SIM: SCALABLE ISLAND MULTICAST FOR PEER-TO-PEER MEDIA STREAMING

Xing Jin *Kan-Leung Cheng* *S.-H. Gary Chan*

Department of Computer Science
The Hong Kong University of Science and Technology
{csvenus, klcheng, gchan}@cs.ust.hk

ABSTRACT

Despite the fact that global multicast is still not possible in today's Internet, many local networks are already multicast-capable (the so-called multicast "islands"). However, most application-layer multicast (ALM) protocols for streaming has not taken advantage of the underlying IP multicast capability. As IP multicast is more efficient, it would be beneficial if ALM can take advantage of such capability in building overlay trees. In this paper, we propose a fully distributed protocol called Scalable Island Multicast (SIM), which effectively integrates IP multicast and ALM. Hosts in SIM first form an overlay tree using a scalable protocol. They then detect IP multicast islands and employ IP multicast whenever possible. Through simulations on Internet-like topologies, we show that SIM achieves much lower end-to-end delay and link stress as compared with traditional ALM protocols.

1. INTRODUCTION

With the popularity of broadband Internet access, there has been increasing interest in media streaming services. Recently, peer-to-peer (P2P) streaming has been proposed and developed to overcome limitations in traditional server-based streaming [1]. In a P2P system, cooperative peers self-organize themselves into an overlay network via unicast tunnels. They cache and relay data for others, therefore eliminating the need for powerful servers from the system. Currently, there are two approaches of overlays for P2P streaming: tree structure and gossip mesh. The first one builds one or multiple overlay tree(s) to distribute data among hosts. Examples include application-layer multicast schemes (e.g., Narada and NICE) and some P2P video-on-demand systems (e.g., P2Cast and P2VoD) [2]. The second one builds a mesh among hosts using gossip algorithms, with hosts exchanging data with their neighbors in the mesh [1]. Despite of their better resilience to network and group dynamics, gossip-based approaches have overall higher control overhead due to data scheduling and mesh maintenance. They also have higher playback delay because data clips are transmitted over multiple paths to a host and the longest one is the video delay. On the contrary, trees introduce lower end-to-end delay and are easier to maintain. We hence adopt a tree-based approach in this paper.

Most previously proposed tree-based ALM protocols (such as Narada, NICE, DT, Scribe, ALMI, etc.) assume that none of the routers are multicast-capable and hence have not considered the use of the underlying IP multicast capability. Although global IP multicast is not available today, many local networks in today's Internet are already multicast-capable. These local multicast-capable domains, or so-called "islands," are often interconnected by multicast-incapable or multicast-disabled routers. Since IP multicast is more efficient than ALM, it would be beneficial if ALM makes use of the local multicast capabilities in building trees. We hence propose a distributed and scalable scheme called *Scalable Island Multicast (SIM)* that combines IP multicast with ALM for media streaming.

In SIM, hosts within an island communicate with IP multicast. They connect across islands with unicast overlay paths. Each host first distributedly joins an overlay tree, which is mainly for monitoring and maintenance purpose. A host then detects and joins its multicast island. Each island in SIM has a unique ingress host. The ingress receives packets from outside of the island through its overlay connection and IP-multicasts them within the island. The other members within the island receives data from IP multicast instead of from their parents in the overlay tree.

We have evaluated SIM with simulations on Internet-like topologies. As compared with other traditional ALM protocols, SIM efficiently combines IP multicast with ALM to achieve low end-to-end delay and link stress.

We briefly review previous work on island multicast as follows. Though protocols such as Scattercast, YOID, UMTP, mTunnel, AMT, Universal Multicast (UM) and Subset Multicast (SM) have been proposed to combine IP multicast with ALM, many of them require special nodes (such as proxies or routers) or manual configuration for inter-host connections [3]. SIM is fully autonomous, does not require any special or super nodes and is scalable to large groups. As op-

This work is supported, in part, by the Area of Excellence in Information Technology of the University Grant Council (AoE/E-01/99), Competitive Earmarked Research Grant of the Research Grant Council (HKUST6156/03E) in Hong Kong and the Innovation and Technology Commission of the Hong Kong Special Administrative Region, China (GHP/045/05).

posed to [4], SIM is fully distributed and scalable. The work in [5] proposes a distributed approach to integrate IP multicast and ALM. Each island has a leader, which identifies some ingress and egress hosts in its island for data delivery. This approach puts heavy control loads on leaders and has complex mechanism for the management of leaders, ingress hosts, and egress hosts. SIM provides a much simpler data delivery method and hence is much more implementable. In SIM, there is no leader, and there is no overhead to select egress hosts.

