KENNESAW STATE
UNIVERSITY

Module 7:

Emulating a Physical System

Dr. Maria Valero

Agenda

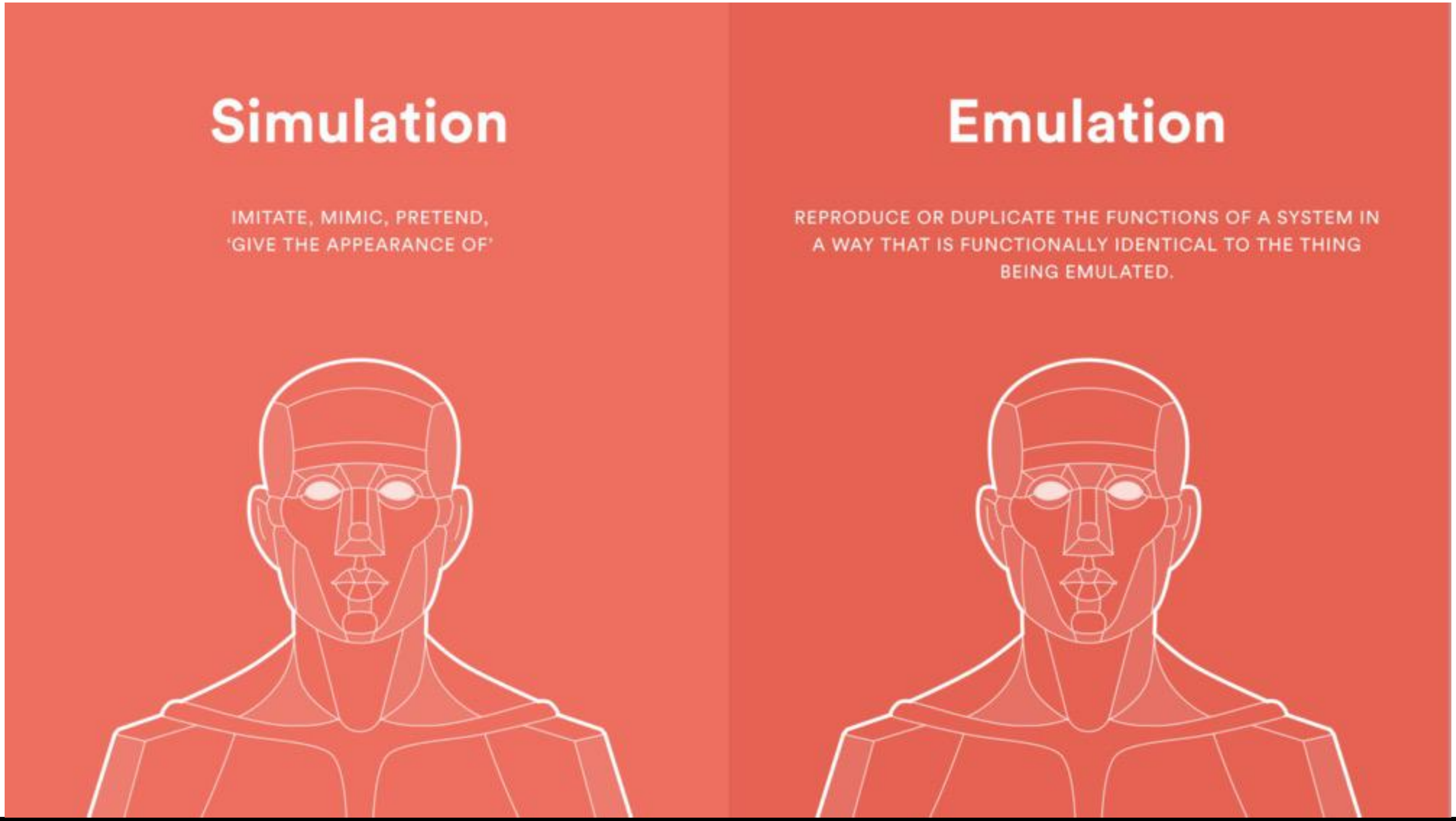
- Introduction to Emulation
- Differences between emulation and simulation
- Types of emulation
- CORE emulator

Introduction to emulation

- The advancement of large-scale computer and communication networks, such as Internet, power grid control networks, heavily depends on the successful transformation from in house research efforts to real productions

- To enhance this transformation, research has created various network test beds that use emulation for conducting medium to large scale experiments

Differences between emulation and simulation



Types of Emulation

- Hardware Emulation: The use of special purpose hardware to emulate the behavior of a yet-to-be-built system, with greater speed than pure software emulation.

