nsdi '24

Klonet

an Easy-to-Use and Scalable Platform for Computer Networks Education

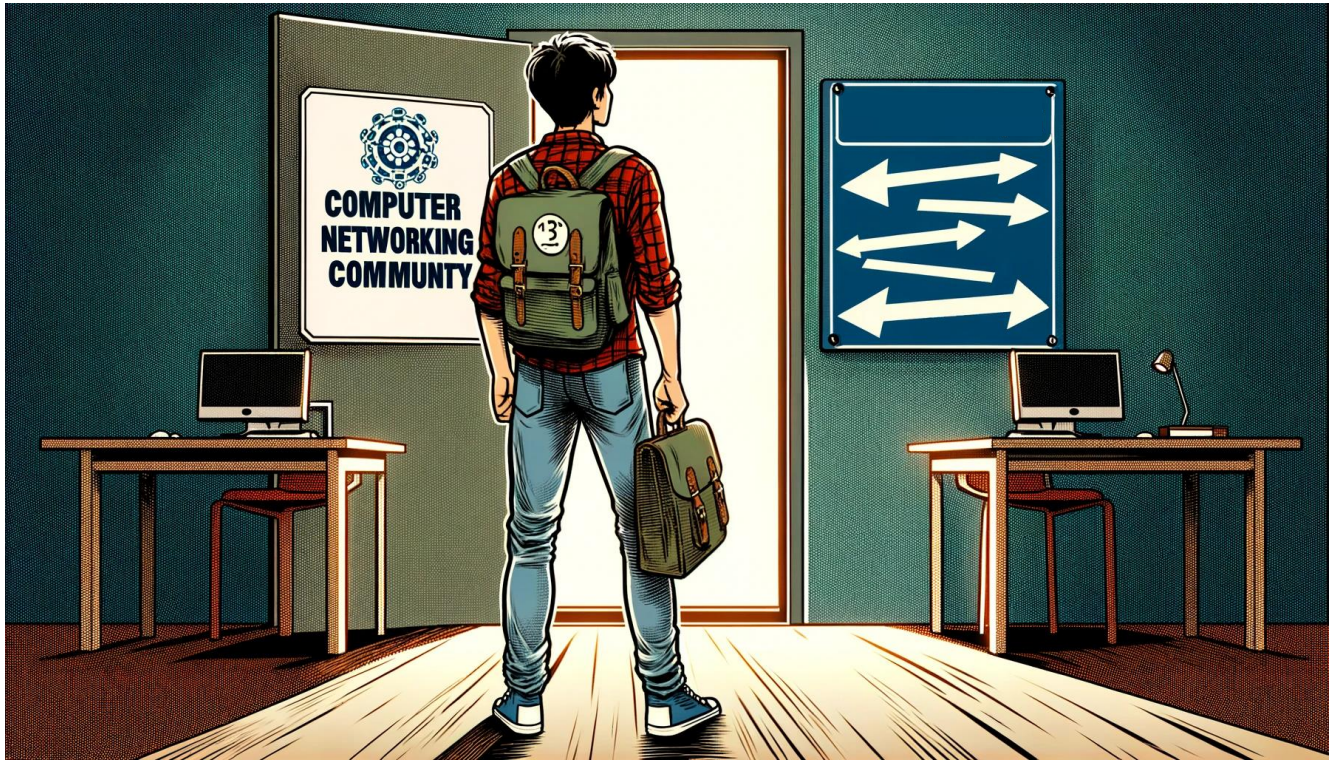
Tie Ma†, Long Luo†, Hongfang Yu†, Xi Chen‡, Jingzhao Xie†, Chongxi Ma†, Yunhan Xie†, Gang Sun†, Tianxi Wei†, Li Chen◊, Yanwei Xu¶, Nicholas Zhang¶

†University of Electronic Science and Technology of China

‡Southwest Minzu University ◊Zhongguancun Laboratory

¶Theory Lab, Central Research Institute, 2012 Labs, Huawei Technologies Co., Ltd.

Education is Vital to Our Community



Education is the gateway for newcomers, shaping the future of computer networking!

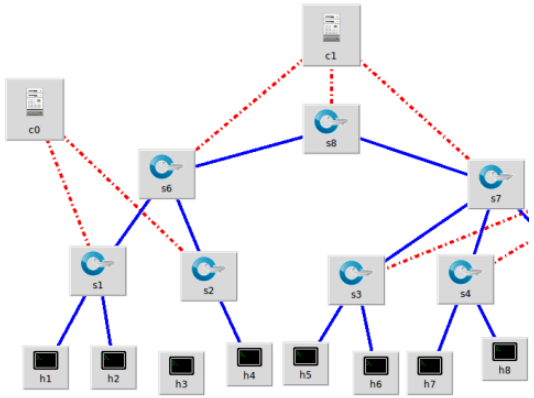
Practice is Important

- Computer Networks: an **abstract** and **complex** discipline
- The **key** to computer networks education: **Practice**
- Ways for students to get practice:

Testbed

Realistic,
but expensive

Emulator



Compromise
choice

Simulator

```
def send():
    msg_send = Message("testMsg");
    scheduleAt(5.0, msg_send)

def handle():
    if recv:
        msg_ack = Message("ack");
        scheduleAt(
            simTime()+1.0, msg_ack)
```

Cheap,
but unrealistic

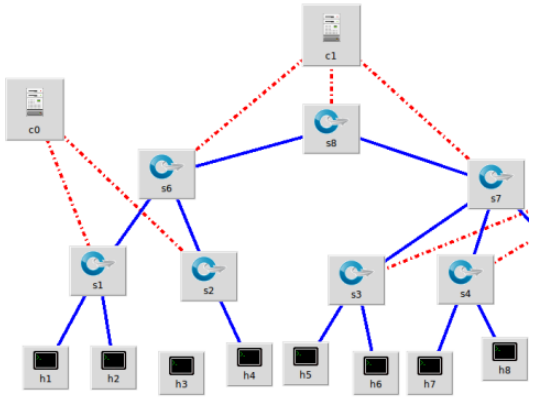
Practice is Important

- Computer Networks: an **abstract** and **complex** discipline
- The **key** to computer networks education: **Practice**
- Ways for students to get practice:

Testbed

Realistic,
but expensive

Emulator



Compromise choice

Simulator

```
def send():
    msg_send = Message("testMsg");
    scheduleAt(5.0, msg_send)

def handle():
    if recv:
        msg_ack = Message("ack");
        scheduleAt(
            simTime()+1.0, msg_ack)
```

Cheap,
but unrealistic

We aim to build a network emulation platform for education

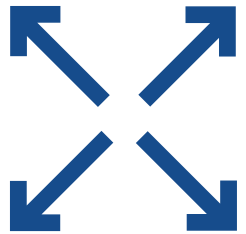
Goals for an ideal education platform

Easy-to-use

- Help students lower the barrier to practice
  - E.g. easy to get started and master basic operations
  - *"The first step is always the hardest"*
- Help tutors improve teaching efficiency
  - E.g. direct students easily and setup environments quickly

Scalable

- Support a large number of concurrent experiments
- Support several large-scale emulated networks