The rest of the paper is organized as follows. In Section 2 we describe the key components in SIM. In Section 3 we present some illustrative simulation results. We conclude in Section 4.

2. SYSTEM DESIGN

2.1. Construction of Overlay Trees

We are interested in building a tree with low end-to-end delay. Clearly, the tree construction mechanism should be distributed so that the system is scalable to large number of users. Furthermore, the algorithm should be simple with low setup and maintenance overhead.

In SIM, a new host first contacts a Rendezvous Point (RP) to obtain a list of current hosts in the system. It pings these hosts and selects k closest ones. Then, it pings the neighbors of these selected hosts, and selects k closest ones from all the hosts (the neighbors and the original k hosts). This process is repeated until the improvement on round-trip time is lower than a certain threshold, or the number of iterations exceeds a certain value. At the end of the process, the new host selects from its current k closest hosts the one with enough forwarding bandwidth as its parent. If there are no qualified hosts, the new host goes back up one level to look for qualified parents.

Figure 1 shows an example of host joining in our scheme. Suppose $k = 2$ and P is a new host. P first obtains a list of hosts from the RP, say, C, D, E, F and G. P then pings all of these hosts and selects two closest ones, say C and D. P then pings all of C's and D's neighbors. It continues selecting two closest hosts from C, D, C's neighbors (i.e., A, F and G) and D's neighbors (i.e., B, H and I). Such iteration stops if any of the above stopping conditions is satisfied. Since the list of hosts returned by the RP is randomly generated, the communication overhead for joining is distributed to all the hosts. In the following discussion, a parent of a host refers to the host's parent in the overlay tree.

2.2. Integrating IP Multicast

After a host joins the overlay tree, it detects its island and joins the island if any. First of all, each host should record its distance from the source on the overlay tree, in terms of round-trip time or hops. The distance from the source can be computed as the sum of its parent's distance from the source and the distance from its parent.

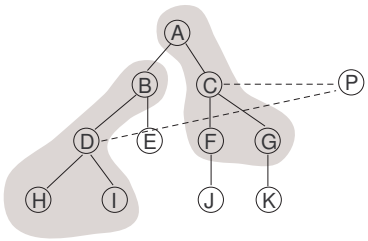


Fig. 1. An example of joining the overlay tree in SIM.

Formation of Multicast Groups: Each streaming session has two unique class-D IP addresses for IP multicast. One is used for multicasting control messages, and the other is used for multicasting streaming data. We call the groups corresponding to these two IP addresses a *CONTROL group* and a *DATA group*, respectively.

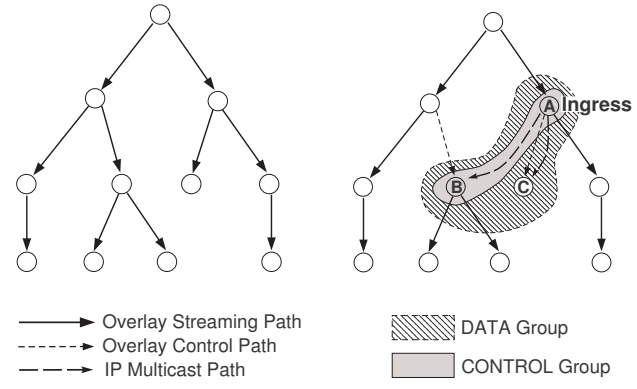
Each island has a unique ingress host, which is responsible for accepting data outside the island and multicasting it within the island. We call a host within the island a *border host* if its *parent* is not inside the island. In SIM, border hosts (including the ingress) join both the CONTROL group and the DATA group, and non-border hosts only join the DATA group.

An ingress host periodically multicasts *KeepAlive* messages in the CONTROL group, which contains information about its distance from the source. It also multicasts streaming data within the DATA group. The ingress is selected from border hosts in the CONTROL group. Initially, the ingress of an island is its first joining host. A new border host substitutes the current one to become an ingress if: (1) The current ingress leaves or fails (detected through the missing of *KeepAlive* messages), or; (2) A border non-ingress host has lower end-to-end distance from the source than the current ingress by a certain threshold.

Island Detection: The two class-D IP addresses are maintained by the RP. When a new host joins the session, it obtains the addresses and a list of current hosts from the RP. The new host then joins the overlay tree as described above. Afterwards, it joins the CONTROL group.

