Issues in Overlay Multicast Networks: Dynamic Routing and Communication Cost

Sherlia Y. Shi Jonathan S. Turner
Applied Research Lab
Washington University in St. Louis
{*sherlia, jst*}*@arl.wustl.edu*

WUCS-*02-14*

May 28, 2002

Department of Computer Science
Campus Box 1045
Washington University
One Brookings Drive
St. Louis, MO 63130-4899

Issues in Overlay Multicast Networks: Dynamic Routing and Communication Cost

Sherlia Y. Shi Jonathan S. Turner
Applied Research Lab
Washington University in St. Louis
{*sherlia, jst*}*@arl.wustl.edu*

Abstract

Overlay networks are becoming a popular vehicle for deploying advanced services in the Internet. One such service is multicast. Unlike conventional IP multicast, which requires universal deployment of network layer mechanisms, the overlay multicast model leverages the existing unicast mechanism and offers many service flexibilities to applications. Implementing multicast without requiring network support eliminates many deployment complexities that IP multicast has faced. However, it also raises new issues in efficient network design. In an earlier paper, we studied multicast routing algorithms designed to optimize resource usage in overlay networks, when multicast session membership is static. In this paper, we study the routing performance in sessions where members can join and leave dynamically. In order to prevent service interruptions, routing in dynamic sessions cannot be as optimized as in the static case, resulting in possible performance degradation. We quantify this effect and show how it can be partially mitigated using a limited form of session rearrangement. We also study the impact of the overlay multicast approach on underlying networks, in order to quantify the difference in cost and performance of overlay multicast and "native" multicast. We demonstrate that overlay multicast is reasonably efficient. Indeed, in many cases it makes more efficient use of network resources than native IP multicast.

1. Introduction

Multicast is a data transmission mechanism that provides efficient simultaneous data delivery to a group of users. Unfortunately, the limited availability of multicast in the public Internet has led researchers to question the validity of the current model and implementations. More importantly, it is motivating a search for efficient alternatives for supporting group communication in the Internet.

One of the possible alternatives is to use an overlay network of *multicast service nodes* (MSN) that act as proxies, forwarding data to and from group members through unicast connections. Among MSNs, data is delivered along a virtual multicast tree, where each tree branch is a unicast connection. Figure 1 shows an example. Although the underlying data transmission is unicast,

this overlay multicast service network still supports the two advantages of multicast over unicast: a) it reduces the transmission overhead on the sender; and b) it reduces the overhead on the network and the time taken for all destinations to receive the data. The first advantage is clear since the sender only needs to transmit one packet to its proxy instead of one copy to each group member. The second advantage has been shown in several previous studies [3, 5, 8, 9] using simulations over various network topologies and using a wide range of multicast tree topologies.

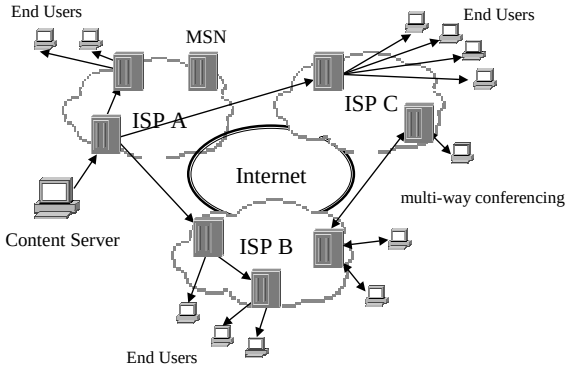


Figure 1: An Overlay Network for Multicast Services

We refer to a generic overlay multicast service as an advanced multicast architecture, or AMcast for short. In AMcast, an MSN serves a client by joining all the multicast groups that the client is interested in and forwards its packets to other MSNs in the session, which forwards packets to other MSNs and to their individual local clients. The MSNs are located (logically) at the network edges, typically co-located with ISP edge routers or co-location service providers; this gives the MSNs faster network access to multiple ISPs. MSNs in a common session, construct a shared bi-directional multicast tree for forwarding packets belonging to the session. The communication channels between MSNs and their clients, and among MSNs both leverage the existing unicast transport mechanisms, so deployment does not require changes to the end system OSs or to network routers.

Previous studies of overlay multicast focus on end-system middleware that allows a collection of hosts to form a multicast session and forward packets to session participants in a cooperative manner [5, 7–9, 15]. While highly flexible, *End System Multicast* (ESM) is constrained by the limited bandwidth typically available at the interface between LANs and WANs. This limits both the number of multicast sessions that can be supported from within a given LAN and the branching factor that can be supported at each node of a multicast tree. For large multicast groups, the limitation on branching factor translates to longer end-to-end delays, making end-system multicast less suitable for real-time applications. In addition, in today's Internet, the available routes to end systems often contain detours to the backbone network even for two nodes that are geographically close by, but in different ISP domains. These detours further increase end-to-end delay and prevent users from taking advantage of their geographical proximity. Proxy-based overlay multicast, on the other hand, allows MSNs to be placed within the network where MSNs can directly peer with multiple ISPs using high speed network access to a large number of end users. The co-location with ISPs also allows MSNs to optimize their routing paths over the underlying network topology, particularly

over the backbone network links. We believe AMcast and ESM offer complementary advantages: ESM is well-suited for small multicast groups or groups where all members have high speed network access; and AMcast offers advantages for large-scale multicast and for small or medium size groups that require real-time, high data rate communication.

