IETF 90: VNF PERFORMANCE BENCHMARKING METHODOLOGY

Contributors:

Sarah Banks:sbanks@akamai.com

Muhammad Durrani: mdurrani@brocade.com

Mike Chen: mchen@brocade.com

Objective

Create comprehensive VNF performance test methodology that provides underlying resource requirements and accurately predicts real world deployment performance

Problem Statement

A) Mix and match server hardware and hypervisors creates high degree of variation between server builds e.g. clock speed, cores, memory, HV etc.

B) Real world VNF performance is directly linked to hardware performance, resource allocation and HV configuration/capabilities (for VM deployments)

C) Considering A and B benchmarking the same VNF s/w performance would be expected to vary on different server builds with different resource assignments

VNF performance benchmark testing requires a method of identifying and removing h/w bottlenecks wherever possible to isolate VNF s/w performance

Solution Approach

- Begin with data plane VNF workload extend to others in future
- Leverage existing applicable benchmark RFC methodologies wherever possible

  2544 – Benchmarking Methodology for Interconnected Devices

  5180 - IPv6 Benchmarking Methodology for Network Interconnect Devices

  3918 – Methodology for IP Multicast Benchmarking

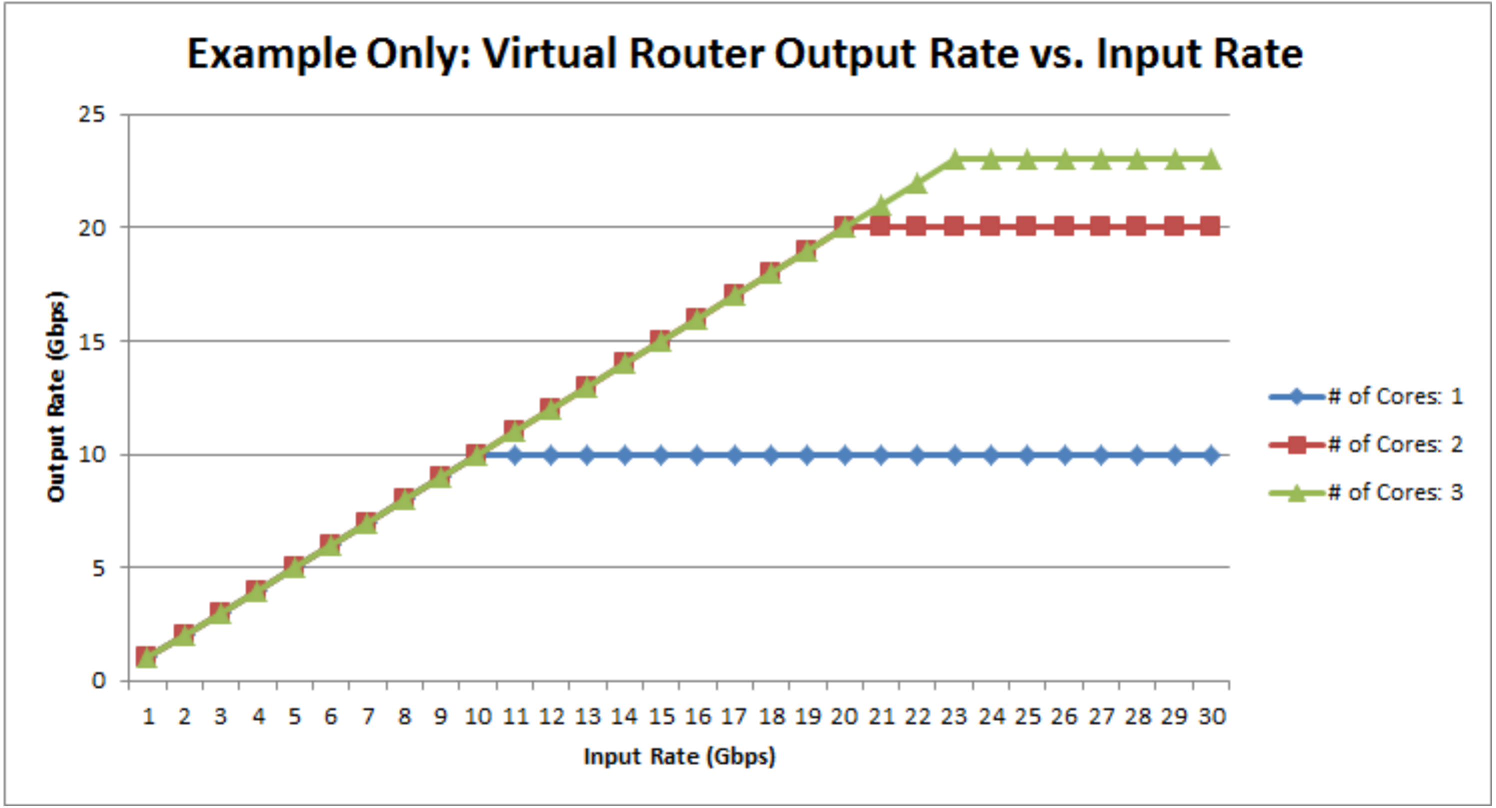
- Develop new methodologies /propose amendments for gaps with existing benchmark RFCs ←previous WG comment

