable reduction of the routing table size compared to both reference schemes. Moreover, the stretch of the resulting multicast routing paths show limited deterioration compared to the minimum value obtained with Steiner Trees.**

## I. Introduction

With the advent of multimedia streaming/content, multicast distribution from a source to a set of destination nodes is (re-) gaining interest as a bandwidth saving technique competing with or complementing cached content distribution. Nevertheless, the scaling problems faced in the 90's when multicast received main attention from the research community remain unaddressed since so far. Indeed, routing protocol dependent multicast routing schemes (such as Distance Vector Multicast Routing Protocol and Multicast Open Shortest Path First) have been replaced by routing protocol independent routing schemes such as Protocol Independent Multicast (PIM) and Core Base Trees (CBT). Overlaying multicast routing on top of unicast suffers however from the same scaling limitations as current unicast routing with the addition of the level of indirection added by the multicast routing application. Multicast routing protocol enables routers to build a (logical) delivery tree between the sender(s) and receivers of a multicast group. Multicast routing table includes the Multicast Routing Information Base (MRIB) and the multicast Tree Information Base (TIB). The MRIB is the topology table, typically derived from the unicast routing table, which carries multicast-specific topology information. The TIB is the collection of routing state created from the exchange of join/prune messages. This table stores the state of all multicast distribution trees at that node.

In this paper, we propose a dynamic compact multicast routing (CMR) algorithm that allows the construction of point-to-multipoint routing paths enabling the distribution of traffic from any source to any set of leaf nodes. The tree determined by this point-to-multipoint routing path is commonly referred to as the Multicast Distribution Tree (MDT) as it enables the distribution of multicast traffic. By means of the proposed scheme, MDTs can dynamically evolve according to the arrival of leaf-initiated join/leave requests. To evaluate the performance of the proposed multicast routing scheme, we measure the stretch of the produced routing paths, the memory size and the number of routing table entries as well as the communication cost, i.e., the number of message exchanged to build the MDT. Two reference multicast routing schemes, based on the Shortest Path Tree (SPT) algorithm and the Steiner Tree (ST) algorithm respectively, are used to compare the performance of the proposed multicast routing scheme. Simulations are performed by running them over synthetic power-law graphs comprising 10k nodes and modeling large-scale topologies such as the Internet.

This paper is organized as follows. Section II introduces the compact routing concept and our main contributions while Section III focus on the proposed dynamic CMR algorithm and the search process segmentation to mitigate the communication cost. Performance results together with their analysis are presented in Section IV. Future work and conclusions are drawn in Section V.

## II. Motivation and Contribution

### A. Compact Routing

Compact unicast routing aims to find the best tradeoff between the memory-space required to store the routing table entries at each node and the stretch factor increase on the routing paths it produces. Such routing schemes have been extensively studied following the model developed in the late 1980's by Peleg and Upfall [1]. Since then, following the distinction operated by Awerbuch [2], various labeled compact routing schemes (nodes are named by polylogarithmic size labels encoding topological information) and name-independent compact routing schemes (node name space is topologically independent) have been designed [3], [4].

As recently formalized in [5], dynamic compact multicast routing algorithms enable the construction of point-to-multipoint routing paths from any source to any set of destinations referred to as leaves. As mentioned above, such routing paths define a distribution tree referred as MDT. The routing algorithm creates and maintains the set of routing states used by each node part of the MDT to derive the entries to forward the multicast traffic received from the source to its leaves.

### B. Contribution

The algorithm proposed in [5] is a labeled and root initiated, dynamic compact multicast routing scheme. The present paper[1] proposes instead the CMR algorithm, a name-independent compact multicast routing scheme for leaf-initiated, distributed and dynamic construction of MDT. In this context, "leaf-initiated" means that the join/leave requests are initiated by the leaves; "distributed" implies that transit nodes process the join/leave requests and compute the routing table entries (no centralized processing by the root); and "dynamic" refers to the on-line capability to timely process the join/leave requests as they arrive without re-computing and re-building the MDT from scratch. The proposed scheme is also characterized by its independence from any underlying unicast routing topology required by leaf-initiated multicast routing schemes such as PIM [6]. In other terms, the local knowledge of the cost to direct neighbor nodes is sufficient for the proposed routing scheme to properly operate. As such, it is actually a true "protocol independent" multicast routing scheme.

