OPENRASE

Open **R**esource **A**llocation to **S**ervice Function Chains **E**mulator

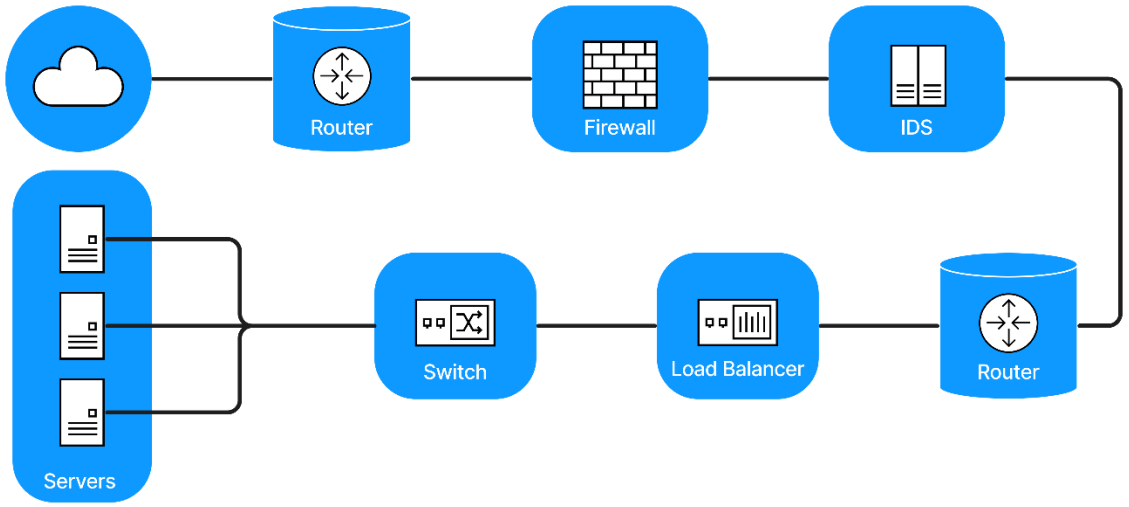
Theviyanthan K.

Background

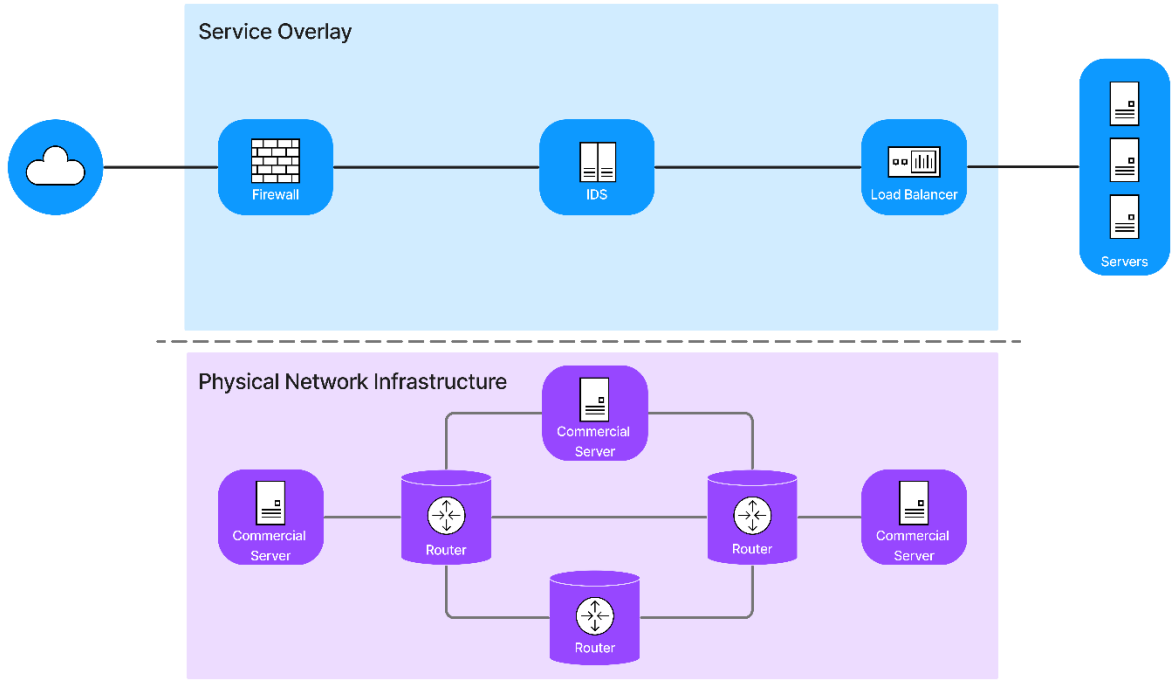
What are Service Function Chains (SFCs)?

