Performance Analysis of a Dynamic Compact Multicast Routing Scheme

Dimitri Papadimitriou, Pedro Pedroso, Piet Demeester

Performance Analysis of a Dynamic Compact Multicast Routing Scheme

Dimitri Papadimitriou (1), Pedro Pedroso (2), and Piet Demeester (3)

(1) Alcatel-Lucent Bell, Copernicuslaan 50, 2018 Antwerpen, Belgium, (2) Universitat Politecnica de Catalunya, Barcelona, Spain, (3) Ghent University, Ghent, Belgium

Les algorithmes de routage compact ont pour objectif de trouver le meilleur compromis entre le nombre de bit mémoire nécessaire pour stocker les entrées des tables de routage et le coût des chemins de routage que ces algorithmes produisent. Dans cet article, nous présentons un algorithme de routage compact permettant la construction dynamique de chemins de routage point-à-multipoint pour la distribution de données depuis une source quelconque vers un ensemble quelconque de destinations. L'arbre déterminé par ce chemin et servant à la distribution du trafic est communément appelé arbre de distribution de trafic multicast. Au moyen de cet algorithme, cet arbre est capable d'évoluer dynamiquement en fonction de l'arrivée des demandes de souscription à la source de trafic associée. Nous étudions les compromis entre l'étirement produit par l'algorithme de routage, la taille et le nombre d'entrées résultantes dans les tables de routage ainsi que le coût de la communication.

Keywords: routage compact, point-à-multipoint, dynamique, performance

1. Introduction

Compact unicast routing aims to find the best tradeoff between the memory-space required to store the routing table (RT) 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 [5]. Since then, in accordance to the distinction between labeled (nodes are named by polylogarithmic size labels encoding topological information) and name-independent (node names are named by topologically independent) compact routing schemes have been designed, notably in [7] and [1]. Recently introduced in [2], dynamic compact multicast routing algorithms construct point-to-multipoint routing paths from any source to any set of destinations (or leaves). These paths define multicast traffic distribution tree (MDT) since the algorithm instantiates local routing state so that each MDT node can derive the entries to forward the multicast traffic received from a given source to its leaves.

In this paper, we propose in Section 2 a dynamic compact multicast routing (CMR) algorithm that enables the construction of point-to-multipoint routing paths for the distribution of multicast traffic from any source to any set of leaf nodes. Compared to [2], the proposed name-independent CMR algorithm is i) *leaf-initiated*: the join/leave requests are initiated by the leaf nodes, ii) *distributed*: transit nodes process the join/leave requests and compute the routing table entries (no centralized processing by the root of the MDT), iii) *dynamic*: join/leave requests are processed on-line as they arrive without re-computing and/or re-building the MDT from scratch, and iv) *protocol independent* of any underlying unicast routing topology: the local knowledge of the cost to direct neighbor nodes is sufficient for the proposed algorithm to properly operate. In Section 3, we evaluate the CMR performance by measuring the stretch of the routing paths it produces, the memory space needed to store the resulting RT entries as well as its communication cost, i.e., the number of message exchanged to build the MDT. For this purpose, we simulate the CMR algorithm on synthetic power law graphs comprising 10k nodes that are representative of the Internet topology. Two reference schemes, the Shortest Path Tree (SPT) and the Steiner Tree (ST) algorithm are used to compare the CMR performance over the same topologies. We conclude this paper in Section 4 by analyzing our results against possible improvements left as future work.

