NFV Network and Compute Intensive H/W Acceleration (using SDN/PI forwarding)

World Class Standards

NFV PoC (Proof of Concept) #21

http://nfvwiki.etsi.org/index.php?title=Network_Intensive_and_Compute_Intensive_Hardware_Acceleration

peter.ashwoodsmith@huawei.com (presenter)
evelyne.roch@huawei.com (contact for info/join etc.)

What

- Multiple vendors collaborating to show different H/W acceleration mechanisms for NVF (list of vendors is on the POC link).
- Demonstrations were given at the recent SDN and OpenFlow world congress of work so far.
- A series of independent tests prior to the conference were also run.
- This is ongoing work, happy to include others that are interested as POC evolves.
- APIs should be of interest to those doing SDN/NFV related research, standards and open source.

Overview

- **Why accelerate VNFs with H/W?**
  - **Reduced required compute resources (S/W 250G-> H/W 1T)**
  - **Reduced power/cooling (S/W 350W @250G-> H/W 170W@1T, less noise!)**
  - **Reduced space (~5x reduced density – 5U to 1U)**
  - **Reduced latency/jitter (skip a hop into / out of CPU + CPU latency)**
  - **Predictable performance (CPU usage changes less impact on perf.)**
  - **Increased security (H/W accelerated harder to snoop)**

- **Challenge is how to get these gains while still using common platforms?**
  - **After all entire point of NFV is to use common servers! Goal is not to go back to where we were pre-NFV.**
  - **SDN can help if we can split VNF into low and high performance parts. VNF and VNF' and then use/define common API's between VNF and VNF'**