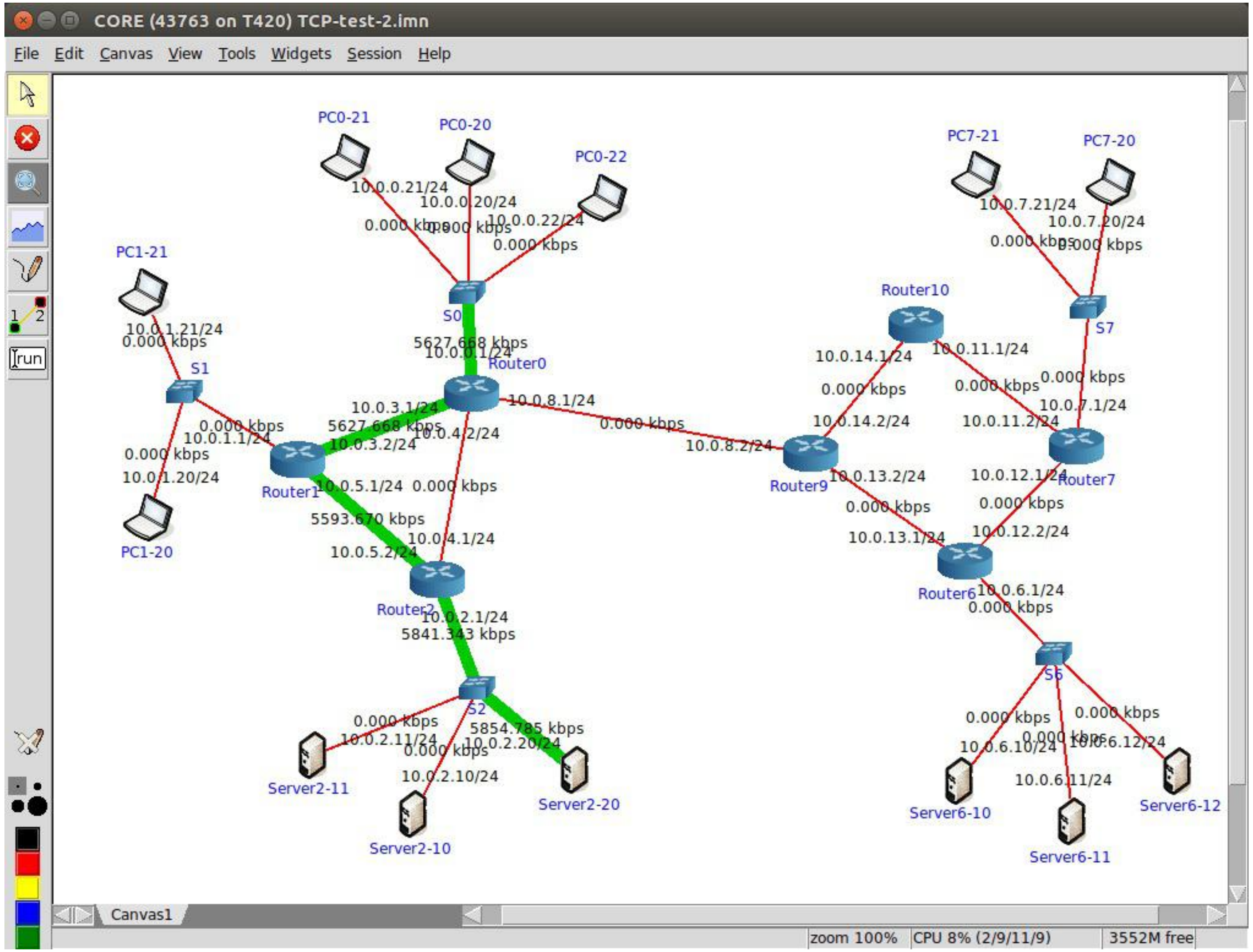
- Software Emulation: Running a program or other software designed to a different system. It is the duplication of functionality of systems.

Example of power grid testbed

CORE Emulator (1)

- The Common Open Research Emulator (CORE) is a tool for emulating networks on one or more machines. You can connect these emulated networks to live networks. CORE consists of a GUI for drawing topologies of lightweight virtual machines, and Python modules for scripting network emulation.

CORE Emulator (2)



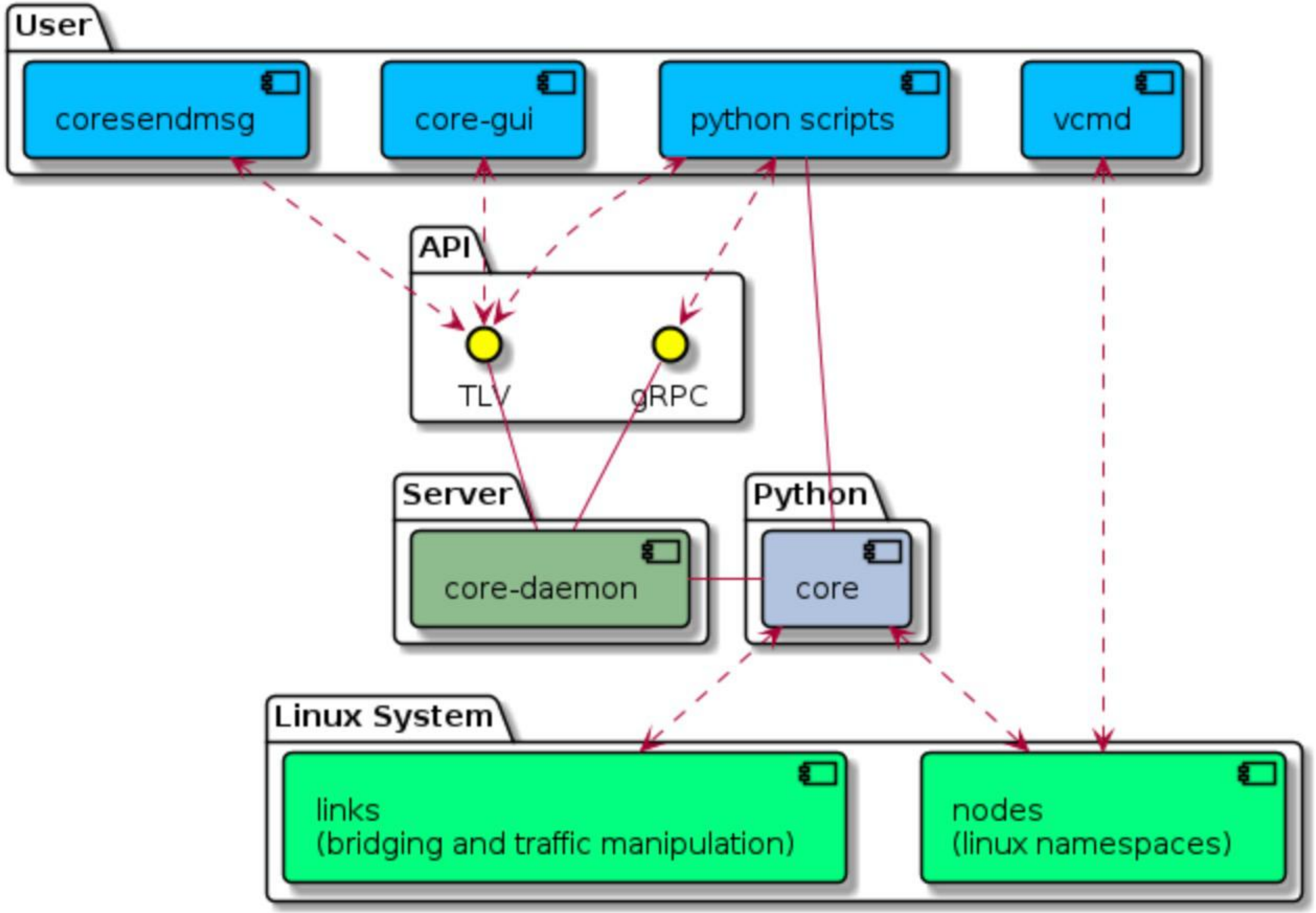
CORE main components (1)

