Research Article

Analytical Performance Evaluation of Different Switch Solutions

Francisco Sans¹ and Eric Gamess²

¹ *Universidad Central de Venezuela, Laboratorio de Comunicación y Redes, Los Chaguaramos, Caracas, Venezuela*

² *Universidad Central de Venezuela, Escuela de Computación, Los Chaguaramos, Caracas, Venezuela*

Correspondence should be addressed to Eric Gamess; egamess@gmail.com

Received 19 July 2013; Revised 27 October 2013; Accepted 28 October 2013

Academic Editor: Liansheng Tan

Copyright © 2013 F. Sans and E. Gamess. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The virtualization of the network access layer has opened new doors in how we perceive networks. With this virtualization of the network, it is possible to transform a regular PC with several network interface cards into a switch. PC-based switches are becoming an alternative to off-the-shelf switches, since they are cheaper. For this reason, it is important to evaluate the performance of PC-based switches. In this paper, we present a performance evaluation of two PC-based switches, using Open vSwitch and LiSA, and compare their performance with an off-the-shelf Cisco switch. The RTT, throughput, and fairness for UDP are measured for both Ethernet and Fast Ethernet technologies. From this research, we can conclude that the Cisco switch presents the best performance, and both PC-based switches have similar performance. Between Open vSwitch and LiSA, Open vSwitch represents a better choice since it has more features and is currently actively developed.

1. Introduction

Virtualization is a concept that in recent years has gained a huge importance. From computing's perspective, virtualization means that computing tasks or programs are run under a virtual environment rather than in a physical hardware. With this concept, one physical component can be logically seen as multiple virtual components. In this way, operators gain great flexibility and abstraction of the underlying physical infrastructure.

One of the major uses of virtualization is the creation of several virtual machines (VMs) in one physical machine also known as virtual desktop infrastructure (VDI). This is done generally in datacenters [1], which provide several services in different VMs, within the same physical machine, allowing a quick provisioning, spilling-over to the cloud, and improving availability during periods of disaster recovery. With the growing adoption of virtualization, it was necessary to develop a new network access layer that provides inter-VM connectivity in the same way as the physical layer does. Hence, software to create virtual switches or virtual routers has recently gain importance. Another use of this software is that a common PC with multiple network interface cards (NICs) can act as a switch or a router.

In a moderate network topology, multiple switches and routers can be present. Obtaining these physical devices can be highly expensive. With the release of software to create virtual devices, a cheaper and more flexible solution has emerged, with the availability of PC-based switches and routers. In a demandant network environment, a device's performance can be decisive, and the choice of a good switch can be significant. Therefore, it is important to know the difference between the performance of a PC-based and an off-the-shelf switch.

In this paper, we propose testbeds and do a performance evaluation with these testbeds with two PC-based switches that are running adequate software: Open vSwitch and LiSA. Their performance is compared with the one of a Cisco switch. We use benchmarking tools to measure round trip time (RTT), throughput, and fairness for the different technologies and make an analytical comparison of the results.

The rest of this paper is organized as follows. Section 2 presents the related work. Section 3 contains a brief description of the switching solutions that we chose for our study. Information about the traffic generation tools and our testbeds is presented in Section 4. An analytical comparison of the obtained results is showed in Section 5. Finally, in Section 6 we discuss the conclusions and future work.

2. Related Work

The virtualization of switches or the transformation of a PC with multiple NICs into a switch has been possible just recently, with the development of the adequate software. Therefore, just a few works have been done in the area of evaluating the network performance of these switch solutions. Rendec et al. [2] presented a detailed architecture description of LiSA, highlighting the advantages of LiSA over the Linux Kernel modules. Then, they evaluated the performance of LiSA, measuring its packet switching capability and its transfer speed. In [3], Pfaff et al. started their work by presenting Open vSwitch. They did a design overview and exposed the different ways of use of this virtual switch. Furthermore, its performance was compared with the Linux bridge, measuring their effective throughput. Pettit et al. [4] extended the previous study over Open vSwitch. An overview of the virtual switch and its integration with a hypervisors as the XenServer was presented. Additionally, the Open vSwitch effective throughput is compared with the one of the virtual ethernet port aggregator (VEPA) [5]. He and Liang [6] also evaluated some Open vSwitch features such as security and network performance. The security evaluation was conducted to see if the virtual switch was able to isolate VM traffic using ping and ARP tests. For the network performance evaluation, they conducted a throughput and CPU usage comparison with and without the use of QoS and VLAN.