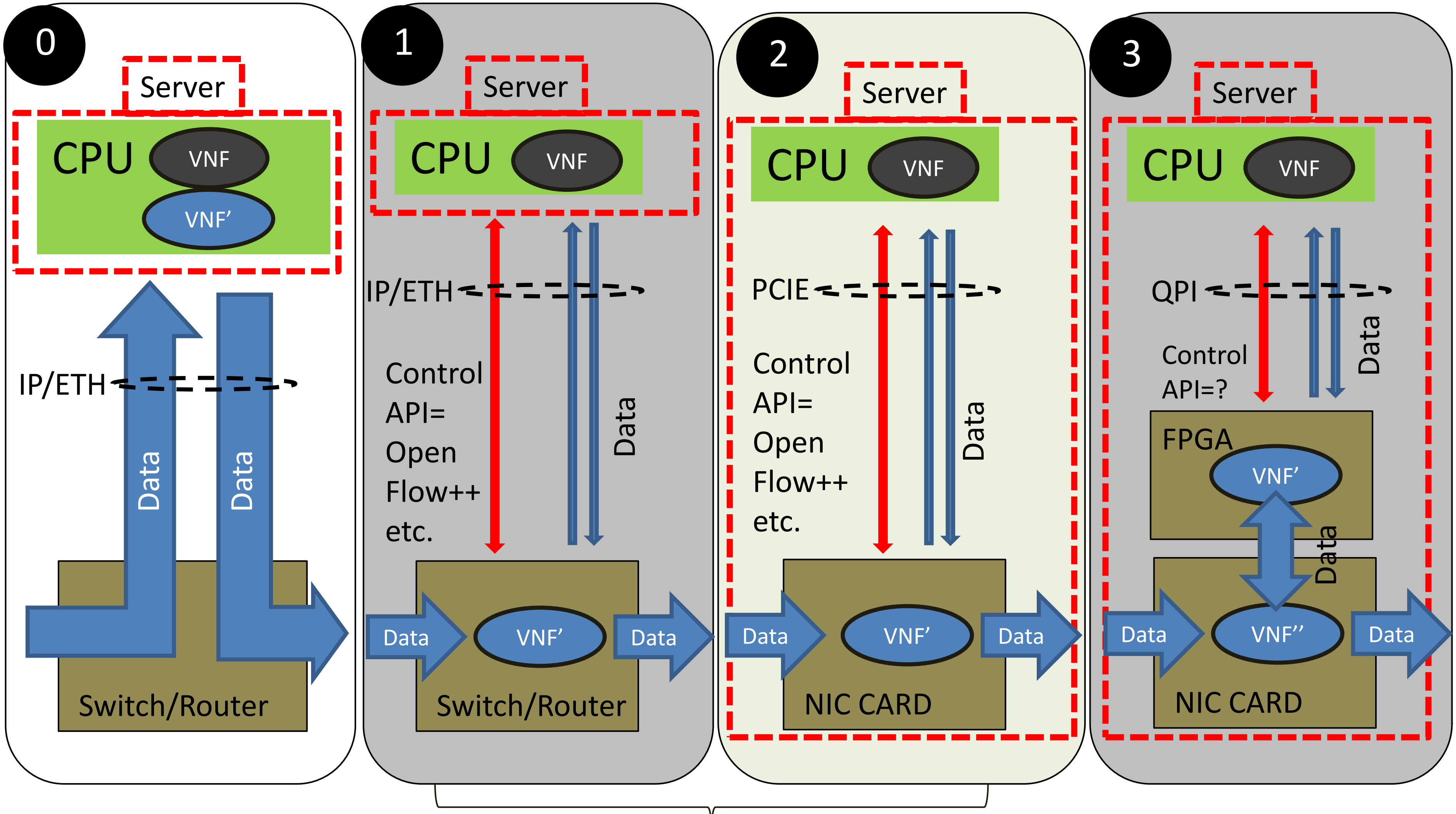
NFV Acceleration Options

VNF’ in CPU i.e. none

VNF’ in Adjacent Switch

VNF’ In Server / NIC

VNF’ in Hybrid FPGA



ETSI POC focused on these two

ETSI Proof Of Concept Overview

- Demonstrated how Network and Compute intensive virtual network functions can take advantage of existing or new H/W to increase performance dramatically.

- Network intensive VNF use case: ❶ ❷ {server & NIC}
  - Acceleration of L7 Load Balancer
    - Demonstrate Benefits of Dynamic Optimization of Packet Flow Routing.
    - **Identify required APIs and OpenFlow extensions.**

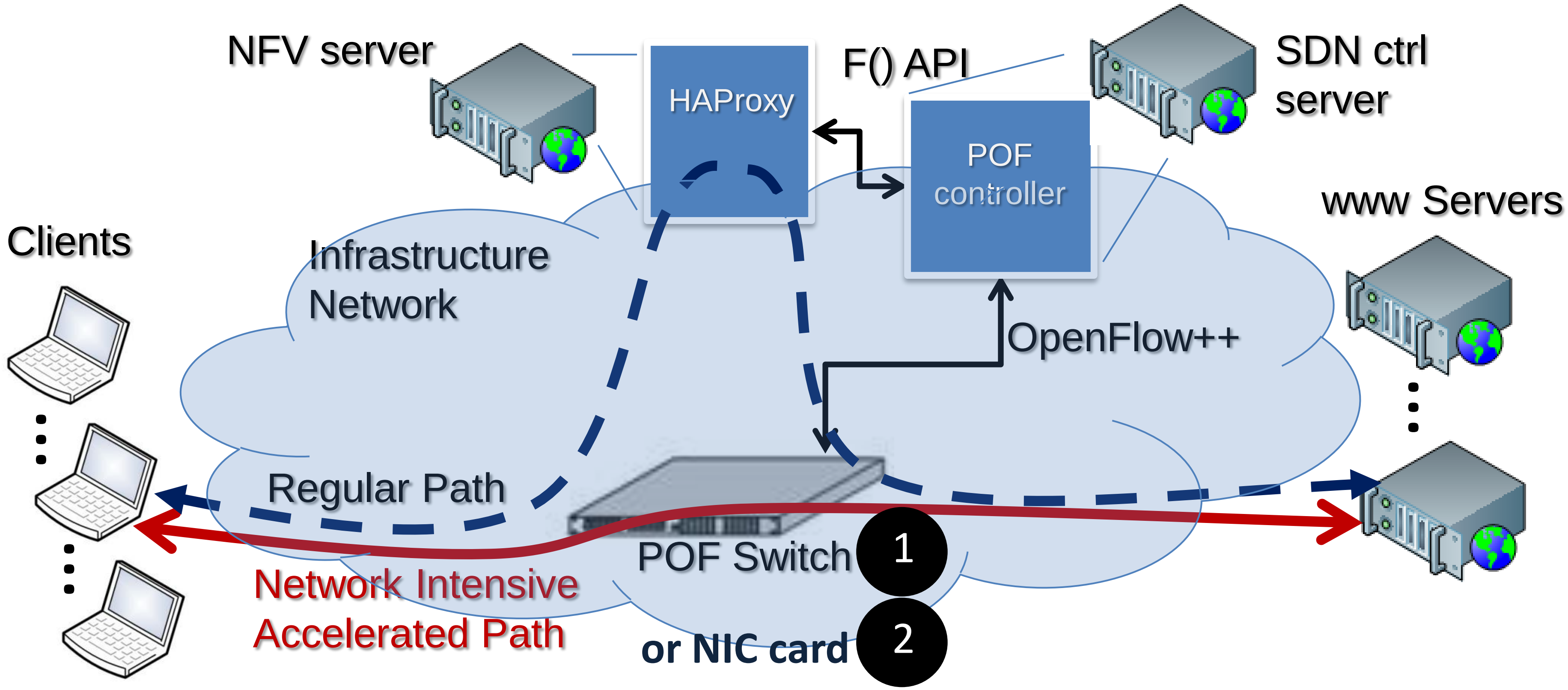
- Compute intensive VNF use case:
  - IPSec authentication and Encryption/decryption Acceleration
    - Address carrier use case for deployment of Open Wi-Fi and LTE Authentication and Encryption Services. ❷ {NIC}
    - **Identify required Hardware Acceleration API extensions**.

