Performance analysis of multicast routing algorithms

D.Papadimitriou, D.Careglio and P.Demeester, Fellow, IEEE

***Abstract*—This paper provides a theoretical performance analysis of different classes of multicast routing algorithms, namely the Shortest Path Tree, the Steiner Tree, compact routing and greedy routing. Our motivation is to determine the routing scheme which would yield the best trade-off between the stretch of the multicast routing paths, the memory space required to store the routing information and routing table as well as the communication cost. For this purpose, we also confront these results to those obtained by simulation on the CAIDA map of the Internet topology comprising 32k nodes.**

***Index Terms*—multicast, routing, algorithm, performance,**

I. Introduction

With the increase of multimedia streaming/content traffic, 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 routing received main attention from the research community remain mostly unaddressed since so far. Indeed, unicast routing dependent schemes (either distance vector-based such as the Distance-Vector Multicast Routing Protocol (DVMRP), or link state-based such as Multicast Open Shortest Path First (MOSPF)) have been supplanted by schemes performing independently from the underlying unicast routing, e.g., Protocol Independent Multicast (PIM) [1]. During the last decade, the single-source variant of PIM, referred to as PIM-SSM, has been deployed in the context of IPTV within Internet Service Provider's (ISP) network (intra-domain multicast). However inter-domain multicast has failed to be widely adopted by most ISPs. The main reasons stem from i) memory scaling when overlaying multicast routing on top of unicast (shortest-path) routing with the addition of a level of indirection, ii) the inter-domain discovery process which prevents shared trees between different domains (thus, defeats the objectives of PIM), and iii) its address space structure (Class-D IP addresses) which requires both hardware and software routers upgrade whereas the corresponding cost cannot be compensated by multicast service revenues when the ISP does not itself provide access to multicast receivers (or sources). Further analysis on current IP multicast routing limits and reasons for its lack of wide-scale deployment in the Internet can be found in [2].

In this context, research efforts dedicated to new multicast routing algorithms have been conducted to move beyond the trade-offs between shared trees and shortest path trees. The objective of this paper is to determine the performance bounds and the best trade-offs one could potentially achieve between the stretch of the routing scheme, the memory space consumed to locally store the routing information (including routing tables) and the communication cost of dynamic multicast routing schemes. The comparative performance analysis is performed against i) two well-known reference algorithms (the Shortest Path Tree and the Steiner Tree), ii) compact multicast routing as developed in the seminal paper of Abraham et al. [3] and the greedy multicast routing scheme recently proposed in [4].

This paper is organized as follows. Section II documents prior work in terms of performance evaluation of multicast routing algorithm and the contribution of this paper. In Section III, we provide an overview of the multicast routing algorithms considered in our study. Section IV details the results of our performance analysis and comparative study in terms of the stretch of multicast routing paths they produce, the memory space they consume, and their communication cost. Finally, Section V concludes this paper.

II. Prior Work and Our Contribution

A. Preliminaries

Consider a network topology modeled by an undirected weighted graph $G = (V, E, \omega)$ where, the set V, $|V| = n$, represents the finite set of vertices or nodes (all being multicast capable), the set $E, |E| = m$, represents the finite set of edges or links, and ω is a non-negative function $\omega: E \rightarrow \mathbb{R}^+$ which associates a non-negative weight or cost $\omega(u, v)$ to each edge $(u, v) \in E$. For $u, v \in V$, the path $p(u, v)$ from vertex u to v is defined as the vertex sequence $[x_0(= u), x_1, \ldots, x_{i-1}, x_i, \ldots, x_p(= v)]$ such that the vertices x_i are all distinct and vertex x_{i-1} is adjacent to x_i, $\forall (x_{i-1}, x_i)_{i=1,\ldots,p} \in E$. Distinction is made between the cost $c(u, v)$ of a path $p(u, v)$ defined as the sum of the weights of the edges on the path from u to v and the length $\ell(u, v)$ of a path $p(u, v)$ which denotes the number of edges the path traverses from u to v. The distance $d(u, v)$ between two vertices u, v of the graph G denotes the length/cost of a shortest/minimum cost path $p(u, v)$ from u to v. The diameter $\delta(G)$ of the graph G is defined as the largest distance between any two vertices $u, v \in V$, i.e., $\delta(G) = max_{u,v \in V}\{d(u, v)\}$.

Let $S, S \subset V$, be the finite set of multicast source nodes and $s \in S$ denote a multicast source node. Let $D, D \subseteq V \backslash S$, denote the finite set of all possible destination nodes that can join a multicast source s and let $d \in D$ denote a destination (or leaf) node. A multicast distribution tree $T_{s,M} = (V_T, E_T)$ is defined as an acyclic connected sub-graph H of G, i.e., a tree rooted at the multicast source node $s \in S$ with leaf node set $M, M \subsetneq D$. The tree $T_{s,M}$ is also referred to as the multicast routing path.

Paper submitted for review on July 21, 2013.

Dimitri Papadimitriou is with Alcatel-Lucent, Antwerp, Belgium (e-mail: dimitri.papadimitriou@alcatel-lucent.com).

Davide Careglio is with the Technical University of Catalonia (UPC), Barcelona, Spain (e-mail: careglio@ac.upc.edu)

Piet Demeester is with the Ghent University and iMinds both located in Gent, Belgium (e-mail: piet.demeester@intec.ugent.be).

