Journal of Information Processing Vol.20 No.2 347–357 (Apr. 2012)

[DOI: 10.2197/ipsjjip.20.347]

Regular Paper

ODMT: On-demand Inter-domain Multicast Tunneling

Achmad Basuki[1,a)] Achmad Husni Thamrin[1] Hitoshi Asaeda[1] Jun Murai[2]

Received: May 30, 2011, Accepted: November 7, 2011

Abstract: The tussle in IP multicast, where different enablers have interests that are adverse to each other, has led to a halt in inter-provider multicast deployment and created a situation in which enabling inter-domain multicast routing is considered a deterrent for network providers. This paper presents ODMT (On-demand Inter-domain Multicast Tunneling), an on-demand inter-provider multicast tunneling that is autonomous in operation and manageable through its definable policy control. In the architectural design, we propose a new approach of enabling inter-provider multicast by decoupling the control-plane and forwarding-plane. Focusing on the control-plane without changing the forwarding-plane, our solution changes the traditional open multicast service model into a more manageable service model in inter-domain multicast operation, hence it eases the Internet-wide multicast deployment.

Keywords: inter-domain IP multicast, on-demand tunnel, overlay network

1. Introduction

Live broadcasts streamed to large audiences spanning several time zones (e.g., important speeches, world sporting events, notable lectures) can generate an unprecedented amount of Internet traffic. US President Obama's speech and the 2010 FIFA World Cup are such examples [1]. The convergence of broadband Internet and broadcasting services to grab larger audiences [2], [3], [4] can increase the amount of video traffic on the Internet beyond what has been currently estimated [5], [6]. These typical resource intensive services (i.e., high bandwidth and low delay) were supposed to be served by IP multicast as was envisioned over two decades ago [7], but unfortunately it is still unavailable on the Internet.

Despite a significant increase in multicast networks deployed for live broadcast services (e.g., IPTV services) [8], [9], such networks are simply *isolated multicast islands* [10] due to the absence of inter-domain multicast. Previous researchers [11], [12], [13], [14], [15], [16] have criticized the *open multicast service* model and its protocol complexity as the root problem in lack of Internet-wide multicast deployment. The open multicast service model and its routing protocol complexities make the operation of inter-domain multicast difficult to manage. The issue with open multicast service model is the lack of control to hosts which send and receive multicast traffic, while the issue with IP multicast protocol complexity is the management problem of massive multicast states on all intermediate (transit) routers. The lack of control function on enabling inter-domainm ulticast is considered a deterrent to network providers and has led to a halt of inter-provider multicast. A scalable and easy to deploy multicast routing protocol is not the only important requirements; we also consider a manageable inter-domain multicast service model to be as important as other criteria in order to revive Internet-wide multicast deployment.

The severity of the inter-domain multicast routing scalability and complexity problem has also meant that a number of proposals have focused on reducing the multicast forwarding state entries and on addressing implementation simplicity [16], [17], [18], [19]. Even though these proposals provide a scalable solution and ease of deployment, they are not principal changes in the open multicast service model. The concern regarding an open multicast service model in terms of inter-domain deployment is an uncontrollable multicast distribution tree that makes monitoring and accounting on multicast access difficult, as discussed in Ref. [16]. This has perhaps contributed to the fact that none of these models has seen deployment.

This paper directly addresses the issue of an open multicast service model in the inter-domain area. We propose a much simpler approach to interconnect isolated multicast islands using an on-demand multicast tunneling mechanism, hence we called it as ODMT. ODMT offers two key advantages over existing solutions: 1) *autonomous and seamless operation of inter-domain multicast* - provides on-demand inter-domain multicast tunneling that avoids deployment dependency on the contiguous upstream provider and does not require modification to the existing multicast routing protocol. 2) *increased possibility of multicast interconnection with flexible operation* - provides a measure of control and deployment strategies by enabling flexible choices of multicast tunnel providers to perform overlay inter-domain multicast peering.

These advantages are achieved through the architectural design. To avoid the complexity of an inter-domain multicast operation, ODMT decouples the control and forwarding planes. This separation enables ODMT to provide on-demand inter-domain

[1] Graduate School of Media and Governance, Keio University, Fujisawa, Kanagawa 252–0882, Japan
[2] Faculty of Environment and Information Studies, Keio University, Fujisawa, Kanagawa 252–0882, Japan
[a)] abazh@sfc.wide.ad.jp

multicast distribution tree and overlay multicast transit through non-contiguous upstream network providers. ODMT adds a few important network components (Section 3) to allow independent deployment by any ISP using their existing multicast routing platform without the need for hardware and software upgrades.

Our key contribution in this paper is a new approach in enabling inter-domain multicast with minimal changes, while at the same time introducing manageable service model of inter-domain multicast operation. We hope this solution will help to overcome the barrier of inter-provider multicast deployment to achieve the vision of ubiquitous multicast availability on the Internet.

The primary focus of this paper is the design and implementation of ODMT. We lay the background of our work in Section 2, then discuss the design approach and implementation in Sections 3 and 4, respectively. We present the deployment issues and analysis of ODMT in Sections 5 and 6 and the related work in Section 7. Finally, Section 8 provides some concluding remarks.

2. Current Multicast Deployment

Efforts to enable multicast on the Internet have been underway since the early 1990s with MBone [20], but, through the years, this effort has diminished and multicast has failed to see Internet-wide deployment. In contrast, IP multicast deployment in certain supervised networks (i.e., IPTV networks, enterprise networks, financial networks) has enjoyed significant success.

2.1 Protocols Evolution

The first multicast routing protocol to support IP multicasting was Distance Vector Multicast Routing Protocol (DVMRP) [21]. DVMRP was introduced to facilitate multicasting at an early stage with a usage model called Any-Source Multicast Model (ASM) [22]. Many-to-many services (i.e., multiplayer games, multi-party conferencing) are classified under this ASM usage model. Todays, the usage of this protocol is considered as legacy deployments. RFC 5110 [23] summarizes the deployment status of different multicast routing protocols and shows that Protocol Independent Multicast - Sparse Mode (PIM-SM) is the only multicast routing protocol that is actually deployed for both inter-domain and intra-domain areas. In the current state, PIM-SM [24] is considered as the de-facto multicast routing protocol.

PIM-SM, as its name implies, builds and maintains the multicast distribution trees (sometimes called multicast forwarding states) independent of any particular underlying topology-gathering protocol to populate its multicast routing table (MRIB). This means that the multicast distribution tree of PIM-SM can be influenced by other routing protocols. PIM-SM was designed to scale up to inter-domain Internet operation [25]; however, since it does not exchange multicast routing tables with other PIM-SM routers, it has to be complemented by other routing protocols. The Multiprotocol Border Gateway Protocol (MBGP) [26] is often used and supports non-congruent multicast routes to the unicast ones. The combination of these two protocols would have been enough to enable multicast on the Internet. Unfortunately, those two protocols are not sufficient to support ASM usage model for inter-domain multicast. ASM usage model requires another protocol to inform the existence of active sources from one domain to the other domains. This requirement makes inter-domain multicast operation becomes inherently complex.

The complexity of employing ASM lies in its way of learning the active sources. A PIM-SM router in a domain needs to first know its PIM rendezvous point (RP) to learn the active sources. It became apparent that when a source is located in another PIM-SM domain, a coordination mechanism between the RPs from multiple PIM-SM domains is needed; Multicast Source Discovery Protocol (MSDP) [27] was invented for this purpose. Therefore, PIM-SM with MSDP and MBGP is a common scheme found in current inter-domain multicast deployment. Unfortunately, this scheme is only valid for IPv4 and MSDP itself has several scalability problems and is quite vulnerable to attacks, as discussed in Ref. [15]. Moreover, the IETF has no intent to define MSDP for IPv6 and has made Embedded-RP available instead [23].

