A Virtual Computer Networking Lab

Mike Zink, Max Ott, Jeannie Albrecht

GEC 20,

March 24th 2015

Sponsored by the National Science Foundation

The "Traditonal" Networks Lab

- Each institution requires a set of hardware (switches, routers, cables, computers)
- Hardware outdates fairly quickly
- Certain aspects are vendor specific
- Equipment is unused for periods of time

The "Virtual" Networks Lab

- Use GENI infrastructure to teach lab
- Individual institutions don't need hardware
- "Guide" students as much as needed
- Teach new technologies (e.g., OpenFlow)

Downside:

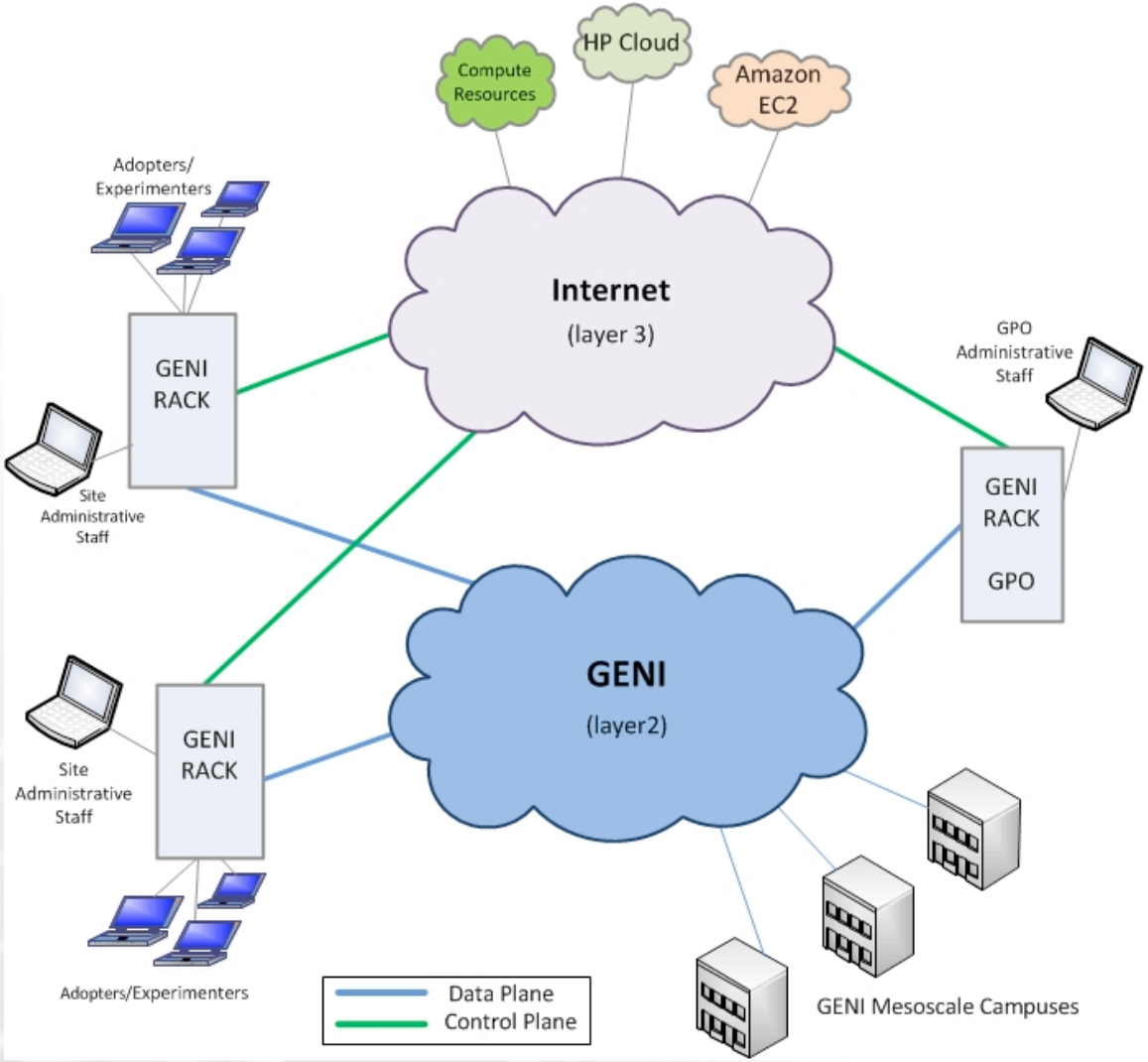
- Students do not touch hardware

Approach:

- Connect local HW to GENI where possible

GENI Technology

- GENI Portal
- GENI APIs
- GENI Racks
- GIMI Tools
- LabWiki



LabWiki as a Classroom Tool

localhost:4000/labwiki

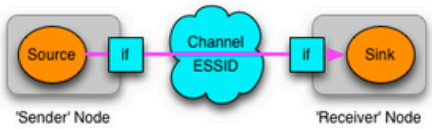
LabWiki

user1 Log out

Plan

Tutorial: First Experiment

As mentioned before, we want to configure an experiment as shown below:



The first step is to describe the experiment in OEDL, the OMF Experiemnt Description Language. To see how this looks for this experiment, open the '1_hello.rb' file in the **Prepare** column.

