OpenRASE: Service Function Chain Emulation

Theviyanthan K.

Theviyanthan Krishnamohan (Thivi)

Third-year PhD student,

University of Glasgow.

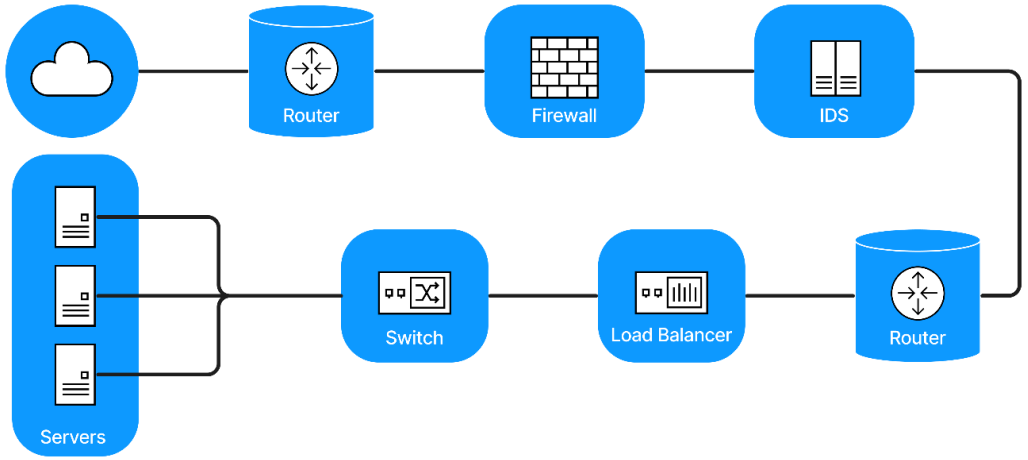
t.krishnamohan.1@research.gla.ac.uk

Background

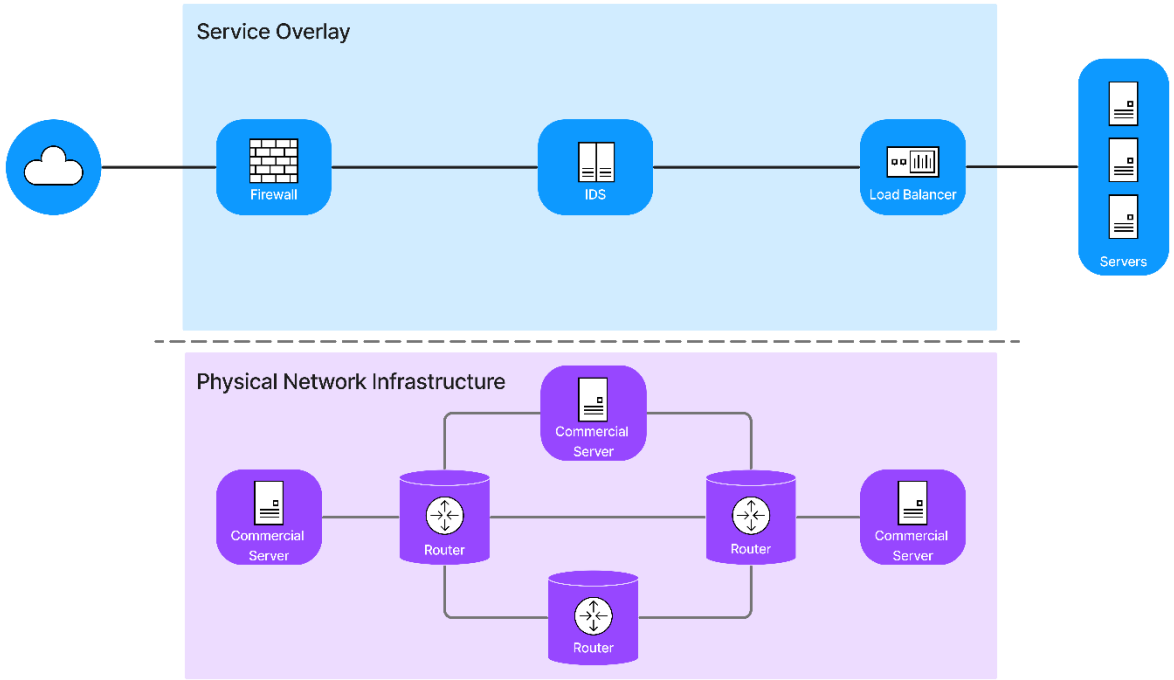
What are Service Function Chains (SFCs)?