The Source-Specific Multicast (SSM) [28] usage model was later invented and is suitable for one-to-many service where the sources are definite (e.g., IPTV, stock quotes, etc.). When SSM is used, inter-domain routing apparently becomes much simpler and does not require a source discovery protocol such as MSDP. The receiver host must signal the router using a specific join message to (source, group) channel. Interested receivers can learn the source address from an out-of-band mechanism, such as Web or through other means to subscribe to the (source, group) channel. The PIM-SM protocol has been designed to support ASM and SSM usage models. Implementation-wise, PIM-SSM is a subset of PIM-SM function, and thus most current routing platforms already support it. The combination of PIM-SSM and MBGP should be sufficient to enable broadcast service (SSM-like model) on the Internet.

2.2 Open Multicast Service Model

The Internet is composed of interconnection between heterogeneous entities called Autonomous Systems (ASes), where the AS interconnection is formed on a contractual basis between ISPs, which mostly classified as *transit* and *peering*. This AS interconnection determines how packets are routed from source and destinations. Gao [29] has identified that more than 99% of Internet paths conform *valley-free* inter-domain path model, in which packets are first forwarded over *uphill* path using only provider links and as soon as possible forwarded to *downhill* path, maybe over a peering link, using only customer links as illustrated in **Fig. 1**. Faratin et al. [30] view today's type of ISPs as *content* ISPs and *eyeball* ISPs, where packets mostly flow from customers of *content* ISPs who provide contents to customers of *eyeball* ISPs who access the contents. We use these concepts to analyze the Internet-wide multicast deployment problem.

Deploying inter-domain multicast on the Internet will obviously help broadcast services scale their distribution to a larger number of potential users, and at the same time save bandwidth. However, this bandwidth saving becomes cost/revenue problem between ISPs as it breaks the common inter-provider peering settlement on the basis of traffic volume [31] and has been regarded as multicast *tussle* [32] as depicted in Fig. 1. This basic multicast tussle has hindered multicast from being widely available on the

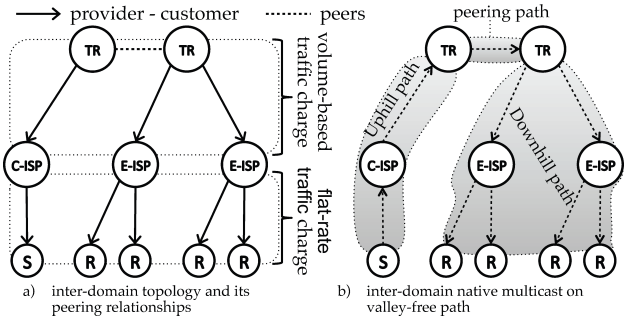

Fig. 1 Inter-domain multicast tussle with regard to peering relationships. A circle describes an AS on the Internet, where C-ISP, TR, E-ISP describe *content* ISP, transit ISP, and *eyeball* ISP, respectively. Enabling multicast between source AS (S) – C-ISP and E-ISP – receiver AS (R) provides a positive incentive as it reduces the possible traffic bottleneck and the outgoing/incoming traffic costs. However, it introduces a negative incentive for transit ISPs, as they see a reduced revenue due to the decreasing transit traffic volume.

Internet.

Open multicast service model has been considered as the root problem in lack of Internet-wide multicast deployment, in which the open service model does not provide enough control to hosts which send or receive multicast traffic. Diot et al. [11] had emphasized that without changing this service model, the Internet-wide deployment of IP multicast will remain stalled. They argued that address allocation, access control and inter-domain management as functions that must be added in an implementation that offers an alternate service model. Realizing this problem and multicast tussle above, we need a more manageable inter-domain multicast service model that can enable each type of ISP to decide their incentives (reducing or increasing traffic volume) to comply with their existing inter-domain peering relationships, which at the end could overcome the barrier to Internet-wide multicast deployment.

3. ODMT Design

In this section, we illustrate the key components of ODMT, as it aims to interconnect the existing multicast islands to enable Internet-wide multicast availability. To achieve this goal, the design of ODMT must meet these requirements.

First, to be practical, ODMT must be seamlessly integrated to the current Internet routing platform. This implies that ODMT must be able to run without requiring neither modification on the current de-facto multicast routing protocol nor hardware upgrades on the existing routing platform. **Second**, ODMT needs to be autonomous in its operation. This implies that ODMT must be able to work on-demand to connect two isolated multicast islands and then disconnect when not needed. Moreover, the control policy to govern the multicast interconnection should be configurable. Therefore, in our design, we do not depend on the availability of contiguous multicast peering. Instead, each site may directly initiate overlay multicast peering to the source's site or via another overlay multicast transit providers based on the local policy choices. **Finally**, the design must support incremental deployment and co-exist smoothly with native inter-domain multicast deployment.

IP Multicast has been considered as an "all or nothing" or "clean-slate" service, which has to be enabled on every node along the path from the sender to the receivers. The multicast states must exist at all the intermediate nodes (transit providers) to guarantee the multicast packet duplication. For inter-domain operation, multicast states are considered undesirable because the transit providers have no control of it. To avoid this situation, bypassing those intermediate nodes using tunneling would be the logical approach. This is similar to the MBone deployment technique [20] where many routers were not multicast-capable. From an operational point of view, instead of using a well-known static tunneling, we need an on-demand inter-domain multicast tunneling. ODMT design relies on the well-known tunneling technique to encapsulate multicast packets crossing the unicast paths in order to interconnect isolated multicast islands. Tunneling is known to have several performance penalties [33]. However, with current advances in hardware-accelerated tunnel interface and hardware-enabled multicast forwarding found in modern routers [34], [35], [36], tunneling no longer imposes significant performance penalties. Our own measurement results in **Table 1** also confirmed this fact. The CPU usage of route processor in each router platform remained low even though tunneling and multicast forwarding were used.

Table 1 Tunneling and multicast forwarding on modern routers with hardware acceleration support. The performance index is the percentage of CPU usage where lower value is better. For the measurement of each of the results, a 30 Mbps UDP stream was generated using iperf [37]. The payload size of each UDP packet was 1,372 bytes. While UDP traffic was streamed, CPU usage was polled from each router every 5 seconds over a period of 10 minutes.

Platform	Tunnel		Native	
	Unicast	Multicast	Unicast	Multicast
Juniper MX80	avg. 0.82% std. 0.38%	avg. 0.83% std. 0.38%	avg. 0.80% std. 0.40%	avg. 0.80% std. 0.44%
Cisco 7200 (NPE-G1)	avg. 3.71% std. 1.57%	avg. 5.02% std. 1.28%	avg. 3.51% std. 0.94%	avg. 4.54% std. 1.24%

3.1 Control- and Forwarding-plane Separation

Instead of proposing a new multicast routing protocol which includes a tunneling mechanism, ODMT decouples the tunnel peering establishment and multicast forwarding mechanism. The advantage of this separation is that, once multicast route to the source site has been established through tunnel peering, the existing multicast forwarding mechanism (i.e., PIM-SM) can be used independently. In this way, ODMT complements the de-facto PIM-SM for inter-domain operation with more controllable multicast distribution mechanisms crossing the Internet.

