IP Multicast in Virtualized Data Centers: Challenges and Opportunities

Olufemi Komolafe
Brocade Communications
Caledonian Exchange
19A Canning Street, Edinburgh EH3 8EG, UK
Email: okomolaf@brocade.com

***Abstract*—The increasing volume and importance of point-to-multipoint traffic in virtualized data centers means the deployment of IP multicast is increasingly attractive. However, concerns about the ability of switches and routers based on commodity hardware to support the conventional IP multicast control plane and data plane, especially when there are thousands of participants in the multicast group communication, results in the infrequent deployment of IP multicast in data centers. This paper discusses the evolution of data center architectures towards the virtualized architectures in which technologies such as VXLAN, VXLAN-GPE, GENEVE, STT and NVGRE are used to build emulated Layer 2 networks that will support multi-tenancy at scale. These technologies are described and compared in terms of a number of factors, with emphasis laid on the manner in which they support multicast. Lastly, innovative approaches that have been proposed to circumvent the obstacles to deploying multicast in the data center IP fabric are also discussed and evaluated.**

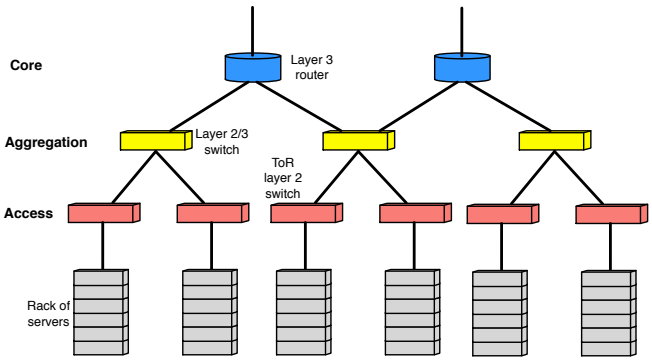

Fig. 1. Canonical data center topology

I. Introduction

In theory, data centers and IP multicast are a good match as traffic flows within data centers are often point-to-multipoint. Such traffic patterns may arise due to management requests to send identical updates to configure numerous servers, computational workloads that require the same data to be disseminated to different workers, monitoring queries sent to poll a large number of devices and the requirement to support multicast-centric applications such as IPTV and the streaming of market trading data. Furthermore, some of the networking protocols and technologies deployed in data centers ideally require multicast for neighbor discovery, adjacency maintenance and so on.

In practice, however, data center operators are typically reluctant to enable IP multicast, with scalability concerns arguably the most often cited deterrent. These concerns are exacerbated by the desire to use commodity hardware or simple switches within data centers. These switches may struggle to hold the multicast state (typical access switches have been found to support less than 1500 multicast forwarding entries [1]) or maintain the protocol exchanges that would be needed to establish multicast communications between potentially thousands of participants as would be the case in large-scale highly-virtualized data centers.

These large-scale data centers must support a large number of tenants where the different tenants are allocated virtual machines (VMs) running on servers distributed throughout the data center. It is essential that tenant VMs can communicate effortlessly, regardless of their physical location within the data center. Additionally, there must be traffic and address space isolation between tenants; from the tenant's perspective, it is the only occupant of the data center. Traffic and address space isolation between tenants is achieved by creating a virtual network instance (VNI) for the tenant, logically connecting the tenant's VMs at Layer 2 or Layer 3 (at the tenant's discretion) to that virtual network and allowing traffic into or out of the virtual network via well-defined gateways. This virtual network is usually realized by encapsulating the traffic from the VMs and tunneling them over the underlying data center network. This virtual network, whose links are essentially tunnels in the data center IP fabric, is referred to as the *overlay network* and the underlying data center fabric that supports the overlay networks is referred to as the *underlay network*.

Much of the debate about data center architectures is about

- the nature of the underlay network: e.g. What is the ideal physical topology? Where should be the demarcation between the Layer 2 and Layer 3 domains?
- the nature of the overlay network: e.g. Do VMs connect to this network at Layer 2 or Layer 3? Which encapsulating protocol should be used?
- the relationship between the underlay and overlay networks: e.g. Where should the tunnel endpoints be situated? How do these endpoints discover each other?

978-3-901882-89-0 @2017 IFIP

This paper explores the challenges and opportunities associated with using multicast in the overlay network and the underlay network in virtualized data centers. Section II gives an overview of the evolution of data center architectures. Section III describes and compares different technologies proposed for implementing overlay networks in terms of a number of factors, including how they handle multicast traffic. Section IV explores innovative proposals to make the widespread usage of IP multicast in underlay networks more popular. Section V concludes the paper.

