Investigating the Performance of Link Aggregation on OpenFlow Switches

Toan Nguyen-Duc[1(✉)], Hoang Tran-Viet[1], Kien Nguyen[1,2], Quang Tran Minh[2], Son Hong Ngo[1], and Shigeki Yamada[2]

[1] Hanoi University of Science and Technology, 1 Dai-Co-Viet Street, Hanoi, Vietnam
{toan.nguyenduc1,hoang.tranviet}@hust.edu.vn, sonnh@soict.hust.edu.vn

[2] National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
{kiennq,quangtran,shigeki}@nii.ac.jp

Abstract. In this paper, we extensively explore the operation of Link Aggregation (LA) on OpenFlow switches in comparison to the LA in conventional switches. The comparison of two LA implementations has been conducted in a real testbed under the UDP and TCP traffic loads. The testbed includes Pica8 P-3925 switches, which support two modes: an OpenFlow switch (i.e., using Open vSwitch) and a conventional switch (i.e., using the operating system called XorPlus). The evaluation results show that two LA implementations achieve similar performance in improving throughput. However, the XorPlus implementation provides a better resilience than the other. Specifically, the LA implementation on XorPlus spends less than 1.49538 seconds to switch the TCP traffic on the faulty link to the other links of a Link Aggregation Group (LAG) while the switchover time is four times longer on the Open vSwitch. In the case of UDP traffic, the maximum switchover time on the Open vSwitch is twice the one on XorPlus.

Keywords: Link aggregation · Resilient · Aggregation bandwidth · Openflow switch · Evaluation

1 Introduction

Link Aggregation refers to the capability of combining multiple physical cables into a logical link. The standardized link aggregation appears in IEEE 802.1AX [1], and is widely used for connecting pairs of networking devices. The link aggregation is prevalent because it provides a cost-effective way to improve bandwidth by simply adding new links alongside the existing ones instead of replacing the existing equipment with a higher-capacity link. For example, a 40Gbps aggregated bandwidth link is formed by aggregating four cables with capacity 10Gbps

V.C.M. Leung et al. (Eds.): TridentCom 2014, LNICST 137, pp. 194–202, 2014.
DOI: 10.1007/978-3-319-13326-3_19

each. Moreover, the link aggregation is also necessary since it increases the network resilience. When a physical cable of the logical link fails, the logical one continues to carry traffic over the remaining cables. Therefore, many network vendors have introduced LA supported in their hardware and software products. However, each vendor has its own commercial solutions which are closed to networking researchers. The same problem occurs in both the traditional manufacturers as well as the ones in the fast growing OpenFlow community.

The OpenFlow technology has been emerging with the concept of software defined networking, which provides innovation and flexibility in network operations and managements [2]. One of key features of the technology is the OpenFlow switch, whose specification is frequently updated by Open Networking Foundation. The switch is an extension of Ethernet switch, which also uses one or several internal forwarding tables. However, the switch has an extra interface, which is used to receive the control instructions from outside. The decoupling design of OpenFlow switch not only increases advanced functionalities but also reduces the cost and complexity of networking hardware. The OpenFlow switch was first deployed in an academic campus network [3], and the OpenFlow features have been supported by many networking vendors. More importantly, the OpenFlow has been successfully deployed in several production networks, such as Google WAN globally interconnecting datacenters [4]. However, several basic but important technologies such as Link Aggregation are recently added to the specification of OpenFlow Switch (version 1.1). Hence, it is necessary to extensively evaluate the technologies' performance in order to support the increasing deployments of OpenFlow.

There exist several related works on evaluations of OpenFlow switches [5, 6]. However, the works mainly focused on evaluating several basic networking parameters on data or control planes such as the throughput and latency. Besides that, there is also an investigation on the performance of specific OpenFlow hardware [7] targeting the scalability of OpenFlow switches. Different to other works, we investigate the operation of LA in OpenFlow switches, and compare its performance with the LA in conventional switches. The comparison of two LA implementations is conducted in a real testbed using Pica8 P-3295 switches [8] with UDP and TCP traffic. The switch supports two modes: an OpenFlow switch and a conventional switch. The mode OpenFlow switch is an open source Open vSwitch with the latest stable version 1.10.0 [9,10]. The mode conventional switch uses Pica8 operating system called Xorplus version 2.0.4. Our aim is to debug the traffic behaviours on the logical links of LA as well as investigate the benefits of LA on the network (e.g., increasing throughput, resilience, etc.). In term of network resilience, we measure the switchover time which is the duration of switching the traffic on the faulty link over the other links in a LAG.

