Selena: Faithful Reproduction Of Network Experiment

D. Pediaditakis, ***C. Rotsos***, A. W. Moore
Computer Lab, University of Cambridge

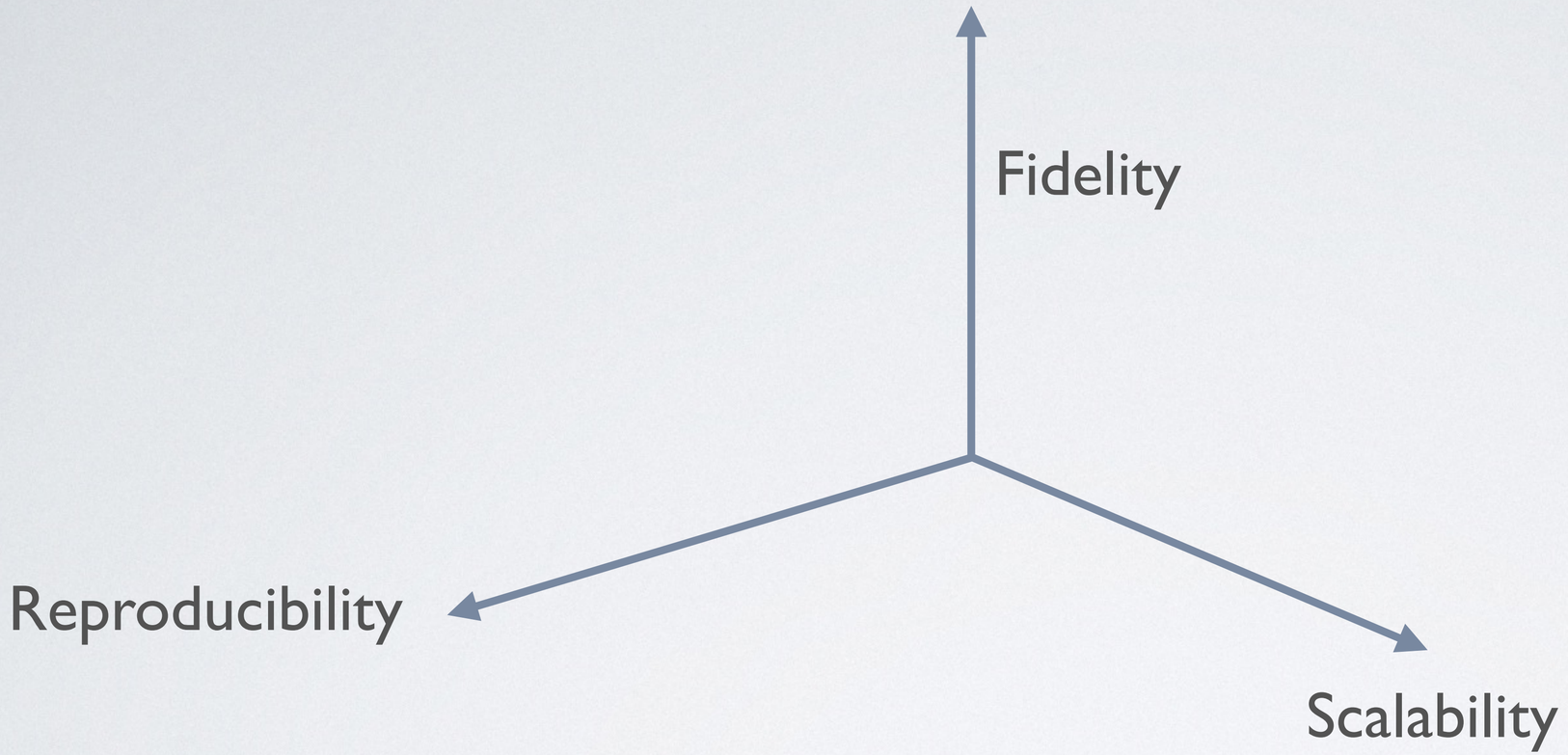
MSN '14, 10/7/14

INTRODUCTION

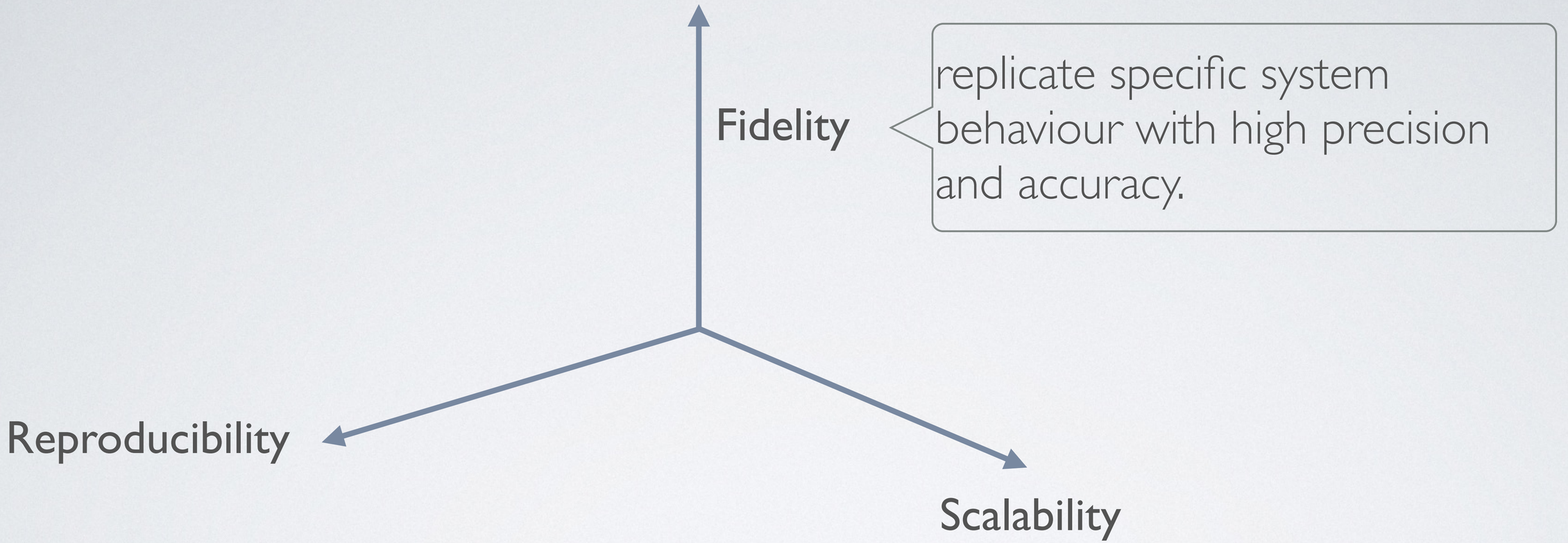
- *How do I evaluate/test my exciting new network architecture/application/protocol?*
  - Network evolution makes the answer **complex**.
  - System sizes make the answer **hard**.

- Current requirements:
  - 10GbE and beyond (100 GbE?) link speeds.
  - Datacenter and Internet host numbers.
  - Close affinity between management and flow timescales.
  - Complex application behaviours.

