A Management Method of IP Multicast in Overlay Networks using OpenFlow

Yukihiro Nakagawa   Kazuki Hyoudou   Takeshi Shimizu

Fujitsu Laboratories Ltd.
Kawasaki, Kanagawa, Japan
{yukihiron, hyoudou.kazuki, shimizu.takeshi}@jp.fujitsu.com

ABSTRACT

Overlay networks stretch a Layer 2 network and increase mobility of virtual machines. VXLAN (Virtual eXtensible LAN) is one of Layer 2 overlay schemes over a Layer 3 network proposed in IETF and its definition covers 16M overlay networks or segments which solves 4K limitation of VLANs. However VXLAN uses IP multicast for the isolation of network traffic by tenant in the shared network infrastructure. IP multicast requires great amount of resources such as IP multicast table and CPU therefore the scalability is to be limited by handling of IP multicast. We propose to manage IP multicast in overlay networks using OpenFlow instead of using dynamic registration protocol such as IGMP. We describe our implementations of VXLAN controller, edge switch with VXLAN gateway and OpenFlow switch. Our method using OpenFlow eliminates periodical Join/Leave messages and achieves more than 4k tenants in our Layer 2 network at server edges, which was not possible before.

Categories and Subject Descriptors

C.2.1 [**Computer Communication Networks**]: Network Architecture and Design; C.2.3 [**Computer Communication Networks**]: Network Operations

General Terms

Experimentation, Design, Management

Keywords

Ethernet switch, Overlay Networks, OpenFlow switch

1. INTRODUCTION

A cloud data center requires dynamic infrastructure and flexible allocation of computing resources on demand. The virtualization technology and virtual machine mobility are important to realize dynamic infrastructure [1, 2]. Overlay networks increase virtual machine mobility significantly by

stretching a Layer 2 domain over Layer 3 network [3, 4]. VXLAN (Virtual eXtensible LAN) is one of Layer 2 overlay schemes proposed in IETF [3]. We picked up VXLAN as an example for clarification although we believe our discussion here applies to not only overlay networks but also common IP multicast problems in IP Network.

We have developed 10Gb Ethernet (10GbE) Layer 2 switches [5, 6, 7] and built Layer 2 networks at server edges. To apply overlay networks in this environment, we encounter following problems: 1) hardware resource or table size for IP multicast, 2) CPU resource for IGMP snooping. In our switches, the table size is 16K and it is not an issue to achieve more than 4k tenants. However we need to solve CPU resource problem. To overcome this, We propose to manage IP multicast in overlay networks using OpenFlow instead of using dynamic registration protocol such as IGMP. With our method, CPU resource problem is solved by eliminating periodical Join/Leave messages and more than 4k tenants can be accommodated in the overlay networks. Also by using OpenFlow, we can make use of multipath in the Layer 2 network.

This paper is organized as follows: Section 2 describes VXLAN overlay networks and the network traffic isolation using IP multicast. Section 3 describes our proposed management method of IP multicast using OpenFlow and an enhancement of OpenFlow action using the vendor extension. Section 4 describes implementations of VXLAN controller, edge switch with VXLAN gateway and OpenFlow switch. We also discuss the efficiency of our method in terms of the number of control messages. Section 5 describes related work and Section 6 provides conclusions.

2. VXLAN OVERLAY NETWORK

VXLAN is an IP multicast application and it uses IP multicast to isolate network traffic.

2.1 Network Traffic Isolation

VXLAN realizes tunneling of Ethernet frames by encapsulating the original frames with additional headers: Outer Ethernet, Outer IP, Outer UDP and VXLAN headers. The outer IP header includes IP address of VXLAN Tunnel End Point (VTEP) which originates and/or terminates VXLAN tunnels. Also VXLAN header includes 24 bit of VXLAN Network Identifier (VNI) or VXLAN segment ID and allows up to 16 million VXLAN overlay networks or segments to coexist within the same physical network infrastructure.

