

Applied Networking Research Workshop 2020



NeST: Network Stack Tester

Shanthanu S Rai, Narayan G, Dhanasekhar M,
Leslie Monis, Mohit P. Tahiliani



Wireless Information Networking Group (WiNG)
Department of Computer Science and Engineering
National Institute of Technology Karnataka, Surathkal, Mangalore, India
July 30, 2020

Network Experimentation

- ▶ Physical testbeds

Network Namespaces

- ▶ Virtualize network stack
- ▶ Complex virtual topologies can be created
- ▶ Minimal system resources

Existing tools

- ▶ Mininet
- ▶ Flexible Network Tester (Flent)
- ▶ Transperf
- ▶ Netesto
- ▶ TEACUP

- ▶ Python package to simplify the process of network experimentation by using Linux network namespaces
- ▶ Intuitive APIs to: build a virtual network, run experiments and collect statistics
- ▶ Simplifies the process to reproduce network experiments
- ▶ Less physical resources, less error prone and less prerequisites
- ▶ Multiple instances of the same network topology can co-exist, and different experiments can be run in parallel on every instance
- ▶ Open source tool released under GPLv2 License
- ▶ Link: <https://nitk-nest.github.io/>

Peer to peer topology

```
# Create two nodes
n0 = Node('n0')
n1 = Node('n1')

# Connect nodes and get corresponding interfaces
(n0_n1, n1_n0) = connect(n0, n1)

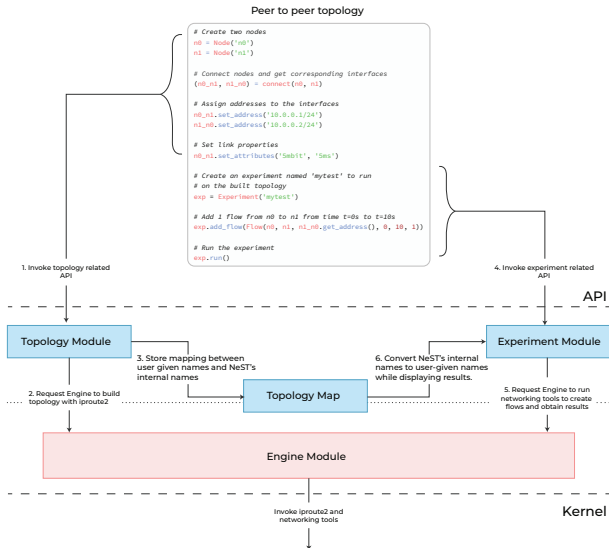
# Assign addresses to the interfaces
n0_n1.set_address('10.0.0.1/24')
n1_n0.set_address('10.0.0.2/24')

# Set link properties
n0_n1.set_attributes('5mbit', '5ms')

# Create an experiment named 'mytest' to run
# on the built topology
exp = Experiment('mytest')

# Add 1 flow from n0 to n1 from time t=0s to t=10s
exp.add_flow(Flow(n0, n1, n1_n0.get_address(), 0, 10, 1))

# Run the experiment
exp.run()
```



Scope

- ▶ Advanced traffic control
- ▶ TCP parameters
- ▶ Netperf, ss, tc
- ▶ Addition of new tools is easy (e.g., httpperf)

Limitations

- ▶ Effects of hardware level optimizations are not seen
- ▶ Lack of support for all implementations of network stacks
- ▶ Lack of advanced debugging functions

Motivation

How accurate are NeST results compared to a physical testbed?

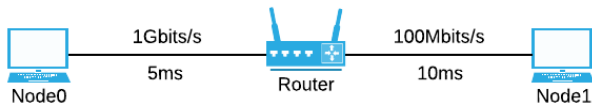
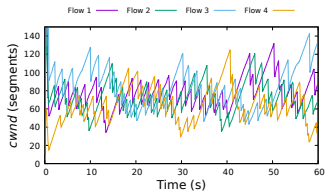


Figure: Simple topology for NeST Validation

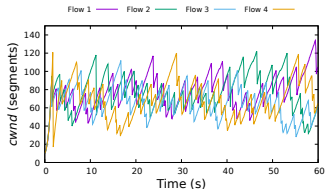
- ▶ 4 CUBIC TCP flows from Node0 to Node1
- ▶ Two experiments run with two different *qdiscs* at Router: CoDel and FIFO.

