

A Hierarchical Overlay Multicast Network

Yingyin Jiang, Min-You Wu, and Wei Shu

Department of Electrical and Computer Engineering, The University of New Mexico

Abstract—The overlay multicast has drawn a lot of attention as an alternative to IP multicast. In the recently proposed overlay multicast solution, most of the implementations do not address the inter-domain administrative issue. For wide area multicast, it is unavoidable to have communication among several administrative domains. At this time, multicast routing cannot assume the knowledge of global network topology, nor can it ignore the routing policy constrained by contractual commercial agreement between administrative domains. This problem can be solved by a hierarchical organization of the routing infrastructure. In this paper, we propose a zone approach of overlay multicast that performs hierarchical routing on two-levels – one level for within each of the administrative domains and another level for among the domains. We compare a single centralized routing to our zone-approach routing for overlay multicast and demonstrate that the performance penalty for the hierarchical organization of routing is small. The results have shown that our distributed design can achieve very close performance to the centralized algorithm. At the same time it enables the composition of different administrative networks, each with its own independent multicast protocol and administrative policy.

I. INTRODUCTION

IP multicast service [1] has been widely accepted as the way to implement multicast for the past decade. However, despite the conceptual simplicity of IP multicast and its obvious benefits, it is yet to take off. As an alternative to IP multicast, *Overlay Multicast* has recently received more attention [2], [3], [4], [5], [6]. Overlay multicast makes deployment of multicast functionality easier since they implement their functionality entirely at IP hosts and require no modifications to the core routing technology. The existing overlay multicast projects can be classified into two categories: *end-to-end* overlay and *proxy-based* overlay. In end-to-end overlay, every member in the multicasting group shares the responsibility to forward data to other members. End hosts self-organize into a multicasting tree. Narada [4], Yoid [2] and ALMI [6] are examples of such a structure. For proxy-based overlay, the multicasting service is fulfilled with the help of multicast proxy nodes, which can duplicate data and forward data to end hosts with a predefined routing algorithm. The proxy-based overlay structures include Scattercast [3] and Overcast [5].

An Internet graph can be logically divided into several Autonomous Systems (ASs), each under its administrative control, having its own routing policy and running its own underlying routing protocol. What's more, the interior structure of an AS is not known to other ASs under different administrative control. Because of these facts, no single multicast protocol has taken over as the only dominant protocol to span the entire internet. Instead we expect islands of these

non-interoperable multicast connectivity. The zone architecture we proposed is based upon this observation of Internet AS structure. It is a proxy-based overlay multicast network that establishes a multicast tree. The zone architecture incorporates the hierarchical structure of Internet into the basic design, instead of the flat structure. The architecture constructs the inter-domain multicast tree via an overlay that is composed of application layer multicast border gateways. A multicast tree is distributively built within each domain and is independent from others. Each zone can implement its own multicast routing subjecting to optimize some certain metrics. A multicast proxy is a replication engine that can be placed to end hosts, network edges, or network cores. Only when proxies are placed to network edges or network cores, can they offer the maximum benefits. However, not everyone has freedom to place a proxy anywhere. In our solution, we assume that, within a zone, the zone administrator has the authority to place proxies anywhere to optimize the local traffic. We also assume that a network provider may place a proxy in its core network.

The major finding of this paper is that when traffic in every zone is optimized, the delay and cost of the whole network can be very close to the global optimal. Thus, with the power of localization, optimal decisions within a zone will result in global optimization in a federation of zones. The rest of paper is organized as follows. Simulation methodology is given in Section 2. Section 3 analyzes the performance result and Section 4 concludes the paper.

II. HIERARCHICAL MULTICAST OVERLAY

In this section, we will present the design of our zone-approach architecture. We first discuss the main components of our architecture that our model builds on and then we will fully investigate the zone approach. Besides the basic components such as node and link, there are three other important components in zone architecture such as zone network, border gateway, similar to those described in [7]. Below we give the definitions.

- **Zone network** – The whole network is divided into several clouds, each of which is a single domain under its own administrative control and employs its own routing policy. A zone does not have any knowledge of other zones, except for their gateways. Each zone network deploys its own intra-domain multicast routing protocols. These protocols may not be the same and they can cooperate with each other using the inter-domain multicast routing architecture we proposed. Each zone network has a num-

ber of border routers that connect the zone network to other zones. Multiple entries for one zone network are allowed in our work.