Ignoring some of the details we can see the definition of two resource groups, **Sender** in line 6 and **Receiver** in line 21.

```
6:   defGroup('Sender', ...
...
21:  defGroup('Receiver', ...
```

There will be more on groups later, but in this

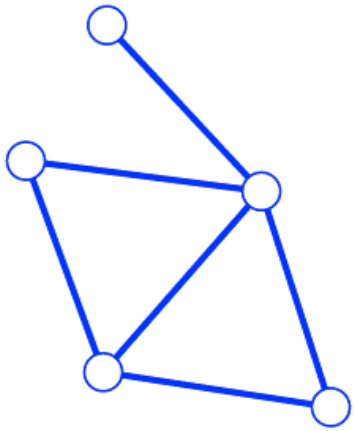
Prepare

git:default:repo/oidl/tutorial/1_hello.r

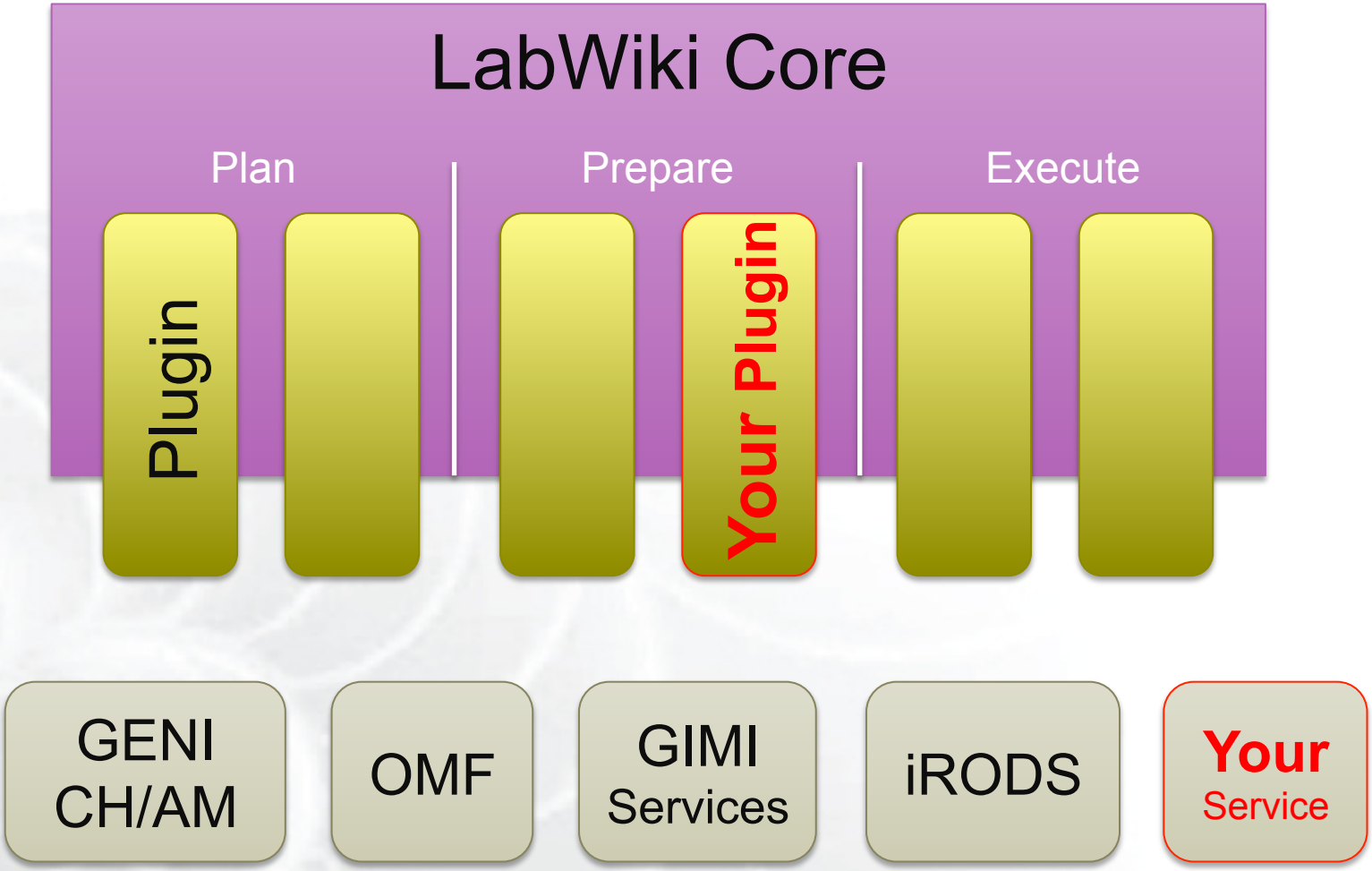
```
1  defProperty('res1', 'omf.nicta.node1', "
2  defProperty('res2', 'omf.nicta.node2', "
3  essid = (0...8).map{65.+(rand(25)).chr}.
4  channel = rand(11)+1
5
6  defGroup('Sender', property.res1) do |no
7    node.addApplication("test:app:otg2") d
8      app.setProperty('udp:local_host', '1
9      app.setProperty('udp:dst_host', '192
10     app.setProperty('udp:dst_port', 3000
11     #app.measure('udp_out', :interval =>
12     app.measure('udp_out', :samples => 1
13   end
14   node.net.w0.mode = "adhoc"
15   node.net.w0.type = 'g'
16   node.net.w0.channel = channel
17   node.net.w0.essid = essid
18   node.net.w0.ip = "192.168.0.2"
19 end
20
21 defGroup('Receiver', property.res2) do |
22   node.addApplication("test:app:otr2") d
23     app.setProperty('udp:local_host', '1
24     app.setProperty('udp:local_port', 30
25     #app.measure('udp_in', :interval =>
```

