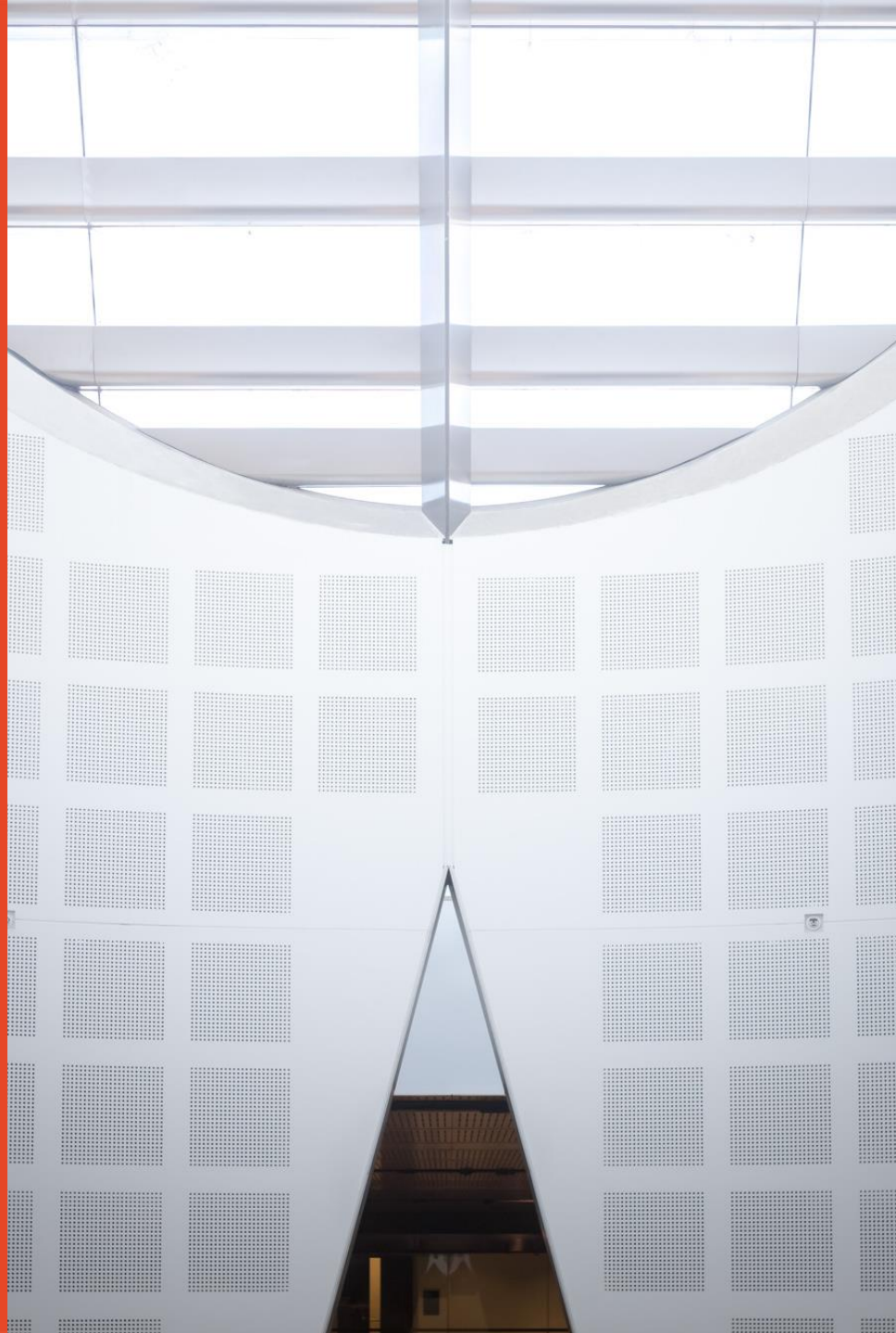
Topic 2: Mininet

Presented by
Dong YUAN

School of Electrical and Computer Engineering

dong.yuan@sydney.edu.au

THE UNIVERSITY OF SYDNEY



Network Emulator

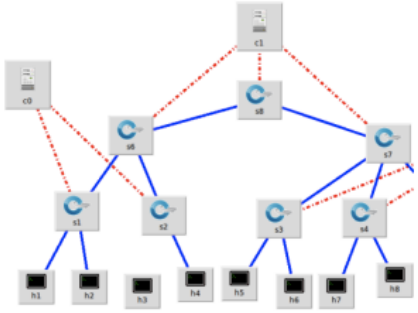
- Education in computer networks
  - Large system with complex structures/layers
  - Practice is important

- Choices of experiment platforms

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

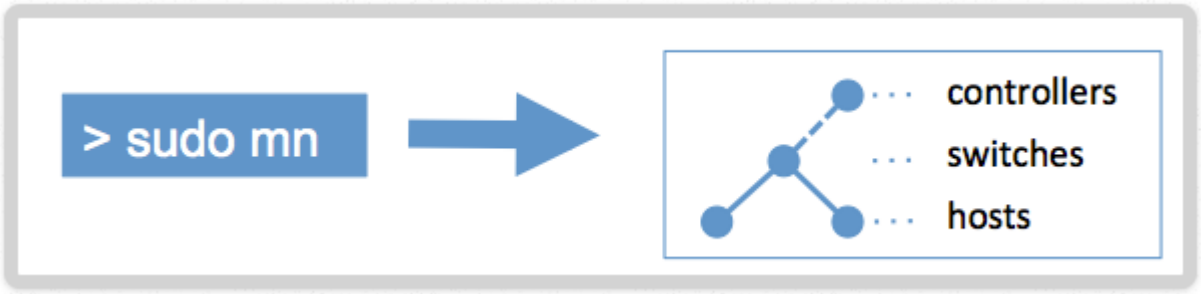
Platforms for Network/Systems experiments

- Hardware Testbed
  - Fast and accurate
  - Expensive, hard to config
- Simulator
  - Inexpensive, flexible, faster even than reality
  - In accurate, cannot be directly used in production environment
- Emulator
  - Inexpensive, flexible, reasonable accurate
  - Slow, some changes may be required to be used in reality

Mininet

- A Network Emulator

- Creates a virtual network on a single machine with a single command
- Virtual network is realistic and can be deployed on real hardware



Mininet (contd.)

