TKN SS18 Lab: SDN Applications
Lab report

Ameya Balkrishna Athale

Immatrikulation nummer: 0390059

Advisor – Dr. Hagen Woesner

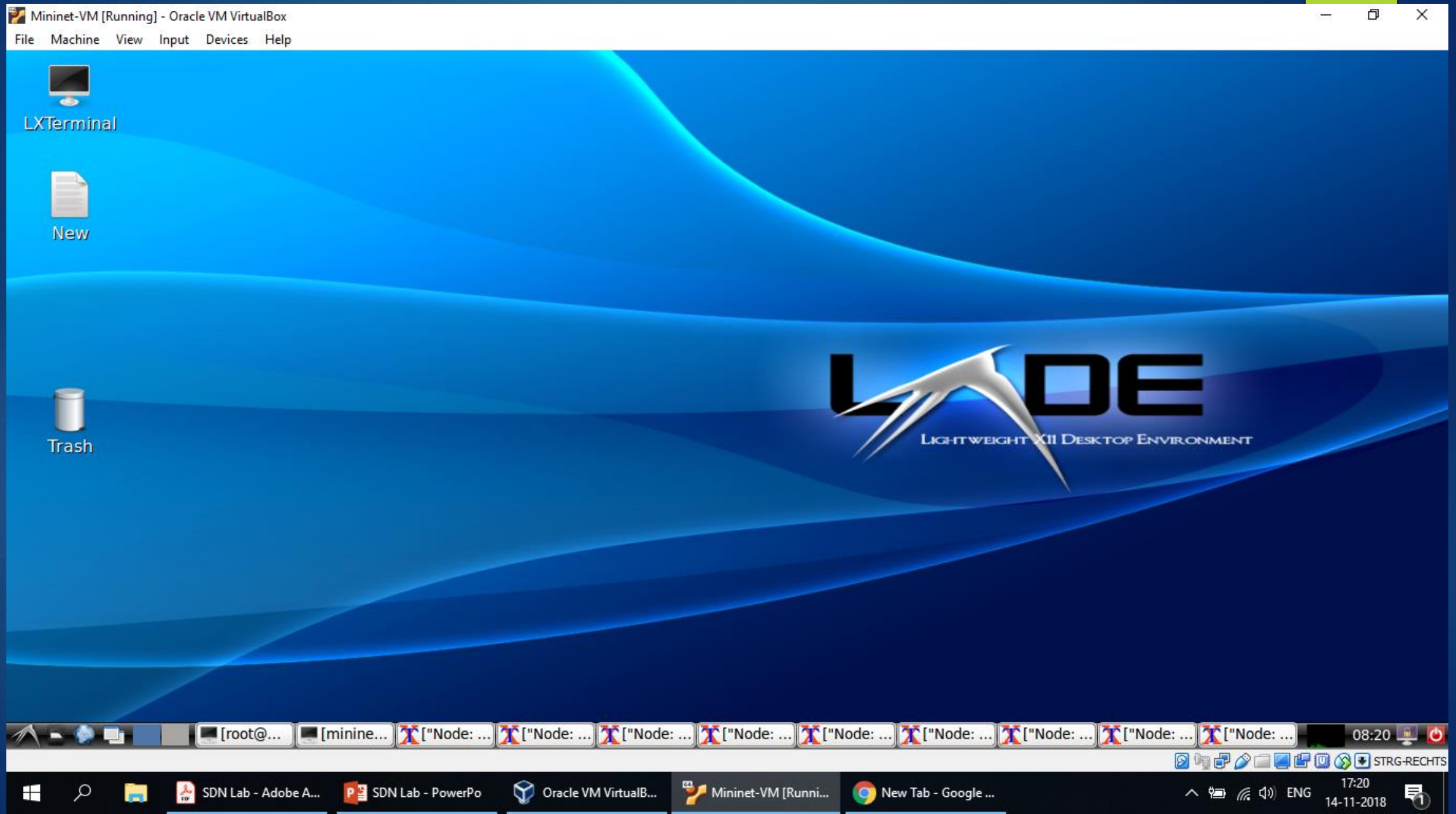
Chapters

- 1. Initial setup
- 2. Some commands
- 3. Executing a topology with 3 hosts, each connected to each of the 3 switches (STP enabled), and iperf readings.
- 4. Ryu Topology Viewer
- 5. Executing a topology with 3 hosts, each connected to the 2 of the 3 switches with link aggregation, and iperf readings