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

A traditional network:

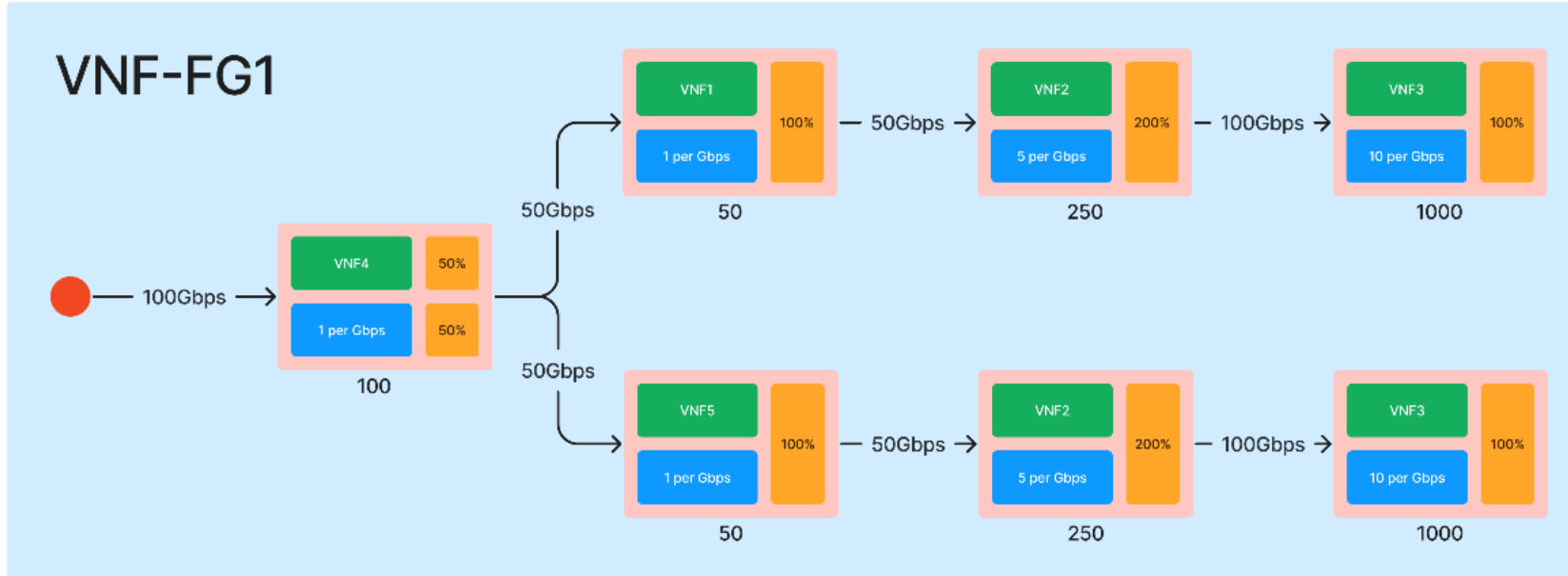