The remainder of paper is organized as follows. In Section 2 we describe the theoretical background of our evaluation including basics of link aggregation, OpenFlow switch, and port mirroring. In Section 3, we present the LA evaluations and results. Finally, we conclude our work in section 4.

2 Theoretical Background

2.1 Link Aggregation

The first standard of Link Aggregation (LA) was introduced in the IEEE 802.3ad [11] in 2000. In 2008, LA was removed from the IEEE 802.3 and added to the IEEE Std 802.1AX-2008. LA is commonly used to connect pairs of networking devices (i.e., switches, routers, etc.), aiming to provide greater bandwidth at the network core. For example, LA which is made up of four links with capacity 10 Gbps each, can carry 40 Gbps. However, the traffic must be a mixture of connections since LA needs to keep the order of the packets. Moreover, LA potentially achieves resilience against link breakage since the failure only affects the carrying traffic that is automatically switched to the others if the connectivity still exists.

LA uses LAG to control static local information and the LA-related information must be configured (e.g., ports on networking devices). Each port has a Port ID and an operational key when it is belong to a LAG. The Port ID is combined with the system ID and the operational key to construct a LAG ID. On the transmitter, LA uses an Aggregator to distribute frame transmissions from a client to the appropriate ports in a LAG. On the receiver, the Aggregator collects received frames from the ports and pass them to the client transparently as shown in Fig. 1. Therefore, the receivers see all links in a LAG as a single logical link. The Aggregator also decides if a new link can join a LAG by comparing its operational Key to the operational Key of the port to which the link connect. Once the local device and its peer agree on the LAG, frames are distributed and collected on the aggregated link. LA uses Link Aggregation Control Protocol (LACP) for dynamic information exchanged between two LA-supported devices. The devices commonly referred to respectively as the "ACTOR" and the "PARTNER". One simple illustration of the communication is shown in

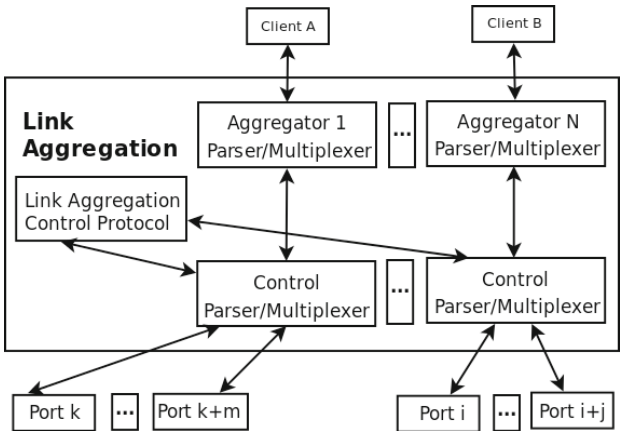

Fig. 1. Link Aggregation 802.1AX block diagram

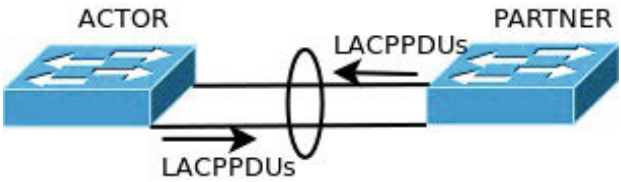

Fig. 2. Dynamic Link Aggregation

Fig. 2. Typically, LACP performs a number of tasks to support the communication between the devices. First, after a LAG is created on both devices, when cables are plugged in to the ports in a LAG, the ACTOR exchanges messages, such as protocol data units (PDUs) with the PARTNER. The messages allow LACP to determine whether or not both peers have the same system ID or have the same speed. If they do, the Collecting and Distributing flags in LACP PDUs are set, the aggregated link is capable of transmitting and receiving traffic. In the second step, LACP monitors the status of individual links to ensure their membership between ACTOR and PARTNER is still valid. In order to perform the monitoring, a periodic timer on the ACTOR will trigger transmission of PDUs to the PARTNER and vice versa. In a failure scenario, three LACPPDUs are exchanged without dependence on timer value. Consequently, the configuration resolves quickly to a stable configuration. In the third and last step, LACP controls the addition of links to the existing LAG as well as removes down links from the group.

