Considerations for Benchmarking Network Virtualization Platforms
draft-skommu-bmwg-nvp-02.txt

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt
The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on December 27, 2018.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in
Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Abstract

Current network benchmarking methodologies are focused on physical networking components and do not consider the actual application layer traffic patterns and hence do not reflect the traffic that virtual networking components work with. The purpose of this document is to distinguish and highlight benchmarking considerations when testing and evaluating virtual networking components in the data center.

Table of Contents

1. Introduction .................................................. 2
2. Conventions used in this document ............................ 3
3. Definitions .................................................. 4
   3.1. System Under Test (SUT) ............................ 4
   3.2. Network Virtualization Platform .................... 4
   3.3. Micro-services ..................................... 6
4. Scope ........................................................ 7
   4.1. Virtual Networking for Datacenter Applications ...... 7
   4.2. Interaction with Physical Devices .................... 8
5. Interaction with Physical Devices ............................ 8
   5.1. Server Architecture Considerations .................... 11
6. Security Considerations ..................................... 14
7. IANA Considerations ......................................... 14
8. Conclusions .................................................. 14
9. References .................................................. 14
   9.1. Normative References .................................. 14
   9.2. Informative References ................................ 15

Appendix A. Partial List of Parameters to Document ............ 16
   A.1. CPU ................................................... 16
   A.2. Memory ................................................ 16
   A.3. NIC ................................................... 16
   A.4. Hypervisor ............................................ 17
   A.5. Guest VM ............................................. 18
   A.6. Overlay Network Physical Fabric ....................... 18
   A.7. Gateway Network Physical Fabric ....................... 18

1. Introduction

Datacenter virtualization that includes both compute and network virtualization is growing rapidly as the industry continues to look for ways to improve productivity, flexibility and at the same time
cut costs. Network virtualization, is comparatively new and expected to grow tremendously similar to compute virtualization. There are multiple vendors and solutions out in the market, each with their own benchmarks to showcase why a particular solution is better than another. Hence, the need for a vendor and product agnostic way to benchmark multivendor solutions to help with comparison and make informed decisions when it comes to selecting the right network virtualization solution.

Applications traditionally have been segmented using VLANs and ACLs between the VLANs. This model does not scale because of the 4K scale limitations of VLANs. Overlays such as VXLAN were designed to address the limitations of VLANs.

With VXLAN, applications are segmented based on VXLAN encapsulation (specifically the VNI field in the VXLAN header), which is similar to VLAN ID in the 802.1Q VLAN tag, however without the 4K scale limitations of VLANs. For a more detailed discussion on this subject please refer RFC 7364 "Problem Statement: Overlays for Network Virtualization".

VXLAN is just one of several Network Virtualization Overlays (NVO). Some of the others include STT, Geneve and NVGRE. STT and Geneve have expanded on the capabilities of VXLAN. Please refer IETF’s nvo3 working group <https://datatracker.ietf.org/wg/nvo3/documents/> for more information.

Modern application architectures, such as Micro-services, are going beyond the three tier app models such as web, app and db. Benchmarks MUST consider whether the proposed solution is able to scale up to the demands of such applications and not just a three-tier architecture.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying significance described in RFC 2119.
3. Definitions

3.1. System Under Test (SUT)

Traditional hardware based networking devices generally use the device under test (DUT) model of testing. In this model, apart from any allowed configuration, the DUT is a black box from a testing perspective. This method works for hardware based networking devices since the device itself is not influenced by any other components outside the DUT.

Virtual networking components cannot leverage DUT model of testing as the DUT is not just the virtual device but includes the hardware components that were used to host the virtual device. Hence SUT model MUST be used instead of the traditional device under test.

With SUT model, the virtual networking component along with all software and hardware components that host the virtual networking component MUST be considered as part of the SUT.

Virtual networking components may also work with higher level TCP segments such as TSO. In contrast, all physical switches and routers, including the ones that act as initiators for NVOs, work with L2/L3 packets.

Please refer to section 5 Figure 1 for a visual representation of System Under Test in the case of Intra-Host testing and section 5 Figure 2 for System Under Test in the case of Inter-Host testing.

3.2. Network Virtualization Platform

This document does not focus on Network Function Virtualization.

Network Function Virtualization (NFV) focuses on being independent of networking hardware while providing the same functionality. In the case of NFV, traditional benchmarking methodologies recommended by IETF may be used. Considerations for Benchmarking Virtual Network Functions and Their Infrastructure IETF document addresses benchmarking NFVs.

Typical NFV implementations emulate in software, the characteristics and features of physical switches. They are similar to any physical L2/L3 switch from the perspective of the packet size, which is typically enforced based on the maximum transmission unit used.