Execute

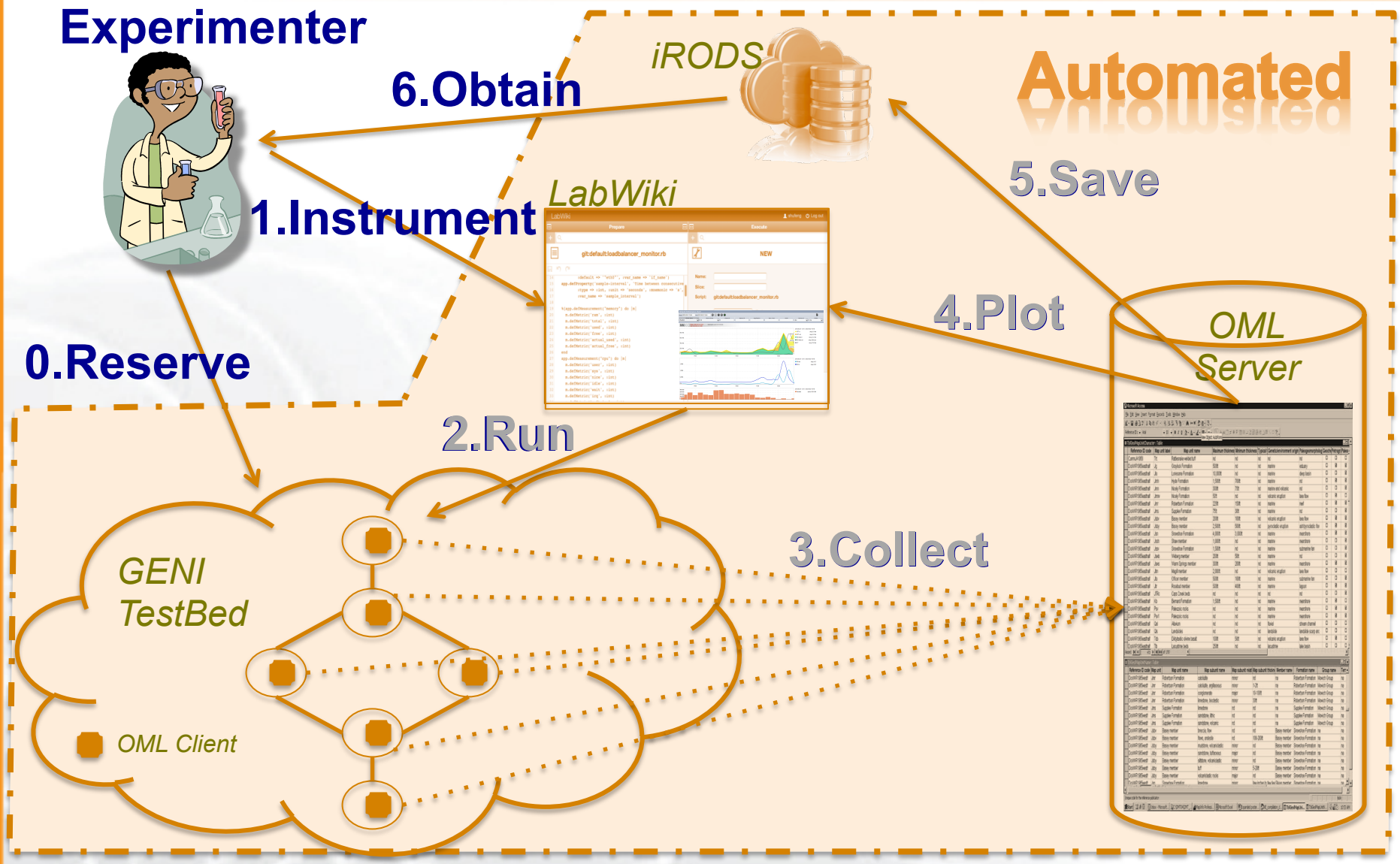
Slice 70256298659140



LabWiki Architecture

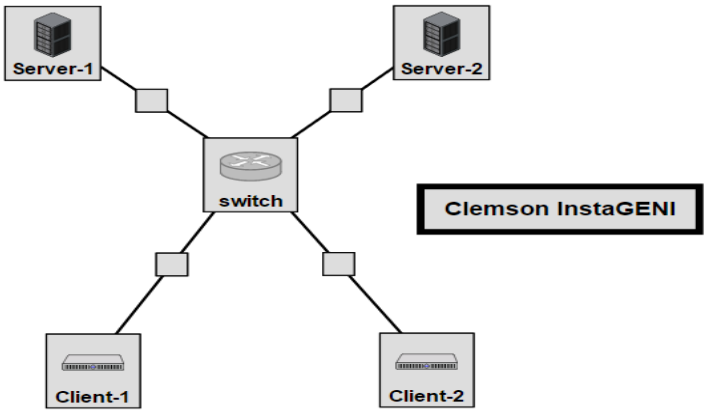


Environment



Assignment I-Data-Center Routing

- Teaches the basics of load balancing for servers in a data-center
- Algorithm can be round robin, throughput-based or random for routing through the OpenFlow switch
- Lessons learned: any-casting, how the packet destination is modified for routing, different algorithms for load balancing, data center technology