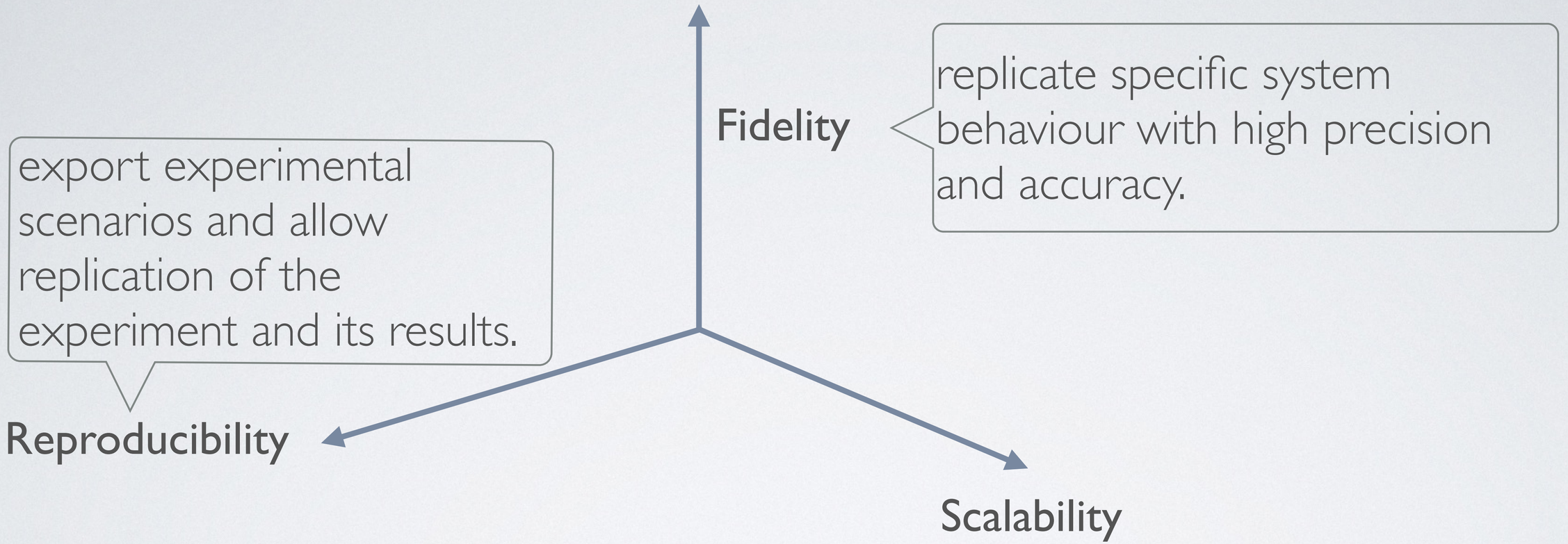
AXIS OF NETWORK EXPERIMENTATION



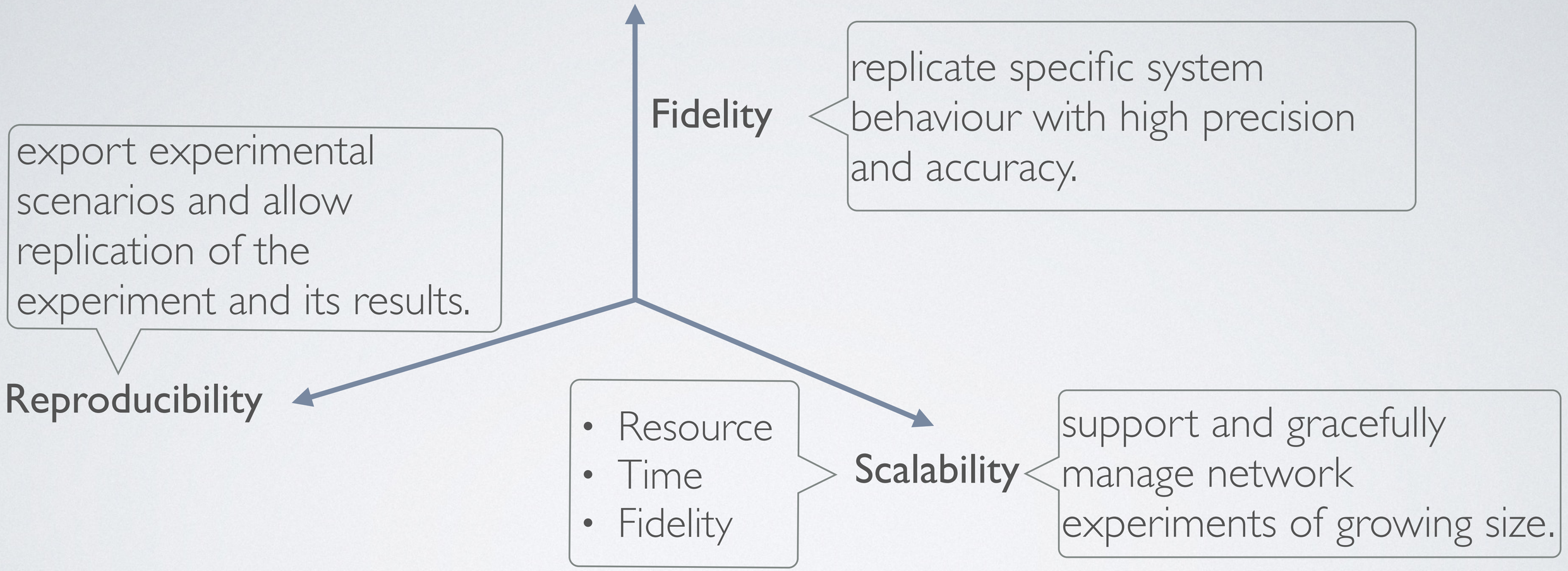
AXIS OF NETWORK EXPERIMENTATION



AXIS OF NETWORK EXPERIMENTATION

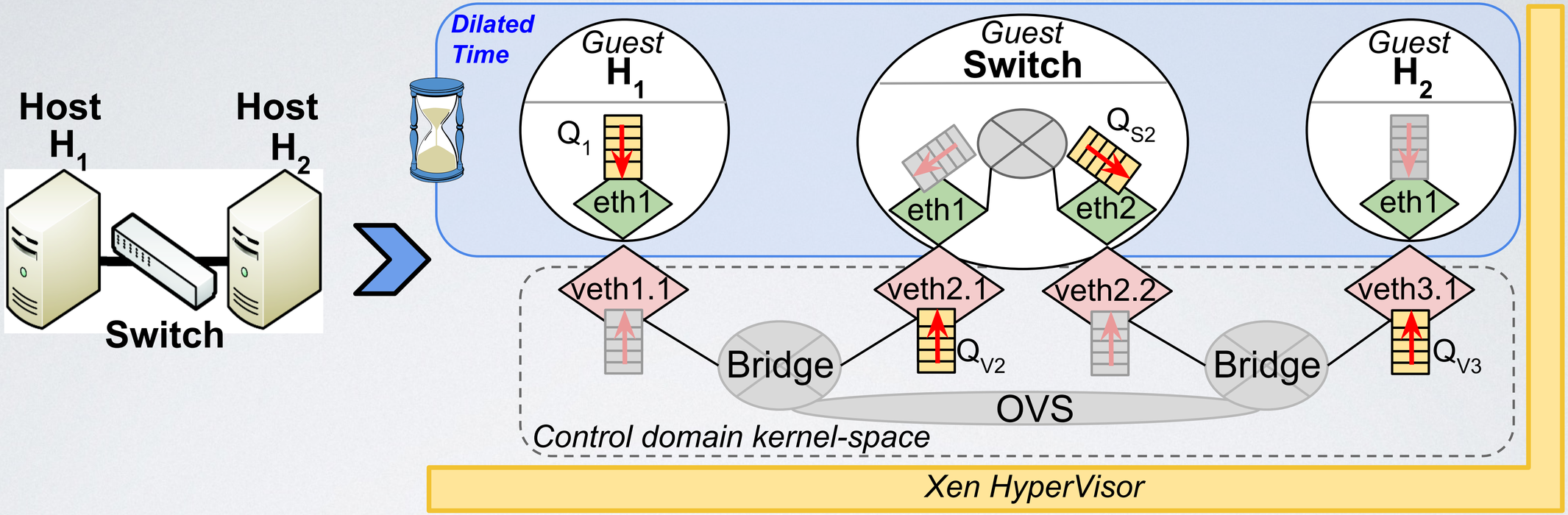


AXIS OF NETWORK EXPERIMENTATION



Scalability dimensions are Pareto complete!

SELENA ARCHITECTURE



Single-host XEN-based emulation framework, *"If it runs on XEN, it runs on SELENA"*.

- *Scalability*: Time dilation for unmodified guests, explicit scalability trade-off control.
- *Fidelity*: Link emulation, OF models.