II. Evolving Data Center Architectures

A. Legacy multi-tenant data centers

Despite a number of different data center network topologies being proposed (for example [2], [3], [4]), the canonical data center network topology remains as shown in Figure 1. In Figure 1, there are rack of servers with each server connected to an access switch, typically positioned at the top of the rack (ToR). The servers could each host numerous VMs, potentially from different data center tenants. These access switches are connected to aggregation switches which may be attached to middleboxes. The middleboxes implement important network functions such as firewall (FW) and load-balancing (LB). These aggregation switches have a Layer 3 connection to core routers, which typically handle the data center's egress and ingress traffic. The servers may host VMs from different tenants and, arguably, the most intuitive way to achieve traffic and address space isolation betweens tenants is to assign each tenant to a different VLAN.

The hypervisors in the servers would typically run a virtual switch (vSwitch) to which all the VMs will be connected via a virtual network interface card (vNIC), as shown in Figure 2. The hypervisor would inform the vSwitch about the vNICs assigned to each VM and the associated VLAN ID. Hence, it may be said that the vSwitch places the port connected to each vNIC in the appropriate VLAN and creates a trunk link towards the ToR switch. Thus, when an untagged Ethernet frame is received from a particular VM, the vSwitch knows the VLAN ID to use to tag the frame before forwarding appropriately. And the converse operation occurs when a tagged Ethernet frame is received from the ToR switch.

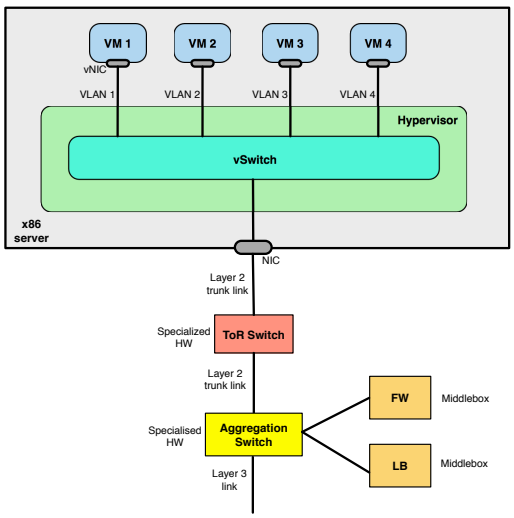

Fig. 2. Potential legacy data center deployment

The restrictions to 4094 VLANs due to a 12-bit VLAN ID field is one of the biggest reasons why VLANs alone are not the main way multi-tenancy is supported. Additionally, VLANs do not create an overlay network; Ethernet frames are tagged rather than encapsulated and so the forwarding tables within the data center fabric may be excessively large. Furthermore, since a VLAN is a broadcast domain, having large VLANs can lead to an excessive volume of broadcast traffic due to convention of flooding broadcast, unknown unicast and multicast (BUM) traffic. Lastly, an instance of Spanning Tree Protocol (STP) [5] runs per VLAN which may lead to a relatively poor utilization of links in the VLAN in an attempt to avoid forwarding loops.

B. Modern multi-tenant data centers

The pervasiveness of virtualization in computing and networking [6] has led to an ever increasing amount of network functions such as routing, network address translation (NAT), load balancing and firewall being realized by VMs running on inexpensive commodity x86 servers rather than expensive proprietary customized hardware. The consensus is that network function virtualization (NFV) [7] will play a pivotal role in future data centers. Consequently, a significant amount of effort has been devoted to developing tools to orchestrate these virtual network functions (VNFs), allowing them to be deployed judiciously within the data center. Typically, these VNFs are chained together into a service chain [8], meaning packets flow between the VNFs, being processed as desired.

