# PIM-Compliant SDN-Enabled IP Multicast Service

Li-Hsing Yen, Ming-Hung Wang, Song-Yu Wu, and Chien-Chao Tseng
Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan 300, R.O.C.
Email: {mhwang408,exz920199}@gmail.com, {lhyen,cctseng}@cs.nctu.edu.tw

***Abstract*—Software-defined networking (SDN) has been a promising solution to multicast streaming data due to its scalability and manageability. However, offering a multicast service that spans a large geographical area is still challenging because we still lack a unified multicast scheme that interconnects independent SDN-managed networks and bridges the service between SDN-based networks and the rest of the Internet. This paper proposes a solution that integrates SDN, CORD, and PIM technologies. We provides preliminary performance evaluation results. This work serves as a stepping stone to the ultimate goal of multicast as a service.**

## I. Introduction

Multicast streaming provides a one-to-many content delivery service that guarantees satisfactory quality of service for streaming data. While the demand for multicast streaming constantly increases, the demand for bandwidth also increases. It is a challenge how to maintain expected quality of service with limited bandwidth capacity.

Software-defined networking (SDN) is a centralized network control and management technology which provies programmability and flexibility. It can facilitate developing an efficient content delivery service that maintains forwarding paths in a dynamic and responsive way for high bandwidth utilization, and thus becomes a promising solution to multicast streaming. However, offering a multicast service that spans a large geographical area is still challenging due to the following reasons. First, the networking area under the control of a single SDN controller (called an SDN *domain*) is physically limited. A large-scale network is usually divided into several SDN domains for management or policy consideration. Thus, a mechanism that connects SDN domains in a large area to provide a unified multicast service will be appealing. Second, not all network domains are SDN-ready. To offer a multicast service that spans both SDN and non-SDN network domains, we need a backward-compatible mechanism to cooperate with non-SDN network domains.

This paper takes the campus network in National Chiao Tung University (NCTU) as an example to address the above-mentioned issues. There are two main campuses (Kuang-Fu and Boai) at NCTU in Hsinchu City. Each campus network has several SDN domains. Depending on the relative locations of multicast source and client (i.e., a receiver), there are four possible types of multicast: 1) Intra-domain multicast, where the source and the client are in the same SDN domain. 2) Inter-domain (intra-campus) multicast, where the source and the client are in the same campus network but in different SDN domains. 3) Inter-campus multicast, where the source and the client are in different campus networks. 4) Internet multicast, where the source is outside the campus network.

We propose a hierarchical architecture that integrates approaches of different layers to support multicasts of all types. For intra-domain multicast, we use SDN-based [1] and traditional L2 multicast approaches for SDN and non-SDN network domains, respectively. For inter-domain and inter-campus multicasts, we additionally use leaf-spine switching architecture [2] together with virtual router (vRouter) which has been used in Central Office Re-Architected as a Data Center (CORD) [3]. For Internet multicast, We use Protocol Independent Multicast (PIM) [4], [5] to provide backward compatibility with traditional IP networks. The result is a unified PIM-compliant SDN-enabled IP Multicast scheme (or PSM for short). We conducted emulations for performance evaluations. The results indicate that the time to process a join request is only a few tens of milliseconds while the response time of receiving the first multicast data is less than three seconds regardless of the type of multicast.

The rest of this paper is organized as follows: Sec. II provides background information and reviews related approaches. Sec. III presents the details of the proposed approaches. Sec. IV shows experimental results. The last section concludes this work.

## II. Related work and Background

Multicast within a local area network (LAN) could be implemented by a broadcast across the whole LAN. However, such a broadcast would waste too much bandwidth. Layer-2 multicast protocols, such as IGMP Snooping [6] and Cisco Group Management Protocol (CGMP) [7], reduce bandwidth consumption by utilizing a logical spanning tree in the LAN. The problem with layer-2 multicast protocols is low link utilization: multicast traffic can only pass through links that are parts of the tree; links not in the tree are not usable.