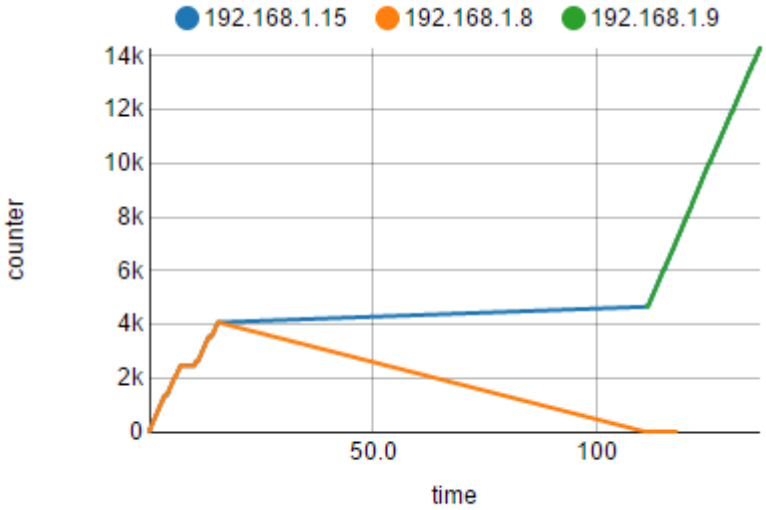
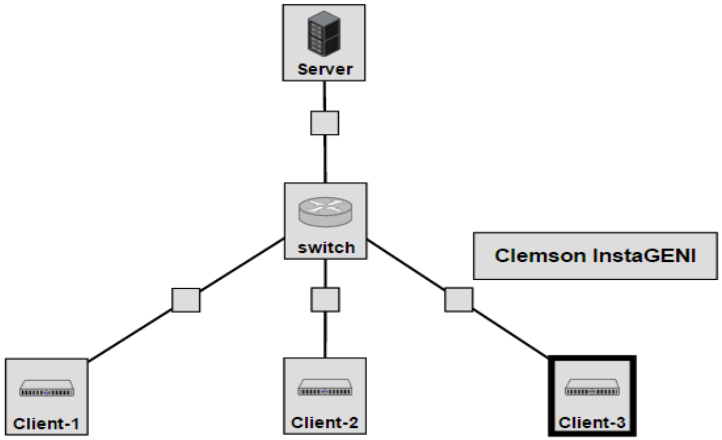
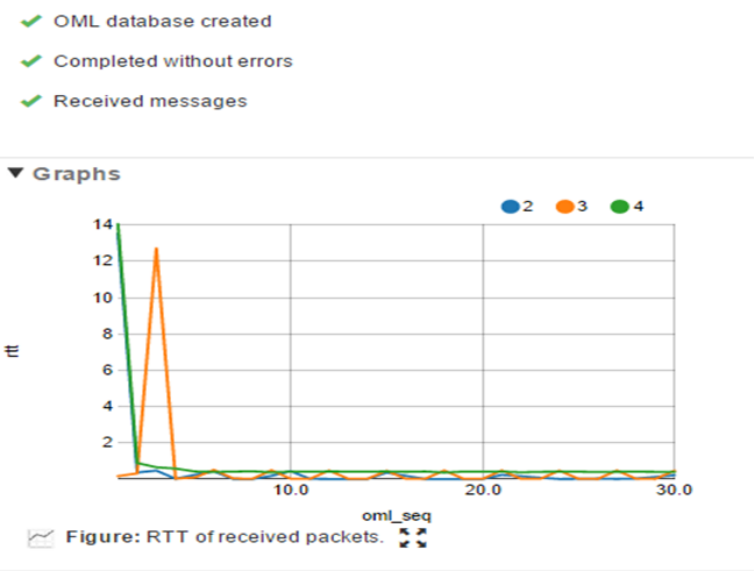


Figure: count of received packets.

Assignment II- Multi-casting

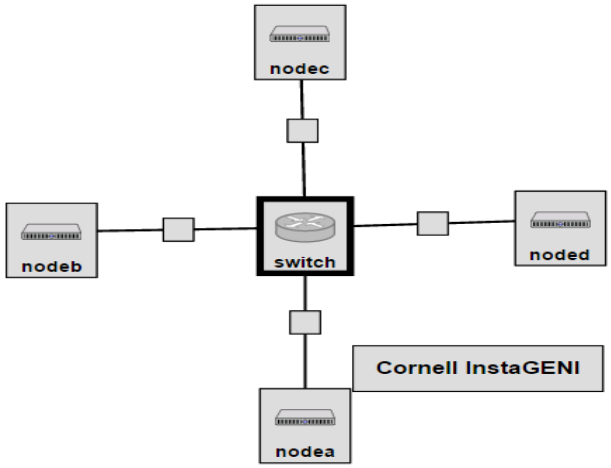
- Teaches the basics of multicasting
- Out-of-band signal - used to send “Join” and “Leave” messages to an OpenFlow controller
- Functionality of the controller is verified through a layer 2 ping
- Lessons learned: multi-casting protocol, packet duplication, layer-2 ping application