In [11], we presented a heuristic multicast routing algorithm, the *Iterative Compact Tree (ICT)* algorithm, for overlay networks that optimizes the *access bandwidth* usage at the MSNs' interfaces, while satisfying applications' end-to-end delay requirements. The ICT algorithm differs from other bandwidth-based overlay routing algorithms [5, 7] in that it optimizes the bandwidth usage for a continuous sequence of session requests, while in [5, 7] each multicast tree is optimized in isolation, with little regard for impact this may have on later arriving session requests. Moreover, none of the previous studies has evaluated routing performance in the presence of dynamic sessions, where participants may join and leave the session throughout the session lifetime. Dynamic sessions have a substantial impact on routing performance, if the tree topology is not rearranged as members join and leave. Since such rearrangement can disrupt the flow of packets in a session, it is preferable (and in some cases may be required) that rearrangement be avoided. In this paper, we quantify the effects of dynamic membership changes on the routing performance of the ICT algorithm.

The paper also studies how the performance of overlay multicast compares to that of native IP multicast. Our comparison of the relative costs takes into account the geographic distances spanned by the links in the underlying network, reflecting the real monetary costs associated with links of greater physical length. We evaluated the multicast trees computed by the ICT routing algorithm and showed that the overlay multicast trees can achieve network efficiency as well as small application delay performance.

The rest of the paper is organized as follows. Section 2 presents some related work; In Section 3, we start by briefly describes the ICT algorithm and then evaluate its performance in dynamic sessions; In section 5, we study the communication cost and other performance characteristics of overlay multicast trees relative to native IP multicast. We conclude in Section 6.

2. Related Work

There are many application-level multicast services appearing in the recent literatures, mostly due to the dwindling usage of the Mbone and the slow deployment of network multicast services. The flexibility of application-level multicast services allow the routing policy to be changed based on the target application requirements. For example, Scattercast [3] uses delay as the routing cost and builds shortest path trees from data sources; Overcast [7] explicitly measures available bandwidth on an end-to-end path and builds a multicast tree that maximizes the available bandwidth from the source to the receivers; and Endsystem multicast [5] uses a combination of delay and available bandwidth, and prioritizes available bandwidth over delay when selecting a routing path. In [9, 15], each application node is assigned a hash identifier and a session is routed based on the bit differences in the node identifiers. The rational behind these *Distributed Hash Table (DCT)* approaches are that they are able to scale to groups of very large size with each group member keeping relatively small size of neighbor information. However, none of these schemes has considered the routing optimization problem over a continuous sequence of multicast sessions or evaluate routing performance with dynamic member joining and leaving the sessions. These two aspects are particular important

to the understanding of the overall network utilization if overlay multicast is to be implemented as a service level infrastructure that is explictly provisioned and managed.

In the ESM context, [4, 9, 15] presented simulation results on the relative effects of overlay multicast to the network level multicast schemes, on the underlying networks and on the application perceived delay performance. All these studies used generated topologies with random link delays. In the AMcast model, the paths in an overlay multicast tree are more influenced by the geographic distance between the MSNs, since intra-ISP paths are typically less circuitous than the end-to-end paths across multiple ISPs. Additionally, all the previous works focus on one specific type of multicast trees, e.g. a shortest path tree, constructed in a static fashion. In our evaluation, we consider a range of different type of multicast trees, ranging from a star to a path, that bound the worse case behavior of all potential overlay trees.

3. Dynamic Routing in Overlay Networks

In AMcast, an end user sends join and leave requests to its proxy MSN and these requests are handled locally by the MSN as whether to create or destroy a connection from the proxy MSN to the user. An MSN joins all sessions that at least one of its users wishes to participate. The routing procedure creates a shared multicast tree among all participating MSNs in a session. There is a delegated MSN for each session that computes the multicast tree and directs other MSNs to form a tree. This choice of centralized computation is necessary to achieve message efficiency by eliminating the need of message exchanges required to coordinate a distributed computation. We should point out that this centralized computation does not create a single spot of failure, since each MSN is potentially a delegate for some number of sessions and the overall computation load is distributed across all MSNs. During periods of heavy network load, we can expect there to be lots of session routing computations being performed concurrently. This means that the overall computational load can be effectively distributed by having different servers do the computation for different sessions. We believe that the greater efficiency of this approach, relative to a distributed routing computation for each session, will more than compensate for any inequities in the load distribution that are likely to arise in practice.

This section reports our new findings on the routing performance of the *Iterative Comapct tree* (ICT) algorithm in dynamic sessions. We first briefly describe the ICT algorithm to provide the necessary background information. More details about the overlay routing algorithms can be found in [11].

3.1. The Iterative Compact Tree Algrorithm

Multicast routing in overlay networks involves building a tree spanning a set of MSNs. The objective of the routing algorithm is two-fold: first, optimize the resource usage on the access links of the MSNs; and second, limit the delay in a multicast session by avoiding excessively long and circuitous routes. The use of access bandwidth at an MSN is determined by its node degree in the multicast tree, i.e. the number of simultaneous connections that an MSN must support for each session; in particular, the routing procedure must never exceed the available interface bandwidth at an MSN.

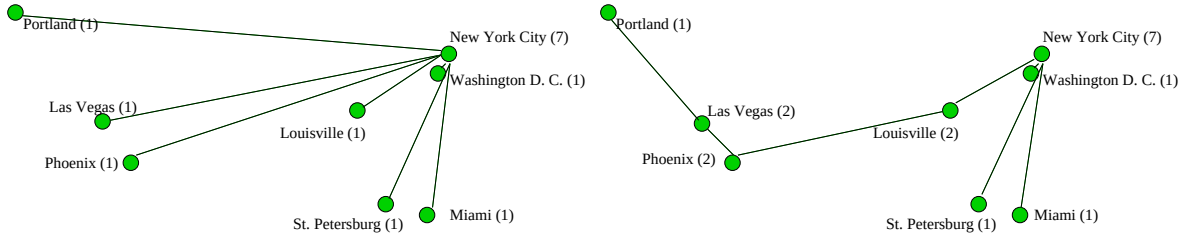
The minimization of the maximum session delay is achieved by constraining the multicast tree diameter.

