Island Multicast: The Combination of IP Multicast with Application-Level Multicast

K.-W. Roger Cheuk S.-H. Gary Chan
Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon
Hong Kong
Email: {rcheuk, gchan}@cs.ust.hk

Jack Y.-B. Lee
Department of Information Engineering
Chinese University of Hong Kong
Shatin, New Territories
Hong Kong
Email: yblee@ie.cuhk.edu.hk

Abstract— **The Internet nowadays consists of multicast-capable domains or "islands" interconnected by multicast-incapable routers. In order to achieve efficient global multicast, we propose and study Island Multicast (IM) where overlay connections are used between islands while IP multicast is used within an island. IM may use any existing application-level multicast protocol to build island overlay. We describe how to elect a representative (or leader) in each island for such a purpose. We also present the mechanisms for electing the bridging nodes for overlay connections. Using Internet-like topologies, we show that IM achieves much higher bandwidth efficiency as compared to using application-level multicast alone, at the cost of a small increase in end-to-end delay.**

I. Introduction

With the availability and penetration of multicast-capable routers, in today's Internet local networks are generally multicast-capable. These multicast domains or "islands" are interconnected by routers which are either multicast-incapable or multicast-disabled (for security or traffic control purposes). In order to achieve global multicast in this environment, application-level multicast (ALM) has recently been proposed. In ALM, group members form an overlay network and content is distributed via unicast by relaying packets from one member to another. This approach, though works well, has not taken advantage of the local multicast capability of an island and hence is not very efficient.

As IP multicast is generally more efficient (in terms of both bandwidth and end-to-end delay), it would be beneficial if ALM can make use of such capability in building multicast trees. We hence propose and investigate a scheme called Island Multicast (IM) that integrates IP multicast with ALM. In IM, a host uses IP multicast if it is in a multicast-capable island and uses unicast to form overlay between islands to achieve global multicast.

One strength of our IM is that it may use any ALM protocols to construct the island overlay tree. We illustrate the operation of IM in Fig. 1, where seven hosts (R1 through R7) belong to the same multicast group on three IP multicast islands [1]. I1 serves as the parent of both I2 and I3. R1 and R3 are the bridge nodes forming an overlay connection for inter-island data delivery from I1 to I2. Likewise, R2 and R6 are the bridge nodes for data delivery from I1 to I3. We call R3 the neighbor bridge node of R1 and R1 the neighbor bridge node of R3. The sender uses IP multicast to send packets to receivers R1 and R2, which in turn forward the packets to islands I2 and I3, respectively. R3, upon receiving a data packet from I1, multicasts it to R4 and R5 by IP multicast. Similarly, R6 forwards the packets to R7 using IP multicast.

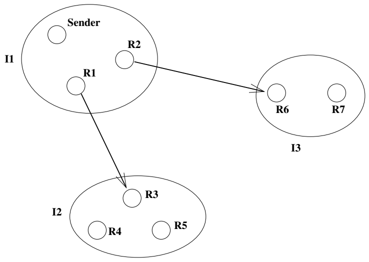

Fig. 1. An example of Island Multicast.

There are several issues that need to be addressed in IM, e.g., how to build an efficient loop-free multicast tree and how hosts join or leave a multicast group. The group members in an island also need to elect a representative (or leader) in order to build an overlay tree between islands. We present a leader election mechanism here. Furthermore, we also need to address the issues of electing bridge nodes and recovering faults.

Using Internet-like topologies, we show by means of simulation that IM indeed can significantly improve bandwidth efficiency as compared to ALM alone, at the cost of a small increase in end-to-end delay. However, contrary to the expectation, when the number of multicast routers in the network increases, the end-to-end delay may actually *increase* before it decreases.

We briefly review previous work as follows. Many proposed ALM protocols such as NICE, DT, etc., assume none of the routers are multicast-capable and hence do not make use of IP

This work was supported, in part, by the Areas of Excellence (AoE) Scheme on Information Technology funded by the University Grant Council in Hong Kong (AoE/E-01/99) and the Competitive Earmarked Research Grant from the Research Grant Council in Hong Kong (HKUST6156/03E).

[1] In this paper, we use node, host and member interchangeably.

multicast capability [1], [2], [3], [4], [5]. Our work addresses how to make use of local multicast capability in overlay construction to improve the efficiency. Though IM is similar to Scattercast and Yoid in terms of having a representative in each island, we study the selection of bridge nodes as well in this paper [6], [7].

