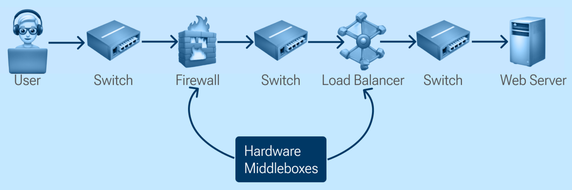
SERVICE FUNCTION CHAIN EMULATION USING OPENRASE

Theviyanthan K.,
School of Computing Science, University of Glasgow

HARDWARE MIDDLEBOXES

Hardware middleboxes are network functions such as firewall deployed on proprietary hardware.

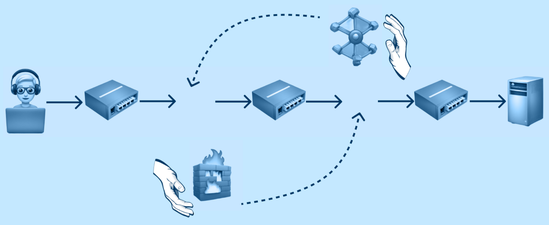


Disadvantages of Hardware Middleboxes

Expensive to buy (CAPEX)
You pay for both software and hardware.

Consumes more power
Each network function has its own hardware. So power consumption is high.

Modifying topology needs human intervention
In the above example, swapping firewall and load balancer has to be done physically as we are dealing with hardware. This makes network autonomy impossible.

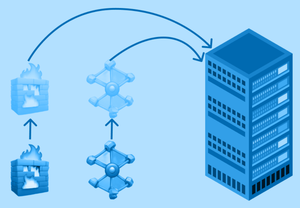


SERVICE FUNCTION CHAINING (SFC)

SFC addresses these disadvantages using NFV and SDN. This enables us to program networks and make them autonomous.

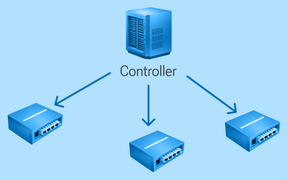
Network Function Virtualisation (NFV)

NFV virtualises network functions and deploys them on servers.



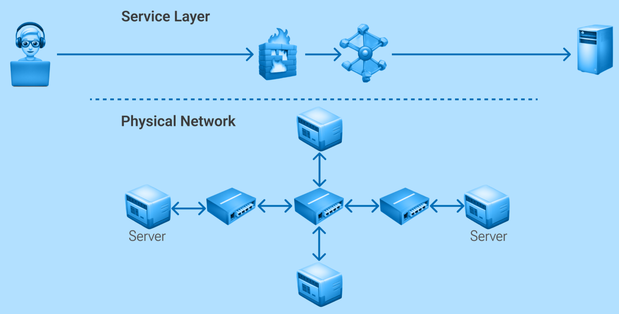
Software Defined Networking (SDN)

SDN allows network switches to be configured by a central controller allowing networks to be programmed.



Service Layer

SFC creates a service layer by using NFV to deploy network functions as Virtual Network Functions (VNF) on the physical network and then by using SDN to link them together.

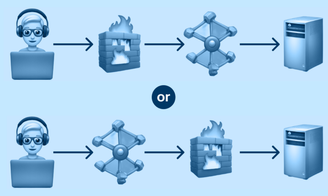


NFV RESOURCE ALLOCATION (NFV-RA)

But SFC comes with its own challenges. One of the challenges is the NFV-RA challenge. This consists of two issues:

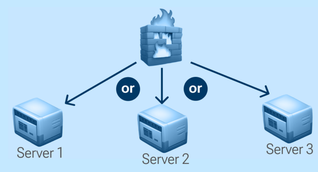
Chain Composition

Deals with the order of network functions in a chain. The order can impact the performance of the chain.