- Border Gateway(BG) – The multicast proxy may or may not be placed on border routers. A multicast border gateway is a border router where a proxy is placed. The inter-domain routing layer is formed and organized among border gateways. Border gateways are organized together to build a routing protocol to enable inter-domain communication and optimize multicast performance. Border gateways in one zone could provide multicast data replication to a peer gateway in a different zone.
- Interior Proxy – The multicast proxy may be placed on an interior node in a zone network. In this case, only client nodes in the same zone can benefit from this deployment of proxy.

A. Metrics

We judge the quality of a multicast tree \mathcal{M} by three metrics: the normalized aggregate delay \mathcal{D} , the normalized aggregate cost \mathcal{B} and *Weighted Sum of Delay and Bandwidth consumption*($WSDB$). The aggregate delay \mathcal{D}_M of a multicast tree is the sum of delays, between the source and every client along the multicast tree. The aggregate cost \mathcal{B}_M of a multicast tree is the sum of bandwidth consumption of all links. These metrics is normalized as: $\mathcal{D} = \mathcal{D}_M / \mathcal{D}_{IPM}$, $\mathcal{B} = \mathcal{B}_M / \mathcal{B}_{IPM}$, where IPM stands for the traditional IP multicast tree, which is commonly used as a reference point for comparison. Due to the SPT-oriented routing algorithm used in IP multicast, \mathcal{D}_{IPM} is the minimal value among all possible \mathcal{D}_M . Based on the same routing, \mathcal{B}_{IPM} is obtained by setting every node as an IP multicast router.

Both \mathcal{D} and \mathcal{B} depend on the routing algorithm used to construct the multicast tree. In order to optimize both metrics simultaneously, we also use the metric *Weighted Sum of Delay and Bandwidth consumption* ($WSDB$), and $WSDB = \lambda \mathcal{D} + \mathcal{B}$, where λ is a weight indicating the relative importance of the delay. In this paper, λ is set to be 1.

B. Multicast Routing Algorithms

Given a proxy placement on the graph, multicast routing is to design a routing algorithm \mathcal{A}_r that builds a multicast tree \mathcal{M} optimized toward certain metrics such as \mathcal{D} , \mathcal{B} or $WSDB$. In our comparison, given the same proxy placement, we compare a globally-constructed multicast tree and a hierarchically-constructed multicast tree, in terms of \mathcal{D} , \mathcal{B} , and $WSDB$.

1) **Global Routing:** In global routing, the multicast distribution tree is built using the knowledge of a global network topology information. Global routing is centralized and not scalable. Although it is an unrealistic model for a large network, it provides a reference for comparison.

Let us first take a look at the routing algorithm in global routing. What we used in our simulation is the *Hybrid Optimal Tree* (HOT) minimizing $WSDB$, a compromise between minimizing delay and minimizing bandwidth [8]. The algorithm

that generates the HOT tree is called Direct-DB (DDB) algorithm, \mathcal{A}_{DDB} — it routes proxies one by one, in ascending order of their shortest distance to the source. The first proxy connects to the source through the shortest path. The following proxies connect to either the source or a placed proxy, whichever minimizes $WSDB$. After the multicast tree is established, each non-proxy node connects to the proxy node that generates the smallest $WSDB$ value. This algorithm generates a tree that balances delay and cost requirements [8].

2) **Zone-approach Routing:** In our zone architecture, a multicast tree is constructed in two levels. Each domain builds its own intra-domain multicast tree, and this multicast tree is built of border gateways and interior proxies. Then a multicast tree of border gateways across domains is built by an inter-domain routing algorithm.

Within a zone, we build a HOT tree using \mathcal{A}_{DDB} , the same routing algorithm as in global routing. Between zones, we could also use the same algorithm to construct a multicast tree to minimize both the cost and delay. However, a simple shortest-path tree (SPT) might be a better choice since it is easier to build with the DVMRP protocol. The SPT tree minimizes \mathcal{D} [8]. The other reasons to use SPT include (1) the delay is more crucial in inter-zone routing; and (2) the cost of multicast between zones is relatively low compared to that within a zone.

An important architecture design choice is whether the zones or gateways should be the node in the inter-zone multicast tree. We were considering inter-domain routing on per-zone basis. There is only one representative border gateway node for each zone, and it works as the sole point that the traffic can flow into this zone. This becomes a suboptimal solution when there are more than one border gateway in a zone. Usually the best route of a multicast stream originating from a zone depends on the proximity of the source node to each border gateway. There may be the situation that some nodes use border gateway A rather than border gateway B to avoid longer delay and others use border gateway B for the same reason. This naive scheme to restrict a zone to a single entry point is proved not to be a good design choice by our experiment.