- SFCs combine Network Function Virtualisation and Software-Defined Networking and create a service overlay over the physical network.

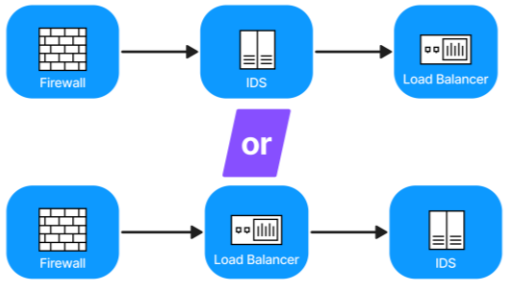
A traditional network:



A Service Function Chain:



Optimisation Challenges

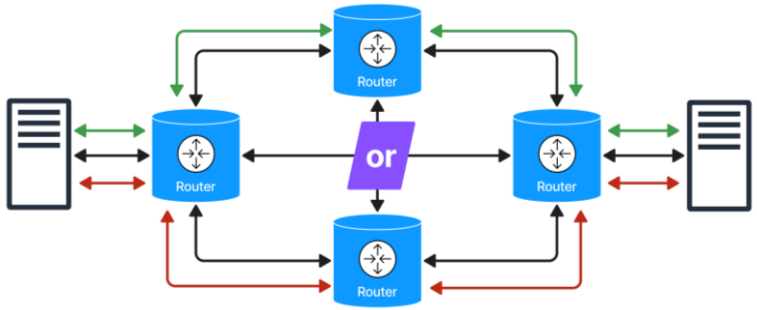


1. Chain composition

How should the Virtual Network Functions (VNFs) be ordered for optimal performance?

2. VNF embedding

Where should the VNFs be deployed for optimal performance?