The separation of tunnel peering establishment which is under the purview of control plane and multicast forwarding mechanism which is under the purview of forwarding plane is depicted in **Fig. 2**. ODMT consists of three main components, namely Controller Node (CoN), Forwarding Node (FoN) and Controller node Resolver (CR). The FoN as its name implies, is a router object controlled by the CoN which supports PIM-SSM routing. The CoN is an authorized controller node for a PIM-SM domain, while the CR is a directory service for CoN lookups. An inter-domain multicast connection using tunnels is facilitated by the CoN and CR for discovering and negotiating with other CoN do-

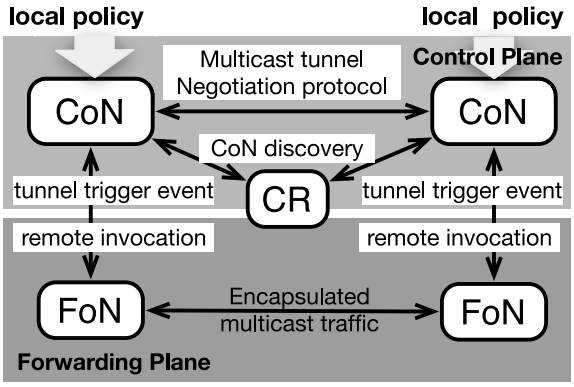

Fig. 2 General abstraction of ODMT.

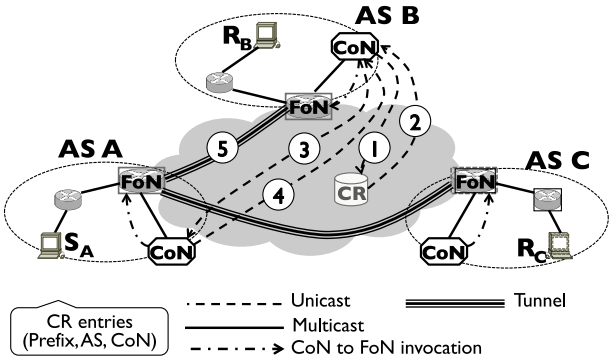

Fig. 3 ODMT working concept. A receiver at AS B network wants to receive multicast packets from a source at AS A network.

mains.

To provide a manageable service model in inter-domain multicast, we leverage the SSM service model into the process of inter-island multicast tunneling; in fact, the findings in Ref. [38] justify this usage model in the inter-domain multicast. We require each receiver site (one multicast island) to explicitly initiate a source-specific multicast join (PIM-SSM join) to a service on other multicast islands. This PIM-SSM join will trigger the tunnel peering creation by the CoN to its FoN and subsequently, CoN will control multicast routes to govern the multicast packets flow over the created tunnel. A pseudo PIM-SSM router function at the CoN is needed to correctly capture the triggered multicast join event. In this design, a measure of control of multicast packet flow is available by defining the local multicast policy to supervise the multicast tunnel creation and forward the multicast joins. This measure of control is in-line with the general direction given by Ref. [32], where ODMT derives its multicast control policy from the local policy inputs.

3.2 Controller Node

CoN holds a key role in this design, as it is meant to be the controller for performing the multicast tunneling between inter-domain routers. Taking advantage of the fact that PIM-SM protocol is independent of the underlying routing protocol, CoN may influence the flow of PIM join messages either through the tunnel link or the native link by injecting static multicast routes. We now use a concrete example to illustrate how ODMT interconnects two isolated multicast islands. **Figure 3** shows the relevant components of ODMT; AS is an autonomous system network representing a Site or a multicast island. We used the terms Site and multicast island interchangeably in this paper. A receiver R_B at AS B wants to receive multicast packets from a source S_A at AS A. For the initial state, the FoN at each site must configure a default multicast route via each site's CoN to ensure that CoN receives PIM (S, G) joins. The following are the steps to interconnect two isolated multicast islands, explaining the steps in Fig. 3 :

1) Upon receiving a PIM (S_A, G) join, CoN_B takes S_A IP address as an input to query CR to discover CoN to connect to.
2) The CR replies to CoN_B with data consisting of CoN_A IP address, IP Prefix and AS number information, which relates to S_A IP address. The CR lookup procedure evaluates the S_A IP address by finding a matching IP prefix with the corresponding CoN IP address.
3) After discovering the CoN_A IP address from the CR, CoN_B initiates a multicast tunnel peering request to the CoN_A.
4) CoN_A will evaluate the request according to its local policy configuration. If the request is approved, CoN_A and CoN_B will start exchanging information of each FoN IP address for a tunnel creation.
5) Finally, each CoN at AS A and AS B will invoke commands for creating the tunnel and enabling the PIM routing function on the created tunnel interface.

Upon the successful tunnel creation, CoN_B will inject a static multicast route for the source destination IP prefix obtained from CR reply into FoN_B in order to divert PIM (S_A, G) join over the created tunnel. Subsequently, FoN_A will receive a PIM (S_A, G) join and forward it to CoN_A. This PIM (S_A, G) join will trigger a lookup procedure at CoN_A similar to the Step 1 above. Since the S_A address belongs to AS A network itself, CoN_A will not query the CR, instead CoN_A must inject a static multicast route at FoN_A to divert the PIM (S_A, G) join to the correct next hop router toward the source S_A. CoN_A will perform route lookup on the unicast routing table at FoN_A for the longest prefix match to the source destination IP S_A. The result will be used for static multicast route injection. Finally, the receiver R_B can start to receive multicast packets from the source S_A. As long as the tunnel between FoN_A and FoN_B is still active (i.e., there are active receivers), other receivers at Site B can receive multicast packets from other sources at Site A, and vice versa without initiating another tunnel request.

3.3 Static Multicast Routes Injection

The initial condition in Section 3.2 requires that the FoN to be configured to have a default multicast route to the CoN. This requirement is important so that the CoN can control every multicast tunnel peering creation. Once a multicast tunnel has been established, ODMT relies on PIM-SM protocol for the multicast routing. However, PIM-SM will not immidiately route multicast packets over the newly created tunnel link because there is no multicast route associated to the tunnel link. Therefore, ODMT needs to inject static multicast routes to influence the multicast routing by PIM-SM over the created tunnel link.

As outlined in the previous section, there are two conditions of

static multicast route injection; 1) The source address is within its own multicast island, and 2) The source address is in another multicast island. To inject a multicast route, CoN needs to know the IP prefix that corresponds to the source IP address. This is for the purpose of aggregating all traffic belonging to a Site over a tunnel link. When the source address is within its own multicast island, the CoN learns the routes from the unicast routing tables of its FoN and injects an appropriate multicast route toward the multicast source. The source IP prefix is obtained from a lookup procedure using the longest prefix match to the multicast source IP address. When the source address is on another multicast island, the source destination IP prefix is obtained from the process of querying CR for CoN.

The use of static routes is very common and useful in network operation, however it has drawbacks, namely it is not controllable and difficult to monitor and troubleshoot. ODMT makes it more manageable by storing every static multicast route injection and monitoring their states periodically. If there are changes in the underlying unicast routing tables, the static multicast route will be updated appropriately.

3.4 Multicast Tunnel Peering Establishment

Tunneling has long been recognized as a possible solution to reduce the number of intermediate nodes that maintain multicast forwarding states along the path from the source to the receivers when the group members are sparse and well spread [39], [40]. ODMT's multicast tunnel peering establishment is employed in the CoN. The tunnel peering establishment describes the negotiation protocol among CoNs. The general view of the negotiation protocol can be summarized in two general functions as was depicted in step 3 and 4 of Section 3.2:

(1) The initiator of multicast tunnel peering - An initiator refers to the Site that initiates a multicast tunnel peering request. An Initiator may act as a receiver-only or a transit Site. The multicast tunnel peering initiation must be preceded by a tunnel trigger events coming from a PIM (S, G) join.

(2) The acceptor of multicast tunnel peering - An acceptor refers to the Site which accepts a multicast tunnel peering request coming from a Site. A Site that has the source or acts as a transit on behalf of the source Site falls into this category.

For both categories, the approval of multicast tunnel peering is subject to the local policy as depicted in Fig. 2. Additionally, multicast tunnel peering states are stored for monitoring purposes and will be deleted when the tunnel is unused.

