Challenges and Advances in Using IP Multicast for Overlay Data Delivery

Xing Jin, Oracle USA

Wanqing Tu, Glyndwr University

S.-H. Gary Chan, HKUST

This work was supported, in part, by the General Research Fund from the Research Grant Council of the Hong Kong Special Administrative Region, China (611107), and the Hong Kong Innovation Technology Fund (ITS/013/08).

ABSTRACT

IP multicast and overlay multicast have been proposed for one-to-many data delivery over the Internet. Compared to overlay multicast, IP multicast is less deployed but can achieve higher delivery efficiency. Researchers hence study how to combine IP multicast with overlay multicast in order to achieve both high deployability and high delivery efficiency. This combination is called island multicast. In this article we present a comprehensive survey of recent research on island multicast. We investigate the general architecture and key components of island multicast. We then discuss the core issue in island multicast: how to set up delivery connections across multiple multicast domains. We finally discuss open issues for future research.

INTRODUCTION

Various network services require point-to-multipoint or multipoint-to-multipoint data delivery among Internet users, such as video streaming and IP television (IPTV). Traditionally, there are two delivery techniques for this purpose: IP multicast (also known as network-layer multicast) and overlay multicast (also known as application-layer multicast [ALM]). In IP multicast routers are responsible for replicating and forwarding packets. In overlay multicast hosts replicate and forward packets. Figure 1 compares IP multicast and overlay multicast, where S is the source, H_1–H_3 are receivers (hosts), and R_1–R_5 are routers [1]. Figure 1a shows the IP multicast case, where routers R_1–R_5 form a router-level spanning tree to replicate and forward packets. In overlay multicast (as shown in Fig. 1b) S establishes unicast connections to H_1 and H_3. H_1 in turn delivers data to H_2. Hence, multicast is achieved via piece-wise unicast connections.

We further compare IP multicast and overlay multicast in Table 1. As shown, IP multicast requires multicast-capable routers, which have not been widely deployed over the Internet. Furthermore, IP multicast can use only UDP, and lacks mature mechanisms for congestion control and loss recovery. As a comparison, overlay multicast does not require any change to the Internet infrastructure. It is based on unicast, and can use existing congestion control and loss recovery solutions for unicast. Hence, overlay multicast is more deployable and manageable. On the other hand, IP multicast is more efficient than overlay multicast. As shown in Fig. 1, in IP multicast a packet is transmitted only once along each link. In overlay multicast a packet is often transmitted multiple times along the same link. In addition, end-to-end delay in overlay multicast is often higher due to host relay. A more detailed comparison between IP multicast and overlay multicast can be found in [2].

Although global IP multicast is not available yet, 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 overlay multicast, it would be beneficial if overlay multicast could make use of local multicast capability for data delivery. Hence, researchers have proposed to integrate IP multicast into overlay multicast (i.e., so-called *island multicast*). The Internet Research Task Force has formed a Scalable Adaptive Multicast Research Group to investigate overlay multicast, IP multicast, and hybrid approaches.

In a typical island multicast protocol IP multicast is used within islands, and islands are connected via unicast connections. While the basic idea is simple, there are many practical issues in system implementation. In this article we present a comprehensive survey of recent research on island multicast. We explore existing island multicast protocols and summarize a series of functional modules for island multicast. We then study the core issue in island multicast: how to connect multiple islands. We classify the state-of-the-art approaches into three categories:

- Island leaders set up interisland paths between themselves for data delivery [3–6].
- Island leaders select some specific hosts within their islands to set up interisland delivery paths [7–9].
- Some protocols do not designate leaders for islands. They determine interisland delivery paths based on a preconstructed overlay [10, 11].