On the other hand, many works have been done to study the performance evaluation of PC-based routers. Gamess and Velásquez [7] made an analysis of the forwarding engine performance of PC-based routers using Solaris, Windows, and Debian for both IPv4 and IPv6. They configured testbeds to evaluate the throughput, the latency introduced by the PC-based router, and the packet loss in a mesh traffic network. Narayan et al. [8] did a similar work but with Fedora, Ubuntu, and Windows Server as operating systems. The throughput, delay, and jitter were measured for TCP and UDP in small testbeds.

Some other works are more general. For example, Chowdhury and Boutaba [9, 10] performed a research on network virtualization and identified areas that need further investigation.

Unlike the previous works, the main contribution of our work is the comparison of the performance evaluation between PC-based and off-the-shelf switches, a comparison which has not been done by previous works to the best of our knowledge.

3. Switching Solutions

3.1. Cisco Switch. Cisco is a manufacture that offers a wide range of network products. One of its most important products are switches. These off-the-shelf switches ease deployment of converged applications and adapt to changing business needs by providing configuration flexibility, support for converged network patterns, and automation of intelligent network services configurations.

In this work, we use a Cisco WS-C3750-24TS-E switch, belonging to the Cisco Catalyst 3750 series. This series of

switches provides L2 and L3 features with support for the IEEE 802.1q, VLAN trunking, inter-VLAN routing, ether-channels (also known as port aggregation or bonding), QoS, and many others more. They are made of 24 Ethernet ports that can function at 10 Mbps or 100 Mbps and 2 SFP-based Gigabit Ethernet ports. We installed Cisco IOS version 12.2(25)SEB2 in the switch and we used the Cisco proprietary command line interface (CLI) to configure it.

3.2. Open vSwitch. Open vSwitch (OVS) [3] is a multilayer virtual switch designed to be flexible, portable and to reside within a hypervisor or management domain, to provide connectivity between the virtual machines and the physical interfaces. It is usually implemented on Xen, XenServer, KVM, and VirtualBox. Due to its popularity, it is also being ported to non-Linux hypervisors and hardware switches.

It can operate as a basic L2 switch in a standalone configuration, supporting VLAN, SPAN, RSPAN, ACL, QoS policies, port bonding, trunking, GRE and IPsec, tunneling, and per-VM traffic policing. Flow visibility with NetFlow and sFlow is also provided. To support integration into virtual environments, OVS exports interfaces for manipulating the forwarding state and managing configuration state at runtime, allowing the specification on how packets are handled based on their L2, L3, and L4 headers.

3.3. LiSA. Linux Switching Appliance (LiSA) [2, 11] is an open-source project that aims for delivering a cheap and efficient solution to small-sized networks. It transforms a standard PC with several NICs into a L2/L3 switch. LiSA tries to resolve Linux VLAN scalability issues and its poor performance with broadcast packets, on both access and trunk ports.

In the Linux kernel, the Layer 2 implementation is not trivial due to the bridge module that is not easily extendible. LiSA uses a framework that hides all the networking internals of Linux and provides an API to configure the switch. The central point for control and management of the switch is a user space CLI, which simulates the IOS CLI of Cisco.

The main task of the module of the kernel is to implement the forwarding logic. In doing so, it has to interact with the forwarding table and the VLAN table of the operating system. To manage VLANs, IEEE 802.1q is supported. With these features, LiSA offers VLAN switching, VLAN tagging, and inter-VLAN routing.

4. Benchmarking Tools and Testbeds

In this section, we present the benchmarking tools selected for our study. We also propose some testbeds to conduct our experiments.

4.1. Test Method. To calculate the RTT, we chose a benchmark developed previously by this research group [12]. As stated in [12], one way delay (OWD) or RTT reported in many evaluation tools is not reliable since they are based on synchronized computers, which is difficult to achieve at the level of microseconds [13]. The key is to take all the timestamps

FIGURE 1: L3 Switch routing.

in the same computer. Our benchmark is based on the client/server model. Basically, a packet (IPv4 or IPv6) of a fixed length is exchanged between the client and the server a number of times (defined by the users). The benchmark takes a timestamp before and after the interchange. The difference of the timestamps is divided by the number of time the packet was sent and received to obtain an average of the RTT over the path and then again by 2 to get the average OWD.

For the throughput, we selected Iperf [14]. This benchmark sends traffic from the client to the server, which then calculates the number of bits per seconds received. In the client, a high bandwidth has to be specified to ensure that the tool saturates the network in order to obtain the maximum throughput. Even though Iperf can measure both TCP and UDP, the tool only allows to specify the bandwidth in UDP. Therefore, we only perform throughput tests for UDP in our experiments.

In our testbeds, the performance evaluation is done by sending packets, with the benchmarking tools mentioned previously, between groups of PCs.

