Performance Evaluation of Networks with Physical and Virtual Links

Bruce Hartpence, Andres Kwasinski
Rochester Institute of Technology
Rochester, New York 14623
Email: bhhics, axkeec@rit.edu

***Abstract*—Understanding network operation and performance issues is critical, particularly in a time when physical and virtual topologies are converging. This convergence is underscored by the adoption of cloud computing services. However, application design scenarios behave differently as do the topologies used for those applications. This paper is an investigation into the performance of virtual and physical Gigabit links after testbed anomalies were detected in the transfer and speed rates. After hundreds of tests we conclude that there significant differences between performance of Gigabit links on native topology types and bridged topologies due to operating system changes, switching methodology and overhead. Further, additional resources allocated to virtual devices does not mean an automatic increase in network link performance.**

KEY WORDS Virtualization, Gigabit, Performance

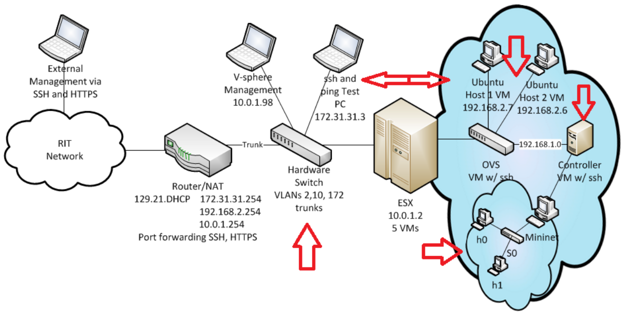

Fig. 1. SDN Testbed

I. INTRODUCTION AND MOTIVATION

Cloud computing embraces the shift towards virtualization. Network servers and services commonly reside within virtual machines and are reached via a combination of virtual and physical links. When adopting new platforms, researchers and production network administrators must have a greater understanding of reasonable expectations, especially when implementing a service level agreement [1]. In 2014, a Software Defined Networking (SDN) testbed was created at Rochester Institute of Technology. This project is documented in [2]. A central theme of an SDN architecture is virtualization and the testbed included a bare metal hypervisor (VMWare ESXi), several virtual machines (Windows and Linux) and a small Mininet deployment. However, as can be seen in the Figure 1 topology, the testbed included both physical and virtual components. During the experiments, certain behavioral differences were noted between the physical and virtual links, most notably the values for throughput and speed. Understanding the performance of the network infrastructure is critical when trying to deploy applications or converting to a virtual environment and users have their own set of expectations. In some cases, the deployment model used can have a drastic and negative impact on the performance of the application [3]. However, network latency issues such as those associated with routing or the number of hops between the source and destination can be equally detrimental. Additionally, assumptions and performance data for network physical and virtual connections may be invalid even if the advertised rates are the same. Some variation between virtual and non-virtual links is expected. In [4], VMWare stated that there is some increase in latency moving data between physical and virtual environments simply because packets must traverse the extra layer of the virtualization stack. But with so many latency sensitive applications (VoIP, banking, health care, etc.) depending on the physical and virtual infrastructure, it is critical that we understand the operation and establish verified performance metrics.

In order to better understand these differences, a second project was started in which these anomalies were more fully investigated. Gigabit links should have predictable performance numbers. However, we may not be able to make assumptions universally because the virtual links are simulated or emulated.But by sending measured traffic over the variety of possible links, the behavior of the links could be determined and the architecture would be more predictable. The virtualization industry is interested in sharing their system improvements as indicated in the VMWare white papers cited here [4], [5]. This paper extends these results with a non-proprietary series of tests that include a larger choice of scenarios.

There have also been a number of projects investigating the performance of cloud systems and their suitability for a variety of research applications. Some of these are documented in [6], [7], and the Magellan Project [8]. The Magellan project has a focus on the resources necessary for scientific computing. However, all of these provide information regarding performance on only the largest systems with dedicated support staff and funding. Smaller research architectures do not receive the same level of support and therefore cannot

978-1-4673-7707-2/15/$31.00 © 2015 IEEE

expect performance tuning or tailored tools. In addition, many of these projects tend to evaluate the performance of the hypervisor and resources allocated to the virtual machines, not the links or connections. In this paper we add the variation in physical and virtual network links. The links and the coupling of physical and virtual topologies are important because this is the most common deployment scenario.