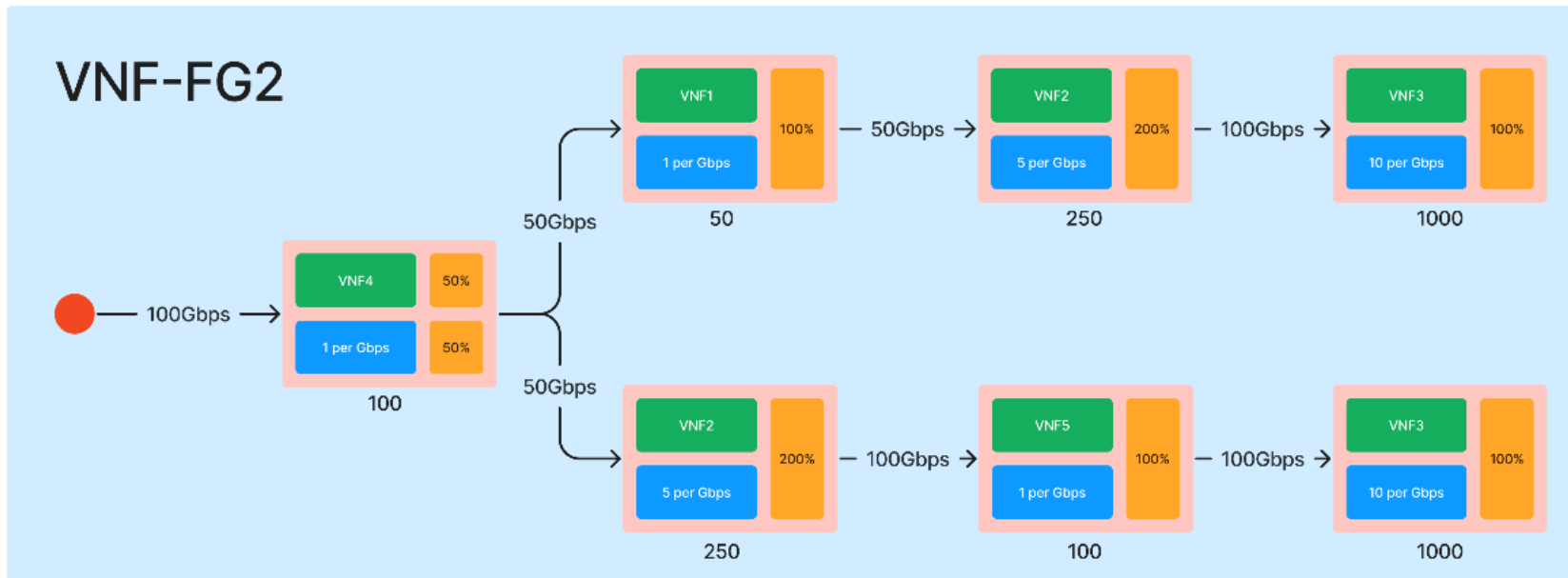
A Service Function Chain:



Network Function Virtualization Resource Allocation (NFV-RA)