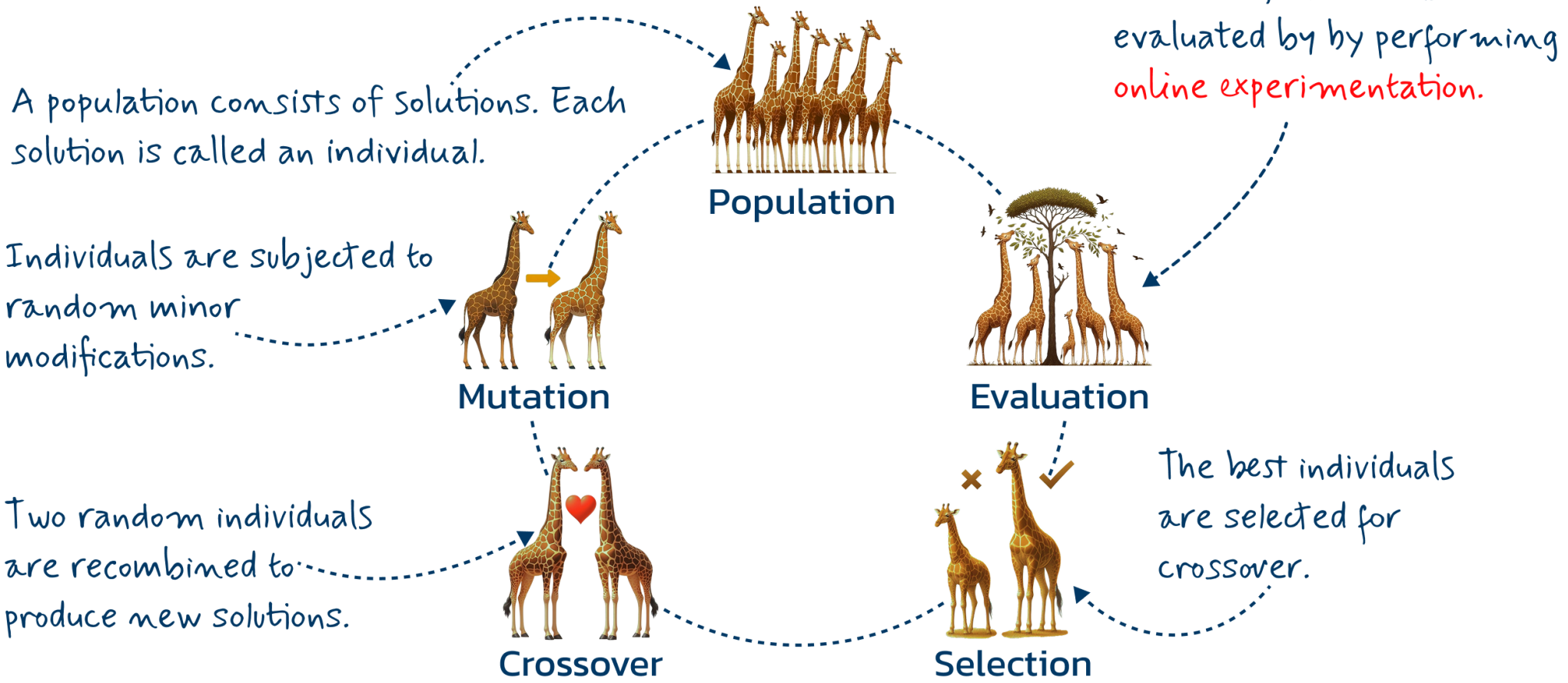
3. Link embedding

How should the VNFs be linked for optimal performance?

- This is an NP-hard optimisation problem [1].

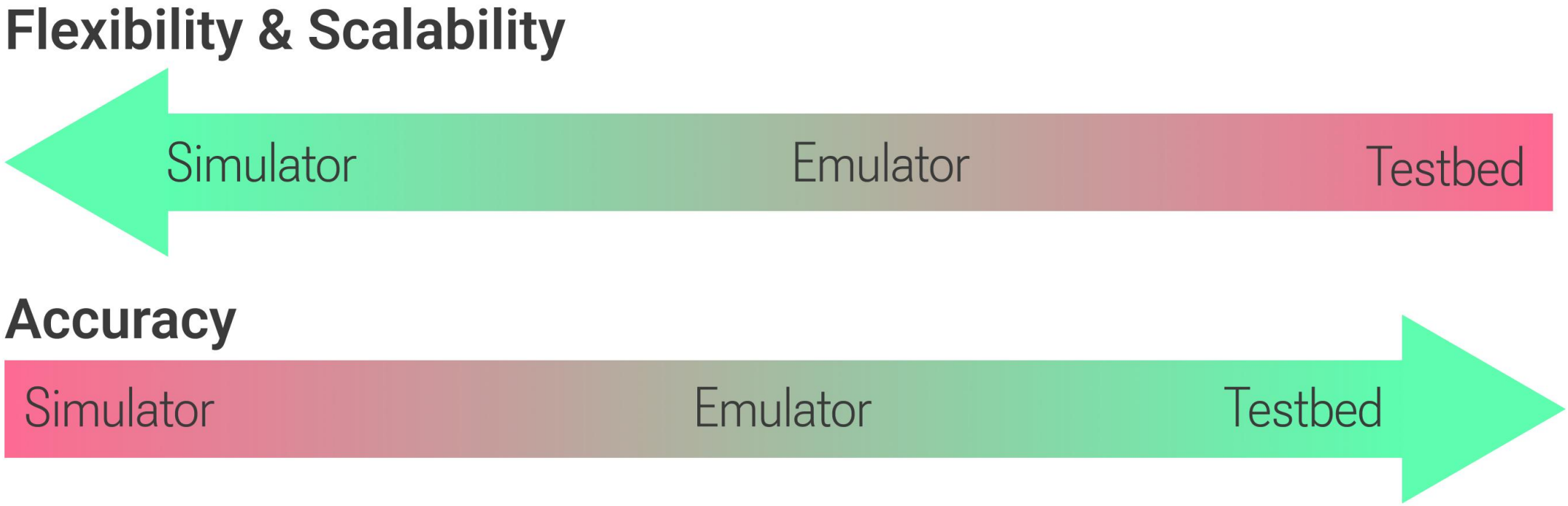
[1] J. Gil Herrera and J. F. Botero, "Resource Allocation in NFV: A Comprehensive Survey," in *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 518-532, Sept. 2016, doi: 10.1109/TNSM.2016.2598420.

