Automated Migration of Port Profile for Multi-level Switches

Yukihiro Nakagawa, Kazuki Hyoudou, Shinji Kobayashi, Osamu Shiraki, and Takeshi Shimizu
Server Technologies Laboratory, IT Systems Laboratories
Fujitsu Laboratories Ltd.
Kawasaki, Kanagawa 211-8588, Japan
E-mail: yukihiron@jp.fujitsu.com, Tel. +81-44-754-2177

Abstract—**The use of virtualization technology has been increasing in the data center to consolidate physical servers into virtual machines and allocate computing resources dynamically by moving virtual machines. IEEE 802.1 Data Center Bridging Task Group is developing a standard to reduce complexity of the network management and automate a task to move network state in an adjacent bridge along with the migration of a virtual machine, which is called Automated Migration of Port Profile (AMPP). However AMPP works between a server and an adjacent bridge only. We propose an extension of AMPP for multi-level switches to further reduce complexity and automate the task. We developed a prototype of the extension and confirmed a network state is moved in a multi-level switch configuration using the standard protocol as it is on the wire. This paper describes IEEE standard based AMPP, our proposed extension of AMPP for multi-level switches, and a prototype of the extension.**

Keywords-Edge virtual bridging; Automated migration of port profile; Virtual machine migration

I. INTRODUCTION

A Cloud Data Center requires dynamic infrastructure and flexible allocation of computing resources to satisfy changing requirements of customers. The virtualization technology and virtual machine mobility are important to realize dynamic infrastructure [1], [2]. When a virtual machine is migrated, it is desired that the network state in the physical switches to be moved together for the network resource optimization and for better security.

To reduce complexity of the network management and move network state automatically, some venders are implementing their own protocols [3], [4] or utilizing hypervisor-dependent APIs [5]. These technologies are proprietary and cannot be used in a heterogeneous environment with physical switches from multiple venders. Also some technologies are lack of synchronization mechanism with hypervisor and therefore it is not guaranteed that physical switch configuration completes before a virtual machine migration ends.

To overcome these problems, IEEE 802.1 Data Center Bridging Task Group is developing a standard for Edge Virtual Bridging (EVB). As a part of EVB, Automated Migration of Port Profile (AMPP) is being defined and it is expected to reduce complexity of the network management and automate a task to move network state in an adjacent bridge along with the migration of a virtual machine [6], [7]. DMTF is also working on Edge Virtual Bridging and defining CIM data model for management, port profile database XML schema, and OVF extension [9], [10].

A dynamic infrastructure should allow flexible resource allocation across large server pools. However the standard-based AMPP works between a server and an adjacent bridge only. And external bridges other than the adjacent bridges need to be configured by a network manager using a proprietary management interface per switch vender. In addition, a management interface is not fast enough for VM mobility. For example, a switch takes one second to process a CLI command comparing a switch firmware takes milliseconds to process a message while a migration of VM with 512MB memory takes only two seconds in 10GbE network in Windows Server 2008 Hyper-V.

We propose an extension of AMPP for multi-level switches to further reduce complexity and automate the task. An adjacent bridge with the extension conditionally forwards control messages to an external bridge based on the internal status. If an adjacent bridge simply forward control messages to an external bridge, a port profile could be removed while a VM is running and AMPP fails. We developed a prototype of the extension and confirmed a network state being moved in a multi-level switch configuration using the standard protocol as it is on the wire. In other words, a bridge with AMPP extension capability can work with an external bridge with standard AMPP capability.

This paper describes IEEE standard-based AMPP, our proposed extension of AMPP for multi-level switches, and a prototype of the extension.

II. IEEE STANDARD-BASED AMPP

IEEE 802.1Qbg standard newly defines protocols for EVB. These are VSI Discovery and Configuration Protocol (VDP), Edge Control Protocol (ECP), and S-Channel Discovery and Configuration Protocol (CDCP).

VDP is an ECP-based protocol to associate port profile to Virtual Station Interface (VSI) of VM. ECP is a transport protocol to carry VDP TLV with re-transmission capability. CDCP is an LLDP-based protocol to configure S-Channel. VDP is the main protocol to realize the standard-based AMPP operation. Therefore we explain the protocol briefly.

978-0-9836283-2-3

This paper was peer reviewed by subject matter experts for publication in the Proceedings of DC–CaVES 2011

A. VSI Discovery and Configuration Protocol (VDP)

VDP simplifies and automates virtual station configuration by enabling the movement of a VSI instance and its related port profile from one EVB bridge to another. VDP protocol defines 4 messages: Pre-Associate, Pre-Associate with resource reservation, Associate, and De-associate messages.