2. CMRA Algorithm

The objective of the proposed algorithm is to minimize the routing table (RT) size at each node $v \in V$, $|V| = n$, at the expense of i) routing multicast packets on point-to-multipoint paths with relative small deviation compared to the optimal stretch obtained with the Steiner Tree (ST), and ii) higher communication cost compared to the shortest path tree (SPT). For this purpose, the CMR algorithm reduces the storage of routing information by maintaining during the MDT construction the direct neighbor-related entries per node v, i.e., only the local routing information (degree(v) entries) instead of global routing information (|n-1| entries). The information needed to reach a given multicast source s is acquired by means of a search mechanism that returns the upstream neighbor 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 the MDT, a multicast routing entry is dynamically created and stored in the so-called tree information base (TIB) from which a multicast forwarding entries is derived. The reduction in memory space consumed by the RT of each node results however in higher communication cost compared to the SPT and ST reference algorithms. In order to keep the communication cost as low as possible, we introduce a maximum (dissemination) path budget in each message so as to discard messages with too long and unneeded range.

2.1 Preliminaries

Consider a network topology modeled by an undirected graph **G** = (V,E,c) where the set V, |V| = n, represents the finite set of nodes or vertices (all being multicast capable), the set E, |E| = m, represents the finite set of links or edges, and c a non-negative cost function c: $E \rightarrow Z^+$ that associates a cost c(u,v) to each link (u,v) $\in$ E. For u, v $\in$ V, let c(u,v) denote the cost of the path p(u,v) from u to v in **G**, where the cost of a path is defined as the sum of the costs along its edges. Let S be the finite set of source nodes, S $\subset$ V, and let D be the finite set of all possible destination nodes that can join a multicast group g, D $\subseteq$ V\{S}. Let s $\in$ S, and d $\in$ D denote a source node and a destination (or leaf) node, respectively. A multicast distribution tree $T_{s,M} = (V_T, E_T)$ is defined as a connected sub-graph of **G** without cycles, i.e., a tree rooted at s $\in$ S with leaf node set M, M $\subseteq$ D. The set M corresponds to the current set of nodes at a given construction step of the MDT identified by the tuple multicast source s-multicast group g (<s,g>).

2.2 Description

The multicast distribution tree $T_{s,D}$ is constructed iteratively. At each step ω (ω = 1,2,..,|D|) of the leaf-initiated construction, a randomly selected node u joins $T_{s,M}$, M $\subseteq$ D. If node u is already part of $T_{s,M}$ (u $\in$ V_T) then it is either a transit or a branching node of the MDT. Otherwise, node u is not part of $T_{s,M}$ (u $\in$ D\{V_T}) and it must search for the least cost branching path from node u to node v $\in$ $T_{s,M}$. Among the set $P_{u,v}$ of possible paths p(u,v) from node u $\notin$ $T_{s,M}$ to node v $\in$ $T_{s,M}$, the least cost branching path p(u,v)* = min{c(u,v) | p(u,v) $\in$ $P_{u,v}$}. The cost c(u,v) of the path p(u,v) is defined as the sum of the cost c(u,w) of the edge (u,w), w being the upstream neighbor node of u (or succ(u)) and the cost c(w,v) of the path p(w,v).

```
Type-R message
If port(pred(w),w) in state open then
    Set port(pred(w),w) to state close
    $\pi(w) \leftarrow \pi(pred(w))$ - c((pred(w),w))
    If $\pi(w) > 0$ then
        If w $\in T_{s,M}$ ($u \neq w = v$) then
            Set c(w,v) $\leftarrow$ 0
            Set port(pred(w),w) to state reach
            Send type-A message to pred(w)
        Else
            If $\pi(w)$ - c(w,succ(w)) > 0 then
                If port(w,succ(w)) in state open then
                    Set port(w,succ(w)) to state wait
                    Send type-R message to succ(w)
                    Wait type-A message from succ(w)
                Else
                    Wait type-A message from succ(w)
            Else ($\pi(w)$ - c(w,succ(w)) $\leq$ 0)
                c(w,v) $\leftarrow$ inf
                Set port(pred(w),w) to state unreach
                Send type-A message to pred(w)
    Else ($\pi(w) \leq 0$)
        c(w,v) $\leftarrow$ inf
        Set port(pred(w),w) to state unreach
        Send type-A message to pred(w)
```

```
Type-A message
While port(w,succ(w)) in state wait or $\tau(w) > 0$
    If c(succ(w),v) finite then
        c(w,v) $\leftarrow$ c(w,succ(w)) + c(succ(w),v)
        Set c(w,v)* = min{c(w,v) | p(w,v) $\in P_{w,v}$}
        Set port(w,succ(w*)) to state reach
        Set port(w,succ(w)) to state unreach
    Else (c(succ(w),v) inf)
        Set port(w,succ(w)) to state unreach

If c(w,v)* finite then
    If port(pred(w),w) in state close then
        Set port(pred(w),w) to state reach
        Send type-A message to pred(w)
Else
    If port(pred(w),w) in state close then
        Set port(pred(w),w) to state unreach
        Send type-A message to pred(w)
```