[Main focus on 2544 as other RFCs refer back to it]

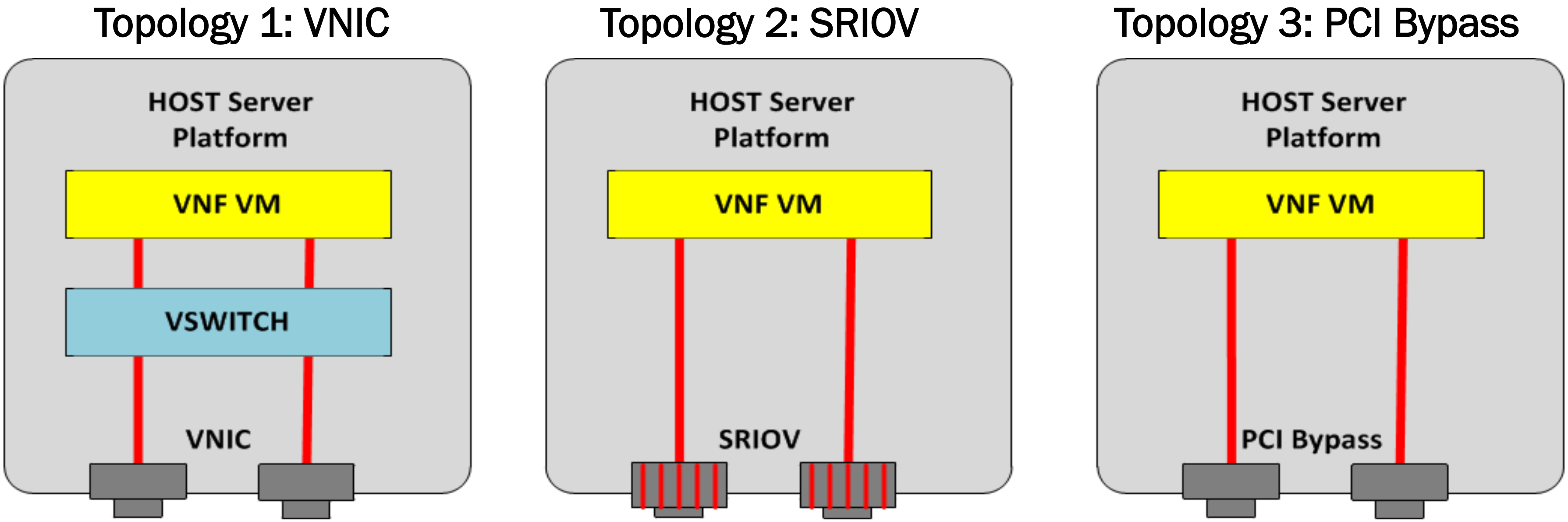
GAP Summary

RFC Reference	GAP/Proposed Ammendment
Testing Constant Resources RFC 2544 **7. DUT Set up**	**VNF can be deployed as VM with flexible h/w resource assignment** Incremental resource assignment approach used
Low count of Destination Networks RFC 2544 **12. Protocol addresses (256)**	**CPU cache destination lookup vs. Main Memory lookup performance difference not exposed with low destination network count** Destination networks increased in chunks
Multiple protocols not used RFC 2544 **17. Multiple protocols**	**CPU is in data path and additional usage from protocols may impact forwarding performance** CPU load introduced by inclusion of mixed protocols e.g. BFD, ARP, IGP/EGP routing updates, authentication, L2 protocols etc
Light Management Polling RFC 2544 11.2 Management frames	**CPU is in data path and heavy management polling may impact forwarding performance** Real management polling stations using SNMP, NETCONF, SSH with custom scripts etc to access DUT [OOB preferred] Mix, scale, polling and frequency for above determined by deployment needs or use case environment e.g. TOR, GW etc.

Throughput Example Results



Benchmark Topologies



Directly connected traffic generator ports used for TX/RX traffic [min 2 per topology]

Common Items

Common per test iterations include:

- Unicast IPv4
- Unicast IPv6
- Combination of IPv4/IPv6 Unicast with % as shown in RFC 5180
- Multicast IPv4 with IGMP joins
- Multicast IPv6 with MLD joins
- Frame sizes from RFC 2544 & Jumbo
- IMIX RFC 6985 (based on 2544 sizes or custom based on deployment req)

Received Traffic Validation: Sequence, TTL decrement, Egress interface selection

GAP Analysis: RFC 2544 A)

CPU core assignment testing

- RFC 2544 Section 7: DUT considered a system with fixed resources
- GAP: VNF can be deployed as VM with flexible h/w resource assignment
- Proposed Amendments:

  Single source, Single Dest/group forwarding performance testing performed iteratively with incremental CPU core assignments per iteration

  Packet sizes used for each additional CPU core added:

    a) RFC 2544 packet sizes & Jumbo

    b) IMIX sizes with sequence made up from above or custom based on deployment use case