- If an island exists, the host receives the ingress's *KeepAlive* messages. The host then detects whether it itself is a border host. If it is, it remains in the CONTROL group and further joins the DATA group; Otherwise, it exits the CONTROL group and joins the DATA group.

  An non-ingress host in the DATA group stops receiving streaming data from its overlay parent. Instead, it accepts data transmitted by IP multicast. The connection to its parent is only used for transmitting control messages. If this host becomes an ingress later, it will



(a) An overlay tree for data delivery; (b) Integrating IP multicast.

Fig. 2. Combining IP multicast and ALM.

resume the overlay connection and accept data from its parent again.

- If the host does not find any island to join, it forms an island (i.e., a CONTROL group and a DATA group) only consisting of itself and becomes the island ingress.

We show an example of data delivery with IP multicast in Fig. 2. Figure 2(a) shows the overlay tree formed as described above. In Figure 2(b), hosts A, B and C join the CONTROL group and detect that they are in the same island. A is elected as the island ingress. B is a normal border host. A and B stay in both the CONTROL group and the DATA group. C is a non-border host, and only stays in the DATA group. A then accepts data from its overlay parent and multicasts them within the DATA group. Note that the incoming overlay paths of B and C are used to deliver control messages instead of streaming data. If A leaves the system, B will be elected as the new ingress since it is the only border host within the island. B will then resume data delivery along its overlay path and multicast data within the island.

2.3. Scheme Extension and Discussion

Using a single tree may not offer satisfactory service, because, firstly, hosts in the system are heterogeneous with different incoming and outgoing bandwidth. A host's incoming path may not be able to provide enough bandwidth for streaming. Secondly, quality degradation at a host (e.g., packet loss or host failure) affects all its descendants. In a highly dynamic P2P system, it is difficult for hosts to achieve high streaming quality with a single tree. To address these problems, we can use multiple description coding (MDC) to encode streaming data into multiple descriptions and distribute the descriptions along multiple trees [6].

In MDC, data is encoded into several descriptions. When all the descriptions are received, the original data can be reconstructed without distortion. If only a subset of the descriptions are received, the quality of the reconstruction degrades gracefully. The more descriptions a host receives, the lower the distortion of the reconstructed data is. Therefore, the source can encode its media content into M descriptions using MDC (where M is a tunable parameter), and transmit the descriptions along M different trees. Note that a host has different descendants in different trees. If the descendants of a host in different trees have low overlap, the packet loss due to a host will be distributed to all its descendants, and hence its impact is reduced.

Another important issue in streaming is loss recovery. Although MDC and multiple-tree transmission can improve resilience, packets may still be lost due to background traffic or path/host failure. A lightweight loss recovery mechanism is hence desired to deal with temporary packet loss. Traditional source recovery and parent recovery schemes have error correlation and implosion problems. To address this, we can consider using lateral error recovery (LER) [7]. LER randomly divides hosts into multiple planes and independently builds an overlay tree in each plane. A host needs to identify some hosts from other planes as its recovery neighbors. Whenever an error occurs, the host performs retransmission from its recovery neighbors.

3. ILLUSTRATIVE SIMULATION RESULTS

We have done simulations on Internet-like topologies to evaluate our scheme. We generate 10 *Transit Stub* topologies with GT-ITM. Each generated topology is a two-layer hierarchical network with a backbone of transit domains. Stub domains are all connected to the backbone. In our simulations, each topology has 8 transit domains and 256 stub domains, consisting of 3200 routers and about 20000 links. A group of hosts are randomly put into the network. A host is connected to a stub router with 1ms delay, while the delay of core links is given by the topology generator. From the stub domains that consist of at least one host, we randomly select some and set them to be multicast-capable. In our scheme, each new host obtains a number of (at most 10) randomly selected hosts from the RP when joining. A new host repeats the pinging iterations for at most 4 times and $k = 5$.

We also implement two tree-based ALM protocols for comparison, i.e., Narada and Overcast [2, 8]. Narada is one of the pioneering ALM protocols and its performance can serve as the benchmark. Overcast achieves low stress from the source to all receivers. We evaluate two important metrics in ALM, i.e., relative delay penalty (RDP) and link stress. RDP is defined as the ratio of the overlay latency from the source to a host to the delay along the shortest unicast path, and link stress is defined as the number of copies of a packet transmitted over a certain physical link.

