EE542 - Reading Assignment – 05

UNO: Unifying host and smart NIC offload for flexible packet processing

Presenter: Boyang Xiao

USC id: 3326-7302-74

Email: boyangxi@usc.edu

Index

- Paper’s main ideas
- Background and motivations
- UNO architecture
- UNO implementation
- UNO evaluation
- Conclusions
- Q&A

Paper's main ideas

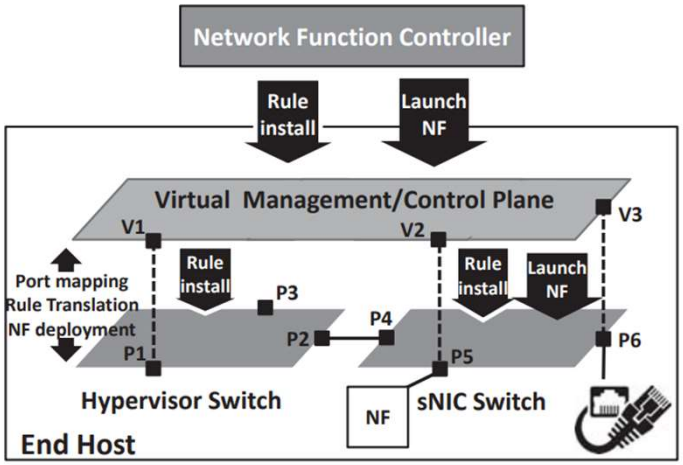
- This paper proposes a generalized SDN-controlled NF offload architecture called **UNO**
- UNO can transparently offload dynamically selected host processors' packet processing functions to sNICs by using multiple switches in the host while keeping the data center-wide network control and management planes unmodified.
- UNO can make optimal use of host's and sNIC's combined packet processing capabilities with local optimization based on locally observed traffic patterns and resource consumption, and without central controller involvement.

Background and motivations

- Network Interface Cards (NICs)
  - Traditional NICs provide several pre-packaged functions to offload routine packet processing to the NIC
  - Advanced NICs (ASIC) equipped with special hardware are purpose built to offload pre-defined packet processing functions
  - Smart NIC (sNIC) refers to a NIC equipped with fully-programmable, system-on-chip multi-core processor on which a full-fledged operating system can execute any arbitrary packet processing functions
- UNO motivation
  - There is no general framework to easily and intelligently offload and/or service chain the functions across hosts and sNICs.
  - We have to seek a design that preserves the benefit of flexible NF placement across both the host and the sNIC, but minimizes the complexity exposed to the data center controller.

UNO architecture

- **UNO** is a framework that systematically and dynamically selects the best combination of host and sNIC processing for NFs using local state information and without requiring central controller intervention
- Two components in UNO:
  - Network function agent: for the management plane (virtualization management)
  - OneSwitch: for the control plane



UNO architecture

- Network Function Agent (NF agent)
  - NF agent is responsible for launching VM/NF instances and configuring OneSwitch (e.g., creating ports) according to management plane policies
  - On the host side, it incorporates additional intelligence to decide whether to deploy NF instances, on a hypervisor or sNIC
  - NF placement decision problem is formulated as an integer linear programming (ILP). And for the problem instance sizes that arise for our application, we can efficiently find the optimal solution using off-the-shelf ILP solvers
  - The NF agent maintains a topology of NFs and VMs in the host and sNIC, as well as resource requirements (hi and ni) of each NF/VM i. When a new NF instance needs to be deployed on the host, NF agent runs the placement algorithm based on the current information

UNO architecture

- OneSwitch
  - OneSwitch constructs a single virtual data plane using the virtual ports created by the NF agent, and exports this virtual data plane to the controller.
  - Two solutions to the rule translation algorithm
    1. **Port-map based rule translation**: We describe our basic approach to translate a virtual rule into a set of physical rules by using virtual-to-physical port mappings (port-map).
    2. **Pitfalls and solutions**: While the above port-map based rule translation may seem straightforward, ambiguity can arise when multiple virtual rules co-exist on the virtual data plane.
  - In UNO, the NF migration must satisfy two requirements. It needs to be done transparently without involving the SDN controller, and without incurring packet loss during migration. Also, during migration it is important to ensure all in-flight packets are processed, and updates to NFs' internal state due to such packets are correctly reflected at the NF instance's new location.