4.2. Testbeds. We consider three testbeds to evaluate the performance of the different switch solutions: (1) L3 switch routing, (2) router on a stick, and (3) fairness. In these testbeds, we distribute up to four PCs connected to a switch with two VLANs.

4.2.1. L3 Switch Routing. Figure 1 shows the general configuration of this testbed. Four PCs are connected to a switch, which implements two VLANs (VLAN10 and VLAN20). A switch virtual interface (SVI) is created for each VLAN in the switch, with a corresponding IP address, allowing inter-VLAN communication. PC1 and PC2 are placed in VLAN10, while PC3 and PC4 are in VLAN20.

The instructions shown in Figure 2 are used to configure the Cisco switch. First, the IPv4 routing capability must be enabled (line 1). Then, the first two FastEthernet interfaces are assigned to VLAN10 and the following two are assigned to VLAN20 (line 3 to line 9). To allow inter-VLAN communication, two SVIs are created and an IP address is assigned to each one. The IP address of the SVI corresponding to VLAN10 is 10.0.0.254 (line 12) and the IP address of the SVI corresponding to VLAN20 is 20.0.0.254 (line 15).

```
(1) ip routing
(2)
(3) interface range fastEthernet 1/0/1-2
(4)   switchport access vlan 10
(5)   switchport mode access
(6)
(7) interface range fastEthernet 1/0/3-4
(8)   switchport access vlan 20
(9)   switchport mode access
(10)
(11) interface vlan 10
(12)   ip address 10.0.0.254 255.255.255.0
(13)
(14) interface vlan 20
(15)   ip address 20.0.0.254 255.255.255.0
```

FIGURE 2: Configuration of the Cisco switch for the L3 switch routing testbed.

```
(1) auto eth<number>
(2) iface eth<number> inet manual
(3)   up ip link set $IFACE up
```

FIGURE 3: File /etc/network/interfaces of OVS.

Figures 3 and 4 show some important instructions for the configuration of OVS. To set a physical interface as a port in the switch, the instructions depicted in Figure 3 must be added to the /etc/network/interfaces file, where *number* is the identifier of a physical interface. The configuration of the switch is shown in Figure 4. In line 1, a virtual switch called SW is created. From line 2 to line 5, all the physical interfaces are added to the switch within their corresponding VLAN. A SVI must be set up in the switch for every VLAN (line 7 and line 9). These SVIs are created as internal to be added as virtual interfaces in the operating system and an IP address is assigned, as specified in line 8 and line 10. Finally, the forwarding capability must be activated to allow the inter-VLAN communication (line 12).

The configuration of LiSA is very similar to the configuration of the Cisco switch, as shown in Figure 5. Every physical interface (eth0, eth1, etc.) is mapped to a virtual interface that can be treated as the Cisco IOS treats its interfaces. The configuration of the interfaces is done one by one; that is, there is no way to configure ranges of interfaces (line 1 to line 11). Finally, a SVI is created for each VLAN (line 13 to line 17) and the corresponding IP address is assigned to each one.

To measure the throughput, we propose the introduction of some background traffic between the PCs. This background traffic does not saturate the network. The idea is to create an additional flow (out of the background traffic) and measure the maximum throughput that can be handled by the switch for this additional flow in the presence of background traffic. To set the background traffic, each PC sends a constant bit rate (CBR) UDP traffic to the other three PCs. We did

```

(1) ovs-vsctl add-br SW
(2) ovs-vsctl add-port SW eth0 tag=10
(3) ovs-vsctl add-port SW eth1 tag=10
(4) ovs-vsctl add-port SW eth3 tag=20
(5) ovs-vsctl add-port SW eth4 tag=20
(6)
(7) ovs-vsctl add-port SW vlan10 tag=10
    -- set interface vlan10 type=internal
(8) ifconfig vlan10 10.0.0.254 netmask 255.255.255.0
(9) ovs-vsctl add-port SW vlan20 tag=20
    -- set interface vlan20 type=internal
(10) ifconfig vlan20 20.0.0.254 netmask 255.255.255.0
(11)
(12) sysctl -w net.ipv4.ip_forward=1

```

FIGURE 4: Configuration of OVS for the L3 switch routing testbed.

```

(1) interface Ethernet 0
(2)     switchport access vlan 10
(3)
(4) interface Ethernet 1
(5)     switchport access vlan 10
(6)
(7) interface Ethernet 2
(8)     switchport access vlan 20
(9)
(10) interface Ethernet 3
(11)     switchport access vlan 20
(12)
(13) interface vlan 10
(14)     ip address 10.0.0.254 255.255.255.0
(15)
(16) interface vlan 20
(17)     ip address 20.0.0.254 255.255.255.0

```

FIGURE 5: Configuration of LiSA for the L3 switch routing testbed.

our experiments for Ethernet (10 Mbps) and Fast Ethernet (100 Mbps) technologies.