Pre-Associate message is used to pre-associate a VSI Instance Identifier with a bridge port. Pre-Associate enables faster response to an associate, by allowing the bridge to obtain the port profile prior to an association. Pre-Associate with resource reservation message is similar with Pre-Associate but it also reserves resources in the bridge to prepare for a subsequent association. Associate message is used to activate an association between a VSI Instance and a bridge port and configure it with the port profile. De-associate message is used to remove an association between a VSI Instance and a bridge port.

B. Examples of VDP messages in VM Migration

Figure 1 shows an example of VDP messages communicated between a server and an adjacent bridge when a virtual machine is migrated. When a migration is initiated, Pre-Associate message is sent from the destination server to the adjacent bridge at a destination port and the bridge is preparing a port profile for the VM by fetching it from a database if necessary. Then a successful response or confirmation is returned to the destination server. During the stop and copy phase, Associate message is sent from the destination server to the adjacent bridge at the destination port and the bridge associates the port profile with the VM. The bridge returns a response to the destination server. On the other hand, De-associate message is sent from the source server to the adjacent bridge at the source port and the adjacent bridge de-associates the port profile with the VM. Then the VM is re-started in the destination server. In this way, the port profile is moved together with the VM and the physical switch is automatically configured.

C. AMPP in Multi-level Switch Configuration

As described above, the standard-based AMPP using VDP works between a server and an adjacent bridge. Therefore when a VM is migrated in a multi-level switch configuration, a network manager needs to configure non-adjacent external bridges as described in Figure 2 (a).

In a blade server configuration, there is a switch blade in the chassis and a VM migration to another chassis always requires an intervention of a network manager. Similarly when we use rack servers, a VM migration to another rack that is connected to another ToR switch requires an intervention of a network manager.

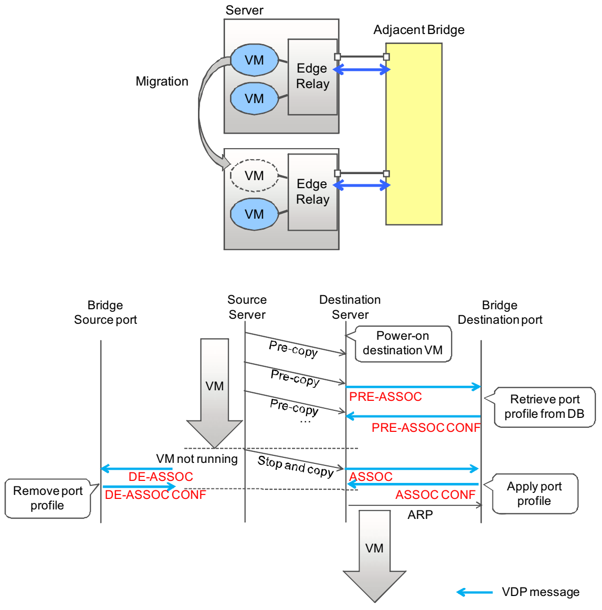
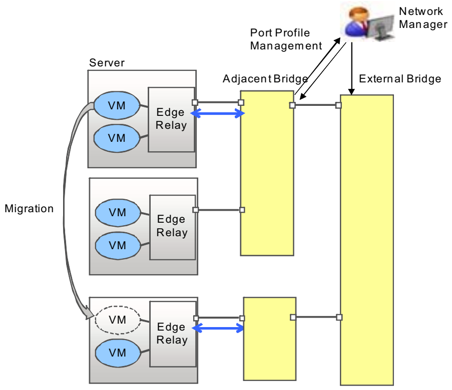
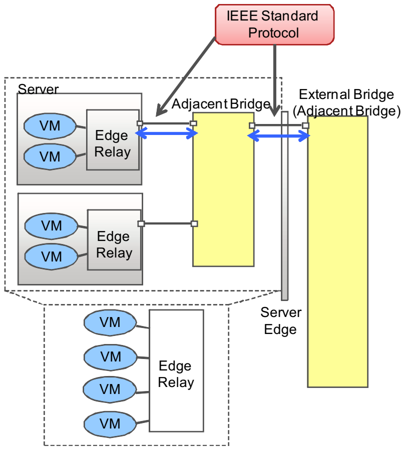


Figure 1. An example of VDP messages in VM migration



(a) Without AMPP Extension



(b) With AMPP Extension

Figure 2. AMPP Extension for Multi-level Switches

III. AMPP EXTENSION FOR MULTI-LEVEL SWITCHES

We propose an extension of AMPP for a multi-level switches so that we can automatically configure physical switches including non-adjacent bridges in addition to adjacent bridges. The extension consists of forwarding of VDP messages and conversion of edge relay mode.