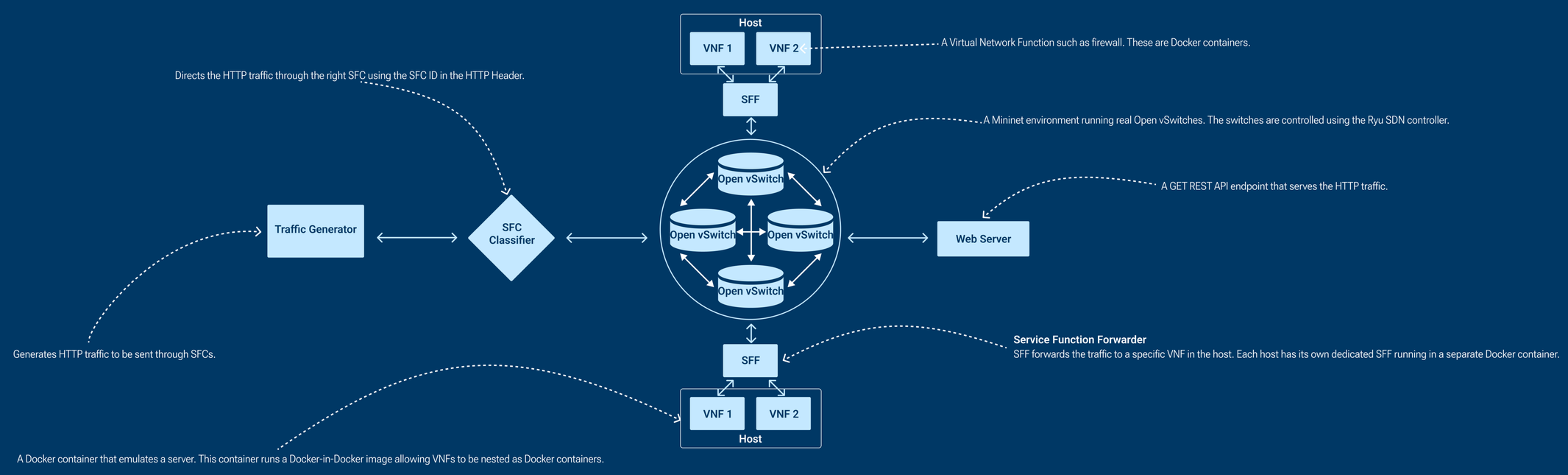
Forwarding Graph Embedding

Deals with where VNFs should be deployed. How VNFs are deployed across the servers available also impacts performance.



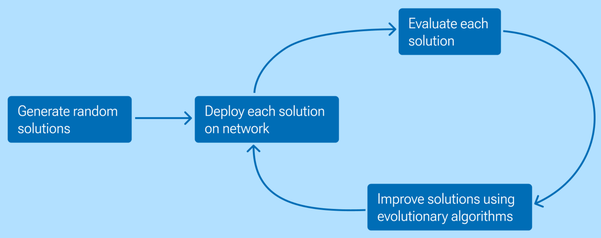
OPENRASE

OpenRASE is an emulator built on top of Docker and Mininet for NFV-RA experiments. It uses Docker containers to emulate servers, and real Open vSwitches for networking.



ONLINE EXPERIMENTAL EVOLUTION

My research goal is to solve NFV-RA through online experimental evolution.



RELATED WORK

Existing tools used to deploy NFV-RA solutions and evaluate them can be categorised as simulators, emulators and testbeds.

Flexibility & Scalability

Simulator — Emulator — Testbed

Accuracy

Simulator — Emulator — Testbed

Emulators provide both accuracy and scalability & flexibility. However, no emulator for evaluating NFV-RA solutions can be found in the literature to the best of my knowledge.

CHALLENGES

Online experimental evolution comes with two major challenges:

High time-consumption

An uncertain environment

So, an algorithm that adopts online experimental evolution should address these challenges.

Therefore, the tool used to deploy and evaluate solutions should not abstract away these challenges so that they can be solved by the algorithm.

EVALUATION

The performance of OpenRASE was compared with ALEVIN, a simulator, which is the most popular tool used to evaluate NFV-RA solutions.

Algorithm used — **Simple Dijkstra**
Available by default in ALEVIN

Metric used — **Acceptance Ratio**

$$\frac{\text{SFC requests that can be deployed}}{\text{Total no. of SFC requests received}}$$

Result
OpenRASE and ALEVIN produced the same acceptance ratio. This shows OpenRASE can replicate the performance of ALEVIN.

In addition, OpenRASE measured the CPU & memory usage of each server and the traffic latency of each SFC. This is not possible in ALEVIN.
OpenRASE also consumed 10 minutes for one experiment and manifested uncertainty (This can be seen in the CPU spikes in idle servers indicated by red circles in the graph).

	ALEVIN	OpenRASE
Acceptance Ratio	✓	✓
CPU Usage	✗	✓
Memory Usage	✗	✓
Traffic Latency	✗	✓

Traffic Latency

CPU Usage