Figure 3 shows a possible deployment where a virtual router (vRouter) runs as a VM and is connected to the vSwitch, as are the tenants' VMs. In this case, the vRouter can maintain a per-tenant VRF to achieve address space and traffic isolation. Additionally, a number of other VNFs are running as VMs on the server and, along with the vRouter, form an NFV service chain. The vRouter is an important component in this architecture and a key requirement is that vRouters should be capable of achieving packet forwarding rates comparable to the traditional routers that they are seeking to replace [9]. To this end, a lot of effort has been devoted to circumventing the performance bottlenecks typically encountered by Linux-based routers. For example, two key performance bottlenecks and the way they are overcome, depicted in Figure 3, are:

- *the Linux kernel forwarding path* is not optimized for packet forwarding. This fact is unsurprisingly given that Linux was designed as a generic OS rather than an OS for packet forwarding. Intel's Data Plane Development Kit (DPDK) [10] offers a way to bypass the Linux kernel forwarding path by providing libraries that allow an optimized user space forwarding process to receive and send packets from the NIC directly. Thus much higher

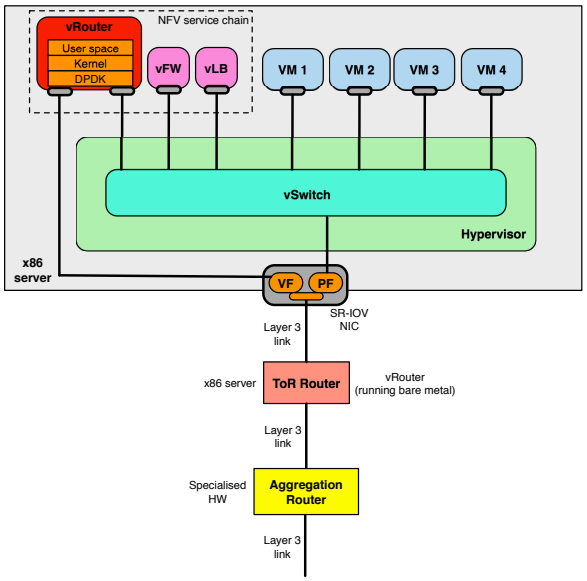

Fig. 3. Potential virtualized data center deployment

forwarding rates than would otherwise be possible are achieved.

- *the vSwitch in the hypervisor* is an I/O bottleneck since it handles all packets destined for all the VMs running. Single-root input/output virtualization (SR-IOV) [11] is a hardware virtualization technology that essentially allows a NIC to appear as multiple virtual functions (VFs), lightweight Peripheral Component Interconnect Express (PCie) [12] devices optimized for data I/O, which are associated with the underlying physical function (PF), the comprehensive PCie device. Thus VMs may be associated directly with VFs so bypassing the vSwitch and achieving forwarding rates comparable with if the VM had exclusive access to the NIC.

The key fact to note is that much of the functionality that was realized using specialized hardware in Section II-B can be realized by VMs running on commodity servers in future large-scale data centers. It is expected that these highly-virtualized data centers will scale to supporting thousands of VMs, which will be orchestrated in a highly dynamic and fluid manner to meet the varying requirements of the data center operator and tenants.

III. Multicast in Data Center Overlay Networks

A. Overview

Regardless of the data center architecture used, there is a requirement to support connectivity between a tenant's VMs located on different physical servers distributed throughout the data center. This requirement, met using an overlay network [13], may be described as to facilitate Layer 2 or Layer 3 connectivity with address and traffic isolation between a number of endpoints over a core network. Put in these terms, it is apparent that overlay networks in data centers are analogous

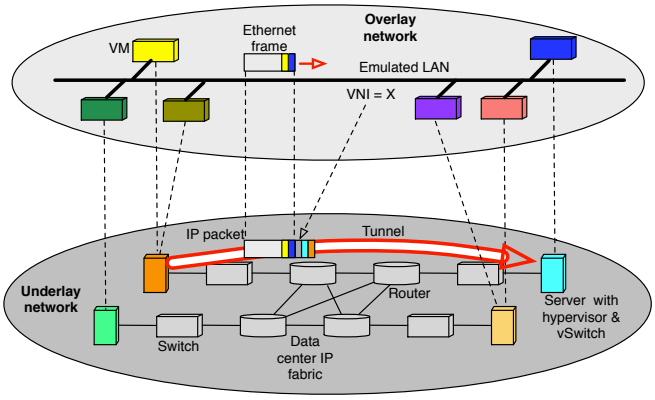

Fig. 4. Examplar overlay and underlay network

to traditional Layer 2 Virtual Private Networks (L2VPNS) and Layer 3 Virtual Private Networks (L3VPNs) [14]. In fact, certain existing L2VPN and L3VPN technologies may be adapted and used to build the overlay network. This category of overlay networks is sometimes referred to as *network-based overlays* because the overlay network endpoints, i.e. the tunnel endpoints, are located within the data center network, typically within the ToR switches or vRouter. On the other hand, in *host-based overlays*, the overlay network endpoints are normally located within the hypervisor. From this distinction, it may be inferred that network-based overlays are mainly promulgated by the traditional network equipment vendors whereas host-based overlays are favored by companies that develop virtualization technologies. Since *network-based overlays* have been in use for longer and their relationship with multicast is better understood and documented, attention is mostly paid to *host-based overlays* in this paper.