This paper makes the following contributions:

- Establishes a set of values that can be considered trusted baselines when operating virtual and non-virtual Gigabit topologies
- Provides performance data for multiple topologies. These range from dedicated physical topologies to bridged and finally fully virtualized topologies.
- Provides an evaluation of the data from three different Gigabit switch vendors. The results show that while vendor offerings behave similarly, they fall short of dedicated virtual topology performance.
- Depicts the performance values for a range of tests including variation in TCP window size.

From our findings, it is clear that major differences can be found between native topology types, that degradation occurs when connecting disparate topologies and that transfers using different operating systems can vary greatly due to operating system programming, switching methodology and overhead.

The rest of the paper is organized as follows; section 2 presents the methodology used, a discussion of some test features and the results for each category of link. Included is an evaluation of the data and the tool used. Section 4 is a discussion about possible reasons for the observed behaviors. Sections 5 concludes.

II. METHODOLOGY

To study the variation in physical and virtual topology behavior and the impact on network metrics, we devised a series of tests to determine if variation existed between the different types of Gigabit links in the testbed and the extent of the variation. The arrows in Figure 1 indicate the locations that were tested.The tool used to send the traffic was iperf [9]. While not exhaustive, this study created a wide variety of experiments in order to provide a reasonable set of comparisons. In all, a series of eight hundred tests on the variety of topologies/paths was completed. A baseline or ”ground truth” with standard Gigabit hardware was first established. Normally, this might be the trusted benchmark data for all Gigabit equipment but as will be shown, the transfer rates vary. Once the data was obtained, it could be organized and compared across the baseline and the other virtual and non-virtual topologies.

A. Tests

The test software used was iperf [9]. iperf can generate either TCP or UDP data streams which can then be used to determine transfer rate, bandwidth, latency and jitter values. Four separate TCP tests were created using different windows sizes. The UDP jitter test was also completed. Thus, for each of the scenarios (17 in total), five tests were run. Each of these five tests were run ten times and the average taken for the final results. In all, more than eight hundred tests were completed. Each of the eight hundred tests ran for thirty seconds. A general list of the scenarios can be seen in Table I. Several other calculations $[MAX, MIN, STDDEV, etc.]$ and measurements $[speed, jitter, packetloss, etc.]$ were completed on the data sets but for brevity, this paper provides the MEAN for the amount transferred in the each scenario. However, some other examples may be given where they add value to the discussion.

TABLE I
TEST SCENARIOS

Physical Sw (D-Link, Cisco, Juniper) with 5 node and memory variants (Win XP, Win 7, Linux)
Virtual Machines (Linux and Win XP)
SDN based (controller, OVS, Win XP, Linux)
Bridged: Phy to VM (Win XP and Linux)
Mininet

Note that for most of test scenarios listed, several subgroups existed. For example the physical switch tests included Cisco 3560, Juniper EX-4200 and D-Link DGS-2205 switches. While the goal was to use the same version of iperf throughout the tests, different operating systems made this impossible. Iperf version 1.7 was used for Windows XP. All other tests used version 2.0.5.

1) TCP Window Size: TCP window size can have a drastic impact on application/network performance. If the window is too small, the sender may have to wait for the receiver to acknowledge (ACK) data which results in under-utilized links or periods of inactivity. Window sizes that are too large may result in a receiver being overrun by data causing discards and retransmissions [10]. Today, most operating systems have the ability to ACK received data very quickly. When this is combined with high speed connections, there are fewer reasons to limit window size. The expectation for these tests was that at lower window sizes, throughput and transfer would suffer. As window size grew, the opposite would occur. The iperf tool controls window size by setting the socket buffer size. Adaptive window size is currently under development. Lastly, clients on different operating systems and on different topologies of the same speed would behave similarly.

