Institute for the Wireless Internet of Things
at Northeastern University

EECE5155: Wireless Sensor Networks and the Internet of Things

Josep Miquel Jornet
Associate Professor, Department of Electrical and Computer Engineering
Director, Ultrabroadband Nanonetworking Laboratory
Member, Institute for the Wireless Internet of Things
Northeastern University
jmjornet@northeastern.edu
www.unlab.tech

Institute for the Wireless Internet of Things at Northeastern

Module L1: Introduction to ns-3 and Wireshark

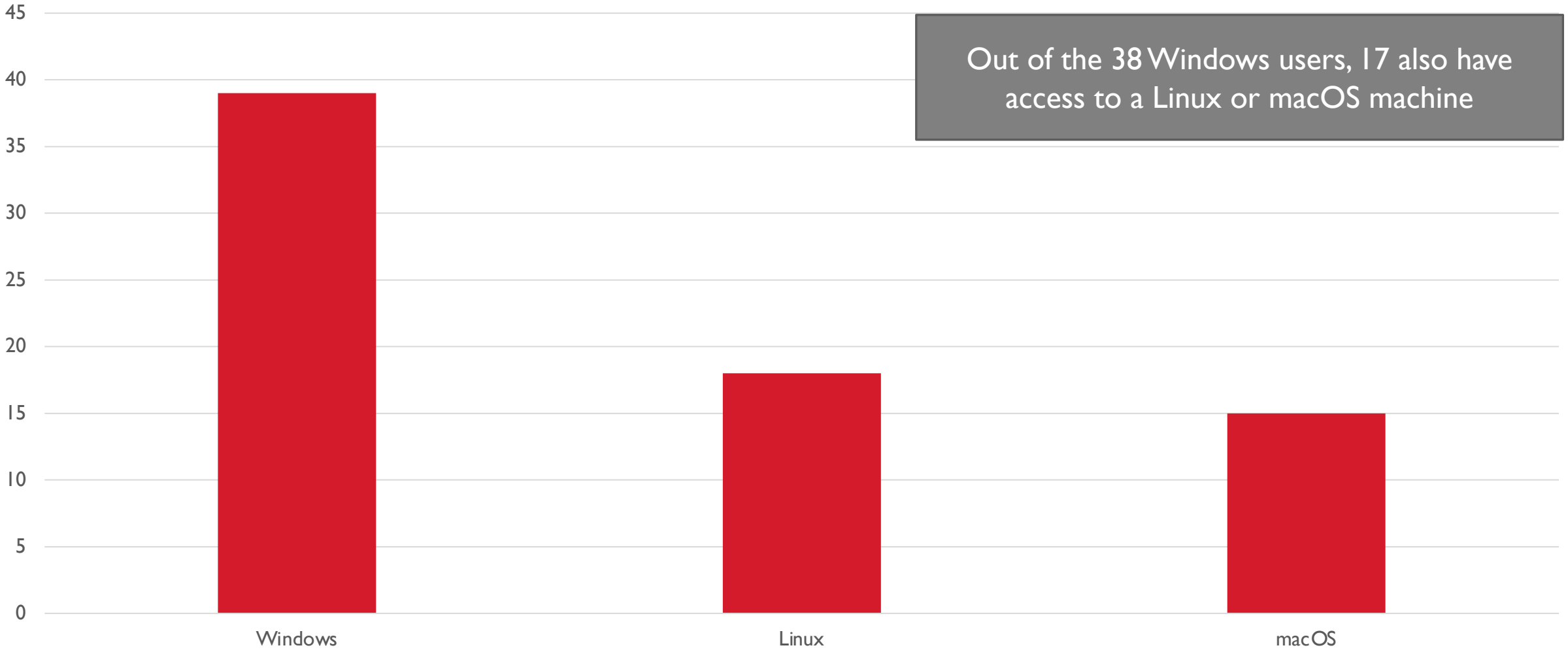
The Tools

Institute for the Wireless Internet of Things at Northeastern

- To complete the computer laboratory assignments, you will need a working installation of ns-3 and Wireshark:
  - **ns-3** is primarily developed on GNU/Linux and macOS platforms, and the minimal requirements to run basic simulations are a C++ compiler (e.g., g++, clang++) and Python (version 3) interpreter.
  - **Wireshark** is a cross-platform application that works in Windows, Linux and macOS

