Bridge-Node Selection and Loss Recovery in Island Multicast

W.-P. Ken Yiu K.-F. Simon Wong S.-H. Gary Chan
Department of Computer Science
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
Email: {kenyiu, cssmw, gchan}@cs.ust.hk

***Abstract*— Island Multicast (IM) has been recently proposed to achieve efficient global multicast, where IP multicast is used within multicast-capable domains (the so-called *islands*) while overlay connections are used to bridge islands. In the previously proposed scheme, the number of ping measurements to find good bridge-nodes is at least proportional to island size, and a leader needs to keep track of all its members in the island. In this paper, we improve the system scalability by presenting a bridge-node selection algorithm where both the numbers of ping measurements and members to keep track of are greatly reduced to some constants. We further propose a recovery scheme for packets lost across islands. Our scheme uses a number of recovery meshes formed by overlays of some randomly chosen nodes. Simulation results show that our bridge-node selection is efficient in terms of control overhead and achieves scalability with little cost in network stress and delay. As compared to traditional source and parent recoveries, our loss recovery scheme substantially reduces both the recovery delay and bandwidth overhead to achieve reliability.**

I. Introduction

With the availability and penetration of multicast-capable routers, the local networks in the Internet nowadays are generally multicast-capable. These multicast domains or so-called "islands" are interconnected by routers which are either multicast-incapable or multicast-disabled (for security and traffic control purposes). In order to achieve efficient global multicast, a hybrid approach called Island Multicast (IM) has been recently proposed [1].

IM organizes the members in a multicast session (or group) into interconnected islands. It is a bi-level architecture: the upper level manages data delivery across islands while the lower level concerns data delivery among members within an island. In the upper level, IM constructs a logical tree to connect the islands using overlay connections (the so-called *bridges*). To construct the tree, each island elects a "leader" to run an overlay multicast protocol, which can be any of the existing ALM protocols. Given the inter-island overlay tree, a pair of bridge-nodes is then selected to connect the islands at the end-points of the bridge together. These node-pairs take the responsibility of inter-island (unicast) delivery. For packet forwarding, the sender delivers packets in its own island using IP multicast. When a bridge-node receives the packets, it forwards the packets to the neighboring islands via unicast. The simplicity and flexibility of the scheme makes it easy to implement and deploy over the Internet [2].

In the previously proposed IM, island leaders are responsible to form bridges connecting islands together. There are three heuristics to select good nodes at the end-points (bridge-nodes), namely, *Closest to Neighbor's Centroid (CNC)*, *Closest to Neighbor's Leader (CNL)* and *Closest Pair (CP)*. In all three heuristics, coordinates of each node are used for computing the network distances between nodes [3], [4]. However, measuring network coordinates requires a set of *distributed landmarks* in the Internet, which leads to implementation complexity. Furthermore, the computed network coordinates cannot reflect the dynamic network distances between nodes. Therefore, periodic ping measurement is more preferable to find network distances between nodes. Both CNL and CP may apply ping measurements in their bridge-node selection. However, the island leaders of these heuristics need to store the full member lists on their own islands and keep track of member status. Furthermore, CNL and CP requires $O(N)$ and $O(N^2)$ ping measurements across islands, respectively, where N is the number of members on each island. This generates much control overhead in terms of bandwidth usage.

In this paper, we propose a simpler scheme for bridge-node selection. Our scheme does not require leaders to keep track of member status. In addition, the scheme needs only a constant number of ping measurements across islands to select bridge-nodes. Our scheme selects a random list of members within an island. These members measure the distances between themselves and a list of members in another island. By simulation, we find that the bridge quality is comparable to the previous approaches based on exhaustive ping measurements.

