

Scalable and Topologically-aware Application-layer Multicast

Yusung Kim Kilnam Chon
Department of Computer Science,
Korea Advanced Institute of Science and Technology
Email: {yskim, chon}@cosmos.kaist.ac.kr

Abstract—We present a scalable and topologically-aware application-layer multicast approach, specially designed for large-scale distributed applications. The proposed approach constructs topologically-aware data paths which are based on topological clustering of multicast group members. The approach does not require any exact network topology information, but instead requires the relative location information of members using landmarks. We partition the members into topologically-aware clusters based on the ordering of their close landmarks. We hierarchically arrange the clusters and separate data paths into two types (i.e., *inside-cluster path* and *outside-cluster path*) to exclude outsider nodes, not belonging to the same cluster, from the inside-cluster paths. Our results on performance evaluation show that constructing topologically-aware data paths can reduce unnecessary high latency and redundant network resource usage with low overhead over existing scalable approaches.

I. INTRODUCTION

An efficient and scalable multicast mechanism is essential to large-scale distributed group communication applications. IP multicast [1] has long been researched and proved the efficiency without redundant packet duplication in network. However, IP multicast has not been widely deployed due to dependency on network-layer routers and thus large parts of the Internet are still incapable of IP multicast [3]. Application-layer multicast moves the role of multicast from routers to end hosts. A major advantage of application-layer multicast does not require any special support from network routers and can be deployed universally [1].

In recent studies, application-layer multicast approaches [3,4] focused on the scalability. The main idea of the approaches is that each multicast participant maintains only state for partial participants. It makes the approaches scalable. However, they paid little effort to ensure that application-layer connectivity is congruent with the underlying IP-level network topology [10]. We can see the importance of topology-awareness in Fig. 2. There are two types of application-layer paths over the physical network which is shown in Fig. 1. The one is a dotted line which is not a topology-aware path. There are three dotted paths which are H1-H4 (i.e., path from H1 to H4), H1-H3, and H3-H2. The sum of the network latency on paths is 500ms. H1-H3 and H1-H4 redundantly include same routers (R1

and R2) and physical links (H1-R1, R1-R2, and R2-H3). H2 is the closest to H1 but is connected to H3. Therefore, the path of H1-H2 via H3 has unnecessary high latency. The other line is a solid line which represents a topology-aware path. The sum of the network latency on paths (H1-H2, H1-H3, and H3-H4) is 310ms which is less than that of the dotted paths. H2 is connected to H1 directly and thus the network latency of H1-H2 is only 10ms. The solid lines include less usage of redundant routers and physical links than the dotted lines do. It shows that a construction of topology-aware paths can reduce unnecessary high latency hops and redundant usage of network resource. Some approaches [5, 6] for application-layer multicast first build a mesh and then construct data paths from the mesh. Mesh based approaches need a global knowledge to construct data paths. It collects complete end-to-end network latency measurement among multicast group members to obtain the global topology information at the application-layer. The information can be congruent with underlying IP-level topology. However, such as the complete end-to-end measurement is high overhead and limits its scalability.

In this paper, we propose a scalable and topologically-aware application-layer multicast approach. The proposed *landmark based approach* constructs topologically-aware data paths based on well-known landmarks which are spread across the Internet. The approach partitions multicast group members into topological clusters. Each cluster includes topologically close members and the clusters are hierarchically arranged according to the global topology information. We separate data paths into two types; inside-cluster path and outside-cluster path to avoid including outsider nodes, belonging to different clusters, among inside-cluster paths. This scheme can reduce unnecessary high latency hops and redundant network resource consumption. Our approach does not require any exact information of underlying IP-level network topology or complete end-to-end measurement. Each member only contacts partial other members and a small number of landmarks. Such low overhead makes our approach scalable.

The rest of the paper is structured as follows: we review the existing approaches for application-layer multicast in Section II. Then we describe our model in Section III, and present the methodology and results on performance evaluation in Section IV. We finally conclude in Section V.