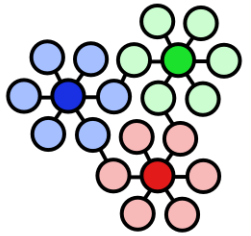
Existing Network Emulators

■ Genaral-purpose Emulators

- Mininet
  - Needs installation (and learning Linux, Virtual Machine, Shell, …)
  - Installing Mininet on a shared server seems a good idea ➡ *Poor isolation and scalability*
  - Lacks a student-friendly GUI
- Emulab
  - Heavy-weight
  - Poor scalability
  - Slow virtual network creation speed
- End-to-end emulators ➡ *Cannot emulate all types of devices (e.g. router)*
- Emulators for special scenarios ➡ *Cannot emulate diverse network scenario*
- …

Beginners

Mininet



StarryNet

Existing Network Emulators

■ Education-purpose Emulators

- Netkit ➡ *Heavy-weight*
- Kathará ➡ *Poor Scalability*
- GNS3 ➡ *Needs installation*
- Mini-Internet ➡ *Poor Network Scenarios*
- SEED ➡ *Lacks Embedding Algorithm*
- IP-mininet ➡ *Needs installation*
- …

SEED Internet Emulator

Existing Network Emulators

Platform Name	Easy-to-use					Scalability
	No Installation Required	GUI and Experiment API	Teaching Tools[1]	Experiment Tools[2]	Rich Node Types	
Mininet [12]	✗	Humble GUI	✗	Limited	✗	✗
Mininet-Hifi [20]	✗	Humble GUI	✗	Limited	✗	✗
Distrinet [15]	✗	No GUI	✗	✗	Limited	✓
Containernet [16]	✗	No GUI	✓	✗	Limited	✗
Vt-Mininet [21]	✗	No GUI	✗	✗	✗	✓
Mininet-Wifi [22]	✗	Humble GUI	✓	✓	✓	✗
Emulab [13]	✓	✓	Limited	✗	Limited	Limited
Netkit [37]	✗	No GUI	✓	✗	✓	✗
Kathará [17]	✗	Humble GUI	✓	✓	✓	✗
Megalos [18]	✗	Humble GUI	✓	✓	✓	✓
GNS3 [19]	✗	✓	✓	✓	✓	✓
SEED [35]	✗	✓	✓	✗	✓	✓
Mini-Internet [36]	✓	No GUI	✓	Limited	✗	✗
IPMininet [14]	✗	No GUI	✓	✓	✗	✗
Klonet (this work)	✓	✓	✓	✓	✓	✓

[1] **Teaching tools** are those designed to facilitate education, *e.g.*, Klonet's scene repository and Mini-Internet's connectivity matrix.
[2] **Experiment tools** are those designed to make experiments easier, *e.g.*, Klonet's traffic generator and IPMininet's IP configuration tools.

Related Network Emulators

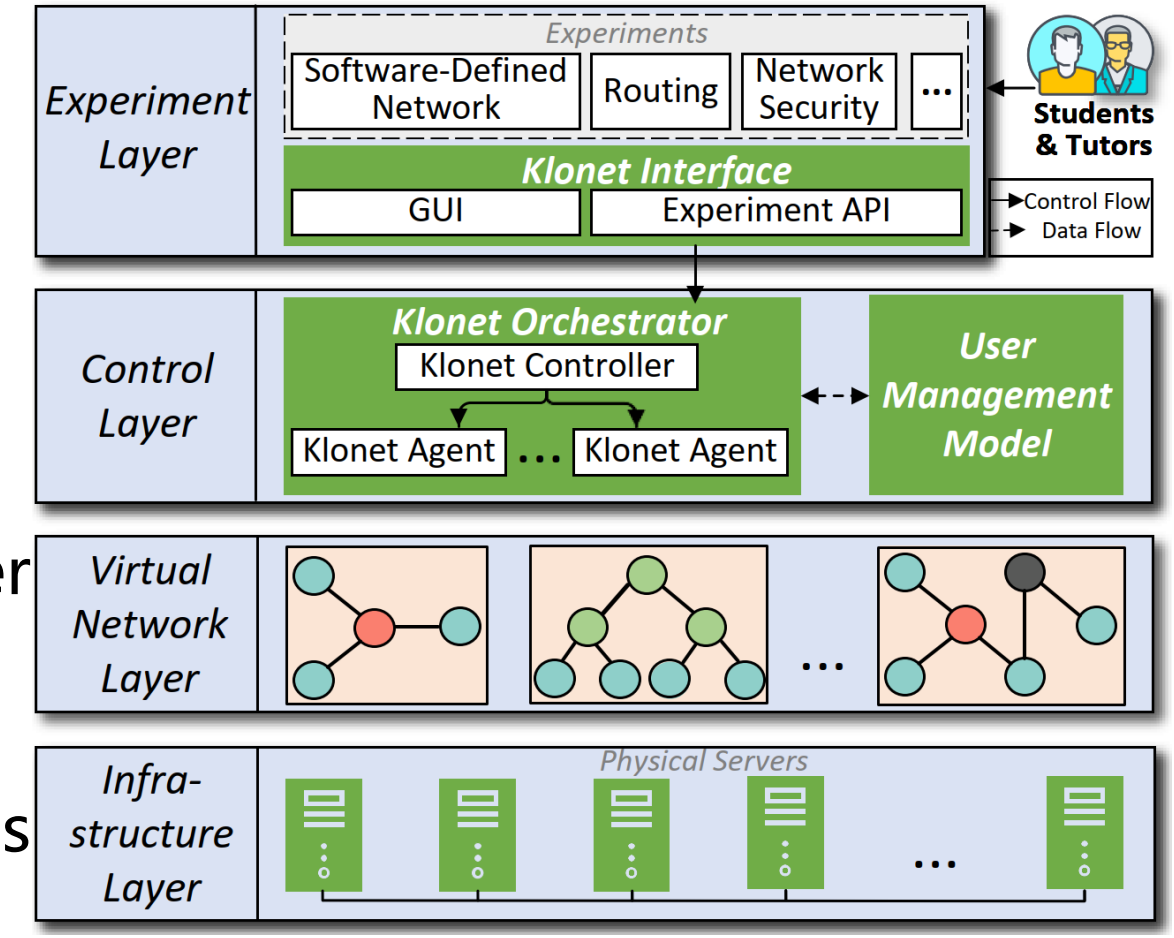
Platform Name	Easy-to-use					Scalability
	No Installation Required	GUI and Experiment API	Teaching Tools[1]	Experiment Tools[2]	Rich Node Types	
Mininet [12]	✗	Humble GUI	✗	Limited	✗	✗
Mininet-Hifi [20]	✗	Humble GUI	✗	Limited	✗	✗
Distrinet [15]	✗	No GUI	✗	✗	Limited	✓
Containernet [16]	✗	No GUI	✓	✗	Limited	✗
Vt-Mininet [24]	✗	No GUI	✗	✗	✗	✓
Mininet-Wifi [17]	✗	No GUI	✗	✗	✗	✗
Emulab [34]	✗	Humble GUI	✓	✓	✓	Limited
Netkit [11]	✗	No GUI	✓	✗	✗	✗
Kathara [27]	✗	Humble GUI	✓	✓	✓	✗
Megalos [18]	✗	Humble GUI	✓	✓	✓	✓
GNS3 [19]	✗	✓	✓	✓	✓	✓
SEED [35]	✗	✓	✓	✗	✓	✓
Mini-Internet [36]	✓	No GUI	✓	Limited	✗	✗
IPMininet [14]	✗	No GUI	✓	✓	✗	✗
Klonet (this work)	✓	✓	✓	✓	✓	✓

[1] **Teaching tools** are those designed to facilitate education, *e.g.*, Klonet's scene repository and Mini-Internet's connectivity matrix.
[2] **Experiment tools** are those designed to make experiments easier, *e.g.*, Klonet's traffic generator and IPMininet's IP configuration tools.