Results from the Logistics Survey

Institute for the Wireless Internet of Things at Northeastern



Out of the 38 Windows users, 17 also have access to a Linux or macOS machine

Setting the Working Environment: ns-3

Institute for the Wireless Internet of Things at Northeastern

- **Case 1:** If you have access to a machine with Linux/Unix or macOS, you can directly install ns-3 following the instructions here:
  - https://www.nsnam.org/wiki/Installation#Installation

Setting the Working Environment: ns-3

Institute for the Wireless Internet of Things at Northeastern

- **Case 2:** If you only have access to a Windows machine or do not have administrator privileges in your Linux/Unix or macOS machine:
  - **Option 1:** You can create your own virtual machine with Linux (more in the next slides)
    - → Think of this a good set of skills to incorporate in your engineering toolbox!
  - **Option 2:** You can download an empty virtual machine on which you can then install ns-3:
    - → It will probably take some time, particularly if you are not familiar with the Linux terminal

    https://northeastern-my.sharepoint.com/:f:/g/personal/jmjornet_northeastern_edu/EuIn-VaIMF9Fqyo5Xmy-IMwBXdLIAnUc2C1IPm3yTqc6Bw?e=ykWGSD
  - **Option 3:** You can download a virtual machine with ns-3 already installed:
    - → **Preferred option for non-Linux users**

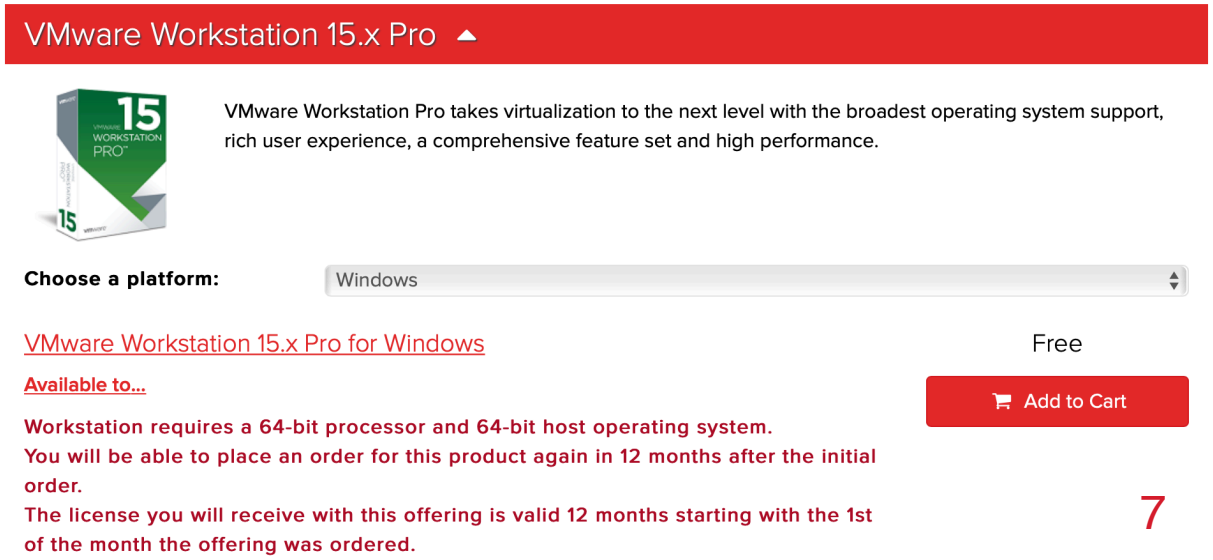
    https://northeastern-my.sharepoint.com/:f:/g/personal/jmjornet_northeastern_edu/Ek2IF72yw59HpQcKY1kyro4Bh27sdvTM3sbZqvMFVIcirw?e=idfIN9

For All: Installing VMware Workstation Pro

Institute for the Wireless Internet of Things at Northeastern

- Whether you are creating your own virtual machine or downloading the class-ready image, you will need VMware Workstation Pro for PC (or VMware Fusion for Mac).