The approaches of UMTP, Mtunnel, AMT are similar to IM [8], [9], [10]. However, they require manual configuration of inter-host connections, while IM does not require that. IM is similar to the framework of Universal Multicast (UM) [11]. However, we study and compare many other bridge-node selection algorithms. The major complexity of UM comes from eliminating routing loops, while IM is inherently loop-free and hence is simpler. Subnet Multicast (SM) also makes use of local multicast capability [12]. However, it is based on a star topology with only one level of overlay tree. This increases the stress of the network. IM is tree-based and hence achieves much better stress performance.

This paper is organized as follows. We first present IM in detail in Sect. II. We then evaluate IM in Sect. III and conclude the paper in Sect. IV.

II. Island Multicast

In this section, we first give an overview of IM. We then present in details the basic mechanisms of IM in terms of join and leave mechanisms, leader elections, bridge improvement, and fault recovery.

A. Overview of Island Multicast

Island Multicast (IM) organizes the members of a multicast session (or group) into an "island overlay." It is a two-level architecture. The upper level concerns delivery between "islands" while the lower level concerns the delivery among members in an island. At the upper level, IM constructs a logical tree to connect the islands. To construct the tree, each island elects one representative or "leader" to run an overlay multicast protocol. Given an inter-island overlay tree, a pair of bridge nodes is then selected for every pair of neighboring islands. These pairs of nodes take the responsibility of inter-island unicast delivery.

In IM, all non-leader nodes join two multicast groups: the DATA group for sending and receiving data and the ALL_MEMBERS group for sending control messages to all members in the same island. For leader nodes, in addition to joining the DATA and ALL_MEMBERS groups, they join the LEADER group for the communication between a member and itself.

Packet forwarding at each node is based on a set of simple rules. Data packets generated from a sender are first sent to the DATA multicast group (if there are other nodes in the island) and to all the nodes on its bridge connectivity list (which contains all its neighbor bridge nodes) via unicast. When a non-sender node receives a data packet, it forwards the packet to all the nodes on its bridge connectivity list (but not to the neighbor bridge node that sends to it, if the packet is from a neighbor island). If that packet is from a neighbor island instead of from the DATA group, the packet is also sent to the DATA IP multicast group if there is more than one member in the island.

In IM, the decision as to which node in an island becomes the effective bridge node to a neighbor island is made by the leader of the island (but the process of computing candidates can be either distributed or centralized). It maintains the current pairs of bridge nodes for inter-island delivery between the island and neighboring islands. These pairs of bridge nodes and the IP addresses of the neighbor islands' leaders are distributed via HEARTBEAT messages to nodes in the island. A non-leader node, upon receiving a HEARTBEAT message, updates its bridge connectivity list.

B. Join and Leave Operations

A joining host has to first determine whether a leader already exists. If not, it declares itself to be a leader. A joining host first subscribes the ALL_MEMBERS group and sends a JOIN message to the LEADER group and waits for a reply. If there is a leader in the island, it immediately multicasts a HEARTBEAT message to the ALL_MEMBERS group as a response to the JOIN message. Upon receiving the HEARTBEAT message, the joining host knows the existence of the leader. If the host does not receive any HEARTBEAT message after sending a certain number of JOIN messages, it declares itself to be the leader. A leader periodically advertises itself by sending a HEARTBEAT message to the ALL_MEMBERS group. It also becomes the bridge node to all neighboring islands and joins the LEADER groups.

Regarding the leave operation, if the node is a leader, it multicasts a LEAVE message to the ALL_MEMBERS group to trigger the leader election process (described later). Otherwise, it multicasts a LEAVE message to the LEADER group to inform the leader of its leaving. A leader, upon receiving a LEAVE message, insert all nodes on the bridge connectivity list of the leaving node into its bridge connectivity list.

C. Leader Election

Leader election process is executed whenever a leader fails or leaves the system. When a host discovers that its leader has failed (as indicated by an absence of HEARTBEAT messages for a certain amount of time) or is leaving (as indicated by receiving the LEAVE message from the leader), it would assume itself to be the new leader by sending, after a random delay, a HEARTBEAT message to the ALL_MEMBERS group. If the host receives a HEARTBEAT message, it suppresses its HEARTBEAT message. In this way, the first node which sends the HEARTBEAT message becomes the new leader. In case of contention, the host with the lexically smallest IP address is chosen as the leader. The new leader then becomes the bridge node to all neighbor islands and joins the LEADER and ALL_MEMBERS groups.