We select representative examples from each category, and discuss their advantages and limi-

0163-6804/09/$25.00 © 2009 IEEE

	IP multicast	Overlay multicast
Multicast tree	Interior tree nodes are routers, and leaves are hosts	Both interior nodes and leaves are hosts
Deployment requirement	Require multicast-capable routers	Can be directly deployed on the current Internet
Transport layer connection	UDP	Can freely choose TCP or UDP
Scalability	Low — multicast-capable routers are not scalable	High — fully distributed and scalable protocols are available
Congestion control/loss recovery	No	May use existing solutions for unicast
Delivery efficiency	High	Low
Example protocols	DVMRP, MOSPF, PIM, CBT	NICE, Narada, Overcast

Table 1. *Comparison of IP multicast and overlay multicast.*

tations. Finally, we investigate two practical issues in island multicast and outline possible directions for future research.

Note that some island multicast protocols build source-specific trees [3, 4, 11], and some build source-unspecific trees [10]. To unify our discussion in this article, we assume that a specific source has been given for tree construction, and the source does not change during data delivery. The rest of the article is organized as follows. In the next section we describe the architecture of island multicast. We then study how to connect islands for data delivery. We then discuss open issues for future research. We conclude in the final section.

Architecture of Island Multicast

Classification of Island Multicast Protocols

Figure 2 shows the main functional modules in island multicast and demonstrates three ways to organize them. In the figure a new incoming host first detects an island and joins the island if there is any. After that, there are two possible ways to set up interisland connections.

Leader-Based Approach — Each island elects a unique leader. Leaders then help connect islands. This approach can be further divided into two categories.

In the first category interisland connections are set up between leaders. Usually, data are distributed via an overlay tree to all leaders. A leader then forwards packets to its island members via IP multicast. This is denoted *approach 1* in Fig. 2. Example protocols include [3–6].

We show an example in Fig. 3a. In the figure eight hosts (labeled H_1 through H_8) are distributed in three islands, I_1, I_2, and I_3. The leaders of the islands are H_1, H_4, and H_8, respectively. All the hosts are receivers. The source transmits packets to leader H_1, which in turn forwards packets to other leaders H_4 and H_8 via unicast. H_1 also relays packets to H_2 and H_3 via IP multicast. Similarly, H_4 multicasts its received packets to H_5 and H_6, and H_8 multicasts its received packets to H_7.

In the second category leaders select some specific hosts to set up interisland connections. Normally, each island has a unique ingress host (usually not the leader), which receives data from outside the island and multicasts them within the island. An island may also have some egress hosts, which forward data to other islands. This is denoted *approach 2* in Fig. 2. Example protocols include [7–9].

For example, in Fig. 3b H_1–H_8 are again eight hosts, among which H_1, H_4, and H_8 are island leaders. H_3 sets up a unicast interisland connection to H_4. Upon receiving data packets, H_3 forwards them to H_4, which in turn multicasts packets within island I_2. Similarly, H_2 forwards packets to H_7, and H_7 multicasts packets within I_3.

In this example H_3 and H_4 use a unicast connection between them to connect islands I_1 and I_2. They are called a pair of *bridge nodes* for the two islands. Likewise, H_2 and H_7 are the bridge nodes for I_1 and I_3. In addition, H_2 and H_3 are egresses of island I_1. H_1, H_4, and H_7 are ingresses of the three islands I_1, I_2, and I_3, respectively.

No Leaders — In a system with no leaders, all hosts join a single overlay after island joining. Based on the overlay, interisland connections can be set up. This is denoted *approach 3* in Fig. 2. A typical example is [10]. A host may also join the overlay before joining an island (e.g., [11]).

Figure 3c shows an example of [10]. The protocol requires a central server, which is usually independent of all hosts (either source or receivers) in the system. A host periodically measures round-trip time (RTT) to some other hosts and reports results to the server. The server then builds a minimum-diameter degree-bounded spanning tree on top of the hosts. Note that an interisland path is assigned a weight equal to its RTT (infinity if unknown), and an intra-island path is assigned weight –1. In this way hosts within the same island are connected together as a cluster, as shown in Fig. 3c. After that, all intra-island connections (i.e., H_1–H_2,

H_1–H_3, H_4–H_6, H_6–H_5, and H_7–H_8 in the figure) are used to deliver only control messages, and intra-island data delivery is achieved by IP multicast. Interisland connections (i.e., H_2–H_7 and H_3–H_4 in the figure) are used to deliver data packets.