Assignment III – Learning Switch

- Teaches the basics of learning switch functionality used by Ethernet switches
- Learning switch implementation through Trema controller
- OEDL script to plot graph
- Lessons learned: learning switch functionality, using different metrics to plot graphs through labWiki



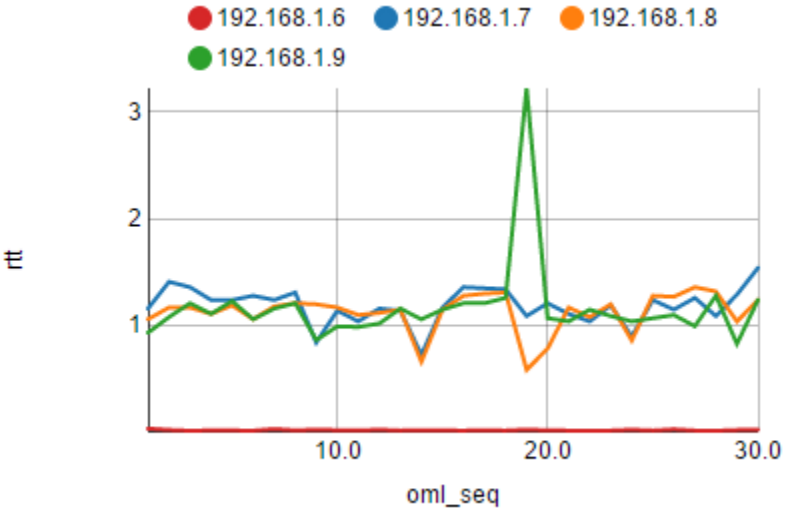
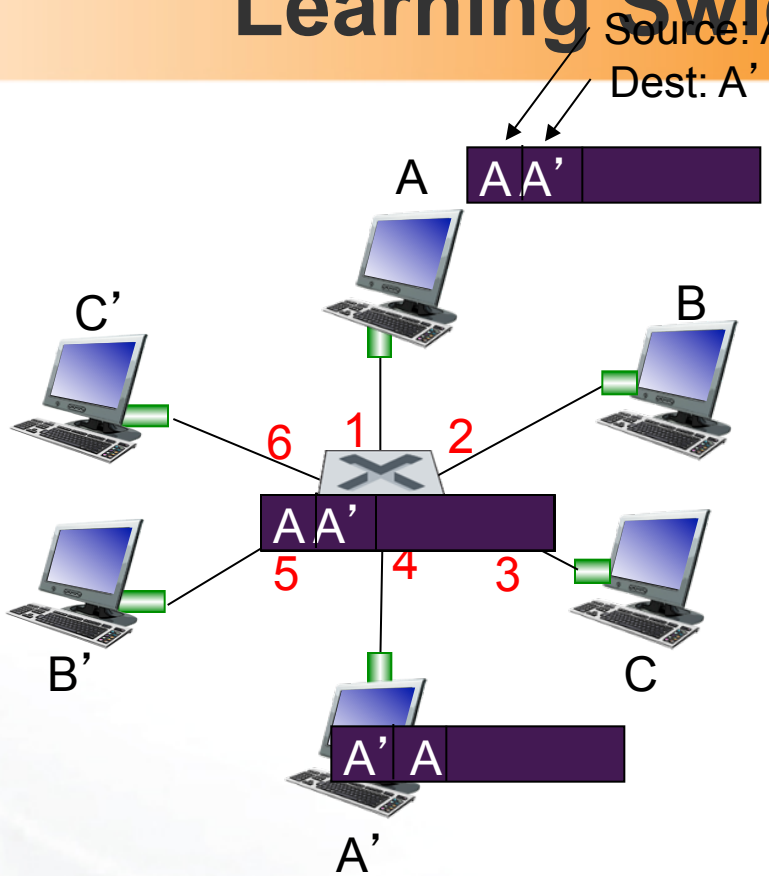
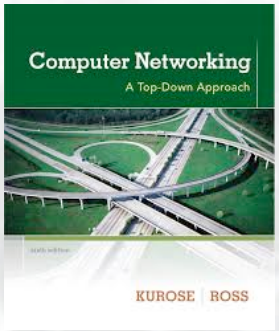


Figure: RTT of received packets.

Learning Switch

- frame destination unknown: *flood*
- destination A location known: selective send
- More info in chapter 5 of "Computer Networks", Kurose & Ross