Figure 4 illustrates an overlay technology creating an emulated LAN between VMs located on different physical servers in the data center. It can be seen that the Ethernet frame from the VM is encapsulated so it traverses the underlay network via the tunnel. Critically, the encapsulation headers must include a reference to the VNI so the packet can be associated with the correct VNI when it is decapsulated. The VMs are unaware of the overlay network, the encapsulation or the underlay network; as far as they are concerned, they are connected to the other VMs in the VNI via a conventional Layer 2 segment, as illustrated Figure 4. Similarly, the core routers in the underlay network are totally oblivious to existence of the overlay network; they are simply forwarding IP packets and are indifferent to the payload.

The function of the overlays may be described as to *map and encap*. *Map*ping a destination MAC address to the corresponding remote overlay network endpoint IP address at the source overlay network endpoint is performed by the control plane. The control plane is described in Section III-B. *Encap*sulating the original frame for transmission across the data center fabric is a function of the data plane which is described in Section III-C.

Control plane	Description	Strengths	Weaknesses	Multicast traffic handling
Data-driven flood and learn	Overlay tunnel endpoints mimic switch behavior by broadcasting ARP requests (to discover the MAC-to-IP mapping) and flooding frames with unknown unicast destination addresses (to facilitate MAC learning)	No control plane protocol	Large traffic volume, multicast required in underlay network	Multicast traffic is flooded to all endpoints (analogous behaviour to switch with IGMP snooping [23] disabled)
Distributed protocol (i.e. MP-BGP EVPN [15])	Each overlay tunnel endpoint peers with BGP route reflector (RR) and exchanges MAC address and IP address information with remote tunnel endpoints via BGP	Flooding eliminated, control plane and data plane separation	Tunnel endpoints required to support BGP	BGP messages exchanged to facilitate multicast tunnel endpoint discovery and to support underlay networks with different multicast capabilities (e.g. source replication, IP multicast forwarding etc.)
Centralized controller	Overlay tunnel endpoints send information about MAC addresses and IP addresses to centralized controller which computes and disseminates globally optimal forwarding state to other tunnel endpoints	Globally optimal and consistent forwarding state, control plane and data plane separation	Scalability and security concerns due to centralized controller	Endpoints send information about attached multicast sources/receivers to controller which is able to program switches/routers in the underlay with correct multicast or source replication forwarding state

TABLE I
COMPARISON OF POPULAR OVERLAY TECHNOLOGIES CONTROL PLANES

B. Control Plane

The control plane is responsible for mapping a destination MAC address to the corresponding remote overlay network endpoint IP address so the frame can be encapsulated correctly and delivered to the correct destination VM. Some of the different control planes that have been proposed for overlay networks in virtualized data centers are summarized in Table I.

C. Data Plane

The most popular data planes have been defined for host-based overlays are described in the following sections and summarized in Table II.

1) VXLAN/VXLAN-GPE: Virtual eXtensible Local Area Network (VXLAN) [16] encapsulates an Ethernet frame in a VXLAN header, a UDP header (with port number 4789) and an IP header. Critically, The VXLAN header contains a 24bit VXLAN network identifier (VNI) field. A desire to carry payloads other than Ethernet was one of the main drivers behind the proposal of a generic protocol extension (GPE) for VXLAN [17]. VXLAN-GPE uses part of the reserved field in the VXLAN header to indicate the payload protocol type and a different UDP port (4790). The VXLAN standard [16] proposed the use of a data-driven flood and learn conrol plane which requires the underlay to support multicast. In the proposal, each VNI is mapped to a multicast group in the underlay network. VXLAN tunnel endpoints (VTEPs) attached to VMs that wish to participate in the VXLAN segment join the multicast group. So essentially, BUM traffic in the VXLAN segment is mapped to multicast in the underlay network. Should the data center operator be unwilling to enable IP multicast in the underlay network, source replication may be used to send unicast copies of the same encapsulated frame to all the VTEPs, an approach that leads to duplicate packets traversing the same links in the overlay and underlay networks. Alternatively, as mentioned earlier a distributed control plane [15] or centralized controller may be use to reduce to amount of point-to-multipoint traffic in the overlay network.