Functional Modules in Island Multicast

We elaborate the functional modules in Fig. 2 as follows.

Island Detection and Joining — Traditional IP multicast does not detect islands, and a host does not know other hosts in its multicast group when joining. To detect islands, a specific detection process is needed. A possible approach is to require some island members to periodically multicast `HeartBeat` messages within their islands [4, 5, 7, 11]. After joining the island, a new host will receive `HeartBeat` messages if there are any. In another approach a new host sends a detection message to the multicast address when joining. Upon receiving the message, existing island members or some specific hosts are required to respond [3, 8, 10]. The new host then knows its island members.

Leader Election — A basic leader election process works as follows. All hosts in the island (or some selected hosts) multicast an election message consisting of their personal information (e.g., locations or IP addresses) within the island. Based on some predefined criteria, a certain host is elected as the leader. In detail, in [3] the first host in the island that requests data from the source is the island leader. In [5, 6] the first member of an island serves as the island leader. In [4, 7–9] the first host to send the election message becomes the leader. Clearly, leader election should be fast. If a leader leaves or fails, a new leader should be elected as soon as possible.

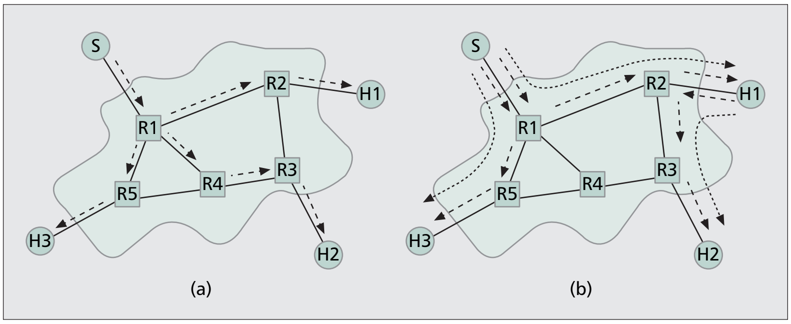
Ingress and Egress Election — The use of ingress and egress can reduce leader workload and improve delivery efficiency. In the next section we discuss how to select ingress and egress hosts.

Overlay Joining — A class of approaches requires all hosts to join an overlay before data delivery. The overlay is used to determine interisland delivery paths. We discuss two typical examples in the next section.

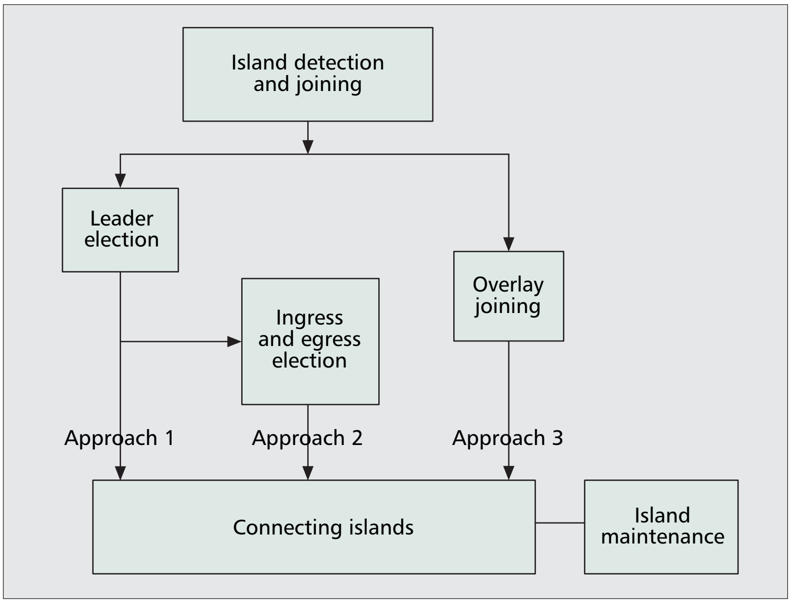
Connecting Multicast Islands — This is the most important component in island multicast. We need to connect multiple islands via unicast connections with the target of low end-to-end delay or high delivery rate. We discuss how to connect islands in the next section.