2) Jitter and Packet Loss: Jitter is the variation in expected arrival time. Excessive jitter can result in unpredictable application or process performance. With enough resources and high link speeds, the expectation is that jitter and packet loss numbers would not only be low (less than 1 percent packet loss and less than 1ms jitter) but that values would be uniform across scenarios. While these were measured, the results are not included in this paper simply because they performed exactly as expected with all scenarios (with the exception of the SDN and bridged topologies) reporting very little jitter and no packet loss.

B. Baseline

The baseline was a topology constructed of switches from various vendors. The rationale is that these network elements are certified by the vendor and then sold as having Gigabit interfaces. They should be representative of the expected performance for Gigabit Ethernet links. Figure 2 depicts the baseline topology in which two Windows 7 hosts are configured as the iperf client and server. Once the tests were completed for one vendor, the switch was exchanged with the next and the tests run again.

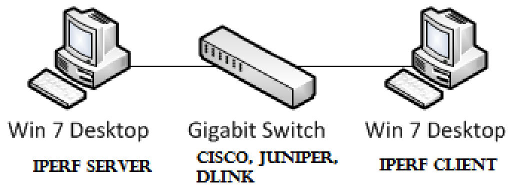

Fig. 2. Baseline Topology

By comparing vendors to each other we can develop a picture of the industry metrics for a particular parameter. Table II contains the baseline values for data transfer in megabytes. For the transfers, the standard deviation values for the tests are included in parenthesis.

TABLE II
BASELINE TRANSFER VALUES (MB) VS. WINDOWS SIZE

Window	Cisco	Juniper	D Link
4K	772.6 (16.85)	788.5 (3.41)	814.9 (33.64)
8K	1798 (6)	1820 (30.66)	1824 (23.75)
16K	1835(5)	1820 (10)	1831 (3)
64K	2793(9)	2725 (33.54)	2800 (16.73)

The D-Link DGS-2205 switch was a small, unmanaged switched commonly used in SOHO environments. Both the Cisco 3560 and Juniper EX-4200 switches were production devices meant for larger networks. The switches had 128K, 128MB and 1GB RAM respectively. The default configurations were used. While they performed similarly, they are rated with different packet per second (pps) capability. At Gigabit speeds, the Cisco had six times the capacity of the D-Link and the Juniper had about ten times the capacity of the Cisco. The low memory would undoubtedly cause scaling problems for the D-Link As can be seen, in each test (and scaling from one test to another) the physical switches parallel each other. As we expected, throughput increased as window size grew. Finally, this switches performed similarly on average but as can be seen by the standard deviation, the switches did not have the same consistency in terms of performance.

III. RESULTS

The results are broken up by the type of connection with a final comparison being provided in a chart.

A. Results: Virtual Machines

In this topology there were two types of virtual machines tested; Linux and Windows XP. The local testbed had insufficient resources for a deployment containing several Windows 7 VMs. The topology for the VMs is the same as that for the desktop machines except that the switch is virtualized. The same series of tests was run between the VMs. However, in order to provide some data that might indicate the impact of a resource change, both the Linux and Windows XP VMs were tested using 1GB and 2GB of memory. Each of the VMs was allocated a Gigabit Ethernet interface via an emulated version of an Intel Gigabit Ethernet network interface card. Since the VMs are communicating with each other, all of the traffic will travel over the internal virtual links. In other words, there are no physical connections. Table III contains the resultant data. The standard deviations are in parenthesis.

TABLE III
VM TRANSFER VALUES (MB)

Test	Linux 1GB	Linux 2GB	WinXP 1GB	WinXP 2GB
Window 4K	991.9(18.9)	994(12.6)	2007(7.8)	2000(28.3)
Window 8K	1700(44.5)	1779(31.4)	2014(20.1)	2013(22.8)
Window 16K	4059(109.3)	4101(85.5)	2424(14.3)	2401(8.3)
Window 64K	10980(218.2)	10800(352.1)	3612(64.2)	3609(76.7)

Comparing the results in Table III to the baseline values in Table II can be seen that amount data transferred between the virtual machines exceeds that sent between the baseline nodes. This behavior is somewhat expected. In a recent white paper discussing the improvements made in various versions of their hypervisor, VMWare states the following;