Thus, we do not build trees of domains, akin to BGP, but rather build trees of multicast gateways. Any subscriber to multicast service gets to choose its gateway node to exit, dictated by the intra-domain multicast routing protocol. Each border gateway maintains a routing table that is used to compute the best route from a client's zone to any other zones in the network. This allows us to select the appropriate exit border gateway for a multicast stream from its source zone. Border gateways in a zone that have service subscribers would join the inter-domain routing layer and participate the routing information exchanging.

Fig.1 shows the routing algorithm $Zone_{SPT-HOT}$ used in the zone-approach routing. It first builds an inter-domain shortest path tree and then a HOT tree using \mathcal{A}_{DDB} in each zone. Since the source node can be any node in the graph, in Step 1, we construct a subgraph $G_{k,bg} = \{n_i | z(n_i) \equiv z(n_s) \text{ and } m(n_i) \equiv 2\}$

The substrate network is a graph G with N nodes $\{n_1, \dots, n_N\}$ and M links, the delay of the link $l_{i,j}$ connecting n_i and n_j is denoted as $d(n_i, n_j)$. There is one source node, n_s , and totally K zones. Each node belongs to a zone $z(n_i) = 1, \dots, K$. Every node is

$$\text{marked by } m(n_i) = \begin{cases} 2 & \text{if } n_i \text{ is a border gateway} \\ 1 & \text{if } n_i \text{ is a proxy only} \\ 0 & \text{otherwise} \end{cases}$$

- step 1) Let $k = z(n_s)$ and a subgraph
 $G_{k,bg} = \{n_i | z(n_i) \equiv k \text{ and } m(n_i) \equiv 2\}$
 Initiate DVMRP from n_s within $G_{k,bg}$,
 every node $n_i \in G_{k,bg}$ will obtain its delay value $d(n_i, n_s)$.
- step 2) Let a subgraph $G_{bg} = \{n_i | m(n_i) \equiv 2\}$
 Initiate DVMRP from $n_i \in G_{k,bg}$ within subgraph G_{bg} ,
 every node $n_i \in G_{bg}$ will obtain its delay value $d(n_i, n_s)$.
 Now, the inter-domain multicast tree has been built.
- step 3) For each zone j , let a subgraph $G_j = \{n_i | z(n_i) \equiv j\} \cup n'_s$
 where n'_s is a virtual source node,
 and $G_{j,bg} = \{n_i | n_i \in G_j \text{ and } m(n_i) \equiv 2\}$
 for $n_i \in G_{j,bg}$, set $d(n_i, n'_s) = d(n_i, n_s)$
 Apply \mathcal{A}_{DDB} algorithm in G_j , with n'_s as the source.

Fig. 1. Routing Algorithm $Zone_{\text{SPT-HOT}}$

consisting of all border gateways in the source zone. Initiating the DVMRP from source node n_s to all nodes in subgraph $G_{k,bg}$, a SPT tree can be obtained and every node $n_i \in G_{k,bg}$ will have its delay value $d(n_i, n_s)$. Then Step 2 builds an inter-domain level SPT tree on $G_{bg} = \{n_i | m(n_i) \equiv 2\}$ across all border gateways in the network. All gateways thus route through this SPT tree to reach a gateway in the source zone and then reach source node n_s , and therefore, obtain delay value $d(n_i, n_s)$. In Step 3, we apply the \mathcal{A}_{DDB} algorithm to minimize $WSDB$ for each zone. First, we build a virtual graph G_j for zone j , including a virtual source node n'_s . The virtual source node n'_s is connected to border gateways $n_i \in G_{j,bg}$ in zone j with $d(n_i, n'_s) = d(n_i, n_s)$, where $d(n_i, n_s)$ is obtained from the inter-domain routing in Step 2. We then apply \mathcal{A}_{DDB} to build the HOT tree on graph G_j with n'_s as the source.

III. PERFORMANCE RESULT

In this section, we will compare our hierarchical zone routing algorithm $Zone_{\text{SPT-HOT}}$ with the results of global routing $Global_{\text{HOT}}$ using a set of transit-stub graphs produced by Georgia Tech Internetwork Topology Generator GT-ITM [9]. The transit-stub model obtains graphs that more closely resemble the internet hierarchy than a pure random graph. It has a similar property of power-law model in its stub domain and a similar topology of BGP graph in its transit domain. In our performance test, all transit-stub graphs have approximately 1,000 nodes. We show results with different zone sizes, changing from 5 zones each of about 200 nodes through 40 zones each of about 25 nodes. All the results are averaged over ten graphs in each figure. In each instance, a random node is assigned as the source, which remains the same for different routing algorithms.