In Ethernet (10 Mbps), every PC sends a CBR traffic of 2.5 Mbps consisting of UDP datagrams with a payload size of 500 bytes to the other PCs, totalling 7.5 Mbps of UDP throughput in both downstream and upstream in each link, leaving approximately 2.5 Mbps bandwidth unused. The additional flow (the one that is different from the background traffic and that saturates the network) is created with Iperf by injecting a 3 Mbps CBR UDP traffic between two PCs.

For Fast Ethernet (100 Mbps), every PC sends a CBR traffic of 25 Mbps consisting of UDP datagrams with a payload size of 500 bytes to the other PCs, totalling 75 Mbps of UDP throughput in both downstream and upstream in each link. This let approximately 25 Mbps bandwidth unused. The additional flow is created with Iperf by injecting a 30 Mbps CBR UDP traffic between two PCs.

FIGURE 6: Router on a stick testbed.

4.2.2. Router on a Stick. The general configuration of the router on a stick testbed is shown in Figure 6. This time, the router performs the inter-VLAN communication. The switch sends the traffic to the router through a trunk link that allows packets from both VLANs to transit. Two IP addresses are configured in the same physical NIC of the router to accomplish the routing process, using subinterfaces.

The configuration of the Cisco switch for this testbed (Figure 7) is similar to the configuration in the L3 switch routing testbed (Figure 2). This time, no SVI is created and a port is assigned as a trunk (line 12). In line 13, the VLANs allowed to pass through the trunk link are indicated and the tagging encapsulation is specified in line 14.

Small modifications are done for OVS in this testbed (Figure 8) in comparison with L3 switch routing testbed (Figure 4). As previously, all interfaces are added to the switch in their respective VLAN. This time no SVI is configured, but an additional interface is set up as a trunk, allowing the traffic of both VLAN10 and VLAN20 (line 7).


```

(1) ip routing
(2)
(3) interface range fastEthernet 1/0/1-2
(4)     switchport access vlan 10
(5)     switchport mode access
(6)
(7) interface range fastEthernet 1/0/3-4
(8)     switchport access vlan 20
(9)     switchport mode access
(10)
(11) interface fastEthernet 1/0/24
(12)     switchport mode trunk
(13)     switchport trunk allowed vlan 10,20
(14)     switchport trunk encapsulation dot1q

```

FIGURE 7: Configuration of the Cisco switch for the router on a stick testbed.

```

(1) ovs-vsctl add-br SW
(2) ovs-vsctl add-port SW eth0 tag=10
(3) ovs-vsctl add-port SW eth1 tag=10
(4) ovs-vsctl add-port SW eth2 tag=20
(5) ovs-vsctl add-port SW eth3 tag=20
(6)
(7) ovs-vsctl add-port SW eth4 trunks=10,20
(8)
(9) sysctl -w net.ipv4.ip_forward=1

```

FIGURE 8: Configuration of OVS for the router on a stick testbed.

Figure 9 depicts some important instructions for the configuration of LiSA for the router on a stick testbed. The configuration is similar to the L3 switch routing testbed (Figure 5) except that we do not create SVIs, but we have an additional interface (Ethernet 4) for the trunk link. Line 14 indicates that this trunk link allows traffic from VLAN10 and VLAN20.

The router represents the main difference between the two testbeds. We used a PC running Debian 6.0.5 with an Intel Core 2 Duo processor of 2.67 GHz, 4 GB of RAM, and the forwarding option activated, for the router. Figure 10 shows some important instructions for the configuration of this router. First, the VLAN module must be installed (line 1), the 802.1q protocol activated at boot time (line 3), and the forwarding option enabled (line 5). Then, the file `/etc/network/interfaces` must be modified to assign two IP addresses in one interface, as depicted in Figure 11.

In this testbed, we also introduced background traffic. To set this background traffic, a CBR traffic is sent from PC1 to PC4, from PC2 to PC3, from PC3 to PC1, and from PC4 to PC2. With this scheme 4 flows pass through the trunk link in both ways, downstream and upstream. The trunk link is the bottleneck link in this testbed because all the inter-VLAN communication traffic must pass through it. To measure the throughput, an additional flow (one that does saturates the network) is established between two PCs in different VLANs.

```

(1) interface Ethernet 0
(2)     switchport access vlan 10
(3)
(4) interface Ethernet 1
(5)     switchport access vlan 10
(6)
(7) interface Ethernet 2
(8)     switchport access vlan 20
(9)
(10) interface Ethernet 3
(11)     switchport access vlan 20
(12)
(13) Interface Ethernet 4
(14)     switchport trunk allowed vlan 10,20
(15)     switchport mode trunk

```

FIGURE 9: Configuration of LiSA for the router on a stick testbed.

```

(1) apt-get install vlan
(2)
(3) echo 8021q >> /etc/modules
(4)
(5) sysctl -w net.ipv4.ip_forward=1

```

FIGURE 10: Configuration of the router for the router on a stick testbed.