> Due to performance improvements in vSphere 4.1, a single VM can saturate a 10Gbps link in both transmit and receive cases. Using regular TCP/IP sockets, virtual machines residing on the same vSphere host can communicate at rates of up to 27Gbps. A single VM with multiple virtual NICs running on vSphere can easily saturate the bandwidth of four 10Gbps NICs (the maximum number of 10Gbps NICs for a cumulative throughput of more than 35Gbps. [5]

The test machine used in this scenario was equipped with 10Gbps interfaces and the associated virtual NIC driver. It can also be seen that the resources allocated to the VMs do not greatly affect the outcome, though there are some minor differences. At this point, variation in the behavior between operating systems and iperf versions becomes clear with the Windows XP virtual machines leading when the TCP window size is small and lagging behind considerably as window size grows. Again, with the age of the operating system this is probably to be expected though we will return to this point later in the paper.

B. Results: SDN

The SDN topology adds a controller which enables communication between the virtual nodes via an OpenFlow enabled switch called openvswitch or OVS [11]. In actual fact, the SDN topology (shown in Figure 3) connects two virtual switches via OVS (middle right) and in this SDN configuration, the controller decides what traffic is permitted.

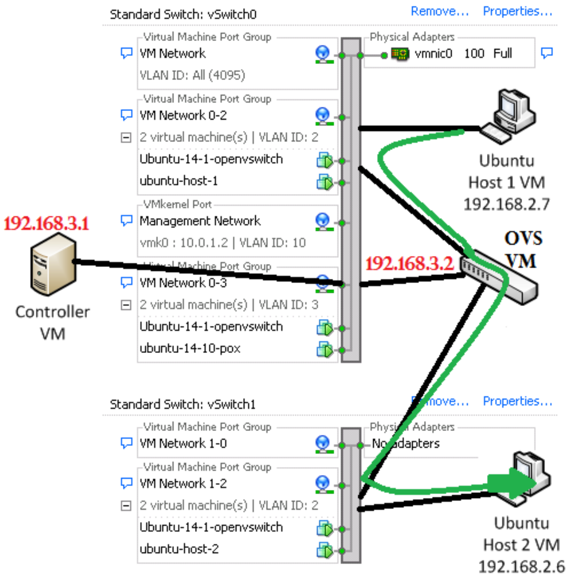

Fig. 3. SDN Topology

After this OpenFlow based communication, the switch installs the necessary entries in the flow table, enabling traffic between the virtual nodes. Because of this extra communication channel, the expectation would be that this configuration would increase the latency and therefore decrease the amount of data transferred between the nodes. The results of the iperf tests are shown in Table IV. Standard deviations are in parenthesis.

TABLE IV
SDN TRANSFER VALUES (MB)

Test	SDN Linux	SDN Win XP
Window 4K	447.7(13.6)	1280(31.9)
Window 8K	765.5(13.7)	1281(17)
Window 16K	1646(49.8)	1559(16.4)
Window 64K	4127(64.8)	2259(54.9)

At small TCP window sizes, the SDN topology transferred less data than both the baseline and the virtual topologies. However, the virtualized nature of the topology seems to have won out as window size grew with SDN catching at least the baseline. Again the behavioral differences between the operating systems is observed.

C. Results: Bridged Connection and Mininet

This section describes two special cases that are meant to fill in the blanks; bridged and Mininet. The bridged and Mininet topologies are shown in Figure 4. In the bridged topology, the Gigabit connection spans the physical and virtual topologies. This test set was added to provide information regarding the connection that would run between virtual and physical hosts. This can be a very important scenario to consider as cloud adoption forces exactly this type of architectures. Given that the connection includes a physical switch and that the virtual link is labeled as Gigabit, the expectation might be that the bridged connection would be very close to the baseline values; that the physical network would be the limiting factor. However, increased latency could occur since as stated in [4], there is the addition of the virtual networking protocol stack which might force the values below the baselines. So, one host was a virtual machine resident on the hypervisor chassis and the other was a desktop node. The nodes are connected via virtual and physical Gigabit switches. Mininet [12] is a network simulator that can create a wide variety of topologies for both experimentation and education. Mininet topologies can also be connected to the physical world. However, the major difference between it and the other topologies examined, is that the Mininet SDN network and nodes reside within a single VM. This means that the resources allocated to a single VM must be split amongst the network nodes because they each ”live” within their own containers. However, the communication is between processes on the same VM. The results for the bridged and Mininet topologies are shown in Table V. For ease of comparison, the average of the baseline transfer MEANs are included. Standard deviations are in parenthesis.