- Employed Protocol Oblivious/Independent Forwarding (POF/PIF) http://www.poforwarding.org/ to program the H/W .. i.e. OF++.

- ONF document ONF2014.451 OF PI: A Protocol Independent Layer describes the benefits of protocol independent packet processing and the role of POF/PIF in next generation of OpenFlow.

Network Intensive VNF Acceleration - L7 Load Balancer

- Modified HAProxy in order to allow it to invoke POF controller APIs.
- Upon receiving new requests, HAProxy invokes POF controller APIs and instructs the underlying H/W to manipulate TCP Seq. and Ack. on both client and server sides and splice the TCP sessions (can't do with vanilla OpenFlow, hence PI forwarding).



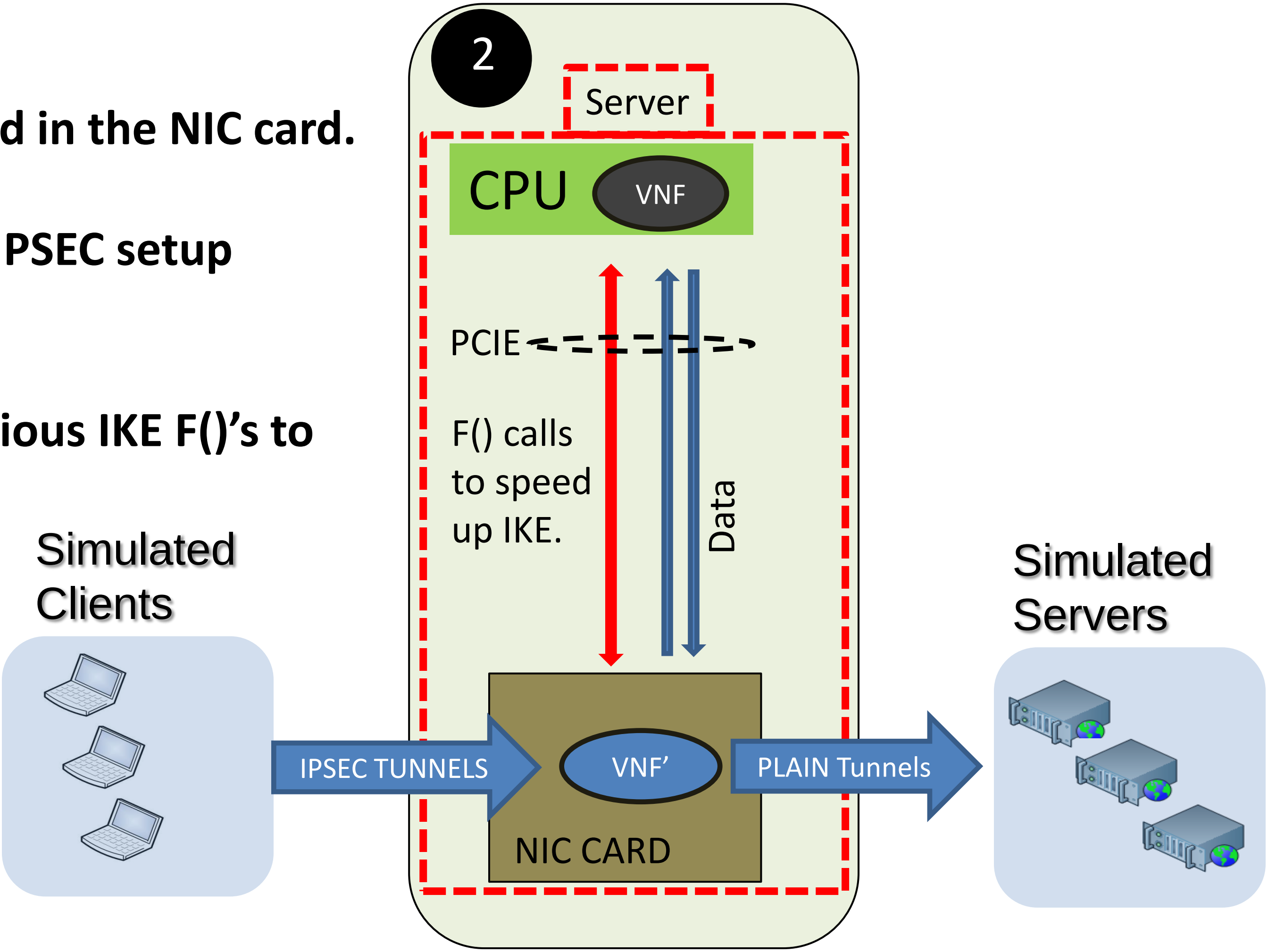
Demonstrated several different implementations from several different vendors.