Besides bridge-node selection, we also investigate providing reliability in IM. Within each island, we assume that Scalable Reliable Multicast (SRM) is used [5]. Observing that packets may be lost in the overlays (which means that all nodes in an island would experience the same packet loss), we propose and study a novel approach to recover errors across islands. On each island, R nodes are randomly selected as recovery nodes. These recovery nodes are randomly assigned to T recovery

This work has been supported, in part, by the Areas of Excellence (AoE) Scheme on Information Technology of the University Grant Council (AoE/E-01/99), and by grants of the Research Grant Council (HKUST6199/02E & HKUST6156/03E) in Hong Kong.

groups, each of which forms an overlay mesh. When SRM is unable to recover errors locally within an island, mesh neighbors are requested for retransmissions. Due to the randomized nature of group mapping, error correlation between mesh neighbors is greatly reduced. This approach hence provides better reliability than the one requesting retransmission solely from upstream islands.

We briefly review related work as follows. Most of the ALM protocols (such as Narada, NICE, DT, ALMI, etc.) are proposed to form efficient overlay multicast tree without considering the presence of multicast-capable domains in the Internet [6]–[9]. SRM has been proposed to achieve efficient reliability for IP multicast [5]. However, SRM can only be used in a multicast-capable domain. Our scheme extends the scope to a global environment with a mix of multicast-capable and multicast-incapable domains. Our recovery scheme shares similar idea of lateral error recovery (LER) to avoid vertical recovery [10]. The difference is the usage of multiple overlay meshes. While multiple trees (which increases the network stress) are used for data delivery in LER, the overlay meshes in our scheme are only used for retransmitting lost packets, and hence do not affect the network performance for data transmission. In addition, our scheme is much simpler as compared with LER.

This paper is organized as follows. First, we describe in detail our bridge-node selection in Section II and recovery scheme in Section III. In Section IV, we then present some illustrative simulation results. Finally, we conclude our work and results in Section V.

II. Efficient Bridge-Node Selection

In order to improve IM performance, bridge-nodes should be selected so that the distance in each inter-island overlay is short (because this reduces both end-to-end delay and link stress). In this section, we present an efficient bridge-node selection scheme which requires only a constant number of ping measurements. Instead of measuring the round-trip time (RTT) between all pairs of nodes between neighboring islands, we randomly select a constant number (C) of peers in each island to perform ping measurements.

This simple approach reduces both the complexity and measurement cost of the system. The key challenge is how to dynamically select random members without having to keep track of all the member status. The protocol is summarized as follows:

1) Leader distributes HEARTBEAT message periodically within its own island using IP multicast. The message also includes a parameter ρ ($0 < \rho \leq 1$) which is used as probability for member reply.
2) Upon receiving a HEARTBEAT message, a non-bridge-node randomly replies (via unicast) with probability ρ, and a bridge-node always replies the message. The absence of bridge-node replies for a certain period of time would trigger bridge-node selection. A leader can dynamically adjust ρ to ensure the total number of replies is roughly equal to C.
3) The leader only keeps the latest set of nodes which reply the HEARTBEAT message. In this way, a leader is not required to detect the failure of a node.
4) During bridge selection, a list of C members in an island is sent to the neighboring islands.
5) The members of an island are randomly paired with those in the neighboring island to form a total of C pairs. These node-pairs perform ping measurements with each other.
6) A leader simply selects the pair with the minimum measured RTT as the bridge-nodes. Given a data flow, the node through which packets enter an island is called *ingress* node, while the one through which packets leave an island is called *egress* node.

Clearly, our approach is suboptimal as compared with CNL or CP approaches. However, we show by simulation (in Section IV) that our approach achieves similar level of performance. Clearly, our scheme reduces the control overhead from $O(N)$ (CNL approach) or $O(N^2)$ (CP approach) to $O(C)$, where C is a constant (the default is 10 in our simulation).

III. Packet Loss Recovery in IM

In this section, we present the mechanism to provide reliability in IM. SRM is used to recover packets lost within an island. For packets lost during their transit across islands, inter-island recovery has to be done. A natural way is to request the upstream hosts for retransmission (i.e., parent recovery). However, such mechanism suffers from the problem of error correlation and implosion [10].

We hence propose and study a novel recovery scheme. Every island leader randomly selects a number R of recovery nodes. The selection may be done in the same way as the one described in our bridge-node selection in Section II. Each recovery node is then randomly assigned to one of T groups. Each group forms an independent recovery mesh. The meshes may be constructed with any existing overlay mesh mechanisms.