The set M corresponds to the current set of nodes at a given construction step of the multicast distribution tree (MDT). The size of the tree $T_{s,M}$ is defined as the size of the connected sub-graph of G, i.e., $|T_{s,M}| = h \leq n$.

B. Prior Work

Prior work on compact multicast routing is, as far as our knowledge goes, mainly concentrated around the routing schemes proposed in the seminal paper authored by Abraham et al. in 2009 [3]. The (universal) compact multicast routing schemes developed in this paper follows an extensive theoretic performance justification and analysis. However, Abraham et al. do not report any numerical analysis whereas upper bounds do not necessarily translate actual performance that can be obtained on graphs underlying large-scale networks such as the Internet. The latter shows properties associated to scale-free graphs (small diameter, high-clustering, and power-law degree distribution) as reported by many studies [8], [9], [10], [11].

On the other hand, the greedy compact multicast routing scheme recently proposed in [4] includes extensive simulation on 16k node topologies. However, this paper provides limited theoretical analysis for the performance bounds of the multicast routing scheme it introduces. In particular, since so far, there was no formal proof that the scheme developed in [4] actually meets the conditions for being qualified as a compact scheme. These conditions are the following i) the stretch of the routing scheme is ideally bound by a constant (it does not grow with the network size), ii) the memory space (in terms of number of bits) required to locally store the routing information scales sub-linearly in the number of nodes n, and iii) node names/labels and header sizes scales (poly-)logarithmically.

Hence, on one hand, we have a detailed theoretical performance analysis and on the other hand, a numerical performance analysis obtained by numerical simulation. In these conditions, theoretic performance comparison limits to worst case analysis whereas numeric results do not easily compare to worst case conditions and provide little theoretic foundation.

C. Our Contribution

In this paper, we close this gap by theoretically analyzing and comparing the performance of two reference multicast routing algorithms (the shortest-path tree and the Steiner tree), compact multicast routing as proposed in the seminal paper of Abraham et al. [3] and the greedy multicast routing scheme recently proposed in [4]. We compare the obtained results in order to determine the routing scheme which would yield the best trade-off between the stretch of the multicast routing paths, the memory space required to store the routing information and routing table as well as the communication cost. We also confront these results to those obtained by simulation on the CAIDA map of the Internet topology comprising 32k nodes as of Jan.2011 [12].

For this purpose, our performance analysis includes the following metrics:

- The *stretch* of the multicast routing scheme is defined as the total cost of the edges of the MDT (as produced by the routing algorithm) to reach a given set of leaf nodes divided by the cost of the minimum Steiner tree for the same leaf set. Note that this definition differs from the one used for (unicast) routing schemes. For the latter, the stretch is defined as the maximum cost of the produced routing path $p(u,v)$ over all node pairs $u,v \in V$ divided by the cost of the corresponding shortest (topological) path.
- The *memory space* (in bits) required at each node to locally store i) the information locally processed by the routing scheme to produce the routing table (RT) entries and ii) the produced RT entries.
- The *communication cost* (also referred to as the message cost) is defined as the number of messages exchanged to build the MDT. This metric is directly related to the leaf join time, i.e., the higher the message cost the longer the time needed for a leaf to join the tree.

We also define the adaptation cost as the number of multicast routing states changes resulting from MDT changes due to arbitrary join-leave sequences or topology changes.

III. Multicast Routing Schemes

To conduct our performance analysis and comparative study, we consider the following routing schemes:

A. Shortest Path Tree

The multicast routing path is constructed as a Shortest Path Tree (SPT) from the information exchanges by means of a loop-avoidance path-vector routing protocol carrying the identifier of the multicast source s and the information of the routing path to reach that source. Without routing policy, this routing path is the shortest path from each receiver to the source s. The SPT algorithm provides the lower bound for the (join) communication cost. Each node keeps the following entries in its local routing table i) an entry per neighbor node to exchange routing messages, ii) an entry per selected path to the multicast source s (derived from the unicast routing table), and iii) a multicast routing entry per source s. This multicast routing scheme corresponds to the currently deployed PIM routing performing on top of a unicast routing protocol such as Border Gateway Protocol (BGP).

B. Steiner Tree

Following the definition of the multicast routing stretch (see Section II), the Steiner Tree (ST) algorithm provides the lower bound in terms of stretch. In order to obtain the near optimal solution for the ST algorithm, we consider a ST-Integer Linear Programming (ILP) formulation. For this purpose, we adapt the formulation provided in [5] for bi-directional graphs. The communication cost for the ST measures at each step of the MDT construction the number of messages originated by the nodes part of the MDT. These messages contain the minimal information for remote nodes not belonging to the MDT to join it. Using this information, each node builds and stores in its routing table a routing entry to reach the closest node that belongs to the MDT. Thus, although the ST computation is processed centrally, the communication cost accounts for the total number of messages exchanged during the MDT building process as an equivalent distributed scenario would perform.

C. Compact Multicast Routing

Compact unicast routing aims at finding 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 seminal paper of Peleg and Upfall [6]. Since the late 1980's, various compact routing schemes have been designed in accordance to the distinction between labeled schemes (where nodes are named by polylogarithmic size labels encoding topological information) and name-independent schemes (where node names are topologically independent). Abraham et al. [3] have recently introduced a dynamic and name-independent compact multicast routing algorithm. This scheme referred in the context of this paper to as Abraham Compact Multicast Routing (ACMR) enables the construction of multicast routing paths from any source to any set of destination nodes (or leaf nodes).