One natural formulation of the routing problem to seek the "most balanced" tree, that satisfies an upper bound on the tree diameter. To explain what is meant by "most balanced", we define the *residual degree* at node v with respect to a tree T as $res_T(v) = d_{max}(v) - d_T(v)$, where $d_{max}(v)$ is the maximum degree that can be supported at node v for a given multicast session and $d_T(v)$ is the degree of v in T. The value of $d_{max}(v)$ is calculated as the available interface bandwidth at v, divided by the bandwidth of the given multicast session. To reduce the likelihood of blocking a future multicast session request, we choose trees that maximize the smallest residual degree. Since the sum of the degrees of all multicast trees is the same, this strategy works to "balance" the residual degrees of different vertices. Any tree that maximizes the smallest residual degree is called a "balanced" tree.

```
Input:  A set of MSNs k = |V|;
   degree constraints d_max(v);
   edge length c(u,v);
   a diameter bound D ∈ Z+;
Output:  tree T

Sort nodes in non-decreasing order of radius max_w c(v,w) as v_1, v_2, ... v_k

/* Balance degree allocation */
initialize all d_A(v) = 1, res_A(v) = d_max(v) - d_A(v)
while Σ_v d_A(v) < 2(k-1) do
  find u of max(res_A(v))
  increment d_A(u), decrement res_A(u)

/* Build small diameter tree */
round = 0
while round < threshold do
  find T of smallest diameter w.r.t. d_A(v)
  if diameter(T) > D then
    i = (round * b) mod k
    increment d_A(v) for v_i ... v_{i+b}
  round = round + 1

/* Adjust degree constraint */
if diameter(T) ≤ D then
  foreach v ∈ V do
    d_max(v) = d_max(v) - d_T(v)
```

Figure 2: Outline of the ICT Algorithm

Figure 2 outlines the algorithmic procedure. The ICT algorithm starts by determining the ideal degree of each node in the multicast session with respect to the objective of maximizing the smallest residual degree. We call this procedure *Balanced Degree Allocation (BDA)*. The total degree required for a multicast session of size k is $2(k-1)$ and each node must have a degree of at least one. After each node is allocated a degree of 1, the remaining allocation of $k-2$ is distributed



(a) Longest Path: Portland → New York → Phoenix; Length = 9416.67 km

(b) Longest Path: Portland → Las Vegas → Phoenix → Louisville → New York → St. Petersburg; Length = 7823.64 km

Figure 3: An Example of the ICT Algorithm with Degree Adjustment

among the nodes by repeatedly adding one to the allocation of the node with the largest residual degree (with respect to the partial allocation). This allocation not only maximizes the smallest residual degree, it also achieves the most balanced possible set of residual degrees. Next, the ICT algorithm attempts to construct a small diameter tree in which the vertices have the assigned degrees. The tree building procedure repeatedly selects a pair of vertices u and v, where u is in the partial tree constructed so far, and v is not. To be eligible for selection, the degree allocation at u must exceed its current degree in the partial tree. The selected vertices are chosen so that the addition of the edge (u, v) will minimize the diameter of the resulting tree. This procedure is repeated for every choice of initial vertex, and the smallest diameter tree is kept.

Unfortunately, this procedure may not produce a tree with the desired diameter. When this occurs, the ICT algorithm systematically "loosens" the degree allocation, and re-tries the tree building procedure. The "loosening procedure" may be repeated, if necessary until a tree of small enough diameter is found. The degree loosening procedure increments by 1, the degree allocations of the b "most central" vertices, where b is a parameter of the algorithm (degree allocations are not incremented for those vertices v already at the limit of $d_{max}(v)$). A vertex u is more central than a vertex v if its radius $max_w c(\{u, w\}) < max_w c(\{v, w\})$. If additional applications of degree loosening are needed, they proceed with the next group of b vertices, going from most central to least central. Figure 3 shows an example. For simplicity, we used geographical distance as routing cost and a diameter bound of 8000 km. Initially, the BDA strategy allocates degrees that only allow the creation of a star topology centered at New York City. However, this tree exceeds the diameter bound. In the degree loosening procedure, the central nodes: Las Vegas, Phoenix and Louisville, are allocated one more degree, allowing a new tree to branch at these nodes. This new tree has a smaller diameter that satisfies the bound and the actual degree usage at the session nodes is still close to the balanced degree allocation. A more detailed description can be found in [11].

3.2. Evaluation Methodology

To evaluate the routing performance of the ICT algorithm, we performed simulations in which multicast sessions are dynamically created, modified and destroyed. The primary performance metric of interest is the fraction of requests that are rejected, because the network is unable to configure

the session with the resources available. There are many possible configurations for the underlying network topologies, the traffic distributions and the session configurations. The choices made here, while not comprehensive, are representative of realistic network configurations and provide useful insights into the effects that key parameters have on the routing performance.

Network Configuration The underlying network topology used in the simulation, spans the 50 largest metropolitan areas in the United States [13]. From the perspective of the overlay multicast service provider, the network is fully connected, since AMcast builds on top of Internet and each MSN can reach others via unicast connections. The geographic distances between MSNs is taken as the edge cost. The diameter bound is fixed at 8000 km, about 1.5 times the largest inter-city distance.