Figures 3 and 4 show the RDP and link stress of different protocols, respectively, where we set 40% stub domains as multicast-capable. Overcast has the largest RDP. This is

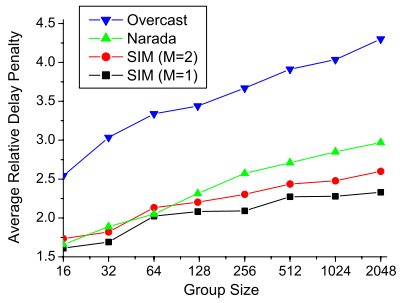


Fig. 3. RDP vs. group size.

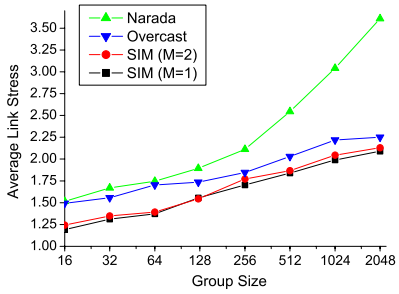


Fig. 4. Link stress vs. group size.

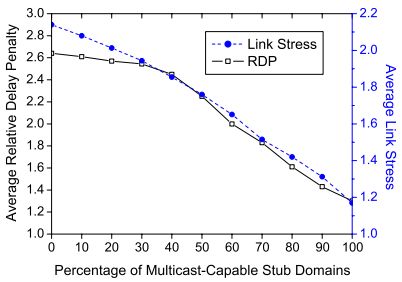


Fig. 5. SIM performance vs. percentage of multicast-capable stub domains.

because it always tries to insert a new host as far from the source as possible. Narada achieves much smaller RDP because it tries to minimize end-to-end delays. SIM has smaller RDP than Narada, especially when the group size is large. It shows that end-to-end delay can be efficiently reduced with our tree construction method and the utilization of IP multicast. If we construct two trees (i.e., $M = 2$), RDP slightly increases. This is because each host has two incoming paths and the slower one determines the end-to-end delay. Figure 4 compares link stress of different protocols. Overcast has much lower link stress than Narada, since Overcast targets maximizing bandwidth and accordingly minimizing link stress. SIM performs better than Overcast and Narada, because it selects appropriate parents for hosts and makes use of IP multicast. The stress does not depend much on the number of trees, because M does not affect the computation of link stress.

Figure 5 shows the RDP and stress of SIM versus different percentages of multicast-capable domains. The group size is 512. As expected, both the RDP and link stress decrease as the percentage of the multicast-capable domains increases. The improvement on RDP is not large when the percentage is less than 30%. This is because two hosts within the same multicast domain are not necessarily close together, therefore selecting a host from other domains as parent may introduce lower delay than simply receiving IP multicast packets in the island. On the other hand, the link stress can be efficiently reduced with the help of IP multicast. Note that even when all the stub domains are multicast-capable, the RDP and link stress are not equal to 1 as in pure IP multicast. This is because the transit domains are not multicast-capable.

4. CONCLUSION

Traditional ALM protocols only make use of unicast connections to form delivery trees and have not fully taken advantage of the local multicast capabilities. In this paper, we propose a fully distributed multicast scheme (called SIM) for media streaming which combines IP multicast with ALM. Hosts in SIM can distributedly detect multicast domains and use IP multicast if possible. Simulations results show that it can achieve low end-to-end delay and link stress.

5. REFERENCES

[1] X. Zhang, J. Liu, B. Li, and T.-S. Peter Yum, "CoolStreaming/DONet: A data-driven overlay network for efficient live media streaming," in *Proc. IEEE INFOCOM'05*, March 2005.

[2] Y. Chu, S. G. Rao, and H. Zhang, "A case for end system multicast," *ACM SIGMETRICS'00*, June 2000.

[3] Y. Chawathe, "Scattercast: An architecture for Internet broadcast distribution as an infrastructure service," PhD thesis, Univ. of California, Berkeley, Dec. 2000.

[4] K.-L. Cheng, K.-W. Cheuk, and S.-H. Chan, "Implementation and performance measurement of an island multicast protocol," in *Proc. IEEE ICC'05*, May 2005.

[5] K.-W. Cheuk, S.-H. Chan, and J. Lee, "Island multicast: The combination of IP multicast with application-level multicast," in *Proc. IEEE ICC'04*, June 2004.

[6] M. Castro, P. Druschel, A-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream: High-bandwidth multicast in a cooperative environment," in *Proc. ACM SOSP'03*, Oct. 2003.

[7] 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 *Proc. IEEE INFOCOM'04*, March 2004.

[8] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O'Toole, "Overcast: Reliable multicasting with an overlay network," in *Proc. OSDI'00*, Oct. 2000.