3.5 Networked-overlay Multicast Transit

In the design of ODMT, a source Site is responsible for handling all multicast tunnel peering requests from many receiver Sites. A source Site may no longer able to accept new multicast tunnel peering requests. In that case, the CoN at the source Site may opt to delegate new multicast tunnel peering requests to other sites that have established multicast tunnel with the source Site (in other words, offloading tunnel requests to a transit Site). We call this procedure as networked-overlay multicast transit mechanism. This mechanism can reduce the inherent drawbacks of tunneling [41]: 1) High fanout of tunnel endpoints at the source Site. 2) Redundant multicast packets from multiple receiver Sites traversing over the same physical link to form tunnels to the source Site.

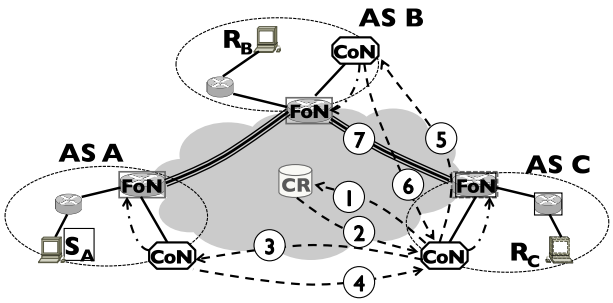

Fig. 4 ODMT Transit concept. CoN at AS A network refuses to accept multicast tunnel peering request from CoN at AS C network, because AS A network cannot accept more tunnel peering. CoN at AS A network suggests CoN at AS C network to negotiate multicast tunnel peering with CoN at AS B network.

We use the concrete example in **Fig. 4** as an illustration to describe the networked-overlay multicast transit mechanism. This example is elaborated from the previous example in Fig. 3 and assumes that AS A and AS B networks are already connected to each other. The receiver R_C at AS C wants to receive multicast packets from the source S_A at AS A. The same initial procedures outlined in the previous example in Section 3.2 also apply to this example, where a PIM (S, G) join triggers the tunnel request initiation and subsequently CoN_C queries the CR to get the CoN of AS A. The only difference is the addition of two steps required to achieve this transit mechanism. The two additional steps are step 4 and 5 as explained below and likewise represented as step 4 and 5 in Fig. 4:

4) CoN_A will evaluate the request according to its local policy configuration. If Site A can no longer accept new multicast tunnel peering request, CoN_A informs CoN_C with a list of CoN which has tunnel peering established or a contract to be a networked-overlay multicast transit on behalf of AS A. In this example, Site B is listed as the transit Site.

5) Upon receiving the list of CoNs, CoN_C will evaluate the appropriate CoN to negotiate with according to the local policy. CoN_C will pick up the CoN address from the list in recursive manner and initiate multicast tunnel peering request until a CoN accepts its request. In this example, Site B has connected to Site A and is willing to be a networked-overlay multicast transit for the multicast sources at Site A. So that, CoN_C initiate a multicast tunnel peering request to CoN_B.

Upon the successful tunnel creation, CoN_C will inject a static multicast route for a source destination IP prefix to FoN_C, which was gathered earlier from the CR reply, to divert PIM (S_A, G) join over the created tunnel. Subsequently, FoN_B will receive the PIM (S_A, G) join from FoN_C. Assuming that FoN_B still has an active (S_A, G) state, the multicast packets will immediately start to flow to FoN_C. Finally after all of the above processes are complete, a receiver R_B will be able to receive the multicast packets from the source S_A.

3.6 Local Policy

ODMT provides a measure of control in the establishment of multicast tunnel peering through the configuration of local pol-

icy, as stated in earlier sections. Local policy is driven by the incentives at each Site, which can be determined through definable metrics and contractual multicast tunnel peering agreement (in the case of networked-overlay multicast transit). The metrics may include the maximum number of tunnel peering that a Site can handle and the policy to offload tunnel peering requests to the transit Sites. In determining the maximum number of tunnel peering, each site's hardware and bandwidth resources must be taken into account.

A source Site that prefers to offload every tunnel request to its networked-overlay multicast transit Site has to define its preferred list of transit CoNs in the local policy. For each multicast tunnel peering from receiver Sites, the source Site will instruct the receiver Site to peer with its preferred list of transit CoNs. Likewise, the receiver Site will select its preferred CoNs during the establishment of multicast tunnel peering using its own local policy.

In the absence of a local policy, for each multicast tunnel peering request from the receiver Site, the source Site will give a list of all connected Sites as its networked-overlay multicast transit. The receiver Site will try to peer with the given list of CoNs until one of the CoNs accepts its request. When no transit Site accepts tunnel request, the receiver Site will re-initiate a tunnel request back to the source Site.

3.7 Controller Node Resolver

To initiate multicast tunnel peering, the CoN at the receiver Site needs to know the IP address of the CoN at the source Site. This mechanism is similar to common bootstrapping communication on the Internet in which a user obtains the IP address of a destination. The information can be acquired via multiple approaches, e.g., directory service, web search, or any out-of-band communication. In our design, we propose Controller node Resolver (CR), which is a CoN lookup service that provides the information regarding IP address prefixes and AS numbers with its corresponding CoN IP addresses. CR is a separate entity in the design of ODMT. While this paper does not discuss the specific implementation of the CR service, the CR service can be implemented in a hierarchical manner like the domain name service (DNS) [42] or in a flat structure [43].

If a source Site wants to be reachable by other Sites, it needs to register its CoN's information at a CR server. When a receiver Site intends to connect to a source Site, the receiver Site will query the CR to retrieve CoN's information on the source Site, similar to querying a DNS server in today's Internet. Once the receiver Site gets the information of the CoN at the source Site, it will start to negotiate the multicast tunnel peering, as described earlier in Section 3.2. When the source Site has the updated information about its CoN due to changes in the network, it needs to update its CR record. This process is similar to dynamic DNS [44], [45] updates or other directory service updates. We do not think record updates to CR or querying CR would cause a significant scalability problem for two reasons. First, the CoN information update is relatively low. A source Site would only change their IP address prefixes and AS number due to network migration or expansion. Second, since the CR can be implemented in a hierarchical manner similar to existing DNS system, the resolution of queries can be distributed. Moreover, the lookup process of transit CoN to the source Site does not involve CR at all.

4. ODMT Implementation

We implemented ODMT in a testbed to verify its design correctness and effectiveness in real deployment. Our development testbed consists of five Cisco 7200 as the border routers and five Cisco 3600 as the internal routers. In this section, we describe the ODMT implementation prototype.

4.1 Overview

The testbed topology is depicted in **Fig. 5** and it consisted of five Sites, represented by AS number 10 to 50. Each Site was configured to run PIM-SSM routing and IGMP version 3 following the configuration guideline outlined in Ref. [46]. Native inter-domain PIM peering was disabled while BGP unicast routing were running between border routers to emulate the isolated multicast environment between Sites.

Each CoN was a host running Linux (Fedora Linux 12 with kernel version 2.6.31) and the prototype of CoN was written using Python script. We modified "qpimd" [47], a fork of Quagga routing suite [48] to act as a pseudo PIM-SSM router at the CoN for the purpose of capturing PIM (S, G) joins. We assigned one host to act as the CR. The CR was a small Python script that read a small database consisting of rows of (*CoN IP address, IP Prefix, AS number*) tuple. For the verification of multicast packet flows, the ssmping tool [49] was installed at each source and receiver host.

CoN controls the multicast tunnel peering on a router via remote command invocations. Each router is represented by an object of FoN class that adheres to the interface depicted in **Fig. 6**. Since router commands are specific to the model of the router, the implementation of FoN class has to be customized for each different router model. Generic Routing Encapsulation (GRE) [50] was chosen as the tunneling protocol because it supports a wide variety of network layer protocol including multicast and is supported by most routers.