No existing emulator can achieve the two educational goals simultaneously

Klonet in a Nutshell

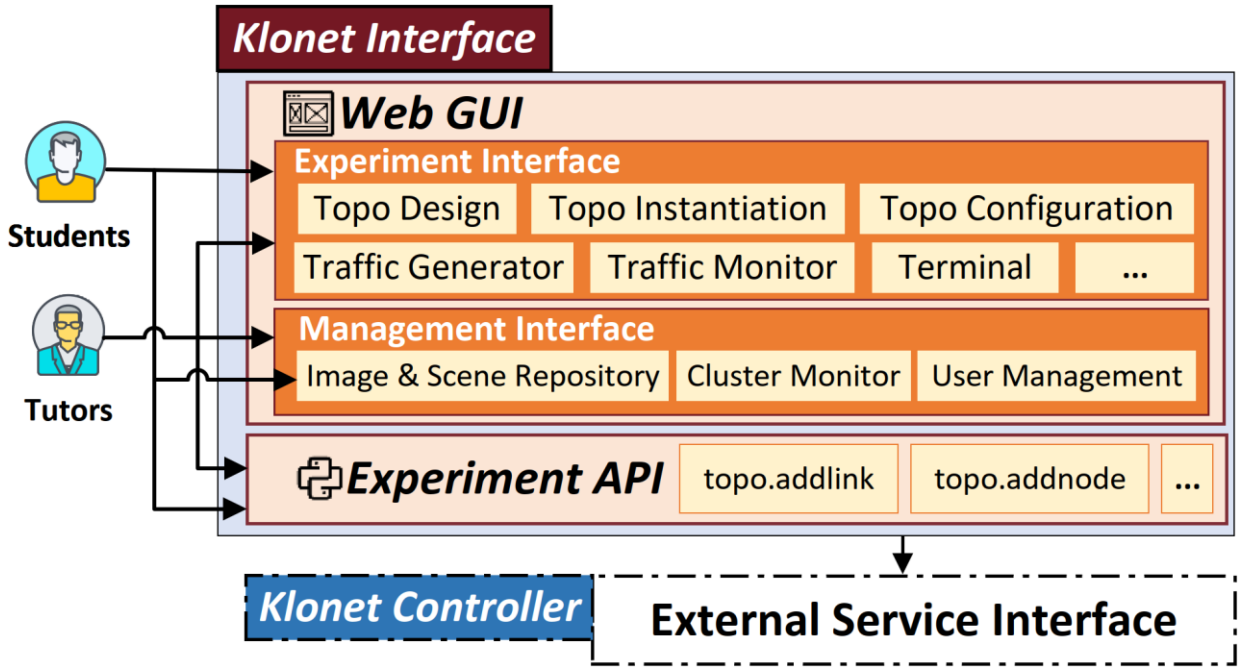
- Adopts a layered architecture
- A shared platform built on a physical cluster, can be easily accessed via its website
- Supports:
  - L2~L7 network experiments
  - Diverse scenarios, e.g. data center networks, wide-area networks
  - Customized topology
  - Mixture of real and virtual devices
  - Rich node types
  - …



Klonet Design Details

■ Achieving Easy-to-use

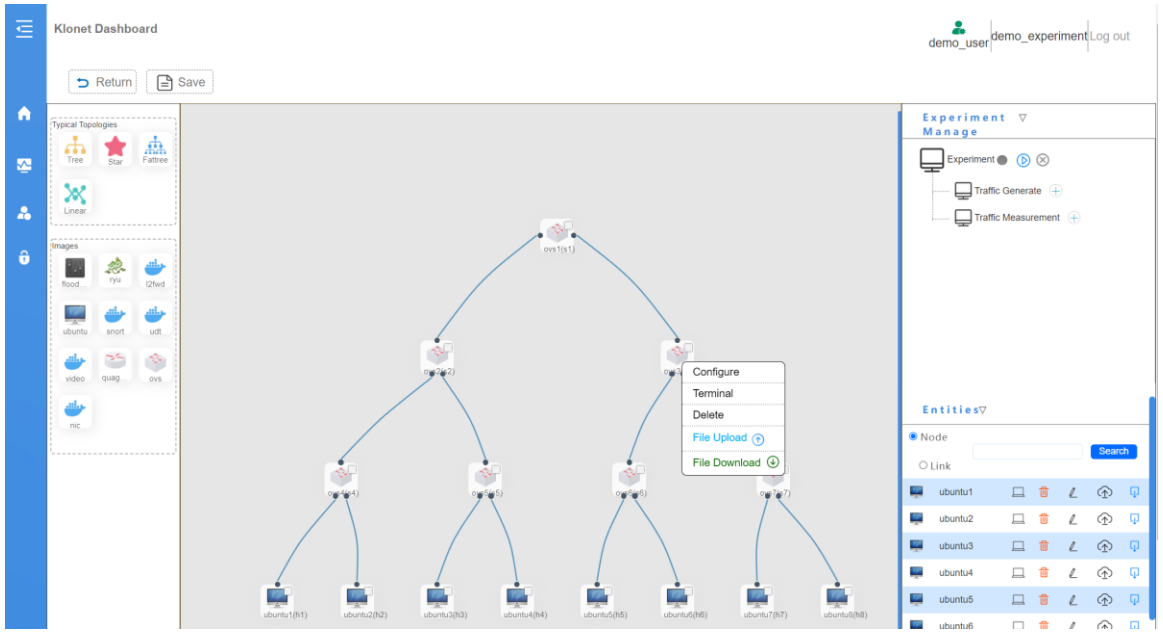
- Browser/Server (B/S) architecture to allow no installation



Klonet Design Details

■ Achieving Easy-to-use

- Browser/Server (B/S) architecture to allow no installation
- Rich Interfaces including Web GUI and Experiment API