Genetic Algorithm

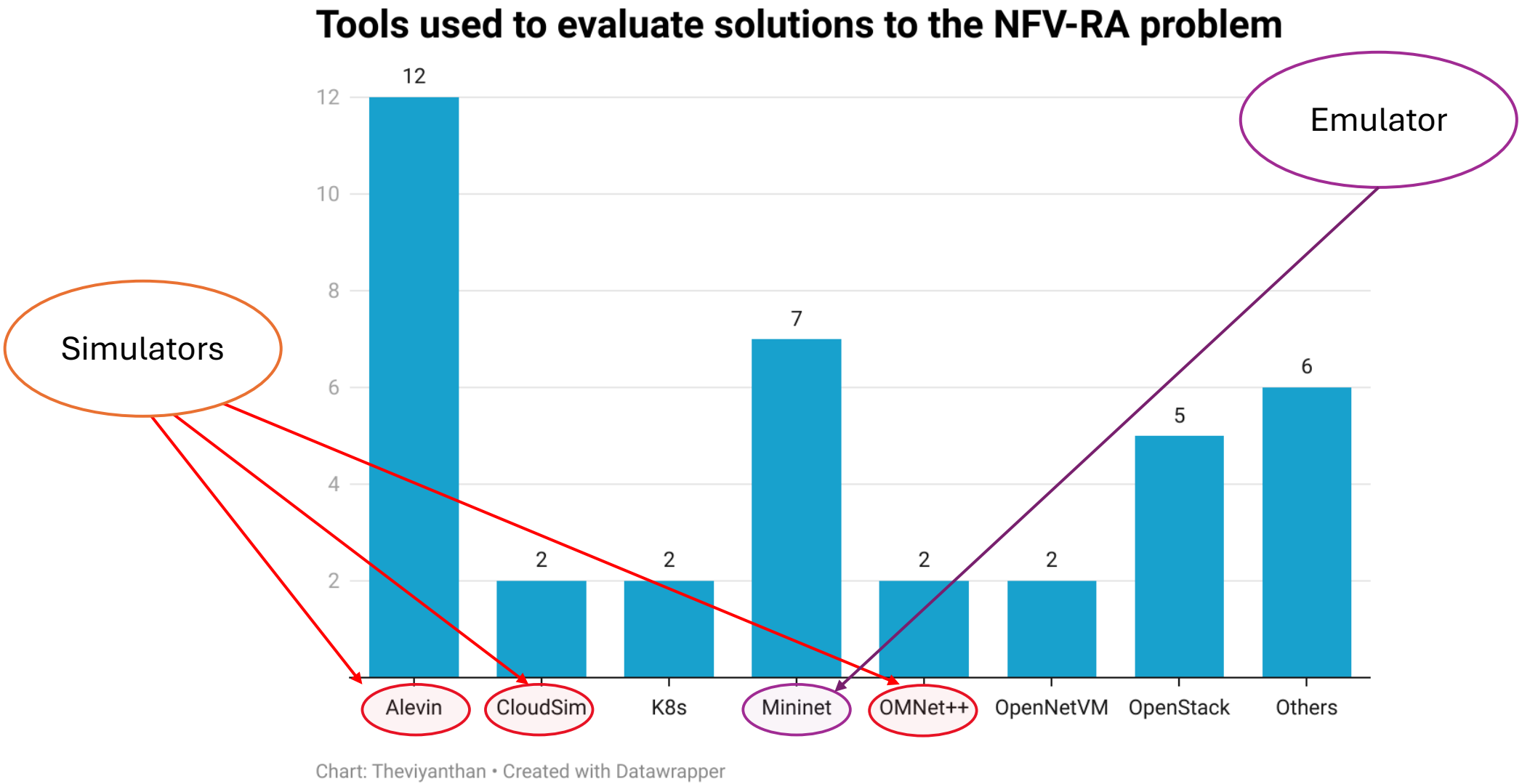


Existing Tools

How to Experiment?



