Research Project: Network Orchestration

A Stream Processing based NFV Platform for 5G on Modern Multicore Processors

PI: Zhang Shuhao (ISTD, SUTD)
Co-PI: Chen Binbin (ISTD, SUTD)
Collaborators:

In the foreseeable 5G era, the IoT ecosystem is expected to accommodate an unprecedented deluge of data traffic generated by hundreds of billions of heterogeneous interconnected devices. One of the key enabling technologies for 5G, network slicing promises to slice the physical network infrastructure into multiple self-contained, isolated, and programmable logical (or virtual) networks to indulge different genres of services demanded by assorted tenants. By replacing the purpose-built, expensive, proprietary network equipment with software network functions consolidated on commodity hardware, Network Function Virtualization (NFV) envisions a shift towards a more agile and open service provisioning paradigm, that is more suitable for the 5G era. To this end, the ETSI ISG NFV has proposed a standard NFV Management and Orchestration (MANO) framework. Many NFV platforms have been proposed in the research literature. However, existing NFV platforms were originally developed for cloud center services. A NFV platform for 5G faces several novel challenges to enable massive development, deployment and flexible management of performance critical IoT-based network services.

- First, there is an increasing demand for quick deployment of 5G mobile network services from the end-users prospective, rather than service providers prospective. Such user-centric services translate to a urgent demand of rapid development of new network services, where NFV frameworks with simple declarative APIs come into play.
- Second, the explosion in the number of connected devices requires the next generation NFV platform to handle enormous and rapid increasing data traffic with ultra-low latency and high-throughput.
- Third, there is a need to handle heterogeneous network slices with varying resource demands and SLA requirements for concurrently execution of many service function chains (SFCs) over 5G infrastructure.

This project will investigate a new approach to address those challenges by developing a novel stream processing based NFV platform for 5G on modern multicore processors. The use case scenario of the proposed NFV platform is shown in Fig.1, such as content distribution network (CDN), intrusion detection system, and security monitoring of vehicle-to-everything (V2X) communication. The envisioned new NFV platform will provide efficient and personalized network function services based on commodity servers. This eliminates the costs of replacing or reconfiguring customized network hardware in response to changing workloads and requirements. For instance, misbehavior detection and trust management for Vehicle-to-Everything (V2X) communication can identify falsified and malicious messages, enabling witness vehicles to report observations about high-criticality traffic events. With the help of a stream processing based NFV platform, we may develop and deploy various customized security monitoring network functions (e.g., firewall and deep packet inspection) on connected vehicles and nearby edge devices (e.g., APs) in a relatively easy and fast manner.