In Ethernet (10 Mbps), every PC sends UDP datagrams with payload size of 500 bytes at 2.0 Mbps to another PC in the other VLAN, as explained previously. With this scheme, 4 flows at 2.0 Mbps pass through the trunk link, totalling background traffic at 8.0 Mbps in both ways. The additional flow is created with Iperf by injecting a 3 Mbps CBR traffic between two PCs in different VLANs, so the flow also has to pass through the trunk link and therefore saturates it.

For Fast Ethernet (100 Mbps), the same scheme as Ethernet is used. This time the PCs send a CBR traffic at 20 Mbps, totalling background traffic of 80 Mbps in both ways, downstream and upstream, in the trunk. The additional flow is created with Iperf by injecting a 30 Mbps CBR traffic between two PCs in different VLANs.

4.2.3. Fairness. To measure fairness, the testbed shown in Figure 12 is used. In this testbed a group of PCs are placed in VLAN10 and one PC is placed in VLAN20. The general idea of this testbed is to saturate the link in VLAN20 by sending UDP traffic from the PCs in VLAN10 to the PC in VLAN20 (called T) through the switch. The VLAN20 link represents the bottleneck and the fairness is calculated according to the amount of bandwidth assigned for each flow from VLAN10 by the switch.

The configuration of the switches is almost the same as the L3 switch routing testbed. Two settings are proposed: (1) two PCs in VLAN10 and (2) three PCs in VLAN10. In both settings, the bandwidth established for each PC is varied

```

(1) auto eth0.10
(2) iface eth0.10 inet static
(3)     address 10.0.0.254
(4)     netmask 255.255.255.0
(5)
(6) auto eth0.20
(7) iface eth0.20 inet static
(8)     address 20.0.0.254
(9)     netmask 255.255.255.0

```

FIGURE 11: File `/etc/network/interfaces` of the router.

FIGURE 12: Fairness testbed.

to study the fairness of the switches with different flows for Ethernet and Fast Ethernet.

For our experiments, all the PCs (PC1, PC2, PC3, and T) have a Debian 6.0.5 operating system, a 2.6.32-5-amd64 kernel, an Intel Core 2 Duo processor of 2.67 GHz, 4 GB of RAM, and a Broadcom NetXtreme Gigabit Ethernet.

Three types of switch are used: one off-the-shelf and two PC-based. The off-the-shelf switch is a Cisco switch Catalyst 3750, as previously mentioned. We used the command `speed 10` and `speed 100` of the Cisco switch to set the desired bandwidth in the ports (10 Mbps or 100 Mbps). The PC-based switch is installed in a PC with an Intel Core 2 Duo processor of 2.67 GHz, 4 GB of RAM, and two NICs (Intel PRO/1000 P Dual Port Server Adapter). We created two partitions in the hard drive of this PC. In the first partition, we installed Debian 7.0 with a 3.2.0-4-amd64 kernel and Open vSwitch version 1.10.0 for the switching engine. In the second partition, we installed CentOS-6.3-i386 and LiSA version 2.0.1 for the switching engine. We used `ethtool` to set the desired bandwidth in the NICs (10 Mbps or 100 Mbps).

5. Results

In this section we show the results obtained by our RTT and throughput measurements in the testbeds presented in

FIGURE 13: RTT with L3 switch routing intra-VLAN communication for Ethernet.

FIGURE 14: RTT with L3 switch routing intra-VLAN communication for Fast Ethernet.

Section 4. We perform experiments for an intra- and inter-VLAN communication, in both Ethernet and Fast Ethernet technologies. First, we introduce the results for RTT, and then we present the results for throughput.

5.1. Round Trip Time

5.1.1. L3 Switch Routing. The RTT for intra-VLAN communication for Ethernet is shown in Figure 13. We varied the UDP payload size for the additional flow from 100 to 1440 bytes. As can be observed, the Cisco switch represents the best performance for the three technologies. The behavior of the two PC-based switches is similar for almost all the payload sizes.

In Figure 14, the RTT results in an intra-VLAN communication for Fast Ethernet are shown. The Cisco switch shows the lowest RTT. OVS and LiSA have a similar behavior, and for a UDP payload size between 750 and 1440 bytes, the RTT stays invariant. This is an unexpected behavior that requires further research.