To evaluate the performance of the CMR, the following performance metrics are considered. The memory complexity (expressed in terms of memory-bit space) of a multicast routing scheme is defined as for its unicast counterpart: the maximum number of memory-bits required to locally store the routing table entries (the {next-hop, destination} information associated to any routing path) produced by the routing algorithm. However, the stretch is now defined as the total weight of edges used by the algorithm to deliver the multicast packet from source s to all leaf nodes $D \subseteq V$, where V is the total number of nodes, divided by the weight of the minimum ST sourced at $s \in S \subseteq V$. In the present context, an additional metric shall be minimized: the communication cost, defined as the number of messages triggered by the sequence of joining/leaving nodes and exchanged for the algorithm to build the MDT. Aiming to mitigate the communication cost, the proposed algorithm segments the searching space into a local and a global space. This segmentation enables to devise a two-stage search process which, as later shown in Section IV, decreases considerably the communication cost induced by the algorithm.

## III. Compact Multicast Routing Algorithm

The objective of the proposed algorithm is to minimize the routing table sizes of each node $n \in V$ at the expense of i) routing the packets on paths with relative small deviation compared to the optimal stretch obtained by the ST algorithm as well as ii) higher communication cost compared to the SPT algorithm. To this end, the CMR reduces the local storage of routing information by keeping only (direct) neighbor-related entries rather than tree structures (as in ST) or network graph entries (as in both SPT and ST). In other terms, the novelty of this algorithm is on maintaining local topology information (|deg(n)| routing table entries) instead of global topology information (|V-1| entries) providing the least cost next hop during the MDT construction. In the CMR context, the information needed to reach a given multicast source s is acquired by means of a search mechanism (explained in Section III.B and III.C) that returns the upstream node along the least cost branching path to the MDT sourced at s. Such mechanism is triggered whenever a node decides to join a given multicast source s as part of a multicast group g. After a node becomes member of a MDT, a multicast routing entry is dynamically created and stored in the local TIB. From these routing table entries, multicast forwarding entries are created. A detailed description of the algorithm can be found in [7].

As stated before, the reduction in memory space consumed by the routing table results however in higher communication cost compared to the reference algorithms, namely the SPT and the ST. Higher cost may hinder CMR applicability to large-scale topologies such as the Internet. Hence, to keep the communication cost as low as possible, the algorithm's search process is segmented in two different stages. The rationale is to put tighter limits and search locally before search globally. Indeed, the likelihood of finding a node of the MDT within a few hops distance from the joining leaf is high in large topologies (whose diameter is logarithmically proportional to its number of nodes) and it increases with the size of the MDT. Hence, as searching in the entire topology every time a leaf node decides to join a MDT is too costly from a communication perspective, we segment the search process by executing first a local search covering the leaf's neighborhood, and, if unsuccessful, executing a global search over the remaining topology. Additionally, a path $p_{budget}$ is used to bound and prevent excessively lengthy or costly path search.

### A. Preliminaries

Consider a network modeled by an undirected, weighted graph $\boldsymbol{G} = (V, E, c)$, with n = |V| where V represents the finite set of nodes all with multicast capabilities, m = |E| where E represents the finite set of undirectional links, and c a non-negative link cost function $c: E \rightarrow Z^+$ that associates a cost c(l) to each link $l \in E$. Let S be the finite set of source nodes, $S \subseteq V$ and let D be the finite set of candidate destination nodes of a multicast group, $D \subseteq V \setminus S$ for a given source $s \in S$. Let $T_{s,M} = (V_T, E_T)$ be a connected sub-graph without cycles of $\boldsymbol{G}$, i.e., a tree rooted at $s \in S$ with $M \subseteq D$. In the context of this paper, the graph $T_{s,M}$, referred to as MDT is dynamically constructed: each step ω, ω = 1,2,..,|D|, a randomly selected node $u \in D \setminus M$ decides to join $T_{s,M}$. If node u is already part of $T_{s,M}$ ($u \in V_T$) then it is either a transit or branching node of the MDT. Otherwise, node u is not part of $T_{s,M}$ ($u \in D \setminus M$) and it must search for the least cost branching path towards a node $v \in T_{s,M}$. Among all possible paths from node u to $v \in T_{s,M}$ of finite cost $c_{u,v}$, the least cost branching path is denoted by $p_{u,v}^* = \min\{c_{u,v} \mid p_{u,v} \in P_{u,v}\}$ and its cost $c_{u,v}^*$. Two types of messages are involved in this process, namely the request (type-R) messages flowing in the upstream direction, i.e., towards the multicast source, and response (type-A) messages sent in the downstream direction, i.e., towards the joining leaf node u. Type-R messages comprise a maximum path budget, $p_{budget}$, that discards messages with too lengthy or too costly dissemination range, and a sequence number {u_id, <s,g>} to prevent duplication of messages, where u_id identifies the leaf

[1] An extended version of this paper is available as technical report [7]

node and <s,g> encodes the multicast source/group pair. Type-A messages comprise the radial cost $c_{w,v}$* (described below) where w is the local node and $v \in T_{s,M}$ and the identifier of the vicinity edge nodes when flag e=0. The flag e distinguishes the messages exchanged during the search stages, both type-R and type-A messages are flagged as *internal*, e=0, if belonging to the local search procedure, and as *external*, e=1, otherwise.

