Network Measurements in Virtualized Networks and its Challenges

Ahmed Abujoda
Institute of Communications Technology
Faculty of Electrical Engineering and Computer Science
Leibniz Universität Hannover

1. Introduction

Virtualization in networks is becoming more popular than ever before. If we look at the end points of the network (clients, servers or peers), we see physical machines which have been virtualized to host more than one virtual machine (VM) so that multiple virtual network nodes which share common physical resources can be created. Shifting to this form of virtualization is growing fast in networks today. According to Gartner [1], 18% of server workloads in 2009 run on virtualized servers; that share will grow to 28% next year and reach almost 50% by 2012. And if we look inside the network core, we can also see a trend toward virtualization. Network components like links, routers and switches are being virtualized. Several solutions have been developed and a considerable amount of research is being carried out to apply virtualization in the network core. One example is developing virtual routers that can be migrated from one physical network infrastructure to another [2]. Another form of virtualization has been implemented in some research testbed networks like PlanetLab and G-Lab. It presents a distributed virtualization approach- each service runs in a slice of the testbed resources. Multiple slices run concurrently on the testbed network, where slices act as network-wide containers that isolate services from each other [3]. A slice consists of a set of VMs, one per each physical node. In summary, virtualization is expanding to cover a considerable part of the network architecture. A vision of a complete virtualized Internet could come true one day.

Since this paper is concerned with network measurements in virtualized networks, a brief introduction to the standard network measurement approaches is indispensable. Network measurements aim at defining and quantifying the behavior and the properties of the network. Several measurement metrics such as delay, loss jitter, capacity and available bandwidth have been considered by network measurements. In addition to that, network topology discovery is also an important part of network measurements. In regard to network measurements methodologies, network measurements can be classified into passive and active measurements. Network measurements in general concern with getting a better understanding about the network.

Network performance, deficiencies and shortcomings cannot be seen without accurate network measurements.

2. Network measurements challenges in virtualized networks

Having a virtualized network raises new challenges to the standard network measurement methods. Through this section, we would like to discuss some of these challenges.

Time stamps: If we consider virtualization at the end nodes, we can see that time stamping is a major challenge. Delay, jitter, capacity and available bandwidth measurements require accurate time stamping at the measuring nodes. Implementing VMs at the end nodes requires a scheduling and switching mechanism between the different operating systems; the physical resources have to be switched in a timely manner between several VMs. As a result of that, packets which belong to a specific VM may be queued until the physical system switches back to that VM. This in turn leads to inaccurate and bad time stamping; the packet may be stamped by the VM several micro- or milliseconds later than the real arrival time of the packet. Obtaining the time stamp of each packet at the scheduling and switching level and transferring it to the upper VMs level could be an approach to overcome this challenge. Another approach would be to modify the architecture of the virtualized system (VMs plus the scheduler) to guarantee a better timely access to the physical network interfaces.

Tools deployment: Beside time stamping, deployment of some standard measurement tools in the virtualized end nodes approaches like G-lab or Planetlab is not directly applicable. Some tools have to be modified or adapted to match the architectural requirements of the virtualized system. For example as explained in [9], in regards to raw sockets PlanetLab uses a protected raw socket type which was created by modifying the standard Linux socket. This protected raw socket type requires that the measurement tool, in order to receive and send packets, has to bind its socket to a specific local port. Another example, as mentioned in [8], tools which use raw kernel sockets can get more accurate time stamps than the tools which use upper layer
sockets. Therefore, the deployment of a tool in a virtualized system like G-lab or planetLab requires a good understanding of the tool methodology and the architecture of the virtualized system.

**Topology discovery:** Considering the virtualization inside the network (interconnection components), we can see also some other challenges. For example, instead of mapping a single physical network topology, we have to distinguish between several virtual topologies and a single physical topology. Active measurement tools like traceroute which rely on the IP protocol will not be able to see the real physical network. The geographical distance to the routers will vary due to the migration process which will lead to more dynamic delay in the network. One approach to tackle this problem is to study the long-term behavior of the new virtualized network in terms of delay and physical distance variation with time. A better understanding of the new networks will help us to build new better discovery topology tools.

**Backbone measurements access:** Furthermore, in today’s networks, where the physical network infrastructure belongs to a single organization, installing measurements tools inside the backbone of the network is not a problem. But in the future when several virtual networks owned by several organizations are deployed in the same physical network, installing measurements tools inside the network backbone will not be possible. Therefore, a shared measurement infrastructure will be required. Access and management of this infrastructure will be a challenge in terms of measurements collection and transfer. With regards to measurements collection, a mechanism is required to distinguish between traffic belonging to different virtual networks. While in case of transfer, a management and processing system which can transfer real-time measurement data to the owner of the virtual network has to be developed. Per-flow monitoring methods and layer two discovery protocols can be one approach to distinguish between different virtual networks.