Chapter 1. Initial setup

- Download an install the latest version of Oracle VM.
- Download and import Mininet 2.2.2 image 64 bit in a VM.
- In network settings, enable the second adapter and choose Host-only adapter.
- Perform **‘sudo apt-get update’** and **‘sudo apt-upgrade’,** post first start of mininet.
- Enable DHCP client for eth1 interface.
- A GUI version of mininet environment can be downloaded for working on multiple sessions. To do that use the following command:
  ‘sudo apt-get update && sudo apt-get install xinit lxde virtualbox-guest-dkms’

Mininet GUI environment



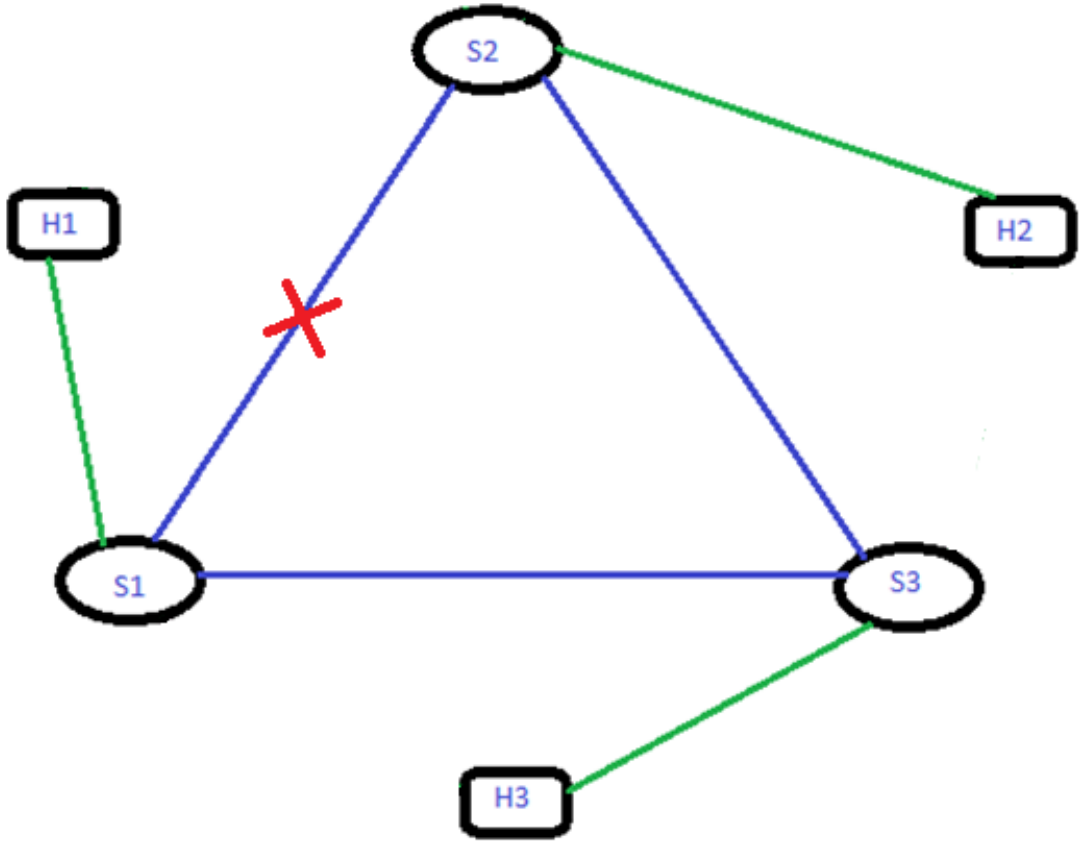
Chapter 2. Some commands

- To initiate mininet with a topology:
  ‘**sudo mn --controller=remote --custom <filename> --topo <topo name>**’
- To specify OpenFlow protocol version:
  ‘sudo ovs-vsctl set bridge <switch_name> protocols=OpenFlow13’
- To enable STP: ‘**ovs-vsctl set bridge <switch_name> stp-enable=true**’
- To run a switch: **‘PYTHONPATH=. ./bin/ryu-manager ryu/app/<filename>.py’**
- IPERF measurement –

Opening a TCP connection: **‘iperf –s’**

Running an iperf – **‘iperf –c <destination IP> -t <t in secs> (optional)**’

Chapter 3



Executing a topology with 3 hosts, each connected to each of the 3 switches, and iperf readings.