Clearly, a large R means higher density of recovery nodes, and hence lower recovery delay. However, since loss is likely correlated with close nodes, this also increases the chance of failed retransmission. Therefore, we expect an optimal R to minimize recovery time. The value of T should also be correlated with R to reduce the chance of requesting nodes in the same island for retransmission. Generally, T should be set slightly higher than or equal to the value of R. (For simplicity, we assume some predetermined values of R and T in this paper. Dynamically adjusting the values of R and T to adapt the network environment will be explored in the future.)

We assume the data source marks each packet with increasing sequence number. The ingress nodes detect errors by gaps in the sequence number. Whenever an error occurs, the ingress node informs its R recovery nodes in the same island via unicast. Instead of requesting upstream nodes for retransmission, these recovery nodes perform retransmission

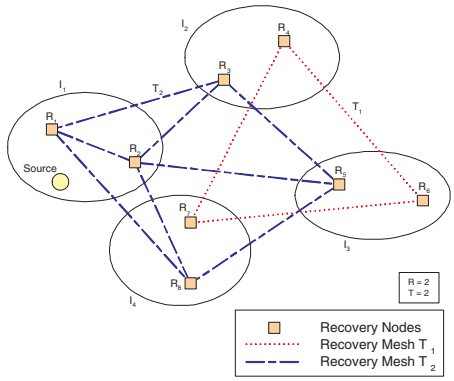


Fig. 1. Illustration of constructing recovery meshes.

with their connected neighbors in the constructed recovery meshes.

Since a recovery node may connect to a number of neighbors in the recovery mesh, retransmission requests are sequenced according to increasing RTT with the connected neighbors. The RTT is measured during the configuration of the recovery meshes. To provide 100% reliability, we request the source for retransmission after a certain attempts with the neighbors in the recovery mesh. In our experiment, we find that only a few (say, 2 to 3) attempts in the mesh can recover most errors. Since most errors can be recovered in the first few trials of retransmission, this greatly reduces the implosion problem at the source.

We illustrate the idea in Figure 1. The squares $R1$ to $R8$ indicate the recovery nodes for the four islands $I1$ to $I4$. In this example, every island selects two recovery nodes (i.e., $R = 2$). Here, two recovery meshes $T1$ and $T2$ are used (i.e., $T = 2$). Suppose $I3$ detects an error, $R5$ and $R6$ are then asked to perform retransmissions. Suppose two attempts are used. $R6$ first seeks help from $R4$ and $R7$. If both $R4$ and $R7$ fail to retransmit the packet to $R6$, $R6$ then performs retransmission with the source. The recovery node successfully repairs the error, multicasts the packet across the island using SRM (the other recovery nodes may abort their recovery process if necessary).

IV. Illustrative Simulation Results

In this section, we present the simulation results of our scheme using Internet-like topologies. We first discuss the experimental setup (Section IV-A), and next evaluate our bridge-node selection scheme (Section IV-B). Lastly, we study the performance of our recovery scheme and compare it with source recovery (Section IV-C).

A. Experimental Setup

We generate a number of *Transit Stub* topologies with the Georgia Tech's Internet topology generator (GT-ITM) [11]. The generated topologies are two-layer hierarchical networks with transit networks (of four transit domains, each with 16 randomly distributed routers on a 1024×1024 grid) and stub networks (of 64 domains, each with 15 randomly distributed routers on a 32×32 grid). A host is connected to a stub router via a LAN (on a 4×4 grid). The leaders run ALMI as the inter-island tree construction protocol [9]. We model 95% of links in the network with the loss rate uniformly distributed between 0 and 1%, and the remaining 5% of links with the loss rate uniformly distributed between 5% and 10% (as according to a study based on real measurement [12]). In our experiments, packets are dropped in a link according to the loss rate of the link.

