A Locating-First Approach for Scalable Overlay Multicast

Mohammed Ali Kaafar, Thierry Turletti, Walid Dabbous
Projet Planète, INRIA-Sophia Antipolis, France
E-mail:{mkaafar, turletti, dabbous}@sophia.inria.fr

I. Introduction

Recent proposals in multicast overlay networks have demonstrated the importance of exploiting underlying network topology data to construct efficient overlays. While they avoid virtual coordinates embedding and fixed landmarks measurements, these topology-aware proposals often rely on incremental and periodic refinements to improve each node's position in the delivery tree. We claim that there are barriers for the scalability of existing overlay multicast protocols. In fact, periodical refinement and control processes induce additional overhead and high communication cost. On the other hand, users attending a video conferencing session or an event broadcast expect an acceptable quality as soon as they join the multicast session. It is then important to overcome an efficiency problem from which almost all current overlay multicast proposals suffer. This problem is the long convergence time to reach a stabilized quality state in the overlay delivery tree.

We propose a novel overlay multicast tree construction scheme, called LCC : Locate, Cluster and Conquer, designed to address the aforementioned scalability and efficiency issues. The scheme consists in two phases: a selective locating phase and an overlay construction phase. Using partial knowledge of location-information for participating nodes, the selective locating phase algorithm consists in locating the closest existing set of nodes (cluster) in the overlay for a newcomer. It allows then to avoid initially randomly-connected structures without using virtual coordinates system embedding nor fixed landmarks measurements. Then, on the basis of this locating process, the overlay construction phase consists in building and managing a topology-aware clustered hierarchical overlay.

II. The Locating Process

By adopting a network positioning strategy similar to the Meridian approach [1], we propose a novel selective locating algorithm to direct newcomers to the "nearest" cluster.

A. Bootstrap and locating request

Each LCC node keeps track of a fixed number of other nodes in the overlay, and organizes them into its locating system, which is a set of non overlapping levels. These levels are represented by intervals $[r_i, r_{i+1}]$, where r_i are exponentially increasing distances from the considered node taken as the origin. Each level is then bounded by a maximum distance, where the $r_i = \alpha e^{i-1}$ for $i \geq 1$ and $r_0 = 0$. Nodes then measure the distances to the set of nodes they are aware of, and affect each node a position in the correspondent level.

It is assumed that there is a global well-known host called Rendezvous Point (RP) in the overlay network, used to bootstrap new members in the overlay. Initially, a newcomer, say node A, has to contact the RP to obtain the identity of a randomly selected boot node, B. A then measures the distance (delay) from itself to B, $d(A, B)$ and affects B a level in its locating system, say level i. If $d(A, B) \leq R_{max}$ (defining the clustering criterion as described in next section), the locating process terminates, and A sends a request to join B's cluster. Otherwise, A contacts B to inform it of its level, and to obtain the identity of known clusters leaders in A's neighborhood. Once node B receives A's request for locating the closest clusters, it simultaneously queries all cluster representative nodes, in the same level than A, as well as all representative nodes in the adjacent levels $i-1$ and $i+1$. However, in this way, node B could query distant nodes whose distance measurements to node A are useless and would introduce additional overhead. We introduce therefore the selection criterion in order to reduce the number of useless probes. It consists in asking only one selected node in a defined area to measure its distance to the newcomer A. In this way, B eliminates nodes that are close enough to the selected node from the candidates to probe A.

B. The selection criterion

Through different requests, each node maintains for each level i a matrix, M^i, representing learned distances of level i's nodes to each other, and to nodes in adjacent levels $i-1$ and $i+1$. Values in M^i are assigned as and when discovered through the other nodes' locating requests. If the distance is not known, it would be set to a large value in the matrix. This would result in selecting the concerned node even though it does not meet the selection criterion. The selection algorithm is described in the following:

B selects a random node, N^i_j, in level i or adjacent levels $i-1$ and $i+1$, and extracts from M^i its known distance vector, V^i_j, which is the j^{th} row in M^i. If $M^i_{jk} = d(N^i_j, N^i_k)$ is less than a threshold value, γ, then node N^i_k is represented by N^i_j. The threshold value is function of $d(A, B)$, and so of the i^{th} level. More precisely, if the newcomer is close to the node B, the aggregation should be fine-grained and B should use a small γ_i value. But, if $d(A, B)$ is large, node B could use a greater γ_i value. In our algorithm, we choose:

$$\gamma_i = \frac{(d(A, B) - r_i)}{r_{i+1}} * d(A, B)$$

Selected nodes to probe the newcomer are represented by a matrix, say S^i. S^i is originally equal to M^i. At each iteration of the aggregation process run at each row of the matrix M^i, S^i is diminished by the columns of nodes that can be represented by the selected node N^i_j. The selection algorithm terminates when rows of S^i contains only distances of representative nodes.

Using this selection criterion, node B is able to reduce the number of selected nodes measuring their distances to A. These nodes have then to report the results back to B. All selected nodes are then stored into a candidate list that identifies a set of candidate cluster leaders list. Finally, the candidate list is sent to the requesting node A.

C. Which cluster to join?

Node A selects cluster leaders sequentially from the candidate list. Among this list, A contacts cluster leaders satisfying the clustering criterion and initiates joining processes to their clusters respectively. If there are no such cluster leaders in the list, A re-initiates the locating process by contacting the cluster leaders sorted in increasing distances. This procedure is repeated until the clustering criterion is met. Finally, it is necessary to set a stop criterion so that the locating algorithm terminates after repeating the procedures C times. If the algorithm ends without satisfying the clustering criterion, A creates its own cluster.

III. The Clustering Process and Overlay Construction

The objective of the clustering process is to maintain appropriate clusters, in terms of nodes proximity both inside the cluster, and between the clusters themselves. It is initiated by every node joining the overlay, once the locating process terminates. On the basis of their locating results, nodes are partitioned into clusters of nodes that can overlap, i.e. a set of nodes could be at the same time members of more than one cluster; these members are called *Edge nodes*. A maximum distance, R_{max}, defines the interval in which other nodes are considered "nearby". During the clustering process, a node decides at which level it will join the overlay. If it creates its own cluster, it joins the overlay at the top-level topology and starts an inter cluster mesh construction. Otherwise, it becomes a cluster member and joins an intra-cluster mesh in order to derive its delivery tree within this cluster. Edge nodes are allowed to join both levels of the overlay in order to allow better inter-cluster connectivity. We emphasize in this work the mechanisms to increase scalability and robustness. In particular we propose a proactive algorithm to manage leaders failures, and new clusters formations afterwards. We also propose new mechanisms, assigning different priority weights to nodes, to smoothly manage migration due to underlying network changes (See [3] for more details).

Since LCC does not specify the protocol to connect the clusters, any existing overlay construction may be used on top of LCC. We choose to construct the LCC overlay by running the MeshTree protocol [2] at both the top-level and the intra-cluster level. MeshTree embeds the delivery tree in a degree-bounded mesh containing many low-cost links. The constructed mesh consists then of two main components: (i) a backbone structure, consisting in a low-cost tree and connecting nodes that are topologically close together, and (ii) additional links to improve the delay properties. While the "Flat" MeshTree first constructs a randomly connected overlay and relies on incremental improvement, which involves adding/deleting links using a set of local rules, the LCC scheme, initially constructs location-aware overlay based on the locating and clustering processes.

IV. Experimentation Results

Using two complementary evaluation methods: extensive simulations and a thorough PlanetLab testing over the Internet[1] (using more than 200 machines), we compare the scalability and efficiency of LCC with that of initially-randomly connected overlays.

In order to compare LCC to multicast overlays relying on periodic refinements, we experiment a variant of LCC, disabling the locating process and setting the $Rmax$=0, thus emulating MeshTree behavior. We call this variant: Randomly connected Overlay or Flat MeshTree. We also introduce a random locating technique, $RLocating$, that does not maintain nodes within levels in the locating system. $RLocating$ requests a randomly selected set of known nodes to measure their distances to the newcomer. We first consider the convergence time property of the LCC overlay. We define the Average Relative Delay Penalty ($ARDP$), as the average ratio between the overlay delay (d') and the shortest path delay in the underlying network (d) from a source s to all other nodes: $\frac{1}{N-1}\sum_{i=1}^{N-1}\frac{d'(s,i)}{d(s,i)}$, where N is the number of nodes in the overlay. Considering that an overlay delivery tree is "efficient" if the $ARDP$ value is less than a threshold value (say 2), one could intuitively conclude that incremental refinements-based approaches incur a long delay before the overlay delivery tree converges to an optimized structure.