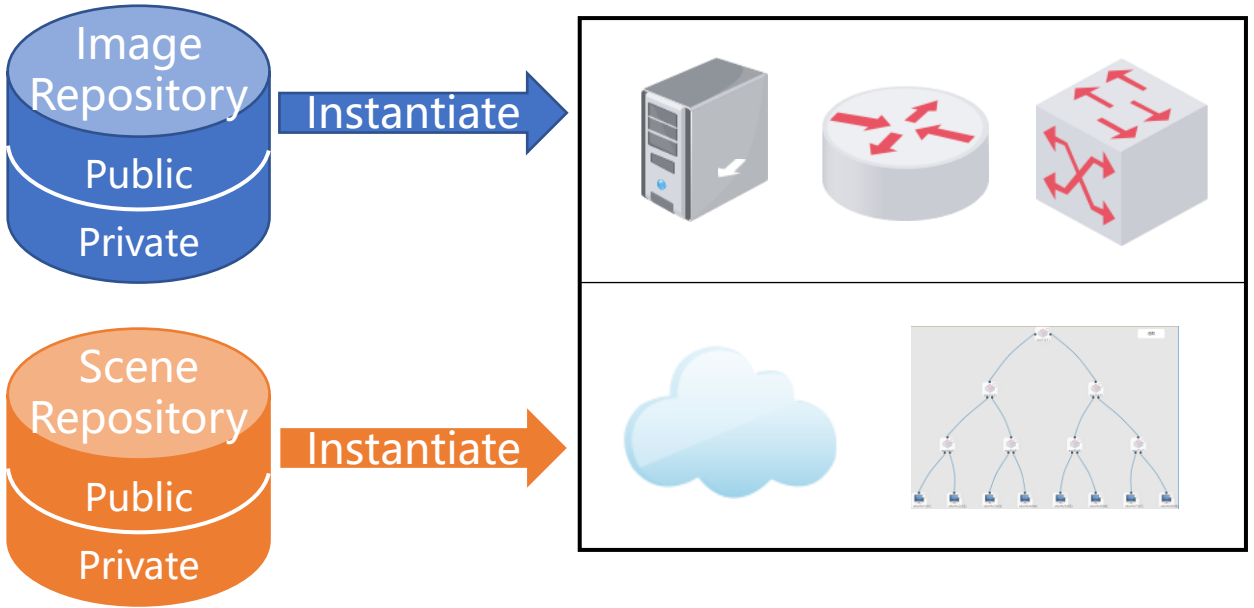



```python
from klonet_api import *
# Get the available images of current student.
images = get_images()
# Select the host(ubuntu) and switch(ovs) image.
ubuntu_image = images["ubuntu"]
ovs_image = images["ovs"]
# Design our topology: h1---s1---h2.
topo = Topo()
h1 = topo.add_node(ubuntu_image, node_name="h1")
h2 = topo.add_node(ubuntu_image, node_name="h2")
s1 = topo.add_node(ovs_image, node_name="s1")
topo.add_link(h1, s1, src_IP="192.168.1.1/24")
topo.add_link(s1, h2, dst_IP="192.168.1.2/24")
# Let Klonet emulate the topology.
deploy(topo)
# Create file in h1 and h2.
exec_cmds_in_nodes(
    {"h1":["touch /log1"], "h2":["touch /log2"]})
```

Klonet Design Details

■ Achieving Easy-to-use

- Browser/Server (B/S) architecture to allow no installation
- Rich Interfaces including Web GUI and Experiment API
- Building image and scene repository to make experiments shareable, scenario-rich, and quick to build



Klonet Design Details

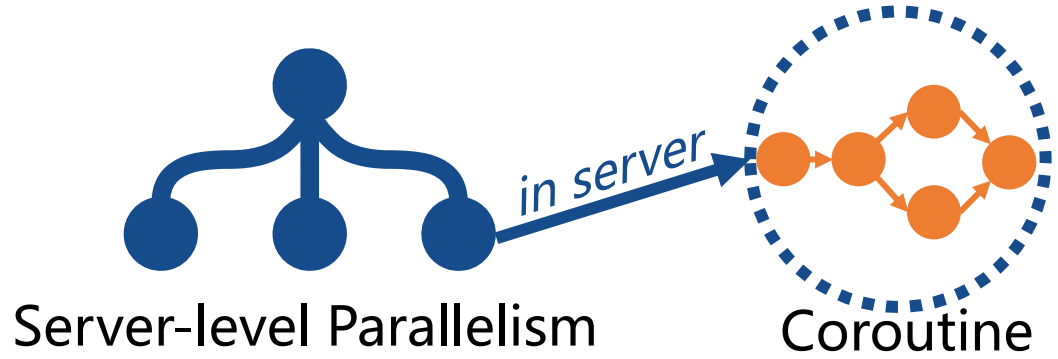
■ Achieving Easy-to-use

- Browser/Server (B/S) architecture to allow no installation
- Rich Interfaces including Web GUI and Experiment API
- Building image and scene repository to make experiments shareable, scenario-rich, and quick to build
- Built-in auxiliary tools (*e.g.* traffic generator, traffic monitor, and typical topology generator, ...) to facilitate experiments

Klonet Design Details

■ Achieving Easy-to-use

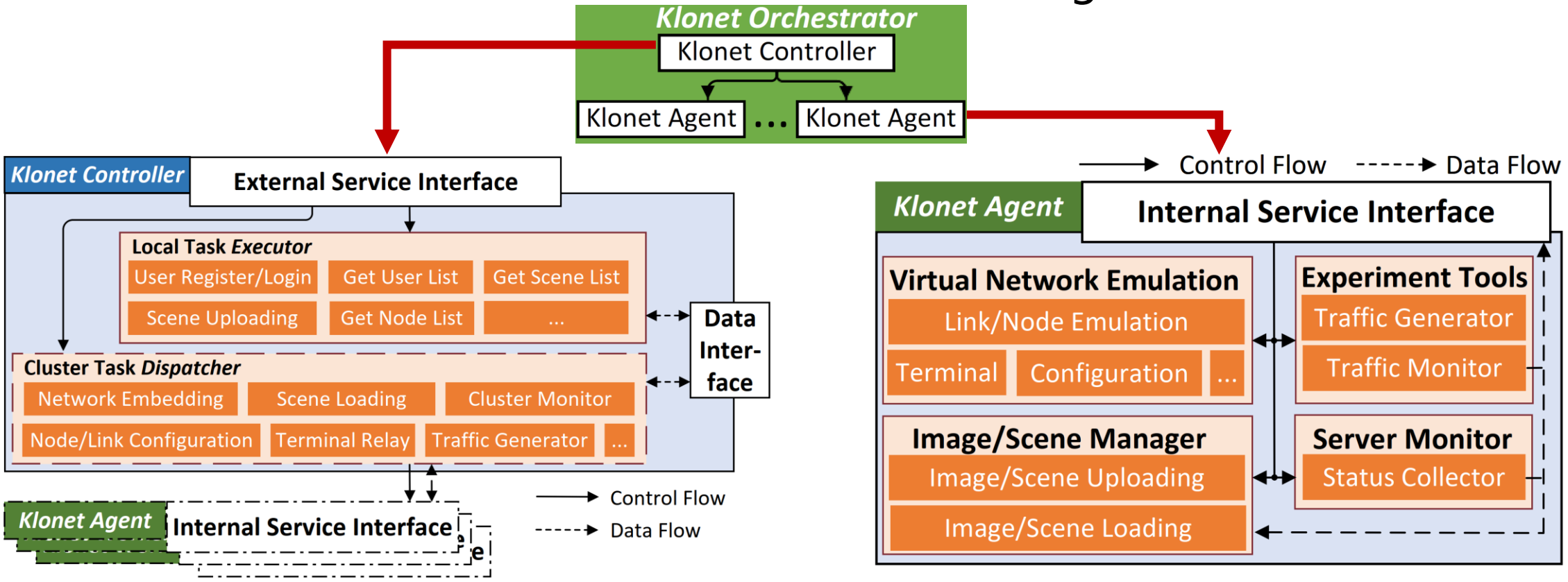
- Browser/Server (B/S) architecture to allow no installation
- Rich Interfaces including Web GUI and Experiment API
- Building image and scene repository to make experiments shareable, scenario-rich, and quick to build
- Built-in auxiliary tools to facilitate experiments
- Apply two parallelization techniques to accelerate creation