### B. Local-Search

This first stage consists in a limited search within a certain perimeter of the topology around the joining leaf u. As illustrated in Fig.1, the contiguous set of nodes covered during this first stage is called vicinity, $B \subseteq V$, where nodes $b \in B$ are referred to as vicinity nodes. The vicinity B is delimited by vicinity edge nodes, $b_v$, i.e., nodes at a given hop-count distance, determined either by one of the two following criteria: i) cost-threshold or ii) number of vicinity nodes proportional to $n^{0.5}$ / log(n). In Section IV, we show that for power law graphs this proportionality leads to the minimum communication cost. During this stage, the $p_{budget}$ of each type-R message carries the criterion value (set at leaf node u) that delimits the vicinity of leaf node u, B(u). If the criterion is set to the cost-threshold, starting from node u, $p_{budget}$ value is decremented at each hop according to the travelled link cost; nodes with $p_{budget} \geq 0$ determine nodes $b \in B(u)$. On the other hand, if the criterion is set to the maximum number of nodes part of its vicinity B(u), $p_{budget}$ is decremented at each hop with the vicinity node's out-degree. In both cases, nodes setting $p_{budget} < 0$ are identified as vicinity edge nodes of B(u). For instance, Fig. 1 assumes a maximum $p_{budget}$ of 8 at node u. At its neighboring node $b_1$, $p_{budget}$ = 8 – (deg(u)=5) = 3. Hence, when the vicinity node $b_1$ forwards a type-R message to its neighbor nodes (except, by application of split horizon, to the node from which the type-R message has been received), the value $p_{budget}$ = 3 – out-deg($b_1$) = 0. Applying this procedure to node $b_2$ leads to the same result since the out-degree of this node is also equal to 3. This procedure settles the maximum reachability of type-R messages with flag e=0 by determining the size of the vicinity |B|, whenever $p_{budget}$ = 0.

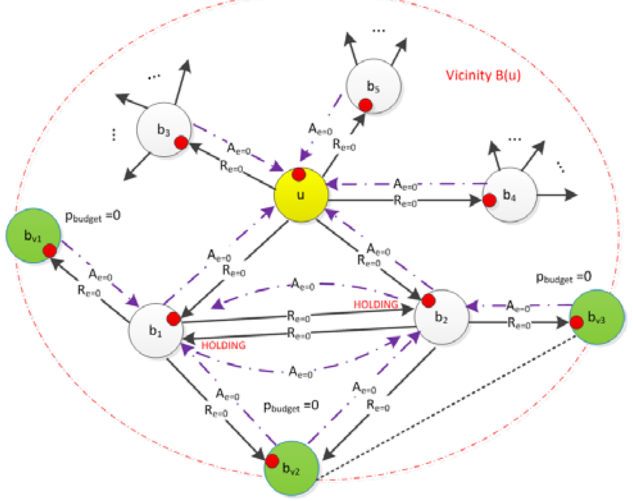

**Fig. 1 - Local Search stage: search the node of the MDT within a limited perimeter called vicinity, B(u).**

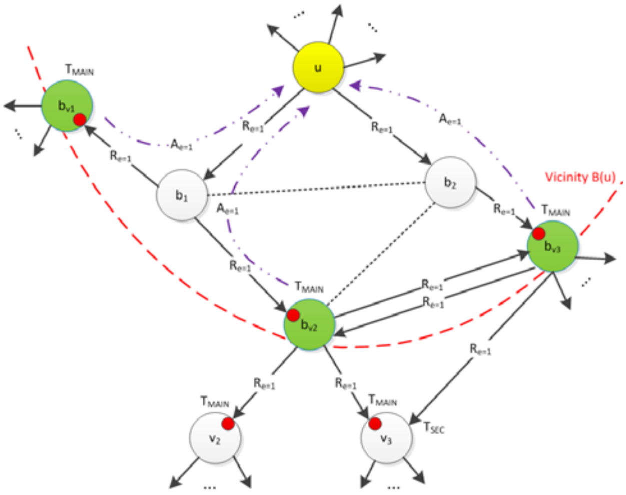

**Fig. 2 - Global Search stage: If local search fails to find a node of the MDT, a search outside the vicinity must be performed.**