Two types of messages are involved at each step of the construction, namely the request (type-R) messages flowing in the upstream direction towards the multicast source s, and the response (type-A) messages sent in the downstream direction towards the joining leaf node u. Type-R messages comprise the following information i) a sequence number {u_{id}, r_{id}} to prevent duplication of messages, where u_{id} identifies the leaf node u and r_{id} identifies its request to join/leave the multicast source/group pair <s,g>, ii) the leaf node u's timer value $\tau(u)$ that sets the waiting time at intermediates nodes before answering back to the downstream neighbor node, and iii) a path budget π, starting from $\pi(u) = \pi_{max}$, set at leaf node u. The π_{max} value is bound by the graph diameter (the length of the longest shortest path) for which approximation algorithms exist, as well as method for computing a lower and upper bounds [4]. Starting from node u, the path budget π is decremented at each node w according to the travelled edge cost. If $\pi(w)$ reaches 0 at node w, the latter does not further propagate the type-R message in order to keep the communication cost as low as possible. Type-A messages sent in response to type-R messages comprise i) the cost c(w,v)* of the locally selected least cost path p(w,v)* from the local node w to $v \in T_{s,M}$.

When leaf node u decides to join the multicast source-group pair <s,g>, it sends a type-R message to all its direct neighbor nodes (succ(u)) to find the least cost branching path p(u,v)* to a node $v \in T_{s,M}$ ($v \in V_T$). The algorithm applied at node w to process and to further propagate type-R messages is described in Frame.I. The processing of type-A message sent in response by node w to its downstream neighbor nodes pred(w) is described in Frame.II. Observe that node w maintains no additional routing information besides the degree(v) entries required at each step of the execution. At waiting timer $\tau(u)$ expiration, if the set of type-A message received by node u is empty or the cost c(succ(u),v) in all received type-A message is set to infinite, node u declares the multicast source s unreachable. Otherwise, leaf node u selects among the received type-A messages the upstream node succ(u) along the least-cost branching path p(u,v)*, $v \in T_{s,M}$. Node u then further proceeds by sending a message to succ(u) to join $T_{s,M}$. At the end of this process, the routing table of each node v belonging to $T_{s,M}$ ($v \in V_T$) includes: i) one RT entry (stored in the multicast routing information base or MRIB) that indicates the upstream neighbor node to which any Join/Prune message is to be sent for that MDT and ii) one multicast traffic routing entry (stored in the TIB) for the tuple <s,g> and the total number of RT table entries.

3. Performance Analysis

To analyze its performance, the CMR algorithm is executed on a large-scale topology (10k nodes) generated by means of GLP [3] that produces power-law graphs representative of the Internet topology. The execution scenarios consider the construction of point-to-multipoint routing paths for multicast groups of increasing size from 500 to 2000 nodes (selected randomly) with increment of 500 nodes. We consider the Shortest-Path tree (SPT) and the Steiner tree (ST) algorithms to compare the performance of the CMR algorithm. The SPT provides the communication cost reference whereas the ST gives the reference in terms of stretch. The SPT is constructed from a loop-avoidance path-vector routing algorithm carrying the multicast source identifier and the routing path to reach that source. Each node keeps an RT entry per neighbor node (to exchange routing message with its neighbors) and a RT per path to the multicast source s. In order to obtain the near optimal solution for the ST, we consider the ST-Integer Linear Programming formulation. For this purpose, we have adapted the formulation provided in [6] for bi-directional graphs. The communication cost for the ST is based on the dissemination from current MDT nodes (at each step of its construction) of the minimal information for remote nodes not (yet) belonging to the MDT to join it. By this mean, 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 construction as a dynamic scenario would perform.