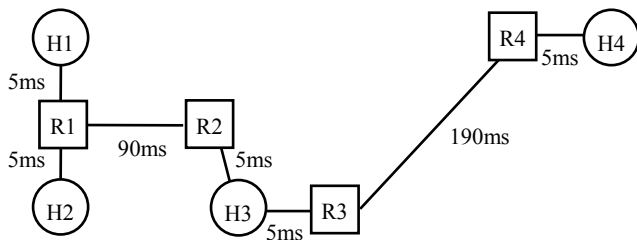


Fig. 1. Physical network topology; H_i is a host, R_i is a router, and N_{ms} is the network latency.

II. RELATED WORK

A number of approaches have explored application-layer multicast. However, being end-to-end, overall paths along the application-layer can get rather long in terms of delays, and data can be replicated several times over some physical links [2]. Having a good level of congruence with the physical topology is challenging and the key to good performance [10]. ALMI [6] takes a centralized approach to construct data paths. Members of a multicast group perform network measurements between themselves. A controller collects the result of measurements from all members, computes a minimum spanning tree based on these measurements, and disseminates routing tables to all members. This approach can consider the global topology at the level of application and obtain a high accuracy of the topological information. However, the overhead of the heavy-weight measurement limits its scalability. NICE [3] and HMTP [4] are tree-first approaches to improve scalability of application-layer multicast. They rely on a recursive algorithm to build the tree; a newcomer first contacts the tree root, chooses the best node among the root and root's children, and repeats this top-down process until it finds an appropriate parent [1]. In these approaches, each member of multicast group only contacts partial members along the tree links. It is low overhead and scalable as compared with centralized approach. However, tree-first approaches are difficult to consider the global topology information and cause to include unnecessary high latency hops and increase the redundant usage of network resource. TAG [7] designs topology-aware application-level multicast protocol. Underlying topology information can be obtained from traceroute or OSPF/BGP routing table dumps. However, some routers do not allow ICMP messages for the traceroute command. The information of OSPF/BGP routing table is not directly available to end-user applications [10]. Instead of such as exact topology information, some approaches use simple topological hints [9,10]. Global Network Positioning (GNP) [9] is based on absolute coordinates computed from modeling the Internet as a geometric space. GNP is applied to estimate host locations to construct a mesh for application-level multicast [8].

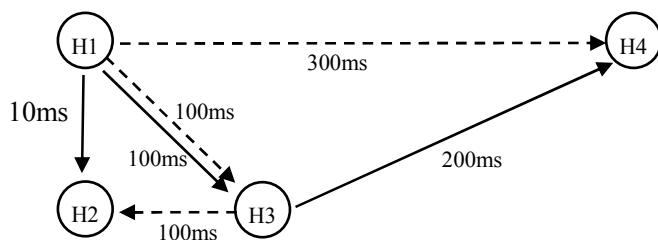


Fig. 2. Comparison of two application-layer paths; Solid lines are topology-aware paths and dotted lines are non topology-aware paths.

We derive our approach from *binning* scheme [10] which is to have a set of nodes independently partition themselves into disjoint bins such that nodes within a single bin are relatively closer to one another than to nodes not in their bin in terms of network latency. The scheme requires a small number (In [10], it shows 8~12 machines should suffice for the current scale of the Internet) of well-known landmark machines spread across the Internet.

III. MODEL

Our approach hierarchically organizes a set of multicast group members into a topological cluster. Each member gets its own relative location information over global topology from landmarks. However, landmarks do not keep any information of members or other landmarks. Neither communication nor cooperation is among landmarks. The proposed approach is a purely distributed model. It means there is no central controller and single point failure. Each member is automatically localized as belonging to its own cluster. Every member measures network latency to each landmark when it joins a multicast group and sorts the landmarks in order of increasing network latency. This ordering of close landmarks presents the member's own relative location information over the global topology. Members that are topologically close have the same ordering of close landmarks and belong to the same cluster. Fig. 3 shows an example of clustering. There are 3 landmarks and 9 hosts (H1~H9). H1 measures network latency to 3 landmarks and has the ordering of (1,3,2) which means (landmark 1, landmark 2, landmark 3). Therefore, H1 belongs to $C[1,3,2]$ which is the cluster that has the ordering of (1,3,2). Each H2~H9 also belongs to its own cluster by the same way. To reduce unnecessary high latency hops and redundant usage of network resource, we separate data paths into two types; inside-cluster path and outside-cluster path. Inside-cluster paths are composed of members in the same cluster. Outside-cluster paths are connected among outsiders that are in different clusters. The separation of inside and outside cluster paths makes sure that topologically close multicast members are first connected by inside-cluster paths without including ones