Setting up and running the topology

- Make sure eth1 and eth0 interfaces have IP addresses.
- Execute the .py topology file from mininet/custom directory with the following command: '**sudo mn --controller=remote --custom <filename> --topo <topo name>**'
- The topology can be designed in the following way in a python file.

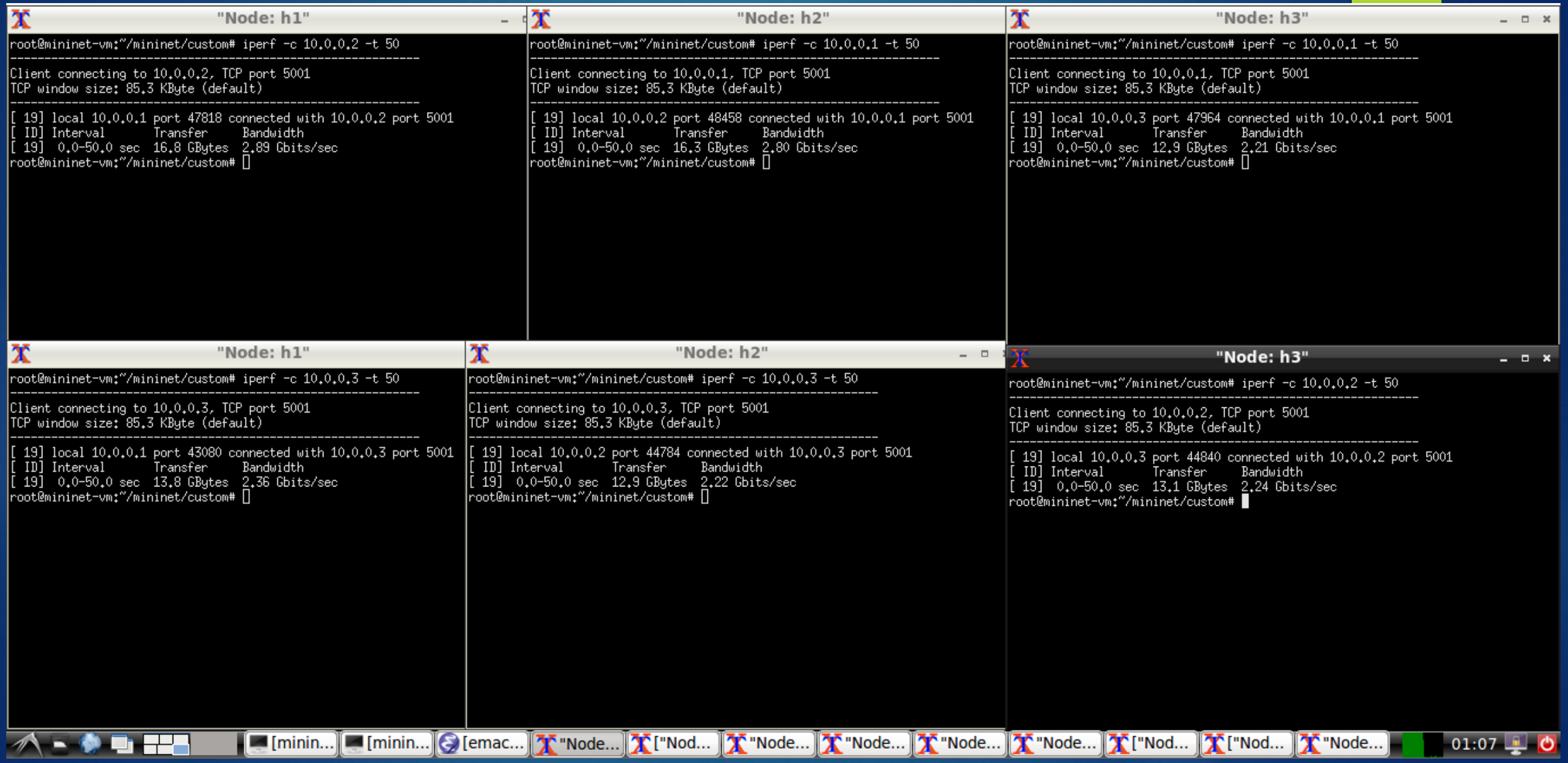
```
# Add hosts and switches
    Host1 = self.addHost( 'h1' )
    Host2 = self.addHost( 'h2' )
    Host3 = self.addHost( 'h3' )
    Switch1 = self.addSwitch( 's1' )
    Switch2 = self.addSwitch( 's2' )
    Switch3 = self.addSwitch( 's3' )

# Add links
    self.addLink( Host1, Switch1 )
    self.addLink( Host2, Switch2 )
    self.addLink( Host3, Switch3 )

    self.addLink( Switch1, Switch2 )
    self.addLink( Switch2, Switch3 )
    self.addLink( Switch3, Switch1 )
```