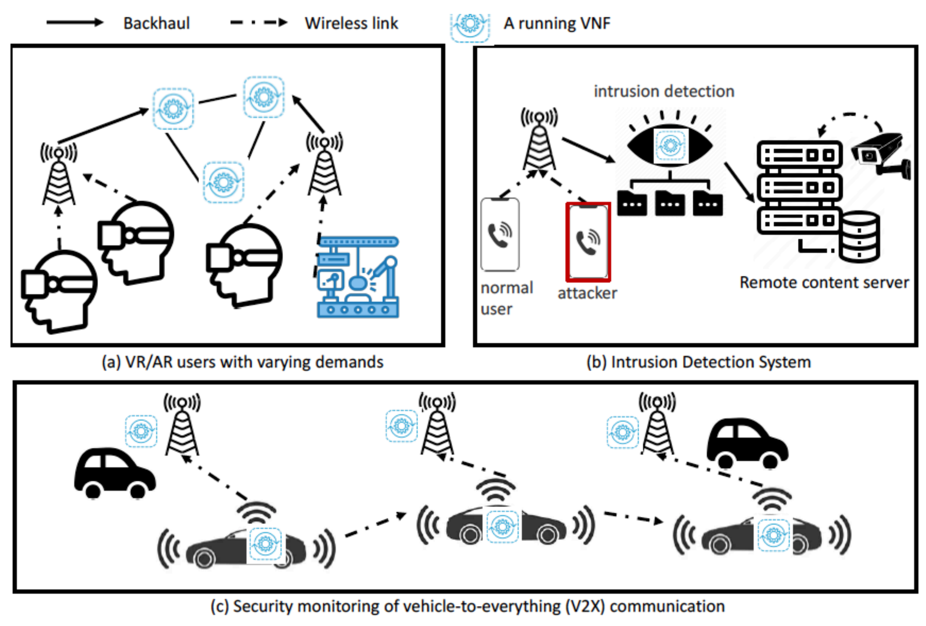

Figure 1 Applications of the proposed NFV platform

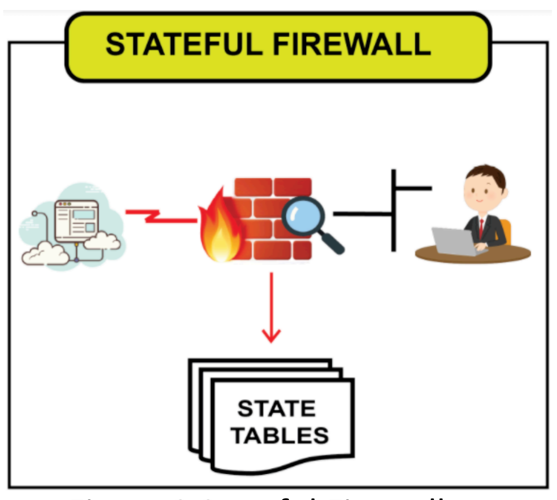

Figure 2 Stateful Firewall

Motivation: To enable network services to handle enormous data traffic with ultra-low latency and high-throughput with Commodity Off-The-Shelf (COTS) is non-trivial. While many existing NFV platforms manage to sustain 40/100 Gbps links using high-speed I/O techniques, their performances have not been tested under 5G/6G configurations. To achieve high scalability, our preliminary study revels that the main difficulty to scale up NFV platforms on multicore processors is to handle highly contended shared state access during VNF executions of immense number of connections. Fig.2 illustrates an overview of a typical stateful VNF -- stateful firewalls. The stateful firewalls screen the entire state of network connections and continually break down the total traffic and information packets setting, looking for a passage to a network instead of discrete traffic and data packets in isolation. It is a difficult task as the states need to be read or updated during the processing of flows of network packages. As such, we can potentially address such challenges by relying on the emerging *concurrent stateful stream processing* techniques. Our prior work has developed TStream[1], which is a novel stateful stream processing engine that is able to scale even under highly contended and dynamic workloads.

The proposed project aims at developing a novel NFV platform for 5G based on a novel stateful SPE – TStream via three key work packages: *(1) API Design; (2) Engine Implementation and Optimization; (3) Runtime Dynamic Control Plane*.

WP1: The first step of this project is hence to *design a way (e.g., APIs) to map VNF expressions to stream processing expressions, e.g., dataflow APIs.* The latter are widely used in stream processing engines (SPEs). Streaming abstraction describes complex streaming computations over time-series data, where packet flows are kinds of streams over the network. With this new understanding, multilayer VNFs can be described based on streaming abstraction. Streaming abstraction describes complex streaming computations over data streams. If we can model network packages as data streams, and model VNF as stream operators, we can utilize existing SPEs, such as Flink and Spark, with well-defined APIs to express complex VNFs and VNF service chains using high-level programming languages, e.g., python/Java. Fig.3 shows an example VNF to cumulatively count the packets called packet counter. We express it with the three-step dataflow API[1]. Note that, the state (count) will be automatically managed by our SPE to relief the burden of VNF developers. This is just one example advantage of utilizing SPE for NFV as our proposed project wish to demonstrate.

Algorithm 1: Packet Counter

```
state count;// number of packet counter from port1.
Function PRE_PROCESS():
    packet ← read and parse from ports;
Function CORE_FUNCTION():
    if packet.port == port1 then
        count ← count + 1;
Function POST_PROCESS():
    return packet; // forward packet to an out port.
```

Figure 3 Packet Counter Expressed in Three-Step Dataflow

WP2: A NFV platform for 5G must be able to achieve both 1) High Packet Rates: must keep up with line rate which is >10MPPS, and 2) Low Latency: used for applications like VoIP and video conferencing. The difficulty arises because most Network Functions (NFs) are stateful and this state need to be shared across NF instances. Developers must deal with the inherent trade-off of maintaining a consistent state shared across packet flows manipulated by multiple NF instances while processing packets at line rate. Due to diverse system requirements, such as managing state beyond main memory, elastic scaling, concurrent readable-and-writable states, and migrating states among shared-nothing architectures, SPEs were designed to be fully aware of state. Those SPEs with built-in state management support are known as stateful SPEs. We propose to rely on stateful SPEs to improve the execution efficiency of stateful VNFs on Multicores. In particular, we need to *implement representative stateful VNFs using the dataflow APIs on modern stateful SPEs, such as TStream*. Multiple questions such as which parallel processing paradigm to adopt will be addressed in the proposed project.

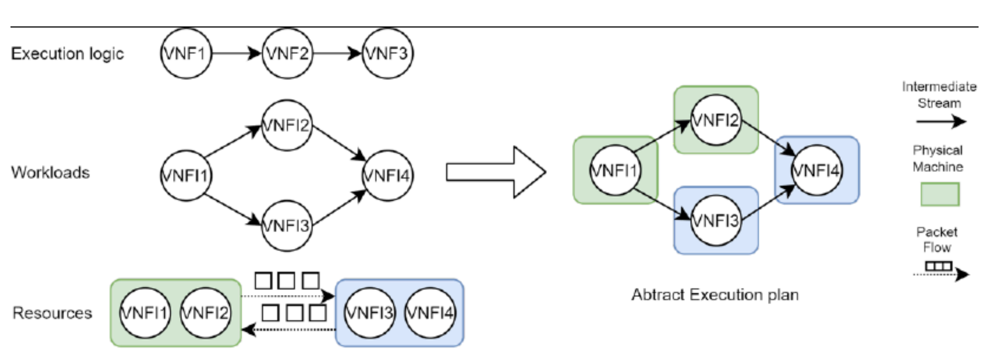

Figure 4 Control Plane Abstraction Example

WP3: We will further *propose a control plane to manage heterogeneous network slices with varying resource and SLA requirements* on top of our SPE-based NFV platform. Specifically, we are to extend it to manage services of heterogeneous network slices with a control plane providing 1) elastic scaling, 2) workload balancing, and 3) function hot swap capabilities. The control plane abstracts complicated NFV runtime and maintains the abstracted execution plan of SFC for high programmability shown in Fig.4

1: S. Zhang, Y. Wu, F. Zhang and B. He, "Towards Concurrent Stateful Stream Processing on Multicore Processors," 2020 IEEE 36th International Conference on Data Engineering (ICDE), 2020, pp. 1537-1548, doi: 10.1109/ICDE48307.2020.00136.