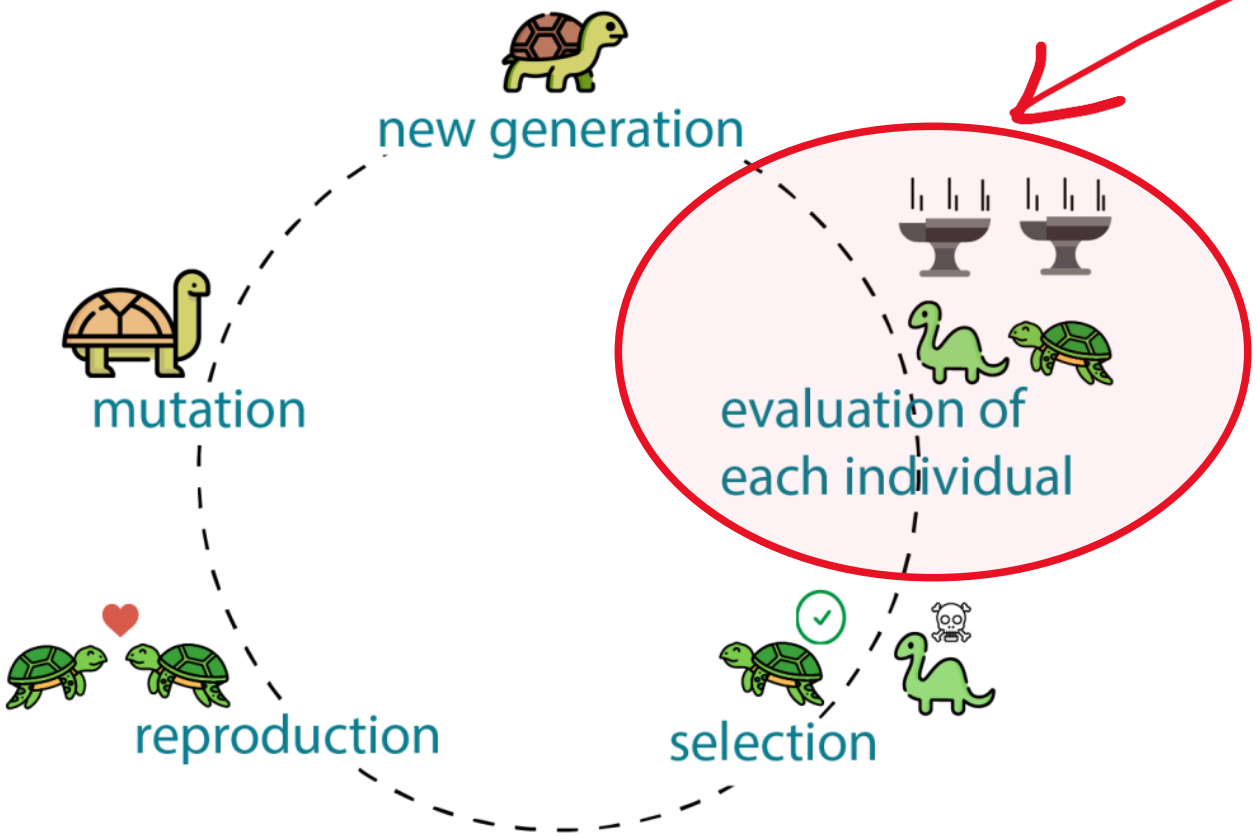
- The NFV-RA problem deals with the composition of and resource allocation to SFCs.
- It is composed of three sub-problems:
  - Virtual Network Function (VNF) Chain Composition (VNF-CC)
  - Virtual Network Function Forwarding Graph Embedding (VNF-FGE)
  - Virtual Network Function Scheduling (VNF-SCH)


VNF-FG1
100Gbps
VNF4
50%
1 per Gbps
50%
100
50Gbps
50Gbps
VNF1
100%
1 per Gbps
50
50Gbps
VNF2
200%
5 per Gbps
250
100Gbps
VNF3
100%
10 per Gbps
1000
VNF5
100%
1 per Gbps
50
50Gbps
VNF2
200%
5 per Gbps
250
100Gbps
VNF3
100%
10 per Gbps
1000


VNF-FG2
100Gbps
VNF4
50%
1 per Gbps
50%
100
50Gbps
50Gbps
VNF1
100%
1 per Gbps
50
50Gbps
VNF2
200%
5 per Gbps
250
100Gbps
VNF3
100%
10 per Gbps
1000
VNF2
200%
5 per Gbps
250
100Gbps
VNF5
100%
1 per Gbps
100
100Gbps
VNF3
100%
10 per Gbps
1000

What am I trying to do?