2.2 OpenFlow Switch

An OpenFlow network consists mainly of OpenFlow switch(es) and OpenFlow controller(s). Essentially, the control of a switch, such as packet routing, is moved to the OpenFlow controller so that the controller administrator has full control over the switch. The controller interacts with the switch by using OpenFlow protocol. The function of the protocol is to install, modify or remove entries on a flow table of the switch. Each entry contains a flow description and a list of actions associated with that flow. When a packet reaches an OpenFlow switch, the switch extracts the packet header (e.g., MAC addresses, IP addresses, and TCP/UDP ports) and compares this information to the flow description of the flow table entries. If a matching entry is found, an action is applied to the packet. For example, the packet may be dropped or forwarded to one or more OpenFlow switch ports. If a matching is not found, the switch will ask the controller for the action that should be applied to all packets from the same flow. The decision is then sent to the switch and saved as an entry in the switch's flow table. The next incoming packets that belong to the same flow are then forwarded through the switch without referring to the controller. The earlier versions of OpenFlow Switch Specifications did not provide LA because their forwarding model simply supported drop, forwarding, and flood packets. Fortunately, since the version 1.1 of OpenFlow Switch specification, the concept of virtual port has been added to

reuse the existing physical ports interface for additional functions like tunneling, and specially link aggregation, etc.

2.3 Port Mirroring

It is not easy to capture the traffic behaviors on individual links in a LAG because all the links are treated as a single one. We find that in order to obtain the behaviors, the technique named port mirroring [12] is extremely useful. The port mirroring is a fundamental option on packet switches and it is configurable remotely. The purpose of technique is to copy all incoming/outgoing packets on a switch port called mirrored port to another port called mirroring port. By doing so, we can monitor the traffic on the mirrored port by placing a packet capturing tool on the mirroring port. Moreover, the technique is also benefit for the evaluation of system resilience as later mention in Sections 3.

3 Evaluation

The testbed in our evaluation consists of two switches and three hosts as illustrated in Fig. 3. As mentioned earlier, the switches run either Open vSwitch or XorPlus. The switches are wired using two cables with capability 1Gbps each, which are formed a LAG. The steps of creating the LAG on the two OpenFlow swiches are implemented mostly following the two manuals of Pica switches [13,14]. In order to effectively collect the results, three Linux hosts, which are equipped with Intel Core I5, 4GB RAM, and Ubuntu 12.04 LTS 64 bit, are used as a traffic generator, receiver, and monitor. On the right side of topology (Fig. 3), the hosts H2 and H3 are attached to the switch SW2, through GE cables. On the left side, the host H1 has four 1Gbps networking cards shown as eth0, eth1, eth2, eth3, respectively. The eth2's duty is to receive the traffic from H2 and H3. The eth0 and eth1 are used to monitor the traffic on each member of the LAG by using Port Mirroring. In the evaluation, we generate the traffic from H2 and H3 to H1 using Iperf [15]. We capture the receiving traffic on the observed hosts using the tool named Bwm-ng [16].