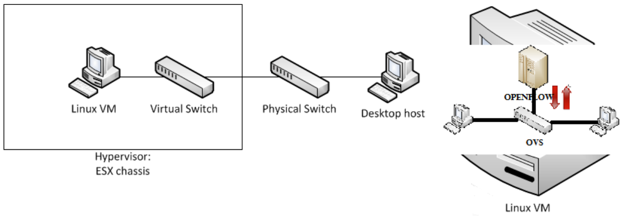

Fig. 4. Bridged and Mininet Topologies

TABLE V
BRIDGED AND MININET TRANSFER VALUES (MB)

Test	Baseline AVG	Bridged Connection	Mininet
Window 4K	792(27.9)	949.1(4.3)	2054(8)
Window 8K	1814(25.4)	952(1.1)	2244(6.6)
Window 16K	1829(9.2)	1321(15.1)	2980(242.4)
Window 64K	2773(40.5)	1768(10.7)	3284(12.8)

Only at very small window sizes (4000 bytes) does the bridged connection exceed that of the baseline. The rest of the time, the bridged connection lags behind. This seems reasonable given that there is an extra logical device in between the nodes. The Mininet topology out-performs the baseline topology by a considerable margin and at small window sizes still records an impressive transfer. This also seems reasonable as the communication is between logical nodes or processes in the same virtual machine. This would also seem to indicate that Mininet does a decent job of emulating the requested link speed.

D. Results: Comprehensive

Figure 5 depicts all of the transfer results in graphical format. The rectangle encloses the baseline values. From this graph it can be seen that for all window sizes and without regard for operating system, dedicated virtual topologies out-perform physical architectures. Mininet based networks also perform very well. SDN typically does not keep pace unless until window size are very large. Bridged topologies containing physical and virtual components perform below that of dedicated physical topologies. Topologies containing Windows XP are almost always slower though as some results show, there are times when Windows XP still performs surprisingly well.