- core-daemon
  - Manages emulated sessions of nodes and links for a given network
  - Nodes are created using Linux namespaces
  - Links are created using Linux bridges and virtual ethernet peers
  - Packets sent over links are manipulated using traffic control
  - Controlled via the CORE GUI
  - Provides both a custom TLV API and gRPC API
  - Python program that leverages a small C binary for node creation

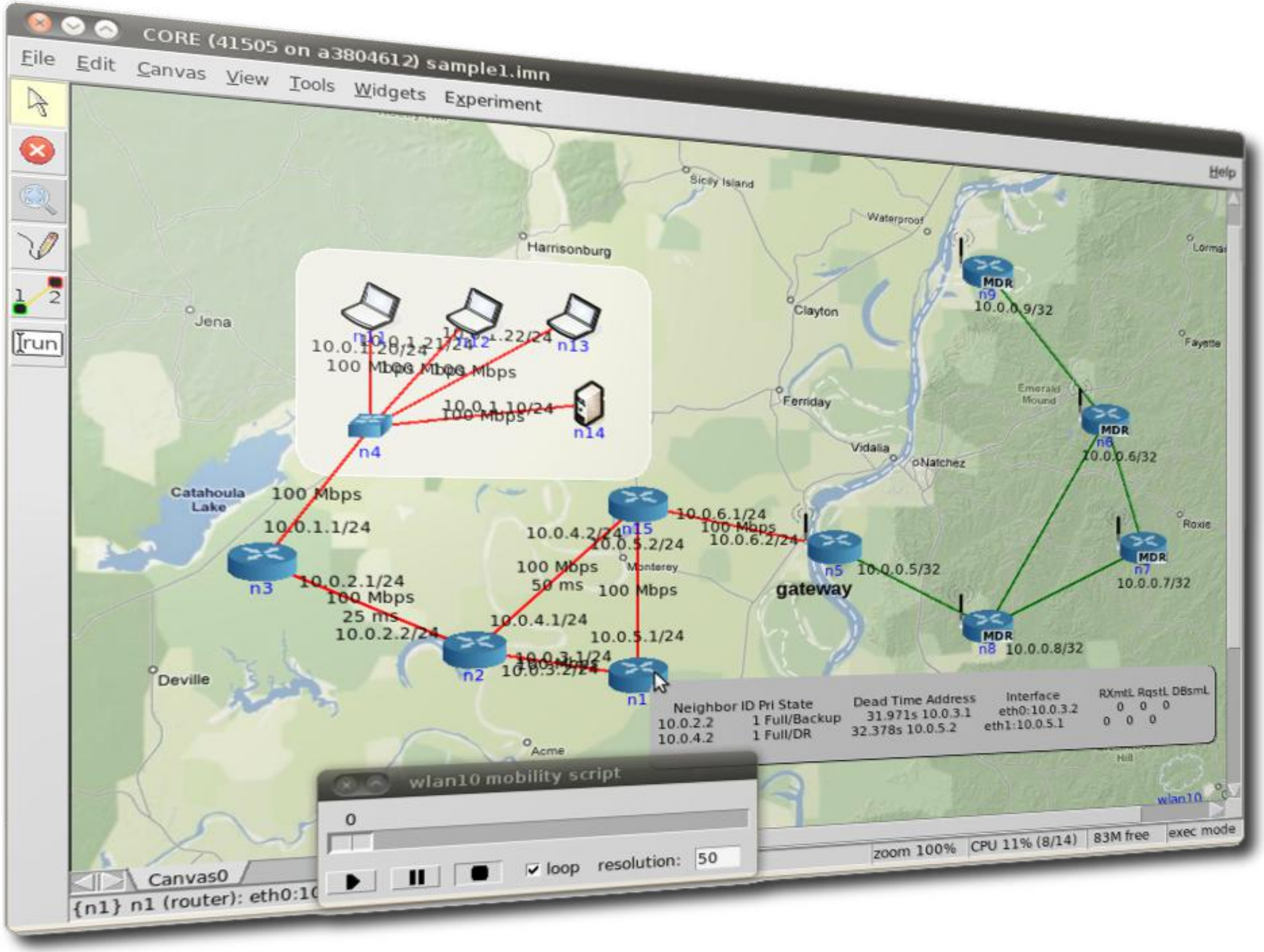
CORE main components (2)

- core-gui
  - GUI and daemon communicate over the custom TLV API
  - Drag and drop creation for nodes and links
  - Can launch terminals for emulated nodes in running sessions
  - Can save/open scenario files to recreate previous sessions
  - TCL/TK program
- coresendmsg
  - Command line utility for sending TLV API messages to the core-daemon
- vcmd
  - Command line utility for sending shell commands to nodes

CORE main components (3)



CORE GUI (1)