In our testbed, FoN is represented by the border router of each AS network. To avoid confusion with term FoN and border router, in this section, we use the term ASR with its index number to represent those two meanings in this testbed. Initially,

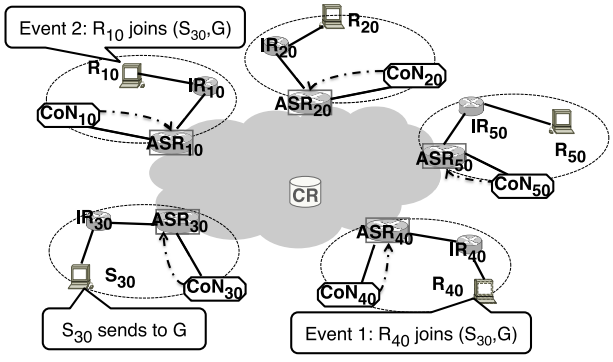

Fig. 5 ODMT test scenario. Each CoN is configured to accept only one multicast tunnel peering request.

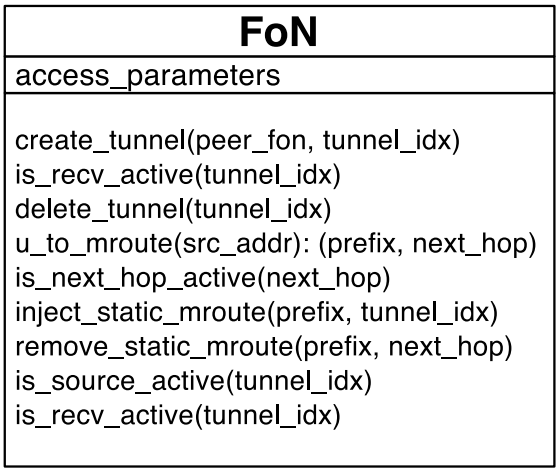

Fig. 6 Abstract interface of FoN (Router) class.

each ASR was configured with a default static multicast route that points to its CoN.

4.2 Test Scenario

We assigned R_{10} to R_{50} as the multicast receivers and S_{30} as the multicast source. We also defined two test scenarios represented by Event 1 and Event 2 as depicted in Fig. 5.

Event 1: a receiver R_{40} initiates a join request (S_{30}, G) to its upstream router IR_{40}. The PIM (S_{30}, G) join state will be created at IR_{40}. IR_{40} will forward PIM (S_{30}, G) join to ASR_{40}. Since ASR_{40} has a static default multicast route to CoN_{40}, the PIM (S_{30}, G) join will then be forwarded to CoN_{40}. The initial multicast join path is $R_{40} \rightarrow IR_{40} \rightarrow ASR_{40} \rightarrow CoN_{40}$.

At this stage, CoN_{40} will start to process the PIM (S_{30}, G) join according the steps described in Section 3.2. CoN_{40} will negotiate with CoN_{30} and the multicast tunnel peering will be established between ASR_{40} and ASR_{30}. Through a static multicast route injection at ASR_{40}, the multicast join path will be changed to $R_{40} \rightarrow IR_{40} \rightarrow ASR_{40} \rightarrow ASR_{30} \rightarrow IR_{30} \rightarrow S30$ and subsequently, R_{40} will receive multicast packets from S_{30}.

Event 2: a receiver R_{10} initiates a join request (S_{30}, G) to its upstream router IR_{10}. While Event 1 is still ongoing, Event 2 is executed. Like Event 1, similar initial processes will be performed in the local PIM domain within AS_{10} network. The initial multicast join path is $R_{10} \rightarrow IR_{10} \rightarrow ASR_{10} \rightarrow CoN_{10}$.

Assuming that the multicast tunnel peering between ASR_{40} and ASR_{30} is still active, thus CoN_{30} will not be able to serve the request from CoN_{10}, because CoN_{30} had been configured to serve only one tunnel peering at a time. Instead, the steps described in Section 3.5 will be used. In brief, CoN_{30} will redirect to CoN_{10} to CoN_{40} and the multicast tunnel peering will be established between ASR_{10} and ASR_{40}. By injecting a static multicast route at ASR_{10}, the multicast join path will be changed to $R_{10} \rightarrow IR_{10} \rightarrow ASR_{10} \rightarrow ASR_{40} \rightarrow ASR_{30} \rightarrow IR_{30} \rightarrow S30$ and subsequently, R_{10} will receive multicast packets from S_{30}.

4.3 Implementation Features

Monitoring of Tunnels and Injected Routes: The tunnel peering and static multicast route injection are performed to enable PIM message delivery. Therefore, to determine when a tunnel and its corresponding static multicast route need to be deleted, we adopted recommendations from PIM-SSM specification [24]. Adopting from PIM join expiry timer, we used 210 seconds for the tunnel expiry timer. A tunnel will be deleted if it is not associated with any multicast state or if there is no flowing multicast traffic after the expiry timer. This tunnel deletion will automatically delete the static multicast routes associated with the tunnel interfaces. In addition, our prototype polls the FoN every 60 seconds to monitor the static multicast routes injection that corresponds to local multicast sources. If there is a change in the unicast routing table for the next hop router of the local multicast sources, the injected static multicast routes will be updated accordingly.

Multiple Channels on a Single Tunnel: Each record on the CR contains an IP prefix with its associated CoN, and several IP prefixes may be associated with a CoN. Let's assume that two PIM joins, (S_1, G_1) and (S_2, G_2) arrive consecutively at a receiver's CoN. The receiver's CoN queries the CR and receives two replies with different IP prefixes and the same CoN IP address (source's CoN). The receiver's CoN establishes a tunnel and injects a static multicast route for (S_1, G_1) first. Before establishing a tunnel for (S_2, G_2) it notes that there is already a tunnel associated with the source's CoN, hence it only injects a static multicast route for S_2 via the existing tunnel without establishing a new tunnel.

Bi-directional Multicast Tunnel: The static multicast route injection that creates multicast join path from the receiver Site to the source Site is only unidirectional. ODMT can handle a bi-directional multicast tunnel easily since it keeps every multicast tunnel peering state. When the CoN at the source Site receives PIM (S_R, G) join, then the same procedure of contacting CR is performed. As soon as the CoN at the source Site receives a reply from CR, it will check its multicast tunnel peering states. If there is an active multicast tunnel peering between the source Site and the receiver Site, multicast tunnel peering negotiation will be skipped. Instead, the source Site only performs multicast route injection to the receiver Site. This will divert PIM (S_R, G) join to the CoN at the receiver Site. At this point, the decision to allow or block this bi-directional multicast flow is determined by the CoN at the receiver Site. This mechanism is fast and efficient because the existing tunnel can be reused for bi-directional multicast flow.

ODMT introduced minimal changes to the networks and at least one CoN is needed for one multicast island to enable interconnection between multicast islands. ODMT eases the management of inter-domain multicast peering.

5. Deployment Issues

This section addresses the possible issues when deploying ODMT.

5.1 Co-existence with native inter-domain IP multicast

ODMT is designed to lessen the barrier of inter-domain multicast deployment, but it is not meant to hamper native inter-domain multicast deployment. ODMT can co-exist with native inter-domain IP multicast that uses PIM-SM and MBGP. ODMT modified the multicast routes through static routing mechanism that only affects its FoN (router), because the injected routes are

not distributed to other routers.

Suppose that at the same router (FoN), MBGP is used for distributing the route reachability for unicast and multicast between two adjacent AS neighbors, the CoN at both ASes will never be triggered for multicast tunnel peering whenever a multicast source and receivers from both ASes are communicating. This behavior is somewhat expected, because there is no point in using multicast tunneling if both ASes are physically adjacent and ODMT should not override the native inter-domain IP multicast.

There may a desirable behavior to make CoN to override the multicast routes from MBGP whenever the source multicast is several hops away (crossing several ASes) from receivers. One way to deal with it is by defining a list of routes that can be overridden by CoN. CoN will inject static multicast routes based on that list. The issue with this deployment model is whenever there is a change to MBGP multicast policy and some routes needs not to be overridden, the list must be updated manually.