In VXLAN, a broadcast packet is sent out to an IP multicast group on which that VXLAN overlay network is re-

alized. To make this work, we need to have a mapping between the VXLAN VNI and the IP multicast group that it will use. This mapping is done at the management layer and provided to the VTEPs through a management channel. Using this mapping, the VTEP can provide IGMP membership reports to the upstream switch/router to join/leave the VXLAN related IP multicast groups as needed [8, 9]. This will enable pruning of the leaf nodes for specific multicast traffic addresses based on whether a member is available on this host using the specific multicast addresses. The IGMP membership report is periodically sent from VTEPs to the upstream switch/router to keep the membership alive. Between routers, a multicast routing protocol like Protocol Independent Multicast - Sparse Mode (PIM-SM) [10] is used to build efficient multicast forwarding trees so that multicast frames are only sent to those hosts which have requested to receive them. In PIM-SM, the membership information is propagated toward a Rendezvous Point (RP) using Join/Prune messages. In addition to the routers, Layer 2 switches on the segment snoop IGMP messages to prune the leaf nodes for the multicast addresses.

2.2 Overlay Network Reconfiguration

Figure 1 shows an example of the network traffic isolation by VXLAN segment when a VM is moved. VXLAN controller is a management layer of VXLAN and manages VTEPs. VXLAN controller configures a mapping between the VXLAN VNI and the IP multicast group. And it also configures VTEP to join the IP multicast group when the VM is moved to the VTEP. For example, VM1 is mapped to VNI 10 at VTEP2 and VNI 10 is mapped to IP multicast group of 224.0.1.0. Then VTEP2 issues IGMP membership report to the upstream router to join the IP multicast group of 224.0.1.0 (VNI 10). The upstream router registers VTEP2 to the IP multicast group. Also the switches between VTEP2 and the router snoop the IGMP membership report and register VTEP2 to the IP multicast group.

The upstream router sends IGMP membership query typically every 125 seconds and individual VTEP replies a membership report within 10 seconds on every IP multicast group it uses. After this registration for the IP multicast group, the migration of VM1 will complete and the hypervisor will send an ARP broadcast packet. This broadcast packet is sent out to the IP multicast group. Upon the reception of the broadcast packet, the association of VM's MAC to VTEP's IP is discovered via source learning. Meanwhile VTEP1 sends IGMP Leave message to the upstream router to unregister membership to the IP multicast group. For the Leave message, the upstream router sends group specific IGMP membership query to check if there is any other VTEPs for the IP multicast group.

The number of periodical IGMP membership reports is $O(g*v)$, here g is the number of IP multicast groups and v is the number of VTEPs. For example, 16k segments * 1k VTEPs becomes 16M messages/10 seconds or 1.6M messages/second. The management CPU needs to process one message every 0.6 microseconds.

3. OPENFLOW CONTROL

We propose to use OpenFlow to solve the dynamic group member joining/leaving problems which are common multicast problems in IP Network.
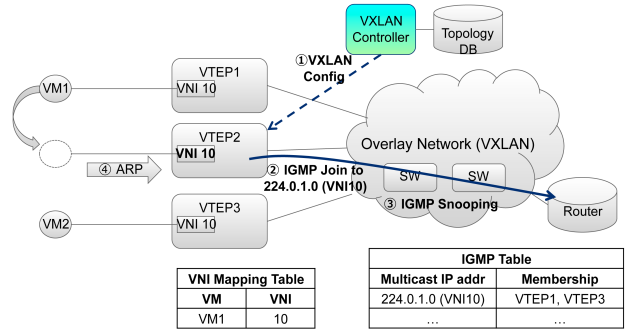

Figure 1: Isolation of Network Traffic by VXLAN Segment using IP Multicast

3.1 IP Multicast Management

As described above, the network traffic isolation is realized by using IP multicast in VXLAN. However IGMP or IGMP snooping requires lots of resources such as IP multicast table and CPU, therefore high-performance, commodity 10GbE switches support small number of IP multicast addresses ranging from 3K to 8K entries [11, 12, 13, 14]. Essentially, IGMP protocol is designed for dynamic registration of the membership in mind so that each application can join/leave the multicast group anytime without any intervention of the management.