Figure 2 (b) shows our idea in which we move the server edge to the outside of the adjacent bridge so that an external bridge works as an adjacent bridge. To realize this, an adjacent bridge forwards VDP messages to an external bridge based on the internal VDP status. Basically we do not forward VDP messages if a VM is migrated to a server that is connected with the same adjacent bridge. And we forward VDP messages if a VM is migrated to a server that is not connected with the adjacent bridge.

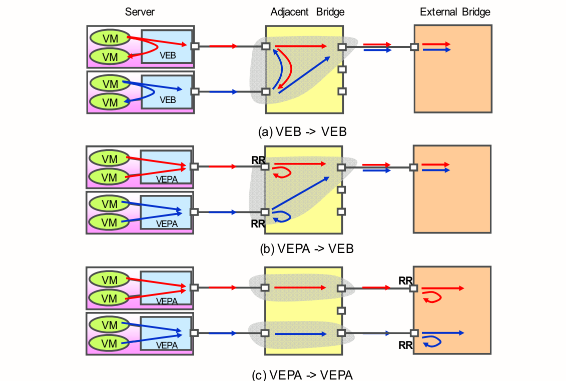
We note that if we forward VDP messages unconditionally, AMPP fails and causes an unexpected result. For example, when we move a VM to a server within the adjacent bridge, a port profile at the external bridge is removed when De-associate message comes after Associate message because De-associate message and Associate message are sent on the same uplink on the adjacent bridge which is connected to the external bridge.

Also EVB bridge capability needs to be advertized on LLDP so that VDP protocol works between an adjacent bridge and an external bridge. The conversion of Edge Relay mode on EVB TLV in LLDP can be done at the adjacent bridge to increase performance. TABLE I shows the combination of Edge Relay modes in a server and an adjacent bridge. The conversion from Virtual Edge Port Aggregator (VEPA) to Virtual Edge Bridge (VEB) enables reflective relay at the adjacent bridge and increases the performance. On the other hand, the conversion from VEB to VEPA causes packet duplication hence it is illegal. For example a VM transmit a multicast packet and the packet is forwarded to another VM via VEB in the server. The multicast packet is also sent to an adjacent bridge and it is reflective relayed to the server, resulting in a duplicated packet. Figure 3 shows handlings of Edge Relay Mode in legal combinations.

In our extension, we use standard protocol as it is on the wire. Therefore we can use a standard-compliant AMPP capable bridge as the most outer physical switch, for example, the external bridge in the configuration of Figure 2(b).

TABLE I. Combination of Edge Relay Mode

Server	Adjacent Bridge	Note
VEB	VEB	
VEB	VEPA	Illegal because of Packet Duplication
VEPA	VEPA	
VEPA	VEB	Performance Optimization



RR: Reflective Relay

Figure 3. Handling of Edge Relay Mode

A. *Forwarding of VDP Messages*

The forwarding decision of VDP TLV is made for each VDP TLV type (Pre-Associate, Pre-Associate with resource reservation, Associate, and De-associate) based on the state (vsiState = PREASSOC, PREASSOCR, ASSOC, DEASSOC) of the VDP state machine.

When the adjacent bridge received a VDP TLV from a server, it retrieves Virtual Station Instance ID (VSIID) in the TLV and search VDP entries with the VSSID. Up to two entries with same VSSID could exist because of transient status in the VM migration.

Table II shows the forwarding conditions of VDP messages. Pre-Associate with resource reservation is omitted in the table for simplicity of explanation. For example, Pre-Associate message is received when vsiState of the reception port is DEASSOC and vsiState of any other port is DEASSOC, the bridge forwards Pre-Associate TLV because the vsiState of the corresponding port of the upper switch is DEASSOC as described in the case P7.

When the forwarding decision of VDP TLV is 'yes', VDP state machine (bridge side) of the VSIID goes to the next state and VDP TLV is forwarded to the uplink port related to the downlink port. It waits for the response from the upper bridge. Upon the reception of the response, it checks an error in TLV Type and Reason. If an error occurs, it cancels the action on the VDP state machine and goes to the previous state. Then it sends the response to the server.

When the forwarding decision of VDP TLV is 'no', VDP state machine (bridge side) of the VSIID goes to the next state and the response is returned to the server for VDP TLV.