Island Maintenance — In a leader-based approach leaders are responsible for the maintenance of their own islands. In an approach with no leaders island maintenance is managed by specific components, depending on system design. In [10] it is managed by a central server, while in [11] it is managed by ingress hosts.

Depending on protocols, island maintenance has different tasks. In [11] an ingress host needs to periodically multicast specific `HeartBeat` messages within its island, which are used for island detection and ingress election. In [8] a leader is responsible for the leaving of its island members. If an ingress or egress leaves, the leader temporarily plays the role of host until a successor has been selected.



■ **Figure 1.** *Examples of IP multicast and overlay multicast (from [1]): a) in IP multicast, packets are replicated and forwarded by routers; b) in overlay multicast, packets are replicated and forwarded by hosts.*



■ **Figure 2.** *Functional modules and classification of island multicast protocols.*

Almost all existing island multicast protocols can be built from the above modules. Some of them (e.g., island detection and joining or leader election) do not have much design space. On the contrary, the core module (i.e., connecting multicast islands) may have diverse designs, leading to different delivery efficiency and different control overhead. In the next section we investigate this issue in detail.

Connecting Multicast Islands

Data Forwarding by Leaders

A system with island leaders may set up interisland connections between leaders. Without loss of generality, a host not within any island is regarded as an island consisting only of itself. The host is also regarded as the leader of its

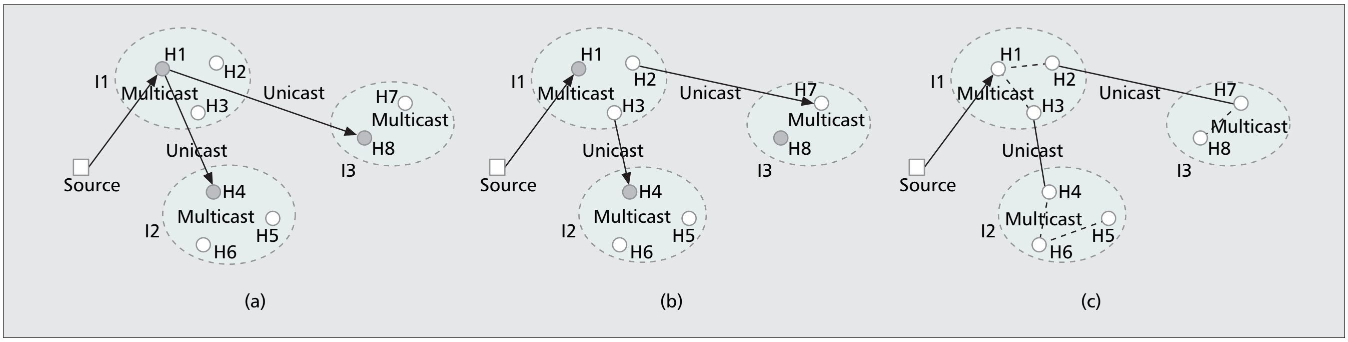


Figure 3. *Examples of island multicast: a) interisland forwarding paths are set up between leaders; b) interisland forwarding paths are set up between pairs of egress and ingress hosts; c) interisland forwarding paths are identified based on an overlay with no need for leaders.*

island. Generally, when a host joins the system, it first detects its island and joins the island if any. Hosts within the same island then maintain a unique leader. Data distribution is achieved in two steps. Leaders first join an overlay tree to obtain data. Each leader then multicasts data within its own island.

In subset multicast, the source sends a copy of data to each of the leaders [3]. Leaders then multicast data within their islands. As each island is connected to the source via a unicast connection, subset multicast is not scalable to large networks with many islands. In HMTP leaders (also called designated members) form an overlay tree through an overlay multicast protocol [4]. As the overlay tree construction process is fully distributed, and the source only needs to forward data to its children in the tree, HMTP is able to accommodate many islands. Other approaches may rely on a central server for tree construction [5, 6].