- As with many other software tools, you can use your university credentials to obtain VMware Workstation for PC (or VMWare Fusion for Mac) for free:
  - Navigate to: https://neu.onthehub.com/WebStore/Welcome.aspx
    - Make sure to login with the university credentials
  - Search for VMware Workstation 15.x Pro
    - (Or Fusion, if you are on a Mac)
  - Add it to the Cart
  - Complete the (free) purchase steps
    - Make sure you save the serial number
  - Download, install and register it



Option 1: Creating your own Virtual Machine

Institute for the Wireless Internet of Things at Northeastern

- After installing VMware Workstation Pro, download your preferred Linux distribution:
  - For its stability and full compatibility with ns-3, we recommend the use of **Ubuntu**:
    - https://ubuntu.com/download/desktop

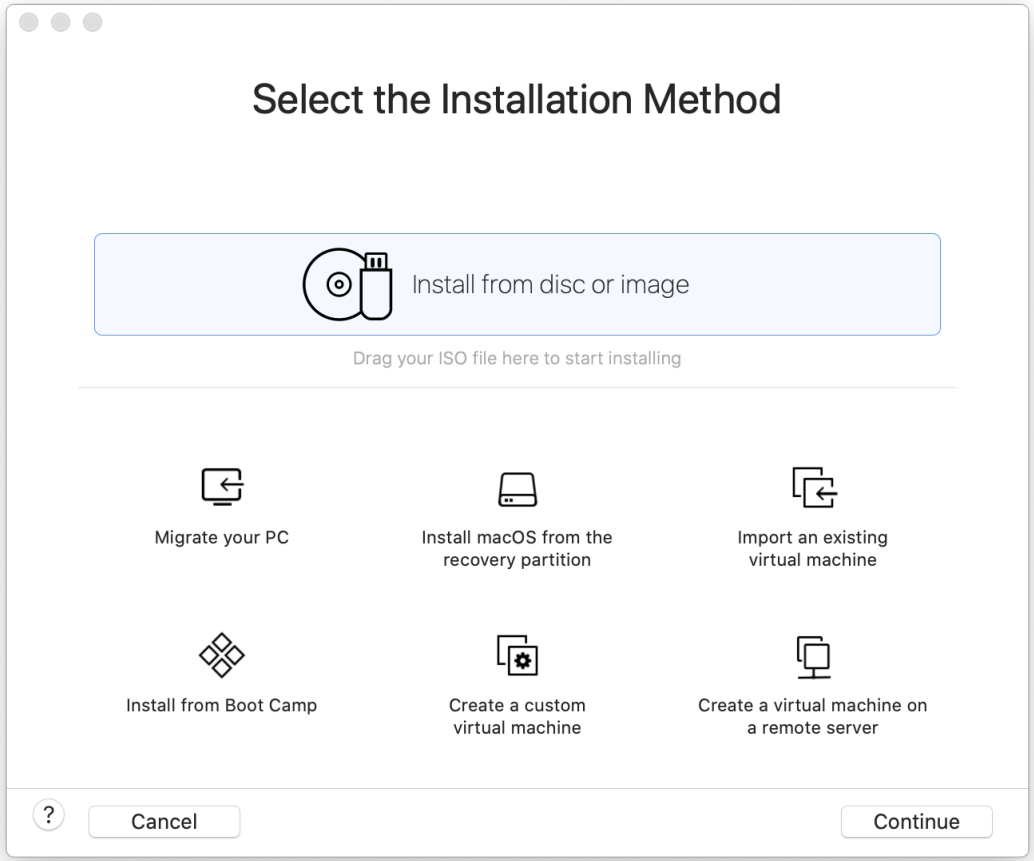
- **DO NOT** proceed with the installation of the ISO image
  - This would change the configuration of your real machine → Remember we want a virtual machine!