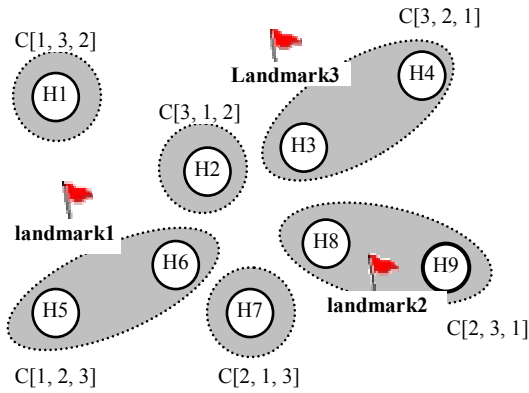


Fig. 3. Topological clustering with 3 landmarks. *Flag* is a landmark, *H_i* is a host, *C_[i,j,k]* is a cluster which consists of members who have the same ordering (*i,j,k*) of the landmarks.

that belong to the different clusters.

We now describe how a host can join to the multicast group and have its own cluster to construct data paths. We assume there is a Rendezvous Point (RP) that has known landmarks. RP only keeps state for directors of tier 1 which is show in Fig. 4. Director is also a member of multicast group to direct other members to reach their own clusters. In Fig. 4, there are 4 landmarks and RP keeps the information of 4 directors (D1~D4). D(*i*) is a director that has ‘landmark *i*’ as its first closest landmark. Each D(*i*) keeps the information of D(*i,j*) in Fig. 5. When H1 that has the ordering of (1,2,3,4) contacts to RP, if there is no D(1) and then H1 takes a role of D(1) because H1 has a ‘landmark 1’ as its first closest landmark. H2 that has the same ordering as that of D(1) contacts RP again, RP let H2 contact D(1). If D(1) does not have D(1,2), H2 takes a role of D(1,2) because H2 has a ‘landmark 2’ as its second closest landmark. Fig. 5 shows completely organized directors for all landmarks. If H that has ordering of (1,2,3,4) joins a tree in Fig. 5. H first contacts RP, RP gives H the information of D(1), and D(1) directs H to D(1,2). H can finally reach C[1,2,3,4] along directors. After clustering, the construction of data paths is as following:

- Inside-cluster paths: Members in a same cluster used Distance Vector routing algorithm. The bottom tier director controls the size of a connected cluster to be less than 10. If it comes to be more than 10, the cluster is divided to two clusters and each cluster selects its own director among cluster members.
- Outside-cluster paths: The outside-cluster paths are composed of directors and the source. Each director knows its own upper director and uses the tree-first approach to coordinate the fan-out degree at a host. The mechanism of the tree-first approach is presented in the section II. Related Work.

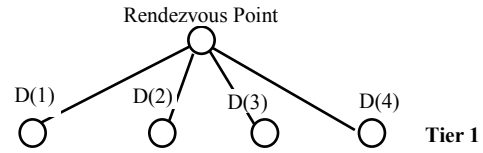


Fig. 4. *Rendezvous Point* and *D(i)*; *D(i)* is a director that has a ‘landmark *i*’ as its first closest landmark. The number of landmarks is 4.