In our experiments, we consider that errors occur within an island can be recovered by SRM within a time estimated as twice the island diameter, which is the worst case for a lost packet to be retransmitted within an island. For recovery mesh, we use a random mesh protocol similar to Narada without overlay improvement [6]. There are two kinds of packets, application data (of size 1024 bytes) and control data (of size 64 bytes).

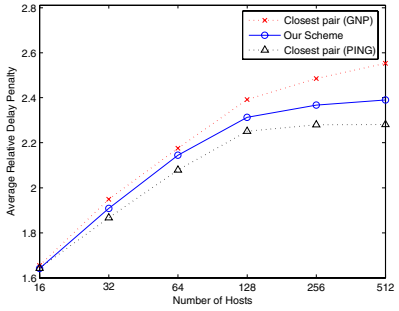
In the simulation, we are interested in the following performance metrics:

- Physical link stress (PLS), defined as the number of duplicated packets transmitted through a given physical link.
- Relative delay penalty (RDP), defined as the ratio between overlay delay to underlay delay of a host from the source.
- Recovery delay (in seconds), defined as the delay from the time an error is detected at a host to the time the packet is completely recovered (including the delay by SRM).
- Retransmission overhead, defined as the total traffic given by the sum of the control packets for requests and retransmitted data packets, per lost packet. We normalize the overhead by the size of a data packet.

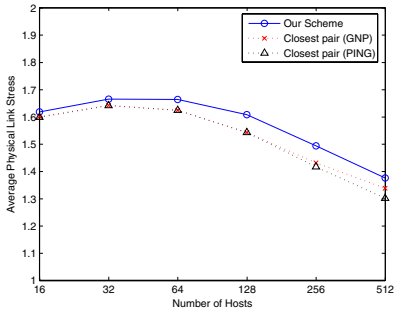
In our simulation, unless otherwise stated, we use the following baseline parameters: number of hosts is 128, $C = 10$ for bridge-node selection, $R = 2$ and $T = 4$ for reliable service.

B. Performance of Bridge-node Selection

We first examine the performance of our bridge-node selection algorithm. The system can obtain the network distances between hosts either by GNP or ping measurements. We compare our scheme in Figure 2 the average RDP and PLS versus the number of hosts with Closest Pair (CP) based on exhaustive search. From Figure 2(a), we see that average RDP of our approach is slightly lower than that of CP using GNP approach and slightly higher than that of CP using ping approach. We compare their PLS performance in Figure 2(b). In general, the average PLS increases at the beginning. This is because when there are too few hosts, IM cannot take advantage of IP multicast within islands, and hence the multicast tree is purely an overlay tree. When there are more and more hosts, IM can use IP multicast to deliver packets. As a result, the average PLS gradually decreases. From the figure, we see that our scheme only suffers slightly in terms of PLS as compared with the CP approaches.



(a) Average RDP versus number of hosts.



(b) Average PLS versus number of hosts.

Fig. 2. Average RDP and PLS of our approach as compared with closest pair approach.

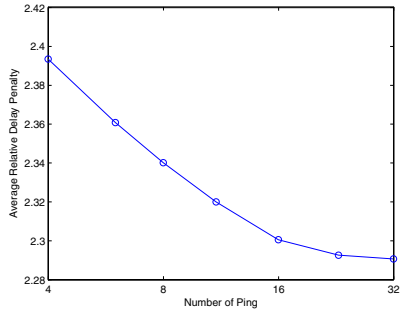


Fig. 3. Average RDP versus the number of ping measurements for our approach.

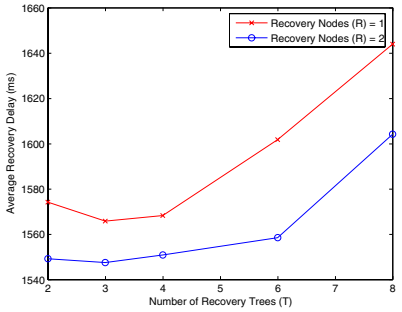


Fig. 4. Effect of R and T.