2) NVGRE: Network Virtualization Using Generic Routing Encapsulation (NVGRE) [20] is similar to VXLAN, with the key difference that the Ethernet frame is encapsulated in a GRE/IP header rather than a VXLAN/UDP/IP header. The GRE header includes a 24bit virtual subnet ID which is used to distinguish between different VNIs. Like VXLAN, NVGRE also supports the mapping of a multicast group in the underlay network to a VNI for BUM traffic. If the underlay does not support multicast, then source replication is suggested as an alternative way of transporting point-to-multipoint traffic.

3) GENEVE: Generic Network Virtualization Encapsulation (GENEVE) [18] aims to be extensible and flexible by supporting the definitions of new extension headers that might emerge in the future and making no assumptions about the nature of the control plane. Thus, GENEVE is defined to have a base header with a number of fixed fields followed by variable length undefined options. An important field in the base header is the virtual network ID. Since GENEVE intentionally does not define a control plane, the specification does not specify the mechanism by which BUM traffic in the overlay network is supported, beyond observing that multicast in the underlay may be useful.

4) STT: To improve performance, some NICs offer a hardware TCP offload capability where the OS passes a large chunk of data (up to 64kbytes) to be transmitted to the NIC, along with some essential metadata. The NIC breaks up the data into TCP segments and, using the supplied metadata, adds the correct TCP, IP and MAC headers in preparation for transmission. Stateless Transport Tunneling (STT) [19] was proposed to exploit the TCP offload capabilities of NICs to achieve great throughput. So a VM can send up to 64kbyte of data to the hypervisor. The hypervisor adds the STT header before sending to the NIC for transmission, along with some metadata. The NIC splits up the data into TCP segments: the first TCP segment contains an IP header, a TCP header, the STT frame header and the beginning of the original payload (i.e. the Ethernet frame to be encapsulated) from the VM.

Subsequent TCP segments contain an IP header, the TCP header and the continuation of the payload. The TCP header is syntactically correct but is incomplete since a TCP session is actually not being established. STT has a 64bit context ID field in its header which may be used to identify the different VNIs. The STT specification intentionally does not define a control plane for the overlay. Hence, there is no special support for multicast defined, beyond the observation that if the underlay supports multicast, then a multicast address may be used as the tunnel destination address.

D. Summary

As Table I highlights, there are different strengths and weaknesses associated with the different control planes presented. The control planes based on MP-BGP EVPN and the centralized controller are the most promising and are attracting the most attention. Control planes based on data-driven flood and learn are deemed to be unsuitable for large scale virtualized data centers due to the large traffic volumes arising from the flooding and also the effective conflation of the control plane and data plane.

Table II summarizes the salient characteristics of the overlay data planes discussed in the preceding sections. There is much ongoing fervent debate regarding the relative merits of the different technologies. It is not yet clear which overlay technologies will emerge as the most durable and popular. Nevertheless, a striking observation from Table II is that these overlay technologies could potentially benefit from the data center underlay supporting IP multicast. Section IV explores some of the issues involved in enabling multicast in the data center IP fabric.

IV. Multicast in data center underlay networks

A. Overview

The role of underlay network is to provide IP connectivity between servers in the data center. Despite the fact that, as seen in Section III, most of the overlay network technologies can potentially benefit from IP multicast in this underlay, IP multicast is typically not enabled. The reasons for the infrequent use of IP multicast are readily apparent by examining the characteristics of the conventional IP multicast control plane and data plane:

- in the *IP multicast control plane*, information primarily flows from receivers to sources, allowing sources and intermediate nodes to know the relative position of the receivers and so create the appropriate forwarding state. Put simply, insight may be gained into the nature of the multicast control plane by answering a question such as "When the last-hop router becomes aware of a downstream receiver, what does it do?" In classical IP multicast, Protocol Independent Multicast (PIM) [24], IGMP [25] and Multicast Listener Discovery (MLD) [26] are the dominant control plane protocols. IGMP/MLD runs between hosts and the last-hop router and is used to track hosts' multicast group membership. PIM runs between between routers in the multicast domain and essentially seeks to build a multicast distribution tree between sources and receivers. PIM is a baroque protocol and, as such, introduces significant overhead and complexity in terms of the volume and type of protocol messages that must be exchanged between routers and the protocol state machinery that must be executed.
- in the *IP multicast data plane*, as would be expected, information flows from sources to receivers. The nature of the multicast data plane may be exposed by answering a question such as "When the first-hop router receives a multicast packet from the source, what does it do?" Typically, the multicast control plane protocols will have built a multicast distribution tree which identifies the outgoing interfaces for a given source and group address pair. Critically, since multicast addresses are not assigned hierarchically and so cannot be easily aggregated, there is usually a requirement to hold per-flow forwarding state in the multicast routers.