The local search starts with the leaf node u sending internal type-R messages (i.e., flag e=0) to all its direct neighbor nodes b (upstream nodes) to find the least cost branching path to a branching node $v \in T_{s,M}$ ($v \in V_T$). Referring to Fig.1, leaf node u sends type-R message to nodes $b_1,\ldots,b_5$. This process continues until the type-R message reaches i) a node $v \in T_{s,M}$ and $p_{budget} > 0$ or ii) a node $v \notin T_{s,M}$ and $p_{budget} = 0$. In the former case, a node belongs to the tree is found; in the latter, a vicinity edge node is reached (node $v = b_v$) but no nodes belong to the tree are found. At this point, node v replies to its neighbor node(s) from which it has received the type-R message(s) with a type-A message. If node $v = b_v$, then the radial cost is set to infinite. If not, the radial cost is computed as follows. Each downstream node w ($w \neq b_v$, $w \notin T_{s,M}$) computes all the branching path costs $c_{w,v}$ from itself to node v (where either $v \in T_{s,M}$ or $v = b_v \neq T_{s,M}$). The cost $c_{w,v}$ is defined as the sum of the cost of edge joining node w to one of its upstream node i and the cost of the path from node i ($i \notin T_{s,M}$) and v ($v \in T_{s,M}$ ). The latter, referred to as the radial cost, is included in the type-A message sent from node i to w. Node w then selects the least cost branching path $p_{w,v}$* and sends the corresponding cost value $c_{w,v}$* to its own downstream node(s). Receiving nodes process this value as the new radial cost and the computation starts again. This stage terminates when node w = u and the leaf node u has received all type-A message (in response to the type-R messages it initiated). If |type-A message| = 0 at waiting timer $w_t$ expiration (set to cope with the maximum round-trip time of a type-R message within B(u)) or the cost value $c_{w,v}$ in all received type-A message is set to infinite, node u declares the multicast source s unreachable and launches the global search method (see Section III.C). Otherwise, the process is completed and leaf node u determines the upstream neighbor node along the least-cost branching path $p_{u,v}$* (= min{$c_{u,v}$ | $p_{u,v} \in P_{u,v}$}) to $T_{s,M}$. Leaf node u then sends to this upstream neighbor node a request message to join $T_{s,M}$.

### C. Global-Search

This stage represents the search of the MDT's branching node outside the vicinity of the leaf node. This process is triggered by the leaf node when the local search phase ends by declaring the multicast source s as unreachable in its vicinity. The global search phase comprises a set of distributed search processes triggered by the leaf node u and started at each vicinity edge node $b_v$ (see Fig.2). Type-R messages marked as external (i.e., flag e=1) are used in this search phase. Two issues can arise here. The first one is that the external type-R messages have to reach the vicinity edge nodes without being flooded inside the B(u) again. For this purpose, the leaf node u sends the external type-R messages directly to each of its vicinity edge nodes $b_v$ along a single path. Indeed, during the local search phase, the internal type-A messages (i.e., flag e=0) received by the leaf node u include the identifier of the node $b_v$ that initiates them. As well, vicinity nodes $b \in B(u)$ keep per vicinity edge node $b_v$, a single active interface from which type-A messages with infinite radial cost have been received (indicating that the neighbor node sits along the path from leaf node u to a given edge node $b_v$). Secondly, to avoid that a node $b \in B(u)$ within the vicinity receives back external type-R messages during the global search stage vicinity edge node $b_v$ filter incoming type-R messages (e=1). The type-A messages sent during the local search are tagged with the flag e=0 sent in response to the reception of type-R message (e=0). Interfaces sending such type-A message are removed from the list of interfaces for forwarding of type-R message (e=1). The exception is for interfaces having received a type-R message (e=1) with leaf node u as sender to enable edge vicinity nodes to send back the answer to node u once the global search completes for that node $b_v$.