TABLE II. Forwarding Conditions of VDP Messages

(a) Pre-Associate

Case	**vsiState of Adjacent Switch**		**Forwarding**	**Note**	**Cf. vsiState of Upper Switch Corresponding Port**
	VDP Reception Port	**Other than Reception Port**			
P1	PREASSOC	DEASSOC	Yes	Keep Alive	PREASSOC
P2		PREASSOC	Yes	Keep Alive	PREASSOC
P3		ASSOC			ASSOC
P4	ASSOC	DEASSOC	Yes		ASSOC
P5		PREASSOC	Yes		ASSOC
P6		ASSOC			ASSOC
P7	DEASSOC	DEASSOC	Yes		DEASSOC
P8		PREASSOC	(Yes)		PREASSOC
P9		ASSOC			ASSOC

(b) Associate

Case	**vsiState of Adjacent Switch**		**Forwarding**	**Note**	**Cf. vsiState of Upper Switch Corresponding Port**
	VDP Reception Port	**Other than Reception Port**			
A1	PREASSOC	DEASSOC	Yes		PREASSOC
A2		PREASSOC	Yes		PREASSOC
A3		ASSOC	(Yes)		ASSOC
A4	ASSOC	DEASSOC	Yes	Keep Alive	ASSOC
A5		PREASSOC	Yes	Keep Alive	ASSOC
A6		ASSOC	Yes	Keep Alive	ASSOC
A7	DEASSOC	DEASSOC	Yes		DEASSOC
A8		PREASSOC	Yes		PREASSOC
A9		ASSOC	(Yes)		ASSOC

(c) De-associate

Case	**vsiState of Adjacent Switch**		**Forwarding**	**Note**	**Cf. vsiState of Upper Switch Corresponding Port**
	VDP Reception Port	**Other than Reception Port**			
D1	PREASSOC	DEASSOC	Yes		PREASSOC
D2		PREASSOC			PREASSOC
D3		ASSOC			ASSOC
D4	ASSOC	DEASSOC	Yes		ASSOC
D5		PREASSOC			ASSOC
D6		ASSOC			ASSOC
D7	DEASSOC	DEASSOC	(Yes)		DEASSOC
D8		PREASSOC			PREASSOC
D9		ASSOC			ASSOC

B. Examples of VM Migration to another Rack

Figure 4 shows an example of VDP messages communicated between a server and an adjacent bridge when a virtual machine is migrated to another server in another rack via multiple ToR switches in the path. When a migration is initiated, Pre-Associate message is sent from the destination server to the adjacent bridge at a destination port. This is a case of P7 described above and Pre-Associate TLV is forwarded to the upper ToR switch.

During the stop and copy phase, Associate message is sent from the destination server to the adjacent bridge at the destination port. This is the case of A1 and the adjacent bridge forwards Associate TLV to the upper ToR switch. When a response is returned from the upper switch, the adjacent bridge returns a response to the destination server.

On the other hand, De-associate message is sent from the source server to the adjacent bridge at the source port. This is the case of D4 and the bridge forwards De-associate message to the upper ToR switch.

In this way, the port profile is moved together with the VM and the physical switch is automatically configured.