- Uses Linux virtual network features
- Arbitrary topologies and nodes
- Research and development of networks
- Test, analyze and predict network behavior
  - Ease of examining network changes before testing/deploying on hardware
  - Recreating real-world network and test cases for a variety of topologies and configurations
- Minimal hardware requirements

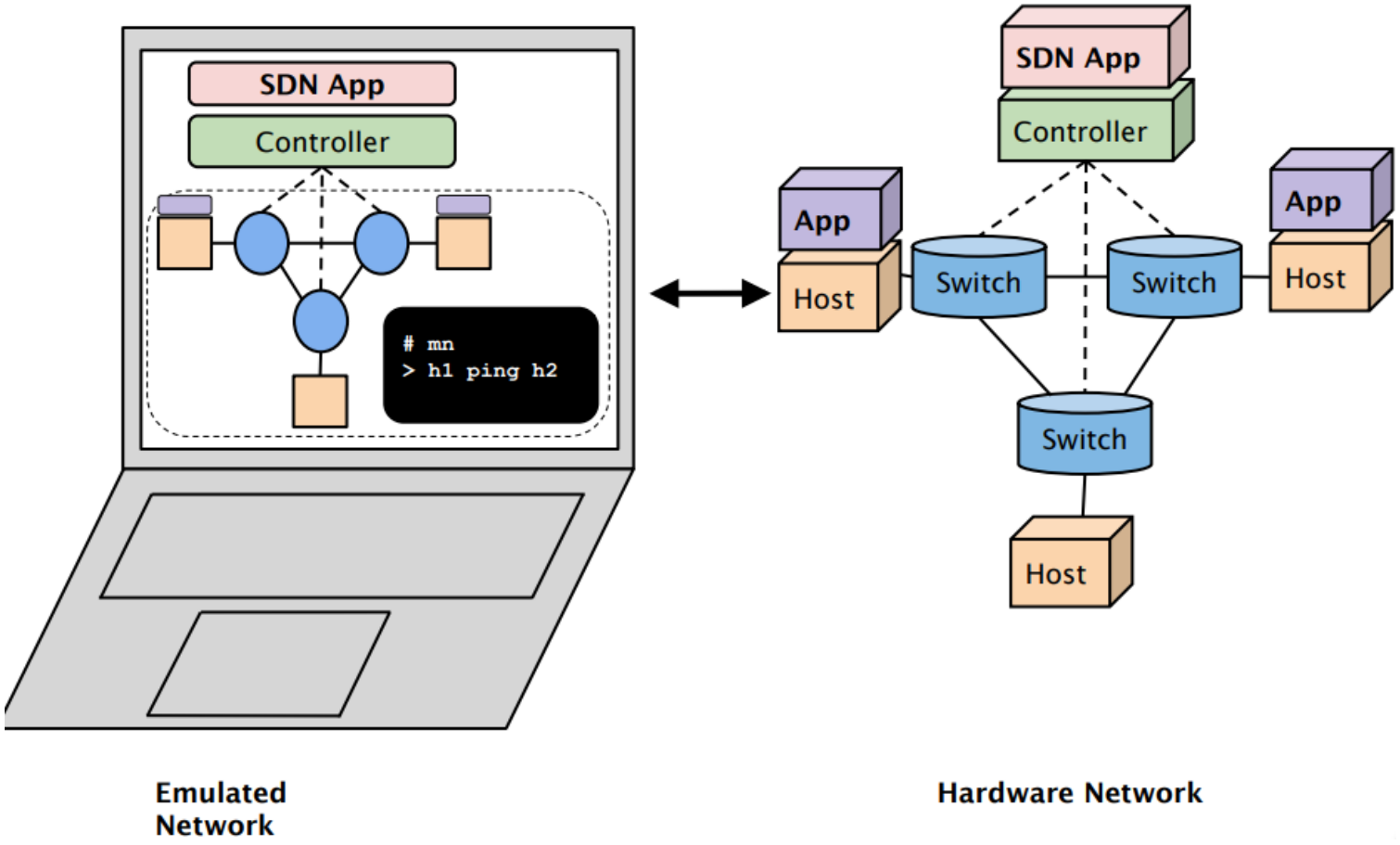
Mininet: Basic Commands and OpenFlow View

- Basic commands:
  - Display an xterm for switch s1
    - mininet> xterm s1

- To view OpenFlow protocol messages, at mininet-VM xterm:
  - sudo wireshark &
  - Capture the interface to controller
  - In wireshark filter box, enter filter to filter OpenFlow messages

Mininet: Apps and Hardware Integration

– Seamless movement of Apps to/from hardware



Mininet: Positives

- Provides a simple and inexpensive network testbed for developing OpenFlow Applications
- Enables complex topology testing (without need to wire up a physical network)
- Multiple concurrent developers can work independently on the same topology
- Usable out of the box without programming
- Includes a topology-aware Command Line Interface (CLI) for running or debugging networks
- Supports system-level regression tests (verifies that the previously developed and tested network still performs the same way after changes)

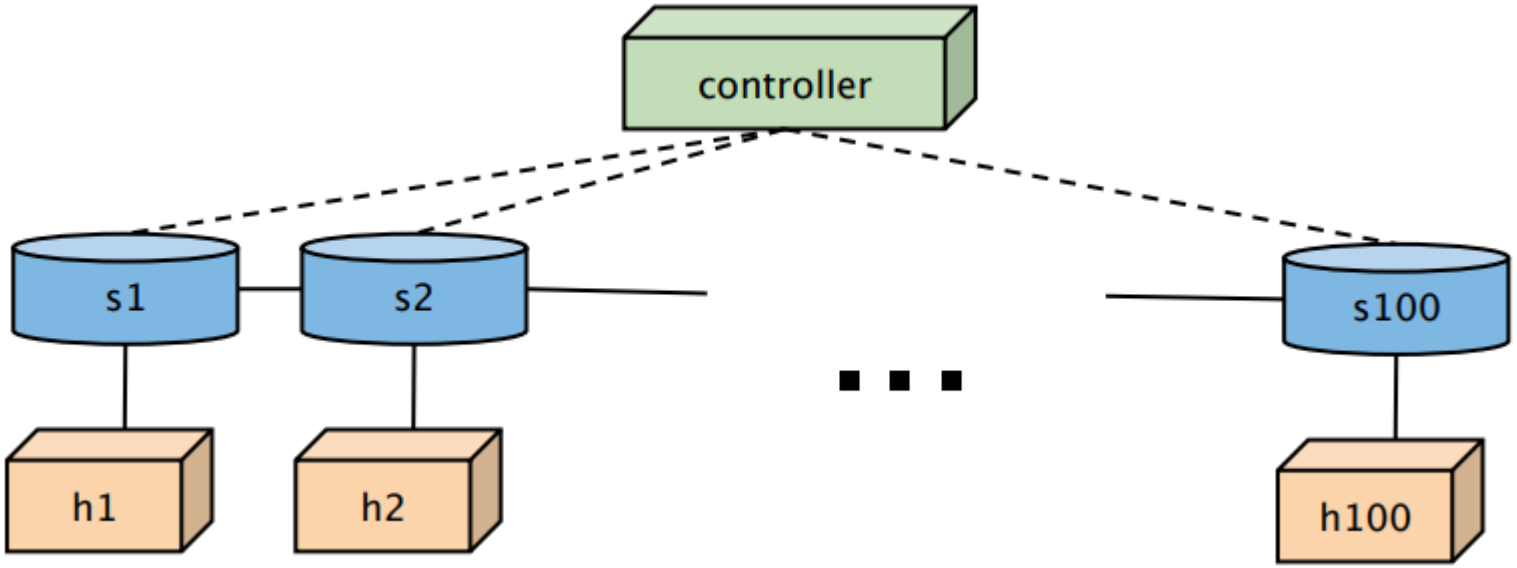
Comparing Mininet to System Virtualization, Hardware Testbeds and Simulators