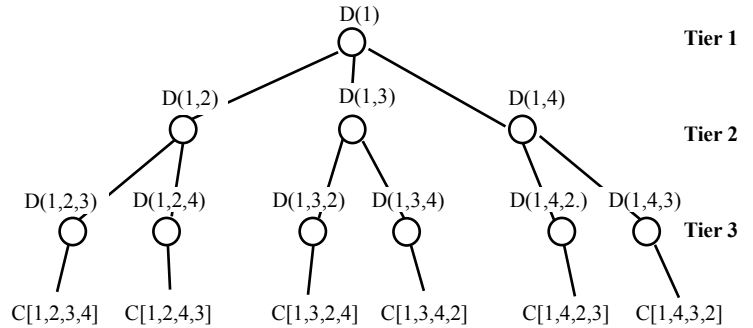


Fig. 5. Hierarchical directors with 4 landmarks; *D(i)*, *D(i,j)*, and *D(i,j,k)* are directors that have (*i*), (*i,j*), and (*i,j,k*) as the order of close landmarks. Non director hosts belong to their own clusters.

IV. PERFORMANCE EVALUATION

We have evaluated the performance among three approaches. The first is the centralized approach [6] which is based on complete end-to-end network latency among multicast group members. When it constructs data paths, it can consider the global topology information in application level. The second is tree-first approach [4] which is for scalability as each member of multicast only contacts partial other members. The last is our model which called landmark based approach.

Two different topologies have been used. One is generated by Inet [11] which is a topology generator. It is an AS-level and the Internet-like topology which follows a power-law. We generated 10,000 nodes and picked up 20 landmarks which are spread across global topology. Inet allocates a ‘weight’ as a ‘link cost’ to each link. We consider the ‘weight’ as network latency. The other topology was built by the measurement experiment on Logistical Backbone (L-BONE) [12] that consists of sharing storage servers distributed in the world. We selected 170 nodes on L-BONE and measured 10KB transfer time as a latency instead of RTT in full mesh. (Most of L-Bone nodes are on PlanetLab[13]) We also selected 6 landmarks among them. We used the four performance metrics to evaluate three approaches. Each metric is as following.

- Average data path latency [4]: It is average latency for data paths constructed by each approach.