By abstracting an island into a leader, the issue of connecting islands becomes constructing an overlay tree among leaders. Existing overlay multicast protocols can then be applied. Therefore, this approach can benefit from the high diversity and efficiency of existing overlay multicast protocols. However, despite its simplicity, this approach has some limitations. First, a leader is responsible for data receiving, data forwarding, and island management. It has high nodal stress and heavy workload. Second, the delivery efficiency may not be high. For example, a supernode is often preferred in leader election due to island management considerations; but the resultant leaders may form a tree with high end-to-end delay. Furthermore, when islands are large it is not efficient to represent an island by a single leader, where two close islands may be connected by a pair of faraway leaders, and end-to-end delay is hence high.

Data Forwarding by Ingresses/Egresses

In order to reduce leader workload and improve delivery efficiency, some approaches select ingress and egress hosts for interisland data forwarding. Leaders can then focus on island maintenance.

Universal multicast is an extension of HMTP [7]. HMTP allows only one designated member in an island, but universal multicast allows multiple designated members in an island. So in HMTP, a designated member of an island is actually the island leader. But for an island in universal multicast, one of its designated members serves as the island ingress, and others serve as the island egresses. In detail, in universal multicast a designated member multicasts its `HeartBeat` messages with a certain time-to-live (TTL) value. These messages reach island members within a certain range of the designated member. Island members that do not receive `HeartBeat` messages then assume that their designated member has left and automatically elect a new designated member. In this way an island may have multiple designated members. After that, designated members within the same island elect a so-called head designated member. Based on the two-level hierarchy of designated members, an overlay tree can be formed on top of all designated members without routing loops and packet duplication in any island. In summary, in universal multicast each island has one ingress (i.e., the head designated member) and multiple egresses (i.e., the designated members other than the head). Furthermore, the head designated member also serves as the island leader for island management.

In universal multicast designated members are elected according to their locations in the island. It works as if dividing a large island into multiple small islands and electing a designated member in each of the small islands. Clearly, the selection of designated members within an island does not consider neighboring islands. Hence, two close islands may be connected by a pair of faraway designated members. To improve delivery efficiency, Cheuk *et al*. propose a set of mechanisms for bridge node selection (note that a pair of bridge nodes consists of one ingress and one egress) [8]. Leaders of islands first form an overlay tree using some overlay multicast protocol. If two leaders are directly connected in the overlay tree, their islands are called a pair of neighboring islands. Given a pair of neighboring islands, their leaders select a pair of bridge nodes to connect the islands. The first class of mechanisms is called *individual bridge node selection*, where a bridge node is selected independent of the other bridge node in its neighboring island. For example, a leader can select from its island members the one closest to the leader of the neighboring island as the bridge node. Another class of mechanisms is called *pair-wise bridge node selec-*

Classification		Connecting islands	Advantages	Limitations	Example protocols
Leader-based	Data forwarding by leaders	Each island elects a unique leader. Leaders set up interisland delivery paths between themselves.	Simple and low control overhead. May use existing overlay multicast protocols.	High nodal stress and heavy workload for leaders.	[3–6]
	Data forwarding by ingresses/egresses	Each island elects a leader. Leaders select ingress and egress hosts to set up interisland delivery paths.	Reduced workload at leaders. High delivery efficiency, especially for large islands.	High overhead for ingress/egress selection and maintenance.	[7–9]
No leaders		All hosts join an overlay tree. Hosts also detect and join islands. Interisland delivery paths are determined based on the tree and island information.	No need for leaders. Easy to implement and low control overhead.	Less control of interisland paths.	[10, 11]

Table 2. *Comparison of island multicast protocols.*

tion, where two neighboring islands cooperatively select a pair of bridge nodes. For example, given a pair of neighboring islands, all members of one island can ping all members of the other island. Among all the pairs, the pair with the smallest distance is selected as the bridge node pair. As all-pair pings are costly and not scalable, Yiu *et al*. propose to randomly ping some host pairs that interconnect two neighboring islands [9].