An SDN controller has the visibility over the whole network topology as well as traffic condition of each physical link. With this ability, the controller can utilize all physical links for layer-2 multicast without creating forwarding loops. Existing SDN-based multicast approaches aim at creating shortest-path trees that minimize delivery delays [8], minimize the time to setup flow entries in SDN switches [9], distribute multicast traffic [10], or minimize packet loss when switching

This work is sponsored in part by Ministry of Science and Technology, Taiwan, under grant numbers 106-2622-8-009-017 and 106-2221-E-009-030-MY3.

978-1-5386-3416-5/18/$31.00 © 2018 IEEE

a multicast tree to another for failure recovery [11]. Bandwidth Aware Multicast Service (BAMS) [1] dynamically maintains multicast trees to exploit the software-definable nature of SDN. It supports load balancing by avoiding congested links and supports multiple multicast sources. However, BAMS considers only a single SDN network domain.

CORD attempts re-architecting traditional telecom central office as a data center to enable the use of commodity building blocks (commodity servers and white-box switches) and to provide the ability to rapidly deploy and elastically scale services [3]. CORD uses a two-tier leaf-spine switching fabric [2] in the data center, where servers and open line terminations (OLTs) are connected to *leaf switches* which are then interconnected by *spine switches*. To connect the CORD-based network to other IP-based networks, we may use virtual router in CORD to interwork with outside IP routers.

PIM is an IP-based multicast protocol which uses routes maintained by an independent unicast routing protocol such as OSPF, RIP, and BGP to build multicast trees. PIM supports *dense* [4] and *sparse* modes [5]. In the dense mode, PIM uses source-based multicast trees, which suits the case when there are few multicast sources or group members are densely distributed. In the PIM spare mode (termed PIM-SM), group members are assumed sparsely distributed such that a *PIM domain* spans multiple geographically separated network domains. PIM constructs a shared multicast tree for each group, where *Rendezvous Point* (RP) is the root node and *Designated Routers* (DRs) are non-root nodes in the multicast tree. Both RP and DRs are IP routers. When a host wants to join a multicast group (to receive multicast data), the router where the host attaches becomes the host's DR and sends a join message toward the RP of the group on behalf of the host. Similarly, when a host wants to send multicast data to a group, the DR where the host attaches registers as a *source* at the RP on behalf of the sending host. All multicast traffic flows through the RP down to every group members (receivers) via each member's DR. When the traffic load between a sender's and a receiver's DRs reaches a threshold, PIM-SM creates a direct shortest path between these two DRs. Consequently, a multicast source with a heavy bandwidth demand can gradually shift its traffic load from the original shared tree to a source-based tree.

Multicast Source Discovery Protocol (MSDP) [12] is to discover multicast sources distributed over multiple PIM domains. MSDP is executed by RPs. A RP detects and keeps track of any new multicast source within its PIM domain and exchange source information with RPs in other PIM domains. When a RP detects a new group join request issued by a client within its PIM domain, the RP sends Source Specific Join message to all the DRs with which sources of the multicast group attach. In this way, clients can receive multicast from sources in other PIM domains.

PIM source-specific mode (PIM-SSM) [13] is a multicast protocol where a multicast receiver can directly join to a particular source without the help of RP. Upon joining a group, the receiver should specify both the group and the IP address of the source (more specifically, the DR of the source). The source discovery here could be done in an out-of-band manner (e.g., through a web page). Alternatively, sources could proactively announce their active sessions using Session Announcement Protocol [14]. Because PIM-SSM creates a shortest-path tree for each source, the resulting multicast forwarding paths are typically shorter than those in a shared tree.

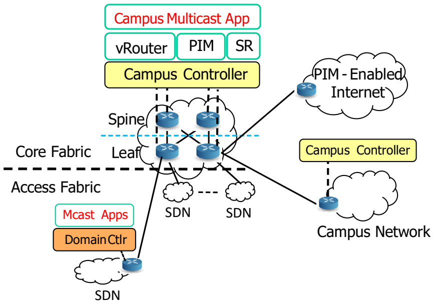


Fig. 1. The architecture of NCTU campus network

## III. The Proposed Approach

### A. Hardware and Software Architecture

Fig. 1 shows the hardware and software architecture for our design in NCTU campus network. Traffic within the campus network is handled by *access fabrics* and *core fabrics*. An *access fabric* refers to an SDN controller (named *domain controller*) together with all switches under its control and management in an SDN domain. A *core fabric* connects all SDN domains inside a campus. Our core fabric is embodied by CORD switching fabric and controlled and managed by a *campus controller*. The campus controller in the main campus also connects the whole campus network to the Internet.