    *Above sizes should be characterized with bare metal also

GAP Analysis: RFC 2544 A)

CPU core assignment results

Results/Data points specific to this test per traffic type:

- Maximum forwarding rate vs. Frame size [one data plot per core assignment]
- Maximum forwarding rate for IMIX sizes vs. CPU cores assigned [single data plot]
- Resource Assignment A: Minimum cores required for rate X with minimum packet size
- Resource Assignment AI: Minimum cores required for rate X with IMIX packet sizes
- Average CPU % utilization for iteration duration at min/middle / max number of cores assigned in testing
- Latency RFC 1242 and Delay Variation RFC 3393 at min/middle / max number of cores assigned in testing

GAP Analysis: RFC 2544 B)

Destination Range Testing

- RFC 2544 Section 12 indicates 256 destination address networks used at random
- GAP: CPU cache destination lookup vs. Main Memory lookup performance difference may not be exposed with low destination network count (applicable to bare metal and VM)
- Proposed amendments:

Iterative forwarding performance test increasing # of destination addresses in blocks e.g. 1K

Packet sizes used for each destination network block increase

a) Minimum packet size and resource starting point: assignment A(min)

b) IMIX sizes with standard sequence or custom based on deployment use case, resource starting point A(imix)

Incremental cores added to A(min) and A(imix) if performance decreases from single destination/group results

GAP Analysis: RFC 2544 B)

Destination Range Test Results

- Results/Data points specific to this test per traffic type:
  - Maximum forwarding rate vs. Unique Destinations [min and IMIX packet sizes plotted]
  - Resource Assignment B(min): Minimum cores needed to forward at rate X with minimum packet sizes to scaled destinations + sufficient main memory for all destination addresses
  - Resource Assignment B(imix): Minimum cores needed to forward at rate X with IMIX packet sizes to scaled destinations + sufficient memory for all destination addresses
  - Average CPU usage % for iteration duration taken with at min/middle/max destinations
  - Latency RFC 1242 and Delay Variation RFC 3393 at min/middle / max number of cores assigned in testing