In Cheuk's and Yiu's approaches selection of bridge nodes depends on the locations of an island and its neighboring islands. By selecting close bridge node pairs, interisland delay can be reduced. As shown in [8, 9], proper selection of bridge nodes can significantly reduce end-to-end delay. As a comparison, in universal multicast the number of egresses in an island is determined by the island size, and two islands may be connected by a pair of faraway hosts. Universal multicast is hence less efficient.

Compared to leader-based data forwarding, this approach can reduce leader workload and often achieve higher delivery efficiency. On the other side of the coin, selection and maintenance of ingress/egress hosts cost additional overhead. In a dynamic system with frequent ingress/egress leaving, the maintenance overhead could be significantly high.

No Leaders

To reduce control overhead and simplify protocol implementation, some protocols do not designate leaders for islands. For example, Cheng *et al*. propose a centralized island multicast protocol [10]. In this approach a central server builds an overlay tree spanning all hosts. Interisland paths are then identified based on the overlay and island information. Please refer to the previous section and Fig. 3c for details of this protocol. As the protocol does not select leaders or bridge nodes, it is simple and highly deployable. To remove the central server from the system and improve system scalability, scalable island multicast (SIM) adopts a distributed tree construction method [11]. SIM first builds an overlay tree spanning all hosts. Hosts then detect a multicast island and join it. Hosts within the same island elect one ingress, which receives data from outside the island and IP multicasts them within the island. All other island members receive data from their ingress via IP multicast. In SIM egresses can be determined based on the tree and island information without further election. Its control overhead is kept low. Furthermore, it is fully distributed and scalable.

Despite their simplicity, these approaches have some limitations. When hosts form an overlay, they do not take island information into consideration. Therefore, the resultant interisland connections may not achieve high delivery efficiency. These approaches also lack mechanisms for flexible adjustment of interisland connections.

Comparison and Discussion

We compare the above island multicast protocols in Table 2. Approaches relying on leaders for data forwarding do not select ingresses or egresses. They have relatively low overhead, and can make use of existing overlay multicast protocols. However, this simple extension of overlay multicast puts heavy control overhead on leaders. Furthermore, in some cases it is not efficient to represent an island by a single leader, where the overall delivery efficiency may not be high.

Approaches using ingresses/egresses for data forwarding address the above two limitations. They put the data receiving and forwarding responsibility on ingresses and egresses. Leaders can then focus on island management. Careful selection of ingresses and egresses also improves delivery efficiency. On the other hand, selection and maintenance of ingresses/egresses lead to a complicated control mechanism and high overhead.

Different from the above leader-based approaches, approaches with no leaders do not select leaders for island connection. They require all hosts to join a single overlay. Based on the overlay and island information, a host can detect whether it is itself an ingress or egress. Some approaches like [11] need to elect ingresses. These approaches are simple to deploy and have low control overhead. On the other hand, as interisland connections are automatically determined, the delivery efficiency may be low if bad connections are used. They are hence not as flexible and adaptive as the approaches with ingress/egress.

Advanced Research Issues

In this section we discuss practical issues when applying island multicast to real network applications.

Loss Recovery

In island multicast a host may suffer packet loss due to path congestion or failure. In some services data packets received after a certain deadline are also regarded as lost. Therefore, timely loss recovery is important. Generally, there are two classes of recovery mechanisms for overlay multicast: proactive or reactive [1]. In a proactive approach a host sends redundant recovery packets besides data packets. If there is loss of data packets at the receiver end, the receiver can use redundant packets to reconstruct the original data. This class of approaches can be directly extended to island multicast.

Reactive recovery retransmits lost packets when loss occurs. Lateral error recovery (LER) is one example [12]. 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 a loss occurs, the host performs retransmission from its recovery neighbors. A variation of LER for recovery at ingress hosts in island multicast has been studied in [9]. It works best when jointly used with intra-island recovery like Scalable Reliable Multicast [13].