D. Bridge Improvement

Initially, a leader node is the bridge node to all neighboring islands. To improve the overall performance, a bridge improvement algorithm is executed at each node to reduce the number

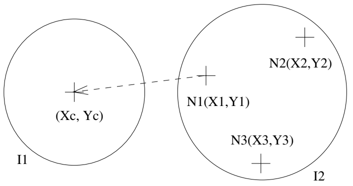

Fig. 2. Closest to neighbor's centroid. In this figure, the three plus signs in the right circle represent the end-hosts and the plus sign in the left circle represent the centroid of island I1. In island I2, the node N_1 is selected as the bridge node to island I1 since it is closest to the centroid of I1.

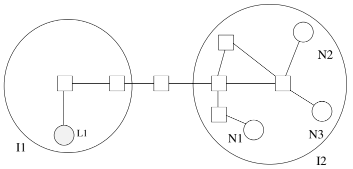

Fig. 3. Closest to neighbor's leader (using network distance). In this figure, the square nodes represent the routers and the circle nodes represent the end-hosts. In island I2, the node N_1 is selected as the bridge node to island I1 since it is closest to the leader L1.

of physical hops in each inter-island unicast connection. This reduces the number of physical links carrying an additional copy of data and the delay in inter-island delivery.

We briefly present in this section the following bridge selection algorithms: Closest to Neighbor's Centroid (CNC), Closest to Neighbor's Leader (CNL) and Closest Pair (CP).

1) Closest to Neighbor's Centroid (CNC): We assume that all nodes have access to a network coordinate service (such as GNP [13]). The leader of an island computes centrally, for each neighbor island, the bridge node in the island which is closest to the neighbor island's centroid among all nodes in the island (Fig. 2). To do this, each JOIN message has to carry the network coordinates of the joining node and the leader has to keep track of all nodes in the island for computing the centroid coordinates. It periodically exchanges the centroid coordinates with neighboring islands' leaders and informs each neighbor island's leader of the bridge node to that neighbor island.

2) Closest to Neighbor's Leader (CNL): In CNL, the node in the island that is closest to the leader of a neighbor island in terms of Enclidean distance (in a network coordinate space) or network distance (in terms of, for example, round-trip time) is selected as the bridge node to that neighbor island (Fig. 3). A non-leader node, upon receiving a HEARTBEAT message, starts this algorithm. To do this, each HEARTBEAT message carries the distance (network distances or Euclidean distances) between each neighbor island's leader and the current bridge node in the island which has an overlay connection to that neighbor island. It also contains the IP addresses of the leaders in the neighboring islands (and their network coordinates if distances are computed in a coordinate space).

When a non-leader node receives a HEARTBEAT message, it determines whether it is closer to at least one neighbor island's leader than the current bridge nodes. If so, it sends a CANDIDATE message to the ALL_MEMBERS group after a random delay; the CANDIDATE message contains the distances to all neighboring islands' leaders. If during the delay it learns from received CANDIDATE messages and the latest HEARTBEAT message that it will not be selected as a bridge node (i.e., for each neighbor island, there is at least one other node that is closer to the neighbor island's leader than itself), it suppresses its CANDIDATE message. For each CANDIDATE message received, the leader updates its table of bridge nodes and informs each neighbor island's leader of its bridge node to that neighbor island.

Eventually, the set of bridge nodes becomes stable. Clearly, if no message is lost and an island has m members, at most $m - 1$ CANDIDATE messages are generated in the island.

3) Closet Pair (CP): This algorithm finds the closest pair of bridge nodes between an island and a neighbor island, in terms of Euclidean or network distance. Each leader keeps track of nodes within its island and exchanges its list of nodes with neighboring islands' leaders. A leader periodically sends a HEARTBEAT message containing the IP addresses or coordinates of the nodes in neighboring islands, and the distance between each pair of bridge nodes.