3. Preliminary measurements

In order to investigate the network measurements challenges in virtualized networks, we carried out some measurements in G-Lab network. G-Lab network is a testbed network controlled by the G-Lab project community. The network consists of over 170 nodes connected together through the German National Research and Education Network, DFN [4], [5]. As explained earlier, G-Lab implements end node virtualization using distributed virtualization approach; a slice, VM per each node, is granted to each user. Research targeting future Internet development is being carried out in G-lab; G-lab may reflect the future virtualized networks. Our measurements in G-lab started with applying topology discovery using traceroute. Traceroute measurements were collected, processed and visualized to extract the topology graph of G-Lab (figure 1). The traceroute version that we used is the pre-installed G-Lab version. To improve topology discovery results, we tried other newer versions and implementations of traceroute in G-lab platform, but these versions were not able to function properly. In addition to that, we complied and run some standard tools to measure other parameters like capacity and available bandwidth. But we encountered difficulties with these tools due to bad time stamp accuracy, high context switching and undeliverable packets. We plan to investigate the reasons behind these difficulties.

The above preliminary results can be explained with referring to section 2. Regarding the topology discovery tools, we can infer that G-Lab architecture prohibits some tools from working properly. However, with regards to the other tools, the scheduling and timing mechanism produce inaccurate time stamping which hinders the functionality of these tools. Furthermore, the G-Lab architecture can be also a problem for these tools.

These results provided us with some indication to the challenges that would be encountered in virtualized networks.

![Figure 1 G-lab traceroute Topology](image)

4. Related work

Yalagandula et al. carried out end-to-end active measurements in PlanetLab using tools like Pathrate, pathchirp and spruce [6]. However, they stated that a significant engineering effort was required to modify these tools in order to run them reliably and with reasonable accuracy in PlanetLab.
Spring et al. compared the time stamps obtained at the slice-level and at the scheduler level (kernel-level) in Planetlab [7]. They showed that the difference between the slice-level and kernel-level time stamps is typically small when the packets are sent out. However, when the packets are received, the difference is typically large and variable. Furthermore, they investigated the possibility of sending packets at precise times (with predefined time space) in the slice-level. Their measurements showed that out of 10 packets sent with 1 ms time space only 20-40% kept their precise timing. They proposed delegating the packet transmission scheduling to the scheduler (Kernel) in order to overcome this problem.

Ongaro et al. discussed the impact of the Virtual Machine Monitor (VMM) scheduler on I/O performance when several VMs run concurrently on the same physical machine [8]. They showed that the VMM scheduler can guarantee the fairness in term of processor resource sharing, but in term of I/O performance fairness the scheduler achieves high and uneven I/O response and make the VM with intensive I/O load performs poorly and unpredictably. One example was the network bandwidth achieved by seven VMs running the same network load. It was found out that the achieved bandwidth was not fairly distributed among the VMs and it varied with almost factor of 1.5 from on VM to another. They discussed and proposed optimization methods to improve the scheduler in term of I/O performance fairness, but more optimization is still also required.

The related work mentioned above can be interpreted in the context of our paper. The first and second related work can help us to understand the measurement challenges (tools deployment and time stamp) in virtualized environment like G-Lab and PlanetLab. The performance of I/O in VMs which was discussed in the third related work can have an impact on the time stamp accuracy achieved in the VMs. The effect of the scheduler on the performance of the network I/O interface can have also a similar effect on the time stamp accuracy. The uneven latency I/O response can result in uneven time stamp accuracies at the different VMs i.e. VMs which have high latency access to the network I/O interface will obtain worse time stamp than the other VMs with lower latency.

5. Conclusion and Future work

Through this paper we have reviewed three major forms of virtualization in networks. We expected that a considerable part of the future Internet and networks will be virtualized. We have identified some challenges that encounter network measurements in virtualized networks. G-lab network was chosen to carry out some preliminary measurements. The results of the preliminary measurements can be considered as a starting point to investigate these challenges further in order to find appropriate solutions.

As future works, we will carry out more measurements in G-Lab to unveil the causes of the measurements challenges and to develop appropriate solutions. On the other hand, we plan to study further the measurements challenges in the network core visualization approach. Developing measurement solutions which can fit to the structure of these networks will be considered. Available technologies like Openflow and NetFPGA could be useful tools for our implementations.

References