Klonet Design Details

■ Achieving Scalability

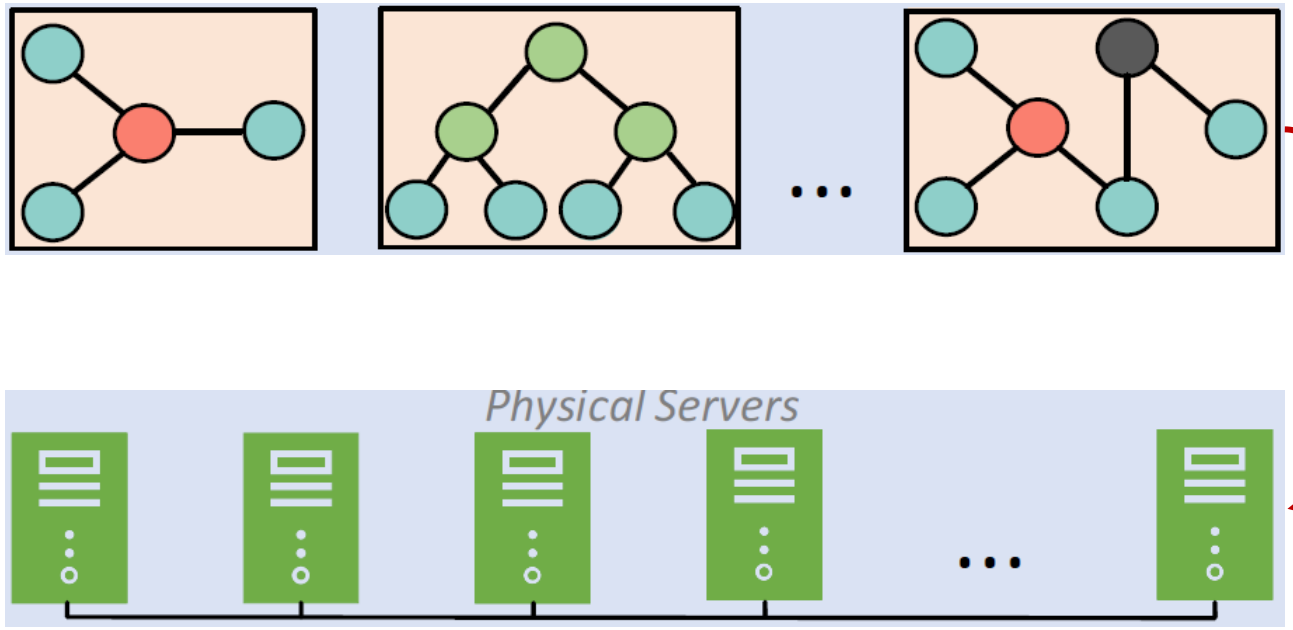
- Distributed orchestrator which enables scaling on clusters



Klonet Design Details

■ Achieving Scalability

- Distributed orchestrator which enables scaling on clusters
- Design a virtual network embedding algorithm to map virtual networks

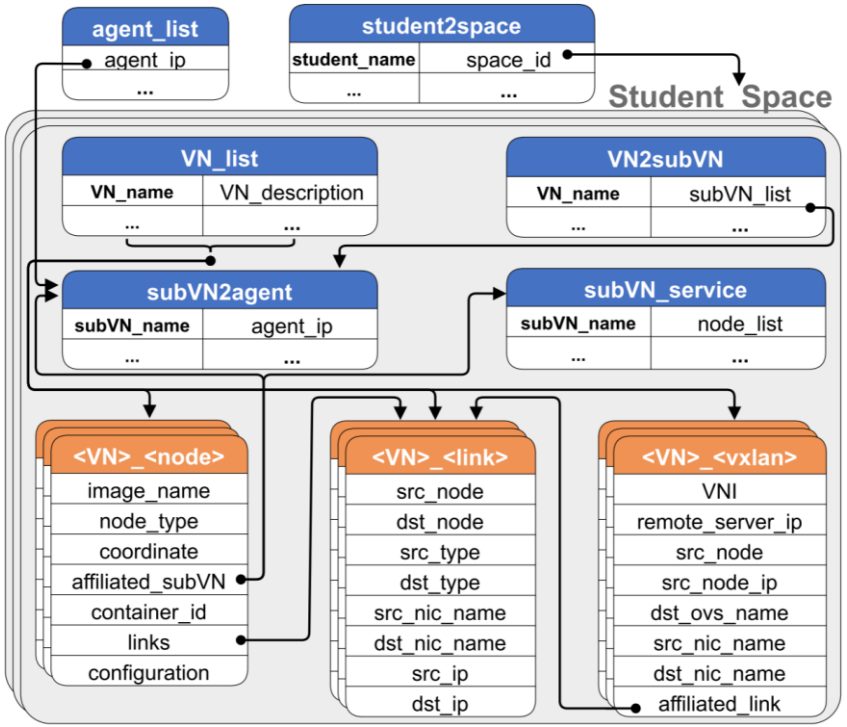


Embed virtual networks without overloading physical servers

Klonet Design Details

■ Achieving Scalability

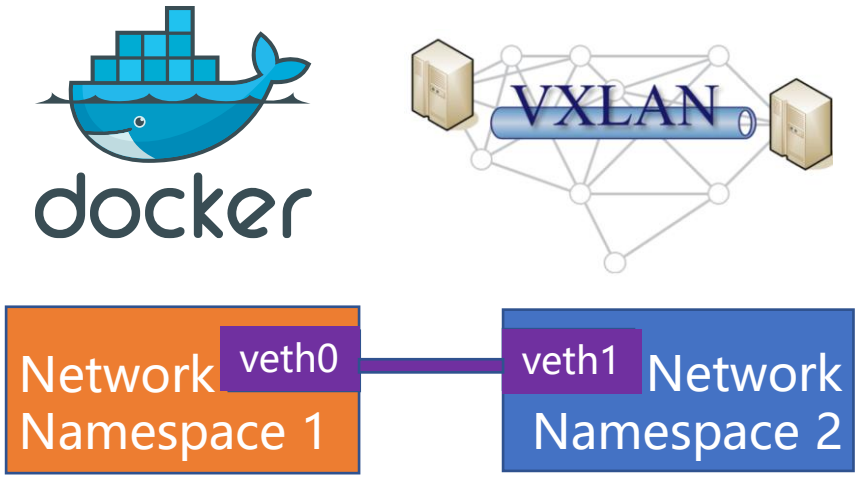
- Distributed orchestrator which enables scaling on clusters
- Design a virtual network embedding algorithm to map virtual networks
- Implement a user management model for multi-user data organization



Klonet Design Details

■ Achieving Scalability

- Distributed orchestrator which enables scaling on clusters
- Design a virtual network embedding algorithm to map virtual networks
- Implement a user management model for multi-user data organization
- Light-weight virtual network emulation
  - Node: Docker container
  - Link: Virtual Ethernet pair and VXLAN
  - Link Properties: Linux traffic control

