Docker Network Tester

A Python3 Wrapper around the Docker SDK for network performance tests over a network of containers.

Flavien Vargues

Erasmus Student

University of Technology of Troyes (France)

Networks & Telecommunications

Goals & Chosen Design

Goal: to build a **scripting/visual interface** to configure and **run a network of VMs**/containers and OpenVSwitches.

User interface could be a **script** that is converted in a VagrantFile for running the network, or an HTTP interface.

Link capacity should be controllable : bandwidth, delay, loss.

Suggested software: Vagrant, OpenVSwitch, **Docker**.

Chosen Design: Building a **Wrapper** to configure and instantiate a **network of containers** in a certain topology.

Python3 around the Docker SDK, usable through any python interface **(shell, jupyter notebook, python file)**

Link capacity controllable : bandwidth, delay and loss.

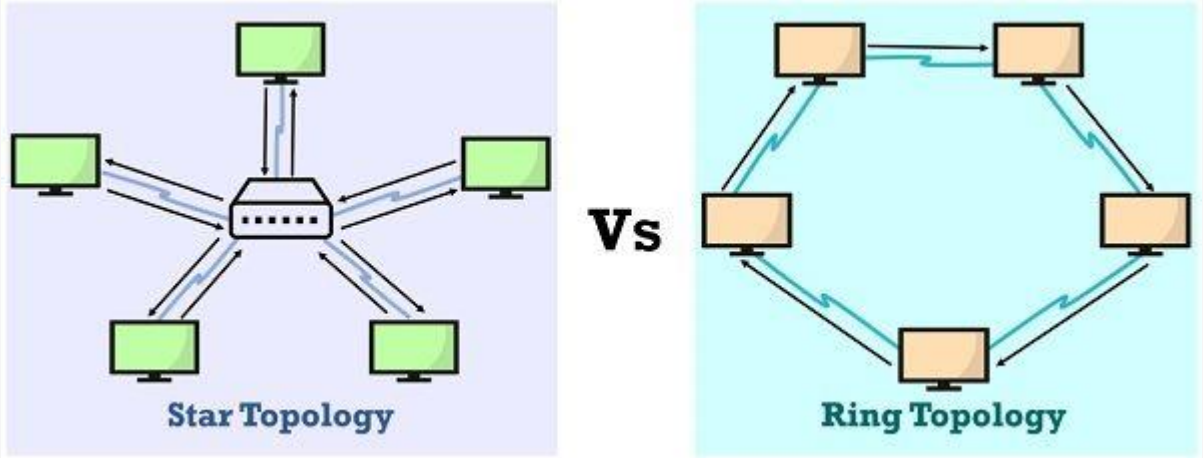
Main software used : **Python3, Docker SDK**

Priorities

1. Easily instantiate the containers in the correct configuration and execute tests.
2. Provide a way to configure the network and traffic control.
3. Facilitate the use of the program for testing.

Design choices & Difficulties

1. How to enforce traffic control ?
    - In container program ?
    - **Docker-tc**
2. How to enforce a topology ?
    - Docker links (legacy) ?
    - Docker-topo?
    - **Docker networks between containers**
3. How to enforce IP routing
    - Manipulation of hosts table ?
    - **Modifying the gateway of IP route**



Available commands

- DNTConfiguration Class
  - Input check of configuration
- DNT Class
  - License()
  - Help()
  - Connect()
  - Build()
  - Destroy()

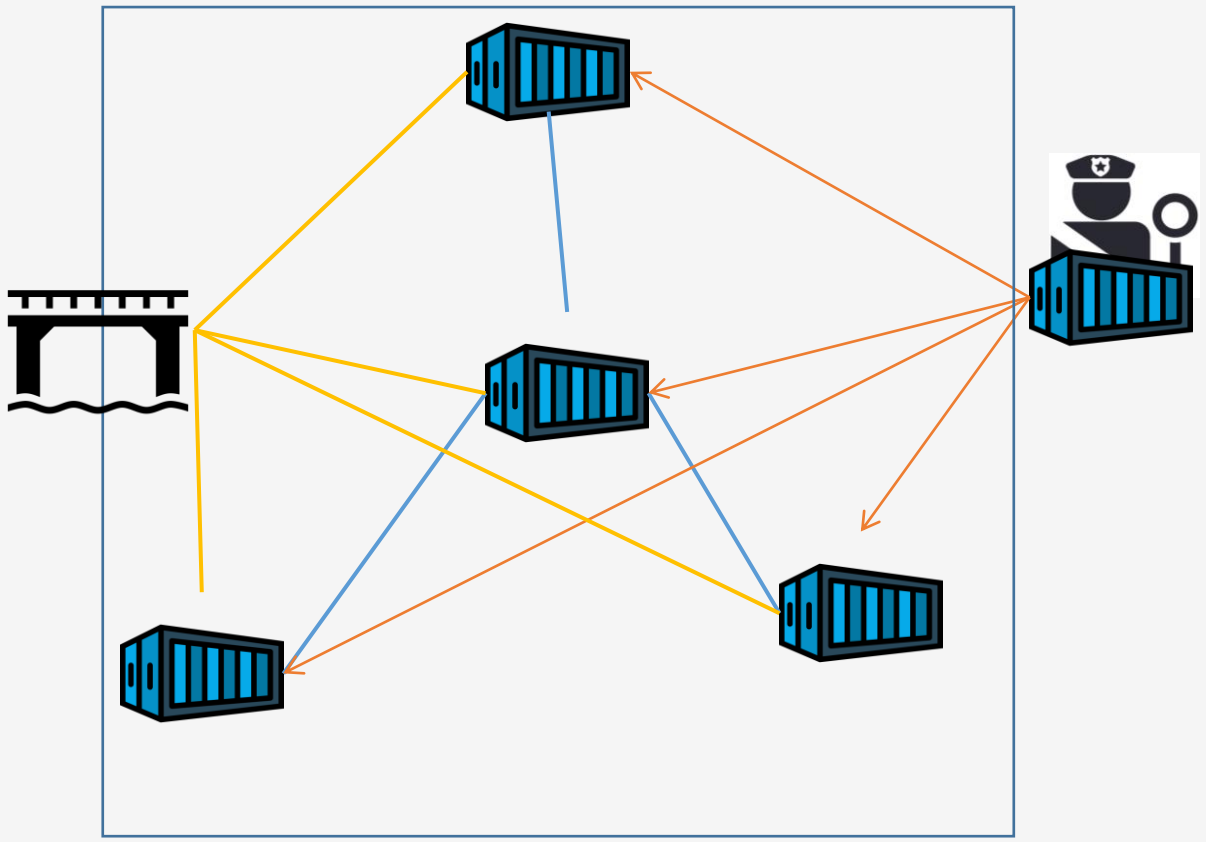
Tests commands: returns python dict()

- Ping
- Traceroute
- Iperf3
- Twamp

How it works

Build() – Where the magic happens !

1. Instantiate the containers (connected to bridge).
2. Connect the containers to networks for each link.
3. Disconnect containers to the bridge.
4. Correct IP routing.



Demonstration