The need to run complex protocols to build and maintain a *multicast distribution tree* and maintain *per-flow forwarding state* are significant obstacles to widespread IP multicast deployment in data center networks (and, in reality, in the Internet in general [36]), especially given the desire to use commodity hardware rather than proprietary specialized forwarding hardware and the need to potentially support multicast group communication between thousands of VMs. Most of the proposals on making the use of IP multicast in data centers commonplace, some of which are discussed in Sections IV-B and IV-C, seek to address one or both of these challenges, often by exploiting idiosyncrasies of the data center network topology and architecture.

B. Avoiding construction of a multicast distribution tree

As alluded to in Table I, one way to avoid running complex protocols to construct and maintain a multicast distribution tree is to use a centralized controller, as defined in the Software Defined Networking (SDN) [37] and OpenFlow [38] architectures. Thus, multicast deployments based on SDN and OpenFlow are attracting much interest [35] and have been proposed for use in data centers. For example, [27] seeks to avoid the use of IGMP to track group membership at the tunnel endpoints but rather use an OpenFlow-based centralized controller to manage the multicast forwarding. A new SDN-based multicast routing algorithm that designed allows commodity switches to be used for multicast has also been proposed [34].

Bit Index Explicit Replication (BIER) [39] is a new paradigm for multicast forwarding that has many attractive properties for use in data centers [40]. Upon entering a BIER domain, the ingress router adds a BIER header to the packet. Critically, this header contains a bit string in which each bit maps to an egress router. If a bit is set, then the packet should be forwarded to the associated egress router. Simple bit-wise operations within the BIER domain result in the packet being forwarded to the correct set of receivers. The fact that BIER does not require the construction and maintenance of a

Data plane	Encapsulation headers	Encapsulation overhead (bytes)	Payload	Overlay network ID	Strengths	Weaknesses	Multicast traffic handling
VXLAN [16]	Ethernet-IP-UDP-VXLAN	50	Ethernet	24bit VNI	Widely deployed, multi-vendor support	Ethernet is only payload	Map VNIs to multicast group in underlay or use source replication
VXLAN-GPE [17]	Ethernet-IP-UDP-VXLAN	50	Ethernet, IP, NSH [21], MPLS	24bit VNI	Similar frame to VXLAN, suports different payloads	Different port from VXLAN	Map VNIs to multicast group in underlay or use source replication
GENEVE [18]	Ethernet-IP-UDP-GENEVE	50 (minimum)	Ethernet	24bit VNI	Extensible, control plane agnostic	Large and variably-sized header makes hardware support challenging	Exploit multicast support in underlay if possible
STT [19]	Ethernet-IP-*TCP*-STT	76 or 58	Ethernet	64bit context ID	Exploit NIC's TCP offload capabilities, 64bits to identify overlay network	Non-standard use of TCP	Exploit multicast support in underlay if possible
NVGRE [20]	Ethernet-IP-GRE	42	Ethernet	24bit virtual subnet ID	Based on GRE which is pervasive, similar to VXLAN	Lack of UDP/TCP header makes ECMP [22] routing support challenging	Map VNIs to multicast group in underlay or use source replication

TABLE II
COMPARISON OF POPULAR OVERLAY TECHNOLOGIES DATA PLANES

multicast distribution tree makes it attractive for use in data centers.

C. Avoiding per-flow forwarding state

Per-flow unicast forwarding state is avoided by assigning addresses hierarchically so routes may be aggregated. In contrast, the assignment of multicast addresses is unregulated in the Internet, making multicast address aggregation difficult. However, given that the data center network is under the exclusive control of the data center operator, it may be possible to assign multicast addresses hierarchically, thus allowing multicast routes to be aggregated. Furthermore, the fixed and regular data center network topology lends itself well to elegant multicast address aggregation schemes. Approaches along these lines have been proposed. For example, [28] exploits the topological properties of data center networks and a centralized controller to allow the total number of multicast groups supported in the data center to far exceed the maximum capacity of any individual switch. Similarly, source replication and a judicious approach to managing multicast addresses in the data center can be used to ensure that the multicast forwarding capacity of switches and routers are not exceeded has been proposed [29].