Traffic Configuration The "traffic density" at each node is proportional to the population of the metropolitan area it serves, so MSNs at larger cities are more likely to participate in a session. We use a Poisson session arrival process, and the session holding times follow a Pareto distribution. We select session size (or, interchangeably session fanout) from a binomial distribution with mean of k and vary k to change the traffic configuration.

For simplicity, all multicast sessions are assumed to have the same bandwidth. Different MSNs were assigned different interface bandwidths, depending on their traffic density and their location. MSNs in more central locations are assigned higher interface bandwidths than those in less central locations, since it is more efficient for multicast sessions to branch out from these locations than from the more peripheral locations. The assignment of interface bandwidth at MSNs is critical to the performance of the routing algorithms. We have dimensioned the network to best carry a projected traffic load, given the specific routing algorithm. The procedure used to perform the dimensioning is described in [12].

Session Configuration The behavior of dynamic sessions depends largely on the applications. For example, a conferencing application may have a large number of members joining the session at the beginning and staying throughout the session; while an on-line chat room may constantly have members joining and leaving the session. Instead of conjecturing any specific profiles, we simply start each session with a random initial session fanout selected from a binomial distribution and generate a number of join and leave requests that are uniformly distributed throughout the session lifetime. In order to isolate the influence of session dynamics, we keep the average session fanout constant at 10, which is the session fanout for which the network was dimensioned. This is done by varying the initial session fanout, so that the average of the initial and final fanouts is 10. Each curve is annotated with a triple of values representing the *(initial fanout, number of join requests per session, number of leave requests per session)*.

4. Routing with Dynamic Sessions

An MSN supporting a multicast session can often handle dynamic membership changes locally. Specifically, when a new user joins a session that an MSN is already supporting (because another of its clients is participating), the addition can be handled locally, without affecting any of the other MSNs. Similarly, the departure of a participant can be handled locally, so long as there are other

clients of the MSN that are still participating. However, in other cases an MSN may need to interact with other MSNs to satisfy join and leave requests on the part of its clients, resulting in changes to the multicast tree topology. There are fewer choices available to the network, when making such adjustments to the tree topology. To avoid disrupting the flow of packets among session participants, it is desirable (and at least for some applications necessary) to avoid large-scale changes to the tree topology. The least disruptive approach is to add a new MSN, by just adding a connection between it and another MSN that is already supporting the session. Similarly, a departing MSN is removed from the tree when it no longer has any clients participating in the session, and it is a leaf in the multicast tree. This approach ensures that packet flows are not disturbed by membership changes. The requirement that an MSN with no clients, remain in a multicast tree if it is not a leaf, is important for session continuity, but does have a negative impact on overall network performance. This effect is quantified by the results presented below. We also show that a significant improvement in performance can be obtained if an MSN with no clients is permitted to drop out of a multicast tree if its degree in the tree drops to two. For this case, we can make a simple adjustment to the tree topology that permits packet flow continuity to be maintained, using special procedures for packet handling during the transition.

When adding a new member to an existing multicast tree, we apply the same strategy as the ICT algorithm. We first identify the set of nodes in the tree at which the addition of an edge to the new MSN would not violate the diameter bound. We then add a connecting edge at the node in this set with the largest residual degree. Ties are broken by selecting the node that results in the smallest diameter tree.

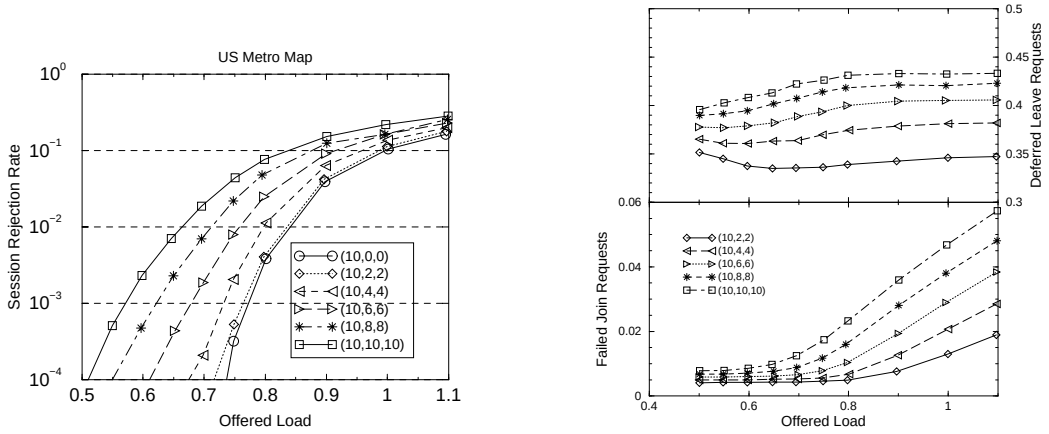


Figure 4: Routing Performance with Dynamic Sessions

We measure the routing performance as the fraction of requested sessions that are rejected. Figure 4 shows the performance of the ICT algorithm with an equal number of join and leave requests in each session. Each session starts with an intial fanout, which appears as the first number in the triple of values identifying each curve. Join and leave requests are distributed randomly over the time interval of the session. The number of join and leave requests appears as the second and third values identifying each curve. We expect many applications to display a different sort of profile, with most joins occuring near the start of the session and most leaves occuring near the end. Since the performance impact of leaves is much greater than that of joins, such applications

are not greatly affected by dynamic membership changes. The random distribution was chosen to emphasize a more challenging case, although not the most extreme.