GAP Analysis: RFC 2544 C)

Real World Background Event Testing

- RFC 2544: section 11 includes modifiers that are generally applicable and section 11.2 indicates 1 management query per second (in-band) section 17 indicates mixed protocol environments are not addressed
- GAP: Explicit testing with DUT CPU loading, in VNF case CPU is in the data path and additional usage may impact forwarding performance [bare metal & VM]
- Proposed Amendments:

Repeat destination address range test with additional CPU loading from:

Inclusion of mixed protocols e.g. BFD, ARP, IGP/EGP routing updates, authentication, L2 protocols etc

Real management polling stations using SNMP, NETCONF, SSH with custom scripts etc to access DUT [OOB preferred]

Mix, scale, polling and frequency for above determined by deployment needs or use case environment e.g. TOR, GW etc.

Noisy Neighbor – Misbehaving VM on same host

GAP Analysis: RFC 2544 C)

Real World Background Event Test Results

- Results/Data points specific to this test per traffic type:
  - Maximum forwarding rate vs. Unique Destinations [min and IMIX packet sizes plotted]
  - Resource Assignment C(min): Minimum cores needed to forward at rate X with minimum packet sizes to scaled destinations + sufficient main memory for all destination addresses
  - Resource Assignment C(imix): Minimum cores needed to forward at rate X with IMIX packet sizes to scaled destinations + sufficient memory for all destination addresses
  - Average CPU usage % for iteration duration taken with at min/middle/max destinations
  - Latency RFC 1242 and Delay Variation RFC 3393 at min/middle / max number of cores assigned in testing