CORE (41505 on a3804612) sample1.imn
File
Edit
Canvas
View
Tools
Widgets
Experiment
Help
run
Harrisonburg
Jena
Clayton
Ferriday
Vidalia
Natchez
Catahoula Lake
Deville
Acme
gateway
n1
n2
n3
n4
n5
n13
n14
n15
10.0.1.1/24
10.0.2.1/24
10.0.2.2/24
10.0.4.1/24
10.0.5.1/24
10.0.6.1/24
10.0.6.2/24
10.0.0.5/32
10.0.0.6/32
10.0.0.7/32
10.0.0.8/32
10.0.0.9/32
100 Mbps
25 ms
50 ms
MDR
Neighbor ID Pri State
Dead Time Address
Interface
RXmtL RqstL DBsmL
10.0.2.2 1 Full/Backup 31.971s 10.0.3.1 eth0:10.0.3.2 0 0 0
10.0.4.2 1 Full/DR 32.378s 10.0.5.2 eth1:10.0.5.1 0 0 0
wlan10 mobility script
loop
resolution: 50
zoom 100%
CPU 11% (8/14)
83M free
exec mode
Canvas0
{n1} n1 (router): eth0:10

CORE GUI (2)

- The GUI is used to draw nodes and network devices on a canvas, linking them together to create an emulated network session.
- After pressing the start button, CORE will proceed through these phases, staying in the **runtime** phase. After the session is stopped, CORE will proceed to the **data collection** phase before tearing down the emulated state.
- CORE can be customized to perform any action at each state.

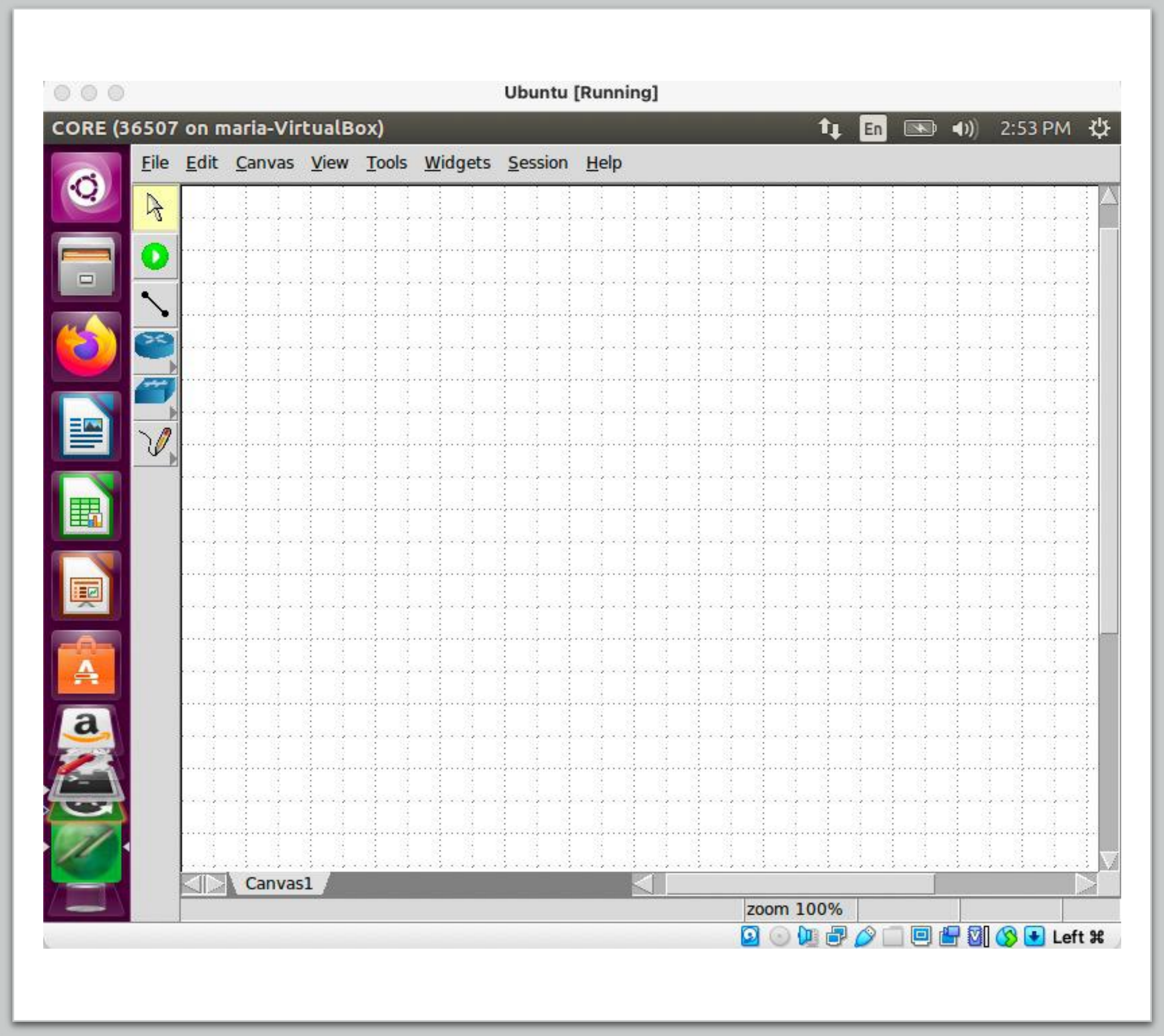
https://coreemu.github.io/core/gui.html

CORE to emulate IoT

- CORE can emulate any type of device, so we can use it for emulating some IoT devices, especially those that collect data from real world and transmit it to another device or server

CORE Example (1)