We propose to use OpenFlow [15] for IP multicast management of VXLAN segments. In our application, the Layer 2 (MAC address) flow table definition is enough for IP multicast management. We can make our Layer 2 switch working as a OpenFlow switch with the flow table. Also we can preregister all VTEP address (unicast) and multicast addresses. We do not need to use Packet-In messages.

Figure 2 shows a management method of IP multicast using OpenFlow for VXLAN segments we propose. VXLAN controller configures a mapping between the VXLAN VNI and the IP multicast group. And it also configures VTEP to join the IP multicast group when a VM is moved to the VTEP. For example, VM1 is mapped to VNI 10 at VTEP2 and VNI 10 is mapped to IP multicast group of 224.0.1.0. Instead of issuing an IGMP membership report to the upstream router to join the IP multicast group, VXLAN controller modifies the flow table in the upstream router for the IP multicast group of 224.0.1.0 (VNI 10) to include VTEP2 as a receiver. VXLAN controller also modifies the flow table in the switches between VTEP2 and the router to include VTEP2 as a receiver of the IP multicast group.

Even though the upstream router sends IGMP membership query typically every 125 seconds, individual VTEP does not need to reply a membership report on any IP multicast group it uses for the isolation of network traffic by VXLAN segment.

After this flow modification for the tenant, the migration of VM1 will complete and the hypervisor will send an ARP broadcast packet. This broadcast packet is sent out to the IP multicast group. Upon the receipt of the broadcast packet, the association of VM's MAC to VTEP's IP is discovered via source learning. Meanwhile VXLAN controller modifies the flow table in the upstream router and switches to exclude VTEP1 from the IP multicast group. No group specific IGMP membership query message is needed to check
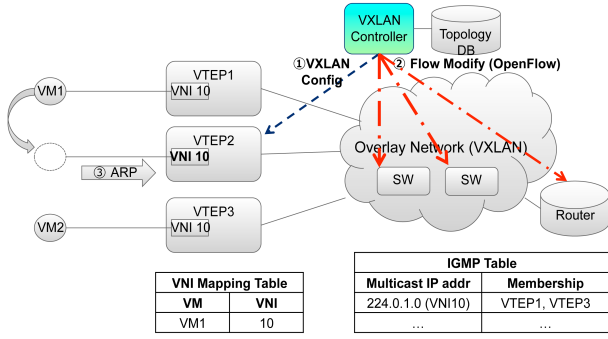
Figure 2: Management of IP Multicast using Open-Flow

Figure 3: Prototype of IP Multicast Management using OpenFlow for VXLAN Segments

Figure 4: 10GbE Switch used for Prototyping of Edge and OpenFlow Switches

if there is any other VTEP for the multicast group because VXLAN controller manages VTEPs based on the network topology and VM locations.

3.2 Protocol Requirements

In OpenFlow protocol, there are two programming modes in the flow table modification: The one is reactive mode and the other is proactive mode. In the reactive mode, an unknown packet is forwarded to a central controller and the controller will decide the action for the unknown packet, for example, to setup the flow table to forward the packet to a specific port. On the other hand, in the proactive mode, the controller will setup the flow table in advance so that the packet is forwarded to a specific port.

In our case, we can use the proactive mode and setup the flow table in advance using the flow table modification message because the VXLAN controller manages the VXLAN segments. No Packet-In message is necessary in the proactive mode, making the ASIC hardware implementation easier.

Also in our case, we need to forward a packet to multiple output ports for IP multicast based on the network topology and VM locations. To realize this, we need to enhance the output action of OpenFlow protocol to specify multiple output ports.

4. IMPLEMENTATION

Figure 3 shows a prototype of IP multicast management using OpenFlow. The prototype includes VXLAN controller, edge switch with VTEP for VXLAN gateway and OpenFlow switch. A VTEP can be located either of hypervisor, edge switch or router. We implemented VTEP at the edge switch for the prototype. Figure 4 shows 10GbE switch we used for prototyping of edge and OpenFlow Switches.