3.1 Stretch

Fig.1 shows the results obtained for multiplicative stretch ratio between the CRM and the ST algorithm and between the SPT and the ST algorithm. From this figure, we can observe that the CMR algorithm leads to a multiplicative stretch slightly higher than 1 (from 1,09 to 1,06 for multicast group sizes ranging from 500 to 2000 nodes) and provides a gain of at least 5% compared to the SPT (for a multicast group size of 500 nodes) that remains approximately constant with increasing group size.

3.2 Communication cost

As shown in Fig.2, the communication cost for the CMR is relatively high compared to the SPT even if twice lower than the communication cost implied by the ST when the multicast group size reaches 2000. 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 the power law graph. These results suggest that further improvements on the communication process are desirable to apply the CMR on power law graphs comprising of the order of 10k nodes. Following the results obtained in [8], enforcing type-R message propagation by parts shows an improvement of at least one order of magnitude on the communication cost at the expense of slightly deteriorating stretch. Using this method, the joining node u searches locally for a node $v \in T_{s,M}$ in its vicinity $B(u) \subset V$ where $|B(u)| \sim n^{0.5}/\log(n)$; if $T_{s,M}$ is unreachable in B(u), node u then initiates a global search on the remaining topology $V \setminus \{B(u)\})$.

3.3 Routing Table Size

Routing tables (RT) include the MRIB, the TIB as well as the unicast RIB entries (for the SPT scheme that relies on the underlying unicast routing topology). As shown in Fig.3, the gain in terms of number of RT entries obtained for the ST compared to the CMR algorithm decreases from 8,8 to 3,4 when the multicast group size increases from 500 to 2000. The same trend is observed when comparing the SPT to the CMR algorithm. As shown in Fig.4, the gain in terms of memory space consumption obtained for the ST compared to the CMR algorithm varies from 10,1 to 3,8 when the multicast group size increases from 500 to 2000. This gain decreases from about 8,8 to 3,4 when comparing the SPT to the CMRA. Even if this gain decreases with increasing multicast group size, both figures show significant benefit when the group size remains relatively small.