PSM software consists of two parts, one running on campus controllers while the other on domain controllers. In a campus controller, we reuse existing CORD Apps to manage and control the core fabric. These Apps include *segment routing* (SR) App for fabric management, vRouter App for Internet connectivity, and PIM App for PIM-SSM management.

We additionally design Campus Multicast App running on every campus controller for the management of multicast in the whole campus network. We also designe the following Multicast Apps to be run on every domain controller: 1) Domain Multicast Administrator, which manages all multicast-related events. 2) Source Listener, which detects active multicast sources and reports any instance to the campus controller. 3) IGMP Speaker, which takes care of IGMP Join/Leave messages issued by clients. 4) Casting Manager, which dynamically maintains the multicast tree inside the domain and manages the bandwidth of the multicast tree. It also installs OpenFlow forwarding rules to relevant switches.

### B. Source Detection and Discovery

Since multicast sources could be anywhere in the network, we have to detect the locations of the sources to deliver

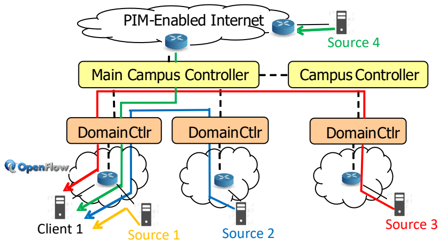


Fig. 2. Four multicast scenarios

multicast data. When a source within an SDN domain sends out the first multicast packet toward a group, the switch serving the source intercepts and forwards the packet to the domain controller using an OpenFlow Packet In message. This message notifies Source Listener of a new source. In this way, the Multicast Apps know the set of all sources within the SDN domain, enabling the handling of intra-domain multicast. The Source Listener also informs all other controllers of this new source to support inter-domain and inter-campus multicasts.

When the source is located at some network in the Internet that supports PIM, we use PIM-SSM to perform multicast and use web-based out-of-band source discovery.

### C. Group Join/Leave Process

Suppose that Client 1 in Fig. 2 sends out an IGMP Join message to subscribe multicast data from a specific source, i.e., source-specific multicast. This message causes the directly attached OpenFlow switch to send a Packet In OpenFlow message to the domain controller. If the source resides in the same SDN domain (e.g., Source 1 in Fig. 2), the controller constructs or modifies an intra-domain multicast tree for this group and installs associated packet forwarding rules in relevant switches. We used BAMS for this part. If the source is not in the same SDN domain, the domain controller passes join information to the campus controller. The campus controller caches source active information from other domain or campus controllers, so it can tell whether the source is in another domain (but still in the same campus, like Source 2 in Fig. 2) or in another campus (like Source 3 in Fig. 2). The former is an intra-campus multicast while the latter is an inter-campus multicast. For an intra-campus multicast, the campus controller instructs involved domain controllers either to construct an inter-domain multicast tree or to create an inter-domain path that directly connects the source's and the client's domains. For an inter-campus multicast, the campus controller needs to notify another campus controller to construct either an inter-campus multicast tree or an inter-campus path that connects the source's and the client's domains.

If the source is outside the campus network (e.g., Source 4 in Fig. 2), the main campus controller sends PIM-SSM join message to the source in the Internet. However, the main campus controller sends out only the first PIM-SSM join request. Any subsequent join request toward the same group only creates a data delivery path from the client sending the request to the core fabric of the main campus controller. This approach reduces the amount of PIM-SSM join requests.