4.1 VXLAN Controller

VXLAN Controller manages VXLAN segments and configures the VTEPs. For the VTEP configuration, VXLAN controller uses CLI and configures a mapping between the VXLAN VNI and the IP multicast group. And it also configures VTEP to join the IP multicast group when a VM is moved to another VTEP. For OpenFlow switch configuration, VXLAN controller uses OpenFlow controller to communicate with OpenFlow switch over the secure channel and configures the flow table.
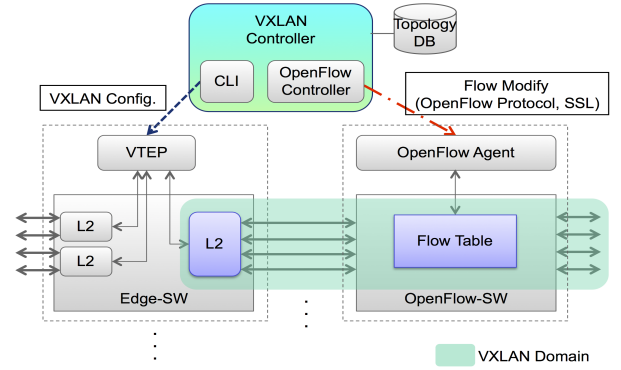
Figure 5 shows examples of the VTEP configuration by using VXLAN controller GUI. A different color shows a different segment. An administrator selects a port for a VXLAN segment and specifies VNI Information such as VNI and the domain IP address for the IP multicast group.

4.2 Edge Switch with VTEP

An edge switch with VTEP function works as a VXLAN gateway. For the prototype, we implemented VTEP function at a dedicated external server that is connected to the edge switch.

Table 1 shows VNI Mapping policy of VTEP at the edge switch. The input frame could be either of untagged or C-tagged. There are four mapping policies: vlan-bind, port-bind, pg-bind and mac-bind. Vlan-bind maps VLAN ID to VNI. Port-bind maps port number to VNI and it pass through both untagged and C-tagged packets. Pg-bing maps port group to VNI when one tenant uses multiple ports. Mac-bind maps source MAC address and VLAN ID if any to VNI and it enables mapping per VM basis for offloading of hypervisor VTEP function.

Figure 6 shows examples of VNI Mapping Policy. There are two port groups (PG1 and 2) configured at VXLAN Gateway 1. VLAN 100 of PG1 and VLAN 100 of PG2 communicates with the storage connected to VXLAN Gateway 2. To isolate tenants or groups, VLAN 100 of PG1 is mapped to VNI 10 and VLAN 100 of PG 2 is mapped to VNI 30. And VXLAN Gateway 2 maps VNI 10 and 30 to VLAN 100 and 200 respectively for the traffic isolation. Also VLANs 1 to 99 are mapped to VNI 40 at VXLAN Gateway 1 while keeping C-tag (inner VLAN Tag) with VLAN handling op-
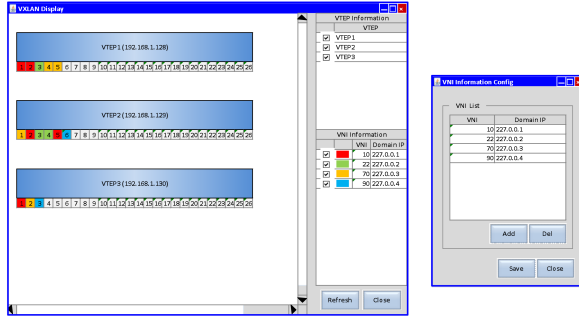
Figure 5: VTEP Configuration by VXLAN Controller

Table 1: VNI Mapping Policy of VTEP at Edge Switch

Input	Mapping policy	Mapping Operation			Inner-Tag handling option
untag	port-bind	port no	->	vni	n/a
	pg-bind	port group	->	vni	n/a
	mac-bind	mac	->	vni	n/a
c-tagged	vlan-bind	vlan id	->	vni	through or del/add
	port-bind	port no	->	vni	through or del/add
	pg-bind	port group	->	vni	through or del/add
	mac-bind	mac+vlan id	->	vni	through or del/add

tion of "through" mode. Server 3 (PG2) can communicate with Server 4 using VLAN in the tenant.