- *Reproducibility*: XAPI-based Python API, unmodified guest support.

REPRODUCIBILITY

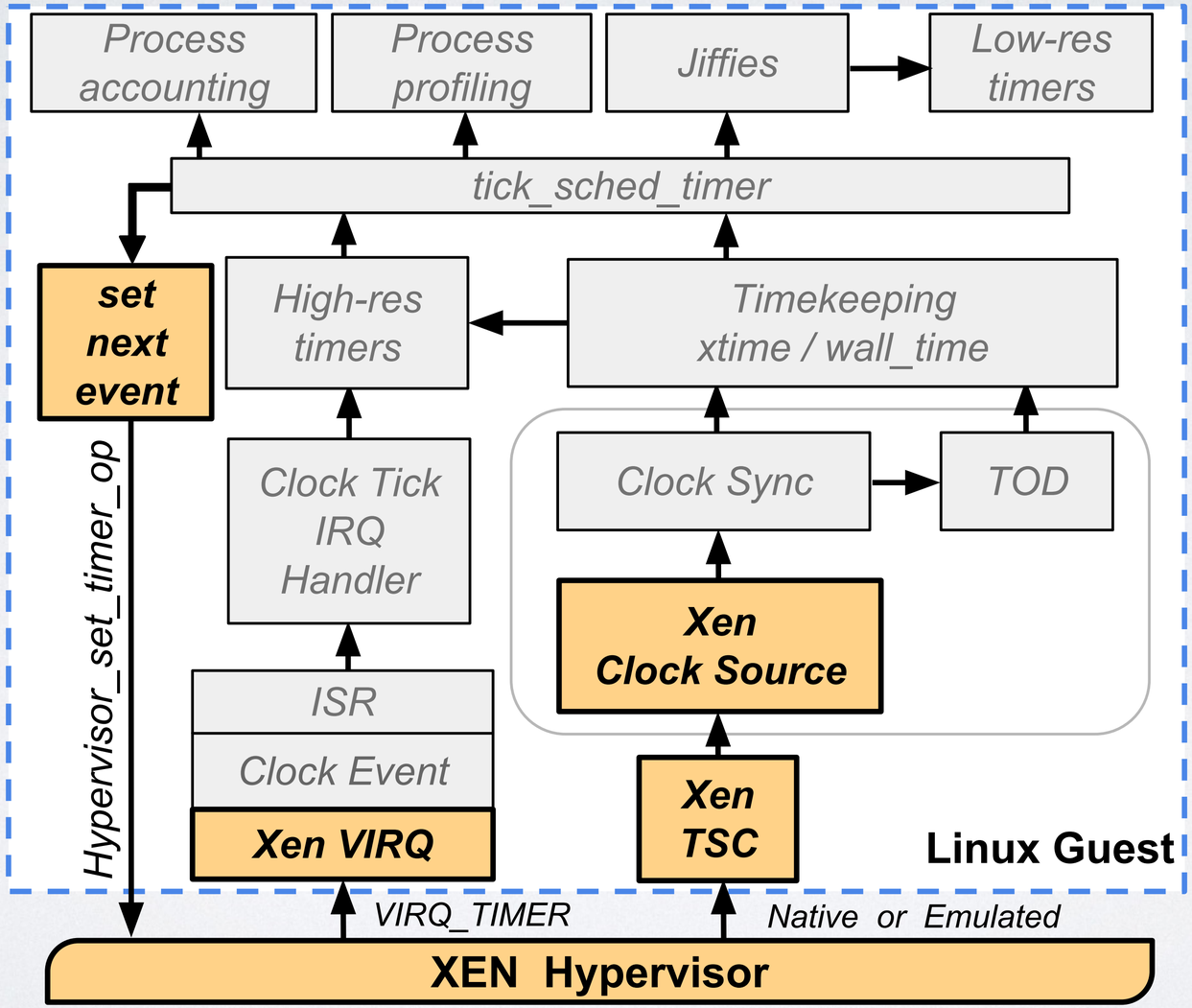
Topology

```
# Node-0 (H1)
newNode(
  0, "H1"                 # unique ID, name label
  NodeType.LINUX_HOST,    # guest template
  [ (1000,                # 1st interface, queue len
    "RANDOM",             # MAC address
    "10.0.1.2",           # IP address
    "255.255.255.0",      # NetMask
    "10.0.1.1"),          # Gateway
    (..), (..) ],         # additional interfaces
  1,                      # number of VCPUs
  "4,5,6,7",              # VCPU Mask
  "512M"  )               # Guest RAM
# Node-1 (H2)
newNode(1, "H2", NodeType.LINUX_HOST, .....)
# Node-2 (Switch)
newNode(2, "Switch", NodeType.LINUX_OVS, .....)
# Link: H1<-->Switch
newLink(
  (0, 0),                 # Node-0, 1st interface
  (2, 0),                 # Node-2, 1st interface
  1000,                   # Link speed in Mbps
  0.2  )                # Latency (NetEm params)
# Link: H2<-->Switch
newLink( (1, 0), (2, 1), 1000, 0.2) )
```