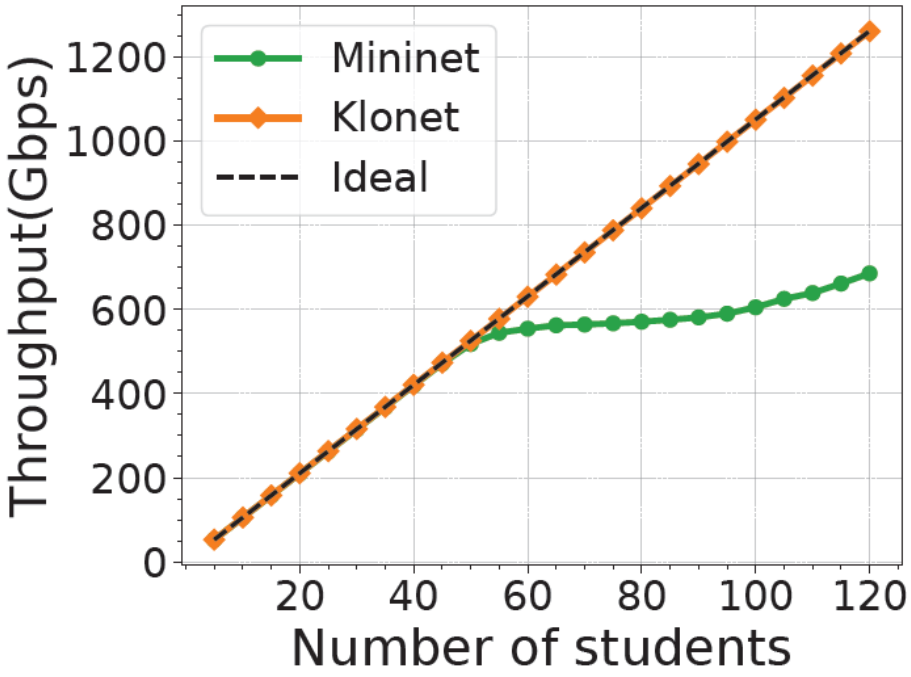


Klonet Design Details

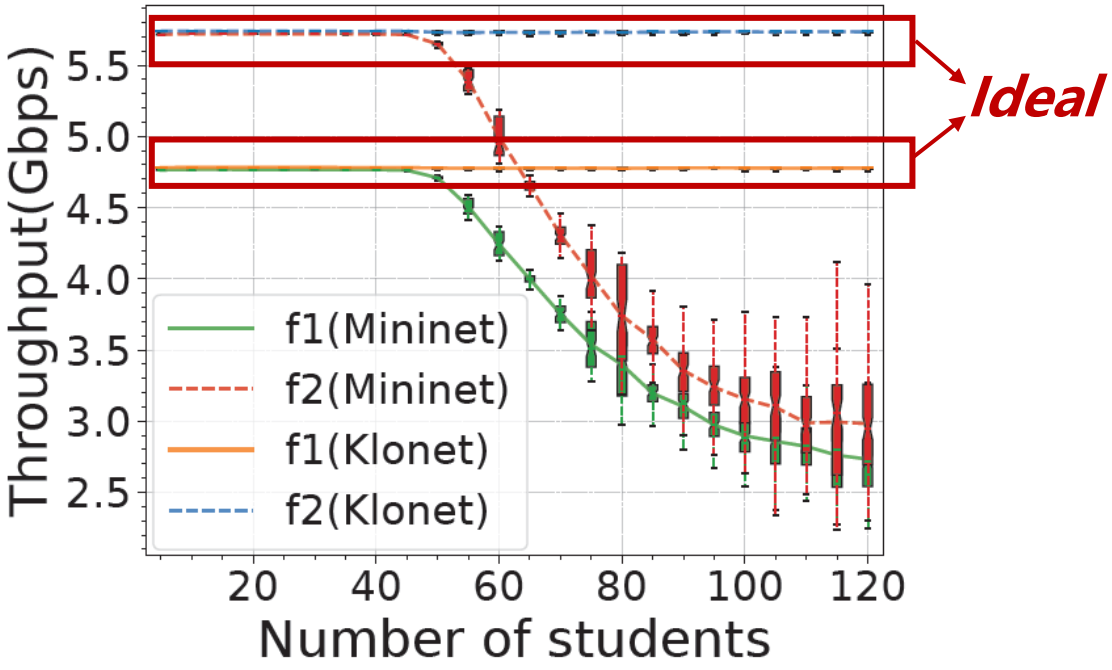
- **Robustness is important for a shared platform**
- **Achieving Robustness from top to bottom:**
  - Klonet orchestrator
  - User management model
  - Virtual networks
  - Cluster

See more details in our paper! (§4.4)

Fidelity Evaluation



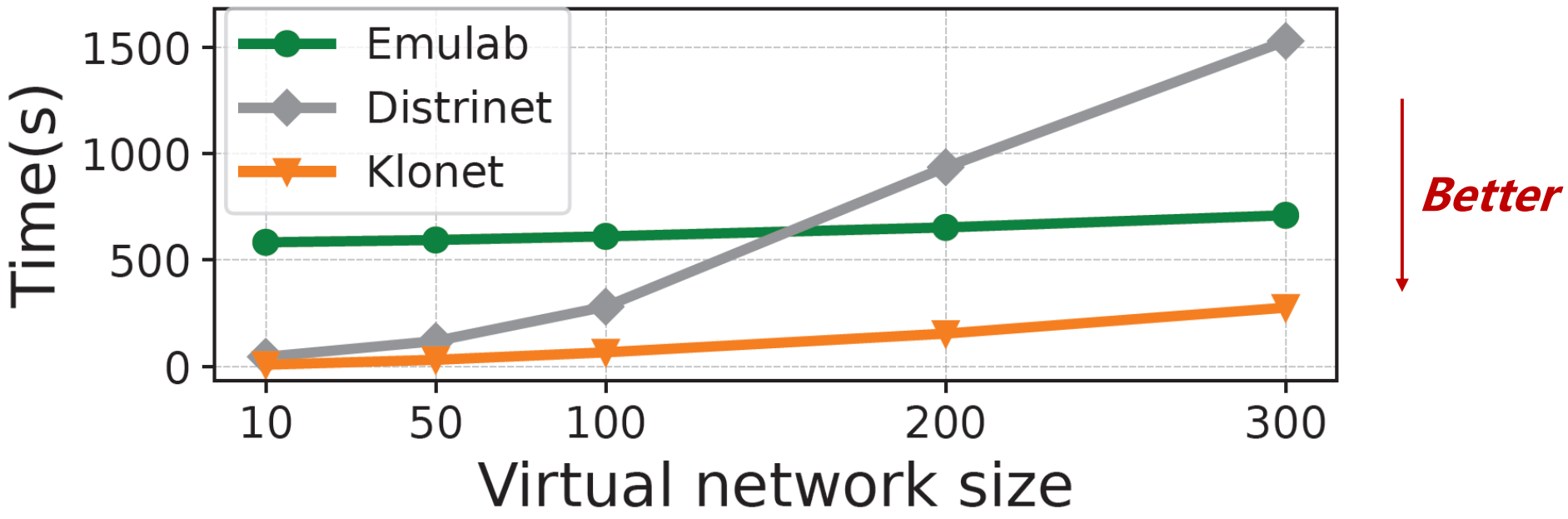
(a) Total throughput.



(b) Throughput distribution.