4.3 OpenFlow Switch

An OpenFlow switch provides a flow-table forwarding model to be managed by the controller. The flow table contains a set of flow entries (header values to match packets against), activity counters, and a set of zero or more actions to apply to matching packets. All packets processed by the switch are compared against the flow table. If a match is found, any actions for the entry are performed on the packet. For example, the action might be to forward a packet to a specified
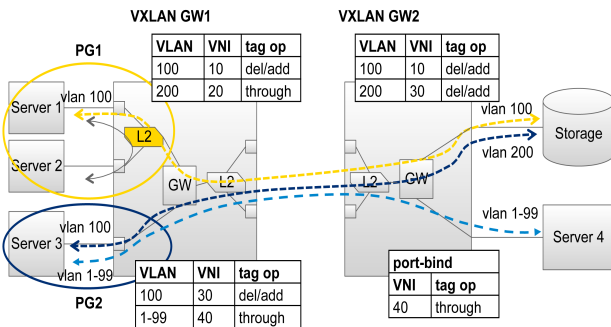

Figure 6: Example Use of VNI Mapping Policy

```
struct ofp_action_output_vendor {
    uint16_t type;        /* OFPAT_VENDOR. */
    uint16_t len;         /* Length is 16. */
    uint64_t portvec;  /* Output port vector. */
    uint8_t pad[4];       /* Pad to 64 bits. */
};
OFP_ASSERT(sizeof(struct ofp_action_output_vendor) == 16);
```

Figure 7: OpenFlow Vendor Action for IP Multicast

output port. If no match is found, the packet is forwarded to the controller over the secure channel. The controller is responsible for determining how to handle packets without valid flow entries.

In our application, the VXLAN controller manages VXLAN segments and proactively configures the flow table for IP multicast group based on the tenant information. Packet-In message is not necessary because all entries for VXLAN segments are configured proactively. The required action for IP multicast address is to forward packet to one or more specified ports. However OpenFlow actions do not provide packet forwarding to multiple destination ports other than "Forward All". Therefore we extend the output action to include output port vector. Figure 7 shows the vendor action for the IP multicast address.

The VXLAN controller configures the flow table using Flow Modify message in OpenFlow protocol. This output action can be mapped to ASIC function. In a Layer 2 switch, the flow table for IP multicast is mapped to the MAC address table and the destination MAC address of the incoming packet is compared against the MAC address table. In a router, the flow table for IP multicast is mapped to the IP multicast table and the destination IP multicast address is compared against IP multicast table.

Our implementation of OpenFlow switch is based on Open-Flow 1.0 [16]. Current OpenFlow specifications are not ASIC friendly and requires an abstraction for the forwarding plane [17, 18]. Open Networking Foundation (ONF) formed a discussion group called Openflow-future and they are working on Forwarding Plane Models (FPMODs).

Our current implementation of OpenFlow switch includes an OpenFlow agent running on a dedicated external server connected to the switch. The agent communicates with the controller over the secure channel and it configures the switch using CLI. The OpenFlow agent will be implemented in the switch firmware for the management when an AISC-friendly advanced OpenFlow architecture becomes available.

4.4 Discussion

Table 2 shows the specifications of commercially available 10GbE switches [11, 12, 13, 14]. The switches typically support 4K entries in the VLAN table and 3K to 8K entries in the multicast table.

Table 3 shows the number of IGMP/OpenFlow messages for IP multicast management. Using IGMPv2, a router sends a general query message to 224.0.0.1 every Query Interval to keep membership of the multicast groups. The VTEPs respond it by the membership reports within Max Response Time. The number of periodical Join messages is g*v for a query where g is the number of multicast group and v is the number of VTEPs.