UNO implementation

- NF Agent and OneSwitch
  - The NF agent exports APIs via which a centralized NFV platform can provision VM/NF instances and their port interfaces on a given host. This northbound interface largely borrows from the OpenStack Compute APIs
  - OneSwitch implementation is based on OpenVirteX (OVX) network virtualization software [34], which can perform basic control plane translation for network slicing.
- Hypervisor/sNIC Switches
  - In UNO architecture, hypervisor and sNIC switches are regular SDN switches controlled by OneSwitch, and thus we base their implementation on OVS.
  - To support rules generated by the UNO's rule translation algorithm, both switches need to handle per-packet flow-id metadata, and perform flow-id based flow matching and push/pop-flow-id actions.

UNO evaluation

- Benefit of offloading evaluation results:
  - Even when dedicated hardware acceleration is available for an NF for either platform (AES-NI/QAT on x86 host or MiCA on sNIC), sNIC's network function processing still consumes less energy than that of the x86 host.
  - Compared to sNIC and x86/AES-NI deployment, the PCIe bandwidth usage with x86/QAT deployment is more than doubled. sNIC deployment is more PCIe band-width efficient than x86/AES-NI deployment because the latter case needs to sustain additional PCIe bandwidth overhead of egress IPsec packets.
  - If UNO is adopted for the SD-WAN application, the number of x86 CPU cores are saved and can be used for tenant applications by offloading to TILE-Gx CPUs. Note also that UNO enables SECGW to be offloaded from TILE-Gx CPUs to the built-in crypto engine.
- Flow Rule translation evaluation results:
  - UNO can reduce the controller overhead by that much by distributing sNIC control on to individual end hosts

UNO evaluation

- Cost of NF migration evaluation results:
  - Deserialization on sNIC takes longer than serialization on x86 host due to the lower single-core performance of sNIC. The trend becomes the opposite when migrating from sNIC to x86 host. The average per-packet latency increases by 40–50ms during migration, compared to migration-free condition.
- Effect of NF placemen evaluation results:
  - When the maximum PCIe bandwidth is small (10Gbps), it is the bottleneck and UNO puts the NFs to the x86 host using more CPU cores. When the maximum PCIe bandwidth increases, UNO offloads more NFs to the sNIC, thus decreasing host CPU utilization while increasing sNIC CPU utilization. UNO's placement algorithm effectively leverages available PCIe bandwidth and sNIC's compute capacity to offload as much NF workload as possible.
  - UNO's current placement algorithm **does not** always improve the latency because its objective is to minimize x86 host resource usage.

Conclusions

- UNO: an SDN-controlled NF offload architecture
- UNO can transparently leverage the smart NIC's programmable compute capabilities to accelerate the NF data plane, and without introducing additional complexity in the data center's centralized management and control planes
- UNO's transparent offload is achieved by two per-host components:
  - the NF agent which intelligently chooses a subset of NFs to offload to the sNIC
  - OneSwitch which abstracts out the offloaded NF data planes from the data center's control plane
- The evaluation results demonstrate the feasibility and the substantial benefits of UNO

Q&A

- **Questions:** What are UNO's two key components and what are their functionalities, respectively?

- Answer:
  1. Net function agent: NF agent intelligently chooses a subset of NFs to offload to the sNIC
  2. OneSwitch: OneSwitch component abstracts out the offloaded NF data planes from the data center's control plane

Thanks for watching!

Presenter: Boyang Xiao

USC id: 3326-7302-74

Email: boyangxi@usc.edu