During the global search phase, the $p_{budget}$ value is bound at node u by a threshold set to the graph diameter (length of the longest shortest path). Approximation algorithms exist to compute this value as well as method for computing a lower and upper bound [8]. Each node $b_v$ sets the maximum waiting timer $w_{b,t}$, $T_{MAIN}$ in Fig. 2, and the subsequent search process proceeds as follows (more details can be found in [7]). For instance, assume that node $b_v$ sends external type-R messages to each of its neighbor nodes except to its downstream node as explained here above. It then waits for receiving the same number of external type-A messages. Upon reception, node $v_b$ determines the least-cost branching path $p_{u,v}$* to $T_{s,M}$ ($p_{u,v}$* = min{$c_{u,v}$ | $p_{u,v} \in P_{u,v}$}), where u = $b_v$. Node $b_v$ is ready to answer back to leaf node u once either of the following is met: i) it receives the entire set of type-A messages from its upstream neighbor nodes (before its waiting timer $w_{b,t}$ expires) or ii) the waiting timer $w_t$ initiated after reception of the first type-R message (e=1) from leaf node u expires. Once one of these two conditions is met, node $b_v$ computes the branching path cost $c_{u,v}$ from itself to any node $v \in T_{s,M}$ using the radial cost $c_{w,v}$ received from its upstream neighbor nodes w and the cost of the link from itself to node w. Node $b_v$ then selects the least cost branching path $p_{u,v}$*, and sends the corresponding cost value, $c_{u,v}$* directly to the joining leaf node u. If |type-A message| = 0 at waiting timer $w_{b,t}$ expiration, the cost value $c_{u,v}$ is set to infinite indicating that the multicast source s is unreachable. Hence, as soon as this search phase terminates, each node $b_v$ returns a unique type-A message (e=1) directly to the leaf node u from which it initially received an external type-R message. Thus, no computation or selection is performed by nodes $b \in B(u)$ along the path taken by the type-A messages (e=1) sent towards the leaf node u. This path is determined by the incoming interface maintained by each node $b \in B(u)$ upon reception of type-R message (e=1) from leaf node u. Fig.2 shows the node $b_1$ receiving two type-A messages from $b_{v1}$ and $b_{v2}$. Contrary to the local search stage, here $b_{v1}$ does not perform any computation or routing decision. It just forwards the incoming type-A (e=1) messages received from nodes $b_{v1}$ and $b_{v2}$ towards the leaf node u that can receive as many type-A messages as its number of vicinity edge nodes. Note that i) the records locally created during the local search phase are subsequently deleted by the node sending a type-A message (e=0) that does not include an infinite cost to a vicinity edge node $b_v$, and ii) the records remaining and/or locally created during the global search phase are deleted by the node sending a type-A message (e=1).

## IV. Performance analysis

The performances of the proposed compact multicast routing algorithm are analyzed by its simulation on a large-scale topology (10k nodes and 35k links) generated by means of GLP [9]. This evolutive topology generator, which relies on generalized linear preferential attachment, produces power-law graphs that are representative of the Internet Autonomous System (AS) topology, in particular, in terms of clustering coefficient. The execution scenario considers the construction of point-to-multipoint routing paths for multicast groups of increasing size from 500 to 2500 nodes (selected randomly) with increment of 500 nodes. Each execution is performed 10 times by considering 10 different multicast sources.

We compare the performance of the proposed CMR algorithm to the Shortest-Path Tree (SPT) and the Steiner Tree (ST) algorithms. In addition to the routing path stretch and memory-bit space consumption, the performance metrics include the communication cost. The SPT algorithm provides the reference for the communication cost. It is constructed from a loop-avoidance path-vector routing algorithm carrying the identifier of the multicast source s and the routing path to reach that source. Each node keeps thus a routing table entry per neighbor node (to exchange messages) and a routing table entry per path to the multicast source s. The ST algorithm provides the reference in terms of stretch. In order to obtain the near optimal solution for the ST, we consider a ST-Integer Linear Programming formulation. For this purpose, we have adapted the formulation provided in [10] for bi-directional graphs. The communication cost for the ST measures at each step of its construction the number of messages initiated by nodes part of the MDT. These messages contain the minimal information for remote nodes not (yet) belonging to the MDT to join it. Using this information, each node knows how to reach the closest node of the MDT. Thus, although the ST is computed centrally, the communication cost accounts for the total number of messages exchanged during the MDT building process as a dynamic scenario would perform.

### *A. Stretch*

Fig.3 illustrates the stretch ratio of the multicast routes (i.e. MDT) set up by the CMR and the SPT algorithms compared to the ST reference algorithm. The multiplicative stretch for the CMR is slightly higher than 1. Its trend curve decreases as the multicast group size increases (from 1.08 up to 1.04 for multicast group size ranging from 500 to 2500). In addition, it remains constant from group size of 2000 to 2500. Compared to the SPT, the CMR maintains a constant average gain of 6.5% along the different group sizes.