My Research

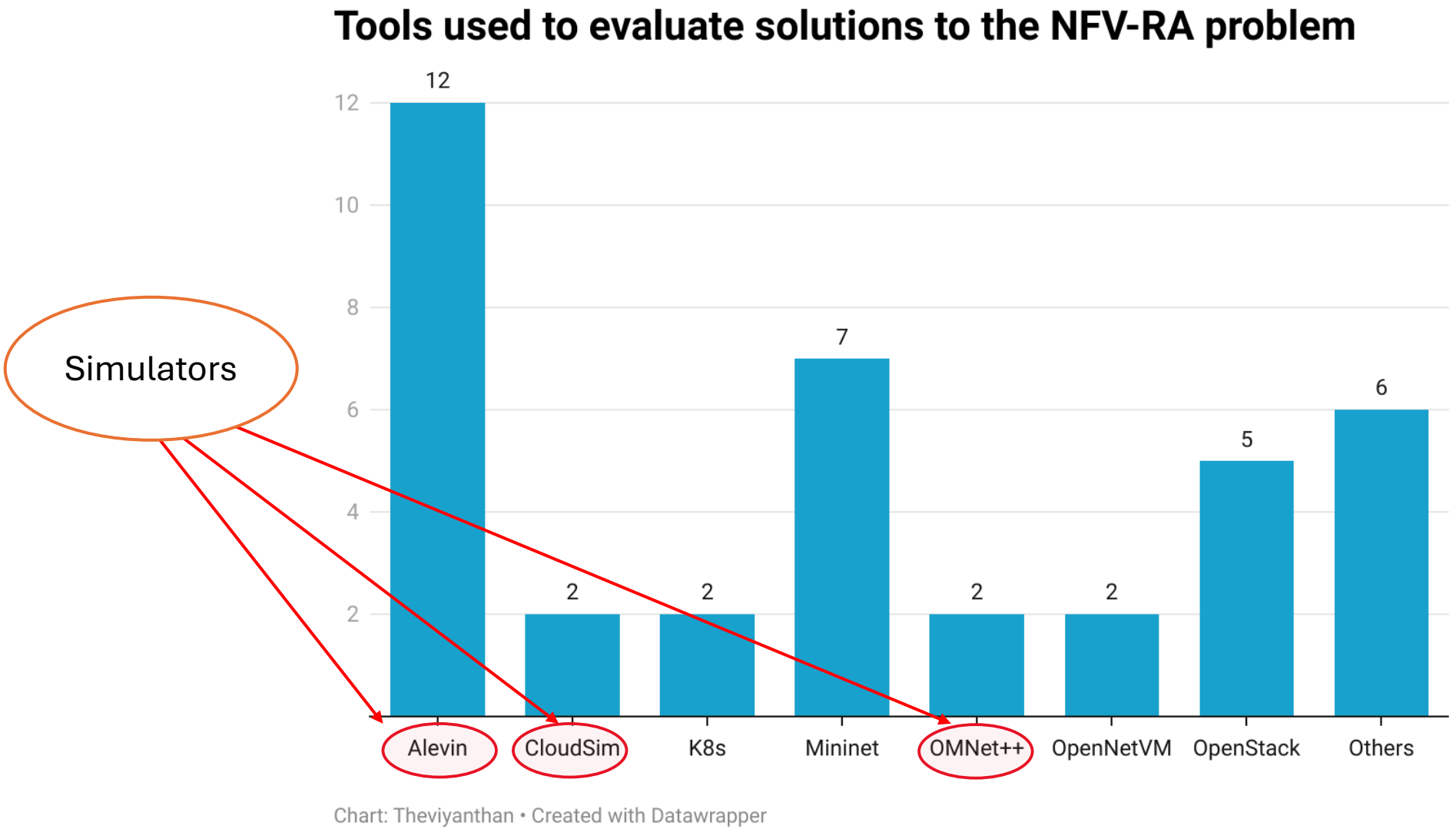


Evaluation is done on a live network, so it is going to be time-consuming.

- I plan to use Genetic Algorithms to solve the VNF-CC and VNF-FGE problems.

Why do I need OpenRASE

Tools used in literature



Why can’t I use a simulator?