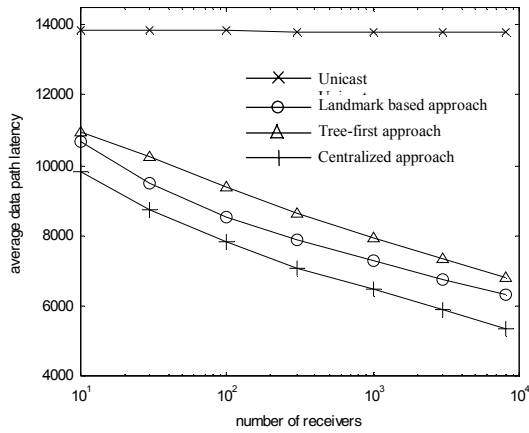


Fig. 6. Average data path latency on Inet

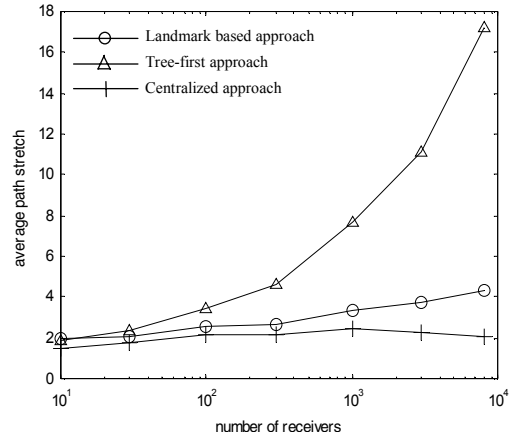


Fig. 7. Average stretch on Inet

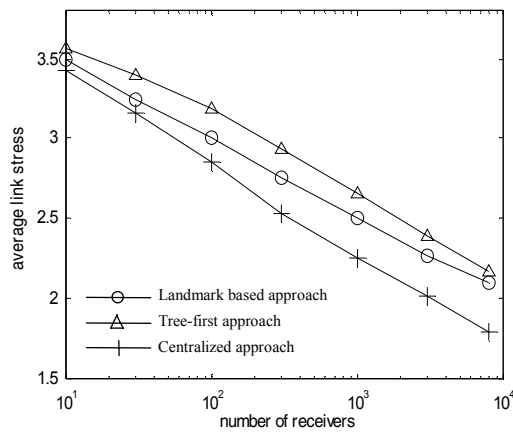


Fig. 8. Average stress on Inet

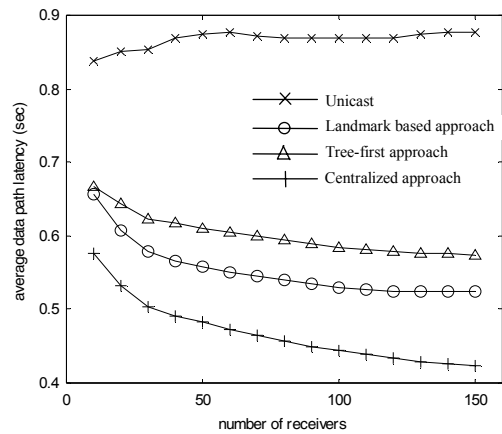


Fig. 9. Average data path latency on L-BONE

- Stretch [5]: The ratio of the relative delay penalty against the direct path of unicast
- Stress [5]: Total number of duplicate copies of a packet over a physical link.
- Control overhead [2]: We count the number of control messages which are generated by members to construct data paths.

In Fig. 6, unicast used direct path from source to each receiver (multicast member). The average data path latency of unicast is almost same all the time. However, three approaches reduced average data path latency more as the number of receivers increased. Centralized approach had the least average latency and landmark based approach had a less average latency than that of tree-first approach. In Fig. 7, we evaluated the stretch of path against unicast on Inet. Centralized approach had about twice stretch against unicast all the time. However, tree-first approach had high stretch up to 17 times. Because tree-first approach could not consider the global topology information, the data paths

might include unnecessary hops. The stretch of landmark based approach was less than 4 times. Fig 8 shows average stress on Inet. Centralized approach had the least stress. Landmark based approach was also less than that of tree-first approach.

In Fig. 9, we presented the average data path latency on L-BONE. The result was similar to Fig. 6. The average data path latency of landmark based approach located between them of different two approaches. Because L-BONE topology was based on end-to-end network latency measurement and we couldn't know the state of physical link, there is no evaluation of average stress on L-BONE. In the stretch evaluation on L-BONE, three approaches had a small difference because the number of receivers was small (less than 170).

We have evaluated the quality of data paths; stress, stretch, and average data path latency. Centralized approach had the highest quality among the three approaches because the data path is based on the global topology information by complete end-to-end network latency measurement.