The plot shown at the left of the figure shows the fraction of sessions that are rejected at initialization; the fraction of rejected join requests is shown separately in the right side bottom figure. As expected, the more dynamic membership changes there are, the worse the routing performance becomes. The number of failed join requests increases with the offered load, since at high load join requests are likely to arrive at nodes that are out of capacity. There are about $0.5\% - 1\%$ failed join requests at the lightest traffic load for each session configuration. This is because a new node, regardless of its location, always joins the tree as a leaf node. When a session is initialized with nodes on both coasts only, a node in the more central location joining the session tree is likely to add excessive distance to the tree diameter and to violate the diameter bound (8000 km). Regardless of the offered load, there is a small fraction of sessions that have such configurations, resulting in the flat tails for all curves. A larger diameter bound would greatly reduce this effect, suggesting that it might be appropriate to relax the diameter bound for dynamic joins.

At higher offered loads, the fraction of failed join attempts is generally smaller than the fraction of sessions that fail when starting up. At an offered load of 70%, the fraction of failed joins remains below 1.5%, while the number of failed session requests is above 5% for the most dynamic case. The top figure on the right side shows the fraction of leave requests for which an MSN with no local clients was compelled to remain in the session because its degree in the multicast tree was greater than one; we call this a "deferred leave request". The fraction of deferred leave requests is significantly higher than the fraction of failed join requests. As these deferred leave requests contribute to the network usage but not to the offered load, they are likely to be the main cause of the routing performance degradation.

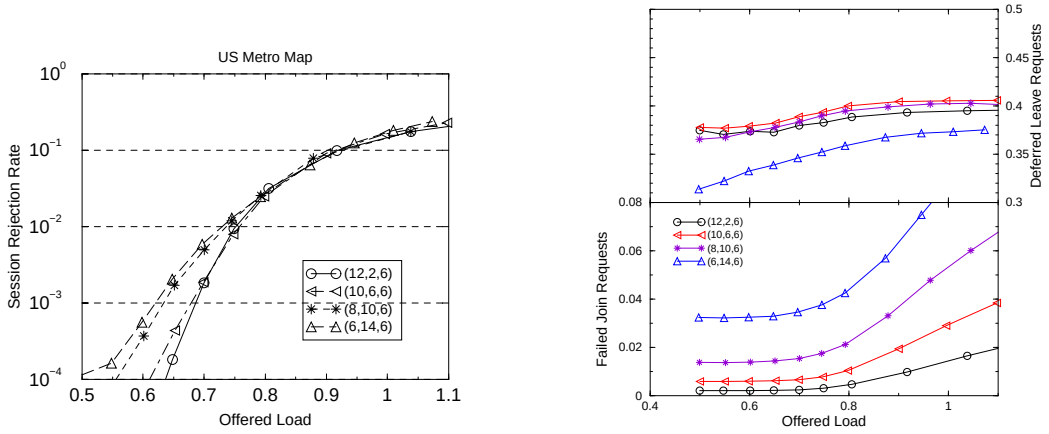


Figure 5: Effect of Dynamic Join Requests

Figures 5 and 6 examine the effects of dynamic join requests and leave requests, respectively. In Figure 5, the number of leave requests is held constant, while the number of join requests is varied from 2 to 14. To keep the average session fanout fixed, the initial fanout is varied. (Without this adjustment to the initial session fanout, the average fanout would vary, clouding the meaning of the results, since the average session fanout itself, has a significant effect on performance.) Similarly,