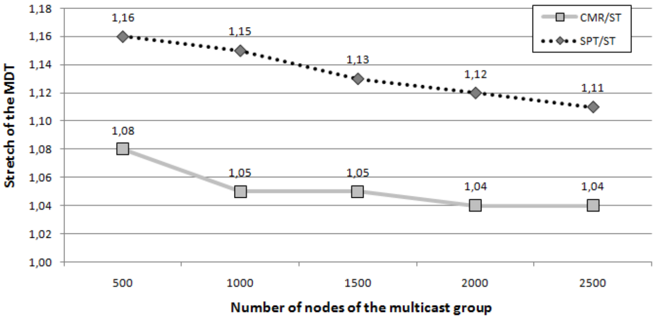

**Fig. 3 – Stretch of the MDT as function of the number of nodes of the multicast group covered.**

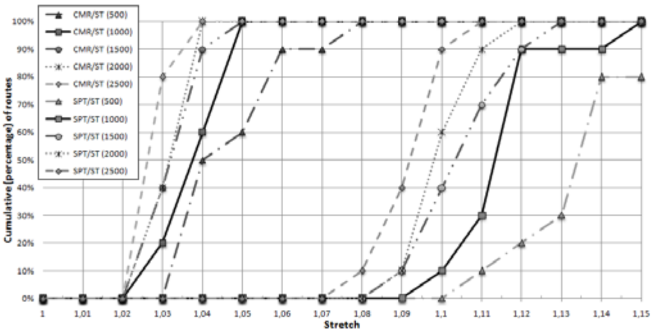

**Fig. 4 – Cumulative percentage of multicast routes as a function of the stretch evolution.**

Another interesting observation is obtained by measuring the cumulative percentage of multicast routes in function of the stretch evolution. In Fig.4, at least 50% of the multicast routes created by the CMR have a stretch lower than the minimum stretch (1.04) reached for all the multicast group sizes. Except for the group size of 500 which has a maximum stretch of 1.08, the other group sizes lead to a maximum stretch less than 1.05. As the multicast group size increases, the percentage of routing paths of lower stretch also increases. Compared to the SPT (right-hand side of Fig.4), for group sizes of 500 nodes, only 10% of the routing paths have a stretch equal or less than 1.11. For group sizes of 2500, only 10% of the multicast routes have a stretch equal or less than 1.08. All point-to-multipoint routing paths produced by the CMR lead to a maximum stretch of 1.08 independently of the multicast group size.

### *B. Routing Table Size*

The routing table (RT) comprises the MRIB, the TIB entries as well as the unicast RIB entries for the SPT scheme that relies on the underlying unicast routing topology. Each RT entry must be encoded using a proper data structure, helping to derive its size (number of bits). For instance, let us consider an interface encoded over 32 bits, an address over 32 bits, an AS over 16 bits (as an AS's path being defined as a sequence of AS's) and cost/distance metric over 16 bits [6]. From Table I, the CMR algorithm shows outstanding performance in terms of the total number of RT entries it produces.

**Table I – Number of RT entries for SPT, ST, and CMR with respect to the multicast group size.**

| | Multicast Group Size | | | | |
|---|---|---|---|---|---|
| | 500 | 1000 | 1500 | 2000 | 2500 |
| SPT | 82,393 | 83,656 | 84,837 | 85,955 | 87,036 |
| ST | 11,354 | 12,504 | 13,587 | 14,626 | 15,642 |
| CMR | 1,416 | 2,596 | 3,707 | 4,770 | 5,805 |

The highest number of RT entries obtained for a multicast group size of 2500 (5,805 entries) is 2.8 times smaller than the number of RT entries produced by the ST algorithm (15,642 entries) and 15 times smaller than the number of the RT entries for the SPT algorithm (87,036 entries). Fig.6 illustrates the relative gain in terms of the total number of RT entries produced by the CMR against the ST and SPT algorithms. An increasing gain as the multicast group size decreases can be observed. Moreover, as the size of the multicast group increases, both CMR and ST algorithms show a similar growing trend compared to the SPT algorithm.