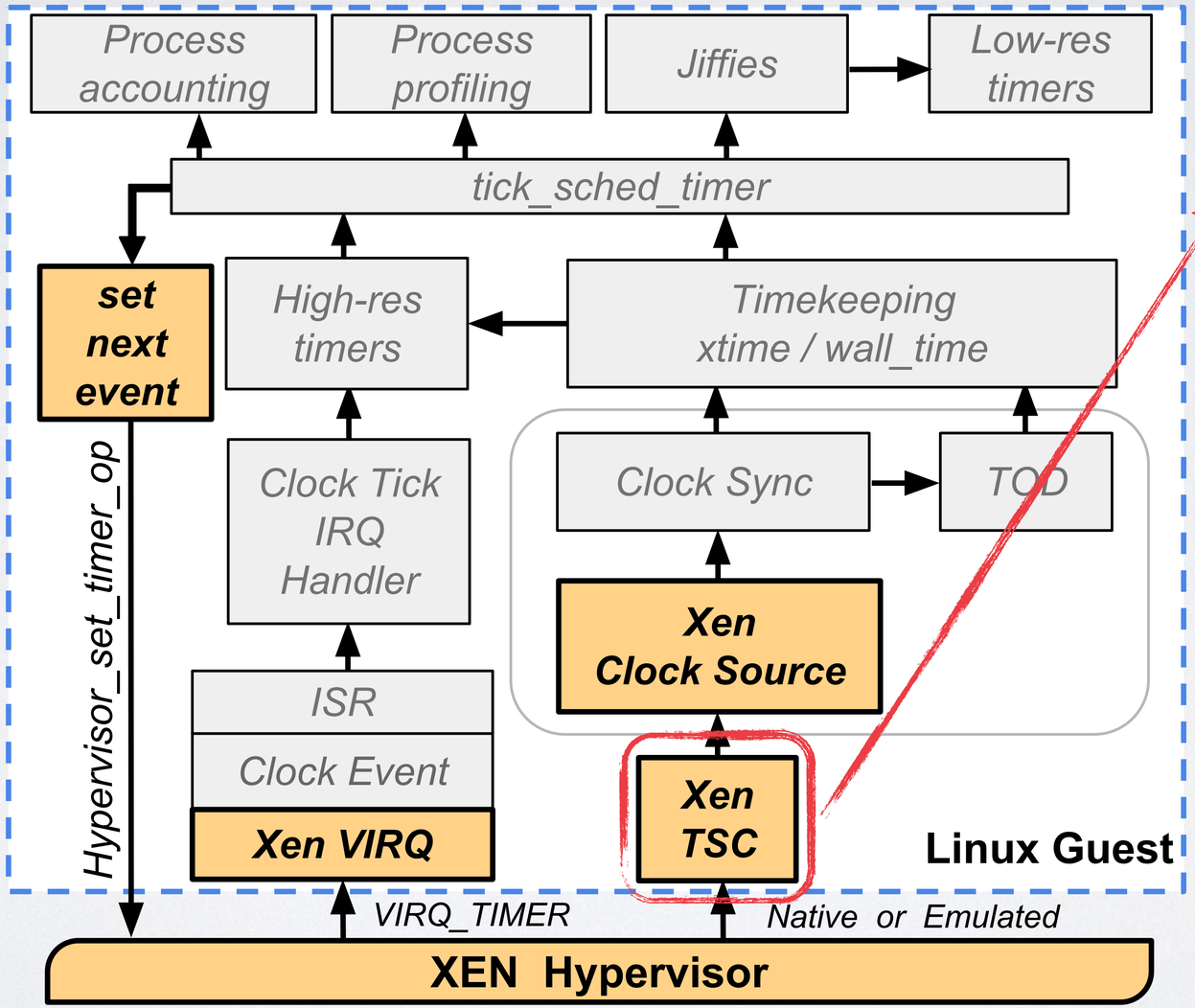
Experiment

```
# Prevent arp broadcasts
setArp(0,"eth1","10.0.1.3","fe:ff:ff:00:01:03"))
setArp(1,"eth1","10.0.1.2","fe:ff:ff:00:01:02"))
# Configure the switch
pushCmd(2,["ovs-vsctl set-fail-mode br0 secure"])
pushCmd(2,["ovs-vsctl add-port br0 eth1"])
pushCmd(2,["ovs-vsctl add-port br0 eth2"])
pushOFRule(2,
  "br0", "add-flow","in_port=1,action=output:2")
pushOFRule(2,
  "br0", "add-flow","in_port=2,action=output:1")
# Run netperf for 10 seconds
pushCmd(0,
  ["netperf -H 10.0.1.3 -t TCP_STREAM -l10 -D1"])
```