Tools used in literature



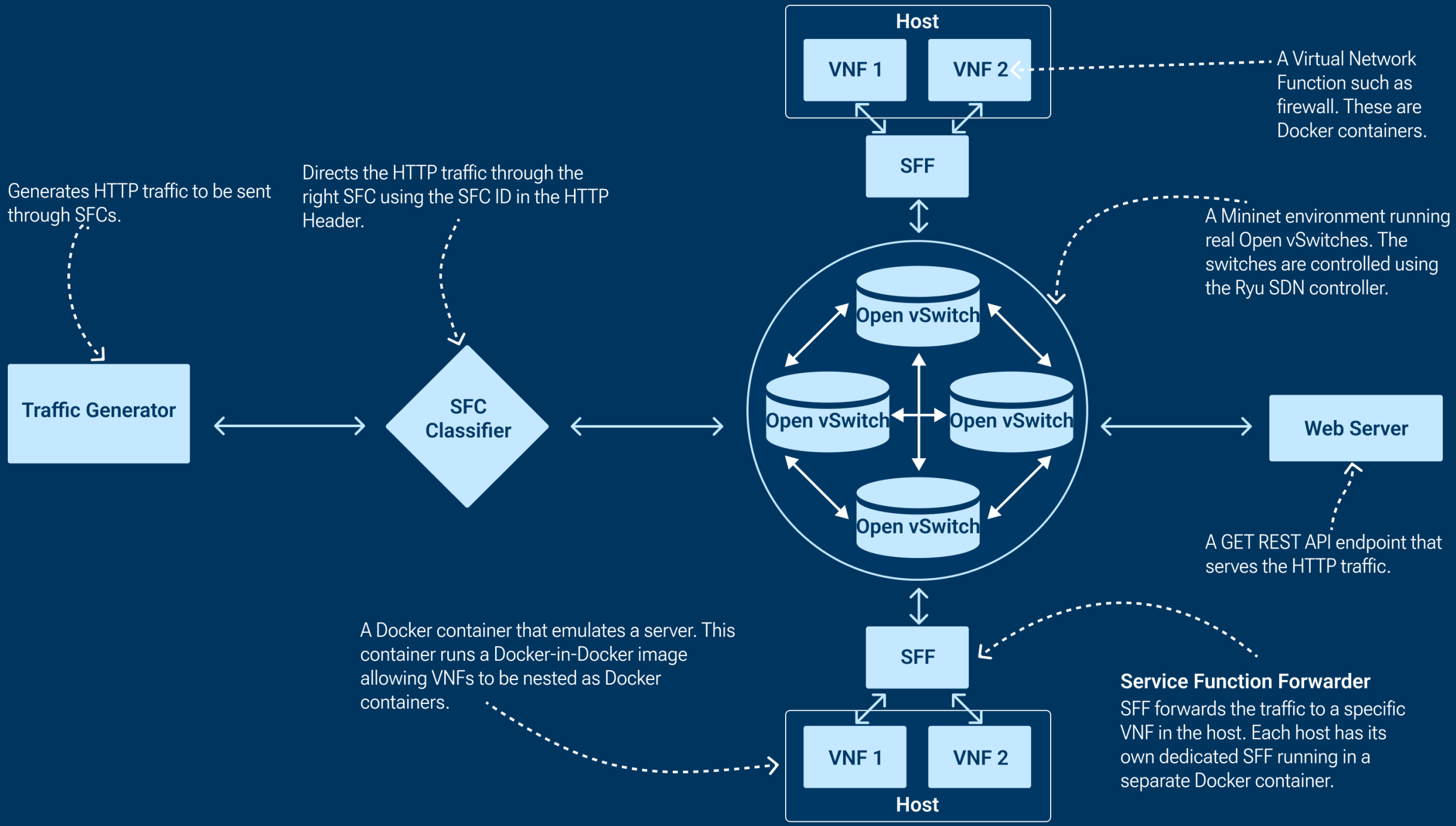
Research Gap

- Simulators such as ALEVIN **do not** provide an accurate evaluation of fitness.
- Testbeds are **not scalable**.
- Emulators such as Mininet **need additional code** to support SFC experiments.

There exists no SFC-specific emulator.

OPENRASE

https://github.com/Project-Kelvin/OpenRASE

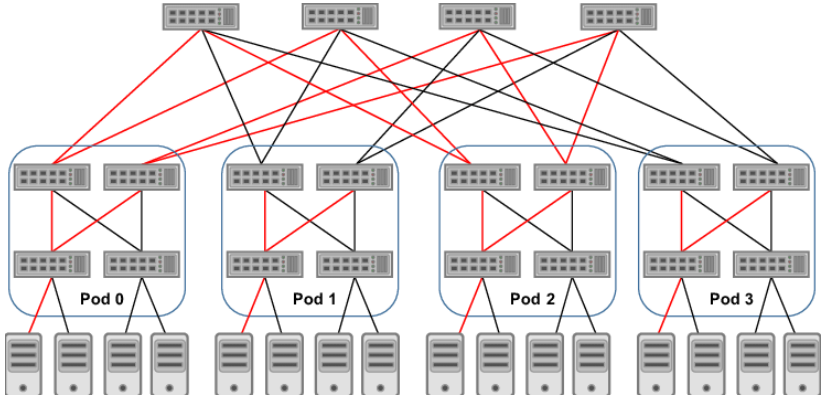

Host
VNF 1
VNF 2
A Virtual Network Function such as firewall. These are Docker containers.
SFF
Generates HTTP traffic to be sent through SFCs.
Directs the HTTP traffic through the right SFC using the SFC ID in the HTTP Header.
A Mininet environment running real Open vSwitches. The switches are controlled using the Ryu SDN controller.
Traffic Generator
SFC Classifier
Open vSwitch
Open vSwitch
Open vSwitch
Open vSwitch
Web Server
A GET REST API endpoint that serves the HTTP traffic.
A Docker container that emulates a server. This container runs a Docker-in-Docker image allowing VNFs to be nested as Docker containers.
SFF
Service Function Forwarder
SFF forwards the traffic to a specific VNF in the host. Each host has its own dedicated SFF running in a separate Docker container.
VNF 1
VNF 2
Host

Evaluation

Experimental Goals

- We demonstrated that:
  - OpenRASE can replicate ALEVIN for static metrics
  - OpenRASE can run experiments and produce dynamic metrics

Experimental Design



- Four unique SFC Requests (SFCRs) were designed based on examples from the literature.
- The greedy Dijkstra algorithm available by default in ALEVIN was used to embed SFCs.
- The acceptance ratio (no. of SFCRs that can be embedded divided by the total no. of SFCRs received), CPU usage of hosts and traffic latency of SFCs were measured.
- 8 experiments were carried out by varying the no. of SFCRs and the no. of CPUs available to hosts.