Option 1: Creating your own Virtual Machine

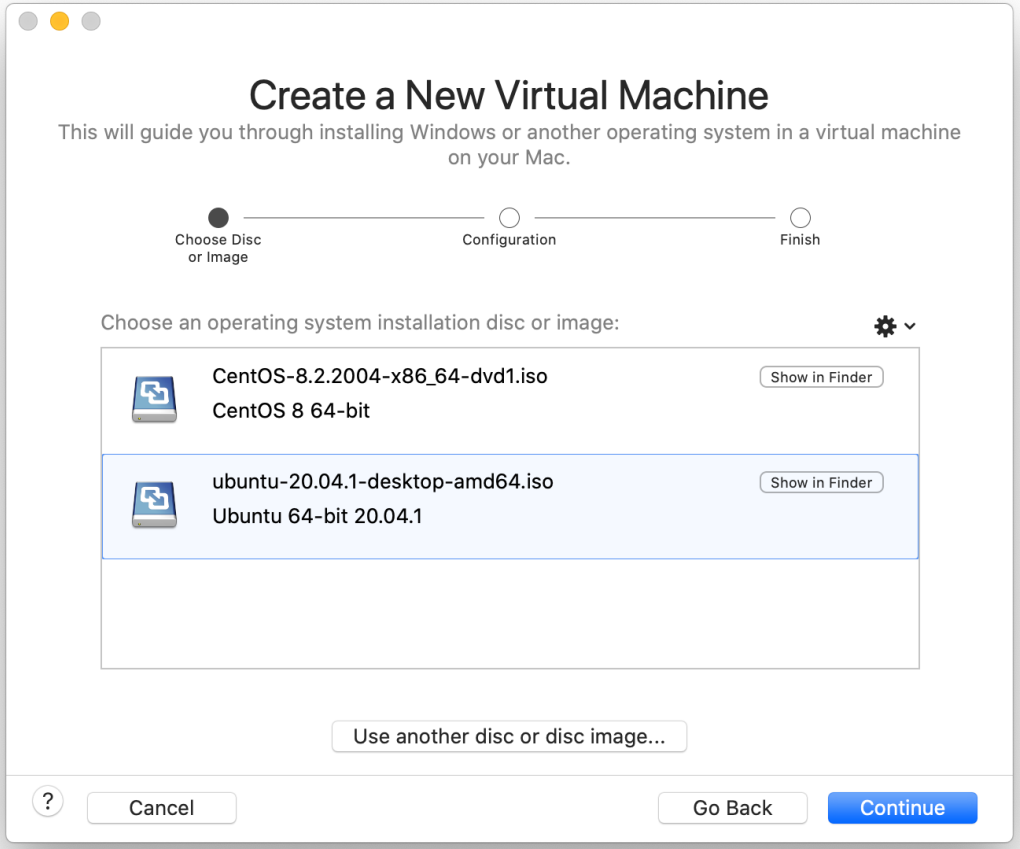
Institute for the Wireless Internet of Things at Northeastern

- Start VMware Workstation (or Fusion, if you are on a Mac)
- Follow the onscreen instructions to create your virtual machine:

1)



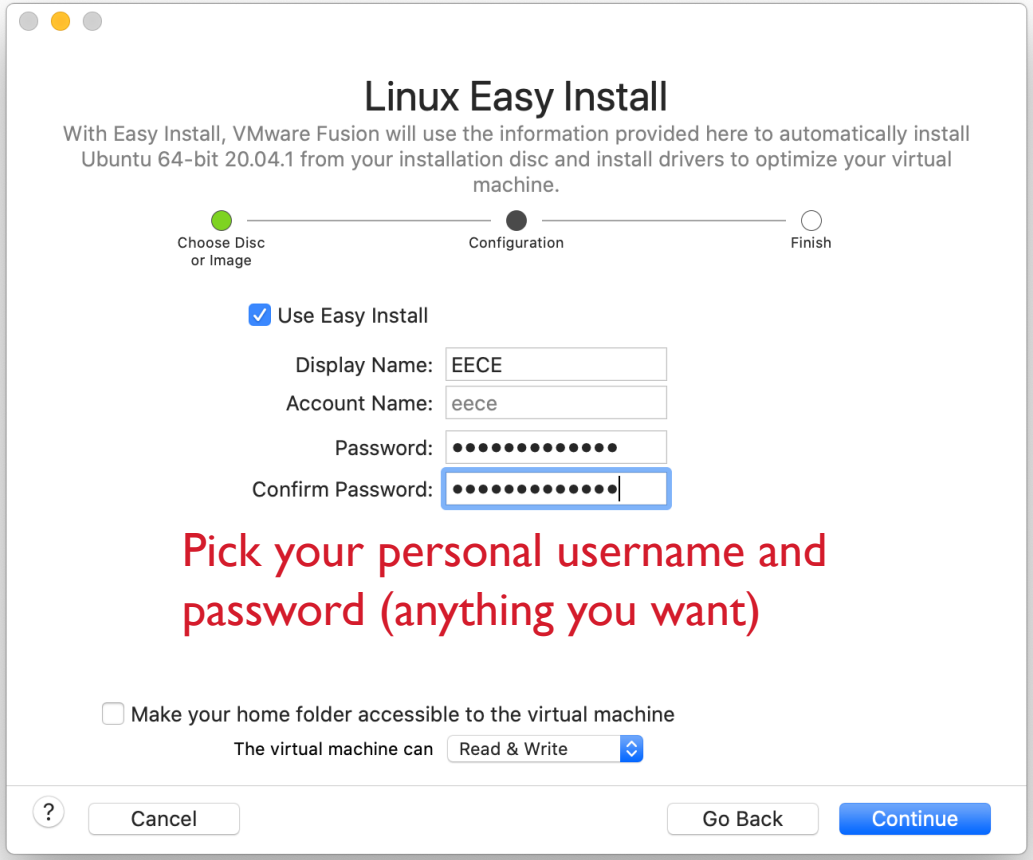
2)