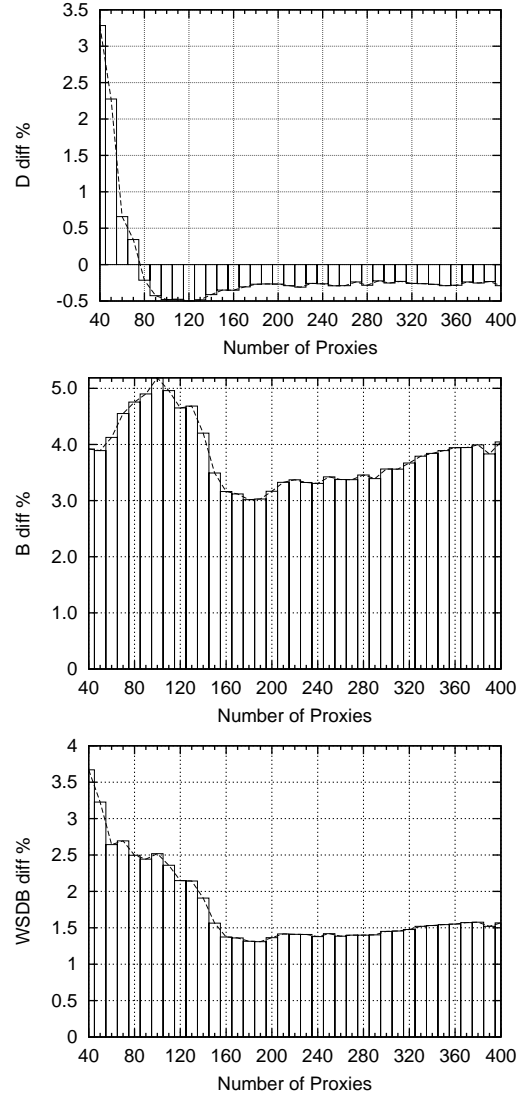


Fig. 2. Comparison for \mathcal{G}

For routing comparisons, we have to decide a proxy placement first. Each zone itself is responsible to place a given number of proxies within zone. They all use the following rule to place proxies. We assume that there are P_{z_i} proxies placed for zone i . If zone i has n border routers, and $P_{z_i} \leq n$, proxies are all placed on border routers randomly. If $P_{z_i} > n$, it would first place n proxies on border routers, the rest $(P_{z_i} - n)$ proxies would be placed randomly on other nodes within the domain. This is the default proxy placement for routing comparisons in this paper. Based on this proxy placement, $Zone_{\text{SPT-HOT}}$ is compared to the globally-constructed HOT multicast tree. $Zone_{\text{SPT-HOT}}$ refers to construct inter-domain multicast shortest path tree and within a zone, an intra-domain HOT tree is used for data distribution. In the simulation, proxies are placed before building zone and global multicast trees.

Fig. 2 are the routing comparison results for graph \mathcal{G} . We show the percentage of performance difference between

the two approaches $Zone_{SPT-HOT}$ and $Global_{HOT}$. The percentage of difference of \mathcal{D} , \mathcal{B} and $WSDB$ is calculated by $\frac{Value_{zone} - Value_{global}}{Value_{global}} \cdot 100\%$. Graph \mathcal{G} is a 1000-node graph with 5 transit zones and 20 stub zones. Each transit zone contains, on an average, 4 transit nodes and each transit node will have 1 stub domain connected to it. Each stub zone has, on an average, 49 stub nodes. Delay and bandwidth consumption are normalized values by the IP multicast tree. We only show results after all transit nodes are placed proxies and each stub zone has at least one proxy placed, which is starting from 40 proxies for graph \mathcal{G} . Considering delay only, $Zone_{SPT-HOT}$ is better than $Global_{HOT}$ with more than 80 proxies since it uses the shortest path tree in inter-domain routing. For metrics \mathcal{B} and $WSDB$, although $Zone_{SPT-HOT}$ is slightly worse than $Global_{HOT}$, it achieves very close results comparative to $Global_{HOT}$. The difference between the two is less than 3.5% for \mathcal{D} , less than 6% for \mathcal{B} and less than 4% for $WSDB$.