- Compared to System Virtualization
  - Boots within seconds; Installs easily; Provides adequate bandwidth; Scales larger.
- Compared to hardware testbeds
  - Inexpensive; always available; quick configuration and reconfiguration
- Compared to Simulators
  - Easy connections to real networks; interactive performance; real and unmodified code.

Mininet: Advantages

– Scalability: huge network with practical performance

mn --topo linear,100 --switch user --controller ref

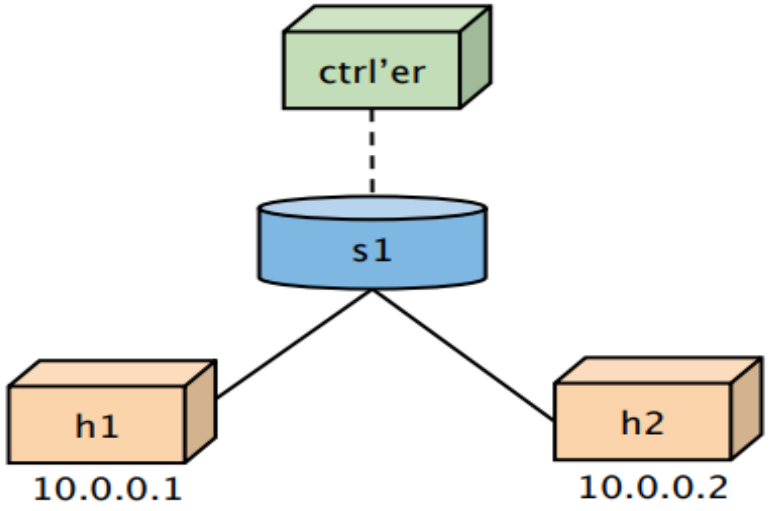


Mininet: Advantages (contd.)

– Ease of Use: having a simple command line tool and/or API that can automatically handle the below basic network setup.

```
sudo bash
# Create host namespaces
ip netns add h1
ip netns add h2
# Create switch
ovs-vsctl add-br s1
# Create links
ip link add h1-eth0 type veth peer name s1-eth1
ip link add h2-eth0 type veth peer name s1-eth2
ip link show
# Move host ports into namespaces
ip link set h1-eth0 netns h1
ip link set h2-eth0 netns h2
ip netns exec h1 ip link show
ip netns exec h2 ip link show
# Connect switch ports to OVS
ovs-vsctl add-port s1 s1-eth1
ovs-vsctl add-port s1 s1-eth2
ovs-vsctl show
# Set up OpenFlow controller
ovs-vsctl set-controller s1 tcp:127.0.0.1
ovs-controller ptcp: &
ovs-vsctl show
```

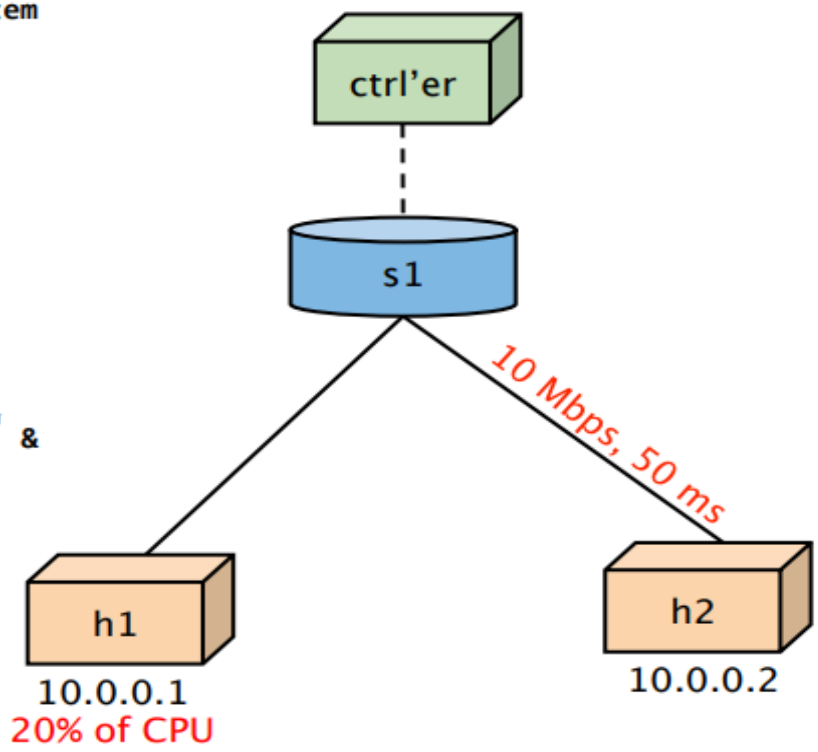
```
# Configure network
ip netns exec h1 ifconfig h1-eth0 10.1
ip netns exec h1 ifconfig lo up
ip netns exec h2 ifconfig h2-eth0 10.2
ip netns exec h1 ifconfig lo up
ifconfig s1-eth1 up
ifconfig s1-eth2 up
# Test network
ip netns exec h1 ping -c1 10.2
```



Mininet: Advantages (contd.)