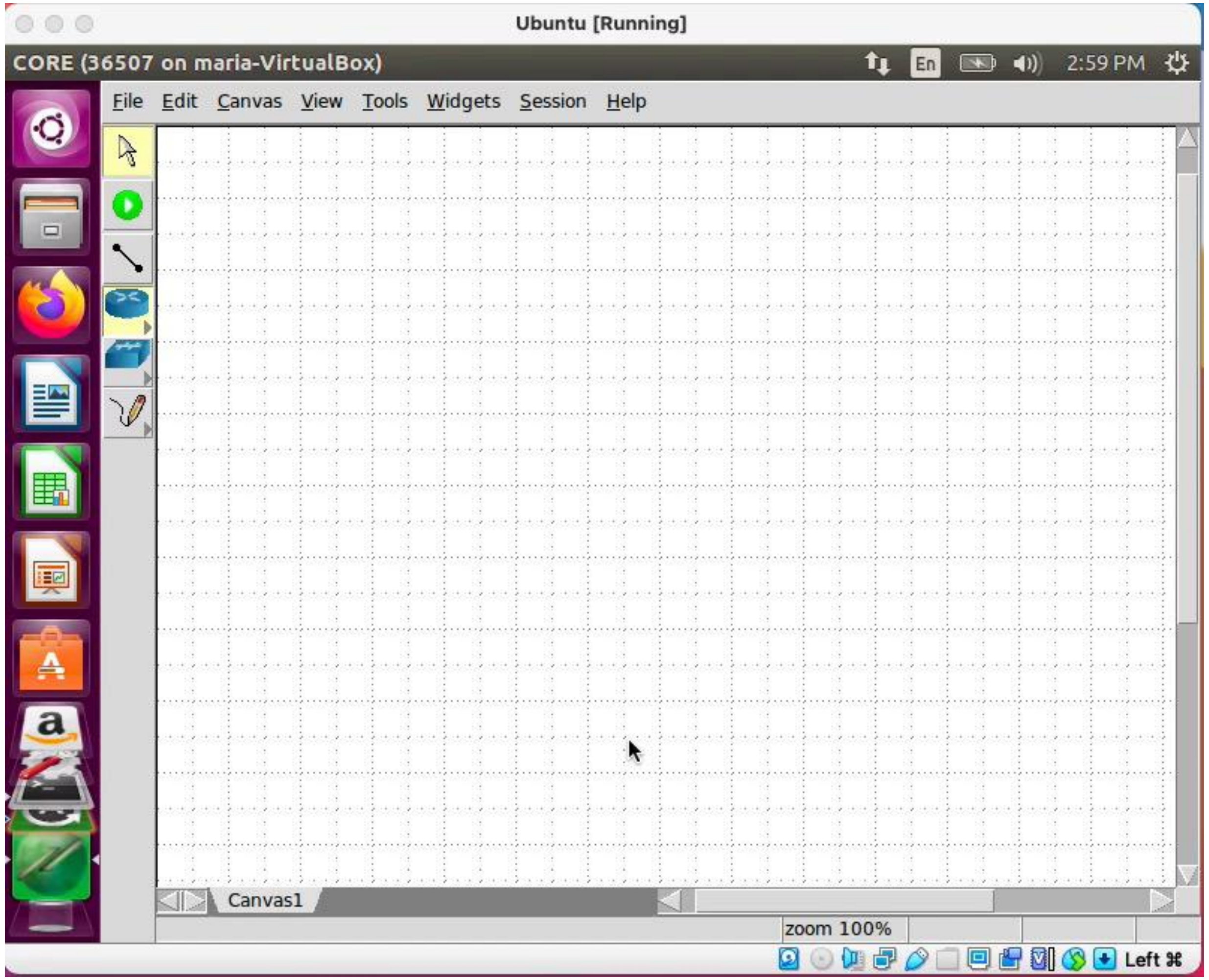
Open CORE



CORE Example (2)

- First, set up a small network simulation scenario on the CORE Network Emulator. In this example, we will use the first three L3 node types: *router*, *PC*, and *host*