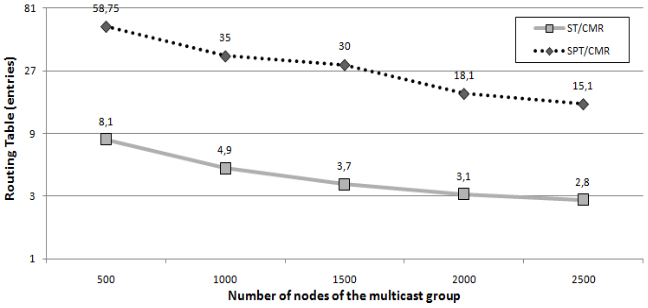

**Fig. 6 - RT size ratio (in terms of number of RT entries) as function of the multicast group size.**

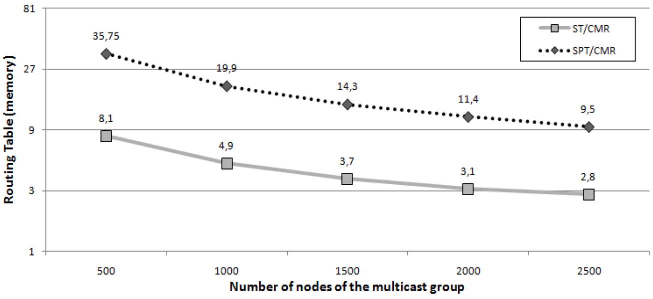

**Fig. 7 – RT size ratio (in terms of memory-bits) as function of the multicast group size.**

Fig.7 depicts the relative gain in terms of the memory-bit space consumed by the total number of RT entries produced by the CMR algorithm against the total number of RT entries produced by the ST and SPT algorithms. As it can be observed, the relative memory gain of the CMR compared to the ST algorithm is never lower than 2.8 (for a multicast group size of 2500) and reaches a maximum of 8.1 as the multicast group size decreases to 500. The same trend is observed when comparing the CMR to the SPT algorithm, the relative memory gain ranges from 9.5 (for a group size of 2500) up to 35.75 (for a group size of 500). Despite of its better communication cost performance (as detailed in Section IV.C), the memory-space consumed and the number of RT entries produced by the SPT algorithm grows exponentially with the size of the multicast group. For the CMR, the curve grows sub-linearly: as the size of the multicast group increases the increment in number of RT entries becomes smaller.

### C. *Communication Cost*

The communication cost is a crucial metric to determine the applicability of the proposed algorithm to power-law topologies comprising of the order of 10k nodes. The two-stage search procedure presented in Section III plays an important role in mitigating this cost. As depicted in Fig. 8, the communication cost ratio for the CMR is relatively high compared to the SPT even if much lower than the communication cost implied by the ST. This observation can be explained by the presence of high degree nodes (nodes that have a degree of the order to 100 or even higher) in power law graphs. However, this communication cost does not take into account for the evolution of the routing topology. This evolution impacts multicast routing algorithms such as the SPT that are strongly dependent on non-local unicast routing information compared to the CMR.

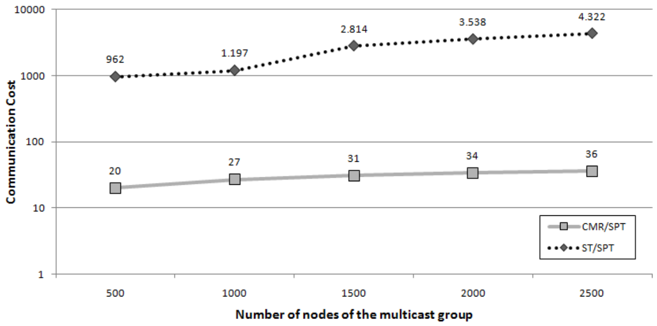

**Fig. 8 – The communication cost ratio as function of the number of MDT nodes covered (multicast group size).**

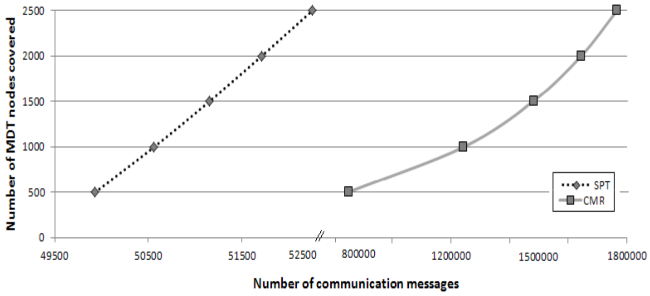

**Fig. 9 – The number of MDT nodes covered (multicast group size) with respect to the number of communication messages.**