Table 2: 10GbE Switch Specification

Switch	Dell Force10 S4810	Cisco Nexus3064	Arista 7050S-64	Juniper QFX3500
Switch capability	1.28Tbps	1.28Tbps	1.28Tbps	1.28Tbps
VLAN table	4k	4k	4k	4k
Mac table	128K	128k	128K	120K
L3 routing table	16K	8K	16K	8k
Multicast table	4K	8K	8K	3.5K

Table 3: Cost of IP Multicast Management

	Number of periodical messages per second	Number of non-periodical messages per second when VM move	Note
IGMP	Query: 1 / interval Join: g*v / (maxresp*gr)	Leave: 1*k Group-specific Query: 1*k Join: v*k	g >> k
OpenFlow	Query: 1 / interval Join: 0	Flow Modify: 2*k	

g: number of multicast group
v: number of VTEPs
k: VM mobility (number of VM move per second)
gr: 1 for IGMPv2, number of group records for IGMPv3
Interval: query interval,125 seconds (default)
maxresp: max response time, 10 seconds (default)

We do not consider membership report suppression in IGMPv2 because the membership report suppression does not work well in bridged LANs using IGMP snooping. A report generated by one host and heard by others that are group members on the same link causes the additional members to suppress their reports. If IGMP snooping switches were to forward reports to all attached interfaces, hosts on same LAN segments with group members may not be noticed. Therefore IGMP snooping switches avoid broadcasting reports out of all interfaces and they forward reports only to the nearest multicast router. The membership report suppression was removed in IGMPv3 [9].

Using IGMPv3, the membership report contains group records to report multicast groups. We do not consider IN-CLUDE (S, G) joins or EXCLUDE (S, G) joins introduced in IGMPv3 although hosts could maintain a list of sources that excluded or included. This is because the set of VXLAN tunnel sources is unknown and may change often, as the VMs come up/go down across different hosts therefore the VTEP will use (*, G) joins.

When a VM is migrated, a VTEP sends a Leave message. Upon the reception, the router sends a group-specific query message to the group IP address and VTEPs subscribe to the multicast group reply it by Join messages. The number of messages is 1*k for Leave messages, 1*k for group-specific messages and v*k for Join messages where k is the VM mobility or number of VM moves per second. Normally k is considered to be much smaller than g.

In our method using OpenFlow, the controller sends Flow Modify messages to install a new flow in the destination VTEP and uninstall an existing flow from the source VTEP. The number of messages is 2*k for Flow Modify messages which is much smaller than the number of messages in the method using IGMP.

4.5 Future Work

OpenFlow actions do not provide packet forwarding to multiple destination ports other than Forward All. Although we extended the output action to include output port vector using the vendor action for the IP multicast address, it is vendor specific and an interoperability issue arises. Therefore we think multicast packet handlings should be included in an updated version of OpenFlow specification.

5. RELATED WORK

There have been new research proposals for data center networks and an overlay network is one of the hottest research and development subjects. In overlay networks, Layer 2 networks are stretched over Layer 3 networks hence reducing broadcast traffic will be becoming more important and various researches are reported.

K. Elmeleegy and A. L. Cox [19] propose EtherProxy in which a proxy is used to suppress broadcast traffic of ARP and DHCP. A. Greenberg et al. [20] propose a directory based system VL2 in which an ARP broadcast packet is converted to an unicast query to the directory. H. Shah et al. [21] propose ARP broadcast reduction in VXLAN.

L. Kreeger et al. [22] propose using a control protocol for inner to outer address mapping in overlay networks to eliminate flooding in the source learning approach.

IEEE 802.1Qbg [23] and DMTF [24] standards define Automated Migration of Port Profile (AMPP) to reduce unnecessary traffic in VM migration. Y. Nakagawa et al. [25] propose AMPP for multi-level switches.

6. CONCLUSIONS

This paper described a management method of IP multicast in VXLAN overlay networks. If we simply apply VXLAN to Layer 2 network, we encounter the dynamic group member joining/leaving problems which are common multicast problems in IP Network. We controlled our Layer 2 switch using OpenFlow to solve the problems. Our method eliminated periodical Join/Leave messages and achieved more than 4k tenants in our Layer 2 network at server edges, which was not possible before. As a future work, we make use of benefits of OpenFlow and enforce multipath control in Layer 2 network.

7. ACKNOWLEDGMENTS

8. REFERENCES

[1] N. Carr. *The Big Switch: Rewiring the World, from Edison to Google.* W. W. Norton and Company, New York, 2008.