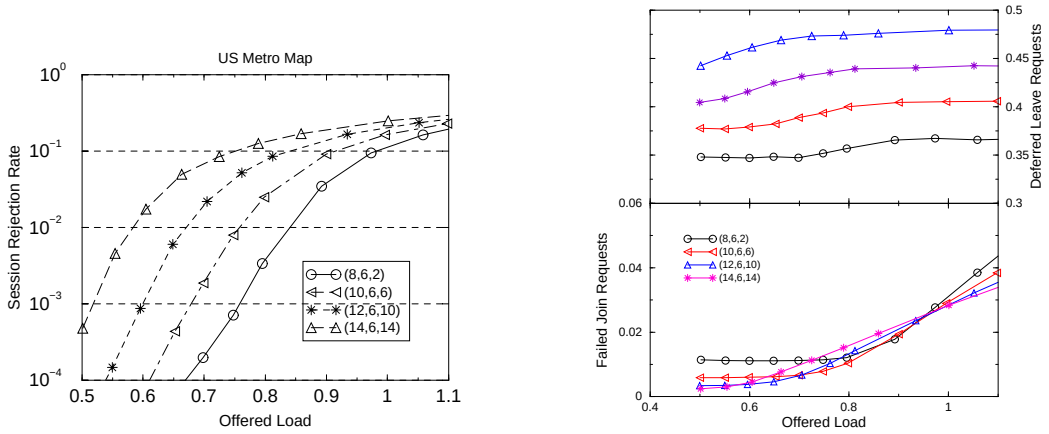


Figure 6: Effect of Dynamic Leave Requests

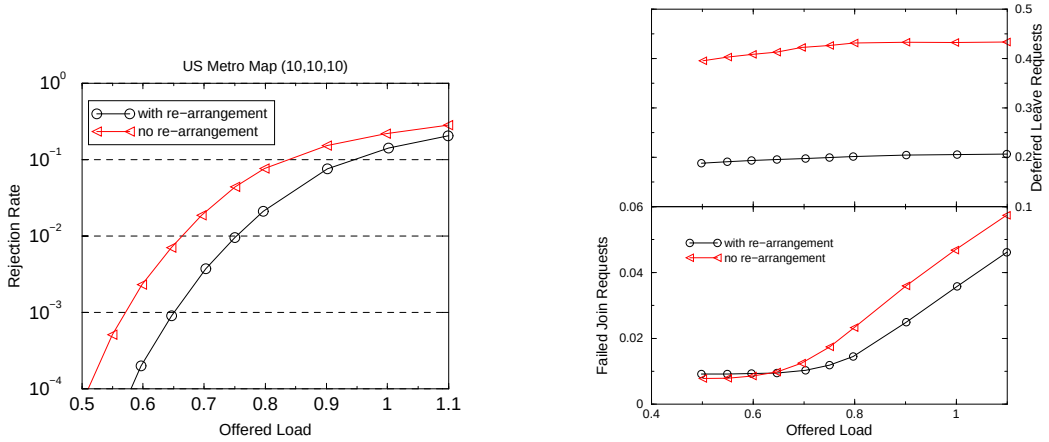


Figure 7: Performance with Tree Re-arrangement

in Figure 6, we vary the number of leave requests from 2 to 14 while holding the number of join requests fixed. Comparing Figure 5 and 6, we clearly observe that the performance degradation caused by the leave requests is much larger than that caused by the join requests. This suggests two things. One is that the ICT algorithm can tolerate the less balanced bandwidth usage caused by the incremental joins quite well. The other is that the restriction of only allowing leaf nodes to drop out of a tree has a big effect on the overall performance. The top curve in Figure 6 shows that nearly 50% of nodes remained in the session even when they were no longer supporting any local clients. This suggests that it may be important to allow some limited rearrangement following leaves.

Figure 7 shows how the performance changes when MSNs with no clients are permitted to drop out of a multicast tree when their degree drops to two. The utilization at which the session blocking probability is equal to 1% increases from about 65% to about 75%, suggesting that a network that allows limited rearrangement can carry about 15% more traffic than a network that does not (assuming a target session rejection rate of no more than 1%).

5. Cost of Overlay Networks

Since the ICT algorithm seeks to optimize the usage of interface bandwidth at MSNs and to limit session delay, it cannot provide any guarantees on the characteristics of the trees it constructs that affect the performance of the underlying network. Depending on resource availability, the characteristics of overlay routing trees can vary significantly. When the load on the MSNs is light, the ICT algorithm tends to create small diameter trees, often a star centered at the node with the highest available bandwidth. Under heavier loads, it tends to produce trees that have more a more even distribution of node degrees. In particular, the trees may have relatively few branching points, and many nodes with two incident edges. The impact of such trees on the underlying network can differ significantly. In this section, we study how the overlay multicast trees created by the ICT algorithm map onto an underlying network, and the resulting loading effects on the network. We also study how this mapping may effect application performance using the following metrics.

Transmission Cost: The transmission cost measures the average network cost of sending a packet from one group member to the rest of the group. It includes the link cost from each multicast client to its designated MSN and the link cost of the multicast tree joining the participating MSNs. As the overlay tree branches at the MSNs rather than at the network routers, the overall transmission cost includes the cost of multiple traversals on some of the physical network links.

Link Stress: The stress on a link measures the number of duplicate packets that travel over that link. The overlay multicast approach can create such duplicate links because the copying occurs within the MSNs, not within the routers. In the reported results, we do not count duplicate packets on the access links connecting MSNs to the underlying network.

Relative Delay Penalty: The RDP is the ratio of the delay between a pair of members along the overlay tree and the delay over their network shortest path. This measures the magnitude of the detour that is taken by a packet sent over the overly tree.

Session Delay Penalty: The SDP is an alternative (and arguably more relevant) measure of application delay performance. The SDP is the ratio of the maximum delay on the tree path between two nodes in a multicast session to the maximum network delay between any pair of nodes in the session. This measures how much worse the worst tree path is to the worst intrinsic delay among the nodes in the session.

5.1. Simulation Topology and Setup

It is difficult to construct a network topology that is representative to the current Internet. Popular topology generators such as GT-ITM [14] assumes certain network hierarchies and generate random graphs for each network layer. Recently the University of Oregon Route Views Project [10] has provided many researchers with access to part of the global routing table exported from about 40 different Autonomous System domains, and which constructs an AS-level network map.

Unfortunately, neither of these approaches captures the geographic properties of the Internet. The cost of network links is intrinsicially a function of the geographic distances spanned those links,

and for backbone links, delays are largely determined by the geographic distance, since queueing delays on the high speed backbone links tend to be small. For this reason, we have chosen a network topology that explicitly seeks to capture the geographic characteristic of a real-world network. We have also used geographic distance as a measure of both link cost and delay.

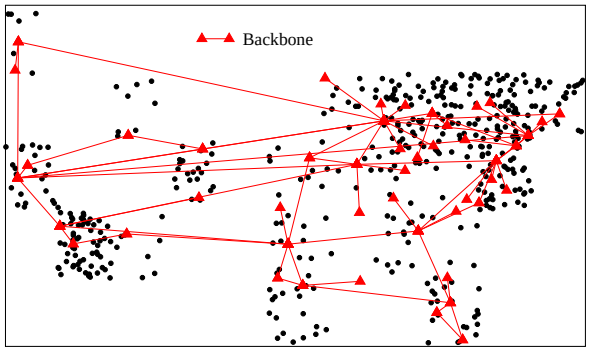


Figure 8: Metro Network Topology

Node Distribution Figure 8 depicts the network topology used in our simulation. The configuration (called the metro topology) contains backbone routers at each of the 50 largest metropolitan areas in the United States. Another 450 nodes are distributed among the 50 metropolitan areas in proportion to their populations.