BACKUP

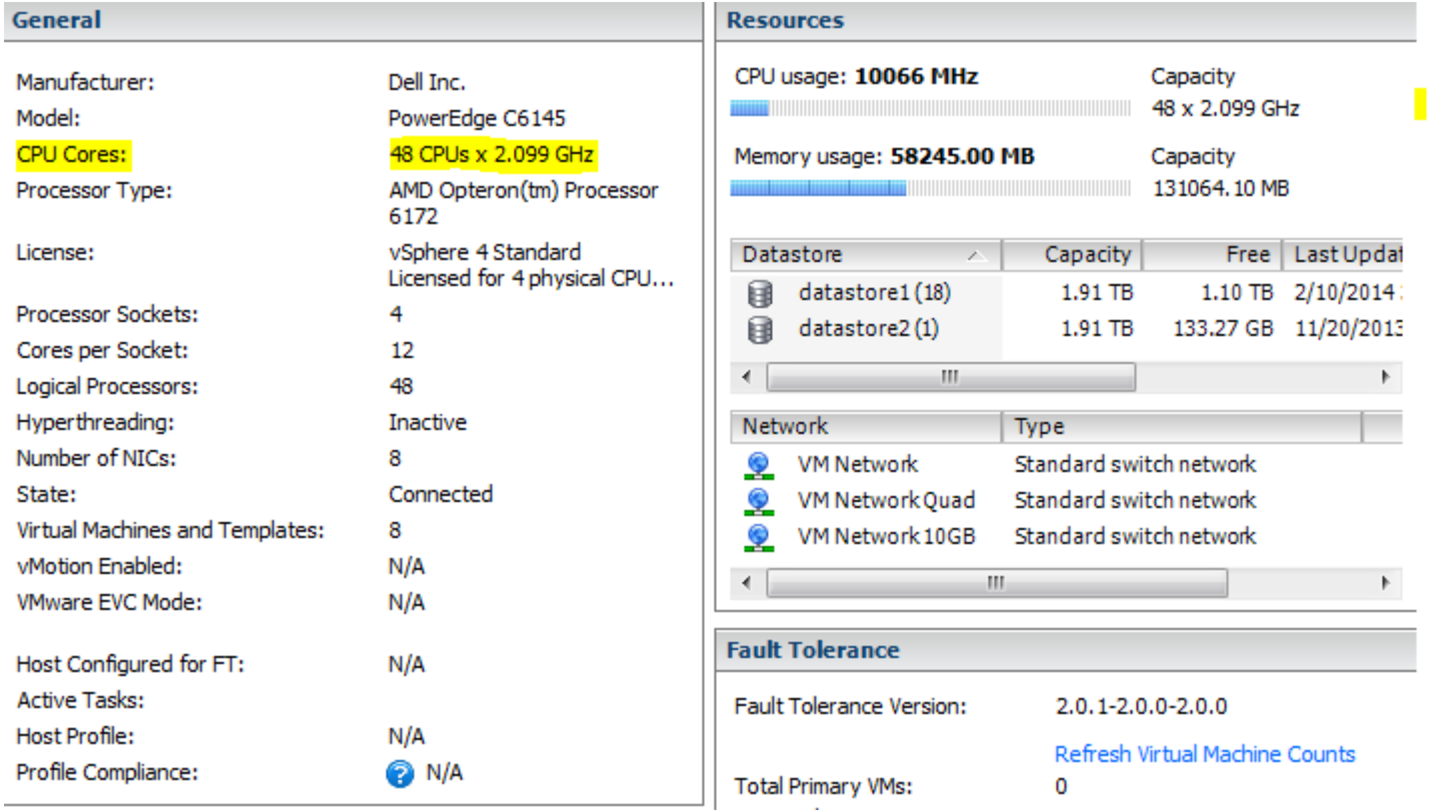
Forwarding Performance Summary Matrix

	Virtual Router Configuration & Resource Assignment									
Traffic Forwarding Performance: a) Maximum throughput for minimum & IMIX packet sizes b) Latency c) Delay Variation L3 forwarding verification: TTL decrement, CRC calculation, Egress interface selection, sequence	Virtual Router on Bare Metal: steady state	Virtual Router on Bare Metal: common background events (e.g. route churn)	Virtual Router VM: Single CPU core assigned - steady state	Virtual Router VM: Single CPU core assigned with common background events (e.g. route churn)	Virtual Router VM: N CPU cores assigned - steady state [incremental core increase to remove CPU bottleneck]	Virtual Router VM: N core assigned with common background events (e.g. route churn)	Virtual Router VNF: Single CPU core assigned - steady state	Virtual Router VNF: Single CPU core assigned with common background events (e.g. route churn)	Virtual Router VNF: N CPU cores assigned - steady state [incremental core increase to remove CPU bottleneck]	Virtual Router VNF: N core assigned with common background events (e.g. route churn)
Unicast Traffic Forwarding										
Unicast IPv4: Single source single destination (CPU cache lookup)										
Unicast IPv4: Single source scaled destination (Main memory lookup)										
Unicast IPv6: Single source single destination (CPU cache lookup)										
Unicast IPv6: Single source scaled destination (Main memory lookup)										
Unicast IPv4: Single Bi-directional flow (CPU cache lookup)										
Unicast IPv4: Scaled Bi-directional flows (Main memory lookup)										
Multicast Traffic Forwarding [Connected Hosts]										
Multicast IPv4 : Single source to single group with directly connected hosts [IGMP joins]										
Multicast IPv4 : Multiple source to single group with directly connected hosts [IGMP joins]										
Multicast IPv4 : Single source to scaled groups with directly connected hosts [IGMP joins]										
Multicast IPv4 : Multiple source to scaled groups with directly connected hosts [IGMP joins]										
Multicast IPv6 : Single source to single group with directly connected hosts [MLD joins]										
Multicast IPv6 : Multiple source to single group with directly connected hosts [MLD joins]										
Multicast IPv6 : Single source to scaled groups with directly connected hosts [MLD joins]										
Multicast IPv6 : Multiple source to scaled groups with directly connected hosts [MLD joins]										

Server Based Hardware Acceleration

- Virtual routers deployed on servers with hardware acceleration e.g. hardware based forwarding via TCAM can and should have baseline performance evaluated via traditional methods

Multicore Processor in HV



THANK YOU