- When the topology gets live, run a controller that is simple_switch_13.py using the following command:

  PYTHONPATH=. ./bin/ryu-manager ryu/app/<filename>.py

Iperf readings

taken after the topology was running with a controller



Iperf readings (50 secs) in a tabular format

Destination / Source	H1	H2	H3
H1	-	2.89 Gbps	2.36 Gbps
H2	2.80 Gbps	-	2.22 Gbps
H3	2.21 Gbps	2.24 Gbps	-

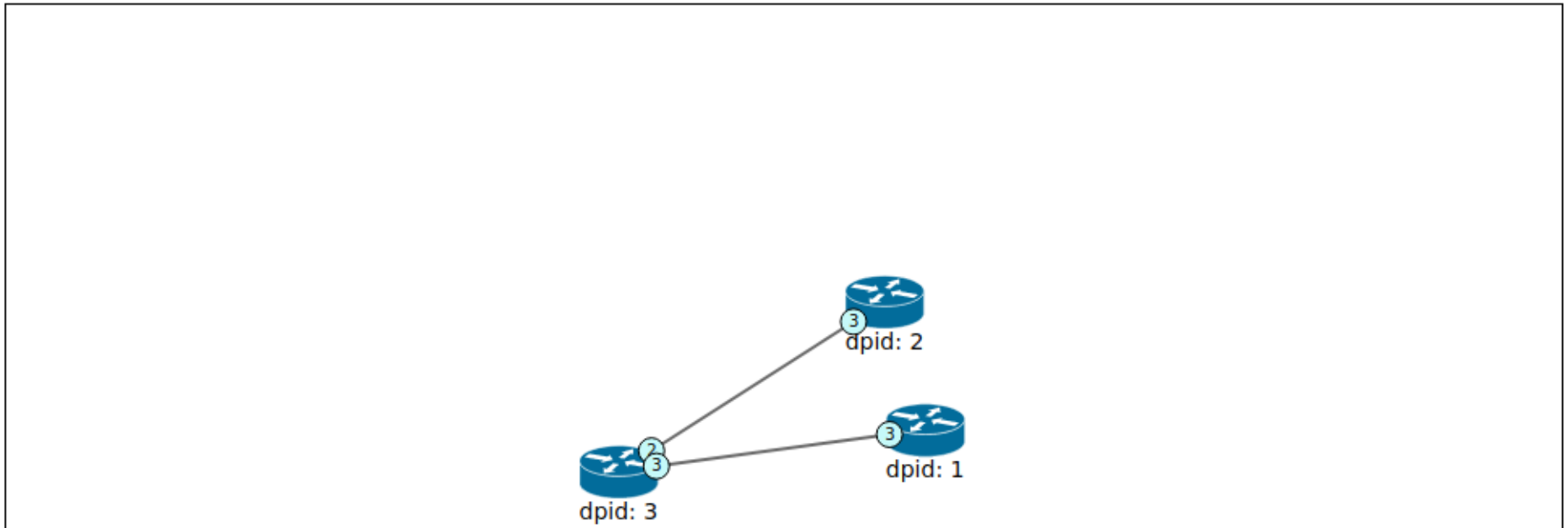
Chapter 4. Ryu Topology Viewer

Ryu topology viewer helps to visualize the topology that is already running.