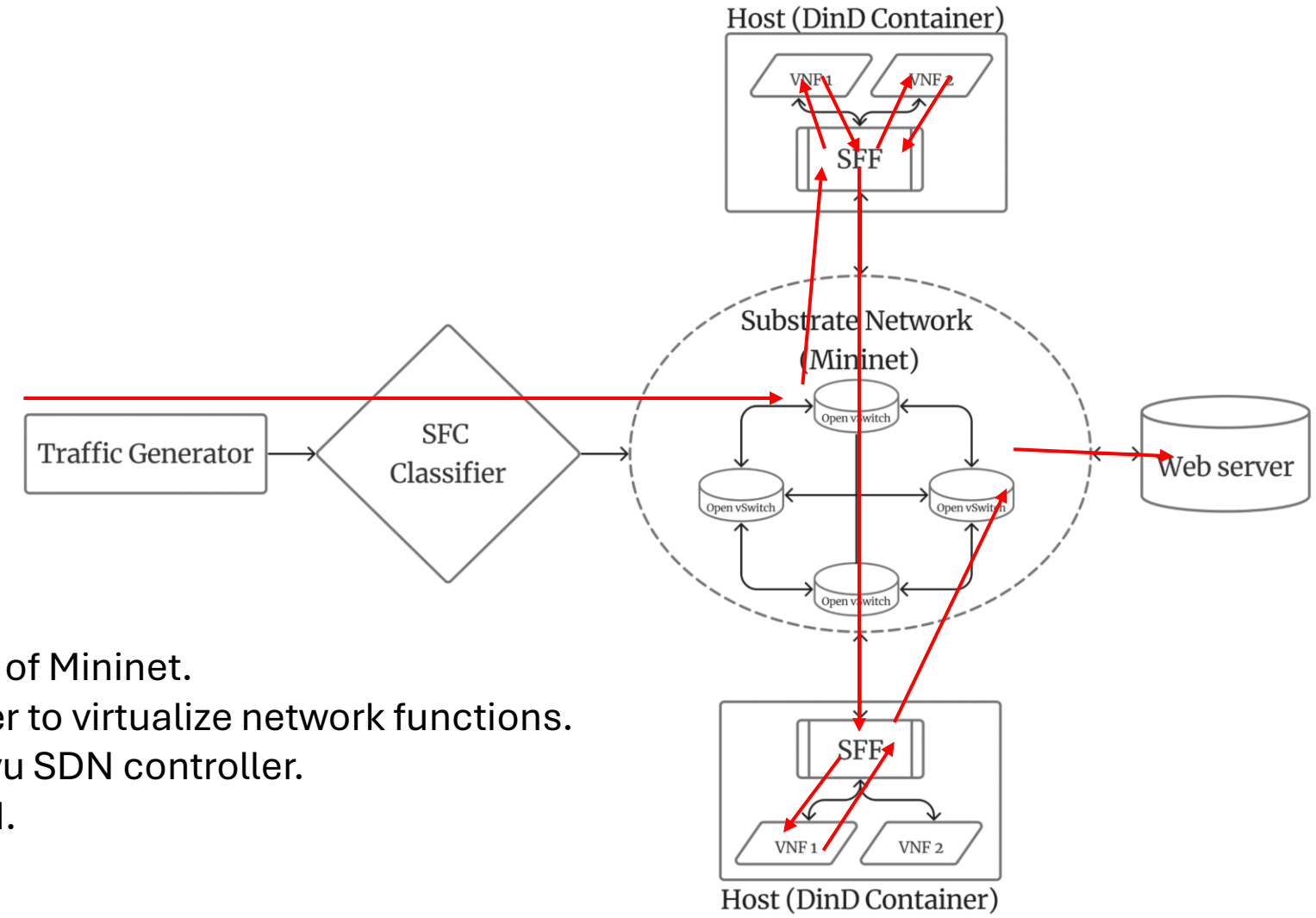
- Simulators don’t provide high fidelity.
- A major challenge in online experimentation is excessive time consumption.
- The use of simulators nullifies this challenge.
- This means an algorithm that runs fast on a simulator may not necessarily be fast on a real network.
- An algorithm’s ability to adapt to an uncertain environment cannot be tested.

Why do I need an emulator?

- Emulators offer higher fidelity than simulators.
- They are closer to real networks. So, online experimentations on emulators consume time just like real networks.
- This means that an algorithm’s ability to address such challenges can be tested with high fidelity.
- Provides a more uncertain environment than a simulator.

How does OpenRASE work?