A node in an island determines the Euclidean distance or network distance to the nodes in neighboring islands. If the distance to a node in a neighbor island is smaller than that of the current pair of bridge nodes connecting the island to that neighbor island, the node sends out a CANDIDATE message to the ALL_MEMBERS group after a random delay. The CANDIDATE message contains the "better" pair of bridge nodes and the corresponding distance of this pair. The leader, upon receiving a CANDIDATE message, updates its table of bridge nodes and informs the corresponding neighbor island's leader the updated pair of bridge nodes.

E. Failure Recovery

There are two types of node failure for which recovery is necessary, namely leader node failure and bridge node failure. When a group member does not receive any HEARTBEAT message for a certain amount of time, it runs the leader election process (as described in Sect. II-C). A failed leader is thus replaced.

For the detection and recovery of bridge node failure, some nodes are chosen as monitor nodes, described as follows. The monitor node for an ingress bridge node (a bridge node which receives inter-island data) is the leader in the same island. For an egress bridge node (a bridge node which sends inter-island data), its monitor nodes are all the downstream bridge nodes it is sending data packets to. In this way, each data packet can function as an implicit HEARTBEAT message of bridge nodes. For example, if a leader receives data packets from the DATA multicast group, it knows that the ingress bridge node in the island is still alive.

However, this is insufficient to completely eliminate the role of explicit HEARTBEAT messages. It is possible that a node is not failed but still is unable to send data packets timely, e.g., there is a failure high up in the island tree and the sender has nothing to send at the moment. Thus, when a bridge node does not receive data for long, it is necessary to send HEARTBEAT messages periodically to its monitor nodes (using the same multicast or unicast channel as it would for sending data packets) to inform them its “aliveness.” The exceptional case happens when an ingress bridge node is also the leader, it will be replaced by the new leader.

How the failure of a bridge node is recovered is described as follows. If a leader detects that the ingress bridge node in the island fails, it becomes the new ingress bridge node and insert all entries on the failed bridge node’s bridge connectivity list into its list. If an egress bridge node detects that the upstream bridge node in the parent island fails, it reports the error to the corresponding parent island’s leader by sending an ERROR message using the ERROR unicast channel. The parent island’s leader then uses itself as the bridge node to the islands that have sent ERROR message. If however the parent island’s leader is the bridge node to the island, there is no need to send ERROR messages because either the ALM protocol fixes the failure (i.e., the parent island contains only the failed node) or the parent island elects another leader.

Upon recovery from a leader failure, the new leader acts as the bridge node to the parent and all child islands. Then a bridge improvement algorithm is executed. Upon recovery from a bridge node failure, all nodes in the island in which a bridge node has failed execute a bridge improvement algorithm. Thus, in both cases, the island overlay would be fully repaired.

III. Simulation Experiments

A. Simulation Environment

For the purpose of simulation, a number (10) of Transit Stub graphs are generated with the Georgia Tech’s Internet topology generator [14]. Each of the generated graphs (Transit Stub) is a two-layer hierarchy of transit and stub networks. There are six transit domains, each with 15 routers. There are 90 stub domains, each with 20 routers. In each simulation, each host is randomly attached to a stub-domain router directly. The delays of these links are uniformly distributed over the interval $[0.1, 3)$.

Unless otherwise specified, the group size is 200 hosts excluding the sender and the CNL algorithm (using half round-trip time as the distance metric) is used. The logical tree connecting islands is the minimum spanning tree of the complete graph connecting the leaders.

We consider two ways to choose some routers to be multicast-capable. The first one chooses a certain percentage of stub-domains as multicast-capable domains, where all the routers inside these domains are multicast-capable. The second one randomly chooses some percentage of routers to be multicast-capable. Unless otherwise specified, the first method is used and the percentage of domains selected is 40%. Note that the second method does not distinguish between stub-domain and transit-domain routers, so at 100% both average link stress and average Relative Delay Penalty (RDP) are 1.

B. Illustrative Results

We first compare the performance of our scheme with SM, modified SM (labelled ”SM (MST)” in some figures) and ALMI [5]. Modified SM is almost the same as SM, except that data is distributed to subnet representatives through an overlay tree consisting of the sender and the representatives instead of using separate unicast streams. The overlay tree is constructed in the way that minimizes the sum of unicast delays. The schemes are compared at different group size, ranging from 25 to 1600. The average stress and RDP results are presented in Fig. 4 and Fig. 5, respectively.