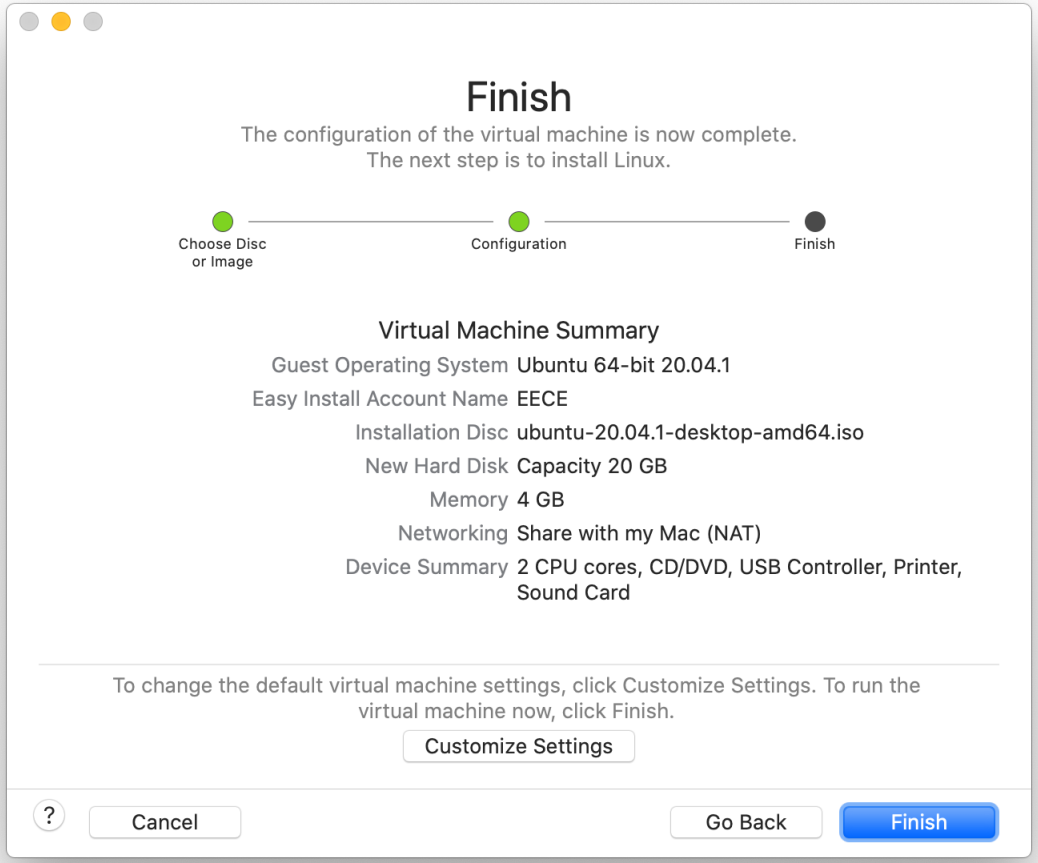
Option 1: Creating your own Virtual Machine

Institute for the Wireless Internet of Things at Northeastern

3)