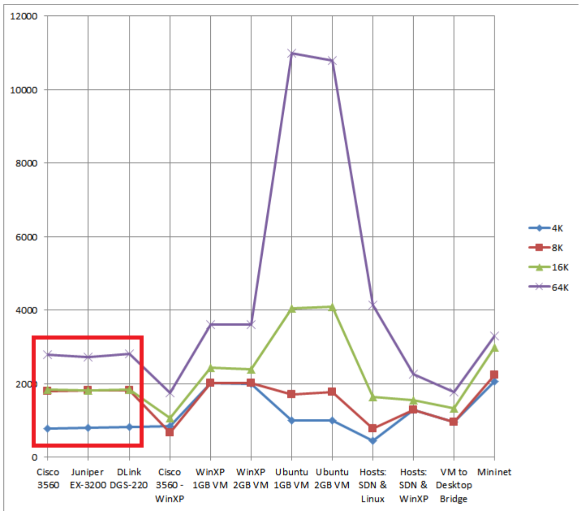

Fig. 5. Comprehensive Results

There were three different operating systems involved in this project; Windows XP, Windows 7 and Ubuntu Linux. Almost without exception Windows XP did not perform well and in fact was only used when resources would not support Windows 7. The older version of iperf was only used with Windows XP. One performance exception was that the Windows XP VMs behaved just as the Linux VMs when changing the VM memory resources. This data caused another series of tests to be added to the project: various operating systems against dedicated physical switches. After seeing the behavior of the baseline and virtual nodes we returned to the baseline topology in order add the following scenarios; Windows 7 with 4GB and 8GB RAM, Ubuntu 14 with 4GB and 8GB of RAM and Windows XP 1GB RAM laptops. Since all of the switches perform similarly, only one need be used and so all of these were run on the Cisco switch. The results of these tests can be seen in Figure 6. Figure 6 shows that until window size reaches 64k, Linux lags behind both Windows 7 and XP. The results for the same operating system but with different amounts of RAM are almost identical which leads to the conclusion that RAM has little impact on the transfer capabilities of the nodes. This is supported by the virtual machine tests. While the tests and scenarios run in this project provide a very good idea of the behavior to be seen in these configurations, there are other topology combinations or tests that might have been added. For example, window size might be allowed to scale beyond 64K. The overall results seem to have verified the performance noted in the VMWare study [5]; namely that completely virtual topologies tend to out-perform physical version. However, this project went further, adding the very important cases for VMs, a bridged topology and Mininet. The project also tested a variety of hardware switches and SDN. Mininet, also a virtual topology, out-performs the physical build. SDN topologies seem to suffer from increased latencies and therefore transfer less data. It should be noted that there are several SDN topology types. Another way to look at the data is in its entirety; for example comparing all of the baseline data to all of the virtual machine data. The Table VI depicts the features of these larger data sets. Note that the baseline data does not include the desktop variations - only Windows 7.

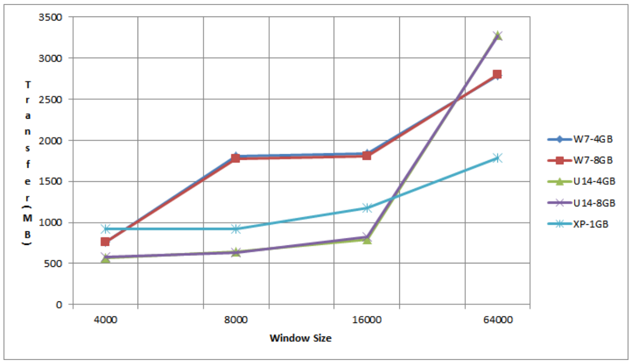

Fig. 6. OS modifications

TABLE VI
SCENARIO TRANSFER VALUES (MB)

Window	Baseline	VMs	SDN	Bridged	Mininet
4K	805.2	1498.2	863.9	949.1	2054
8K	1527.7	1876.5	1023.3	952	2244
16K	1639.3	3246.3	1602.5	1321	2982
64K	2515.8	7250.3	3193	1768	3286

This approach simplifies things and it may provide a glimpse into what a heterogeneous topology consisting of many different node and link types might be present. Again it can be seen the results fall within the behavior outlined in this paper.

E. Tool Validation

Part of the project was to verify that iperf was reporting data in a consistent fashion and representing the transferred data properly. In examining the calculations that were the focus of this paper, the value used for a megabyte of data (MB) was 1,048,576 bytes. Converting this to bits gives 8,388,608 bits. On a network, 1Mbps = 1,000,000 bits per second. In a sample calculation, iperf transfers 48.6MB in 30sec and reports a speed of 13.6Mbps. Performing the calculation: 48.6

* 8,388,608 / 30 = 13.589Mbps. Using Wireshark, the ”conversation” between the iperf client and server can be followed. For the same calculation, all of the packets traveling between the client and server were captured. In total, 34,927 frames were sent, each carrying a payload of 1460 bytes giving a transfer speed of 13.598Mbps. 34,927 * 1460 * 8 / 30 = 13.598Mbps. 1460 bytes is used because this is the payload after removing the overhead for each frame. These calculations indicate that iperf and Wireshark agree on the values, verifying the tool used for the tests.

IV. DISCUSSION

Some of the observations that can be made from these results can be summarized as follows;

- Physical switches have the same behavior regardless of vendor or memory.
- Native virtual topologies outperform physical topologies.
- Nodes of the same operating system behave similarly, regardless of memory allocation. This is true for bare metal and virtual nodes.
- Windows and Linux have very different performance profiles.
- But perhaps most importantly, assumptions of network performance referencing baseline or benchmark data cannot be relied upon because of the differences in node and topology behavior.