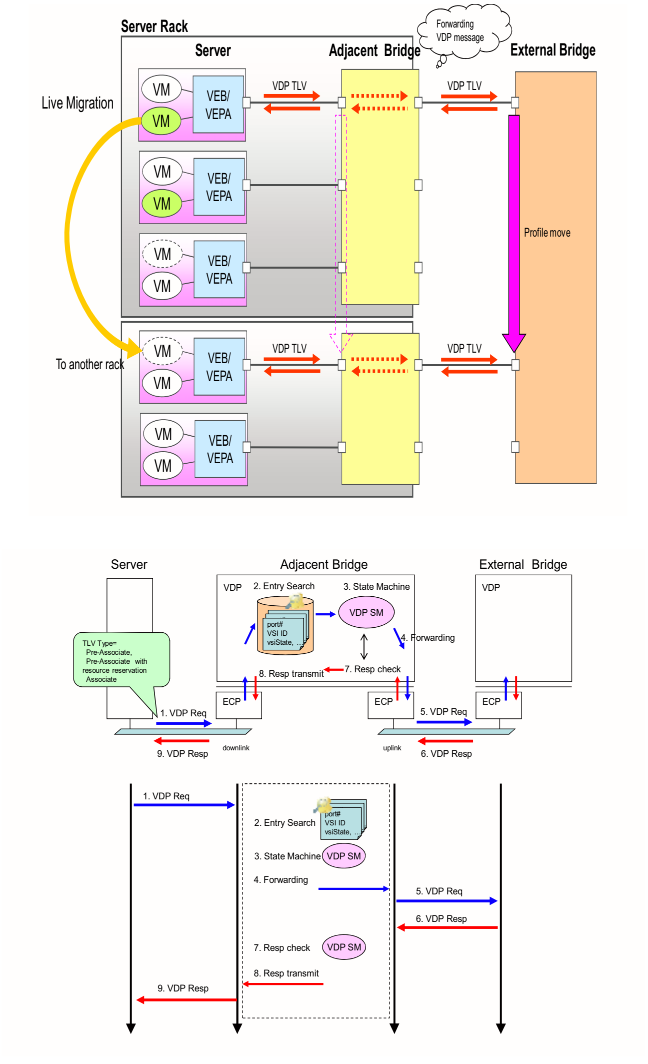


Figure 4. Inter-rack VM Migration and VDP Messages

C. Examples of VM Migration within Rack

Figure 5 shows an example of VDP messages communicated between a server and an adjacent bridge when a virtual machine is migrated to a server in the rack via a ToR switch. When a migration is initiated, Pre-Associate message is sent from the destination server to the adjacent bridge at the destination port. This is the case of P9 where vsiState of the reception port (destination port) is DEASSOC and vsiState of another port (source port) is ASSOC, and the bridge does not forward Pre-Associate TLV because the vsiState of the corresponding port of the upper ToR switch is ASSOC.

During the stop and copy phase, Associate message is sent from the destination server to the adjacent bridge at the destination port. This is the case of A3 and the adjacent bridge does not forward an unnecessary Associate TLV to the upper ToR switch because vsiState of the upper switch is ASSOC although forwarding of Associate TLV is acceptable.

On the other hand, De-associate message is sent from the source server to the adjacent bridge at the source port. This is the case of D6 and the bridge does not forward De-associate message to the upper ToR switch. In this case, forwarding of De-associate TLV is unacceptable because if De-associate message is forwarded, the associate on the destination port is removed while VM is running.

In this way, the port profile is moved together with the VM and the physical switch is automatically configured.

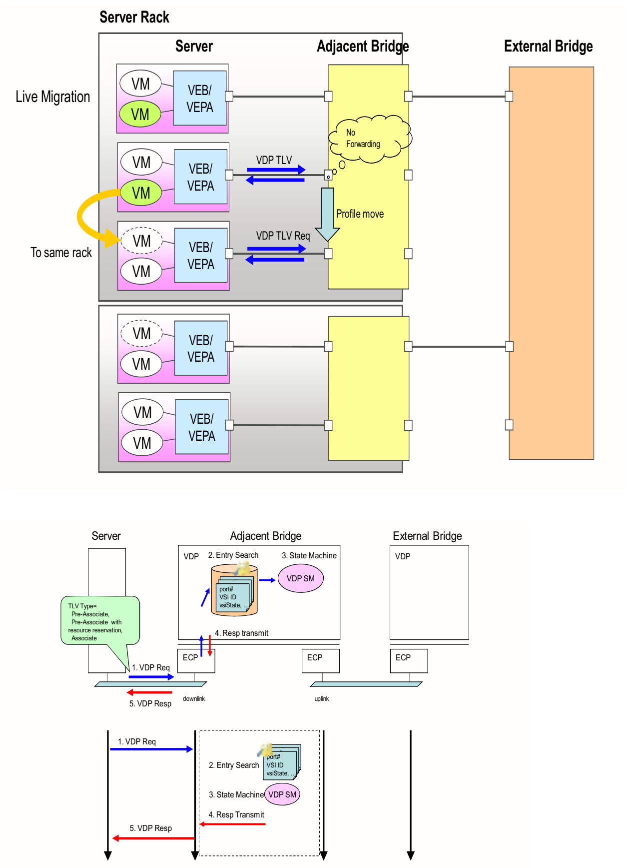


Figure 5. Intra-rack VM Migration and VDP Messages

D. Topology for AMPP Extension

In the AMPP extension, we forward VDP messages to an upper-level switch. In other words, AMPP extension works in a topology like a tree structure where we can decide an uplink to which we forward VDP messages. Our switch blade has a port grouping feature in Intelligent Blade Panel (IBP) firmware mode and the port group has only one active uplink for downlink ports [11].

IV. PROTOTYPE

We developed a prototype of the AMPP extension. The prototype consists of Port Profile Engine for AMPP with an extension for multi-level switches, EVB Packet Analyzer with Protocol Checker and Visualization Tool of EVB messages.

A. Port Profile Engine

Figure 6 shows the module structure of Port Profile Engine. Port Profile Engine consists of EVB, VDP and ECP daemons. EVB daemon is a master daemon and initiates VDP and ECP daemons. EVB daemon communicates with an adjacent bridge using EVB TLV in LLDP and configures Edge Relay parameters. VDP daemon communicates with ECP daemon using TCP socket and ECP daemon communicates with Layer 2 protocol daemon using Layer 2 socket. ECP daemon guarantees that VDP TLV in ECP packet is received by the adjacent bridge using L-ACK.