[2] P. Dawson and T. Bittman. Virtualization Changes Virtually Everything. *Gartner Special Report*, March 2008.

[3] M. Mahalingam, D. Dutt, K. Duda, P. Agarwal, L. Kreeger, T. Sridhar, M. Bursell, and C. Wright. VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks. *Internet Draft*, February 2012.

[4] M. Sridharan, K. Duda, I. Ganga, A. Greenberg, G. Lin, M. Pearson, P. Thaler, C. Tumuluri, N. Venkataramiah, and Y. Wang. NVGRE: Network Virtualization using Generic Routing Encapsulation. *Internet Draft*, September 2011.

[5] T. Shimizu, Y. Nakagawa, S. Pathi, Y. Umezawa, T. Miyoshi, Y. Koyanagi, T. Horie, and A. Hattori. A Single Chip Shared Memory Switch with Twelve 10Gb Ethernet Ports. *Hot Chips 15*, August 2003.

[6] Y. Nakagawa, T. Shimizu, Y. Koyanagi, O. Shiraki, S. Kobayashi, K. Hyoudou, T. Miyoshi, Y. Ogata, Y. Umezawa, T. Horie, and A. Hattori. A Single-Chip, 10-Gigabit Ethernet Switch LSI for Energy-Efficient Blade Servers. In *GreenCom 2010*, pages 404–411, December 2010.

[7] Y. Nakagawa, T. Shimizu, T. Horie, Y. Koyanagi, O. Shiraki, T. Miyoshi, Y. Umezawa, A. Hattori, and Y. Hidaka. *Energy-Aware Switch Design*. IGI Global, Pennsylvania, 2012.

[8] W. Fenner. Internet Group Management Protocol, Version 2. *RFC 2236*, November 1997.

[9] B. Cain, S. Deering, I. Kouvelas, and B. Fennerr. Internet Group Management Protocol, Version 3. *RFC 3376*, October 2002.

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

[11] DELL Force10. *Specifications: S4810 High-Performance 10/40 GbE Top-of-Rack Switch*, 2012.

[12] Cisco. *Nexus 3064 Switch Data Sheet*, 2012.

[13] Arista. *7050S-64 10/40G Data Center Switch Product Brief*.

[14] Juniper. *QFX3500 Switch Datasheet*, 2012.

[15] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. OpenFlow: Enabling Innovation in Campus Networks. In *SIGCOMM CCR*, pages 38(2):69–74, 2008.

[16] The OpenFlow Switch Consortium. *OpenFlow Switch Specification Version 1.0.0*, December 2009.

[17] S. Shenker. *Future of Networking, and the Past of Protocols*, October 2011.

[18] N. Yadav. *SDNs, OpenFlow 1.x, OpenFlow 2.0...*, December 2011.

[19] K. Elmeleegy and A. L. Cox. EtherProxy: Scaling Ethernet By Suppressing Broadcast Traffic. In *INFOCOM*, 2009.

[20] A. Greenberg, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. Maltz, P. Patel, and S. Sengupta. VL2: A Scalable and Flexible Data Center Network. In *SIGCOMM*, 2009.

[21] H. Shah, A. Ghanwani, and N. Bitar. ARP Broadcast Reduction for Large Data Centers. *Internet Draft*, October 2011.

[22] L. Kreeger, D. Dutt, T. Narten, D. Black, and M. Sridharan. Network Virtualization Overlay Control Protocol Requirements. *Internet Draft*, January 2012.

[23] IEEE. P802.1Qbg/D2.2 Draft Standard for Local and Metropolitan Area Networks-Virtual Bridged Local Area Networks -Amendment XX: Edge Virtual Bridging. *IEEE Draft Standard 802.1Qbg*, February 2012.

[24] DMTF. Virtual Networking Management White Paper, Version 1.0.0. *DSP2025*, February 2012.

[25] Y. Nakagawa, K. Hyoudou, S. Kobayashi, O. Shiraki, and T. Shimizu. Automated Migration of Port Profile for Multi-level Switches. In *DC-CaVES*, pages 22–29, September 2011.