Another interesting approach to avoid having to maintain per-flow forwarding state for multicast is to encode the multicast tree into a Bloom filter, a probabilistic data structure for storing sets which supports membership queries [30]. This Bloom filter is usually carried in the packet's header and each router in the multicast domain can then use simple logical operations to decide whether the packet should be forwarded out of a specific interface. There have been a number of promising studies on applying Bloom filters to multicast forwarding in data centers, for example [31], [32], [33].

From the discussion on BIER forwarding Section IV-B it can be seen that BIER does not require per-flow multicast forwarding state, a property that makes BIER attractive for use in data centers.

D. Summary

It is encouraging to note that, despite the previously-mentioned obstacles to deploying IP multicast in data centers, there appears to be a consensus emerging that future large-scale highly-virtualized data centers must have a better solution to handling the increasing amount of point-to-multipoint traffic than the unsatisfactory fallback of source replication. Thus, numerous interesting innovative proposals are under development.

V. CONCLUSION

The challenges faced by an operator wishing to enable IP multicast within the data center IP fabric are significant, mostly due to the unsuitability of the conventional IP multicast control plane and data plane to large-scale, highly virtualized data centers. Nevertheless, the technologies used to achieve multi-tenancy at scale by building emulated Layer 2 networks could potentially benefit from the use of IP multicast within the data centers. Given the anticipated increase in the volume and importance of point-to-multipoint traffic flows, it is unsurprising that significant effort is being devoted to seeking innovative approaches to overcome some of the challenges that prevent the widespread usage of IP multicast in data centers. SDN-based approaches, in which a centralized controller optimized for use within the data center instructs the switches and routers of how to forward multicast traffic, are very attractive. BIER is also a particularly promising approach as it addresses most of the problems associated with using IP multicast within data centers and is currently being standardized within the IETF, with support from many of the leading equipment vendors.

REFERENCES

[1] D. Newman, 10 Gig access switches: Not just packetpushers anymore, Network World, Volume 25, Issue 12, Mar 2008

[2] M. Al-Fares, A. Loukissas, A. Vahdat, A Scalable, Commodity Data Center Network Architecture, Proc. ACM SIGCOMM 2008.

[3] R. Mysore *et al*, PortLand: A Scalable Fault-Tolerant Layer 2 Data Center Network Fabric, Proc. ACM SIGCOMM 2009.

[4] C. Guo *et al*, Bcube: A High Performance, Server-centric Network Architecture for Modular Data Centers, Proc. ACM SIGCOMM 2009.

[5] R. Perlman, An Algorithm for Distributed Computation of a Spanning Tree in an Extended LAN, ACM SIGCOMM Computer Communication Review, Volume 15, Issue 4, Sept 1985.

[6] R. Jain, S. Paul, Network Virtualization and Software Defined Networking for Cloud Computing: A Survey, IEEE Communications Magazine, November 2013.

[7] Network Functions Virtualisation: An Introduction, Benefits, Enablers, Challenges and Call for Action, https://portal.etsi.org/NFV/NFV_White_Paper.pdf.

[8] P. Quinn (Ed), T. Nadeau(Ed), Problem Statement for Service Function Chaining, RFC 7498, IETF, April 2015.

[9] Vyatta 5600 Performance Test Executive Summary, https://www.sdxcentral.com/wp-content/uploads/2014/10/Vyatta-5600-Performance-Test-Executive-Summary.pdf.

[10] DPDK Boosts Packet Processing, Performance, and Throughput, http://www.intel.com/content/www/us/en/communications/data-plane-development-kit.html.

[11] PCI-SIG SR-IOV Primer: An Introduction to SR-IOV Technology, http://www.intel.co.uk/content/www/uk/en/pci-express/pci-sig-sr-iov-primer-sr-iov-technology-paper.html.

[12] Specifications, http://pcisig.com/specifications.

[13] T. Narten (Ed), E. Gray (Ed), D. Black, L. Fang, L. Kreeger, M. Napierala, Problem Statement: Overlays for Network Virtualization, RFC 7364, IETF, October 2014.

[14] P. Knight, C. Lewis, Layer 2 and Layer 3 Virtual Private Networks: Taxonomy, Technology and Standardization Efforts, IEEE Communications Magazine, June 2004.