Network Connectivity The connecting links were selected as follows: first a backbone network spanning the 50 cities was configured by using links from the AT&T backbone map [1]. Next, we added "star links" connecting each local node to its nearest backbone node. Finally, we computed a network level minimum spanning tree (MST) over the entire set of nodes and added the resulting links to the network. This last step was done to provide some routing diversity for local nodes, so they weren't completely limited to the paths through their backbone node.

Client and Server Placement We placed MSNs at each of the 50 backbone cities, as these are places where most data centers are likely to be located. For each multicast session simultated, we randomly selected a number of multicast clients among all nodes. A client is assigned to its closest server and only those servers that have attached clients participate in the multicast session.

We simulated the dynamic arrival and departure of multicast sessions over time and compared the overlay multicast trees created by the ICT algorithm with two network level multicast trees: **Steiner tree** – A Steiner tree is the optimal multicast tree in terms of total cost, however, the computation of the optimal Steiner tree is NP-complete [6], so we compare to an approximate Steiner tree computed using the well-known MST heuristic, which has a worst-case approximation ratio of 2 [2] and which typically produces near-optimal trees. **Shortest path tree** – The shortest path tree is widely used in IP multicast and in some of the application-level multicast schemes [3, 5]. Each sender in a multicast group uses its own source-rooted shortest path tree. The shortest path tree is the best tree from the standpoint of network delay, but can make very inefficient use of network bandwidth, especially in richly connected network topologies. It also requires the maintenance of far more state information in the network for multicast trees (quadratic, rather than linear). When comparing the transmission cost, we use the average cost of all the shortest path trees in a session.

5.2. Evaluation of the Trees Produced by ICT

In each simulation run, we generate dynamic session arrivals and departures. For each session, we randomly select a number of clients out of the entire 500 nodes and assign each client to its nearest MSN. We use the ICT algorithm to create an overlay tree among the session MSNs. Each tree, including the overlay tree branches and the connections from the clients to the MSNs, is evaluated according to the four metrics. For the graphs presented here, we used a fixed session size of 100 nodes. The results for other session sizes deviate little from this set, since most MSNs (an average of 40 out of the total 50 MSNs) already participate in each session, and the variation of the session size only affects the number of connections from the clients to the MSNs, which has less influence on the results. We vary the x-axis value as the offered load to the MSNs to examine the impact of the load factor on the output trees from the ICT algorithm. The capacity assigned to each MSN and the diameter bound were configured as in the previous section.

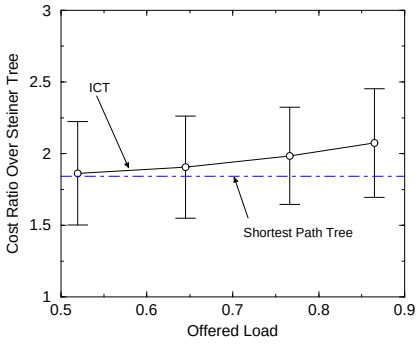
Figure 9(a) shows the relative cost of the trees generated by the ICT algorithm and the shortest path tree, to the approximate Steiner tree cost. The average cost of the shortest path trees holds constant at about 1.8 times the Steiner tree cost. The average ICT cost rises slowly with the increase of network load, from 1.8 to 2.1 times the Steiner tree cost. This shows that the ICT algorithm uses network resources nearly as efficiently as the shortest path tree, although both clearly deviate significantly from the ideal presented by the Steiner tree.

Figure 9(b) shows the average and maximum link stress for the ICT overlay trees. Here, we assume MSNs are co-located with backbone routers and only show the stress on the backbone links since they are more expensive resources. There are a total of 68 backbone links connecting the 50 cities. On average, the number of duplicate packets carried by each backbone link is less than 1.5. For network level multicast trees, the link stress is always one.

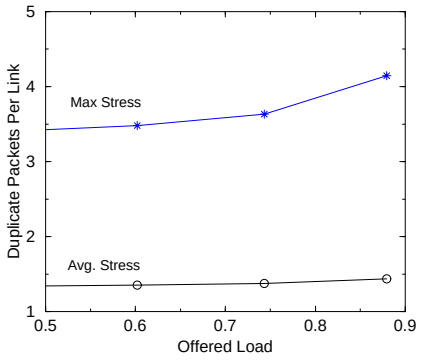
Figure 9(c) shows the average and the 90th percentile of the RDP value for the overlay trees and the average value for the approximate Steiner tree. The shortest path trees have RDP of one, however in any general topology, no single tree can optimize the delay between every pair of members due to the existence of alternate routing paths. The average value for the ICT trees is slightly higher than that of the approximate Steiner trees, while the maximum RDP value is much higher than that of the Steiner tree, which is about 2.2 (not shown). We observe that the large RDP values mostly occur between members that have small network delay but relatively large delay along the overlay tree. The absolute delay is bounded at 8000 km by the ICT algorithm.

Figure 9(d) shows the SDP values for the ICT trees and the approximate Steiner trees. Again, the SDP value for shortest path trees is always one. Because the ICT algorithm uses a delay bound on the tree diameter, its SDP value is actually smaller than that of the Steiner tree and acceptably close to the ideal achieved by the shortest path trees.