The ACMR scheme is i) name-independent, ii) leaf-initiated since join requests are initiated by the leaf nodes but it requires the prior local dissemination of the node set already part of the MDT or keeping dedicated center nodes informed about the nodes that have already joined the MDT, iii) dynamic since requests can be processed on-line as they arrive without re-computing and/or re-building the MDT, iv) (partially) centralized since it requires tree routing information processing by the root of the MDT (i.e., the multicast source node) for each join request after their processing (i.e., mapping) by pre-determined center nodes, and v) dependent of an underlying sparse tree cover grown from a set of center nodes (which induce node specialization driving the routing functionality). It is important to emphasize that the sparse tree cover underlying the ACMR scheme is constructed off-line and requires global knowledge of the network topology to properly operate.

D. Greedy Multicast Routing

The Greedy Compact Multicast Routing (GCMR) scheme proposed in [4] enables the leaf-initiated construction of multicast routing paths from any source to any set of leaf nodes. This scheme aims at minimizing the routing table size (thus the memory space) of each node at the expense of i) multicast routing paths with relative small deviation compared to the optimal stretch obtained by the ST algorithm, and ii) higher communication cost compared to the SPT algorithm. This algorithm minimizes the storage of routing information by requiring only direct neighbor-related information obtained locally and proportionally to the node degree. Thus, it doesn’t rely on the knowledge of non-local topology/path information (as it is the case for the SPT) or requiring the construction of global structures such as sparse covers (as it is the case for the ACMR scheme) or tree structures (as it is the case for the ST). In other terms, it only requires maintenance of local routing information while providing the next hop along the least cost branching path during the MDT construction. The challenge consists thus in limiting the communication cost, i.e., the number of messages exchanged during the search phase, while keeping the best possible stretch-memory space tradeoff.

During the MDT construction, the routing information needed to reach a given multicast source s is acquired by means of an incremental two-stage search process. This process, triggered when any node $u \in V,\ u \notin T_{s,M}$ decides to join a given multicast source s, starts with a local search covering the leaf node's neighborhood. The latter is also referred to as the vicinity ball $B(u)$ of node u. The rationale is the following: the probability of finding a node $v \neq u$, $v \in T_{s,M}$ within a few hops distance from the joining node u is high in large graphs whose diameter $\delta(G)$ is logarithmically proportional to its number of nodes n, i.e., $\delta(G) \sim log(n)$. Moreover, this probability increases with the size of the MDT. If the local search performed over the joining node's vicinity ball $B(u)$ is unsuccessful, the search process is then continued over the remaining unexplored topology without requiring global knowledge of the current MDT. For this purpose, a variable path budget π is used to limit the distance travelled by leaf initiated requests in order to prevent costly (in terms of messaging) global search. In both searching phases, the returned information provides the upstream neighbor node along the least cost branching path to the MDT rooted at the selected multicast source node s. When reaching the joining node, this information enables selection of the least cost branching to the MDT. The routing table of each node $v \in T_{s,M}$ includes consequently the following entries i) one entry that indicates the upstream neighbor node to which the join message is sent for each multicast source s and ii) one entry to enable routing of incoming multicast traffic (originated by that source s) from its incoming port to a set of outgoing ports.

The GCMR scheme is i) name-independent, ii) leaf-initiated; however, compared to the ACMR scheme it operates without requiring prior local dissemination of the node set already part of the MDT or keeping specialized nodes informed about nodes that have joined the MDT, iii) dynamic, iv) distributed since transit nodes process homogeneously the incoming requests to derive the least cost branching path to the MDT without requiring any centralized or specialized processing by pre-determined or dedicated nodes, and v) independent of any underlying topology construction, and performing in absence of an underlying unicast routing topology since the local knowledge of the cost to direct neighbor nodes is sufficient for the GCMR scheme to properly operate.

IV. Performance Analysis

In this section, we analyze and compare the performance of the multicast routing schemes introduced in Section III for join only events to the multicast source (but no leave events). This case is appropriate for settings where once a node joins an MDT it will not leave it until the multicast session ends.

A. Stretch

1) ACMR

The stretch of the ACMR scheme as determined by the Lemma.7 of [3] is $O(min\{log(n), log(\Delta)\}.log(n))$ competitive compared to the stretch of the ST algorithm. The quantity Δ called the aspect ratio of the graph G is defined as the ratio between the maximum distance $max\ d(u,v)$ and the maximum distance $\min d(u,v)$ for any node pair $u,v \in V$ (see [3]). Note that when the minimum distance is equal to 1, then the aspect ratio Δ corresponds to the diameter $\delta(G)$ of the graph G.

Using the CAIDA maps of the Internet topology comprising 16k (Jan.2004) and 32k nodes (Jan.2011), the measured ratio $\Delta = \delta(G) \simeq 10$. These results are confirmed by the systematic routing path length measurements documented in [7]. Consequently the stretch of the ACMR scheme is $O(log(n))$. Note here that compared to other studies, the present paper makes a clear distinction between the average path length and the diameter of the graph (i.e., the maximum path length). Moreover, since the diameter of the unweighted graph underlying the Internet topology is of the order of $log(n)$, the stretch upper bound of the ACMR scheme is $O(\delta(G))$.