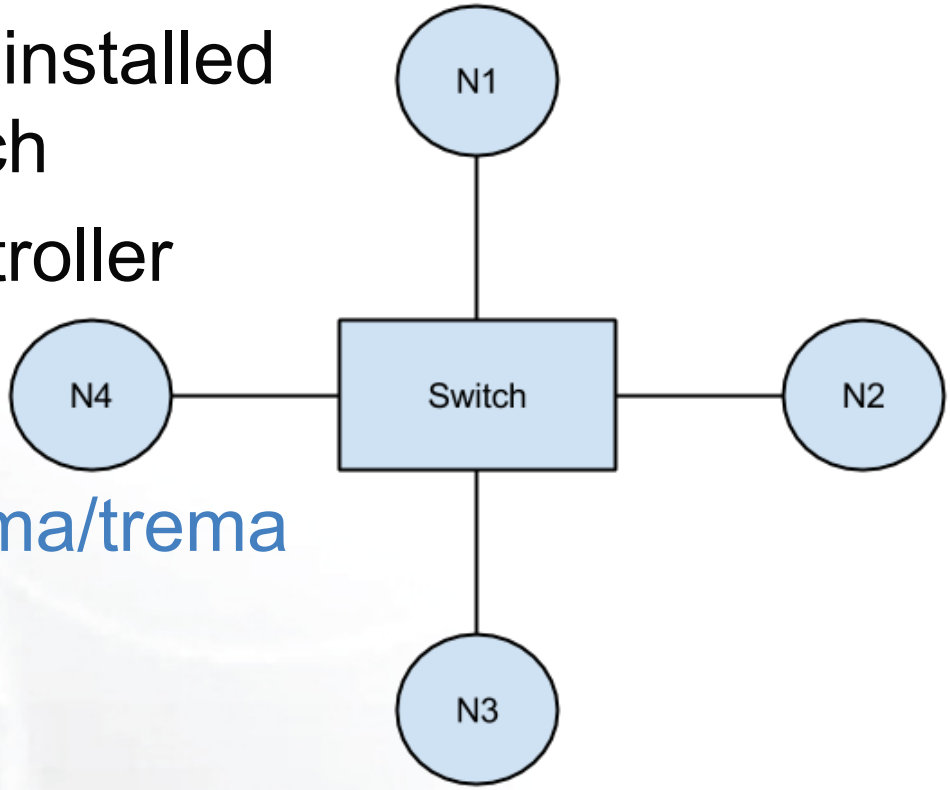
MAC addr	interface	TTL
A	*1*	*60*
A'	*4*	*60*

Switch table (initially empty)

Assignment Topology

- N1 – N4: regular end systems
- Switch: VM with OVS installed
  ➔ sw-based OF switch
- Trema-based OF controller running on Switch

https://github.com/trema/trema



Goal: Implement learning switch and verify its correct behavior

Methodology

- Have students go through material in book
- Provide assignment instructions in LabWiki
- Provide experiment script in LabWiki that allows verification of learning switch
- Provide OF controller template for students to complete

Assignment IV – IP Routing

- Teaches the very basics of IP routing
- Use ping to verify routing
- Can be easily extended:
  - Geographically distributed topology
  - Build routing mechanisms on top
- Lessons learned: static IP routing, forwarding, impact of route on RTT

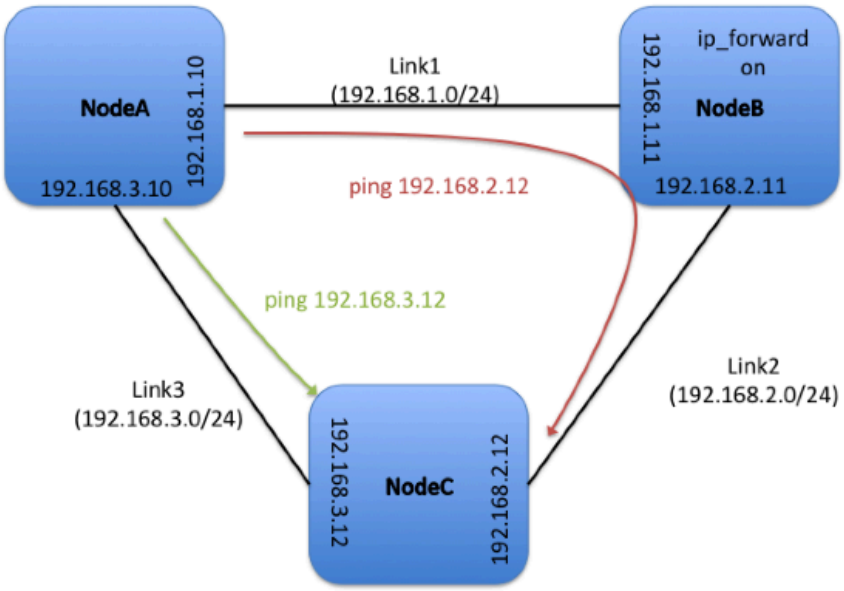
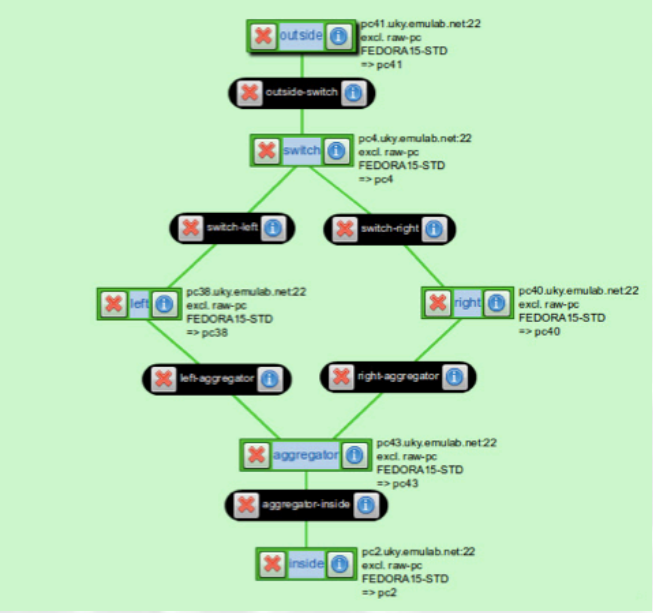


Figure 1 Topology and routing

Assignment V – OF Load Balancer

- Teaches programming of OpenFlow controller
- No OpenFlow knowledge required
- Lessons learned: IP routing and forwarding