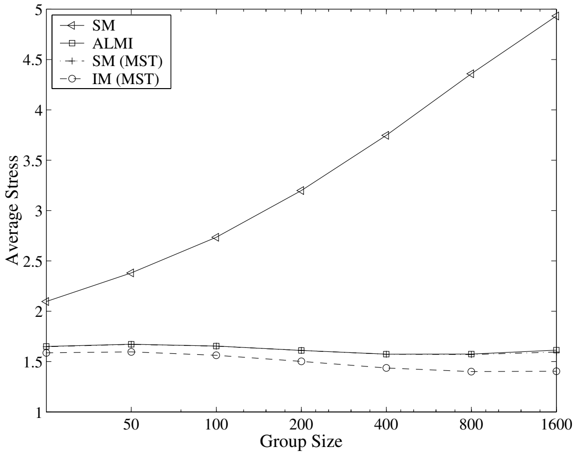

Fig. 4. Stress (Average) vs. Group Size.

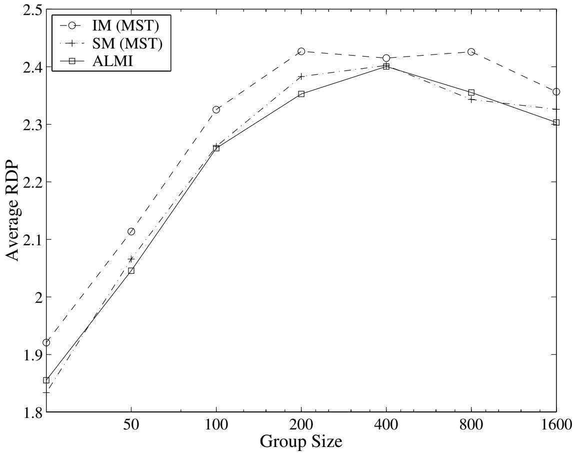

Fig. 5. RDP (Average) vs. Group Size.

Fig. 4 shows the average stress against group size. SM has the largest stress. The average stress increases rapidly with increasing group size because it uses the star topology. It can be seen that modified SM shows no noticeable improvement in terms of stress. But with our scheme, stress is $7-14\%$ smaller than using ALMI alone, because it considers the selection of bridge nodes and makes use of IP multicast. While an improvement of about 7 – 14% is not large of a figure, the average stress for ALMI is low at around 1.6 and therefore there is not much room for improvement.

The RDP results are shown in Fig. 5. Note that the RDP curve for SM is not shown as the RDP is very close to one for all group sizes. SM clearly outperforms the other two schemes in terms of end-to-end delay, but at the cost of suffering much

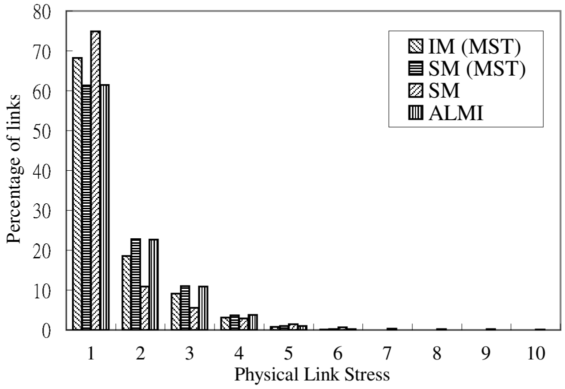

Fig. 6. Link Stress Distribution.

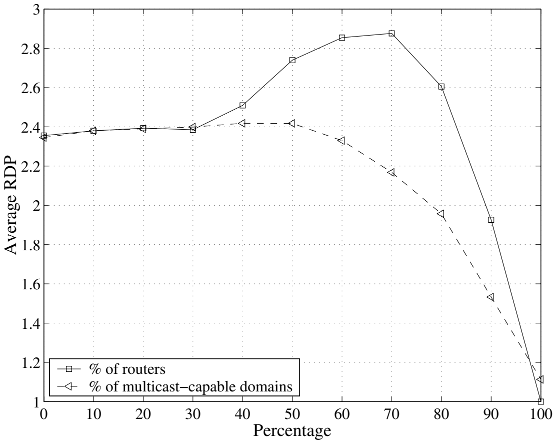

Fig. 7. RDP (Average) vs. Percentage of IP multicast routers or domains.

higher stress. As can be seen from Fig. 4 and Fig.5, our scheme reduces stress at the cost of a small increase in RDP (which is less than about 5%).