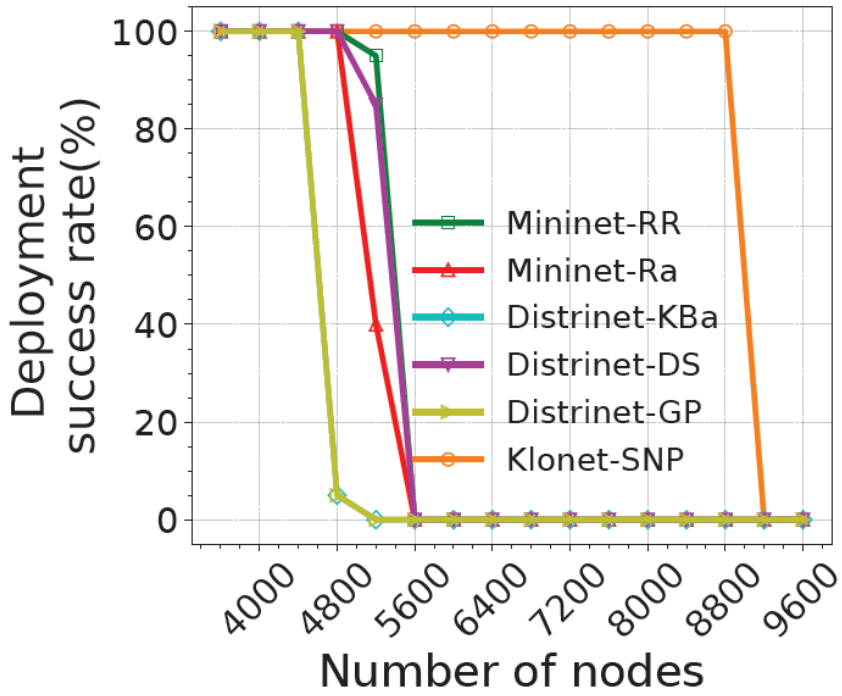
Klonet can support more students to conduct experiments simultaneously

Creation Time Evaluation

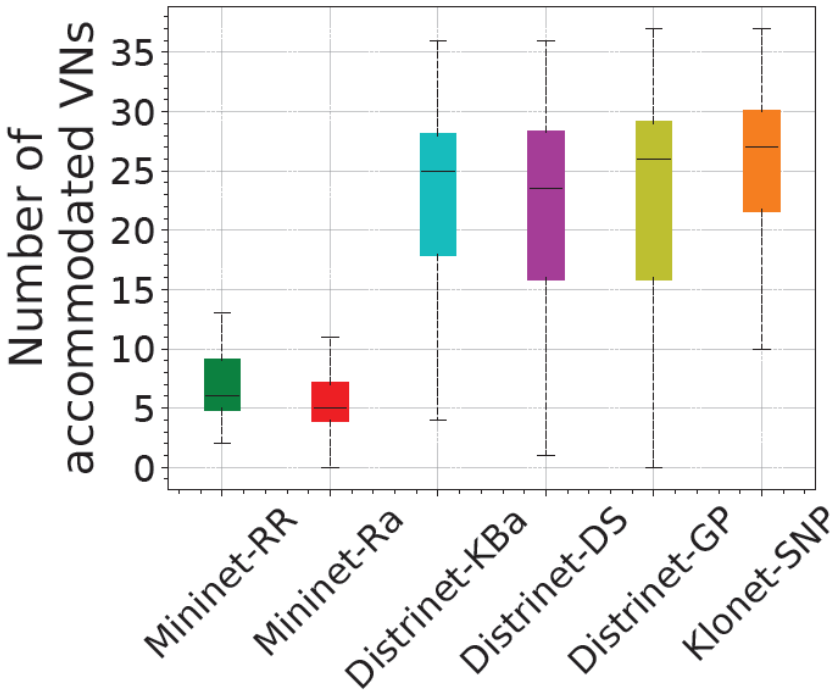


Klonet has a faster speed to deploy VNs

VNE Algorithm Evaluation



(a) Success rate.



(b) Success number.

Better

Klonet has a more efficient Virtual Network Embedding (VNE) algorithm

Use Cases

Implement algorithm in programmable networks

~150 stu./yr.

Networking virtual devices with Raspberry Pi

~40 stu./yr.

Implement network configuration such as NAT

~80 stu./yr.

Collaborate to achieve routing

~30 stu./yr.

Self-learners verify basic knowledge

~100 stu./yr.

Even several research works

E.g. a work using Klonet has been published in RTSS 2022

Use Cases

Implement algorithm in programmable networks

~150 stu./yr.

Project Ⅰ: Playing with algorithms (§6.1)

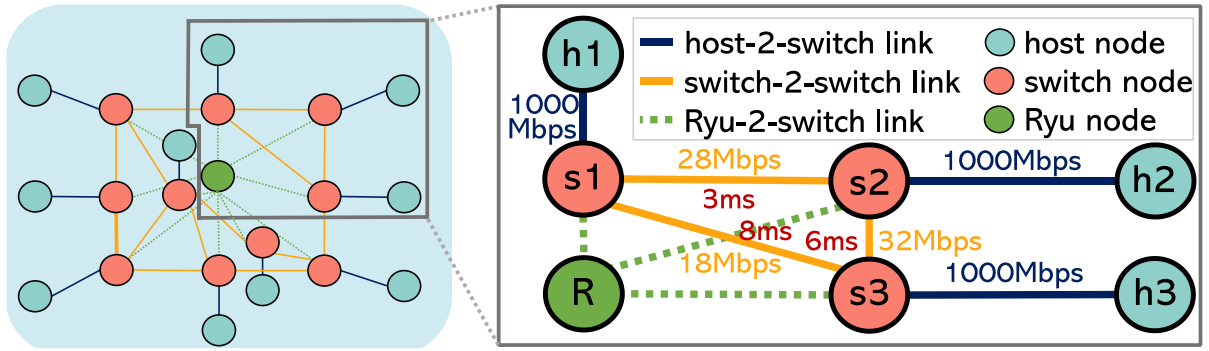
Collaborate to achieve routing

~30 stu./yr.

Project Ⅱ: Intra-domain Routing (§6.2)

Use Cases -- Project Ⅰ: Playing with algorithms

- Help students gain practical experience with algorithm performance in a programmable network

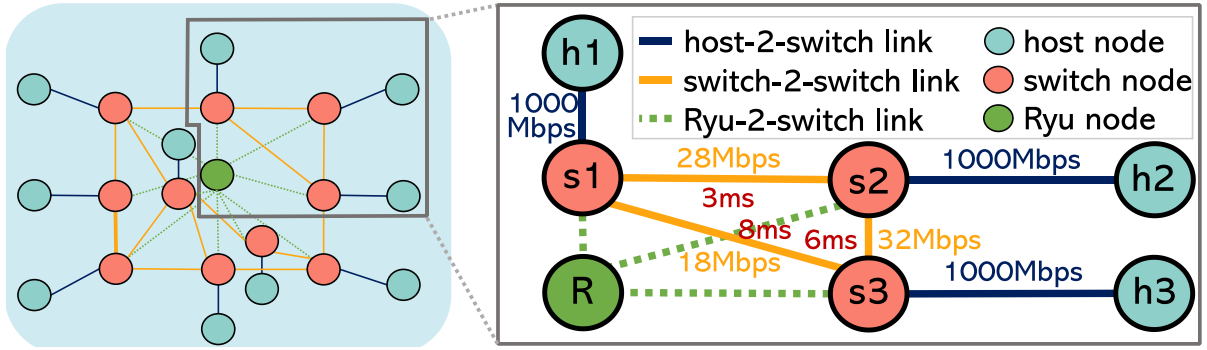


- **Learning outcomes**
  - Understand how OpenFlow works
  - Write algorithms in OpenFlow controllers
  - Identify the performance of algorithms

Challenges:

1. How to focus on learning SDN networks rather than building them?
2. How to quickly replay experiments for self-improvement or tutor assessment?
3. How to interact with the network, control it, and observe its performance?