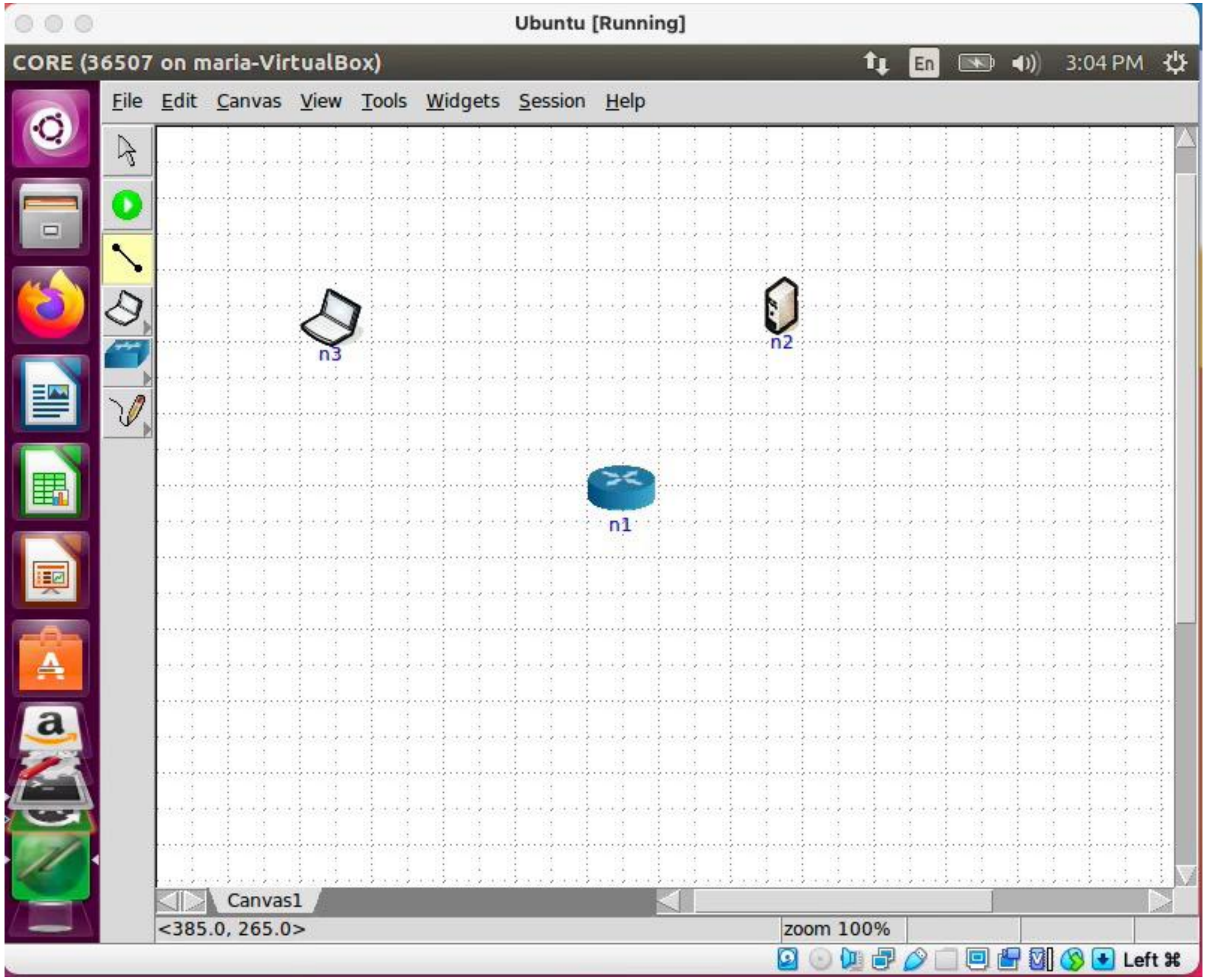
- *(RUN THE VIDEO →)*



CORE Example (3)

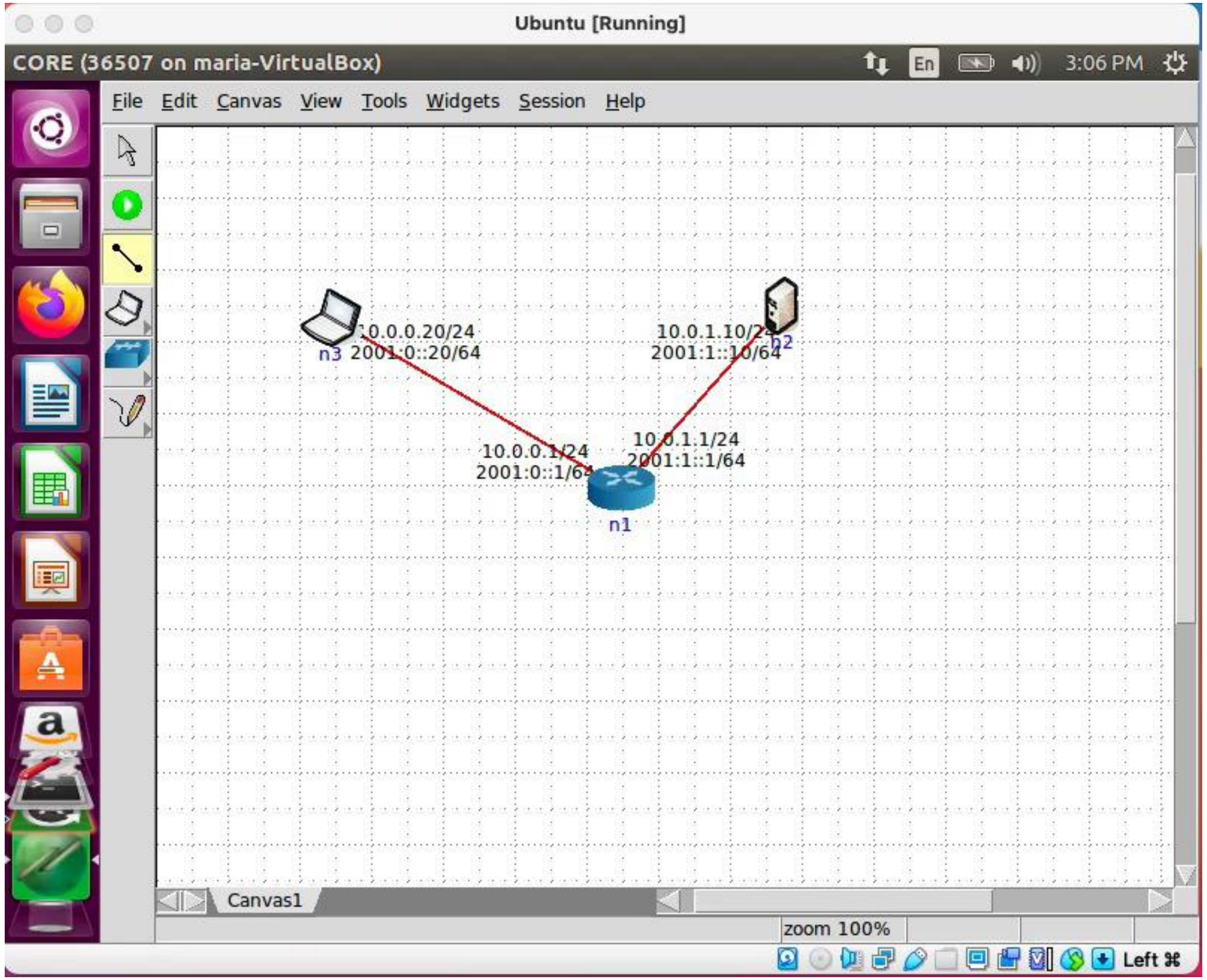
- Connect the devices in the following way

- *(RUN THE VIDEO →)*



CORE Example (4)