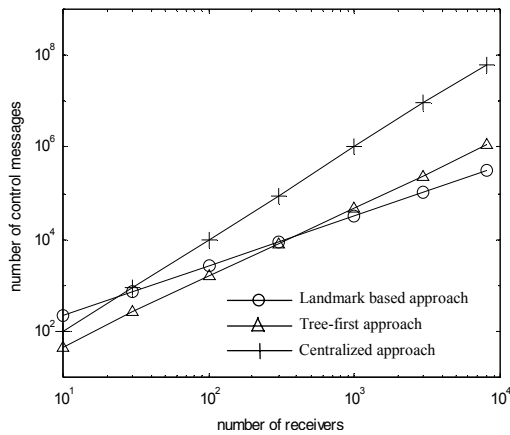


Fig. 10. Number of control messages on Inet

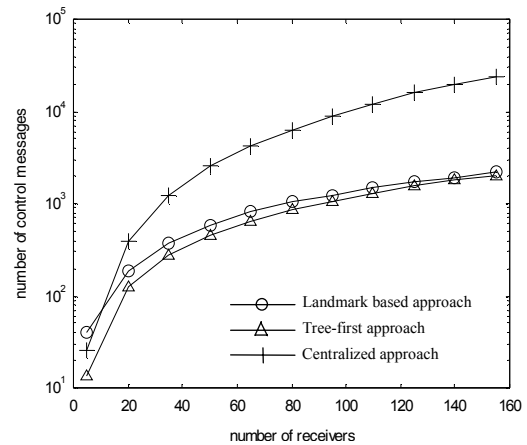


Fig. 11. Number of control messages on L-BONE

However, the complete end-to-end measurement limits its scalability. Fig. 10 compares the number of control messages on Inet. The number of control messages in the central approach was up to 100 times more than those of other approaches. When the receiver size is less than 30, the number of control messages in landmark based approach was the highest because each receiver had to measure network latency to 20 landmarks. As the number of receivers increased, the slope of landmark based approach was close to that of tree first approach. The number of control messages is related to control overhead [2]. Landmark based approach had low overhead as much as tree-first approach did. Fig. 11 also presents the number of control messages on L-BONE. Landmark based approach also had low number as much as tree first approach did.

V. CONCLUSION

Traditional application-layer multicast approaches do not need an additional network infrastructure. Such schemes are easy to deploy but are challenging to be congruent with IP network topology. In this paper, we presented a new approach for application-layer multicast using landmarks. Our approach is scalable and constructs topologically-aware data paths. Using a small number of landmarks, we partition members of multicast group into topologically-aware clusters and hierarchically arrange these clusters according to global topology. Our approach separates data paths into two types; inside-cluster path and outside-cluster path to avoid including outsider nodes, are not in the same cluster, among inside-cluster paths. Results on performance evaluation comparing our approach against existing approaches in two different topologies show that our approach has low overhead, moderately link stress and average data path latency, and low stretch.

An open problem would be the selection of landmarks; the number of landmarks and the location of them. We also plan to study a bandwidth-aware approach while we have focused only into network latency. In some data intensive applications, bandwidth is an important factor of topology awareness as much as the network latency.

REFERENCES

- [1] S. Deering and D. Cheriton, "Multicast routing in datagram the Internetworks and extended LANs," in *ACM Transactions on Computer Systems*, Vol. 8, No. 2, May 1990.
- [2] A. El-Sayed, V. Roca, and L. Mathy, "A survey of proposals for an alternative group communication service," *IEEE Network*, v. 17 no. 1, pp. 46-51, January, 2003
- [3] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in *Proc. ACM SIGCOMM*, Pittsburgh, PA, USA, August 2002.
- [4] B. Zhang, S. Jamin, and L. Zhang, "Host multicast: a framework for delivering multicast to end users," in *Proc. IEEE INFOCOM*, New York, NY, USA, June 2002.
- [5] Y. Chu, S. Rao, and H. Zhang, "A case for end system multicast," *ACM SIGMETRICS*, Santa Clara, CA, USA, June 2000.
- [6] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel, "ALMI: an application level multicast infrastructure," in *Proc. 3rd USENIX Symp. The Internet Tech. and Sys.*, San Francisco, CA, USA, March 2001.
- [7] M. Kwon and S. Fahmy, "Topology-Aware Overlay Networks for Group Communication," in *Proceedings of ACM NOSSDAV*, pp. 127-136, May 2002.
- [8] T.S. Eugene Ng and Hui Zhang, "Predicting the Internet network distance with coordinates-based approaches," in *Proceeding of INFOCOM 2002*. New York, June 2002.
- [9] WC. Wong and SH. Chan, "Improving delaunay triangulation for application-level multicast," in *Proceedings of IEEE Globecom 2003*, San Francisco, CA, December 2003
- [10] S. Ratnasamy, M. Handley, Richard Karp, and S. Shenker, "Topologically-aware overlay construction and server selection," in *Proc. IEEE INFOCOM*, New York, NY, USA, June 2002.
- [11] Inet, <http://topology.eecs.umich.edu> (Accessed: 29 February 2004).
- [12] Lbone, <http://loci.cs.utk.edu> (Accessed: 29 February 2004).
- [13] PlanetLab, <http://www.planetlab.org> (Accessed: 29 February 2004)