- Performance: experiments should match results on hardware.
  - Performance setup in Linux

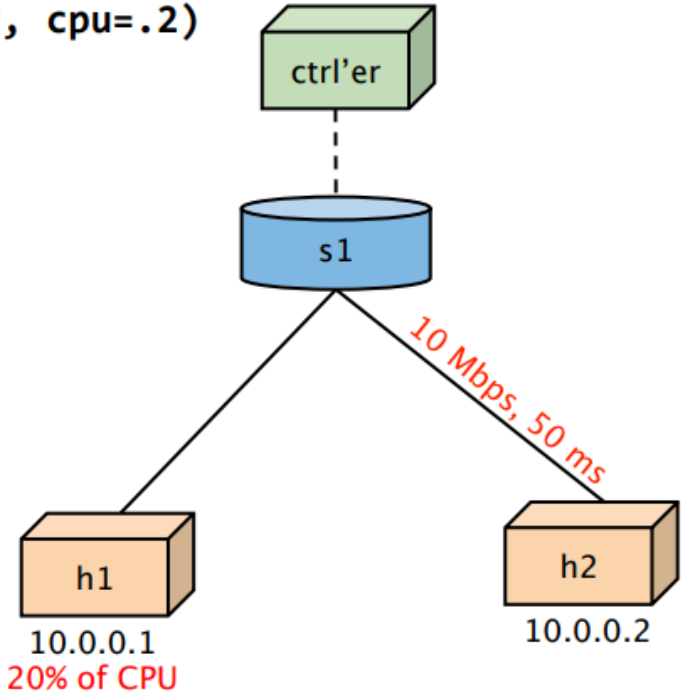
```
# Limit link bandwidth and add delay
tc qdisc add dev s1-eth2 root handle 5: tbf rate
  10Mbit burst 5k latency 12ms
tc qdisc add dev s1-eth2 parent 5:1 handle 10: netem
  delay 50ms
ip netns exec h1 ping -c4 10.2
ip netns exec h2 iperf -s >& /dev/null &
ip netns exec h1 iperf -t 5 -c 10.2
# Limit CPU bandwidth
cgcreate -g cpu:/h1
cgset -r cpu.cfs_period_us=100000 /h1
cgset -r cpu.cfs_quota_us=20000 /h1
ip netns exec h1 bash -c "while true; do a=1;done" &
cgclassify -g cpu:/h1 $!
```



Mininet: Advantages (contd.)

- Performance setup in Mininet

```
# Limit link bandwidth and add delay
net.addLink(h2, s1, cls=TCLink,
            bw=10, delay='50ms')
# Limit CPU bandwidth
net.addHost('h1', cls=CPULimitedHost, cpu=.2)
```



Mininet: Limitations

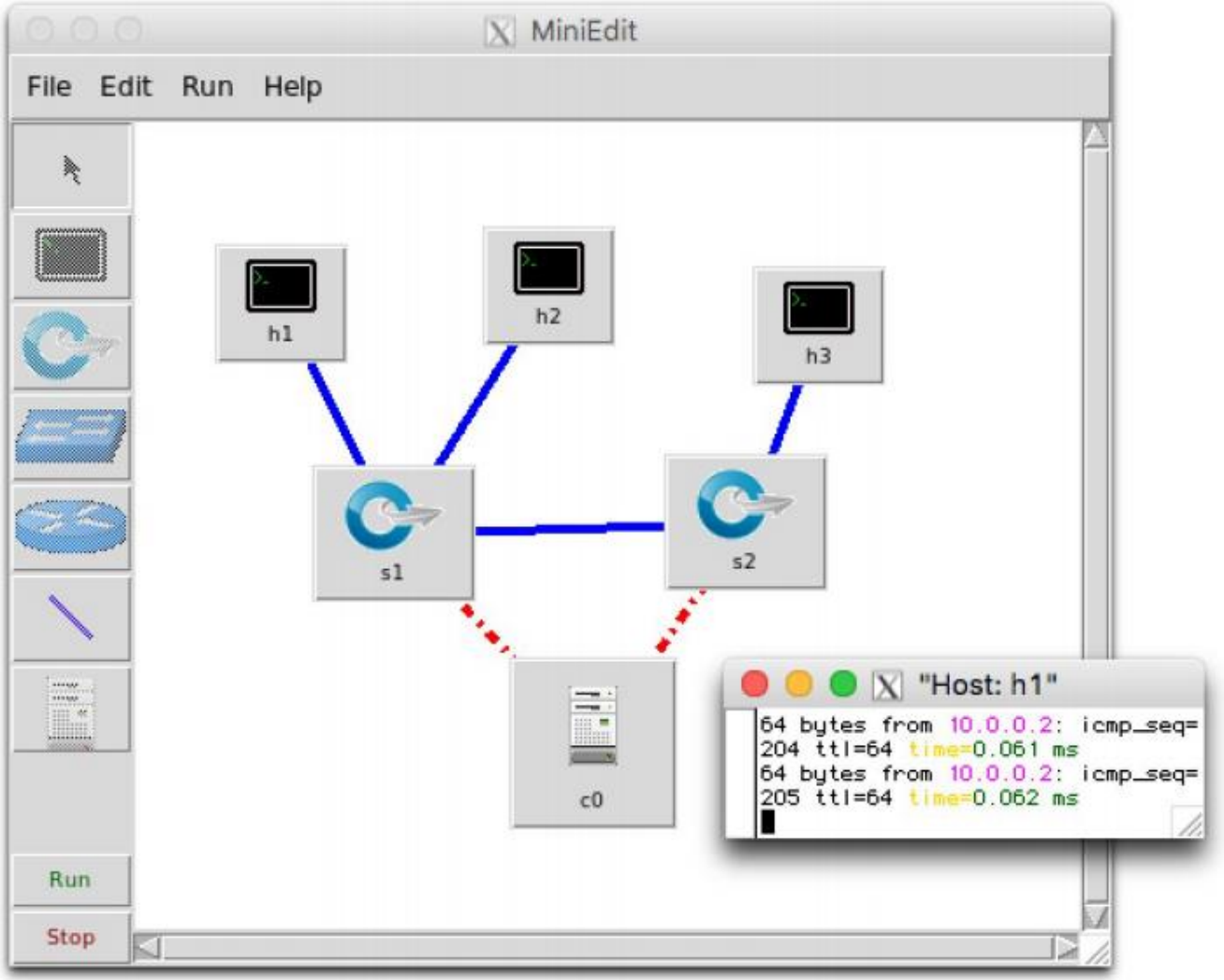
- Networks in mininet cannot exceed the CPU or bandwidth available on a single server. For a server with 3 GHz of CPU that can switch about 10 Gbps of simulated traffic, those resources need to be balanced and shared among the virtual hosts and switches
- At the moment, mininet cannot run non-Linux compatible OpenFlow switches or applications. However, in practice this has not been a major issue.
- If you need custom routing or switching behavior, Mininet won't write your OpenFlow controller for you. You will need to find or develop a controller with the features you require.