Results

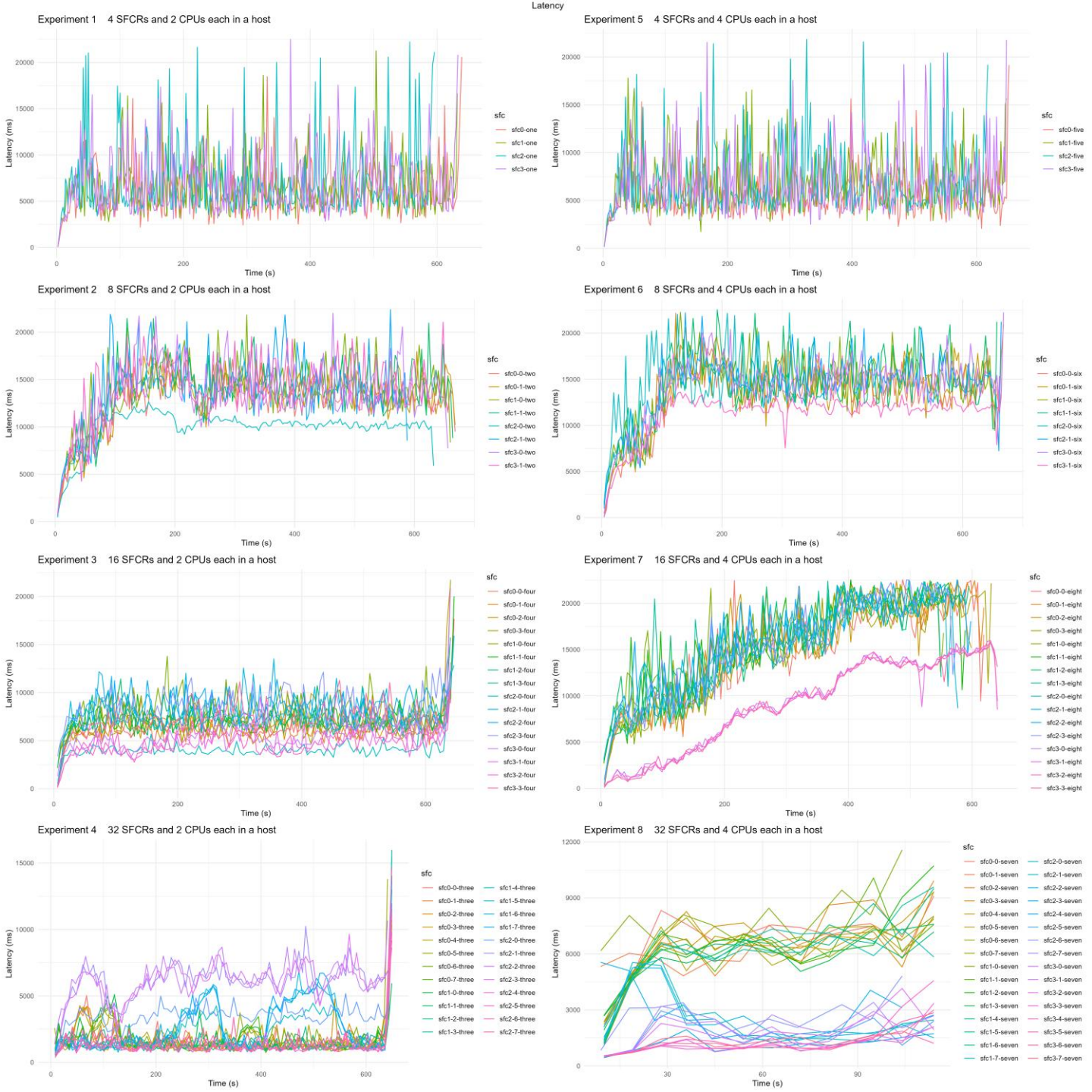
Experiment	CPUs per host	SFCRs Duplicates	Submitted SFCRs	Acceptance Ratio		Deployment Time (s)
				OpenRASE	ALEVIN	OpenRASE
1	2	1	4	1	1	34.02
2	2	2	8	1	1	44.43
3	2	4	16	1	1	75.76
4	2	8	32	0.75	0.75	90.81
5	4	1	4	1	1	28.58
6	4	2	8	1	1	34.39
7	4	4	16	1	1	55.04
8	4	8	32	1	1	110.69