When a client leaves the group, controllers need to do some housekeeping like modifying or removing OpenFlow rules. If the leaving client is the last group member in the SDN domain but there is still some source in other domains, the domain controller additionally notifies its upstream campus controller of this using a Leave message. The campus controller then modifies switching rules in the core fabric to stop forwarding multicast messages of this group to the last client's SDN domain. If the client is not only the last group member in its SDN domain, but also the last one in its campus network, the campus controller additionally notifies the other campus controller of this for housekeeping. If the leaving client is the last group member in the whole campus network, the main campus controller sends a PIM Leave to all routers in the Internet with which MSDP peerings have been established.

### D. Data Delivery

Intra-domain multicast traffic within an access fabric is handled based on OpenFlow rules. For multicasts beyond a single access fabric, a tunnel is created to forward multicast traffic between an access fabric and a core fabric. This is to pass traffic through legacy (i.e., non-SDN) networks as well as to aggregate access network traffic so as to decrease the number of flows. Traffic forwarding inside a core fabric is done by SR, where leaf switches attach and detach Multi-Protocol Label Switching (MPLS) labels for every packet received while spine switches forward packets between leaf switches based on MPLS labels. Therefore, an inter-domain multicast packet will be encapsulated by the source's access fabric and tunneled to the core fabric. The leaf switch in the core fabric that receives the tunneled packet decapsulates the packet, attempts matching the packet to a multicast address, attaches the corresponding MPLS label, and forwards the packet to a spine switch. The spine switch performs a label lookup to forward the packet to all the leaf switches that connect to some access fabric where at least one client resides. The leaf switch then encapsulates the packet again and forwards it to all the access fabrics that serve clients. The packet will be decapsulated at access fabrics and therein forwarded to clients. Forwarding of inter-campus multicast is similar. The only difference is that there is a tunnel created between two core fabrics. For multicast from the Internet, packets all arrive at the core fabric managed by the main campus controller. Since the paths from this core fabric to all access fabrics where clients of the multicast group reside all have been created upon the client's join requests, the delivery of the multicast packet simply follows existing forwarding rules.

## IV. Performance Evaluation

We did not perform experiments on our campus network because the impact of background traffic cannot be easily accounted. We therefore conducted emulations to evaluate the performance of PSM. We used two physical hosts, one Core i5 3470 with 16 GB RAM and the other Core i7 870 with 14 GB

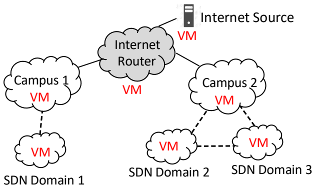


Fig. 3. Simulation environment

TABLE I
MEASURED LATENCY

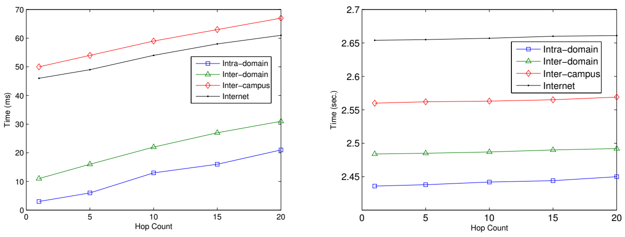
| Component | Latency |
|---|---|
| Controller-Controller | 2 ms |
| Controller-Switch | $< 1$ ms |
| Multicast Tree Calculation | 1 ms |
| OvS Tunnel Setup | 3 ms |
| Flow Rule Installation (per rule) | 1 ms |
| Source Location | 1 ms |
| Core Fabric Segment Routing Setup Per Path (if needed) | 45 ms |

RAM, to host seven virtual machines (VMs). In the Core i5 physical host, two VMs emulated campus core fabrics while another VM was used as an Internet router (running Quagga 0.99.23). In the Core i7 physical host, three VMs emulated access fabrics while another VM was a multicast source in the Internet. The simulated network topology is shown in Fig. 3.

We first measured the processing latency of a members join request. It counts from the time at which when a domain controller receives a join request to the time when all relevant flow rules for the join request have been installed. Table I lists the components of the whole latency and corresponding measured results.