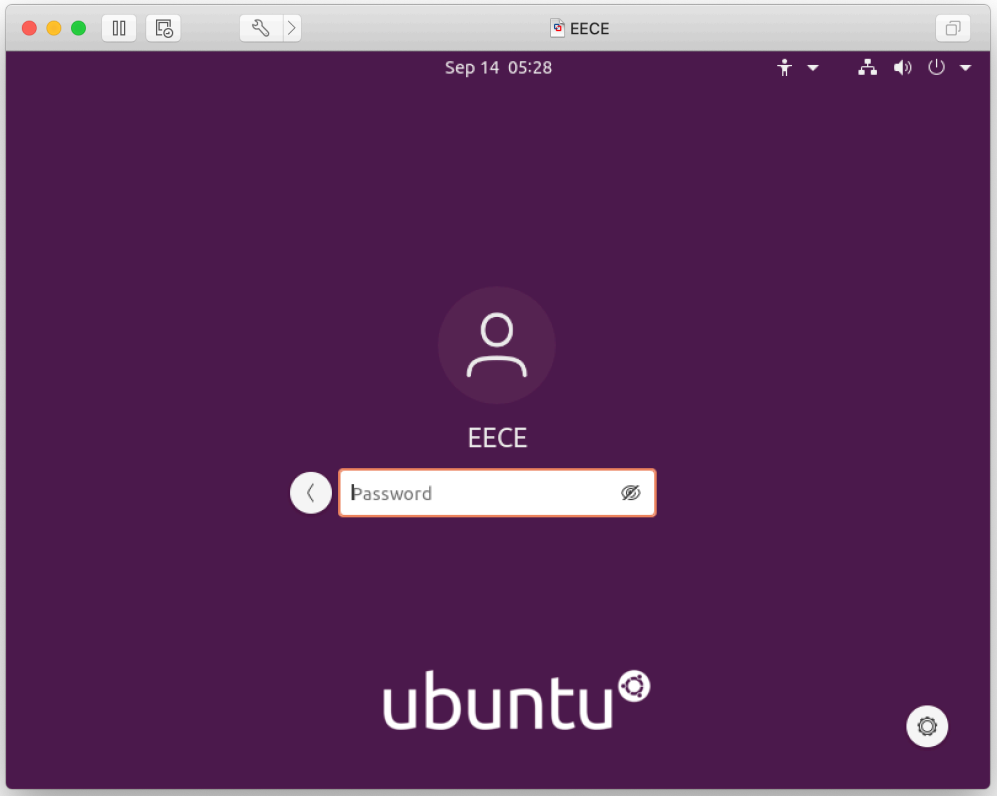
4)

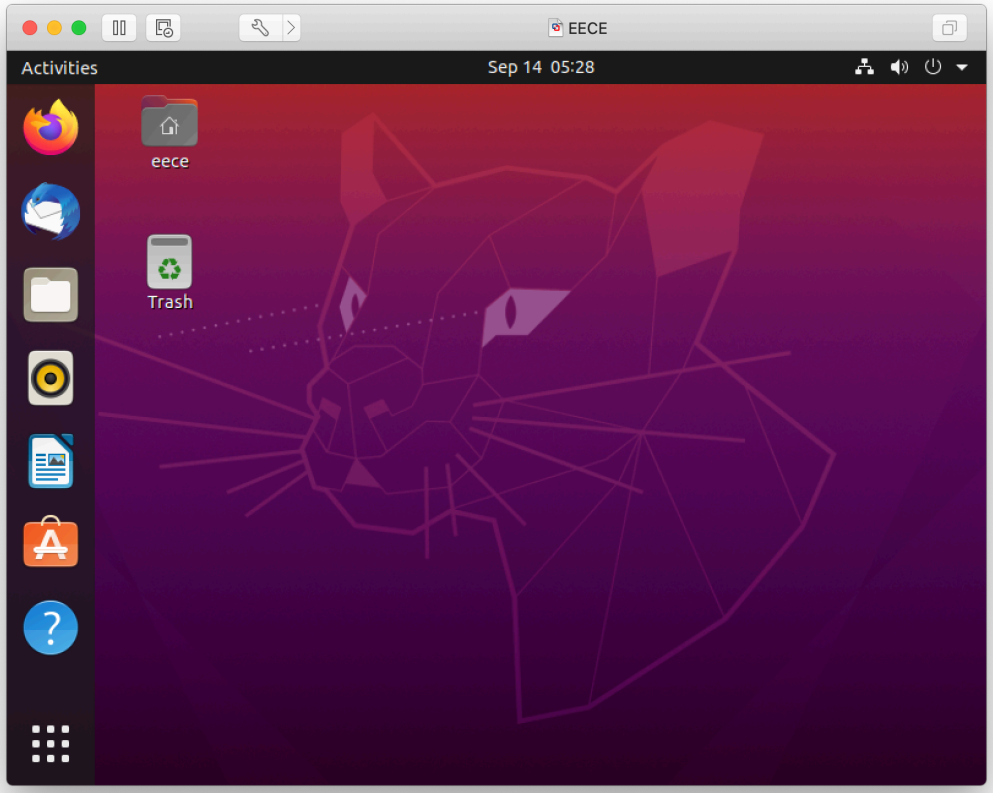


Option 1: Creating your own Virtual Machine

Institute for the Wireless Internet of Things at Northeastern

- Follow the on-screen instructions to finalize the creation of your own virtual machine





Congratulations, your machine is ready!!!
Now follow the instructions for Option 2 to install ns-3