- OpenRASE was able to replicate ALEVIN's outputs when it comes to static metrics like acceptance ratio.
- ALEVIN, being a simulator, cannot measure dynamic metrics like CPU usage and latency, which OpenRASE is able to.

CPU Usage


CPU Usage
4 SFCRs and 2 CPUs each in a host
4 SFCRs and 4 CPUs each in a host
8 SFCRs and 2 CPUs each in a host
8 SFCRs and 4 CPUs each in a host
16 SFCRs and 2 CPUs each in a host
16 SFCRs and 4 CPUs each in a host
32 SFCRs and 2 CPUs each in a host
32 SFCRs and 4 CPUs each in a host
CPU usage (%)
Time (s)
host
h1
h10
h11
h12
h13
h14
h2
h3
h4
h5
h6
h7
h8
h9

Traffic Latency

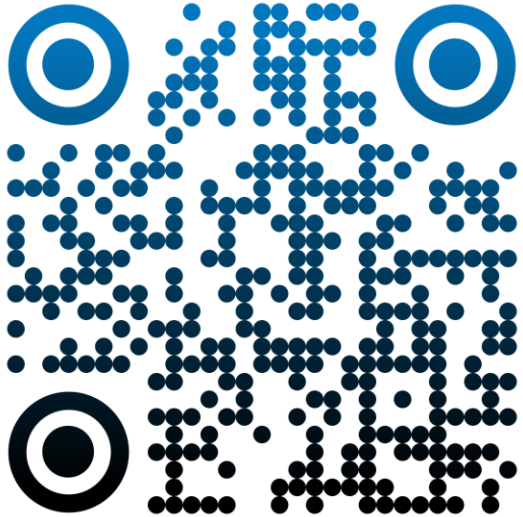


Summary

- Optimally embedding SFCs is NP-hard.
- GAs one way of solving this problem. But they require a way to evaluate solution fitness.
- We need an emulator as they are more scalable than testbeds and more accurate than simulators.
- We created OpenRASE to emulate SFCs.

OpenRASE is Open Source!

- You can use it for your SFC experiments.
- Your contributions are welcome!



https://github.com/Project-Kelvin/OpenRASE

Questions

Thank You

Appendix

Why Genetic Algorithms?

- It is a heuristic algorithm that can solve NP-hard problems.
- It can adapt to an uncertain/unknown environment.
- It is an underutilised algorithm in the SFC realm. Only 12/163 surveyed studies use GAs.

ALEVIN vs. OpenRASE

	ALEVIN	OpenRASE
Test	Simulated	Emulated.
Hosts	Simulated	Docker containers as hosts
Switches	Simulated	Open vSwitches using Mininet
VNFs	Simulated & abstract	7 distinct, real VNFs. Allows addition of more VNFs
VNF resource requirements	Arbitrary, static, & user-specified	VNF demands are calibrated through benchmarking
VNF behavior	Static	Dynamic based on input traffic
Deployment	Simulated	Emulated real code deployment using Docker containers and Mininet
Programming language	Java based	Python based
Tool resource requirements	Low	High
Use case	Rapid designing and prototyping	High-fidelity testing
New resource demands and metrics	Can be added	Adding new demands and metrics need changes to the emulator