5.2 Placement of FoN

ODMT design recommends a FoN to be a border router of an AS, with a CoN located on the same link. However in a deployment, a FoN may be an internal router of an AS. If such a deployment is needed, all routers in an AS must be configured to have a default static multicast route toward the FoN. This configuration is required so that all PIM-SSM joins to the sources outside of an AS will always be forwarded to the FoN. Subsequently, the PIM-SSM joins will trigger the establishment of multicast tunnel peering at the CoN.

5.3 Single CoN with Multiple FoNs

The size of a network varies from one AS to another. For example, a Tier-1 ISP or Large ISP can have many border routers and can advertise many IP prefixes. There may be a need to operate several FoNs with a single CoN. The CoN and its FoNs must be connected to each other via manual multicast tunnels. The advantage of this deployment model is CoN has complete control over its FoNs, so the coordination between FoNs can be easily managed. For example, if there is PIM (S_1, G) join received from one FoN, while another FoN has the tunnel state, instead of initiating another multicast tunnel peering with the source Site, CoN can directly create multicast tunnel peering between its FoNs. The issues of this deployment model are that a single CoN may not scale to handle monitoring and control of many FoNs through remote commands invocation.

5.4 Multiple CoNs and Multiple FoNs

This deployment model may emerge from the same motivation as in the case of Single CoN with Multiple FoN, where the size of the network is considerably large and there are many border routers. This deployment model can gain the same advantages as above, but it scales better. However, it poses a different issue on the synchronization between CoN, because basically each CoN is independent of each other. There is a possibility that several CoNs may perform multicast tunnel peering to the same source network. This inefficiency can be alleviated by configuring each CoN within an AS to prioritize its local CoNs over other foreign CoN.

6. Deployment Scenario Analysis

CoN is the central component in determining the deployment scenario of ODMT. The CoN at source AS (S) provides control based on the input of local policy whether to directly accept a multicast tunneling request from a receiver AS (R) or to offload it to other sites which provide multicast transit tunnel on behalf of the source AS. To analyze how and when ODMT could overcome the barrier in inter-domain multicast deployment, we use several possible deployment scenarios in **Fig. 7**. Reduced traffic or bandwidth savings due to multicast does not always translate into benefits for an ISP. We evaluate each scenario using three parameters: 1) the number of routers that maintain multicast forwarding states, 2) elimination of redundant traffic due to the number of redundant tunnels, and 3) increased traffic opportunity by the number of tunnels.

Figure 7 a) shows inter-domain native multicast deployment scenario, in which IP multicast introduces bandwidth savings for all parties with the consequence of maintaining multicast forwarding states on all ASes. For transit ISPs, this bandwidth saving does not provide benefit due to the reduced traffic volume from their customers (*content* and *eyeball* ISPs). With this loss, transit ISPs will be better off without multicast routing, this causing the isolation of multicast among ASes. Multicast deployment scenario using tunneling to interconnect directly between source AS (S) and receiver ASes (R), depicted in Fig. 7 b), could be regarded as a means to avoid the multicast isolation problem. Figure 7 c) describes a scenario, where source AS (S) offloads tunnel requests to its upstream *content* ISP based on a prior agreement to provide multicast transit tunnel. Figure 7 d) shows a scenario, where source AS (S) has multicast transit tunnel agreement with transit ISPs. The deployment scenario in Fig. 7 e) shows that transit ISPs are bypassed by the multicast transit tunnel agreement between the source AS and the *eyeball* ISPs and in Fig. 7 f) shows a deployment scenario where the source AS has transit tunnel agreement with various ISPs (*content*, transit, and *eyeball* ISPs).

The evaluation on each scenario is presented in **Table 2**, where each type of ISP, namely *content*, transit, and *eyeball* ISPs are being evaluated according to the three parameters above. In summary, 1) the *content* ISP: increased traffic opportunity is most preferable followed with low elimination of redundancy and low number of routers with multicast states, such as scenario in *b* and *c*. 2) the transit ISP: similar to *content* ISP, such as scenario in *b*, *c*, and *d*. 3) the *eyeball* ISP: elimination of redundancy is most desirable followed by low number of routers with multicast states, such as scenario in *a*, *e*, and *f*. There is no concern of increased traffic opportunity for the *eyeball* ISP, since their focus is on the *downhill* path traffic where the reduce of inbound traffic would be beneficial for them.

7. Related Work

The efforts of providing multicast reachability in the Internet is a longstanding goal of the research community and the industry for more than two decades. ODMT drawn on many existing ideas.

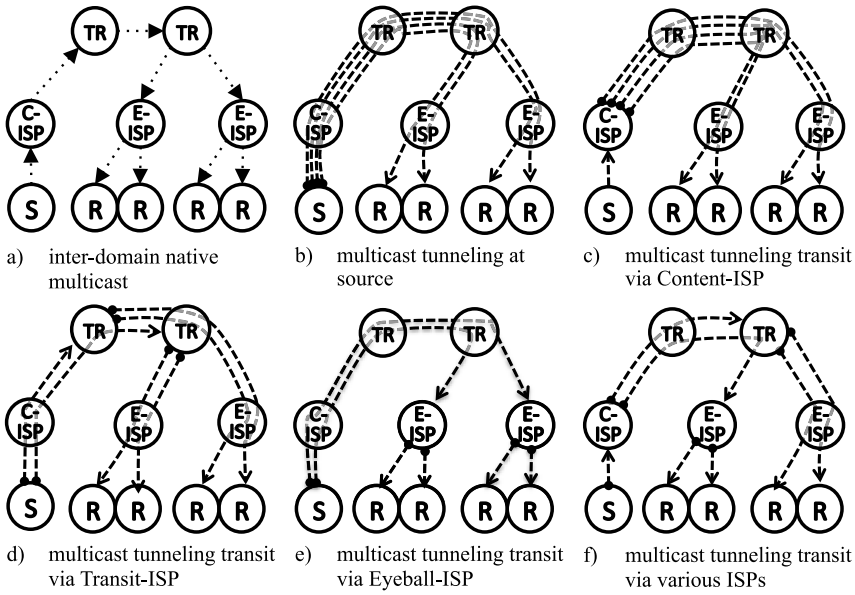

Fig. 7 Native multicast and ODMT deployment scenarios. The inter-domain peering relationships between circle ASes in these scenarios follows the defined relationships in Fig. 1. a) A dotted line with solid-end arrow describes native multicast path hop by hop over circle ASes. For b) to f), a dashed line with open-end arrow describes multicast tunneling between circle ASes and the solid-start arrow for each dashed line indicates the starting point of tunnel, which could be directly from S (source AS), or multicast transit tunnel via C-ISP (*content* AS), TR (transit AS) or E-ISP (*eyeball* AS).

Table 3 Multicast service models comparison.

	ASM	SSM	AMT [19]	ODMT
Routing tree type	Any	Per-source, unidirectional only	Any	Per-source prefix, bidirectional
Sender authorization	None	None (single source)	None	via CoN
Receiver authorization	None	None (hook provided)	None	via CoN
Protocol to manage inter-domain core	PIM, MBGP, MSDP (IPv4)	PIM-SSM	Relay nodes	CoN and CR
Modification to packet formats	No	No. (IGMPv3 - IPv4, MLDv2 - IPv6)	No	same as SSM

Table 2 Multicast deployment scenarios evaluation. Here, "good" indicates providing positive incentive according to each parameter evaluation on each type of ISP.