The link stress distributions are shown in Fig. 6. Only the percentages for link stress of 10 or less are shown. The maximum link stress of SM is 200, which cannot be seen from the figure. This is because SM aggregates unicast streams for each subnet and links near the sender will suffer from large stress if it need to send data to a lot of subnets. Our scheme, modified SM and ALMI show much less traffic concentration, with the maximum link stress of nine. But, in our scheme, more links are of lower stress than modified SM and ALMI.

Finally, we evaluate the effectiveness of our scheme at different percentage of multicast-capable domains or IP multicast capable routers. In Fig. 7, we show the average RDP versus the percentage of multicast-capable domains or multicast-capable routers in the network. In the first selection method, as expected, the RDP decreases as the percentage of multicast-capable domains increases. However, for the second one, the RDP first increases and then decrease. As the number of multicast routers increases, islands begin to form. Because the selection is random, it is likely that some islands are "belt-shaped" (a chain of multicast routers). Since each island has only one ingress bridge node, it may not be possible for it to be close to all the egress bridge nodes on the same island, especially "belt-shaped" islands. This increases delays, and hence the RDP. On the other hand, as the number of multicast routers increases, more and more large islands are formed. Since intra-islands packet delivery is based on IP multicast which employs shortest-path routing, the RDP decreases. There is hence a peak in RDP as shown.

IV. Conclusions

The Internet today consists of multicast-capable "islands" and multicast-incapable regions interconnected by multicast-incapable routers. In order to enable global multicast efficiently, multicast features should be used within an island while the islands are interconnected by unicast connections. In this paper, we propose a scheme called Island Multicast (IM), which organizes multicast delivery into two levels: at the upper level inter-island overlay is established, while at the lower intra-island level IP multicast is used. We presented the basic mechanisms for this scheme.

Based on IM, we demonstrated that it can make significant improvement in average stress, at the cost of a small increase in end-to-end delay. And if IP multicast routers are randomly placed, quite contrary to what was expected, the average RDP may increase with the number of multicast routers.

References

[1] J. Liebeherr, M. Nahas, and W. Si, "Application-layer multicasting with delaunay triangulation overlays," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 1472–1488, Oct 2002.

[2] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in *Proceedings of ACM SIGCOMM 2002*, Aug 2002.

[3] M. Castro, P. Druschel, A.-M. Kermarec, and A. I. T. Rowstron, "Scribe: A large-scale and decentralized application-level multicast infrastructure," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 1489–1499, Oct 2002.

[4] Y.-H. Chu, S. G. Rao, S. Seshan, and H. Zhang, "A case for end-system multicast," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 1456–1471, Oct 2002.

[5] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel, "ALMI: An application level multicast infrastructure," in *Proceedings of the 3rd USENIX Symposium on Internet Technologies and Systems*, Mar 2001.

[6] Y. Chawathe, S. McCanne, and E. Brewer, "An architecture for internet broadast distribution as an infrastructure service," *PhD thesis, University of California, Berkeley*, Dec 2000.

[7] P. Francis, "Yoid: Your own internet distribution," *http://www.icir.org/yoid/*, Feb 2004.

[8] R. Finlayson, "The UDP multicast tunneling protocol," *draft-finlayson-umtp-09.txt*, Nov 2003.

[9] P. Parnes, K. Synnes, and D. Schefstrom, "Lightweight application level multicast tunneling using mtunnel," *Computer Communications*, pp. 1295–1301, Apr 1998.

[10] D. Thaler, M. Talwar, L. Vicisano, and D. Ooms, "IPv4 automatic multicast without explicit tunnels (AMT)," *draft-ietf-mboned-auto-multicast-02.txt*, Feb 2004.

[11] B. Zhang, S. Jamin, and L. Zhang, "Universal IP multicast delivery," in *Proceedings of the International Workshop on Networked Group Communication (NGC)*, Oct 2002.

[12] J. Park, S. J. Koh, S. G. kang, and D. Y. Kim, "Multicast delivery based on unicast and subnet multicast," *IEEE Communications Letters*, vol. 5, pp. 1489–1499, Apr 2001.

[13] T. S. E. Ng and H. Zhang, "Predicting internet network distance with coordinates-based approaches," in *Proceedings of INFOCOM 2002*, June 2002.

[14] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *Proceedings of INFOCOM 1996*, pp. 594–602, Mar 1996.