TABLE III shows the list of processing in VDP daemon. VDP daemon keeps VDP Instance with related information including VSI Instance ID (VSIID), port profile ID (VSI Type ID), VDP state machine status (vsiStatus) and bridge port number.

In this prototype, we implemented Port Profile Engine as a program running in a remote server connected to the switch. LLDP packet with EVB TLV and ECP packet with VDP TLV are forwarded to the remote server and they are processed by Port Profile Engine. Port Profile Engine specifies a designation port for the packet to be transmitted. And the switch sends out the packet on the designated port.

Port profile engine will be implemented in the switch firmware for the management when the IEEE 802.1Qbg standard completes.

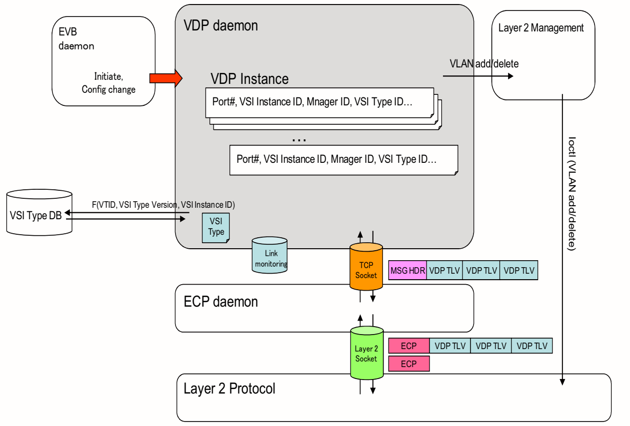


Figure 6. Module Configuration of Port Prfile Engine

TABLE III. Procesing in VDP Daemon

Processing	Description
VDP Initialization	Initialize VDP Data Transmit/Receive Interface to/from ECP module
VDP Data Reception	Receive VDP data in receeived ECP from ECP module
VDP Instance Search	Search VDP Instance using VSIID in VDP TLV
VDP State Machine	VDP State Machine
VDP Forwarding	When VDP forwarding mode enabled, forward VDP TLV to upper bridge if necessary
VDP Data Transmission	Transmit VDP data to ECP module for transmission
VDP Disconnect	Disconnect VDP Data Transmit/Receive Interface to/from ECP module

B. *EVB Packet Analyzer with Protocol Checker*

Figure 7 shows configuration of EVB packet analyzer and a visualization tool. TABLE IV shows features of EVB Packet Analyzer. EVB Packet Analyzer is a standalone program and captures LLDP / ECP Packets and analyzes EVB TLV in LLDP packets and VDP TLV in ECP packets. The analysis results are displayed on the window of EVB Packet Analyzer with packet filtering applied if filtering is configured. The analysis results can be output to Visualization Tool described later. It also has protocol checking capability and checks ECP protocol and VDP protocol if it is enabled.

Through the prototyping, we noticed that we cannot distinguish VDP request from VDP response with success by just looking at VDP TLV. To distinguish request from response, we looked at a control tag by which Port Profile Engine designates an output port for an outgoing frame. From EVB Packet Analyzer point of view, it is better to have an indication on the VDP TLV.

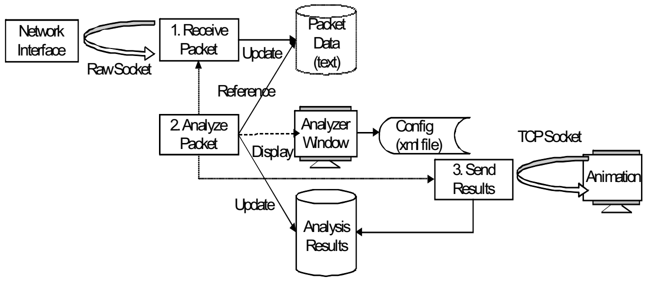


Figure 7. EVB Packet Analyzer and Visualization Tool

TABLE IV. Features of EVB Packet Analyzer

Features	Description
Packet Reception	Receive ECP and LLDP packets and save packet data
Packet Analysis	Analyze VDP TLV and EVB TLV and save the analysis results.
Packet Filter	Filter packet based on the filter setting
Display of Analysis Results	Display analysis result with filtering applied on the analyzer window
Output of Analysis Results for Visualization	Output analysis results to visualization tool for the animation
ECP Protocol Checker	Check transition of ECP state machine based on packet analysis on ECP Packet
VDP Protocol Checker	Check transition of VDP state machine based on packet analysis on VDP TLV

C. *Visualization Tool of VDP Messages*

The visualization tool inputs results of analysis from EVB Packet Analyzer and generates animation of VDP messages in VM migration based on the inputs. The visualization tool of VDP messages is used to explain the protocol behavior intuitively in a demonstration.