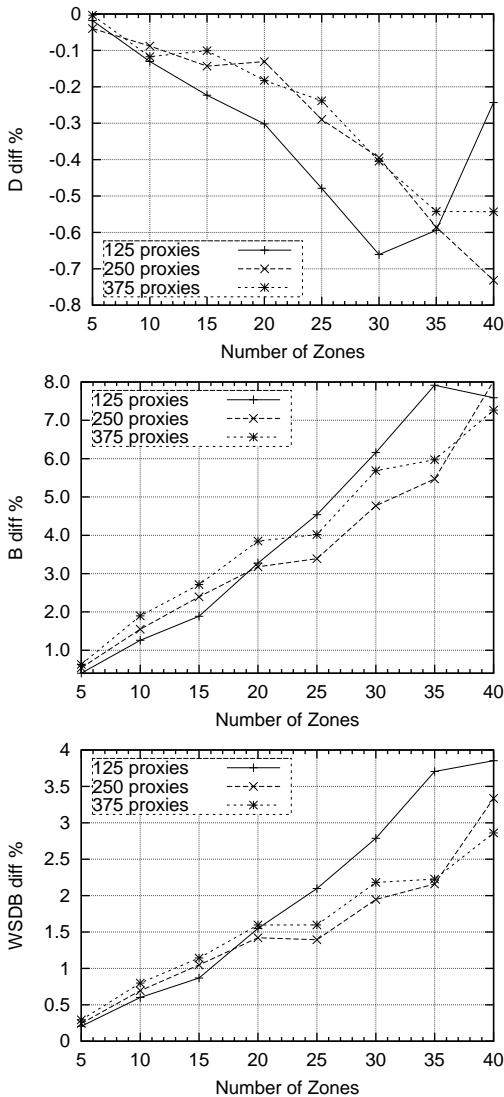


Fig. 3. Comparison of Global and Zone routing with varying zone sizes

We also conducted an interesting experiment aiming to answer the question that whether the size of a zone affects the performance of $Zone_{SPT-HOT}$. We obtained a set of 1000-node graphs of different average zone sizes by changing the setting of the total number of zones during the graph generation. As shown in Fig.3, the y-axis shows the percentage of \mathcal{D} , \mathcal{B} and $WSDB$ differences for $Zone_{SPT-HOT}$ compared to the result of $Global_{HOT}$. Three curves with different number of proxies placed are included in the figures. For delay \mathcal{D} , $Zone_{SPT-HOT}$ is better than $Global_{HOT}$ since the SPT tree is used for inter-domain routing but the differences are less than 1%. For bandwidth consumption \mathcal{B} , difference between $Zone_{SPT-HOT}$ and $Global_{HOT}$ is growing when the average size of a zone is getting smaller in the graph. $WSDB$ also has similar result as bandwidth consumption \mathcal{B} . The $WSDB$ performance of the zone approach is getting further away from the global optimum when there is a larger number of zones in the same size of graph.

IV. CONCLUSION

This work is an effort towards a distributed hierarchical multicast routing approach that does not assume the knowledge of global network topology for overlay multicast networks. We compare a centralized multicast routing algorithm to our zone-approach multicast routing algorithm and demonstrate that the performance penalty for the hierarchical organization of routing is small. The results have shown that our distributed design can achieve very close performance to the centralized algorithm. At the same time hierarchy enables the composition of different administrative networks, each with its own independent multicast protocol and administrative policy. Further work with this distributed design is to utilize the same idea in design of a distributed proxy placement algorithm, based on these routing results.

REFERENCES

- [1] S. Deering, "Multicast routing in a datagram internetwork," *Ph.D. dissertation, Stanford University, Palo Alto, California*, 1991.
- [2] P. Francis, "Yoid: Your own Internet distribution," Tech. Rep. at www.aciri.org/yoid, UC Berkeley ACIRI Tech Report, Apr. 2000
- [3] Y. Chawathe, "Scattercast: An Architecture for Internet Broadcast Distribution as an Infrastructure Service," *Ph.D. thesis, Department of EECS, UC Berkeley*, Dec. 2000.
- [4] Y. Chu, S. Rao, and H. Zhang, "A case for end system multicast," in *ACM Sigmetrics*, 2000.
- [5] J. Jannotti, D. K. Gifford, K. L. Johnson, M.F. Kaashoek, and J. W. O. Jr., "Overcast: Reliable multicasting with an overlay network," in *5th Symposium on Operating System Design and Implementation(OSDI)*, Dec. 2000.
- [6] Dimitrios Pendarakis Tellium, "ALMI: An Application Level Multicast Infrastructure," In *Proc. of 3rd Usenix Symposium on Internet Technologies and Systems*, March, 2001.
- [7] Y. Chawathe and M. Seshadri, "Broadcast Federation: An application-layer Broadcast Internetwork," in *the 12th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, May 2002.
- [8] Min-You Wu, Yan Zhu and Wei Shu, "Proxy Placement for Server-Based Multicast," submitted.
- [9] GT-ITM, <http://www.cc.gatech.edu/projects/gtitm/>.