A more general LER-based recovery scheme for island multicast has been studied in SIM, which does not need additional intra-island recovery [11]. In LER trees in different planes are kept at similar sizes in order to balance recovery loads. But the balancing of multiple trees in a dynamic system requires high control overhead. SIM hence uses only one plane and builds a single delivery tree. The recovery neighbor of a host should satisfy the following requirements:

- Not reside in the host's subtree
- Not be the host's ancestor
- Not reside in the same island as the host

Here the third requirement is used because loss correlation between members of the same island is high. SIM's recovery scheme does not need to balance multiple trees, thereby introducing lower control overhead. However, it is not as efficient as LER. In LER a host and its recovery neighbor have disjoint overlay paths from the source, leading to low loss correlation between them. But SIM's recovery scheme cannot guarantee disjoint overlay paths from the source between a host and its recovery neighbor.

Forming a Mesh Network for High-Bandwidth Delivery

A popular application of island multicast is multimedia streaming. However, it has been shown that tree-based overlay does not perform well for streaming applications [14]. A tree is fragile and prone to severe service disruption, and an interior tree node might not have enough bandwidth for streaming to its children.

To address these issues, mesh-based multiple path delivery has been proposed [14]. In this approach hosts form a mesh network. Each host has multiple neighbors in the mesh, and periodically exchanges data with its neighbors. Thus, each host has multiple incoming paths. Even if a few neighbors or paths fail, the host can still receive data from other neighbors.

To integrate IP multicast into a mesh network, there are some additional issues to address. Within an island, if IP multicast paths cannot provide enough residual bandwidth, we need to set up other incoming paths for island members. However, this is not as simple as in a pure overlay network. For example, it is not clear whether two hosts in the same island should be neighbors in the mesh. On one hand, members within the same island often have high-bandwidth connections between each other. On the other hand, as mentioned, island members are often highly loss correlated. Therefore, it is a goal of future research to reduce loss correlation between island members and achieve efficient intra-island data exchange. Furthermore, a mesh network itself has high control overhead for mesh construction and maintenance. When using IP multicast, additional control mechanisms for island maintenance or leader election are introduced. We need to simplify system design and reduce control overhead.

Mesh network itself has high control overhead for mesh construction and maintenance. When using IP multicast, additional control mechanisms for island maintenance or leader election are introduced. We need to simplify system design and reduce control overhead.

Concluding Remarks

IP multicast and overlay multicast have their own advantages and limitations. A promising approach is to combine them together: so-called island multicast. In this article we study the general architecture and functional modules of island multicast. According to the method of connecting islands, we divide the existing solutions into three classes. We investigate representative examples in each category and qualitatively compare them. We finally outline some open issues for future research.

References

[1] X. Jin, W.-P. Yiu, and S.-H. Chan, "Loss Recovery in Application-Layer Multicast," *IEEE Multimedia*, vol. 15, no. 1, Jan.–Mar. 2008, pp. 18–27.

[2] A. Ganjam and H. Zhang, "Internet Multicast Video Delivery," *Proc. IEEE*, vol. 93, no. 1, Jan. 2005, pp. 159–70.

[3] J. Park *et al.*, "Multicast Delivery based on Unicast and Subnet Multicast," *IEEE Commun. Lett.*, vol. 5, no. 4, Apr. 2001, pp. 1489–99.

[4] B. Zhang, S. Jamin, and L. Zhang, "Host Multicast: A Framework for Delivering Multicast to End Users," *Proc. IEEE INFOCOM '02*, June 2002, pp. 1366–75.

[5] B. Chang, Y. Shi, and N. Zhang, "HyMoNet: A Peer-To-Peer Hybrid Multicast Overlay Network for Efficient Live Media Streaming," *Proc. AINA '06*, Apr. 2006.

[6] S. Lu *et al.*, "SHM: Scalable and Backbone Topology-Aware Hybrid Multicast," *Proc. ICCCN '07*, Aug. 2007, pp. 699–703.

[7] B. Zhang *et al.*, "Universal IP Multicast Delivery," *Comp. Net.*, vol. 50, no. 6, 2006, pp. 781–806.