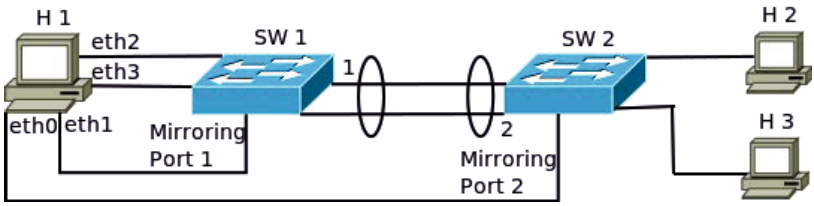

Fig. 3. Evaluation Testbed

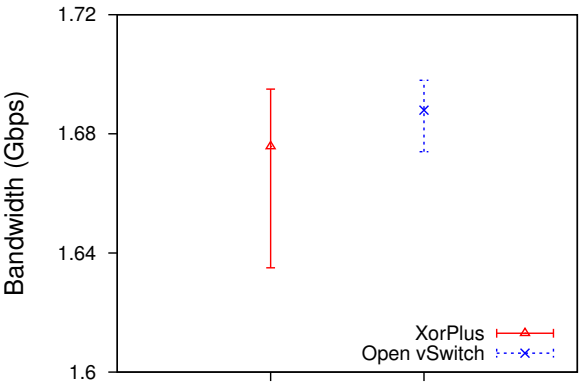

Fig. 4. TCP-related aggregation bandwidth measurement

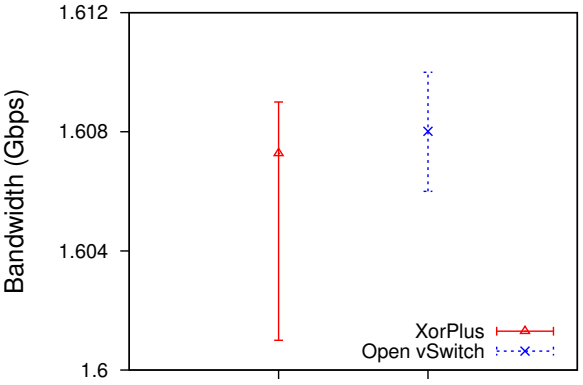

Fig. 5. UDP-related aggregation bandwidth measurement

3.1 Evaluating the Aggregation Bandwidth

The measurement procedure with TCP traffic is as follows. Initially, an Iperf server is started on H1 listening for incoming requests. Then, an Iperf client is enabled on H2, that attempts to establish communication with the Iperf server. After 5 seconds, another Iperf client, which share the same destination with the one on H2, is triggered on the host H3. The Iperf clients will be stopped after 90 seconds. We run the procedure 50 times on the two types of switches and the results for Open vSwitch and Xorplus are shown in Fig. 4. Figure 4 shows that the maximum TCP bandwidth of the LAG is about 1700 Mbps on Open vSwitch. This is only marginally higher than the maximum bandwidth of the LAG on XorPLus. We have also repeated the same procedure for the UDP traffic. The UDP bandwidth results are shown in Fig. 5. Similar to TCP traffic,

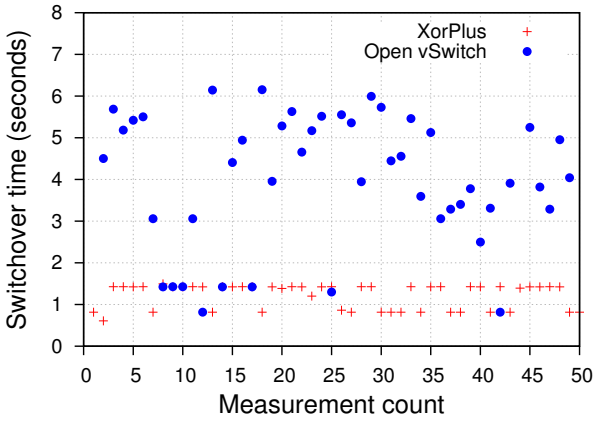

Fig. 6. TCP-related switchover time measurement results

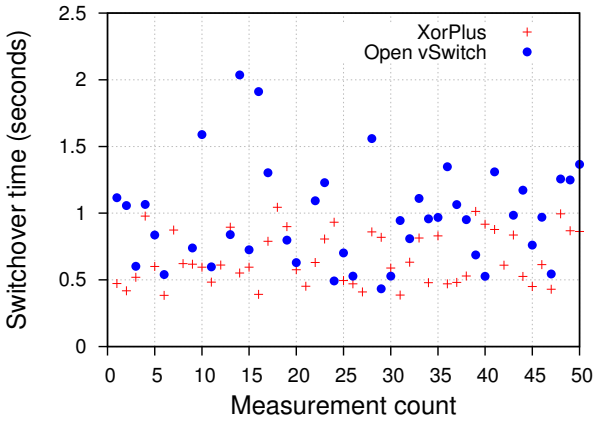

Fig. 7. UDP-related switchover time measurement results

the maximum UDP bandwidth of the LAG is 1600 Mbps and it is slightly higher bandwidth comparing to the other.

3.2 Evaluating the System Resilience

In order to evaluate how LA improves the system resilience, we observe the network performance when link failures occur. The focusing parameter is the switchover time, which is defined as the duration of switching the traffic from a faulty link to another aliveness link. Specifically, the switchover time t is the difference between the timestamp t1 carried by the last packet on the faulty link and the timestamp t2 on the first packet that goes over the alive one. The shorter switchover time, the fewer packet loss, and hence the shorter one leads

to a better system resilience. In this evaluation, we use the same scenario as in Figure 3. The host H1 has three 1Gbps NICs called eth0, eth1 and eth2. The NICs are used to capture traffic on a computer aiming to avoid the problem of clock synchronization on different machines. As shown in the figure, Port 1 of SW1 connected to the first link of the LAG and Port 2 of SW 2 connected to the second one. SW1 was configured to copy the traffic on port 1 to another port. This port was then connected to H1's eth1. Similarity, the traffic on port 2 of SW2 was mirrored to another port which was connected to H1's eth0. As a result, H1 can monitor the network traffic on each link of the LAG. Tcpdump[17] was used to capture the details of packets being received over H1's eth0 and eth1.