2) GCMR

For unweighted (weighted) graphs, the stretch of the GCMR scheme is determined by Lemma 1 (respectively, Lemma 2).

Lemma_1: The stretch upper bound of the GCMR scheme is $O(\frac{\delta(G)\,+\,1}{2})$.

Proof: Assume that nodes $v, w \in T_{s,M}$ and are respectively at distance $d(s,v)$ and $d(s,w)$ from the multicast source s such that $d(s,v) > d(s,w)$. Assume also that node u decides to join the multicast tree $T_{s,M}$ by appropriate setting of its path budget $\pi(u)$.

If the following inequality is verified

$$min_i\{d(u,v_i) \mid d(u,v_i) < \pi(u) \land v_i \in T_{s,M} \land v_i \in p(s,v)\} < min_j\{d(u,w_j) \mid d(u,w_j) < \pi(u) \land w_j \in T_{s,M} \land w_j \in p(s,w)\}$$

then, node u will subsequently select the shortest branching path to the node $v_i^* = min_i\{d(u,v_i) \mid d(u,v_i) < \pi(u) \land v_i \in T_{s,M} \land v_i \in p(s,v)\}$. From this routing decision, the increase of the multicast routing path stretch is given by the formula $d(u,v_i^*) + d(v_i^*,s)$. Moreover, resulting from the path budget constraint ($d(u,v_i^*) < \pi(u)$), the upper bound of the stretch increase is determined by $d(u,v_i^*) + d(v_i^*,s) < \pi(u) + d(v_i^*,s)$. Since $d(v_i^*,s) < d(v,s)$ (otherwise node v_i^* could not be selected at all), we obtain the following upper bound to the stretch increase: $\pi(u) + d(v,s)$. Moreover, by replacing the source s by any node $x \in T_{s,M}$ we can generalize this upper bound to $\pi(u) + d(v,x) > \pi(u) + d(v_i^*,x) > d(u,v_i^*) + d(v_i^*,x)$.

On the other hand, assume that there exists a node $w_j^* \in T_{s,M}$ and $w_j^* \in p(s,w)$ such that i) $d(w_j^*,s) < d(v_i^*,s)$ and ii) $d(u,w_j^*) + d(w_j^*,s) < d(u,v_i^*) + d(v_i^*,s)$ whilst node w_j^* is not reachable by node u due to the path budget constraint, i.e., $d(u,w_j^*) > \pi(u)$. Then the routing path stretch would increase by $d(u,w_j^*) + d(w_j^*,s)$. This increase is minimum when $d(w_j^*,s)$ is minimum, i.e., when $w_j^* = min_j\{d(w_j,s)\}$. Since by construction $d(u,w_j^*) > \pi(u)$, we could have obtained as result of the selection of node w_j^* the following lower bound to the stretch increase: $\pi(u) + min_j\{d(w_j^*,s)\}$. Moreover, by replacing the source s by any node $y \in T_{s,M}$ we can generalize this lower bound to $d(u,v_i^*) + d(v_i^*,y) > d(u,w_j^*) + d(w_j^*,y) > \pi(u) + min_j\{d(w_j^*,y)\}$.

As the maximum (minimum) distance is given by $\delta(G)$ (1, respectively) and the maximum (minimum) path budget is set to $\delta(G)$ (1, respectively), the stretch of the GCMR scheme is $O\left(\max\{\frac{\delta(G)\,+\,1}{2}, \frac{2\delta(G)}{\delta(G)\,+\,1}\}\right) = O(\frac{\delta(G)\,+\,1}{2})$. □

Lemma_2: the stretch increase of the GCMR scheme is dominated by the sum (over all join events) of the ratio between the minimum distances $min_v\{d(u_i,v) | v \in B(u_i) \land v \in T_{s,M}\}$ and $min_w\{d(u_i,w) | w \notin B(u_i) \land w \in T_{s,M}\}$ such that $min_w\{d(u_i,w)\} < min_v\{d(u_i,v)\}$.

Proof: Let $G = (V,E,\omega)$ be a weighted undirected graph; two cases can occur when considering a joining node u_i depending on whether the multicast source s belongs or not to the vicinity ball of the joining node u_i:

i) If the multicast source $s \in B(u_i)$, then local search initiated by node u_i will find the least cost branching path to the tree $T_{s,M}$. This condition is verified when the path budget $\pi(u_i)$ value is sufficient for the request message to reach the source node s from the joining node u_i. It is obvious to see that when this condition is met, the resulting stretch increase is minimal.

ii) If the multicast source $s \notin B(u_i)$, then the following alternative can occur:

ii.a) If $\exists$ node v such that $v \in T_{s,M}$ and $v \in B(u_i)$, then the local search process initiated by the joining node u_i will find the actual least cost branching path if and only if there no other node $w \in T_{s,M}$ and $w \notin B(u_i)$ that can be found at shorter distance, i.e., $d(u_i,w) < d(u_i,v)$. Indeed, the distance limit set by the joining node u_i on the local search process by means of the path budget $\pi(u_i)$, allows (if it exists) to reach a node v such that $v \in T_{s,M}$ and $v \in B(u_i)$; thus, before triggering the global search process. However, due to the finite size of the ball $B(u_i) < \sqrt{n}$ (see [4]), when decrementing the path budget $\pi(u_i)$ such node v can be found during the local search phase even though $\exists$ node $w \in T_{s,M}$ and $w \notin B(u_i)$ such that $d(u_i,w) < min_v\{d(u_i,v)\}$ over all node v such that $v \in B(u_i)$ and $v \in T_{s,M}$. Hence, the stretch increase is bound by the fraction of such nodes conditioned by the current number of nodes already belonging to the tree $T_{s,M}$. The stretch increase can thus be derived from the following formula:

$$\sum_{i=1}^{M'\subseteq M} \frac{min_j\{d(u_i,v_j) \mid v_j \in B(u_i) \land v_j \in T_{s,M}\}}{min_k\{d(u_i,w_k) \mid w_k \notin B(u_i) \land w_k \in T_{s,M}\}} \tag{1}$$

ii.b) If $\nexists$ node v such that $v \in B(u_i)$ and $v \in T_{s,M}$, then the global search process initiated by node u_i will find the least cost branching path. This condition is verified if the path budget $\pi(u_i)$ value is sufficient for the request message to reach node $v \in T_{s,M}$ from the joining node u_i. When this condition is met, the resulting stretch increase is minimal. □

3) Comparative Analysis

The stretch upper bound of the multicast routing paths produced by the ACMR scheme even if universal (i.e., applicable to any graph) is 2 times higher than the one produced by the GCMR scheme. It also important to note that the stretch of the GCMR scheme has a second order dependence on the network size (due to its dependence on the diameter $\delta(G)$). On the other hand, the ACMR scheme shows a first and a second order dependence on the network size (due to its dependence on the number of nodes n and the diameter $\delta(G)$ and the number of nodes n).

Fig.1 depicts the routing scheme stretch obtained by simulation of the ST, the SPT, the GCMR and the ACMR scheme (for different values of the parameter k). The simulations are performed on the CAIDA map of the Internet topology comprising 32k nodes. The scenario executed simulates the construction of multicast routing paths for leaf node set of increasing size from 500 to 4000 nodes with increment of 500 nodes. Each execution is performed 10 times by considering 10 different multicast sources. From this figure, we can observe that the upper bound for the ACMR scheme is not reached (its maximum value reaches 2.15 for $k = 1.5$). Moreover, the stretch of the GCMR scheme is in average still twice better than the stretch of the ACMR scheme with a maximum value of 1.08 (for 500 leaf nodes) and a minimum value of 1.03 (for 4000 leaf nodes). Note also that the comparative gain is weakly influenced by the value of the parameter k. This parameter characterizes the sparse tree cover construction: the higher the value of k, the lesser the number of trees in the sparse tree cover (TC).

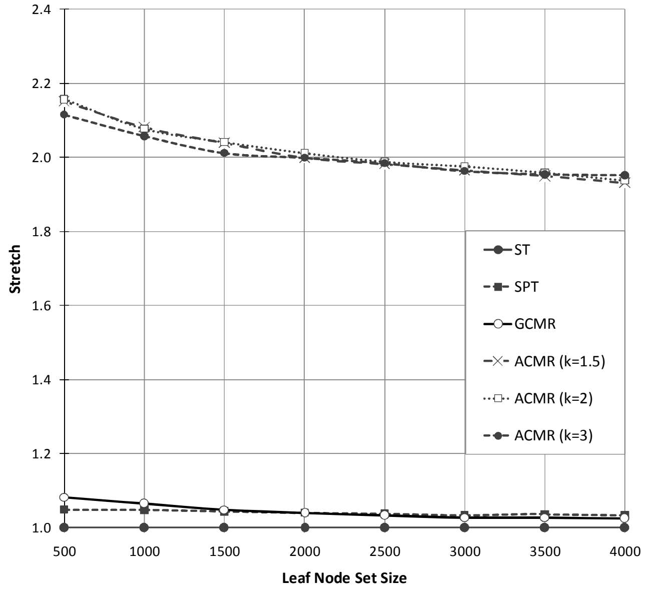

Fig.1: Stretch in function of the leaf node set size

B. Memory

1) ACMR

The memory space consumed by the ACMR scheme as documented in Section 6.1 of [3] comprises the space required to store the following routing information:

1) Each node $v \in V$ stores the tree routing information $\tau(T, v)$ for all the trees T in its own label $SPLabel(v)$[1], which yields a total memory of $O(log^3(n).log(\Delta)/log(log(n)))$.
2) For each radius $r \in R = \{0,1, \ldots, log(\Delta)\}$ and tree T belonging to the sparse tree cover $TC_{k,2^r}(G)$, the center node $c(T_i(v))$ of node $v \in T$ stores the label of all nodes contained in the ball $B(v, 2^r)$, which leads to a total memory over all $|R|$ radii of $O(kn^{1+1/k}log(\Delta))$ bits.
3) Each node $v \in V$ stores $O(log(\Delta))$ labels of size $\tilde{O}(kn^{1/k})$ to reach the center nodes $c(T_i(v))$ for all radii $r \in R = \{0,1, \ldots, log(\Delta)\}$, which leads to a total memory of $O(kn^{1+1/k}log(\Delta))$.

Thus, the ACMR scheme consumes in total $\tilde{O}(kn^{1+1/k})$ bits. As the value of the parameter k ranges in the interval $[1, log(n)]$, we obtain respectively as upper bounds $\tilde{O}(n^2)$ and $\tilde{O}(n^{1+1/\log(n)})$. Note that the memory consumption of the ACMR scheme is independent of the MDT size.