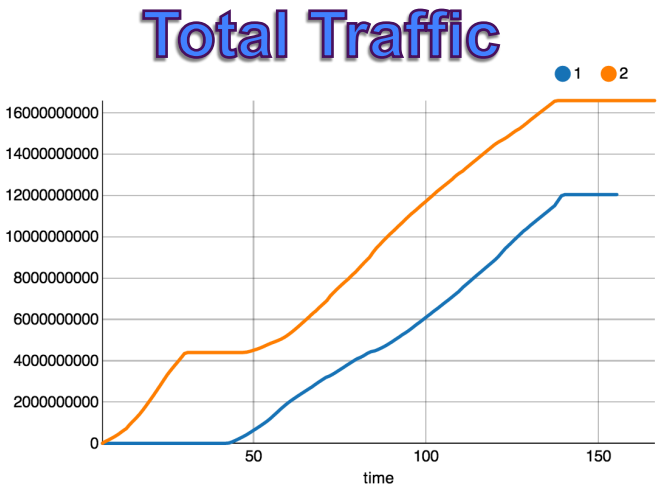


Figure: Cumulated Number of Bytes on each Path

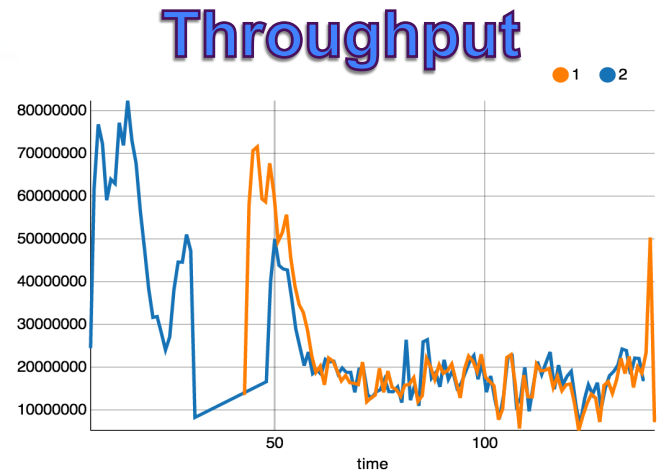
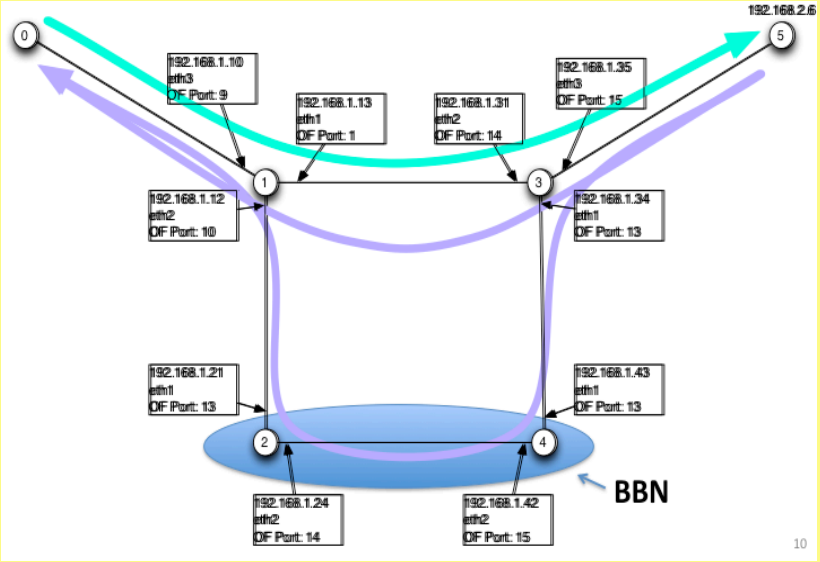


Figure: TCP Throughput (Bytes-per-Second) on each Path

Assignment VI – DASH Video

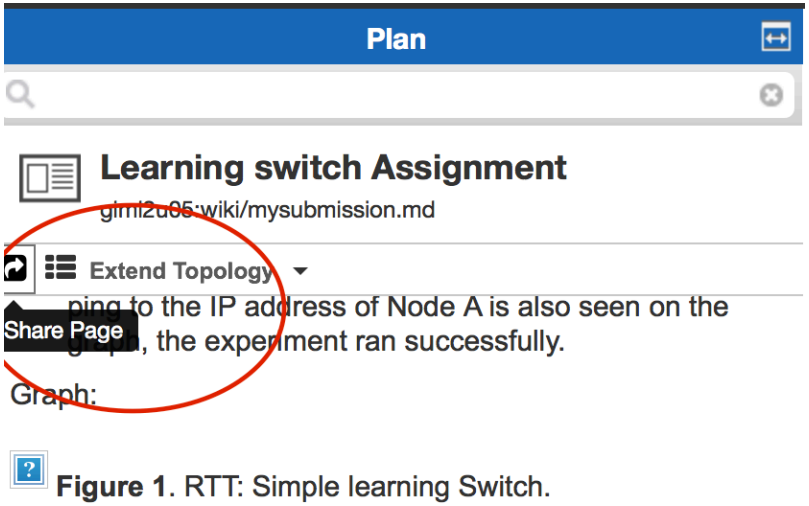
- Teaches how to measure video quality
- Uses alternative paths to show impact of RTT on video quality
- Lessons learned: OpenFlow basics, DASH basics, network measurement