For All: Running the Virtual Machine

Institute for the Wireless Internet of Things at Northeastern

- **Option 1:** If you have created your own virtual machine using VMware Workstation Pro, you can now use the same tool to run the virtual machine
- **Options 2 and 3:** If you have downloaded the course virtual machine, you will still need to download and install the VMware Workstation Pro
  - For that, follow the steps in Slide 6 to purchase (for free), download, install and registering it
  - After that, open the tool and select "Open a Virtual Machine", and select the course virtual machine
    - **Important Note:** You might receive multiple warnings:
      1. If asked, say "No" to trying to connect to virtual device ide1:0
      2. If asked, indicate that you have copied the virtual machine
      3. If asked, indicate that you want to take over the existing user

      (You might not see any of these, it all depends on your system)

For All: Speeding up the Virtual Machine

Institute for the Wireless Internet of Things at Northeastern

- If you feel that your virtual machine is very slow, you can adjust the amount of resources that you want to allocate to it. For that,
  - Suspend or power off the virtual machine
  - Click on "Edit virtual machine settings"
  - Select the amount of memory (at least 4 Gb) and the number of processors (at least 2) that you want your virtual machine to have

Options 1 & 2: Installing ns-3 in the Virtual Machine

Institute for the Wireless Internet of Things at Northeastern

- Whether you created or download the virtual machine, you are ready to install ns-3 from the terminal
  - Continue with the steps here:
    - https://www.nsnam.org/wiki/Installation#Ubuntu.2FDebian.2FMint
      - **Note:** you will need root privileges to run the installations commands. For that, add sudo in front of each command
        - For example, start with: `sudo apt-get install g++ python3`
          - The first time it will ask you for your password (N0rthe@stern. if using the course machine)
      - **Note:** If the pip package is not found, use: `sudo apt-get install python3-pip`
      - **Note:** If after "baking", the cMake is missing, follow the instructions here to install it: https://vitux.com/how-to-install-cmake-on-ubuntu-18-04/
        - If the cMake installation is giving an error about not finding openSSL, try with: `sudo apt-get install libssl-dev`
- **Important:** There is some merit in learning how to do all these things, but if this is very far from your comfort zone, just go with Option 3! (the goal is to get to use ns-3)

Options 1 and 2: Installing Wireshark

Institute for the Wireless Internet of Things at Northeastern

- The last step is to install Wireshark in your virtual machine. For that,
  - Within the terminal:
    - `sudo apt install wireshark`

- (Wireshark is already installed in Option 3)

Setting the Working Environment

Finally, we are ready to use the system!

Network Simulator 3

ns-3

Institute for the Wireless Internet of Things at Northeastern

- A discrete-event network simulator targeted for research and educational use
  - **Network simulator:**
    - A software tool that predicts the behavior of a network, without an actual network being present
      → Imitates a real telecommunication network without the real cost
  - **Discrete-event:**
    - Each event occurs at a particular instant in time → Marks a change of state in the system
    - No change in the system is assumed to occur during events
      → The simulation can directly jump in time from one event to the next → Huge time savings!
- **Other simulators:**
  - OPNET, OMNET, NetSim, etc.

Simulation Workflow

- **Topology Definition:**
  - How many nodes compose the network?
  - How are these nodes? Which application are they running?
  - How are the nodes connected between them? What type of links are they using?

- **Model Development:** to specify the protocols that each component utilizes:
  - Which protocols are the nodes utilizing? Are they already implemented in ns-3? Are we developing new protocols?

- **Node and Link Configuration:** to set the default values for the different parameters in each model:
  - Assign addresses to the nodes, determine the packet length, maximum number of retransmissions, fix the modulation and coding techniques in use, etc.

Simulation Workflow

- **Execution:** to let the nodes generate new events and react accordingly by following the defined models:
  - Packets are generated, sent, retransmitted if necessary, new packets are generated in reply, etc.
  - In the meantime, packets are traced and the information is logged

- **Performance Analysis:** to extract information from the packet traces and logs
  - End-to-end delay, end-to-end packet delivery probability, energy consumption, etc.