We next show in Figure 3 the average RDP versus C (number of ping measurements between two islands). Clearly, the average RDP decreases when C increases, because the system has a higher probability to find a close pair of nodes when C is large. However, a larger C implies more overhead in terms of ping measurements. Therefore, we should not have C too large. In our experiments, we find that $C \approx 10$ is a reasonably good parameter.

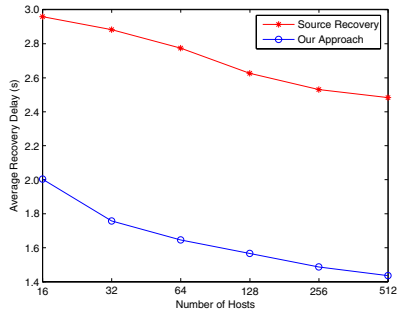
C. Performance of Reliable Multicast

Figure 4 shows the average recovery delay versus T given R. As T increases, the recovery delay first decreases and then increases. This is because as T increases, the recovery nodes within the same island are distributed into different meshes, and the error correlation between neighboring nodes decreases. However, as T further increases, the recovery nodes are spaced out, making the recovery process ineffective. Given R, there is hence an optimal T to minimize the delay.

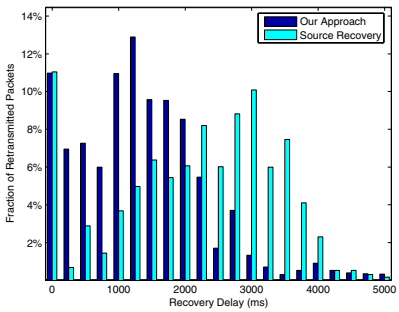
We then compare the recovery delay of our scheme with source recovery where every host requests retransmission from the data source. We show in Figure 5(a) the average recovery delay versus the number of hosts. The average delay for both schemes decreases as the number of hosts increases. This is because there are more hosts located in the same island, allowing most lost packets to be recovered within an island by SRM, whose recovery delay is very small (less than 100 ms). Clearly, our scheme achieves a substantially better performance as compared with source recovery, because the average distance to the recovery neighbors in our scheme is much shorter than that to the source in source recovery. In addition, Figure 5(b) compares the distributions of recovery delay in both schemes. Clearly, the distribution of our scheme is skewed towards left, leading to its low recovery delay.

We compare in Figure 6 the retransmission overhead versus the number of hosts. The retransmission overhead does not depend much on the number of hosts in the network. Since recovery nodes in our scheme request neighboring islands for retransmission, there is no implosion problem as in source recovery. (The implosion may further increase packet loss at the source, which triggers more retransmission requests in source recovery.)

Note that neighboring recovery nodes may not have requested packets. Therefore, after a few retransmission requests (two attempts in our simulation), the requesting node would ask the source for retransmission. Figure 7 shows the fraction of retransmissions from the source in our scheme. Note that almost all retransmission requests (over 90%) are handled by recovery neighbors instead of the source. This greatly relieves the processing and network load at the source for handling retransmission requests. When the number of hosts is small, the number of nodes in a mesh is small. As a result, some nodes may not even have two neighbors for retransmission before asking the source. Therefore, there is a relatively high fraction of retransmissions directed to the source. As the number of hosts increases, the recovery neighbors are closer and the probability of packet loss in retransmissions becomes



(a) Average recovery delay versus the number of hosts as compared with source recovery.



(b) Distribution of recovery delay for different schemes.

Fig. 5. Comparison with source recovery in terms of recovery delay.

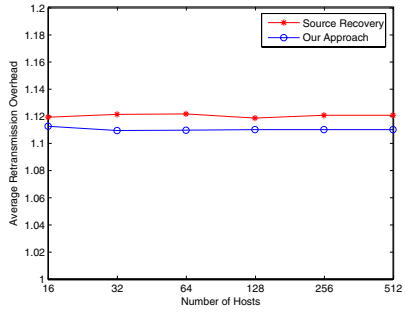


Fig. 6. Average retransmission overhead per lost packet versus number of hosts.