V. EVALUATION

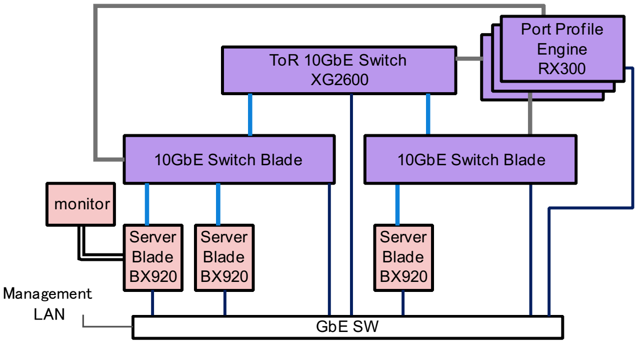
We evaluate our prototype of AMPP extension in a multi-level switch configuration. For the evaluation system, we use a blade server and a ToR switch where a multi-level switch configuration is typical and VM migration of inter-chassis always requires configuration changes both in switch blade and ToR switch.

A. *Evaluation System*

Figure 8 shows an evaluation system for AMPP extension prototype. The evaluation system mainly consists of three BX920 server blades equipped with 10GbE NICs and two 10GbE switch blades, one 10GbE ToR switch XG2600 and RX300 rack servers [13], [14].

Port Profile Engine is running on RX300 and processes EVB packets coming into the switch it manages. The switch blade works as an adjacent bridge with AMPP extension enabled. EVB Packet Analyzer with Protocol Checker and Visualization Tool are also running on one of RX300s and EVB Packet Analyzer captures and analyzes mirrored control packets. Visualization Tool generates animation of VDP messages from the analysis results.

Linux LLDP agent with IEEE 802.1Qbg patch [15] is running on the BX920 server blade. We used the agent on the server side to make sure interoperability with our prototype on the switch side. The LLDP agent advertises EVB capabilities on EVB TLV in LLDP packet and supports ECP protocol to carry VDP TLV in ECP packet.



(a) System Configuration

(b) ToR 10GbE Switch XG2600 (c) BX900 10GbE Switch Blade

Figure 8. Evaluation System

B. Protocol Operation

In the first evaluation, we evaluate EVB capabilities exchanges using EVB TLV in LLDP. The switch blade communicates with a station and works as an adjacent bridge from the station point of view. Also the switch blade communicates with ToR switch and works as a station from ToR switch point of view. The conversion of Edge Relay Mode from VEPA to VEB is recognized by ToR switch correctly.

In the second evaluations, we evaluate VDP protocol for port profile movement along with VM migration. Figure 9 shows captured packets by EVB Packet Analyzer and visualizations of VDP messages in inter-chassis VM migration by Visualization Tool. In inter-chassis VM migration, VDP TLVs from a station are forwarded to ToR switch and the port profile is moved both in switch blades and ToR switch along with VM migration. In intra-chassis VM migration, VDP TLVs from a station is not forwarded to ToR switch and the port profile is moved within the switch blade only.

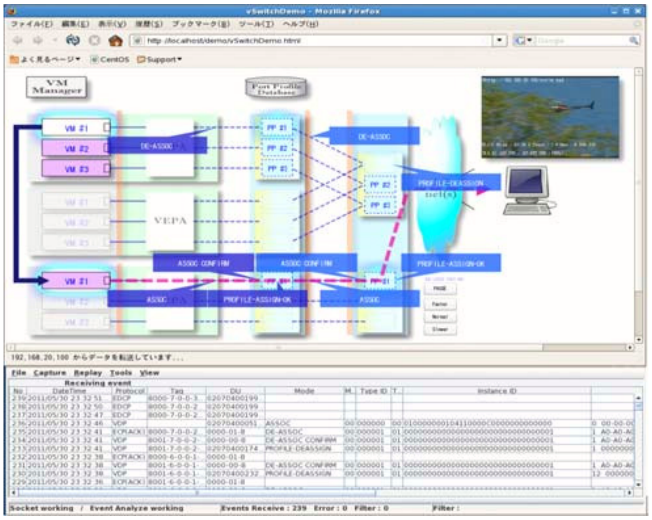


Figure 9. Visualization of VDP Messages in VM Migration

C. Scalability

One concern regarding AMPP extension is scalability of Port Profile Engines in a large-scale multi-level switch configuration. Since we do not have a prototype at scale, we use measurements from our existing system to project the requirements of large systems. Figure 10 shows the amount of VDP messages Port Profile Engines (PPE) expected to process at a switch stage in multi-level switch configuration with locality as a parameter. PPE-100 means 100% of locality where VDP messages are processed in the first stage only. PPE-75 means 75% of VDP messages are processed in the first stage locally and 25% of messages are forwarded to an upper switch, and so on. For a comparison, it also shows the amount of VDP messages in a flat configuration using Port Extension in concept (PEC) where all messages are forwarded and processed in the most upper switch and have nothing to do with locality. Port Profile Engines need to handle less VDP messages in a large-scale multi-level switch configuration.