[15] A. Sajassi (Ed), J. Drake (Ed), A Network Virtualization Overlay Solution using EVPN, Internet draft, IETF, December 2016.

[16] M. Mahalingam *et al*, Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks, RFC 7348, IETF, August 2014.

[17] L. Kreeger (Ed), U. Elzur (Ed), Generic Protocol Extension for VXLAN, Internet draft, IETF, April 2016.

[18] J. Gross (Ed), I. Ganga (Ed), Geneve: Generic Network Virtualization Encapsulation, Internet draft, IETF, July 2016.

[19] B. Davie (Ed), J. Gross, A Stateless Transport Tunneling Protocol for Network Virtualization, Internet draft, IETF, April 2016.

[20] P. Garg (Ed), Y. Wang (Ed), NVGRE: Network Virtualization Using Generic Routing Encapsulation, RFC 7637, IETF, September 2015.

[21] P. Quinn (Ed) and U. Elzur (Ed), Network Service Header, Internet draft, IETF, September 2016.

[22] C. Hopps, Analysis of an Equal-Cost Multi-Path Algorithm, RFC 2992, IETF, November 2000.

[23] M. Christensen, K. Kimball, F. Solensky, Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches, RFC 4541, IETF, May 2006.

[24] B. Fenner, M. Handley, H. Holbrook, I. Kouvelas, Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised), RFC 4601, IETF, August 2006.

[25] B. Cain, S. Deering, I. Kouvelas, B. Fenner, A. Thyagarajan, Internet Group Management Protocol, Version 3, RFC 3376, IETF, October 2002.

[26] L. Vida (Ed), L Costa (Ed), Multicast Listener Discovery Version 2 (MLDv2) for IPv6, RFC 4310, IETF, June 2004.

[27] Y. Nakagawa, K. Hyoudou, T. Shimizu, A Management Method of IP Multicast in Overlay Networks using OpenFlow, Proc. of Workshop on Hot topics in Software Defined Networks (HotSDN), ACM, 2012.

[28] X. Li, M. Freedman, Scaling IP multicast on datacenter topologies, Proc. of Conference on Emerging Networking Experiments and Technologies, (CoNEXT), ACM, 2013.

[29] Y. Vigfusson, H. Abu-Libdeh, M. Balakrishnan, K. Birman, R. Burgess, G. Chockler, H. Li, Y. Tock, Dr. Multicast: Rx for Data Center Communication Scalability, Proc. of European Conference on Computer systems, (EuroSys) 2010.

[30] X. Tian, Y. Cheng, Bloom Filter-based Scalable multicast: Methodology, Design and Application, IEEE Network, Volume 27, Issue 6, December 2013.

[31] D. Li, J. Yu, J. Yu, J. Wu, Exploring Efficient and Scalable Multicast Routing in Future Data Center Networks, Proc. International Conference on Computer Communications (INFOCOM), IEEE, 2011.

[32] D. Li, Y. Li, J. Wu, S. Su, J. Yu, ESM: Efficient and Scalable Data Center Multicast Routing, IEEE/ACM Transactions on Networking, Volume 20, Issue 3, June 2012.

[33] P. Jokela, A. Zahemszky, C. Rothenberg, S. Arianfar, P. Nikander, LIPSIN: Line Speed Publish/subscribe Inter-networking, Proc. ACM SIGCOMM 2009.

[34] A. Iyer, P. Kumar, V. Mann, Avalanche: Data Center Multicast using Software Defined Networking, Proc. of International Conference on Communication Systems and Networks (COMSNETS), 2014.

[35] W. Gu, X. Zhang, B. Gong, L. Wang, A Survey of Multicast in Software-Defined Networking, Proc. of International Conference on Information Engineering for Mechanics and Materials (ICIMM), 2015.

[36] C. Diot, B. Levine, B. Lyles, H. Kassem, D. Balensiefen, Deployment Issues for the IP Multicast Service and Architecture, IEEE Network, Volume 14, Issue 1, January 2000.

[37] Open Networking Foundation, http://www.opennetworking.org/.

[38] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, OpenFlow:Enabling Innovation in Campus Networks, ACM SIGCOMM Computer Communication Review, Volume 38, Issue 2, April 2008.

[39] I. Wijnands (Ed), E. Rosen (Ed), Multicast using Bit Index Explicit Replication, Internet draft, IETF, July 2016.

[40] N. Kumar *et al*, BIER Use Cases, Internet draft, IETF, July 2016.