In this evaluation, we also use both UDP and TCP traffic on the two types of switches. Each experiment has been repeated 50 times with random link failures, and the values of switchover time are presented in Fig. 6 and Fig. 7. Figure 6 shows that in the cases of TCP traffic the maximum switchover time on Open vSwitch is 6.15464 seconds, which is nearly four times longer than the one on Xorplus. However, the minimum switchover time on two types of switches is almost similar (0.815891 seconds compare to 0.607892 seconds). Figure 7 shows the UDP traffic switchover time on the switches. We can see the Open vSwitch spends maximum of 2.036 seconds to switch UDP traffic, which is twice longer than that time on Xorplus. Hence we confirms that LA enhances the system resilience since it can automatically switch the traffic. Besides that, the evaluation results also show that among two switches the conventional switch has a better resilience performance in term of switchover time.

4 Conclusion

We have evaluated the LA performance on OpenFlow switches and compared it against the one on conventional switches. The evaluation results confirm that the LA can spread the traffic load to all links in a LAG. Consequently, the aggregation bandwidth is increased linearly with the number of the aggregated links. Comparing the LA performance on the two types of switches, we found that the LA on OpenFlow switch provides a slightly lower throughput than the other. We also observed that LA enhances the system resilience since the capability of automatic switching the traffic on the faulty link to the aliveness links lets the LA keeps the logical link alive. Moreover, the LA implementation on XorPlus spends maximum of 1.49538 second to switch a TCP flow in the failure scenario. This period is four times faster than the one on OpenFlow switch. In the case of UDP traffic, it takes less than 2.036s for the LA implementation on the OpenFlow switch, but still twice longer than the LA on the conventional switches. We conclude that LA on Open vSwitch can be an alternative to the one on the conventional switch in improving throughput. However, the switches need to be furthermore optimized to achieve the equivalent performance in increasing system resilience.

In the future, we plan to extend the investigation on a LAG made up of more links. Moreover, we also plan to investigate the paths form by several

LAGs through multiple switches (i.e., multi-hop). Besides that, we are going to study the behavior and performance of LA under more complicated scenarios such as high throughput, or delay sensitive environments. Finally, we will explore the Fast Failover function of OpenFlow and compare the values of switchover time achieved by the Fast Failover and the one by LA.

References

1. IEEE Std 802.1AX-2008 (2008). http://standards.ieee.org/findstds/standard/802.1AX-2008.html
2. Lara, A., Kolasani, A., Ramamurthy, B.: Network innovation using openflow: A survey. IEEE Trans. Communications Surveys Tutorials **16**(1), 493–512 (2014)
3. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J.: Openflow: enabling innovation in campus networks. SIGCOMM Comput. Commun. Rev. 38(2) (2008)
4. Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Venkata, S., Wanderer, J., Zhou, J., Zhu, M., Zolla, J., Hölzle, U., Stuart, S., Vahdat, A.: B4: experience with a globally-deployed software defined wan. In: Proc. ACM SIGCOMM 2013, pp. 3–14 (2013)
5. Mateo, M.P.: OpenFlow Switching Performance. Master's thesis, Politecnico di Torino (July 2009). http://www.openflow.org/downloads/technicalreports/MS_Thesis_Polito_2009_Manuel_Palacin_OpenFlow.pdf
6. Bianco, A., Birke, R., Giraudo, L., Palacin, M.: Openflow switching: Data plane performance. In: Proc. IEEE International Conference on Communications (ICC), pp. 1–5 (2010)
7. PPELMAN, M.A.: Performance Analysis of OpenFlow Hardware. Master's thesis, University of Amsterdam (December 2012). http://www.delaat.net/rp/2011-2012/p18/report.pdf
8. Pica8 P-3295 Switch specification. http://www.pica8.org/document/pica8-datasheet-48x1gbe-p3290-p3295.pdf
9. Open vSwitch. http://openvswitch.org/
10. Pfaff, B., Pettit, J., Koponen, T., Amidon, K., Casado, M., Shenker, S.: Extending networking into the virtualization layer. In: Proc. of Workshop on Hot Topics in Networks (HotNets-VIII) (2009)
11. IEEE 802.3ad Link Aggregation Task Force (2000). http://www.ieee802.org/3/ad/
12. Zhang, J., Moore, A.: Traffic trace artifacts due to monitoring via port mirroring. In: Proc. IEEE End-to-End Monitoring Techniques and Services 2007, pp. 1–8 (2007)
13. PicOS 2.0.1 L2/L3 Configuration Guide. http://www.pica8.org/document/picos-2.0.1-l2-l3-configuration-guide.pdf
14. PicOS 2.0.1 OVS Configuration Guide. http://www.pica8.org/document/picos-2.0.1-ovs-configuration-guide.pdf
15. Iperf. http://iperf.sourceforge.net/
16. Bwm-ng. http://sourceforge.net/projects/bwmng/
17. Tcpump. http://www.tcpdump.org/