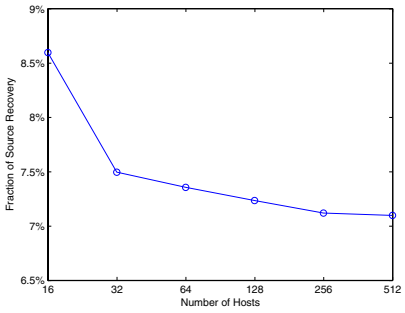


Fig. 7. Fraction of source recovery used in our retransmission scheme.

lower, leading to higher chance of successful recovery in the first few attempts.

V. Conclusions

Island Multicast (IM) takes advantage of IP multicast-capability in local networks to improve the efficiency of data delivery in terms of network stress and delay. Due to its simplicity and ease of deployment, IM is a promising solution for global multicast. However, the previously proposed IM suffers from management and maintenance problem because leaders need to keep track of all member status in their own islands. Furthermore, the bridge-node selection mechanism incurs much bandwidth overhead. We describe in this paper, a new bridge-node selection algorithm to address the above problems. Our scheme reduces much control overhead in terms of bandwidth and storage. The requirement for ping measurements is reduced from $O(N^2)$ or $O(N)$ to $O(C)$, where N is the island size and C is a constant ($C \approx 10$). Our approach achieves comparable network performance in terms of stress and delay as the previous scheme.

We also address the loss recovery issue in IM, where SRM is used for intra-island recovery. We present a new scheme to recover errors across islands, and hence extend the scope of error recovery to global multicast environment. Via simulation based on Internet-like topologies, we show that our scheme ourperforms source recovery in terms of recovery delay.

References

[1] K.-W. R. Cheuk, S.-H. G. Chan, and J. Y.-B. Lee, "Island Multicast: The Combination of IP Multicast with Application-Level Multicast," in *Proceedings of IEEE ICC'04*, June 2004, pp. 1441–1445.

[2] K.-L. Cheng, K.-W. Cheuk, and S.-H. G. Chan, "Implementation and Performance Measurement of an Island Multicast Protocol," in *Proceedings of IEEE ICC'05*, May 2005.

[3] T. S. E. Ng and H. Zhang, "Predicting Internet Network Distance with Coordinates-Based Approaches," in *Proceedings of IEEE Infocom'02*, June 2002.

[4] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A Decentralized Network Coordinate System," in *Proceedings of IEEE Sigcomm'04*, Aug. 2004.

[5] S. Floyd, V. Jacobson, C.-G. Liu, S. McCanne, and L. Zhang, "A Reliable Multicast Framework for Light-Weight Sessions and Application Level Framing," *IEEE/ACM Transactions on Networking*, no. 6, pp. 784–803, Dec. 1997.

[6] Y.-H. Chu, S. G. Rao, S. Seshan, and H. Zhang, "A Case for End-System Multicast," *IEEE Journal on Selected Areas in Communications*, pp. 1489–1499, Oct. 2002.

[7] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable Application Layer Multicast," in *Proceedings of ACM Sigcomm'02*, Aug. 2002.

[8] J. Liebeherr, M. Nahas, and W. Si, "Application-Layer Multicasting with Delaunay Triangulation Overlays," *IEEE Journal on Selected Areas in Communications*, pp. 1472–1488, Oct. 2002.

[9] 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.

[10] K.-F. Wong, S.-H. Chan, W.-C. Wong, Q. Zhang, W.-W. Zhu, and Y.-Q. Zhang, "Lateral Error Recovery for Application-Level Multicast," in *Proceedings of IEEE Infocom'04*, Mar. 2004, pp. 2708–18.

[11] K. Calvert, J. Eagan, S. Merugu, A. Namjoshi, J. Stasko, and E. Zegura, "Extending and Enhancing GT-ITM," in *Proceedings of the ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research (MoMeTools)*, Aug. 2003, pp. 23–27.

[12] V. N. Padmanabhan and L. Qiu, "Network Tomography Using Passive End-to-end Measurements," in *Proceedings of DIMACS Workshop on Internet and WWW Measurement, Mapping and Modeling*, Feb. 2002.