Use Cases -- Project Ⅰ: Playing with algorithms



Challenges:

1. *How to focus on learning SDN networks rather than building them?*
2. *How to quickly replay experiments for self-improvement or tutor assessment?*
3. *How to interact with the network, control it, and observe its performance?*

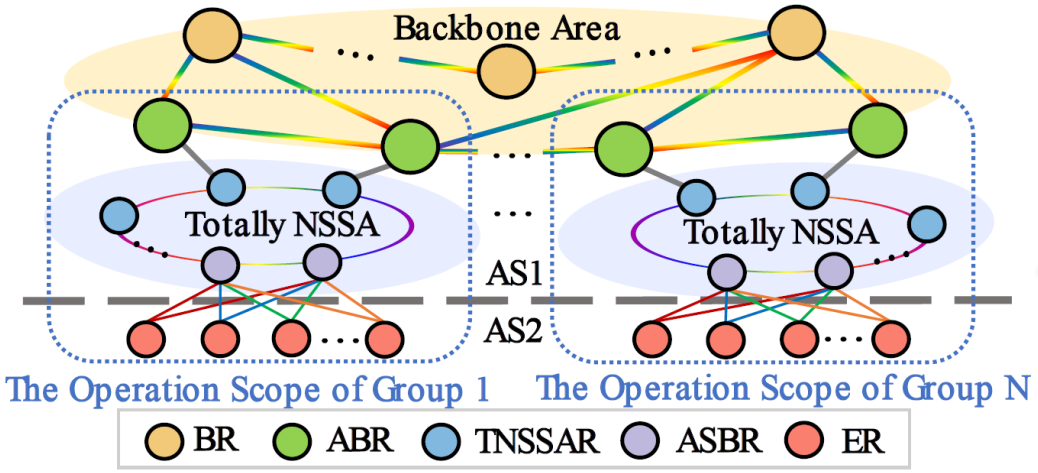
■ **Using Klonet's**

- **Image repository:** tutors can design new experiments quickly (*Address challenge 1*)
- **Scene repository:** students can start and refine experiments easily, TAs can correct and grade experiments easily (*Address challenge 1, 2*)
- **Web terminal, SSH connection:** students can program networks, control networks, and interact with networks easily (*Address challenge 3*)

Klonet holds great usability for both tutors and students

Use Cases -- Project Ⅱ: Intra-domain Routing

- **Help students gain a hands-on experiment that involves realistic network operations**

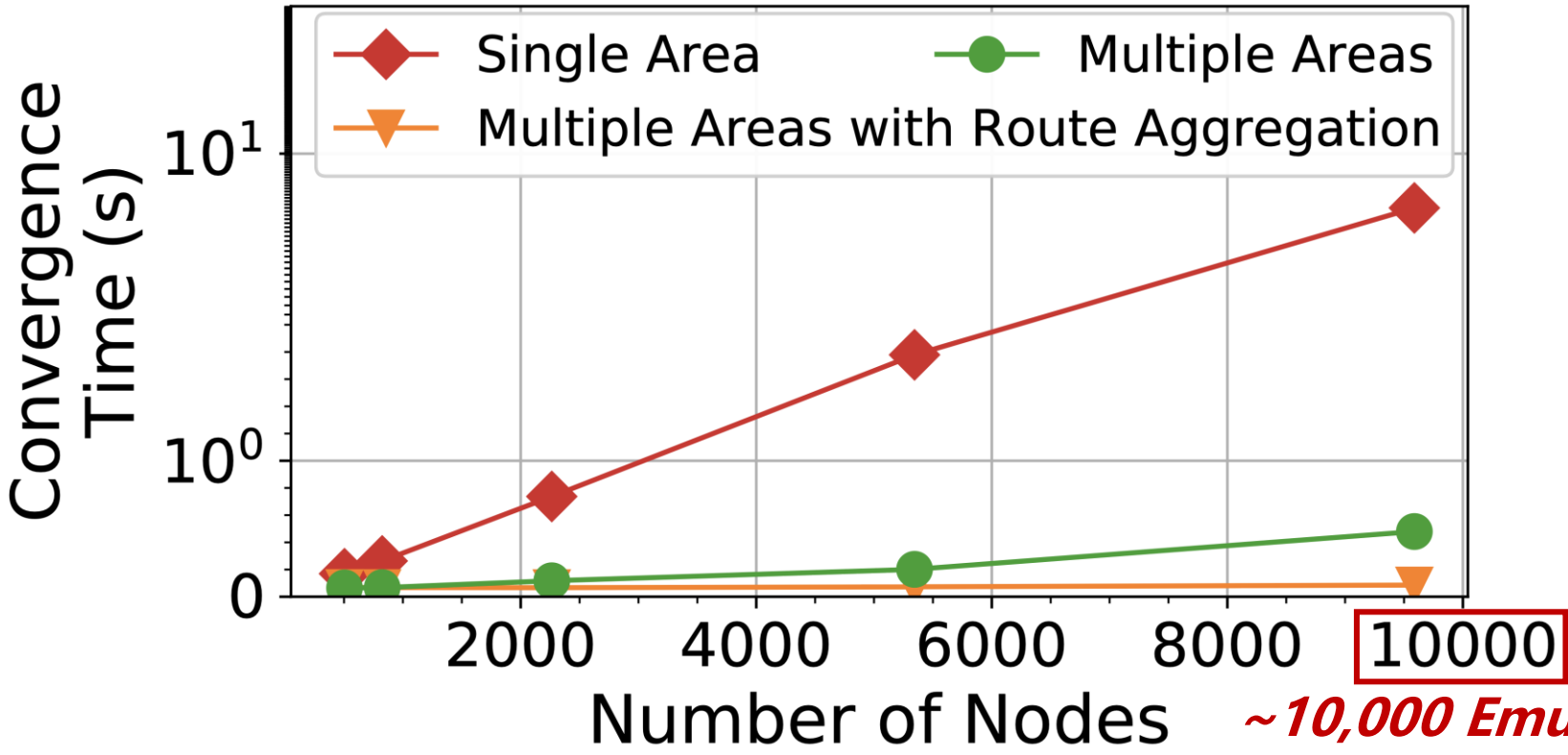


Challenges:

1. Massive scale as if in an enterprise network (up to ~10,000 routers)
2. Configure as if in an enterprise network
3. Observe as if in an enterprise network

- **Learning outcomes**
  - Build and operate enterprise network
  - Have a deeper understanding of OSPF and splitting areas
  - Observe and learn the benefits of route aggregation

Use Cases -- Project Ⅱ: Intra-domain Routing



~10,000 Emulated Routers

Klonet's virtual networks can scale to a very large size while maintaining the network's configurability and observability

Conclusion

- We present **Klonet**, an **easy-to-use** and **scalable** platform for computer networks **education**
- Klonet has been in development for **4 years** and in operation for **2 years**
- Klonet has been adopted in **3 universities** and **4 courses**, serving more than **800 students**
- **We call for more attention to be invested in computer networks education, for the future of our community!**

Thank You!

✉ **yuhf@uestc.edu.cn**

✉ Or if you have a WeChat account:

Welcome to contact us for discussion or cooperation!