2) GCMR

Per multicast source s, each node $v \in T_{s,M}$ stores in its local routing table one entry to the selected upstream node and one multicast routing entry. The memory-bit space consumed by the multicast routing entry, which indicates the outgoing ports for the incoming multicast traffic is proportional to the local tree out-degree d_k. Assuming an optimal port identifier encoding proportional to $log(n)$ at each node, the total memory space consumed by the MDT constructed by means of the GCMR scheme is $O(h\ log(n))$, where h is the size of the MDT. The latter equals n when the MDT covers the entire network.

3) Comparative Analysis

Depending on the value of the parameter k, the GCMR scheme (for $h = n$) is $\tilde{O}(n)$ competitive for $k = 1$ and $\tilde{O}(n^{1/log(n)})$ competitive for $k = log(n)$ compared to the ACMR scheme. The main difference between them consists in that the GCMR scheme depends explicitly on the MDT size whereas the ACMR scheme depends on the network size.

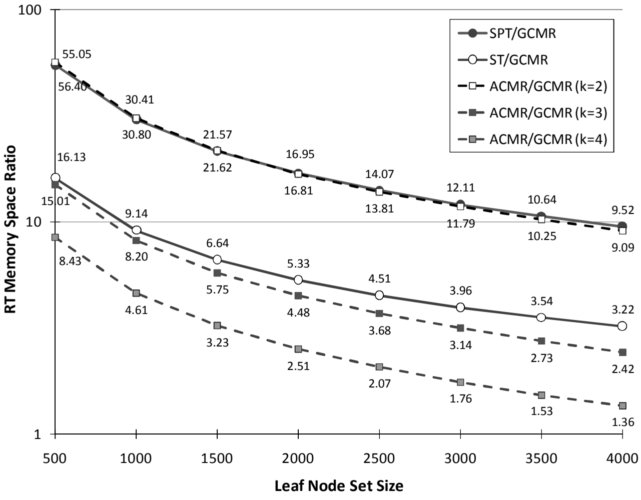

Fig.2: Memory space consumption ratio in function of the leaf node set size

Fig.2 depicts the memory consumption ratio of the ST, the SPT and the ACMR scheme (for different values of the parameter k) against the GCMR scheme. This ratio provides a good indication of the achievable reduction in terms of the memory space required to store the routing information and routing table entries produced by these algorithms. The results are obtained by means of simulation on the CAIDA map of the Internet topology comprising 32k nodes. The scenario executed simulates the construction of multicast routing paths for leaf node set of increasing size from 500 to 4000 nodes with increment of 500 nodes. Each execution is performed 10 times by considering 10 different multicast sources.

From Fig.2, we can observe that for a leaf set of 500 nodes the memory space consumption ratio between the ACMR and the GCMR scheme decreases from 56,40 (for $k = 2$) to 8,43 (for $k = 4$). This ratio decreases as the size of the leaf node set increases. When the size of the leaf set reaches 4000 nodes, this ratio drops to 9.09 (for $k = 2$) and 1.36 (for $k = 4$). These results confirm that the gain in memory space consumption obtained with the GCMR scheme decreases against the ACMR scheme as the size of the MDT increases. The dependency of this gain with respect to the parameter k finds its origin in the underlying sparse tree cover construction that the ACMR scheme requires: the higher the value of the parameter k, the sparser the tree cover. As the value of this parameter increases to its maximal value $log(n) \sim \delta(g)$ and the size of the leaf node set increases to n, the gain in memory space consumption tends to 1. However, this situation is

[1] The label SPLabel(v) stores the label λ(T,c(T)) given by Lemma 9 of [3] for each tree T part of the sparse tree covers containing node v.

unlikely to occur in practice as it would imply that the MDT comprises all network nodes.

C. *Communication Cost*

In order to analyze the communication cost it is important to distinguish between adaptive and oblivious routing. A main property of the ACMR scheme variant documented in Section 6.2 of [3] is the construction of MDTs that are oblivious, i.e., the multicast routing path from the source s to a given leaf node is irrespective of the other leaves. Due to obliviousness, when other nodes join and leave the MDT, this does not affect the multicast routing path to that leaf. In contrast, the GCMR scheme is adaptive, i.e., routing decisions may be modified once there is a change in the information that has lead to that decision. This implies that even if the GCMR scheme is competitive compared to the ACMR scheme, interleaved sequences of join and leave events may increase the message cost. For this purpose, we distinguish between the "join" communication cost from the adaptation cost, i.e., the additional message cost to restore the optimal multicast routing path when nodes that previously joined the tree leave the MDT before the multicast session ends.

1) *Join Communication Cost*

a) ACMR

The total communication cost of the ACMR scheme can be derived from the Lemma.7 of [3]. In case of join-only events, the communication cost is $O(2^{\rho+2}.2|M|.log(\Delta).log(n))$, where |M| is the size of the leaf node set.