Architecture



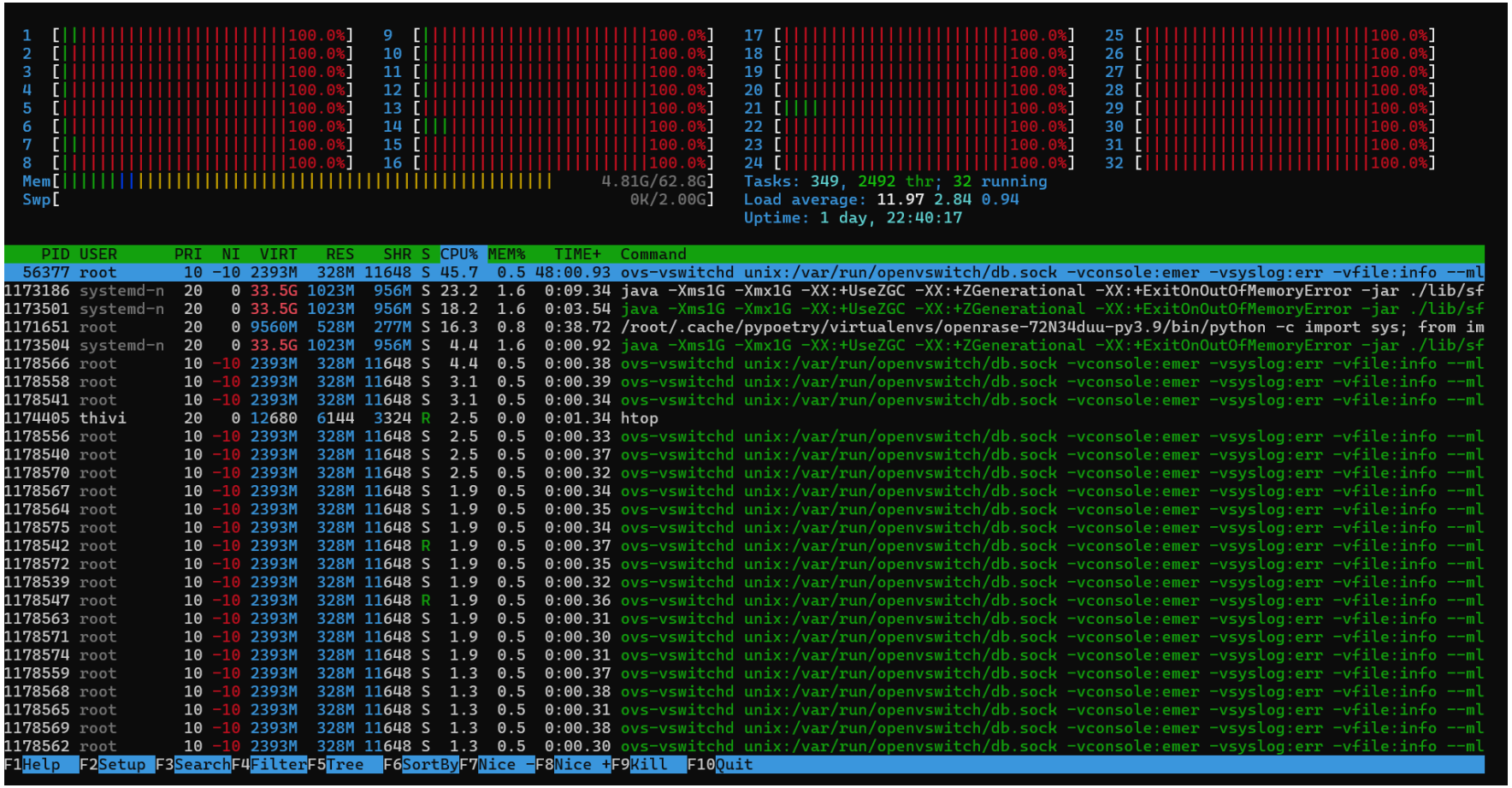
- Built on top of Mininet.
- Uses Docker to virtualize network functions.
- Uses the Ryu SDN controller.
- HTTP based.

Future Work

- Evaluating OpenRASE
- Developing Genetic Algorithms on top of OpenRASE

Challenges

Running a 4ary fat tree topology with a Ryu controller running the REST router app exhausts all CPU cores.