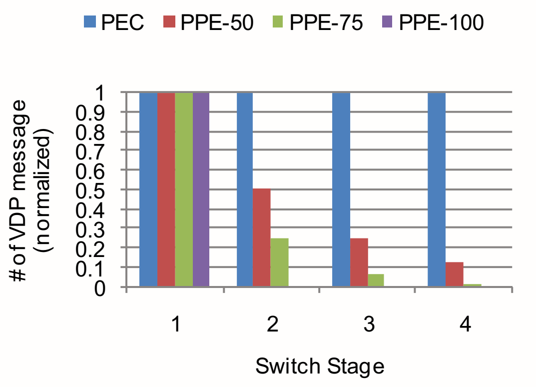


Figure 10. Amount of VDP Messages

VI. CONCLUSION

This paper describes an extension of AMPP for multi-level switches to reduce complexity and automate the network management in a Cloud Data Center where VM mobility is important for the dynamic resource allocation. We developed a prototype of the extension and confirmed a network state being moved in a multi-level switch configuration using the standard protocol as it is on the wire. We will continue developing new functions for dynamic infrastructure as part of its efforts to promote research and development of high-performance data center network.

ACKNOWLEDGMENT

We would like to thank Y. Mochizuki and M. Kawata at Fujitsu Advanced Engineering Limited, and N. Matsuoka and J. Tanaka at Fujitsu Laboratories Ltd. for their help and insightful comments. We also express our gratitude to T. Horie and A. Hattori at Fujitsu Laboratories Ltd. for their support of this project.

REFERENCES

[1] N. Carr, *The Big Switch: Rewiring the World, from Edison to Google.* New York, U.S.: W. W. Norton & Company, 2008, pp. 75-79.

[2] P. Dawson and T. Bittman, "Virtualization Changes Virtually Everything," *Gartner Special Report,* March 28, 2008.

[3] Cisco Systems, Inc., "Cisco VN-Link: Virtualization-Aware Networking, *White Paper*, 2009. Available:
http://www.cisco.com/en/US/solutions/collateral/ns340/ns517/ns224/ns892/ns894/white_paper_c11-525307.pdf

[4] BLADE Network technologies Inc., VMready. Available:
http://www.bladenetwork.net/vmready.html

[5] Arista Networks Inc., VM Tracer. Available:
http://www.aristanetworks.com/en/products/eos/vmtracer

[6] IEEE P802.1Qbg/D1.5 Draft Standard for Local and Metropolitan Area Networks-Virtual Bridged Local Area Networks -Amendment XX: Edge Virtual Bridging, IEEE Draft Standard 802.1Qbg, March 2011.

[7] R. Recio, S. Krishnasamy, and R. Sharma, "Ethernet Virtual Bridging Automation Use Cases," *DC CAVES workshop ITC 22,* September 2010.

[8] IEEE P802.1Qbh/D2.0 Draft Standard for Local and Metropolitan Area Networks-Virtual Bridged Local Area Networks -Amendment: Bridge Port Extension, IEEE Draft Standard 802.1Qbh, April 2011.

[9] DMTF, "Virtual Networking Management white Paper," *DSP2025,* February 2011.

[10] H. Shah, "Management Standards for Edge Virtual Bridging (EVB) and Port Profiles", Management Developers Conference 2010, November 2010.

[11] 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," *GreenCom 2010,* December 2010.

[12] "Fujitsu Develops World's First Technology Employing 10 Gbps Virtual Switch to Substitute for On-Server Virtual Switch Functions," Available:
http://www.fujitsu.com/global/news/pr/archives/month/2010/20100610-02.html, June 10, 2010.

[13] "PRIMERGY BX900 S1, State of the Art Blade Server Chassis – the Dynamic Cube", *Fujitsu PRIMERGY BX900 Data Sheet,* 2009.

[14] "10GbE Ultra-low Latency Switch XG2600", *Fujitsu XG2600 Data Sheet,* 2010.

[15] open-lldp lldpad (LLDP Agent Daemon). Available:
http://www.open-lldp.org/git/?p=lldp/open-lldp.git;a=summary