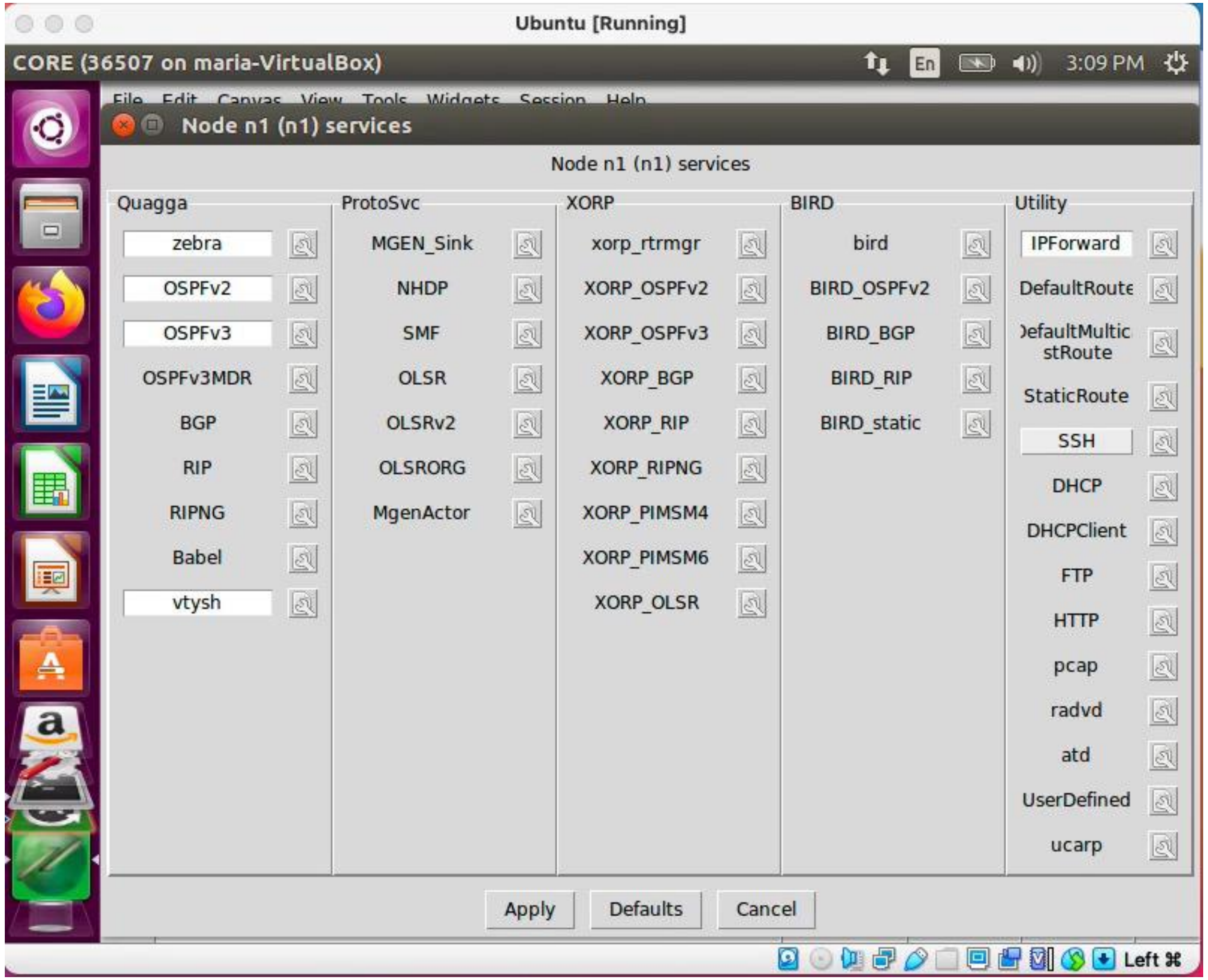
- To see the CORE Services window, right-click on the router node and select *Services…* from the drop-down menu.
- *(RUN THE VIDEO →)*



CORE Example (5)

- The File TAB
  - In the service configuration window's Files tab, we see that the *zebra* service will use a configuration file stored at the location shown in the *Filename* field. If multiple configuration files are needed, they can be selected from a drop-down list.

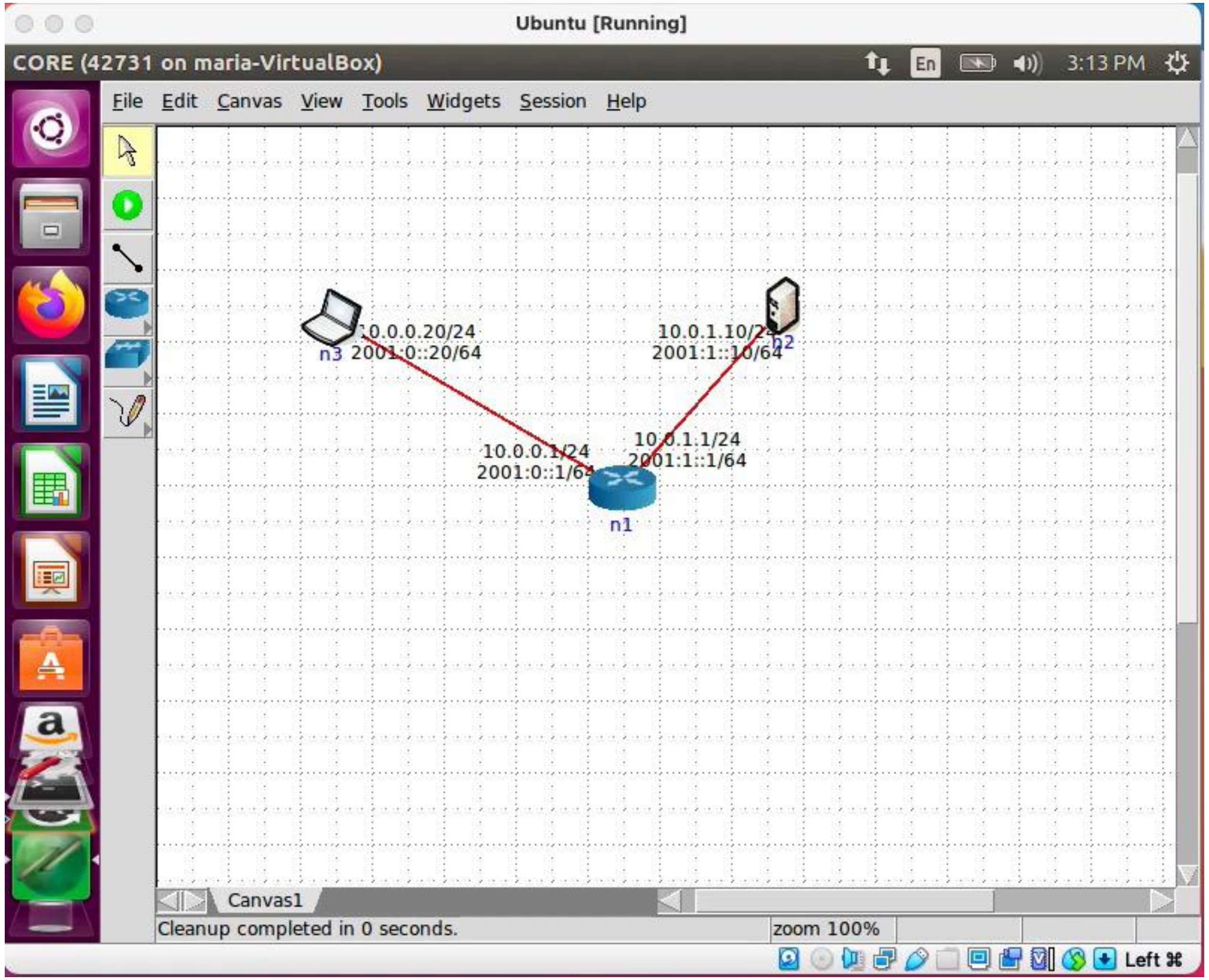
(RUN THE VIDEO →)