Network Intensive VNF Acceleration - Results

- Dynamic Optimization of Packet Flow Routing
  - Can be implemented on NIC or Infrastructure Switch
  - Decreases CPU utilization by 50 - 90%
  - Decreases bandwidth utilization of the link between server and infrastructure switch dramatically.
  - 8 to 25 times increase in demonstrated throughput delivered by VNF:
    - Limit now becomes NIC or infrastructure switch capacity
- Enhanced resiliency
  - Existing flows maintained in case of VNF failure
- Decreased Latency
  - One hop (VNF) removed from end-to-end path

Network Intensive VNF Acceleration – IPSec Gateway

- Modified OpenSwan in order to allow the use of H/W acceleration APIs.
- Analysis of run time and replacement of high runner F()s with FPGA via API.
- FPGA was embedded in the NIC card.
- Initially focused on IPSEC setup rates.
- So we offloaded various IKE F()'s to FPGA.
- Other vendors did different offloads and architecture.



Compute Intensive VNF Acceleration - Results

- Tests executed and hosted by The European Advanced Networking Test Center (EANTC) in their lab in Berlin.
  http://www.eantc.de/fileadmin/eantc/downloads/test_reports/2014/EANTC-ETSI_NFV_PoC21-v3.0.pdf

- Authentication acceleration
  - Can be performed by different types of hardware accelerators
  - Increases transaction rates per server from 160 sessions/s to 2K – 4K sessions/s for 14K tunnels.
  - Number of tunnels could go well beyond 14K.
  - 14K limit was due to unavailability of equipment in test environment.

- Encryption/Decryption acceleration
  - We did not do this but other vendors did, but clearly same approach moving the encrypt/decrypt F()'s to FPGA with common API will have significant throughput results but requires standard API's.

Next Steps in the ETSI POC

- **Generalize the acceleration concepts to other VNFs**
  - **Video trans-coding.**
  - **Service chaining.**
  - **Others based on interest.**

- **Standardize protocol extensions for SDN Controller for higher performance packet forwarding capabilities.**
  - **OpenFlow ++ (Protocol Independent/Agnostic additions)**

- **Standardize API's for higher performance packet compute termination etc. type capabilities.**
  - **Where to do this work, open to suggestions.. openNfv?**

References

- **http://nfvwiki.etsi.org/index.php?title=Network_Intensive_and_Compute_Intensive_Hardware_Acceleration**
- http://www.poforwarding.org/
- http://www.eantc.de/fileadmin/eantc/downloads/test_reports/2014/EANTC-ETSI_NFV_PoC21-v3.0.pdf
- ONF2014.451 OF PI: A Protocol Independent Layer