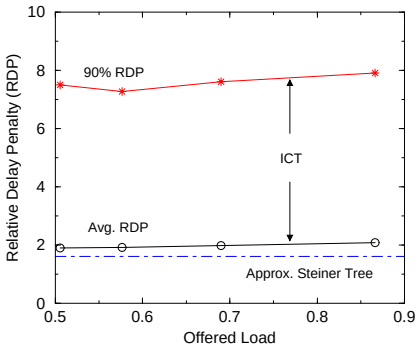
We have not compared our results directly with the results previously reported for application layer multicast. The main disadvantage of application layer multicast, as opposed to overlay multicast, comes from the existence of multiple independent AS domains that restricts the available routes to applications, while in the overlay model, MSNs can directly peer with backbone routers, resulting in the overlay topology to better align with the underlying network topology. This partly explains the values reported here are somewhat smaller than reported previously in [4, 9].



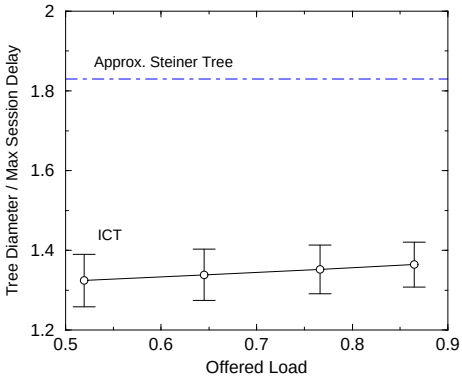
(a) Relative tree cost to the approximated Steiner tree cost.



(b) Average and maximum link stress.



(c) Relative delay performance for pairwise session members.



(d) Delay ratio of tree diameter to the maximum network delay in a session.

Figure 9: Evaluation of Trees Output from the ICT Algorithm

We conclude in this section that with a careful design and prudent engineering practice, not only is overlay multicast feasible, it is also as efficient, in terms of overall tree cost, as the shortest path trees used in the IP multicast model.[1] The RDP performance, on the other hand, is somewhat less encouraging. One particular concern is that two locally closeby clients can not take advantange of their proximity in the overlay model. Although for most applications, it is not critical to have the smallest delay possible as long as the maximum can be bounded, for applications that can take advantage of this speedup for part of its groups, it is conceivable to have some form of multicast gateway services that brige the native multicast in local networks to the overlay multicast session. We will leave this as future work.

[1]The more recent IP multicast as in PIM-SM uses a shared multicast tree which is a shortest path tree rooted at the RP. The optimal placement of the RP in general is NP-hard. The common engineering practice is to place it close to the source, thus the overall cost is still in the ball park of our study.

6. Conclusion

Multicast can be provided either as a basic network service or at a higher level. In this paper, we studied the routing behavior in overlay multicast networks with dynamic sessions and evaluated the communication cost of the overlay networks with a variety of multicast trees. We showed that the overlay network model has the advantage of adopting optimal topology design and route selection mechanisms that results in greater service flexibility without sacrificing much efficiency. In fact, its routing algorithms can create trees that are more efficient than in the traditional IP multicast model. On the other hand, the amount of duplicate packets and delay penalties are kept small. The ICT algorithm, therefore allows service providers to optimize the utilization of their overlay networks, while keeping the network overhead small.

We are currently conducting analysis on the performance bound of the routing algorithms. This will complement our current evaluation of the routing algorithms using simulations. Another direction of the future work is to investigate the system architecture of providing the overlay network services. Without re-inventing the wheel, we are looking into existing programmable router platforms for the possibility of integrating with our overlay service.

References

[1] AT&T U.S. Network Map. http://www.ipservices.att.com/backbone.

[2] Bharath-Kumar and J. M. Jaffe. Routing to Multiple Destinations in Computer Networks. *IEEE Transactions on Communications*, 31(3):343–351, March 1983.

[3] Y. Chawathe. *Scattercast: An Architecture for Internet Broadcast Distribution as an Infrastructure Service*. PhD thesis, University of California, Berkeley, August 2000.

[4] Y. Chu, S. Rao, and H. Zhang. A Case For EndSystem Multicast. In *Proceedings of ACM Sigmetrics*, Santa Clara, CA, June 2000.

[5] Y. Chu, S. G. Rao, S. Seshan, and H. Zhang. Enabling Conferencing Applications on the Internet Using an Overlay Multicast Architecture. In *Proc. ACM SIGCOMM 2001*, San Diego, CA, August 2001.

[6] M. R. Garey and D. S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness*. San Francisco : W. H. Freeman, 1979.

[7] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O. Jr. Overcast: Reliable Multicasting with an Overlay Network. In *Proc. OSDI'01*, 2000.

[8] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel. ALMI: An **A**pplication **L**evel **M**ulticast **I**nfrastructure. In *3rd Usenix Symposium on Internet Technologies and Systems (USITS'01)*, San Francisco, CA, March 2001.

[9] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Application-level Multicast using Content-Addressable Networks. In *Proc. 3rd International Workshop on Networked Group Communication (NGC)*, November 2001.

[10] University of Oregon Route Views Project. http://www.routeviews.org.

[11] S. Shi and J. Turner. Routing in Overlay Multicast Networks. In *Proc. of IEEE INFOCOM'02*, June 2002.

[12] S. Shi, J. Turner, and M. Waldvogel. Dimension Server Access Bandwidth and Multicast Routing in Overlay Networks. In *11th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'01)*, June 2001.

[13] U.S. Census Bureau. http://www.census.gov/population/www/estimates/metropop.html.

[14] E. W. Zegura, K. Calvert, and S. Bhattacharjee. How to Model an Internetwork. In *Proc. of IEEE INFOCOM*, San Francisco, CA, 1996.

[15] S. Zhuang, B. Zhao, A. D. Joseph, R. H. Katz, and J. Kubiatowicz. Bayeux: An Architecture for Wide-Area, Fault-Tolerant Data Dissemination. In *Proc. NOSSDAV'01*, June 2001.