- Run mininet with any desired topology.
- In another console, navigate to the directory where python files for running switches are located and execute the following command in another window:
  '***ryu run gui_topology/gui_topology.py ´<filename>.py --observe-links***'
- Make sure there are no errors present post execution.
- Go to a browser and type in the eth0 IP address of mininet followed by colon and the port number 8080.
- E.g. 192.168.1.1:8080

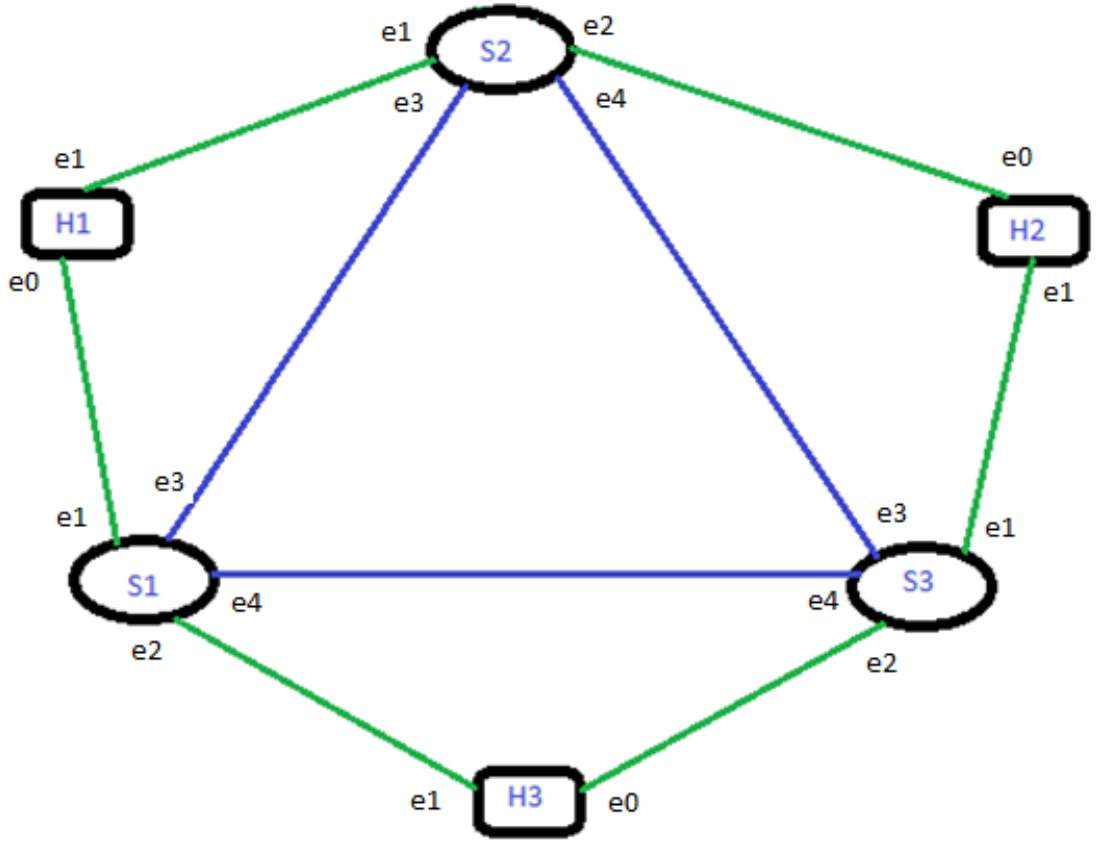
192.168.137.219:8080

Ryu Topology Viewer



mininet@mi... [mininet@mi... [emacs24@... ["Node: h1"] ["Node: h3"] ["Node: h2"]

Chapter 5



Executing a topology with 3 hosts, each connected to 2 of the 3 switches.

Requirements:
- To have equal throughput across all the links.
- To use all the links.
- As much as it gets.

Setting up and running the topology

- Follow the initial steps from the 3rd chapter.
- The topology can be designed in the following way in a python file which will create 3 switches and 3 hosts, each host connected to 2 switches

```
# Add hosts and switches
    Host1 = self.addHost( 'h1' )
    Host2 = self.addHost( 'h2' )
    Host3 = self.addHost( 'h3' )
    Switch1 = self.addSwitch( 's1' )
    Switch2 = self.addSwitch( 's2' )
    Switch3 = self.addSwitch( 's3' )
# Add links
    self.addLink( Host1, Switch1 )
    self.addLink( Host1, Switch2 )
    self.addLink( Host2, Switch2 )
    self.addLink( Host2, Switch3 )
    self.addLink( Host3, Switch3 )
    self.addLink( Host3, Switch1 )
    self.addLink( Switch1, Switch2 )
    self.addLink( Switch2, Switch3 )
    self.addLink( Switch3, Switch1 )
```

Accomplishing the requirements.