TIME DILATION



TIME DILATION

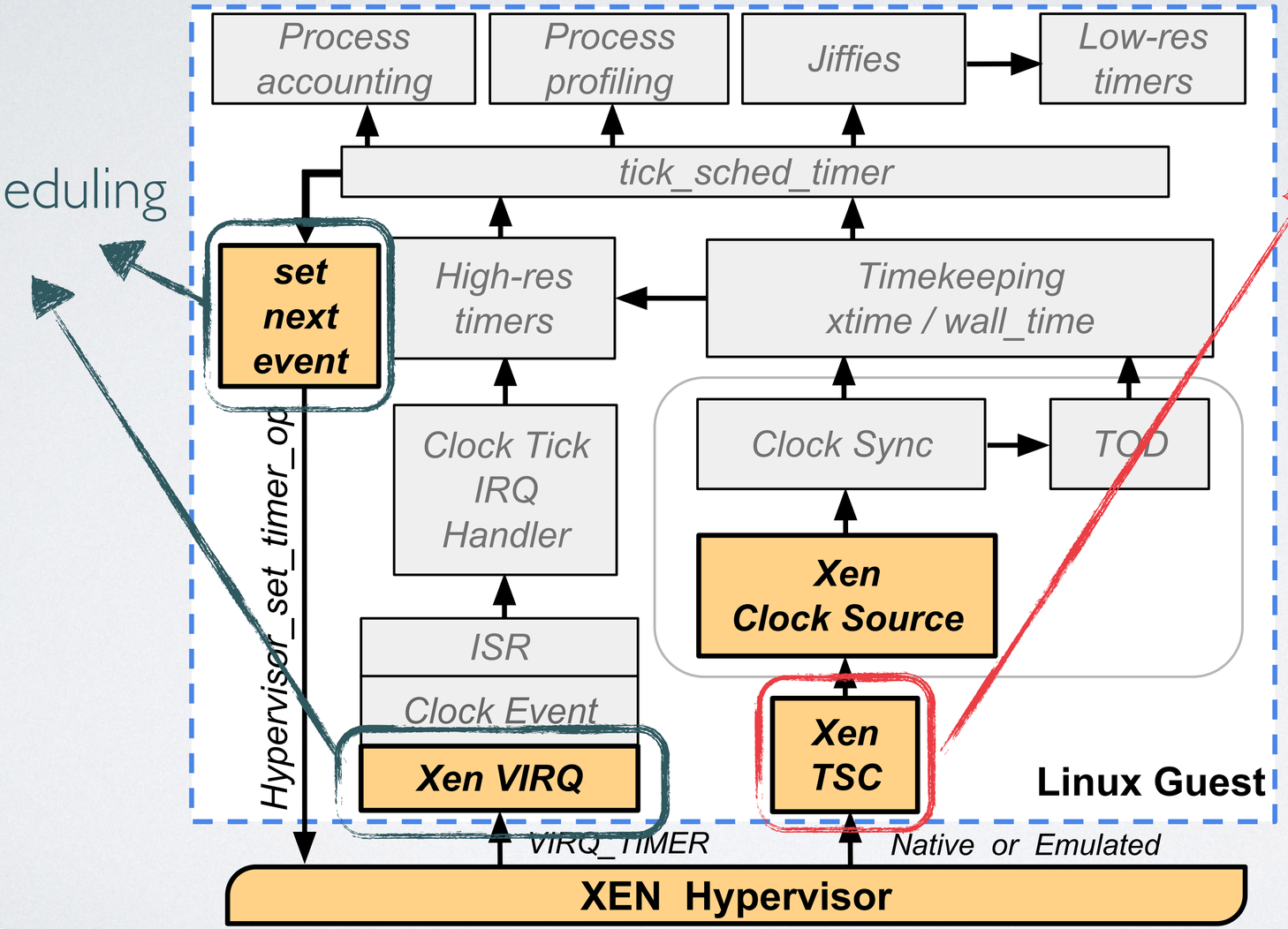


Time source

```
struct shared_info {
     struct pvclock_vcpu_time_info {
          ….
          u64 tsc_timestamp;
          u64 system_time;
          u32 tsc_to_system_mul;
          …
     }
}

or

void pv_soft_rdtsc(struct vcpu *v,
       struct cpu_user_regs *regs,
       int rdtscp);
```

TIME DILATION

Time Event Scheduling



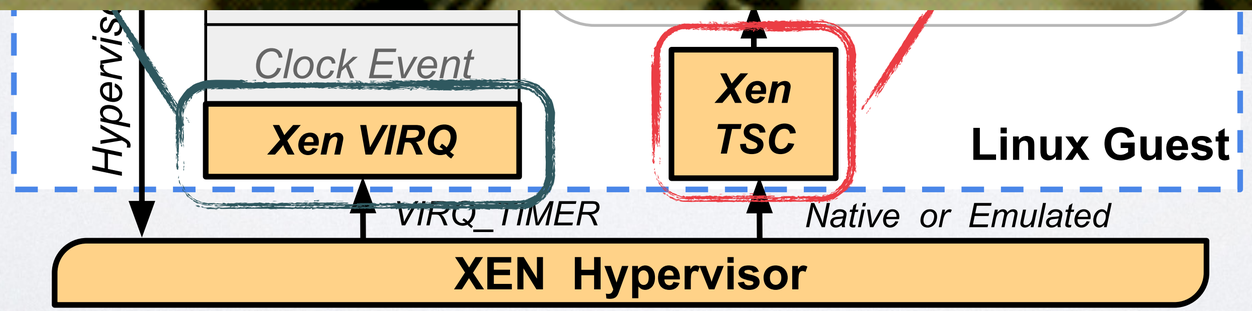
Time source

```
struct shared_info {