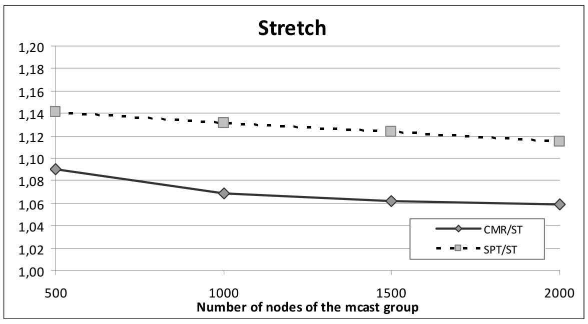

Fig.1: Stretch in function of the multicast group size

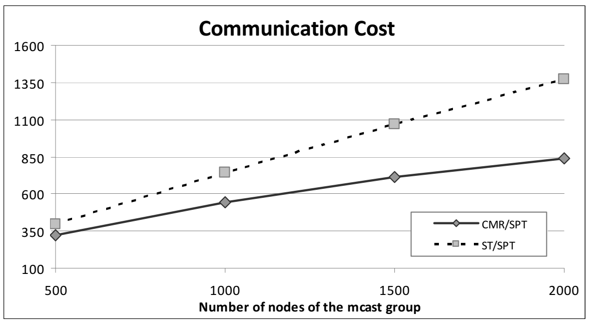

Fig.2: Communication cost in function of the multicast group size

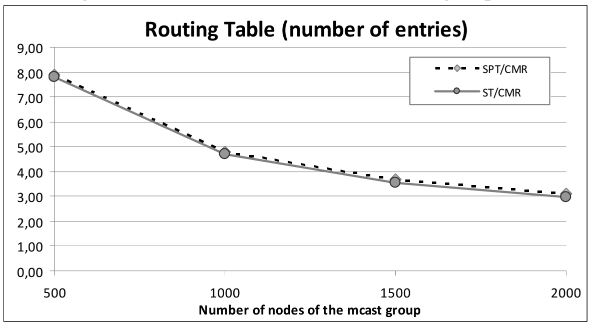

Fig.3: RT entries in function of the multicast group size

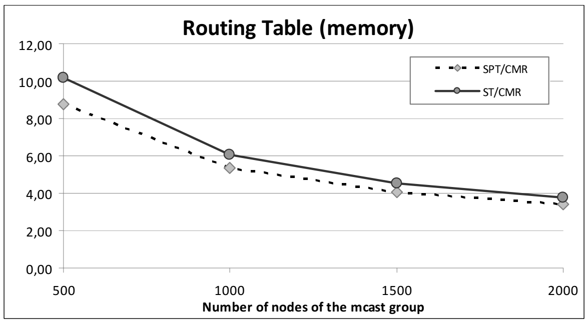

Fig.4: RT memory in function of the multicast group size

4. Conclusion

This paper proposes a leaf-initiated name-independent compact multicast routing algorithm with distributed computation of the RT entries. Our performance analysis considers the multiplicative stretch, the memory space consumption by the RT entries produced by the algorithm and the communication cost required to build point-to-multipoint routing paths. For this purpose, the proposed CMR algorithm has been executed over power-law graphs of 10k nodes and its performance compared to the SPT and ST algorithms ran over the same topologies. The results obtained by simulation show that the CMR algorithm provides a basis for balanced stretch and memory space consumption. Compared to the SPT, the gain in memory space results from the elimination of the underlying unicast RT entries whereas, compared to the ST, this gain is mainly due to the elimination of the RT entries required at each step of the routing path construction. Our initial results show that the communication cost, inherent to the CMRA search phase, can be significantly reduced by segmenting this process. Work is currently conducted to confirm if this technique can effectively decrease the communication cost.

References

[1] I.Abraham, C.Gavoille, D.Malkhi, N.Nisan, and M.Thorup, Compact name-independent routing with minimum stretch, *ACM Transactions on Algorithms (TALG)*, vol.4, no.3, art.37, June 2008.

[2] I.Abraham, D.Malkhi, and D.Ratajczak, Compact Multicast Routing, *Proc. of 23rd International Symposium on Distributed Computing (DISC'09)*, pp.364–378, Elche, Spain, September 2009.

[3] T.Bu, D.Towsley, On distinguishing between Internet power law topology generators, *Proc. of IEEE INFOCOM'02*, vol.2, pp.638–647, New-York (NY), USA, June 2002.

[4] C.Magnien, M.Latapy, and M.Habib, Fast computation of empirically tight bounds for the diameter of massive graphs, *Journal of Experimental Algorithmics (JEA)*, vol.13, art.10, February 2009.

[5] D.Peleg and E.Upfall, A Trade-off between Space and Efficiency for Routing Tables, *Journal of the ACM*, vol.36, no.3, pp.510–530, 1989.

[6] Sage's Graph Library, free open-source mathematics software system (http://www.sagemath.org/).

[7] M.Thorup, and U.Zwick, Compact routing schemes, *Proc. of 13th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA'01)*, pp.1–10, Heraklion (Crete), Greece, July 2001.

[8] 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.