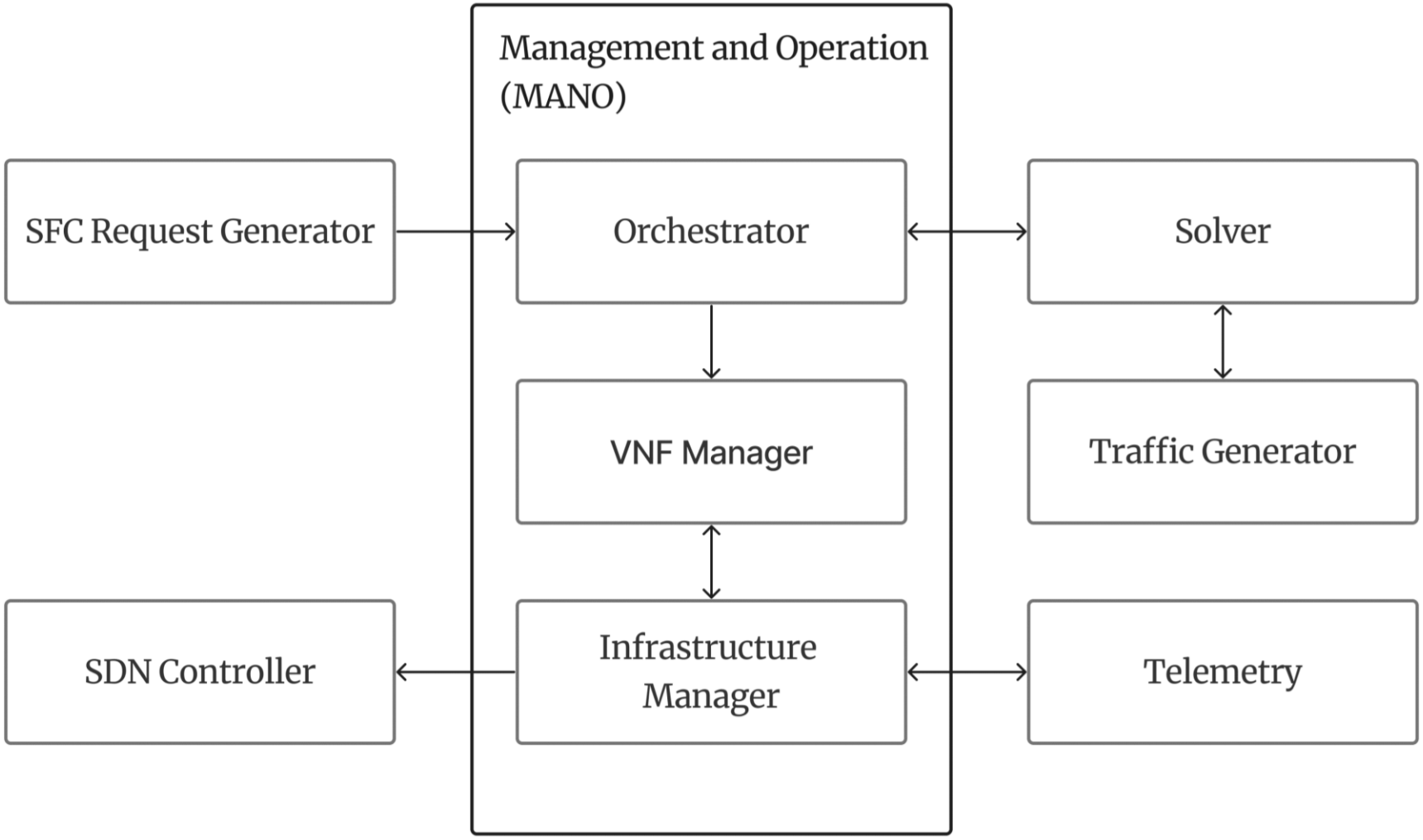
Questions?

Appendix

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

Design


Management and Operation (MANO)
SFC Request Generator
Orchestrator
Solver
VNF Manager
Traffic Generator
SDN Controller
Infrastructure Manager
Telemetry