We set up one source node and one client node and varied the number of hops in both of the source's and the client's SDN domains. We measured the latency of member join process for four different types of multicasts. Fig. 4(a) show the result. The hop count value is the number of switches between the source (resp. client) and the top-layer switch in the source's (resp. client's) access fabric. The latency generally increases as the hop count value increases. The reason is that the emulated controller did not actually install all flow rules to all involved switches at the same time. The controller has some internal scheduling rule to do the flow installations.

Internet and inter-campus multicasts both had higher latencies than the other two types of multicasts because it took significant time for these two types of multicasts to install flow rules on core fabrics. The inter-campus multicast had higher latencies than the Internet multicast because extra work was done in the source's access fabricfor for inter-campus multicast. In contrast, we ignored join delay caused by source in the Internet when measuring latencies for Internet multicast.



(a) (b)

Fig. 4. (a) Latencies of member join process for various types of multicasts (b) Multicast latencies for various types of multicasts

We also measured multicast latency, counting from the time when a client sends out an IGMP join message to the time when the client receives the first multicast data. Fig. 4(b) shows the results. Here the latencies are higher in Internet multicast than in inter-campus multicast. The reason is that the latency due to PIM join counts for Internet multicast. This part of latency is not significant though, because we used only a VM for the PIM router in the PIM-enabled Internet.

## V. CONCLUSIONS

This paper proposes PSM, an SDN-based multicast protocol with service coverage spanning multiple SDN domains. PSM combines SDN, CORD, and PIM. SDN technology helps dynamic bandwidth management and multicast tree construction. The CORD fabric architecture helps fast and flexible data switching among different SDN domains. The PIM protocol enables interworking with traditional IP networks. In the future, PSM could be integrated with CORD overlay part and become a part of CORD service.

## ACKNOWLEDGMENT

We thank Wei-Cheng Wang for his valuable suggestion on the design and implementation of the multicast protocols.

## REFERENCES

[1] M.-C. Chan *et al.*, "SDN supported multicast for live streaming service in campus/enterprise network," *IEEE Realiability*, in press.

[2] S. Das *et al.* (2016, Mar.) CORD fabric, overlay, virtualization, and service composition. Open Networking Lab. [Online]. Available: http://opencord.org/wp-content/uploads/2016/03/CORD-Fabric.pdf

[3] L. Peterson *et al.*, "Central office re-architected as a data center," *IEEE Communications Magazine*, no. 10, pp. 96–101, Oct. 2016.

[4] A. Adams *et al.*, "Protocol independent multicast - dense mode (PIM-DM): Protocol specification (revised)," RFC 3973, Jan. 2005.

[5] B. Fenner *et al.*, "Protocol independent multicast - sparse mode (PIM-SM): Protocol specification (revised)," RFC 4601, Aug. 2006.

[6] M. J. Christensen *et al.*, "IGMP and MLD snooping switches considerations," RFC 4541, May 2006.

[7] "IP Multicast Technology Overview." [Online]. Available: http://tinyurl.com/cisco-cgmp

[8] L. Bondan *et al.*, "Multiflow: Multicast clean-slate with anticipated route calculation on openflow programmable networks," *Journal of Applied Computing Research*, vol. 2, no. 2, pp. 68–74, Jul. 2012.

[9] Cesar A. C. Marcondes *et al.*, "Castflow: Clean-slate multicast approach using in-advance path processing in programmable network," in *Proc. IEEE Symposium on Computers and Communications*, 2012.

[10] W. Cui and C. Qian, "Dual-structure data center multicast using software defined networking," *CoRR*, 2014. [Online]. Available: http://arxiv.org/abs/1403.8065

[11] D. Kotani *et al.*, "A design and implementation of OpenFlow controller handling IP multicast with fast tree switching," in *Proc. IEEE Int'l Symp. on Applications and the Internet*, 2012.

[12] B. Fenner and D. Meyer, "Multicast source discovery protocol (MSDP)," RFC 3618, Oct. 2003.

[13] D. Meyer *et al.*, "Source-specific protocol independent multicast in 232/8," RFC 4608, Aug. 2006.

[14] M. Handley *et al.*, "Session announcement protocol," RFC 2974, Oct. 2000.