	Content ISP	Transit ISP	Eyeball ISP
number of routers with multicast states	high is bad high: a,c medium: f low: b,d,e	high is bad high: a,d medium: f low: b,c,e	high is bad high: a,e medium: f low: b,c,d
elimination of redundant tunnels	high is bad high: a medium: d,e,f low: b,c	high is bad high: a medium: e,f low: b,c,d	high is good high: a,e medium: f low: b,c,d
increased traffic opportunity	high is good high: b,c medium: d,e low: a,f	high is good high: b,c medium: d,f low: a,e	Not applicable *eyeball* ISP does not have incentive in this parameter

Reduction of multicast states using tunnels: Tunneling solution has long been recognized as a possible solution to reduce the multicast states [18], [19], [39], [40]. However, all of these approaches are unaware of high fanout tunnel end-points and redundant tunnel link problem. Our design overcomes this by proposing the concept of networked-overlay multicast transit to reduce the high fanout and redundant link at the source Site.

Reduction of protocol complexities: The complexity of multicast routing protocol lies in the way of learning the active sources. A number of new proposals solved that problem [16], [51], [52] by decoupling the multicast group membership discovery and multicast forwarding lookup. From this point of view, our design does not propose a new multicast forwarding mechanism, instead we focused on the control-plane to make the multicast states creation and the forwarding more controllable.

Alternative service model: Rajahalme in Ref. [32] had explored the multicast tussle on Internet and proposed an *incentive-informed* inter-domain multicast model that derived incentives from the local unicast route policy. To some extent, our approach is in-line with their operational direction regarding the deployable multicast solution and we came out with a concrete solution. However, ODMT introduces a networked-overlay multicast transit mechanism with regard to the input of local policy.

Table 3 shows the comparison of multicast service model, extending the analysis by Ref. [11]. ODMT provides a measure of control in the establishment of multicast tunnel peering through the configuration of local policy. The inter-domain connection is manageable through the arrangement of tunnel negotiation protocol among CoNs at each site and the coordination with CR.

8. Conclusions

ODMT solves the problem of IP multicast tussle among content providers, ISPs, and users by providing a manageable inter-

domain service model, where each ISP can decide whether to provide multicast transit tunnel on behalf of the source Site by evaluating their own incentives. ODMT provides an on-demand inter-provider multicast tunneling that is manageable via the local-policy of each participating site. Our approach lowers the barrier of inter-provider multicast interconnection and reduces multicast operational complexity.

Reference

[1] Labovitz, C.: World Cup versus the Internet, Arbor Networks (online), available from ⟨http://asert.arbornetworks.com/2010/06/world-cup-versus-the-internet/⟩ (accessed 2010-12-22).

[2] Merkel, K.: HbbTV – A hybrid broadcast-broadband system for the living room, EBU (online), available from ⟨http://tech.ebu.ch/docs/techreview/trev_2010-Q1_HbbTV.pdf⟩ (accessed 2010-12-17).

[3] Google TV: TV meets web, Web meets TV (online), available from ⟨http://www.google.com/tv/⟩ (accessed 2010-06-17).

[4] YouView TV: YouView will change the way you watch TV forever (online), available from ⟨http://www.youview.com/⟩ (accessed 2011-01-06).

[5] Cisco Systems: Cisco Visual Networking Index: Forecast and Methodology, 2009–2014 (online), available from ⟨http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf⟩ (accessed 2010-06-01).

[6] Labovitz, C., Iekel-Johnson, S., McPherson, D., Oberheide, J. and Jahanian, F.: Internet Inter-Domain Traffic, *Proc. ACM SIGCOMM 2010*, pp.75–86 (online), DOI: http://doi.acm.org/10.1145/1851182.1851194 (2010).

[7] Deering, S.E.: Multicast routing in internetworks and extended LANs, *Proc. ACM SIGCOMM 1988*, pp.55–64, ACM (online), DOI: http://doi.acm.org/10.1145/52324.52331 (1988).

[8] Cha, M., Choudhury, G., Yates, J., Shaikh, A. and Moon, S.: Case study: Resilient backbone network design for IPTV services, *WWW IPTV Workshop* (2006).

[9] Karpilovsky, E., Breslau, L., Gerber, A. and Sen, S.: Multicast redux: A first look at enterprise multicast traffic, *WREN '09*, pp.55–64, ACM (online), DOI: http://doi.acm.org/10.1145/1592681.1592691 (2009).

[10] Jin, X., Tu, W. and Chan, S.-H.G.: Challenges and advances in using IP multicast for overlay data delivery, *IEEE Communications Magazine*, Vol.47, No.6, pp.157–163 (online), DOI: 10.1109/MCOM.2009.5116814 (2009).

[11] Diot, C., Levine, B.N., Lyles, B., Kassem, H. and Balensiefen, D.: Deployment issues for the IP multicast service and architecture, *IEEE Network*, Vol.14, No.1, pp.78–88 (online), DOI: 10.1109/65.819174 (2000).

[12] Sharma, P., Perry, E. and Malpani, R.: IP multicast operational network management: Design, challenges, and experiences, *IEEE Network*, Vol.17, No.2, pp.49–55 (online), DOI: 10.1109/MNET.2003.1188287 (2003).

[13] Rajvaidya, P. and Almeroth, K.C.: Multicast routing instabilities, *IEEE Internet Computing*, Vol.8, No.5, pp.42–49 (online), DOI: 10.1109/MIC.2004.48 (2004).

[14] Sarac, K. and Almeroth, K.C.: Monitoring IP multicast in the Internet: Recent advances and ongoing challenges, *IEEE Communication Magazine*, Vol.43, No.10, pp.85–91 (online), DOI: 10.1109/MCOM.2005.1522129 (2005).

[15] Asaeda, H., Suzuki, S., Kobayashi, K. and Murai, J.: Architecture for IP Multicast Deployment: Challenges and Practice, *IEICE Trans. Communications*, Vol.89, No.4, p.1044 (2006).

[16] Ratnasamy, S., Ermolinskiy, A. and Shenker, S.: Revisiting IP multicast, *Proc. ACM SIGCOMM 2006*, pp.15–26, ACM, New York, NY, USA (online), DOI: http://doi.acm.org/10.1145/1159913.1159917 (2006).

[17] Stoica, I., Ng, T.S.E. and Zhang, H.: REUNITE: A recursive unicast approach to multicast, *Proc. IEEE INFOCOM 2000*, Vol.3, pp.1644–1653 (online), DOI: 10.1109/INFCOM.2000.832563 (2000).

[18] Costa, L.H.M.K., Fdida, S. and Duarte, O.: Hop by hop multicast routing protocol, *Proc. ACM SIGCOMM 2001*, pp.249–259 (online), DOI: http://doi.acm.org/10.1145/383059.383079 (2001).

[19] Thaler, D., Talwar, M., Aggarwal, A., Vicisano, L. and Pusateri, T.: Automatic IP Multicast Without Explicit Tunnels (AMT), Internet Draft (2010). draft-ietf-mboned-auto-multicast-10.txt.

[20] Thyagarajan, A., Casner, S. and Deering, S.: Making the Mbone real, *Proc. INET, Honolulu*, pp.465–473 (1995).

[21] Waitzman, D., Partridge, C. and Deering, S.: Distance Vector Multicast Routing Protocol, RFC 1075 (Experimental) (1988).

[22] Deering, S.: Host extensions for IP multicasting, RFC 1112 (Standard) (1989). Updated by RFC 2236.

[23] Savola, P.: Overview of the Internet Multicast Routing Architecture, RFC 5110 (Informational) (2008).

[24] Fenner, B., Handley, M., Holbrook, H. and Kouvelas, I.: Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised), RFC 4601 (Proposed Standard) (2006). Updated by RFCs 5059, 5796.

[25] Deering, S., Estrin, D.L., Farinacci, D., Jacobson, V., Liu, C.-G. and Wei, L.: The PIM architecture for wide-area multicast routing, *IEEE/ACM Trans. Netw.*, Vol.4, pp.153–162 (online), DOI: http://dx.doi.org/10.1109/90.490743 (1996).