Mininet: Additions

- Ease of use
- Free and permissively Berkeley Software Distribution (BSD) Open Source license
- Strong users and support community
- Parametrized topologies
- Simple Python Application Programming Interface (Python API)
- Predictive Accuracy
- Easy interface for contribution on GitHub
- Useful for development, teaching and research

MiniEdit

– Mininet's Graphical User Interface (Mininet GUI)

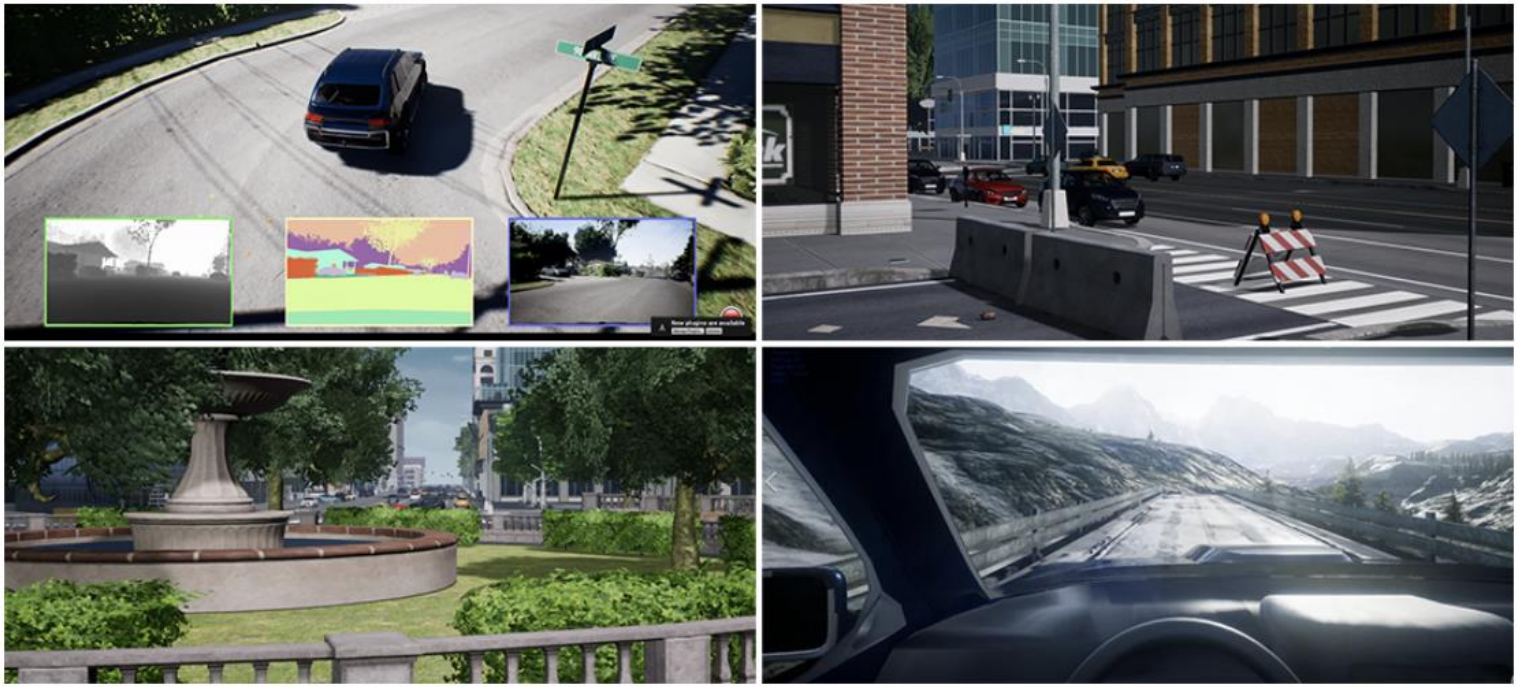


Other Network Emulators

- Netkit
- Kathará
- GNS3
- Mini-Internet
- SEED
- IP-mininet
- Klonet (very new, published in NSDI 2024)