With regard to the switches, it is unlikely that the same results would occur in loaded scenarios. But, the results provide a solid foundation for experimenting with various architectures. Switch configuration is another area for conjecture because device operation can vary greatly. In order to get one look into the operation of the network elements, we can examine the traffic coming from each one of them while they were in a default configuration. Neither the D Link or the ESXi virtual switch generated management or overhead traffic. This does not mean a complete absence of processes, but they were not active on the network. However, the Cisco and Juniper were both engaged in operations such as spanning tree, management protocols (discovery, link management, etc.) and the dynamic host configuration protocol. A very interesting set of results can be seen in the data from the nodes. Why does doubling the RAM have so little effect? We can hypothesize that operating systems compartmentalize memory for system and processes. The iperf program may have been running in a space that was separate. In addition, the nodes (like the switches) were not loaded with network traffic. Fully loaded or busy nodes might have markedly different results. That Linux and Windows have different performance profiles may be because of the socket code used for each case. Certainly different development processes are in play. One interesting footnote is that while Windows out-performed Linux in many cases (below 64k TCP windows), the Linux results were far more stable and consistent. Lastly, this project establishes performance numbers that can be relied on and that can serve as baselines for any project. Performance testing, tuning and reevaluation is critical because of the changing nature of topologies and the integration of virtual and physical connectivity. Values that work in one circumstance to not hold true for others as is evidenced by the bridged topology.

V. CONCLUSION

In this paper we investigated the performance variations seen in Gigabit links when comparing virtual and non-virtual topologies. One important outcome was to establish a ground truth baseline for a Gigabit link. It was revealed that the operating system itself has a great impact on link performance and throughput; much more than the switch manufacturer. Additionally, there is a marked difference in values obtained from the same tests. This shows that some vendors and situations have less variability than others. The more than eight hundred tests showed that over the scenarios there were clear differences in the behavior of the configuration/topologies tested. Virtual links tend to out-perform physical links which calls into question the veracity of the virtual provisioning or the performance one can expect when presented with data derived from virtual machines only. This is shown in the bridged topology results. What is also clear it that the test data accumulated for a particular topology cannot be applied to another topology. Lastly SDN topologies may suffer from latency issues associated with the addition of the controller. The tests run here show that there are marked differences in performance between network links, even if they are labeled as the same type.

REFERENCES

[1] N. Huber, M. von Quast, M. Hauck, and S. Kounev, “Evaluating and modeling virtualization performance overhead for cloud environments.,” in *CLOSER*, pp. 563–573, 2011.

[2] B. Hartpence, “The rit sdn testbed and geni,” *RIT Digital Media Library*, 2015.

[3] L. Ramakrishnan, R. S. Canon, K. Muriki, I. Sakrejda, and N. J. Wright, “Evaluating interconnect and virtualization performance for high performance computing,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, no. 2, pp. 55–60, 2012.

[4] VMWare, “Network i/o latency on vmware vshpere 5,” *Performance Study Whitepaper*, 2012.

[5] VMWare, “Vmware vsphere 4.1 networking performance,” *Performance Study Whitepaper*, 2011.

[6] K. R. Jackson, L. Ramakrishnan, K. Muriki, S. Canon, S. Cholia, J. Shalf, H. J. Wasserman, and N. J. Wright, “Performance analysis of high performance computing applications on the amazon web services cloud,” in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, pp. 159–168, IEEE, 2010.

[7] J. J. Rehr, F. D. Vila, J. P. Gardner, L. Svec, and M. Prange, “Scientific computing in the cloud,” *Computing in science & Engineering*, vol. 12, no. 3, pp. 34–43, 2010.

[8] L. Ramakrishnan, P. T. Zbiegel, S. Campbell, R. Bradshaw, R. S. Canon, S. Coghlan, I. Sakrejda, N. Desai, T. Declerck, and A. Liu, “Magellan: experiences from a science cloud,” in *Proceedings of the 2nd international workshop on Scientific cloud computing*, pp. 49–58, ACM, 2011.

[9] “https://iperf.fr/,”

[10] U. o. S. C. ISI, “Rfc 793 transmission control protocol,” *IETF*, 1981.

[11] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “Openflow: enabling innovation in campus networks,” *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[12] B. Lantz, B. Heller, and N. McKeown, “A network in a laptop: rapid prototyping for software-defined networks,” in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, p. 19, ACM, 2010.