- Here, the idea is to divide the topology in 3 different network.
- When the network goes live after executing the topology and starting up the controller **(simple_switch_13.py)**, the following things need to be configured on each host.
- The directly connected destination host's interface must be in the same network of the source host's interface. So, in the topology H1's 'e1' interface is connected to H2's 'e0' interface via switch.

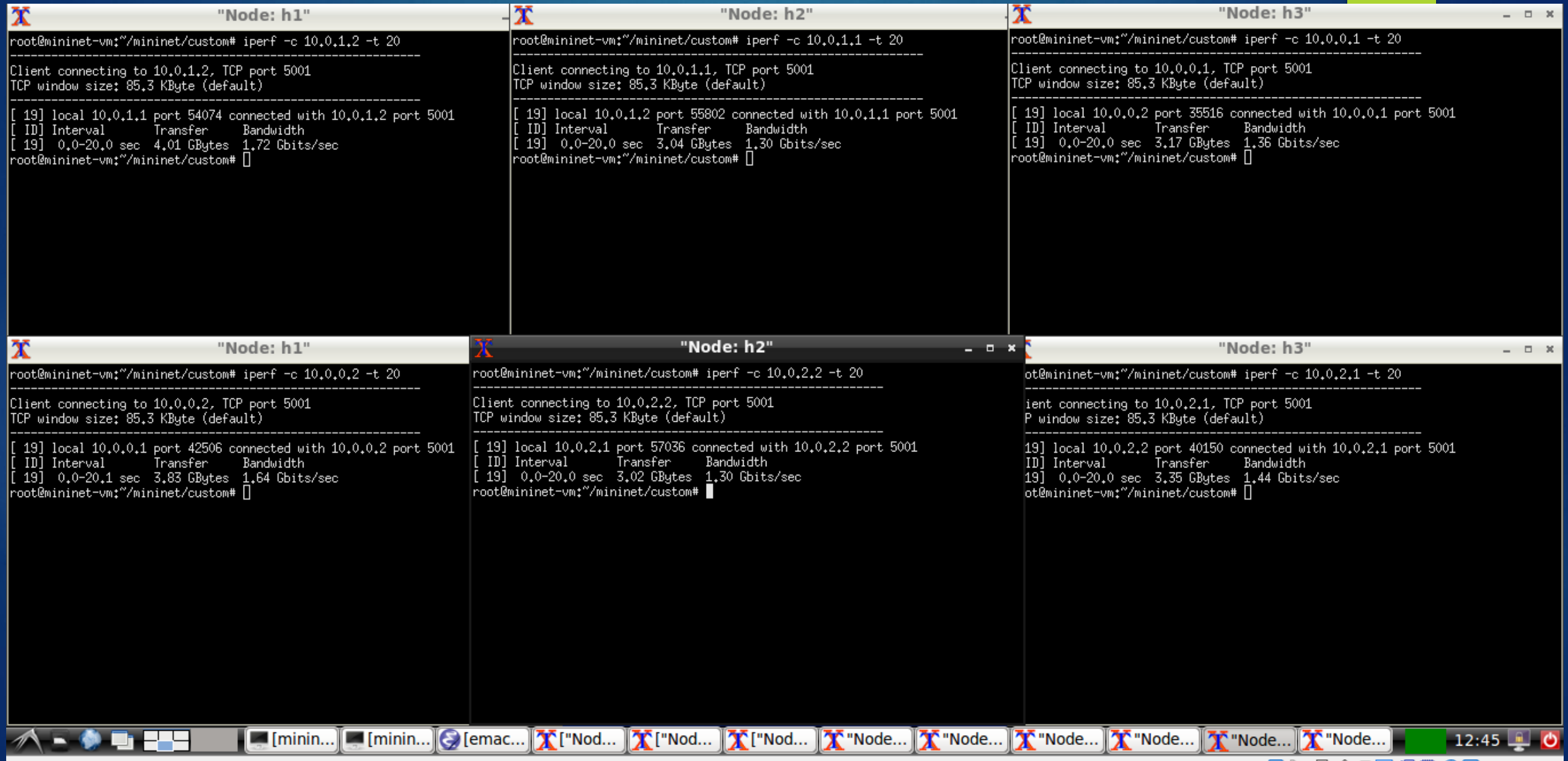
Likewise,

Connections	Network
H1 e0 <-> H3 e1	10.0.0.0/29
H1 e1 <-> H2 e0	10.0.1.0/29
H2 e1 <-> H3 e0	10.0.2.0/29