Since the exponent ρ is at maximum equal to 1 (following the inequality $\rho \leq \log(\delta(G))$ with $\delta(G) \doteq 10$), we obtain for the total communication cost of the ACMR scheme $O(16|M|.log(\Delta).log(n))$. Moreover, as the minimum distance of the unweighted graph underlying the Internet topology is equal to 1, the aspect ratio Δ corresponds to the diameter $\delta(G)$ of the graph G; hence, we obtain for the total communication cost $O(16|M|.log(n))$.

b) GCMR

In the GCMR scheme, each join event as initiated by a node $u_i \in V$ to reach a node $v \in T_{s,M}$ results in a communication cost equal to:

$$C(u_i) = 2\mu_i X_i + 2m(1 - X_i) \tag{2}$$

In (2), μ_i and m are respectively the number of edges in the vicinity ball $B(u_i)$ of the joining node u_i and the total number of edges $|E|$ in the graph. The Boolean variable $X_i = 1$ when at least one node $v \in T_{s,M}$ is comprised in the vicinity ball $B(u_i)$ of the joining node u_i. Thus, when all the multicast distribution tree nodes $v \in T_{s,M}$ are outside the vicinity ball $B(u_i)$, the communication cost $C(u_i) = 2m$.

The total communication cost, i.e., the cost to build the entire MDT, is thus determined by the sum of the individual communication costs $C(u_i)$ induced by all nodes $i = 1, \ldots, |M|$ joining the multicast tree $T_{s,M}$:

$$C(T_{s,M}) = \sum\nolimits_{\mathrm{i}} [2\mu_i X_i + 2m(1 - X_i)] \tag{3}$$

As already shown in [4], defining a vicinity ball size proportional to $\sqrt{n}/log(n)$ minimizes the number of messages exchanged during the construction of the MDT and thus the communication cost. To further reduce the communication cost of the GCMR scheme, each multicast source s constructs a vicinity ball $B(s)$ whose number of edge is given by μ_s. This vicinity ball shall demonstrate the following properties i) its size at least as large as the average size of leaf node's vicinity ball, and ii) the radius locally computed from its outgoing ports is inversely proportional to the neighbor's node degree. Subsequently, when a request message reaches the boundary nodes of the ball $B(s)$ of the multicast source s, the message is directly routed along the shortest path to the source s. This enhancement prevents searching at the neighborhood of the multicast traffic source. The total communication cost is thus determined by:

$$C(T_{s,M}) = \sum\nolimits_{\mathrm{i}} [2\mu_i X_i + 2(m - \mu_s)(1 - X_i)] \tag{4}$$

2) *Comparative Analysis*

Simulations performed on the CAIDA map of the Internet topology comprising 32k nodes show that the communication cost ratio of the GCMR scheme is relatively high compared to the SPT algorithm. As depicted in Fig.3, the communication cost ratio between the GCMR scheme and the SPT algorithm increases from 2,69 (for leaf set of 500 nodes) to 8,17 (for leaf set of 4000 nodes). The ratio's slope decreases as the leaf node set increases until reaching a saturation level around 10. It is worth mentioning that the memory and the capacity required to process communication messages are relatively limited.

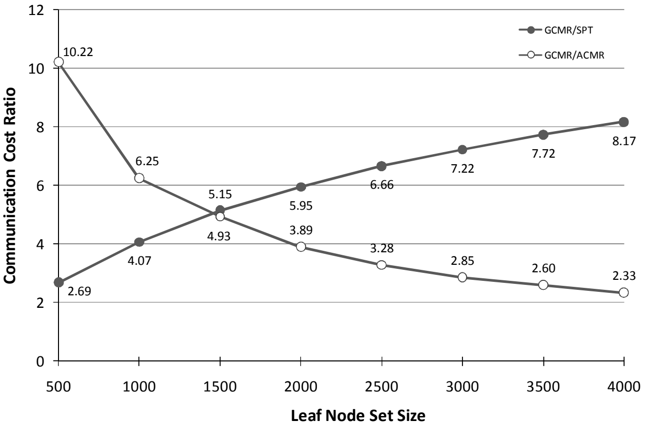

Fig.3: Communication cost ratio in function of the leaf node set size

When comparing the communication cost of the ACMR scheme against the GCMR scheme for the same topology, the opposite trend can be observed from Fig.3. Note here that the communication cost for the ACMR scheme accounts also for the hidden cost associated to the exchange of multicast routing information between joined branching points (for each joining node u_i) and the multicast source node s. The communication cost ratio between the GCMR scheme and the ACMR scheme decreases from 10,22 (for leaf set of 500 nodes) to 2,33 (for leaf set of 4000 nodes). The gain factor observed when decreasing the size of the leaf node set plays in favor of the ACMR scheme and underlines that improvement(s) should be further considered to reduce the join communication cost of the GCMR scheme.

3) *Adaptation Cost*

In order to evaluate the adaptation cost of the GCMR scheme, we are interested in determining the maximum number of re-routing events that this scheme requires to adapt the MDT upon occurrence of leave events. Remember that the

ACMR scheme is oblivious (when nodes leave the MDT, the multicast routing path to the remaining leaf nodes is not affected); hence, there is no additional adaptation cost.

For the GCMR scheme, which is adaptive, the situation completely differs; in particular, when the node $v \in T_{s,M}$ leaves the MDT after an arbitrary sequence of $\sigma\,(1 \leq \sigma < |M|)$ dependent join events, each involving at least one of the nodes along the path $p(v,s)$ from the leaving node v to the multicast source s of the MDT. In this case, a certain number of re-routing events are required to restore the optimal multicast routing path.