ExoGENI



For Students

- Prepared experiment script (OEDL)
  - Read-only
  - Create copy or run as is



- Assignment
  - Execute experiment
  - Generate results
  - Write comments
  - Submit

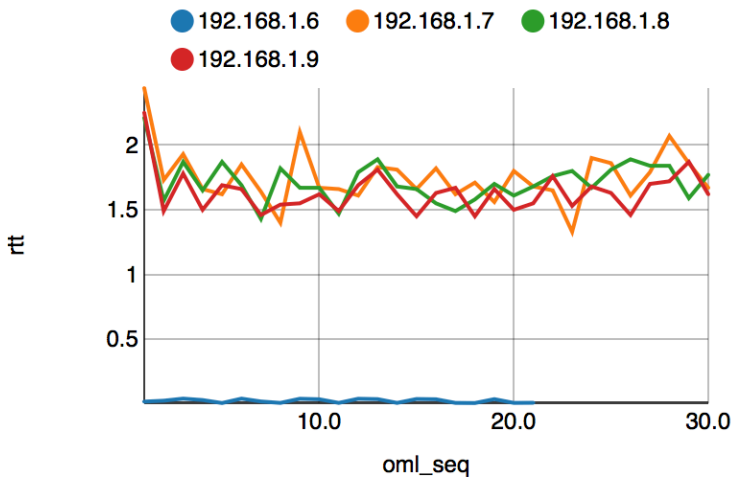
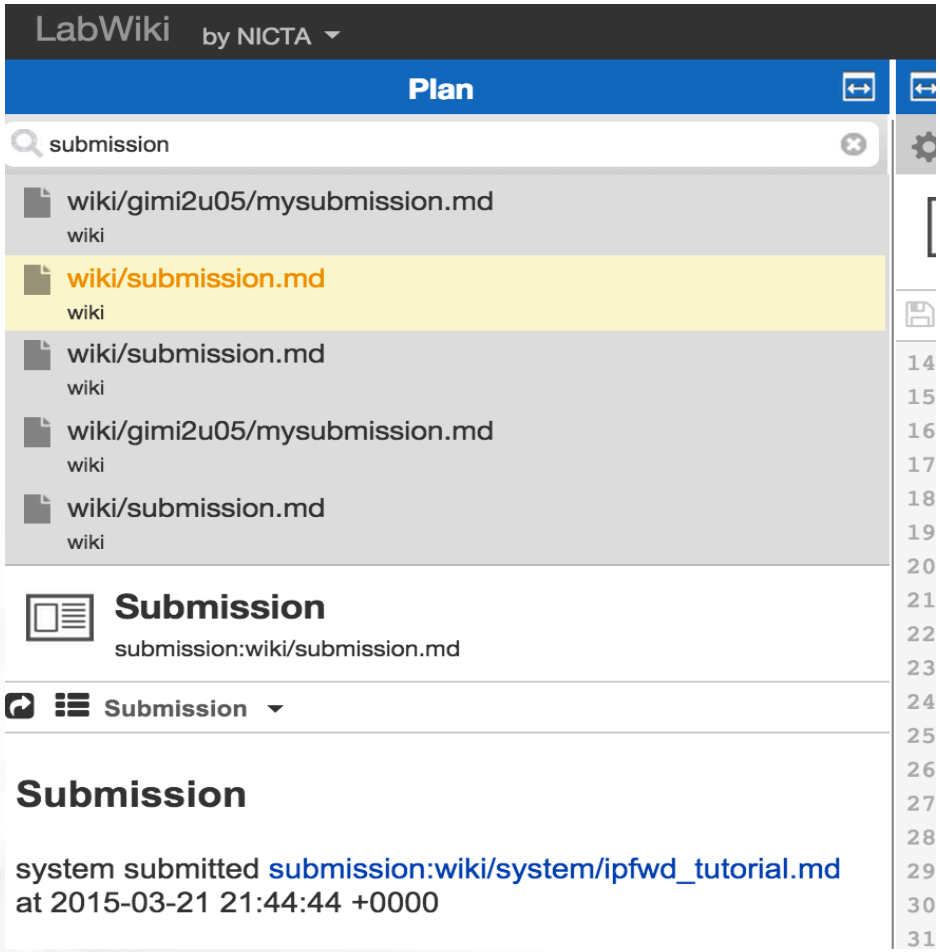


Figure: RTT of received packets.

For Educators

- Create Experiment Script (OEDL)
  - Unlimited update
- Create instructions (Wiki)



For Educators

- Auto checking
  - Custom event trigger
  - View submissions

```
defEvent(:MY_EVENT, every: 0.5) do
# Query for some measurements...
  # returns an array where each element is a hash representing a row from the DB
  query = ms('ping').select { [ :remote] }
  data = defQuery(query)

  triggered = false
  if !data.nil? && !(last_row = data.pop).nil? # Make sure we have some data
    next if peak_list.include?(last_row[:remote]) # Do nothing if we have seen this sample before
    if !peak_list.include?(last_row[:remote])
      peak_list << last_row[:remote] # record that sample, so we dont trigger on it again
    end
    if peak_list.include?('192.168.1.9')&&peak_list.include?('192.168.1.7')&&peak_list.include?('192.168.1.8')

      triggered = true
    end
  end
  triggered
end
onEvent :MY_EVENT do
  group('Source3').startApplications
end
```