Experiment 1: Plots

CoDel

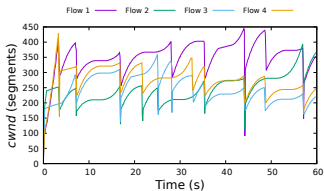


(a) cwnd: NeST

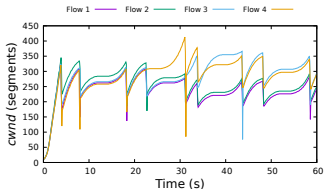


(b) cwnd: Physical Testbed

FIFO



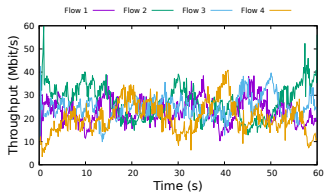
(a) cwnd: NeST



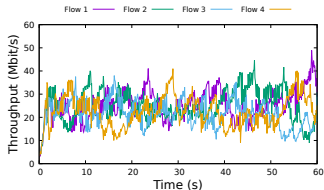
(b) cwnd: Physical Testbed

Experiment 1: Plots

CoDel

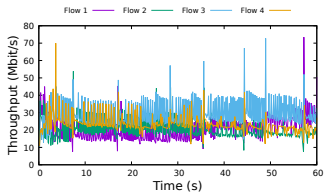


(a) Throughput: NeST

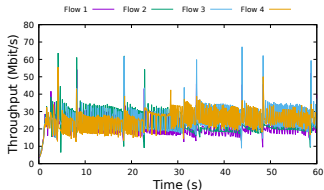


(b) Throughput: Physical Testbed

FIFO



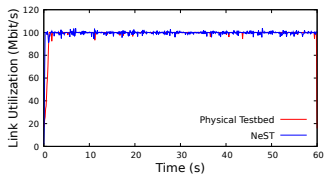
(a) Throughput: NeST



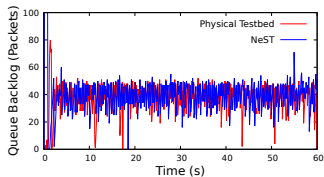
(b) Throughput: Physical Testbed

Experiment 1: Plots

CoDel

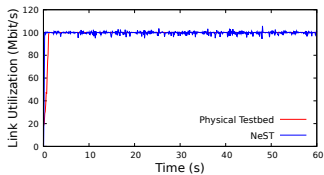


(a) Link Utilization

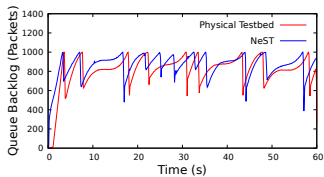


(b) Queue Backlog

FIFO



(a) Link Utilization



(b) Queue Backlog

Motivation

How does NeST perform in emulating and running experiments on a fairly complex topology?

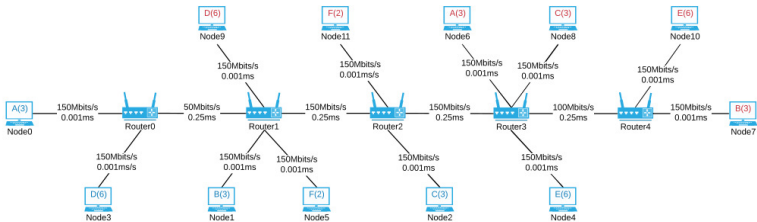
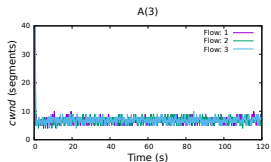
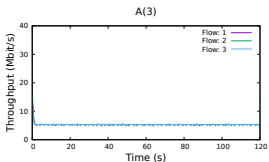


Figure: Complex topology for NeST validation

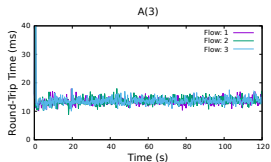
Experiment 2: Plots



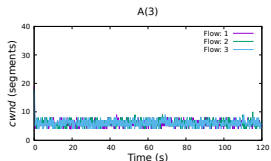
(a) *cwnd*: NeST



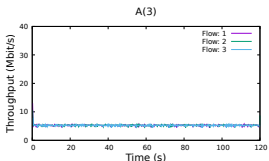
(a) Throughput: NeST



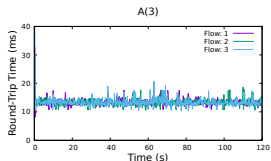
(a) RTT: NeST



(b) *cwnd*: Flent

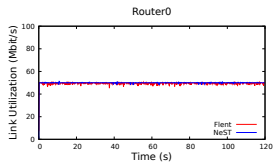


(b) Throughput: Flent

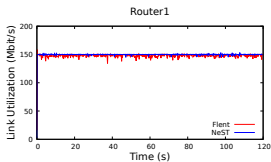


(b) RTT: Flent

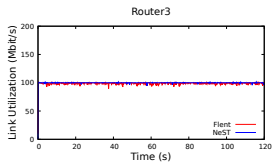
Experiment 2: Plots



(a) Link Utilization: Router0



(b) Link Utilization: Router1



(c) Link Utilization: Router3

To view other plots, please check: <https://gitlab.com/nitk-nest/nest-anrw20>

- ▶ NeST can be obtained from PyPI (Python Package Index)
- ▶ NeST is open source software. Contributions are welcome.
- ▶ Website: <https://nitk-nest.github.io/>

Contact us:

Shanthanu S Rai
shanthanu.s.raig
@gmail.com

Narayan G
gnarayang
@gmail.com

Dhanasekhar M
sekhardhana529
@gmail.com

Leslie Monis
lesliemonis
@gmail.com

Mohit P. Tahiliani
tahiliani
@nitk.edu.in