[8] K.-W. Cheuk, S.-H. Chan, and J. Lee, "Island Multicast: The Combination of IP Multicast with Application-Level Multicast," *Proc. IEEE ICC '04*, June 2004, pp. 1441–45.

[9] W.-P. Yiu, K.-F. Wong, and S.-H. Chan, "Bridge-Node Selection and Loss Recovery in Island Multicast," *Proc. IEEE ICC '05*, May 2005, pp. 1304–1308.

[10] K.-L. Cheng, K.-W. Cheuk, and S.-H. Chan, "Implementation and Performance Measurement of an Island Multicast Protocol," *Proc. IEEE ICC '05*, May 2005, pp. 1299–1303.

[11] X. Jin, K.-L. Cheng, and S.-H. G. Chan, "Scalable Island Multicast for Peer-to-Peer Streaming," *Advances Multimedia*, vol. 2007, 2007, article ID no. 78913.

[12] W.-P. Yiu *et al.*, "Lateral Error Recovery for Media Streaming in Application-Level Multicast," *IEEE Trans. Multimedia*, vol. 8, no. 2, Apr. 2006, pp. 219–32.
[13] S. Floyd *et al.*, "A Reliable Multicast Framework for Lightweight Sessions and Application Level Framing," *IEEE/ACM Trans. Net.*, vol. 5, no. 6, 1997, pp. 784–803.
[14] X. Zhang *et al.*, "CoolStreaming/DONet: A Data-Driven Overlay Network for Peer-to-Peer Live Media Streaming," *Proc. IEEE INFOCOM '05*, Mar. 2005, pp. 2102–11.

BIOGRAPHIES

XING JIN (xing.jin@oracle.com) received his B.Eng. degree in computer science and technology from Tsinghua University, Beijing, China, in 2002, and his Ph.D. degree in computer science and engineering from the Hong Kong University of Science and Technology (HKUST), Kowloon, in 2007. He is currently a member of technical staff in the Systems Technology Group at Oracle, Redwood Shores, California. His research interests include distributed information storage and retrieval, peer-to-peer technologies, multimedia networking, and Internet topology inference. He is a member of Sigma Xi and the IEEE Communications Society Multimedia Communications Technical Committee.

WANQING TU (w.tu@glyndwr.ac.uk) received her Ph.D. degree in computer science from City University of Hong Kong, Kowloon, in 2006. She is currently a lecturer in the School of Computing and Communications Technology, Glyndwr University, Wrexham, United Kingdom. Her research interests include QoS, overlay networks, wireless mesh networks, end host multicast, and distributed computing. She received the Embark Postdoctoral Fellowship of Ireland in 2006.

S.-H. GARY CHAN (gchan@cse.ust.hk) received his B.S.E. degree (Highest Honor) in electrical engineering from Princeton University, New Jersey, in 1993, with certificates in applied and computational mathematics, engineering physics, and engineering and management systems, and his M.S.E. and Ph.D. degrees in electrical engineering from Stanford University, California, in 1994 and 1999, respectively, with a minor in business administration. He is currently an associate professor with the Department of Computer Science and Engineering, HKUST, and an adjunct researcher with Microsoft Research Asia, Beijing. His research interests include multimedia networking, peer-to-peer technologies and streaming, and wireless communication networks. He is a member of Tau Beta Pi, Sigma Xi, and Phi Beta Kappa. He served as a Vice-Chair of IEEE Communications Society Multimedia Communications Technical Committee from 2003 to 2006. He was a Guest Editor for *IEEE Communication Magazine*, Special Issue on Peer-to-Peer Multimedia Streaming (2007) and Springer *Multimedia Tools and Applications*, Special Issue on Advances in Consumer Communications and Networking (2007). He was Co-Chair of the Multimedia Symposium for IEEE ICC 2007. He was the Co-Chair for the workshop on Advances in Peer-to-Peer Multimedia Streaming for the ACM Multimedia Conference 2005 and the Multimedia Symposia for IEEE GLOBECOM 2006 and IEEE ICC 2005.