Network Virtualization platforms on the other hand, are closer to the application layer and are able to work with not only L2/L3...
packets but also segments that leverage TCP optimizations such as Large Segment Offload (LSO).

NVPs leverage TCP stack optimizations such as TCP Segmentation Offload (TSO) and Large Receive Offload (LRO) that enables NVPs to work with much larger payloads of up to 64K unlike their counterparts such as NFVs.

Because of the difference in the payload, which translates into one operation per 64K of payload in NVP versus ~40 operations for the same amount of payload in NFV because of having to divide it to MTU sized packets, results in considerable difference in performance between NFV and NVP.

Please refer to figure 1 for a pictorial representation of this primary difference between NPV and NFV for a 64K payload segment/packet running on network set to 1500 bytes MTU.

Note: Payload sizes in figure 1 are approximates.
NPV (1 segment)       NFV (40 packets)

Segment 1          Packet 1
+-------------------+    +-------------------+
    Headers          |    Headers          |
    +-----------------+    +-----------------+
    | Pay Load - upto 64K |    | Pay Load < 1500  |
    +-----------------+    +-----------------+
                     +-------------------+
Packet 2
+-------------------+
    Headers          |
    +-----------------+
    | Pay Load < 1500 |
    +-----------------+
                     +-------------------+
Packet 20
+-------------------+
    Headers          |
    +-----------------+
    | Pay Load < 1500 |
    +-----------------+
                     +-------------------+

Figure 1  Payload NPV vs NFV

Hence, normal benchmarking methods are not relevant to the NVPs.

Instead, newer methods that take into account the built in advantages of TCP provided optimizations MUST be used for testing Network Virtualization Platforms.

3.3. Micro-services

Traditional monolithic application architectures such as the three tier web, app and db architectures are hitting scale and deployment limits for the modern use cases.

Micro-services make use of classic unix style of small app with single responsibility.
These small apps are designed with the following characteristics:

Each application only does one thing - like unix tools
Small enough that you could rewrite instead of maintain
Embedded with a simple web container
Packaged as a single executable
Installed as daemons
Each of these applications are completely separate
Interact via uniform interface
REST (over HTTP/HTTPS) being the most common

With Micro-services architecture, a single web app of the three tier application model could now have 100s of smaller apps dedicated to do just one job.

These 100s of small one responsibility only services will MUST be secured into their own segment - hence pushing the scale boundaries of the overlay from both simple segmentation perspective and also from a security perspective

4. Scope

This document does not address Network Function Virtualization has been covered already by previous IETF documents (https://datatracker.ietf.org/doc/draft-ietf-bmwg-virtual-net/?include_text=1) the focus of this document is Network Virtualization Platform where the network functions are an intrinsic part of the hypervisor’s TCP stack, working closer to the application layer and leveraging performance optimizations such TSO/RSS provided by the TCP stack and the underlying hardware.

4.1. Virtual Networking for Datacenter Applications

While virtualization is growing beyond the datacenter, this document focuses on the virtual networking for east-west traffic within the datacenter applications only. For example, in a three tier app such web, app and db, this document focuses on the east-west traffic between web and app. It does not address north-south web traffic accessed from outside the datacenter. A future document would address north-south traffic flows.
This document addresses scale requirements for modern application architectures such as Micro-services to consider whether the proposed solution is able to scale up to the demands of micro-services application models that basically have 100s of small services communicating on some standard ports such as http/https using protocols such as REST.

4.2. Interaction with Physical Devices

Virtual network components cannot be tested independent of other components within the system. Example, unlike a physical router or a firewall, where the tests can be focused directly solely on the device, when testing a virtual router or firewall, multiple other devices may become part of the system under test. Hence the characteristics of these other traditional networking switches and routers, LB, FW etc. MUST be considered.

- Hashing method used
- Over-subscription rate
- Throughput available
- Latency characteristics

5. Interaction with Physical Devices

In virtual environments, System Under Test (SUT) may often share resources and reside on the same Physical hardware with other components involved in the tests. Hence SUT MUST be clearly defined. In this tests, a single hypervisor may host multiple servers, switches, routers, firewalls etc.,

Intra host testing: Intra host testing helps in reducing the number of components involved in a test. For example, intra host testing would help focus on the System Under Test, logical switch and the hardware that is running the hypervisor that hosts the logical switch, and eliminate other components. Because of the nature of virtual infrastructures and multiple elements being hosted on the same physical infrastructure, influence from other components cannot be completely ruled out. For example, unlike in physical infrastructures, logical routing or distributed firewall MUST NOT be benchmarked independent of logical switching. System Under Test definition MUST include all components involved with that particular test.
Inter host testing: Inter host testing helps in profiling the underlying network interconnect performance. For example, when testing Logical Switching, inter host testing would not only test the logical switch component but also any other devices that are part of the physical data center fabric that connects the two hypervisors. System Under Test MUST be well defined to help with repeatability of tests. System Under Test definition in the case of inter host testing, MUST include all components, including the underlying network fabric.