CORE Example (6)

- Running the emulator
  - Click on the run symbol to start the emulation
  - If you double-click on any of the devices, you will open a terminal emulating that specific device

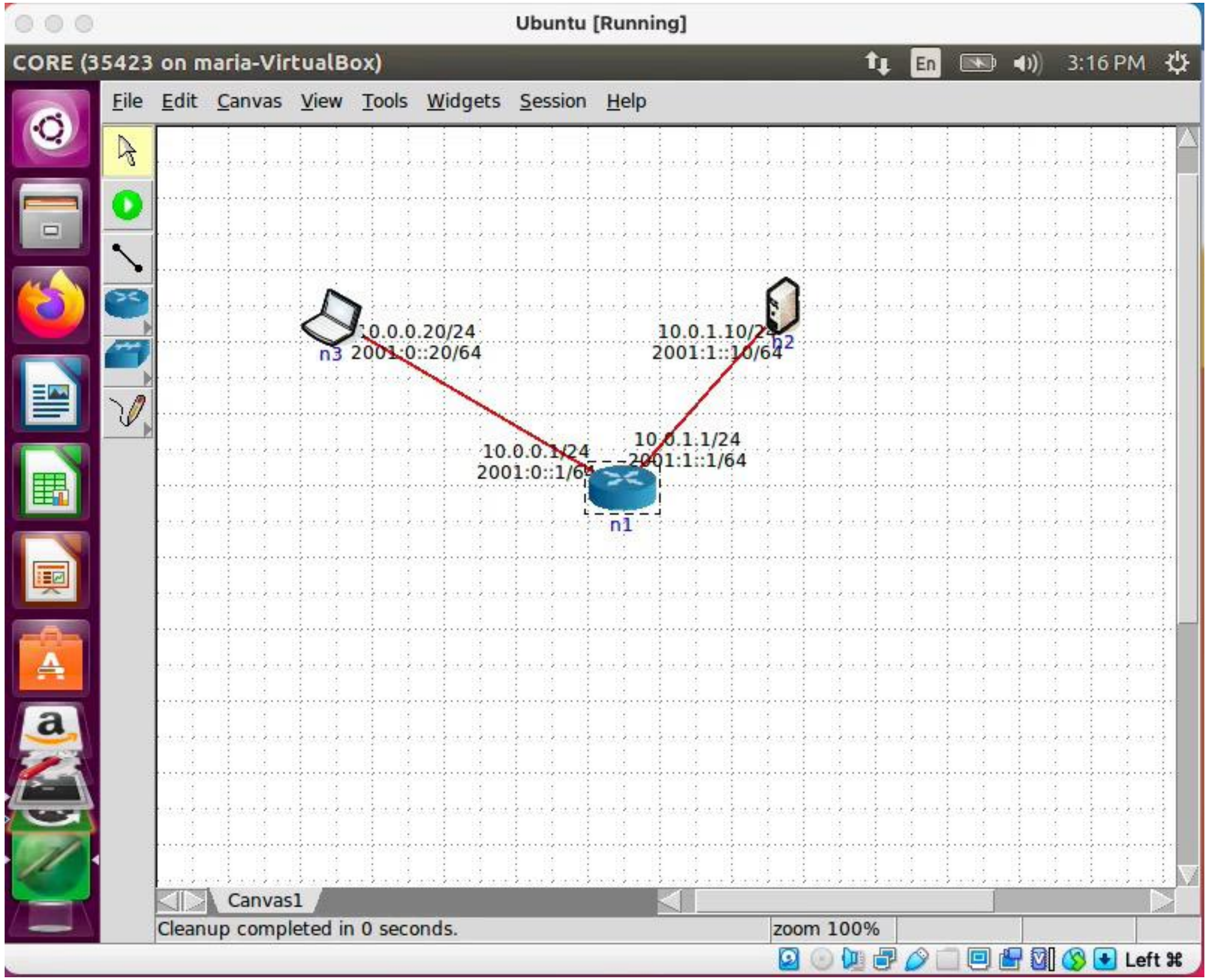
(RUN THE VIDEO →)



CORE Example (7)

- Executing Actions
  - If we double-click on a device, in the terminal, we can perform actions like Linux commands or even ping other devices

(RUN THE VIDEO →)



CORE Example (8)

- You can create python scripts inside each device.
  - Look at one example that creates a python script to print a vector
  - Note that at the beginning export TERM=linux is to enable "nano" in CORE

(RUN THE VIDEO →)