Figure 15 shows the RTT results in an inter-VLAN communication for Ethernet. As it can be observed, the Cisco

FIGURE 15: RTT with L3 switch routing inter-VLAN communication for Ethernet.

FIGURE 16: RTT with L3 switch routing inter-VLAN communication for Fast Ethernet.

switch presents the lowest RTT for all the payload sizes. OVS and LiSA show almost the same results for all the cases.

Similarly to Ethernet (Figure 15), we can observe in Figure 16 that the Cisco switch has the best performance for RTT in Fast Ethernet for the inter-VLAN communication. OVS and LiSA have similar results for almost all the cases, but with a strange behavior. The RTT for some ranges of UDP payload size stays invariant (e.g., between a 1000 and 1440 bytes), which is an unexpected result. It is worth remembering that, in the case of the intra-VLAN communication for Fast Ethernet (Figure 14), this strange result was also reported.

We can also observe that the RTT for intra- and inter-VLAN communication is almost the same. That is, the overhead introduced by the L3 routing is almost imperceptible since it is done within the switch.

5.1.2. Router on a Stick. The RTT in an intra-VLAN communication is not reported for this testbed, because it is exactly the same as the one of the L3 switch routing testbed and shows the same results (Figure 13 for Ethernet and Figure 14 for Fast Ethernet).

FIGURE 17: RTT with router on a stick inter-VLAN communication for Ethernet.

FIGURE 18: RTT with router on a stick inter-VLAN communication for Fast Ethernet.

Results for the inter-VLAN routing for Ethernet are shown in Figure 17. Once again, the Cisco switch presents the lowest RTT. OVS and LiSA switches show similar results.

By comparing Figure 15 with Figure 17, we can observe how the RTT has increased since the L3 routing is done by the router and is not anymore integrated in the switch.

Figure 18 depicts the results for Fast Ethernet. OVS and LiSA show an unexpected behavior as presented in the L3 switch routing testbed for Fast Ethernet, in both intra- and inter-VLAN communication. The Cisco switch presents the best performance for all the cases.

By comparing Figure 16 with Figure 18, we can observe how the RTT has increased since the L3 routing is done by the router, and is not anymore integrated in the switch.

5.2. Throughput

5.2.1. L3 Switch Routing. In Figure 19 the throughput results in an intra-VLAN communication for Ethernet are shown.

FIGURE 19: UDP throughput with L3 switch routing intra-VLAN communication for Ethernet.

FIGURE 21: UDP throughput with L3 switch routing inter-VLAN communication for Ethernet.

FIGURE 20: UDP throughput with L3 switch routing intra-VLAN communication for Fast Ethernet.

FIGURE 22: UDP throughput with L3 switch routing inter-VLAN communication for Fast Ethernet.

We varied the UDP payload size for the additional flow from 100 to 1440 bytes. The Cisco switch presents a better performance for datagrams with a UDP payload size greater than 500 bytes. OVS and LiSA have similar results. For Fast Ethernet, the behavior is maintained and is almost the same for the three switches as shown in Figure 20.

In Figure 21 we can observe the throughput measured for inter-VLAN communication for Ethernet for this testbed. The Cisco switch presents a better performance for datagrams with a UDP payload size greater than 250 bytes. OVS and LiSA have a similar behavior for all packet sizes.

The results for throughput for inter-VLAN communication for Fast Ethernet are shown in Figure 22. The behavior is almost the same as in Ethernet but with results ten times greater. Again, the Cisco switch presents the best throughput performance for datagrams with a UDP payload size bigger than 500 bytes, and OVS and LiSA have similar results.

5.2.2. Router on a Stick. For the router on a stick testbed, the intra-VLAN communication results are not reported, since they were already measured with the L3 switch routing testbed (Figure 19 for Ethernet and Figure 20 for Fast

Ethernet). Only the inter-VLAN communication is done for this testbed.

Figure 23 depicts the throughput for Ethernet. The Cisco switch exhibits the best performance for the three switches, for datagrams with a UDP payload size greater than 250 bytes. The Cisco switch presents throughput results close to the maximum bandwidth established for this testbed (3 Mbps) for datagrams of large size. Again, OVS and LiSA present similar throughput.

The throughput for Fast Ethernet is shown in Figure 24. For all the payload sizes, OVS and LiSA present similar results. Like in Ethernet, the Cisco switch presents the best performance for datagrams with a UDP payload size greater than 250 bytes. Furthermore, the Cisco switch throughput is almost equal to the maximum bandwidth established for this testbed (30 Mbps) for datagrams of large size.