Figure 2 is a visual representation of system under test for inter-host testing.
Virtual components have a direct dependency on the physical infrastructure that is hosting these resources. Hardware characteristics of the physical host impact the performance of the virtual components. The components that are being tested and the impact of the other hardware components within the hypervisor on the performance of the SUT MUST be documented. Virtual component performance is influenced by the physical hardware components within the hypervisor. Access to various offloads such as TCP segmentation

Legend
VM: Virtual Machine
VW: Virtual Wire

Figure 3 Inter-Host System Under Test
offload, may have significant impact on performance. Firmware and
driver differences may also significantly impact results based on
whether the specific driver leverages any hardware level offloads
offered. Hence, all physical components of the physical server
running the hypervisor that hosts the virtual components MUST be
documented along with the firmware and driver versions of all the
components used to help ensure repeatability of test results. For
example, BIOS configuration of the server MUST be documented as some
of those changes are designed to improve performance. Please refer
to Appendix A for a partial list of parameters to document.

5.1. Server Architecture Considerations

When testing physical networking components, the approach taken is
to consider the device as a black-box. With virtual infrastructure,
this approach would no longer help as the virtual networking
components are an intrinsic part of the hypervisor they are running
on and are directly impacted by the server architecture used.
Server hardware components define the capabilities of the virtual
networking components. Hence, server architecture MUST be
documented in detail to help with repeatability of tests. And the
entire hardware and software components become the SUT.

5.1.1. Frame format/sizes within the Hypervisor

Maximum Transmission Unit (MTU) limits physical network component’s
frame sizes. The most common max supported MTU for physical devices
is 9000. However, 1500 MTU is the standard. Physical network
testing and NFV uses these MTU sizes for testing. However, the
virtual networking components that live inside a hypervisor, may
work with much larger segments because of the availability of
hardware and software based offloads. Hence, the normal smaller
packets based testing is not relevant for performance testing of
virtual networking components. All the TCP related configuration
such as TSO size, number of RSS queues MUST be documented along with
any other physical NIC related configuration.

Virtual network components work closer to the application layer then
the physical networking components. Hence virtual network
components work with type and size of segments that are often not
the same type and size that the physical network works with. Hence,
testing virtual network components MUST be done with application
layer segments instead of the physical network layer packets.

5.1.2. Baseline testing with Logical Switch

Logical switch is often an intrinsic component of the test system
along with any other hardware and software components used for
testing. Also, other logical components cannot be tested independent of the Logical Switch.

5.1.3. Repeatability

To ensure repeatability of the results, in the physical network component testing, much care is taken to ensure the tests are conducted with exactly the same parameters. Parameters such as MAC addresses used etc.,

When testing NPV components with an application layer test tool, there may be a number of components within the system that may not be available to tune or to ensure they maintain a desired state. Example: housekeeping functions of the underlying Operating System.

Hence, tests MUST be repeated a number of times and each test case MUST be run for at least 2 minutes if test tool provides such an option. Results SHOULD be derived from multiple test runs. Variance between the tests SHOULD be documented.

5.1.4. Tunnel encap/decap outside the hypervisor

Logical network components may also have performance impact based on the functionality available within the physical fabric. Physical fabric that supports NVO encap/decap is one such case that may have an impact on the performance observed. Any such functionality that exists on the physical fabric MUST be part of the test result documentation to ensure repeatability of tests. In this case SUT MUST include the physical fabric

5.1.5. SUT Hypervisor Profile

Physical networking equipment has well defined physical resource characteristics such as type and number of ASICs/SoCs used, amount of memory, type and number of processors etc., Virtual networking components performance is dependent on the physical hardware that hosts the hypervisor. Hence the physical hardware usage, which is part of SUT, for a given test MUST be documented. Example, CPU usage when running logical router.

CPU usage changes based on the type of hardware available within the physical server. For example, TCP Segmentation Offload greatly reduces CPU usage by offloading the segmentation process to the NIC card on the sender side. Receive side scaling offers similar benefit on the receive side. Hence, availability and status of such hardware MUST be documented along with actual CPU/Memory usage when the virtual networking components have access to such offload capable hardware.
Following is a partial list of components that MUST be documented both in terms of what is available and also what is used by the SUT:

* CPU - type, speed, available instruction sets (e.g. AES-NI)
* Memory - type, amount
* Storage - type, amount
* NIC Cards - type, number of ports, offloads available/used, drivers, firmware (if applicable), HW revision
* Libraries such as DPDK if available and used
* Number and type of VMs used for testing and
  o vCPUs
  o RAM
  o Storage
  o Network Driver
  o Any prioritization of VM resources
  o Operating System type, version and kernel if applicable
  o TCP Configuration Changes - if any
  o MTU
* Test tool
  o Workload type
  o Protocol being tested
  o Number of threads
  o Version of tool
* For inter-hypervisor tests,
  o Physical network devices that are part of the test

! Note: For inter-hypervisor tests, system under test is no longer only the virtual component that is being tested but the entire fabric that connects the
6. Security Considerations

Benchmarking activities as described in this memo are limited to technology characterization of a Device Under Test/System Under Test (DUT/SUT) using controlled stimuli in a laboratory environment, with dedicated address space and the constraints specified in the sections above.

The benchmarking network topology will be an independent test setup and MUST NOT be connected to devices that may forward the test traffic into a production network, or misroute traffic to the test management network.

Further, benchmarking is performed on a "black-box" basis, relying solely on measurements observable external to the DUT/SUT.

Special capabilities SHOULD NOT exist in the DUT/SUT specifically for benchmarking purposes. Any implications for network security arising from the DUT/SUT SHOULD be identical in the lab and in production networks.

7. IANA Considerations

No IANA Action is requested at this time.

8. Conclusions

Network Virtualization Platforms, because of their proximity to the application layer and since they can take advantage of TCP stack optimizations, do not function on packets/sec basis. Hence, traditional benchmarking methods, while still relevant for Network Virtualization Platforms, are not designed to test Network Virtualization Platforms. Also, advances in application architectures such as micro-services, bring new challenges and need benchmarking not just around throughput and latency but also around scale. New benchmarking methods that are designed to take advantage of the TCP optimizations or needed to accurately benchmark performance of the Network Virtualization Platforms.

9. References

9.1. Normative References

9.2. Informative References

Appendix A. Partial List of Parameters to Document

A.1. CPU

CPU Vendor
CPU Number
CPU Architecture
# of Sockets (CPUs)
# of Cores
Clock Speed (GHz)
Max Turbo Freq. (GHz)
Cache per CPU (MB)
# of Memory Channels
Chipset
Hyperthreading (BIOS Setting)
Power Management (BIOS Setting)
VT-d

A.2. Memory

Memory Speed (MHz)
DIMM Capacity (GB)
# of DIMMs
DIMM configuration
Total DRAM (GB)

A.3. NIC

Vendor
Model
Port Speed (Gbps)
A.4. Hypervisor

Hypervisor Name
Version/Build
Based on
Hotfixes/Patches
OVS Version/Build
IRQ balancing
vCPUs per VM
Modifications to HV
Modifications to HV TCP stack
Number of VMs
IP MTU
Flow control TX (send pause)
Flow control RX (honor pause)
### A.5. Guest VM

<table>
<thead>
<tr>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guest OS &amp; Version</td>
<td></td>
</tr>
<tr>
<td>Modifications to VM</td>
<td></td>
</tr>
<tr>
<td>IP MTU Guest VM (Bytes)</td>
<td></td>
</tr>
<tr>
<td>Test tool used</td>
<td></td>
</tr>
<tr>
<td>Number of NetPerf Instances</td>
<td></td>
</tr>
<tr>
<td>Total Number of Streams</td>
<td></td>
</tr>
<tr>
<td>Guest RAM (GB)</td>
<td></td>
</tr>
</tbody>
</table>

### A.6. Overlay Network Physical Fabric

<table>
<thead>
<tr>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vendor</td>
<td></td>
</tr>
<tr>
<td>Model</td>
<td></td>
</tr>
<tr>
<td># and Type of Ports</td>
<td></td>
</tr>
<tr>
<td>Software Release</td>
<td></td>
</tr>
<tr>
<td>Interface Configuration</td>
<td></td>
</tr>
<tr>
<td>Interface/Ethernet MTU (Bytes)</td>
<td></td>
</tr>
<tr>
<td>Flow control TX (send pause)</td>
<td></td>
</tr>
<tr>
<td>Flow control RX (honor pause)</td>
<td></td>
</tr>
</tbody>
</table>

### A.7. Gateway Network Physical Fabric

<table>
<thead>
<tr>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vendor</td>
<td></td>
</tr>
<tr>
<td>Model</td>
<td></td>
</tr>
<tr>
<td># and Type of Ports</td>
<td></td>
</tr>
<tr>
<td>Software Release</td>
<td></td>
</tr>
<tr>
<td>Interface Configuration</td>
<td></td>
</tr>
<tr>
<td>Interface/Ethernet MTU (Bytes)</td>
<td></td>
</tr>
</tbody>
</table>
Flow control TX (send pause)

Flow control RX (honor pause)