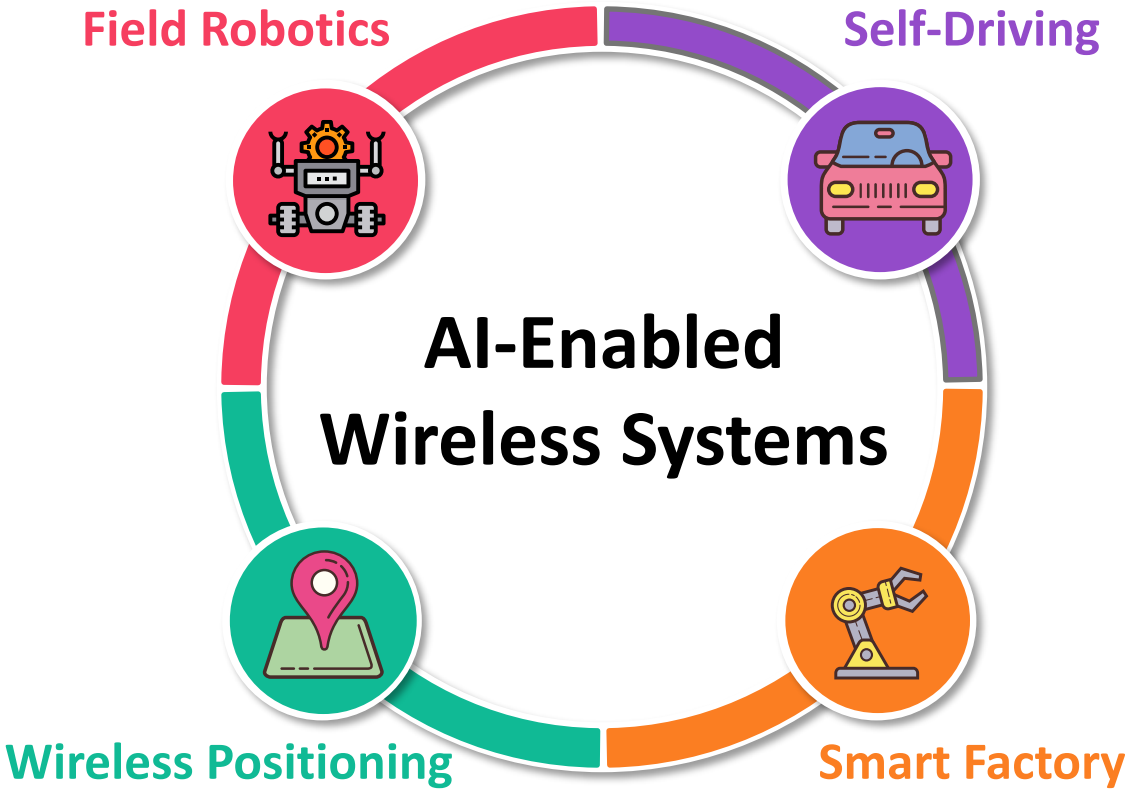
Simulators in other areas

- Deep Reality Simulation - AirSim
  - Autopilot
  - UAV
  - Poacher Detection (nature, war, etc.)



Our Work: WirelessDT

A Digital Twin Platform for AI-Enabled Wireless Systems



Industry Professionals
Modelling, Managing, Monitoring

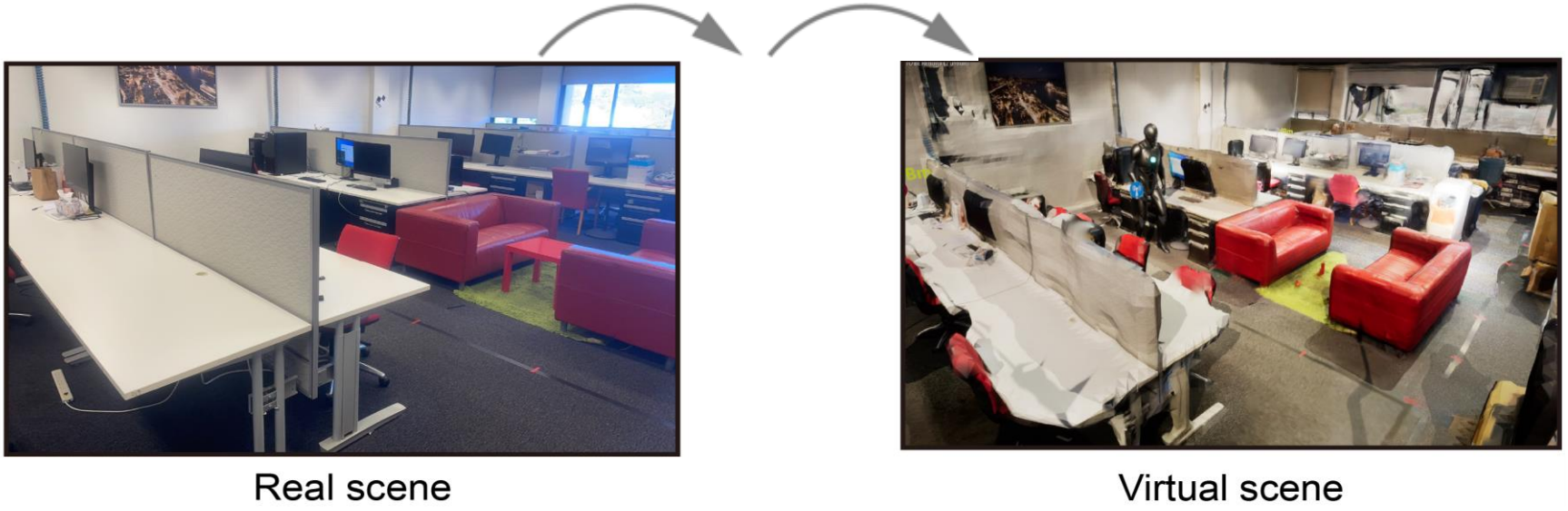
Software Engineers
Building, Testing, Debugging

Researchers
Innovating, Evaluating, Analyzing

Environmental Reconstruction

- To generate realistic virtual scenes efficiently for simulation, we propose a NeRF-based approach, integrated with a Monte Carlo-based calibration process.

- To enhance simulation accuracy, we synchronize the real-world measurement data to the simulator and employ a Monte Carlo sampling technique to calibrate material parameters, such as the signal attenuation factor of walls