     struct pvclock_vcpu_time_info {
          ….
          u64 tsc_timestamp;
          u64 system_time;
          u32 tsc_to_system_mul;
          …
     }
}

or

void pv_soft_rdtsc(struct vcpu *v,
       struct cpu_user_regs *regs,
       int rdtscp);
```

TIME DILATION

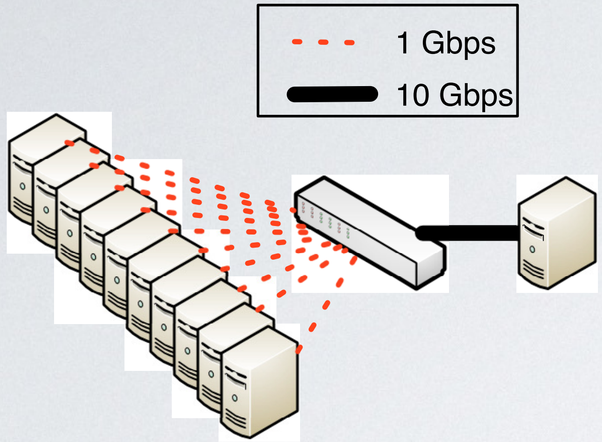
Time Eve

```
_time_info {

amp;
e;
tem_mul;
```

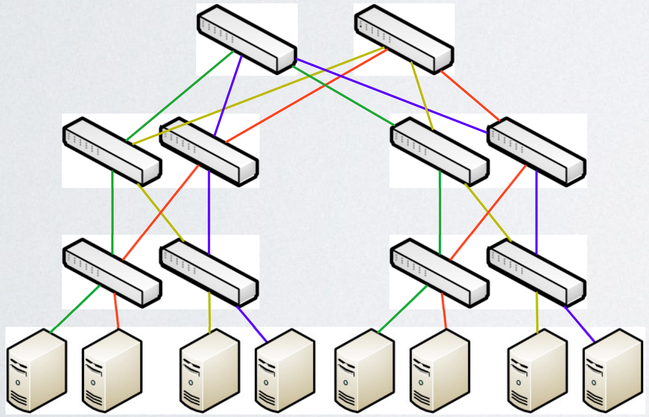



```
void pv_soft_rdtsc(struct vcpu *v,
    struct cpu_user_regs *regs,
    int rdtscp);
```