[1]The LCC mechanism has also been implemented as a library that will be available at `http://www-sop.inria.fr/planete/software`

Fig. 1 illustrates this convergence time, plotting $ARDP$ versus the multicast session time in both simulations (Overlay size = 2000 nodes) and PlanetLab testbed. We set the periodical improvement period to 30 seconds, for each of the receivers. We can see that in LCC, $ARDP$ rapidly decreases to a value less than 2 after the first 200 seconds, i.e. less than 7 improvement rounds per node (Note the good matches between the experimental and simulation curves). For the randomly connected overlay, it takes much more time to $ARDP$ to stabilize (more than 1000 seconds). This indicates that LCC can converge very quickly. In fact, it also induces less improvement rounds and link adjustments during overlay growth or frequent membership changes as shown in Fig. 2, with in average 70% less link adjustments.

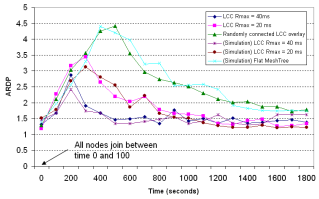

Fig. 1. Convergence Time property.

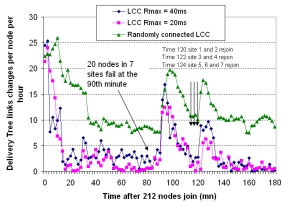

Fig. 2. Link Adjustment rate.

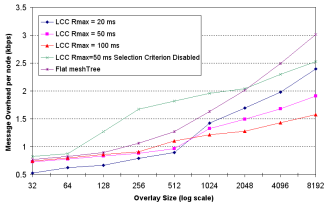

Fig. 3. Protocol overhead.

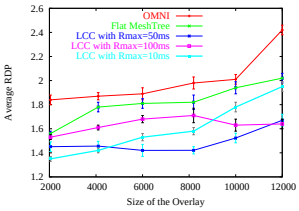

Fig. 4. ARDP Comparison.

We ran simulations to evaluate the control traffic overhead in the overlay during multicast session and observed the protocol behavior in large size overlay. In Fig. 3, we observe the importance of the selection criterion during the locating process. When the selection criterion is enabled, the overhead is insensitive to the overlay size. Disabling the selection criterion, boosts the message overhead due to useless measurement operations during the locating process. Since control messages are not spread outside the clusters, top-level nodes perform "good" results and stabilize the overhead value while the overlay size is increasing. Hence, we observe that the LCC nodes, for R_{max} of 50 ms and 100 ms in the plot, incur in average less than 2 kbps message overhead, in a 8000-nodes overlay. Finally, We plot the $ARDP$ variation according to the overlay size for different overlays in Fig. 4. We observe that the $ARDP$ values for different R_{max} in LCC are roughly maintained at values between 1.2 and 1.6, scaling to large size overlays.

We also studied the locating process efficiency and accuracy. Results showed that the selective locating process is fast, accurate and entails modest resources. On the other hand further simulations show the robustness of the constructed overlay. Future works will include the extension of the scheme to multi-layer hierarchy for scalability purposes, and investigation of techniques to secure the overlay.

References

[1] B. Wong, et al. *A Lightweight Approach to Network Positioning without Virtual Coordinates*. In ACM SIGCOMM 2005.

[2] S. W. Tan, et al. *Meshtree: A Delay optimised Overlay Multicast Tree Building Protocol*. Technical Report 5-05, University of Kent, University of Kent, April 2005.

[3] M. A. Kaafar, T. Turletti, and W. Dabbous. *Locate, Cluster and Conquer: A scalable Topology-Aware Overlay Multicast*, Tech. Report RT-0314, Sophia Antipolis, November, 2005.