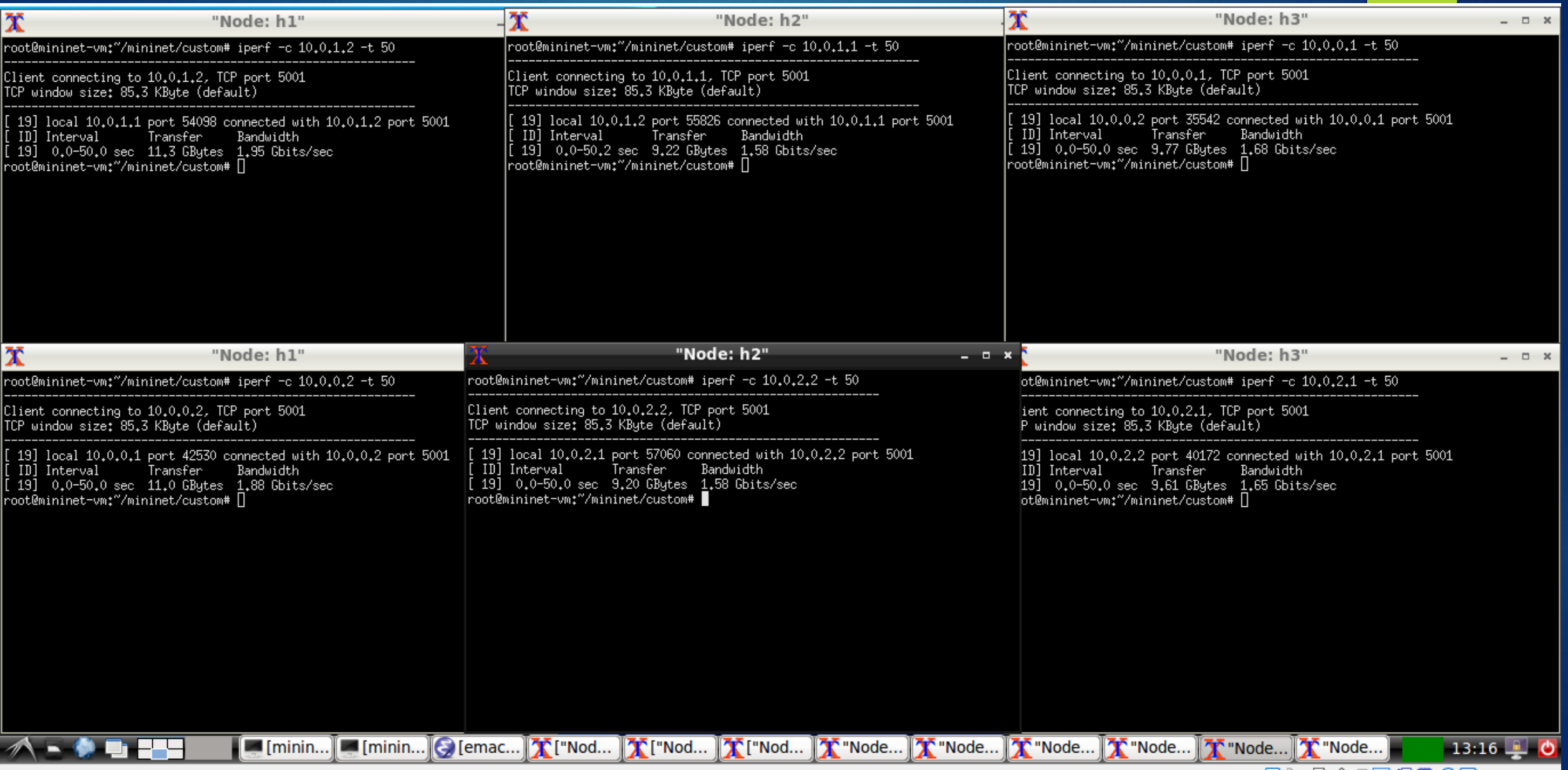
- This way all the links will be used with equal throughput and every host will be reachable from every other host.
- Configuration on each host:

H1	H2	H3
Ip addr add 10.0.0.1 dev h1-eth0	Ip addr add 10.0.1.2 dev h1-eth0	Ip addr add 10.0.2.2 dev h1-eth0
Ip addr add 10.0.1.1 dev h1-eth1	Ip addr add 10.0.2.1 dev h1-eth1	Ip addr add 10.0.0.2 dev h1-eth1

Iperf readings 20 secs



Iperf readings 50 secs

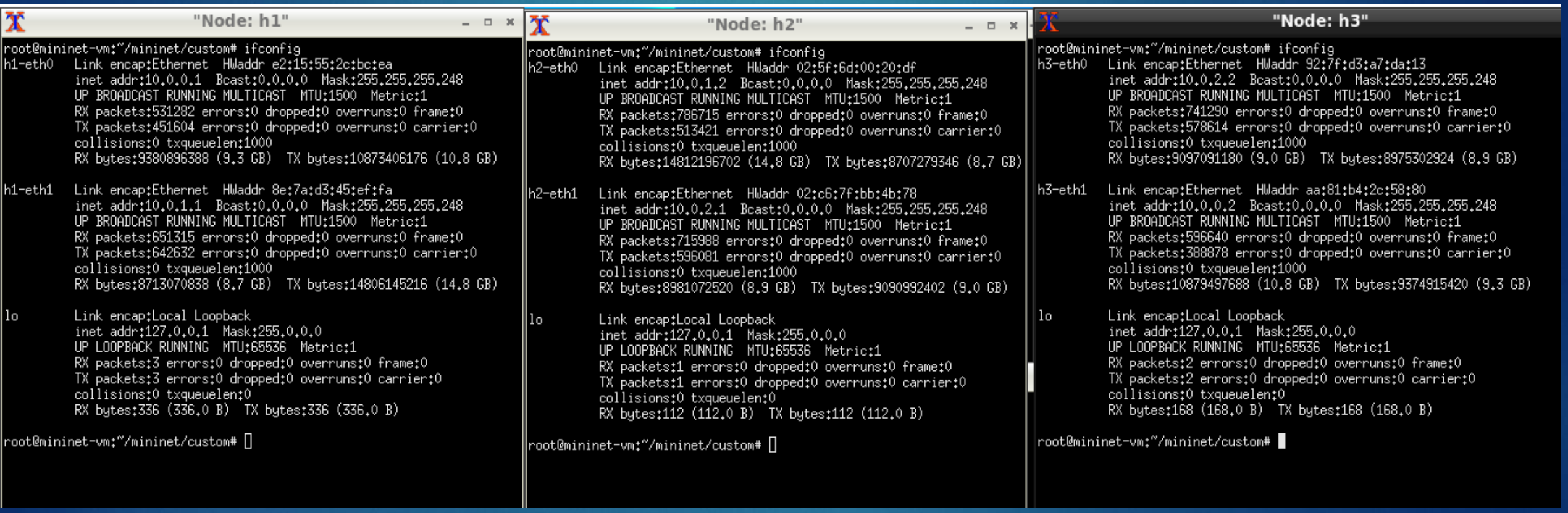


Iperf readings in a tabular format

Source \ Destination		H1		H2		H3	
		20 secs	**50 secs**	**20 secs**	**50 secs**	**20 secs**	**50 secs**
H1	**Bandwidth**			1.72 Gbps	1.95 Gbps	1.64 Gbps	1.88 Gbps
	Transfer			4.01 GBs	11.3 GBs	3.83 GBs	11.0 GBs
H2	**Bandwidth**	1.30 Gbps	1.58 Gbps			1.30 Gbps	1.58 Gbps
	Transfer	3.04 GBs	9.22 GBs			3.02 GBs	9.20 GBs
H3	**Bandwidth**	1.36 Gbps	1.68 Gbps	1.44 Gbps	1.65 Gbps		
	Transfer	3.17 GBs	9.77 GBs	3.35 GBs	9.61 GBs		

Ifconfig for all 3 hosts

Each host can be seen utilizing its both the links



The end