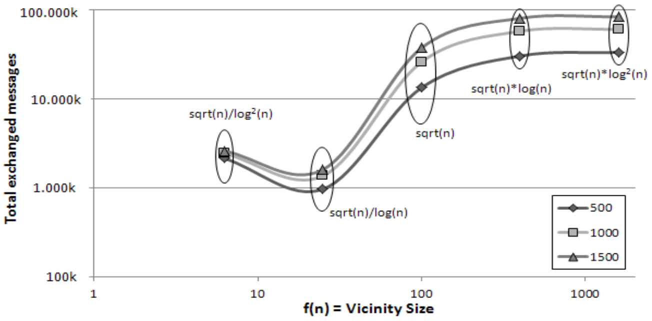

**Fig.10 – Number of exchanged messages according to the Vicinity size (defined by local search stage), n = 10k.**

Between the CMR and the SPT algorithm, the difference of scale in terms of the number of messages exchanged can be observed from the curves of Fig.9. Despite their noticeable difference (maximum of 52,252 SPT-messages vs. 1,765,403 CMR-messages), these curves show that the communication cost for the SPT algorithm grows linearly with the multicast group size whereas the CMR has a concave curve, meaning a sub-linear dependence on the group size. Moreover, as depicted in Fig.8, the communication cost curve for the CMR decreases as the number of nodes composing the multicast group increases. This trend leads us to expect that a saturation level can be reached around a cost ratio not higher than 40 as the multicast group size continues to grow. It is worth mentioning that the memory-space and the processing capacity consumption by communication messages are relatively small. Fig.10, shows that defining the vicinity size proportionally to $n^{0.5}$ / log(n) achieves the minimum number of messages exchanged and thus the minimum communication cost

## V. Conclusion

This paper introduces the first known name-independent compact multicast routing (CMR) algorithm enabling the leaf-initiated, distributed and dynamic construction of MDT. The performance obtained shows substantial gain compared to the ST (minimum factor of 2.8 for multicast group size of 2500, i.e., 25% of the nodes) in terms of the RT entries and memory space required to store them. The stretch deterioration compared to the ST ranges from 8% to 4% (for multicast group size of 500 to 2500, respectively); thus, decreasing with increasing group sizes. The proposed two-phase search process -local search first covering the leaf's node vicinity, and if unsuccessful, a global search over the remaining topology-enables to keep its communication cost within reasonable bounds compared to the reference SPT scheme and sub-linearly proportional to the multicast group size.

Further work will be nevertheless conducted to further decrease the communication cost of the CMR so as to reach this saturation level for smaller multicast group sizes. Another main area of investigation involves the investigation of the CMR performance on real topologies such as the CAIDA Internet topology maps which comprise 16k and 32k nodes.

## References

[1] D. Peleg and E. Upfall, “A trade-off between space and efficiency for routing tables,” *J. ACM*, vol. 36, no. 3, pp. 510–530, Jul. 1989.

[2] B. Awerbuch, A. Bar-Noy, N. Linial, D. Peleg, “Compact distributed data structures for adaptive routing,” *Proc. 21st annual ACM STOC'89*, Seattle, WA, United States, pp. 479–489, May 1989.

[3] M. Thorup, and U. Zwick, “Compact routing schemes,” *Proc. 13th Annual ACM SPAA'01*, Heraklion, Crete, Greece, pp. 1–10, Jul. 2001.

[4] I. Abraham, C. Gavoille, D. Malkhi, N. Nisan, M. Thorup, “Compact name-independent routing with minimum stretch,” *ACM Trans. Alg.*, vol. 4, no. 3, art. 37, Jun. 2008.

[5] I. Abraham, D. Malkhi, D. Ratajczak, “Compact multicast routing,” *Proc. 23rd Int. Symp. DISC'09*, Elche, Spain, pp.364–378, Sep. 2009.

[6] B. Fenner, M. Handley, H. Holbrook, I. Kouvelas, “Protocol Independent Multicast - Sparse Mode (PIM-SM),” *Internet Engineering Task Force (IETF)*, RFC 4601, Aug. 2006.

[7] P. Pedroso, D. Papadimitriou, D. Careglio, “A name-independent compact multicast routing algorithm”, available as Technical Report, UPC-DAC-RR-CBA-2011-15, March 2011

[8] C. Magnien, M. Latapy, M. Habib, “Fast computation of empirically tight bounds for the diameter of massive graphs,” *J. Exper. Alg.*, vol. 13, art. 10, Feb. 2009.

[9] T. Bu, D. Towsley, “On distinguishing between Internet power law topology generators,” *Proc. IEEE Infocom'02*, pp. 638–647, New York, NY, USA, Jun. 2002.

[10] Sage's Graph Library. Available at http://www.sagemath.org/.