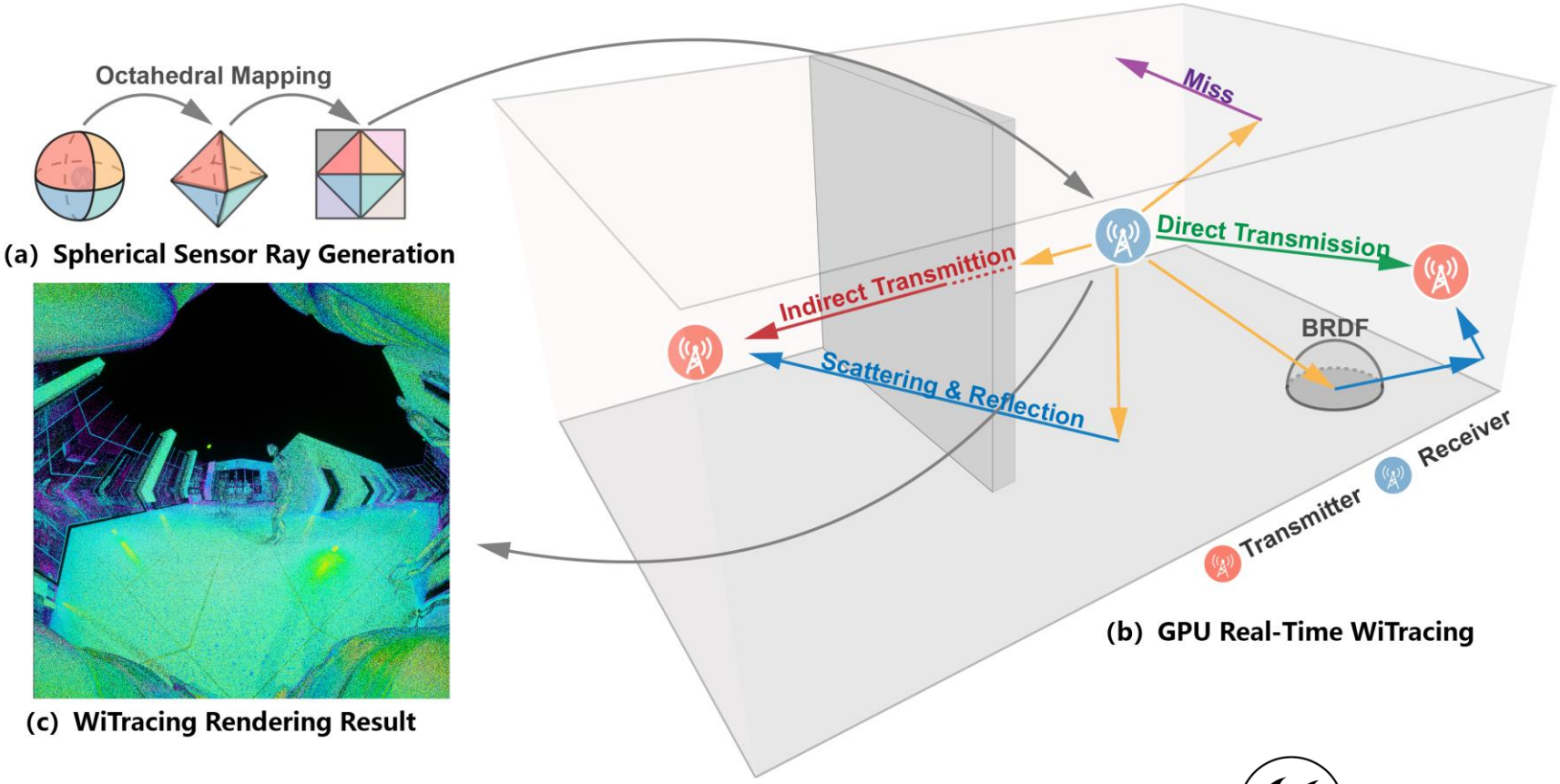


Real scene

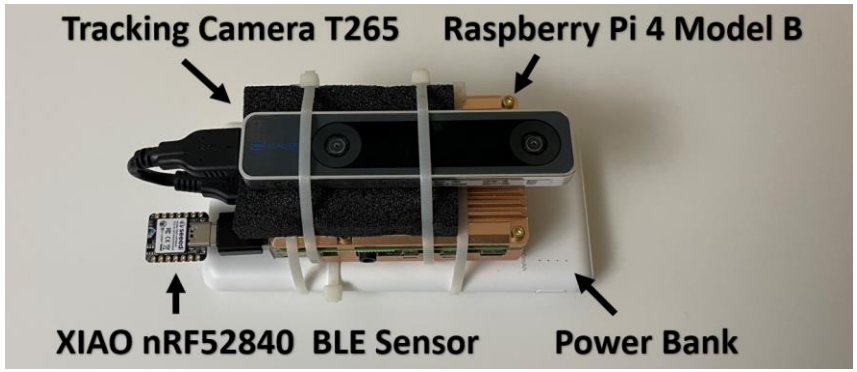
Virtual scene

Render the wireless signal with WiTracing Engine

Overview of the WiTracing process workflow

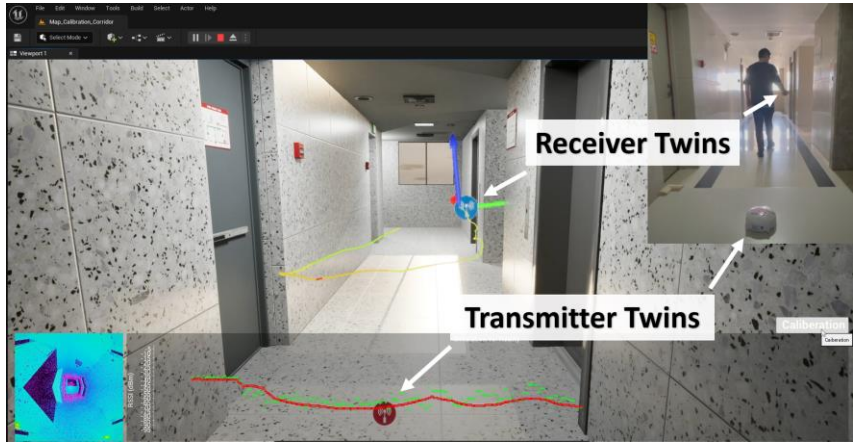


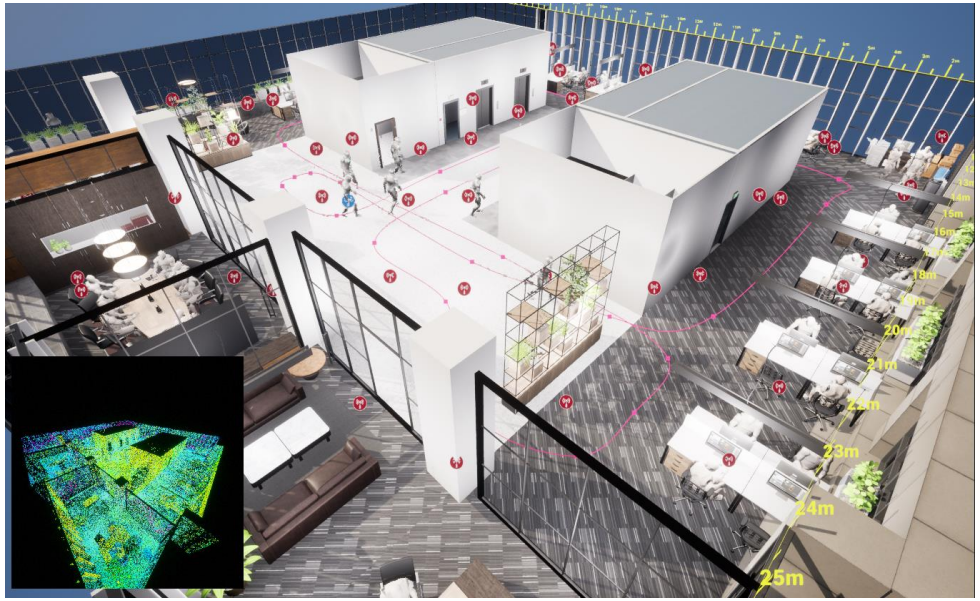
A Demo for Indoor Localisatin



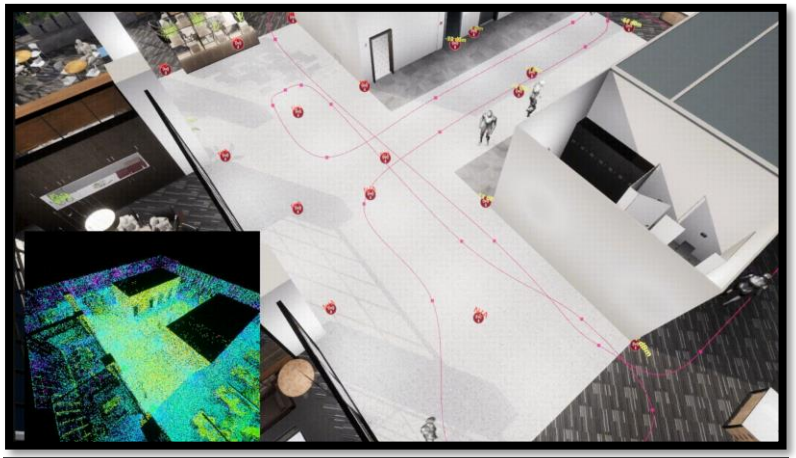
Full Demo Video

https://youtu.be/9KI-3jgMBUA