Theorem_1: the number of events triggered by a node leaving the MDT after an arbitrary sequence of $\sigma\,(1 \leq \sigma < |M|)$ dependent join events is $O(\sigma.(\delta(G)-1))$.

Proof: consider the node $v \in T_{s,M}$. For each leaf node w_i, let $p(w_i,v_j)$ denote the least cost branching path from node w_i to v_j such that $v_j \in p(v,s)$, where s is the source node of the MDT. Assume also that node u wants to join the tree $T_{s,M}$ knowing that the path $p(u,v)$ is such that $d(u,v) = min_j\{d(u,v_j)\} > d(u,w_i)\ \forall i$. It follows that node u selects the path $p(u,w)$ such that $d(u,w) = min_i\{d(u,w_i)\}$ to join the tree $T_{s,M}$.

If afterwards node w leaves the tree $T_{s,M}$, the following conditions must be verified to trigger a re-routing event:

(1) For a node y along the path $p(u,v)$: $d(y,v) < d(y,u) + d(u,w) + d(w,v)$
Note that if node $y \equiv$ node u, then $d(u,v) < d(u,w) + d(w,v)$

(2) For a node x along the path $p(u,w)$: $d(x,u) + d(u,v) < d(x,w) + d(w,v)$

Moreover, following the triangular inequality, $d(w,v) < d(w,u) + d(u,v)$; otherwise, node w wouldn't have selected node v as branching node. Hence, inequality (2) can be rewritten as $d(x,u) + d(u,v) < d(x,w) + d(w,u) + d(u,v)$; thus, $d(x,u) < d(x,w) + d(w,u)$.

The minimum number of re-routing events is determined by the number of nodes along the path $p(x,y)$ when its distance $d(x,y) = d(x,w) + d(w,v) + d(v,y)$ is minimum. From inequality (1), this minimum distance is equal to 3.

The maximum number of re-routing events is determined by the number of nodes along the path $p(x,y)$ when its distance $d(x,u) + d(u,y)$ is maximum. From inequality (2), the maximum distance verifies the following $d(x,u) + d(u,y) < d(x,w) + d(w,v) - d(y,v)$.As these distances are upper bounded by $\delta(G) - 1$, the following inequality holds $d(x,u) + d(u,y) < \delta(G) - 1$. □

Since the diameter $\delta(G)$ of the unweighted graph G underlying the Internet topology grows proportionally to $log(n)$; the number of re-routing events is limited. Derivation of the corresponding message exchange depends on the aspect ratio of the multicast distribution tree. Further investigation would enable determining the total message cost depending on the aspect ratio of the multicast tree.

V. Conclusion

This paper theoretically analyzes and compares the performance of two reference multicast routing algorithms (the shortest-path tree and the Steiner tree), the compact multicast routing scheme as proposed in the seminal paper of Abraham et al. [3] and the greedy multicast routing scheme recently proposed in [4]. We also confront these results to those obtained by simulation on the CAIDA map of the Internet topology comprising 32k nodes as of Jan.2011. Compared to the ACMR scheme, the GCMR scheme provides a better tradeoff between the memory space it requires to locally store the routing information (including the routing table entries) and the stretch factor increase multicast routing paths it produces. On the other hand, the results obtained for the join communication cost ratio between the ACMR scheme (but also the SPT algorithm) and the GCMR scheme show that further improvement are still required for the latter.

Moreover, the adaptive property of the GCMR scheme should induce a limited number of re-routing events in case of finite sequences of join and leave events compared to the obliviousness property of the ACMR scheme. Future work will determine if these theoretical performance results can be verified by simulation for interleaved sequences of join and leave events but also on non-stationary topologies.

References

[1] B.Fenner, et al., Protocol Independent Multicast - Sparse Mode (PIM-SM), Internet Engineering Task Force (IETF), RFC 4601, Aug.2006.

[2] C.Diot et al., Deployment Issues for the IP Multicast Service and Architecture, IEEE Network, vol.4, no.1, pp.78-88, Jan/Feb.2000.

[3] I.Abraham, D.Malkhi, and D.Ratajczak, Compact multicast routing, Proc. of 23rd Int'l Symposium on Distributed Computing DISC'09, Elche, Spain, pp.364–378, Sep.2009.

[4] P.Pedroso, D.Papadimitriou, D.Careglio, Dynamic compact multicast routing on power-law graphs, 54th IEEE Globecom, Houston (TX), USA, Dec.2011.

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

[6] 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.

[7] B.Huffaker, M.Fomenkov, D.Plummer, D.Moore, and k.claffy, Distance Metrics in the Internet, IEEE Int'l Telecommunications Symposium (ITS), Brazil, pp.200–202, Sep.2002,

[8] F.Chung, L.Lu, The Average Distance in a Random Graph with Given Expected Degrees, Internet Mathematics, vol.1, no.1, pp.91–114, 2003.

[9] D.J.Watts, S.H.Strogatz, Collective dynamics of "small-world" networks, Nature 393, pp.440-442, 1998.

[10] M.Faloutsos, P. Faloutsos, and C.Faloutsos, On power-law relationships of the Internet topology, Proc. ACM SIGCOMM 1999 and in ACM Computer Communication Review, vol.29, pp.251-263, 1999.

[11] M.E.J. Newman, The Structure and Function of Complex Networks, SIAM Reviews, vol.45, no.2, pp.167-256, 2003.

[12] CAIDA Map. Available at http://as-rank.caida.org/data.