[26] Bates, T., Chandra, R., Katz, D. and Rekhter, Y.: Multiprotocol Extensions for BGP-4, RFC 4760 (Draft Standard) (2007).

[27] Fenner, B. and Meyer, D.: Multicast Source Discovery Protocol (MSDP), RFC 3618 (Experimental) (2003).

[28] Holbrook, H. and Cain, B.: Source-Specific Multicast for IP, RFC 4607 (Proposed Standard) (2006).

[29] Gao, L.: On inferring autonomous system relationships in the Internet, *IEEE/ACM Trans. on Networking*, Vol.9, No.6, pp.733–745 (online), DOI: 10.1109/90.974527 (2001).

[30] Faratin, P., Clark, D., Bauer, S., Lehr, W., Gilmore, P. and Berger, A.: The Growing Complexity of Internet Interconnection, *Communications & Strategies*, Vol.1, No.72, pp.51–72 (online), available from ⟨http://ideas.repec.org/a/idt/journl/cs7203.html⟩ (2008).

[31] Kosiur, D.: *IP Multicasting: The complete guide to interactive corporate networks*, John Wiley & Sons, Inc. New York, NY, USA (1998).

[32] Rajahalme, J.: Incentive-Informed Inter-Domain Multicast, *Proc. INFOCOM IEEE Conf. Computer Communications Workshops*, pp.1–6 (online), DOI: 10.1109/INFCOMW.2010.5466671 (2010).

[33] Zhang, B. and Mouftah, H.T.: Forwarding state scalability for multicast provisioning in IP networks, *IEEE Communications Magazine*, Vol.41, No.6, pp.46–51 (online), DOI: 10.1109/MCOM.2003.1204747 (2003).

[34] Francois, P. and Bonaventure, O.: An evaluation of IP-based fast reroute techniques, *CoNEXT '05*, pp.244–245, ACM, New York, NY, USA (online), DOI: http://doi.acm.org/10.1145/1095921.1095962 (2005).

[35] Cisco Systems: Cisco 7600 Series Route Switch Processor 720 (online), available from ⟨http://www.cisco.com/en/US/prod/collateral/routers/ps368/product_data_sheet0900aecd8057f3b6.html⟩ (accessed 2010-01-01).

[36] Juniper Networks: Tunnel Service PIC (online), available from ⟨http://www.juniper.net/us/en/local/pdf/datasheets/1000092-en.pdf⟩ (accessed 2011-08-10).

[37] Iperf: Measurement tool for TCP/UDP bandwidth performance (online), available from ⟨http://sourceforge.net/projects/iperf/⟩ (accessed 2011-08-10).

[38] Beverly, R. and Claffy, K.C.: Wide-area IP multicast traffic characterization, *IEEE Network*, Vol.17, No.1, pp.8–15 (online), DOI: 10.1109/MNET.2003.1174172 (2003).

[39] Tian, J. and Neufeld, G.: Forwarding state reduction for sparse mode multicast communication, *Proc. IEEE INFOCOM 1998*, Vol.2, pp.711–719 (online), DOI: 10.1109/INFCOM.1998.665093 (1998).

[40] Blazević, L. and Le Boudec, J.-Y.: Distributed core multicast (DCM): A multicast routing protocol for many groups with few receivers, *SIGCOMM Comput. Commun. Rev.*, Vol.29, No.5, pp.6–21 (online), DOI: http://doi.acm.org/10.1145/505696.505698 (1999).

[41] Inoue, T. and Kurebayashi, R.: An Analysis of Tunneling Impact on Multicast Efficiency, *IEICE Trans. on Communications*, Vol.E89-B, No.1, pp.38–46 (online), DOI: http://dx.doi.org/10.1093/ietcom/e89-b.1.38 (2006).

[42] Mockapetris, P. and Dunlap, K.J.: Development of the domain name system, *SIGCOMM Comput. Commun. Rev.*, Vol.18, pp.123–133 (online), DOI: http://doi.acm.org/10.1145/52325.52338 (1988).

[43] Cox, R., Muthitacharoen, A. and Morris, R.: Serving DNS Using a Peer-to-Peer Lookup Service, *Peer-to-Peer Systems*, Druschel, P., Kaashoek, F. and Rowstron, A. (Eds.), Lecture Notes in Computer Science, Vol.2429, Springer Berlin/Heidelberg, pp.155–165 (2002).

[44] Vixie, P., Thomson, S., Rekhter, Y. and Bound, J.: Dynamic Updates in the Domain Name System (DNS UPDATE), RFC 2136 (Proposed Standard) (1997). Updated by RFCs 3007, 4035, 4033, 4034.

[45] Wellington, B.: Secure Domain Name System (DNS) Dynamic Update, RFC 3007 (Proposed Standard) (2000). Updated by RFCs 4033, 4034, 4035.

[46] Cisco Systems: Cisco IOS IP Configuration Guide, Release 12.2: Configuring Source Specific Multicast (online), available from ⟨http://www.cisco.com/en/US/docs/ios/12_2/ip/configuration/guide/1cfssm.html⟩ (accessed 2010-08-01).

[47] da Silva Marques, E.: qpimd - PIM Daemon for Quagga (online), available from ⟨http://www.nongnu.org/qpimd/⟩ (accessed 2010-08-

01).
[48] Quagga: Routing Software Suite (online), available from ⟨http://quagga.net/⟩ (accessed 2010-11-15).
[49] Veenas, S.: SSMPING Tool (online), available from ⟨http://www.venaas.no/multicast/ssmping/⟩ (accessed 2010-09-01).
[50] Farinacci, D., Li, T., Hanks, S., Meyer, D. and Traina, P.: Generic Routing Encapsulation (GRE), RFC 2784 (Proposed Standard) (2000). Updated by RFC 2890.
[51] Cho, T.W., Rabinovich, M., Ramakrishnan, K.K., Srivastava, D. and Zhang, Y.: Enabling Content Dissemination Using Efficient and Scalable Multicast, *Proc. IEEE INFOCOM 2009*, pp.1980–1988 (online), DOI: 10.1109/INFCOM.2009.5062120 (2009).
[52] Tian, X., Cheng, Y. and Shen, X.: DOM: A scalable multicast protocol for next-generation internet, *IEEE Network*, Vol.24, No.4, pp.45–51 (online), DOI: 10.1109/MNET.2010.5510918 (2010).

Achmad Basuki received his B.E. from Brawijaya University, Indonesia in 2000 and has been joined as a full-time lecturer in computer science at Faculty of Mathematics and Natural Science from 2003 to 2006. He received master's degree in 2008 from Graduate School of Media and Governance Keio University. Japan and is now a Ph.D. candidate at the same university. His current research interests include multicast and its deployment issues.

Achmad Husni Thamrin is Assistant Professor at Keio University. He is a graduate of Keio University, Graduate School of Media and Governance (Ph.D. 2005, M.M.G., 2002). His research interests include multicast, Internet over broadcast media, and peer-to-peer networks.

Hitoshi Asaeda is Associate Professor of Graduate School of Media and Governance, Keio University. He received his Ph.D. in Media and Governance from Keio University in 2006. From 1991 to 2001, he was with IBM Japan, Ltd. From 2001 to 2004, he was a research engineer specialist at INRIA Sophia Antipolis, France. His research interests are IP multicast routing architecture and its deployment, dynamic networks and streaming applications. He is a member of ACM, IEEE, IPSJ, and WIDE Project.

Jun Murai is Professor of Faculty of Environment and Information Studies, Keio University. He received his M.E. and Ph.D. in Computer Science from Keio University in 1981 and 1987 respectively. He was a director of Keio Research Institute at SFC, the president of Japan Network Information Center (JP-NIC), board director of ICANN. Adjunct Professor at Institute of Advance Studies, United Nation University. He also teaches computer network and computer communication.