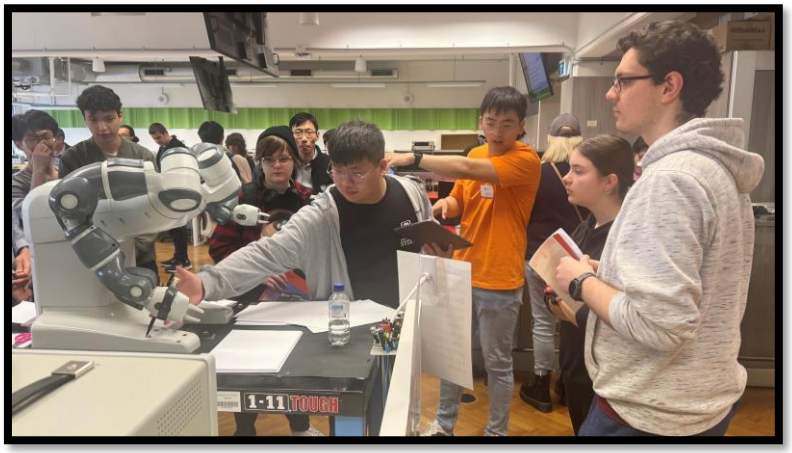
Some Demos



GPU-Based Real-Time Wireless Simulation Technology

Various Industry Applications Scenarios

High Recognition in Top Conference (ICSE and SigComm)

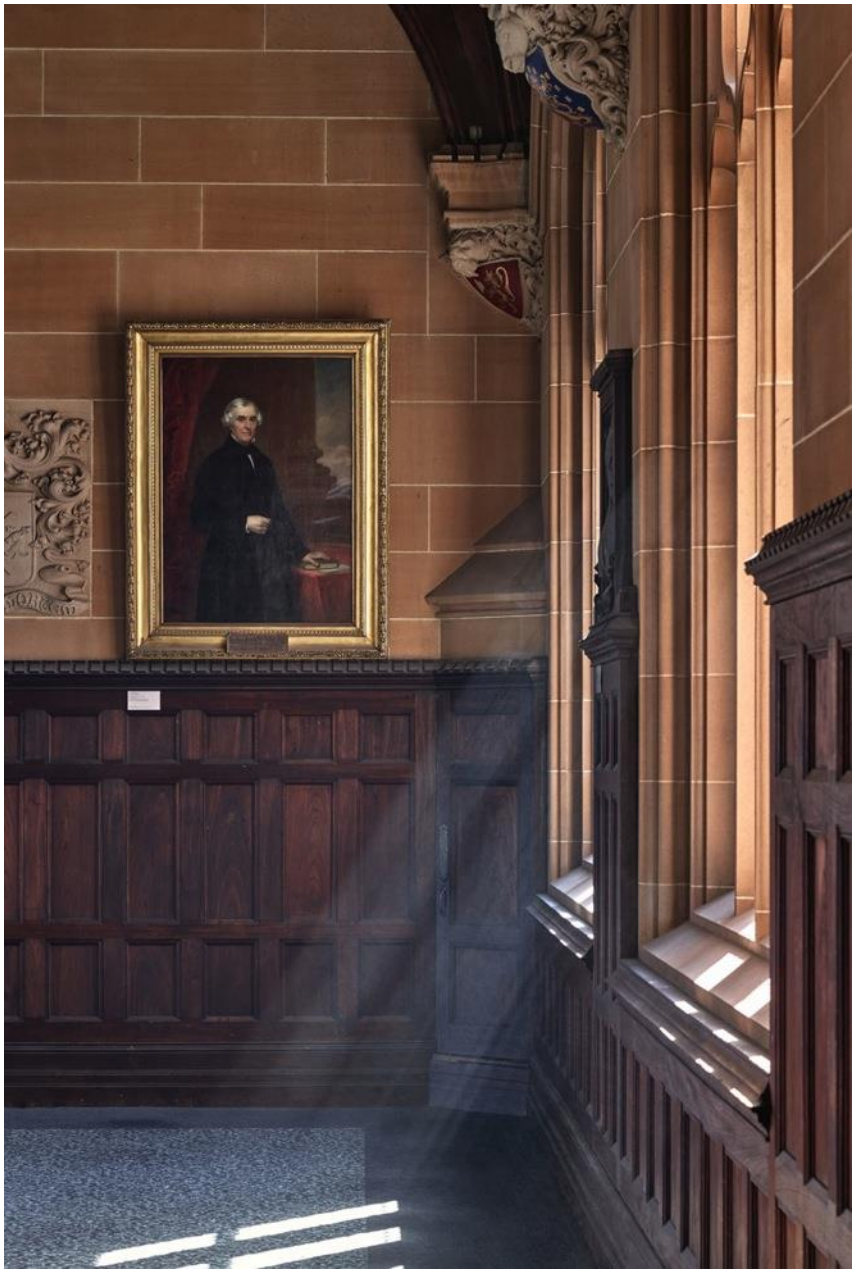


Significant Public Interest (USYD OpenDay)

Thank you!

End

http://www.openvswitch.org//support/ovscon2015/16/1305-lantz.pdf



THE UNIVERSITY OF SYDNEY