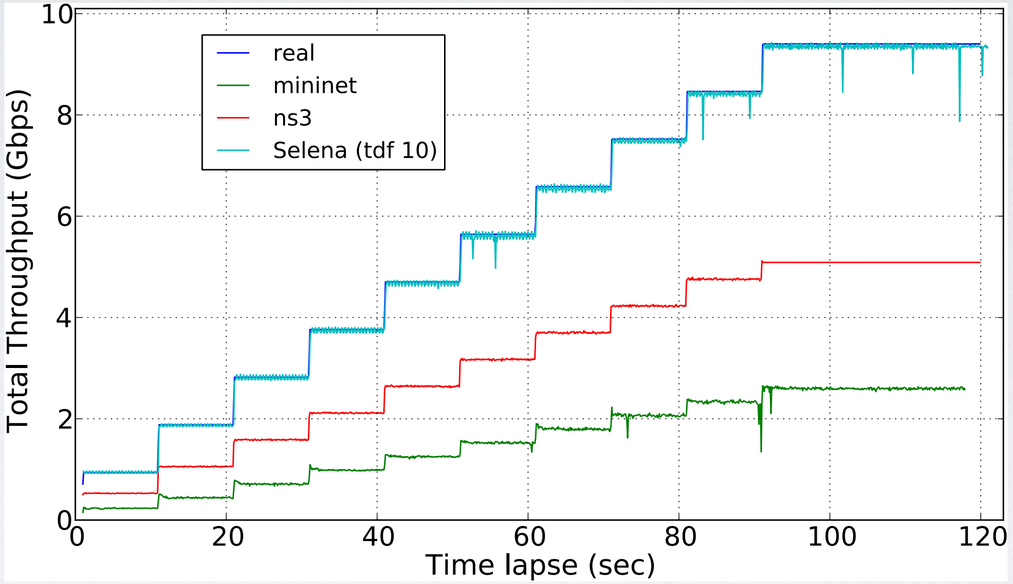
EVALUATION



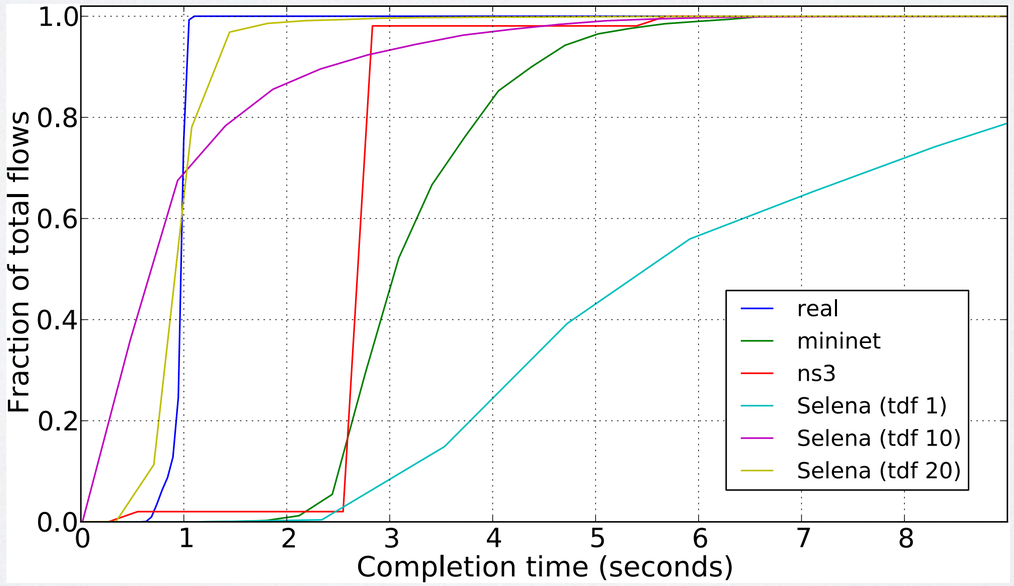
Star Topology



Fat-Tree Topology



Execution time
ns3 : 175m 24s
selena : 20m



Execution time
ns3 : 172m 51s
selena : 40m

SUMMARY

- Network emulation is a multidimensional problem.

- SELENA: scalable emulation platform for large network experiments.
  - Compatible with any cloud application running on PV guests.
  - Experimental reproducibility.
  - Explicit trade-off control between execution time and fidelity.
  - In-depth understanding of TDF and scheduling interplay.
  - Improved results with respect to state of the art.

QUESTIONS?

dimosthenis.pediaditakis@cl.cam.ac.uk
charalampos.rotsos@cl.cam.ac.uk
andrew.moore@cl.cam.ac.uk