5.2.3. Fairness. For the testbed with 2 PCs, the results for Ethernet are shown in Figure 25. There are 5 experiments: (1) 7 Mbps/7 Mbps, (2) 4 Mbps/8 Mbps, (3) 3 Mbps/9 Mbps, (4) 7 Mbps/9 Mbps, and (5) 5 Mbps/8 Mbps. The first experiment (shown as 7/7 in Figure 25) means that both PC1 and PC2

FIGURE 23: UDP throughput with router on a stick inter-VLAN communication for Ethernet.

FIGURE 24: UDP throughput with router on a stick inter-VLAN communication for Fast Ethernet.

are sending a 7 Mbps flow to T (see Figure 12). The second experiment (shown as 4/8 in Figure 25) means that PC1 is sending a 4 Mbps flow to T, while PC2 is sending an 8 Mbps flow to T. The other 3 experiments have to be interpreted in a similar way to the first ones.

For each experiment, we have 6 bars in Figure 25. The first 2 bars represent the UDP throughput received in T from the flows of PC1 and PC2, respectively, when using a Cisco switch. The third and fourth bars correspond to the UDP throughput received in T from the flows of PC1 and PC2, respectively, when using an OVS switch. The last 2 bars characterize the UDP throughput received in T from the flows of PC1 and PC2, respectively, when using a LiSA switch. For all cases, the switches use almost all the bandwidth available (close to 10 Mbps) forwarding the incoming flows. The bandwidth distribution for each combination of flows is proportionally fair.

Figure 26 shows the result of fairness experiments with 2 PCs for Fast Ethernet. The tested flow settings were 70 Mbps/70 Mbps, 40 Mbps/80 Mbps, 30 Mbps/90 Mbps, 70 Mbps/90 Mbps, and 50 Mbps/80 Mbps. In this case, the distributions made by the switches are not as stable as in Ethernet. For

FIGURE 25: Fairness experiments with 2 PCs in Ethernet.

FIGURE 26: Fairness experiments with 2 PCs in Fast Ethernet.

example, we can observe that, in the case of the third experiment (shown as 30/90 in Figure 26) for a Cisco switch, the proportionality between 2 flows is not maintained. Generally, the Cisco switch presents a bigger difference of proportions than the PC-based switches.

In Figure 27, the results of fairness experiments for Ethernet with 3 PCs are shown. The tested flow settings were 4 Mbps/4 Mbps/4 Mbps, 4 Mbps/4 Mbps/8 Mbps, and 8 Mbps/8 Mbps/8 Mbps. For each experiment, we have 9 bars. The first 3 bars are the UDP throughput received in T when using a Cisco switch. The fourth, fifth, and sixth bars represent the UDP throughput received in T when using an OVS switch, while the last 3 bars represent the UDP throughput received in T when using a LiSA switch. For the first two settings, OVS and LiSA have a similar behavior that maintains a proportional distribution of the bandwidth. The Cisco switch presents the worst distribution of the bandwidth in all the tests.

The results for Fast Ethernet with 3 PCs are shown in Figure 28. The tested flow settings were 40 Mbps/40 Mbps/40 Mbps, 40 Mbps/40 Mbps/80 Mbps, and 80 Mbps/80 Mbps/80 Mbps. Like in Ethernet, OVS and LiSA show

FIGURE 27: Fairness experiments with 3 PCs in Ethernet.

FIGURE 28: Fairness experiments with 3 PCs in Fast Ethernet.

similar behavior for almost all cases. The PC-based switches maintain an acceptable proportional distribution of the bandwidth. The most unstable switch for this testbed is the Cisco switch, which does not do a fair distribution of the bandwidth.

6. Conclusions and Future Work

In this paper, several testbeds were proposed to do a comparison of the performance evaluation of a Cisco switch and two PC-based switches (OVS and LiSA). We measured the RTT and the throughput at the level of UDP for every switch.

For the RTT, we compared the intra- and inter-VLAN communication capabilities of the switches for every testbed. The Cisco switch always showed the lowest RTT of all the switches. For Ethernet, the PC-based switches presented a similar performance, which was not far from the performance of the Cisco switch. Nevertheless, the results of the PC-based switches showed a strange behavior for Fast Ethernet. For some ranges of UDP payload size, the results stayed invariant. We could not determine the cause of this anomaly.

We also compared the throughput for the three switches for the proposed testbeds. In our experiments, we observed that, for greater datagram sizes of the additional flow, the Cisco switch presented the best performance. In the L3 switch routing testbed, OVS and LiSA had a similar performance for both Ethernet and Fast Ethernet. In the router on a stick testbed, the Cisco switch stands over the PC-based switches for big datagram payload sizes. In both Ethernet and Fast Ethernet, the Cisco switch almost reached the maximum expected throughput.

The fairness distribution of the bandwidth was also measured. In almost all the experiments proposed, OVS and LiSA showed an almost fair distribution of the bandwidth. The worst fairness was presented by the Cisco switch, where some instability was present, specially in Fast Ethernet.

According to our research, we can conclude that the Cisco switch presents a better performance than the PC-based switches, and therefore, it is a better solution, but at a more expensive cost. OVS and LiSA have a similar performance. Nevertheless, the OVS virtual switch has more functionalities, and, for that reason, it represents a better choice for virtualization.

As it can be observed in the experiments, the results of the RTT in Fast Ethernet in the PC-based switches presented a strange behavior. We plan to study this behavior with more details in future works. We also propose to do an extension of our study with the consideration of other features of switches such as QoS and bonding (etherchannel). Finally, we are projecting to evaluate different versions of TCP [15] when passing through the studied switch solutions.

Acknowledgment

The authors want to thank the CDCH-UCV (Consejo de Desarrollo Científico y Humanístico) which partially supported this research under Grant no.: PG 03-8066-2011/1.

References

- [1] X. Wang, X. Liu, L. Fan, and X. Jia, "A decentralized virtual machine migration approach of data centers for cloud computing," *Journal of Mathematical Problems in Engineering*, vol. 2013, Article ID 878542, 10 pages, 2013.
- [2] R. Rendec, I. Nicu, and O. Purdila, "Linux multilayer switching with LiSA," in *Proceedings of the 5th RoEduNet IEEE International Conference*, Sibiu, Romania, June 2006.
- [3] B. Pfaff, J. Pettit, T. Koponen, K. Amidon, M. Casado, and S. Shenker, "Extending networking into the virtualization layer," in *Proceedings of the Workshop on Hot Topics in Networks (HotNets '09)*, New York, NY, USA, October 2009.
- [4] J. Pettit, J. Gross, B. Pfaff, M. Casado, and S. Crosby, "Virtual switching in an era of advanced edges," in *Proceedings of the 2nd Workshop on Data Center, Converged and Virtual Ethernet Switching (DC CAVES '10)*, Amsterdam, The Netherlands, September 2010.
- [5] Hewlett-Packard Corp and IBM and Broadcom and BNT and Brocade and Chelsio and Emulex and Juniper and QLogic and IEEE, "Edge virtual bridge proposal, version 0, rev 0.1," Technical Report, IEEE, 2010.

- [6] Z. He and G. Liang, "Research and evaluation of network virtualization in cloud computing environment," in *Proceedings of the 3rd International Conference on Networking and Distributed Computing (ICNDC '12)*, Hangzhou, China, October 2012.
- [7] E. Gamess and K. Velásquez, "IPv4 and IPv6 forwarding performance evaluation on different operating systems," in *Proceedings of the 34th Latin American Computing Conference (CLEI '08)*, Santa Fe, Argentina, September 2008.
- [8] S. Narayan, S. S. Sodhi, P. R. Lutui, and K. J. Vijayakumar, "Network performance evaluation of routers in IPv4/IPv6 environment: a testbed analysis of software routers," in *Proceedings of the IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS '10)*, pp. 707–710, Beijing, China, June 2010.
- [9] N. M. M. K. Chowdhury and R. Boutaba, "Network virtualization: state of the art and research challenges," *IEEE Communications Magazine*, vol. 47, no. 7, pp. 20–26, 2009.
- [10] N. M. M. K. Chowdhury and R. Boutaba, "A survey of network virtualization," *Computer Networks*, vol. 54, no. 5, pp. 862–876, April 2010.
- [11] A. Faur and G. Milesen, *STP/RSTP implementation in LiSA [Bachelor's thesis]*, University Politehnica of Bucharest, Bucharest, Romania, 2009.
- [12] K. Velásquez and E. Gamess, "A survey of network benchmark tools," in *Machine Learning and System Engineering*, vol. 68 of *Lecture Notes in Electrical Engineering*, pp. 465–480, 2010.
- [13] H. Marouani and M. R. Dagenais, "Internal clock drift estimation in computer clusters," *Journal of Computer Networks and Communications*, vol. 2008, Article ID 583162, 7 pages, 2008.
- [14] NLANR/DAST, "Iperf homepage," <http://iperf.fr/>.
- [15] K. Tsiknas and G. Stamatelos, "Comparative performance evaluation of TCP variants in WiMAX (and WLANs) network configurations," *Journal of Computer Networks and Communications*, vol. 2012, Article ID 806272, 12 pages, 2012.