- **Graphical Visualization:** to represent the gathered data into more convenient graphical representations

Programming in ns-3

- Many models are already developed:
  - **Physical Layer:**
    - Wired links: point-to-point, shared bus, etc.
    - Wireless links: line-of-Sight, non-line-of-sight, indoor and outdoor propagation models, etc.
  - **Data Link Layer:**
    - CSMA, CSMA-CA, CSMA-CD
  - **Network Layer:**
    - Routing Protocols: AODV, DSDV, DSR, OLSR
    - IPv4, IPv6
  - **Transport Layer:**
    - TCP, UDP
  - **Full standard implementations:**
    - IEEE 802.11 and variants, WiMax, LTE, LoRaWAN

Programming in ns-3

- ns-3 is written in C++ and uses Python for scripting

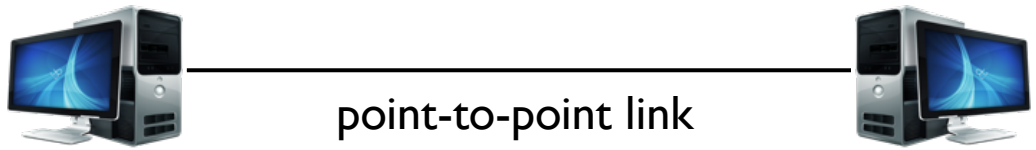
- `waf`
  - A new Python-based build system
  - Used to compile and build a ns-3 simulation

Key Abstractions

- **Node:**
  - The device to which we will add functionalities, e.g., applications, protocol stack, network card, etc.
- **Application:**
  - The user program which generates some activity, e.g., HTTP, FTP, etc.
- **Channel:**
  - The physical link connecting different nodes, e.g., PointToPointChannel (wired point-to-point link), WifiChannel (wireless link), CsmaChannel (ethernet-like link), etc.
- **Net Device:**
  - The network adapter installed in a node to be able to use the channel, e.g., PoinToPointNetDevice, WifiNetDevice, CsmaNetDevice, etc.
- **Topology Helpers:**
  - Helps with the installation/setup of the protocol stack and the Net Device in the nodes

First Example

- Located in source/ns-3.31/examples/tutorial/first.cc



Node 0: UDP Echo Client

point-to-point link
IP Block: 10.1.1.0/24
Data Rate: 5 Mbps
Propagation Delay: 2 ms

Node 1: UDP Echo Server

- UDP Echo is an application in which the client sends some information to the server, who replies back with the same packet:
- We will set up the client to send just one packet at time t=2 sec

First Example

- **Main function:**
  - int main (int argc, char *argv[]) {

- **Topology Helper:**
  - **NodeContainer:** defines the nodes in the network
    - NodeContainer nodes;
    - nodes.Create (**2**);

  - **PoinToPointHelper:** connects the PointToPointNetDevice over the PoinToPointChannel
    - PointToPointHelper pointToPoint;
    - pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
    - pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));

First Example

- **NetDeviceContainer:** installs the PointToPoint devices to the nodes that we have defined
  - NetDeviceContainer devices;
  - devices = pointToPoint.Install (nodes);

- **InternetStackHelper:** sets the protocol stack to our nodes
  - InternetStackHelper stack;
  - stack.Install (nodes);

- **Ipv4AddressHelper:** sets the IP addresses to the nodes
  - Ipv4AddressHelper address;
  - address.SetBase ("10.1.1.0", "255.255.255.0");
  - Ipv4InterfaceContainer interfaces = address.Assign (devices);

First Example

- **Applications:**

  - **UdpEchoServerHelper:** sets the specific application server that will be running in of the nodes
    - UdpEchoServerHelper echoServer (9);
    - ApplicationContainer serverApps = echoServer.Install (nodes.Get (1));
    - serverApps.Start (Seconds (1.0));
    - serverApps.Stop (Seconds (10.0));

  - **UdpEchoClientHelper:** sets the clients application running on the other node
    - UdpEchoClientHelper echoClient (interfaces.GetAddress (1), 9);
    - echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
    - echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
    - echoClient.SetAttribute ("PacketSize", UintegerValue (1024));
    - ApplicationContainer clientApps = echoClient.Install (nodes.Get (0));
    - clientApps.Start (Seconds (2.0));
    - clientApps.Stop (Seconds (10.0));

First Example

- **Simulator:**
  - Simulator::Run ();
  - Simulator::Destroy ();
  - return 0;
  - }
- Time to run the simulation:
  - cd
  - cd source/ns-3.31
  - cp examples/tutorial/first.cc scratch/myfirst.cc
  - ./waf
  - ./waf --run scratch/myfirst
- Output:
  - At time 2s client sent 1024 bytes to 10.1.1.2 port 9
  - At time 2.00369s server received 1024 bytes from 10.1.1.1 port 49153
  - At time 2.00369s server sent 1024 bytes to 10.1.1.1 port 49153
  - At time 2.00737s client received 1024 bytes from 10.1.1.2 port 9

First Example

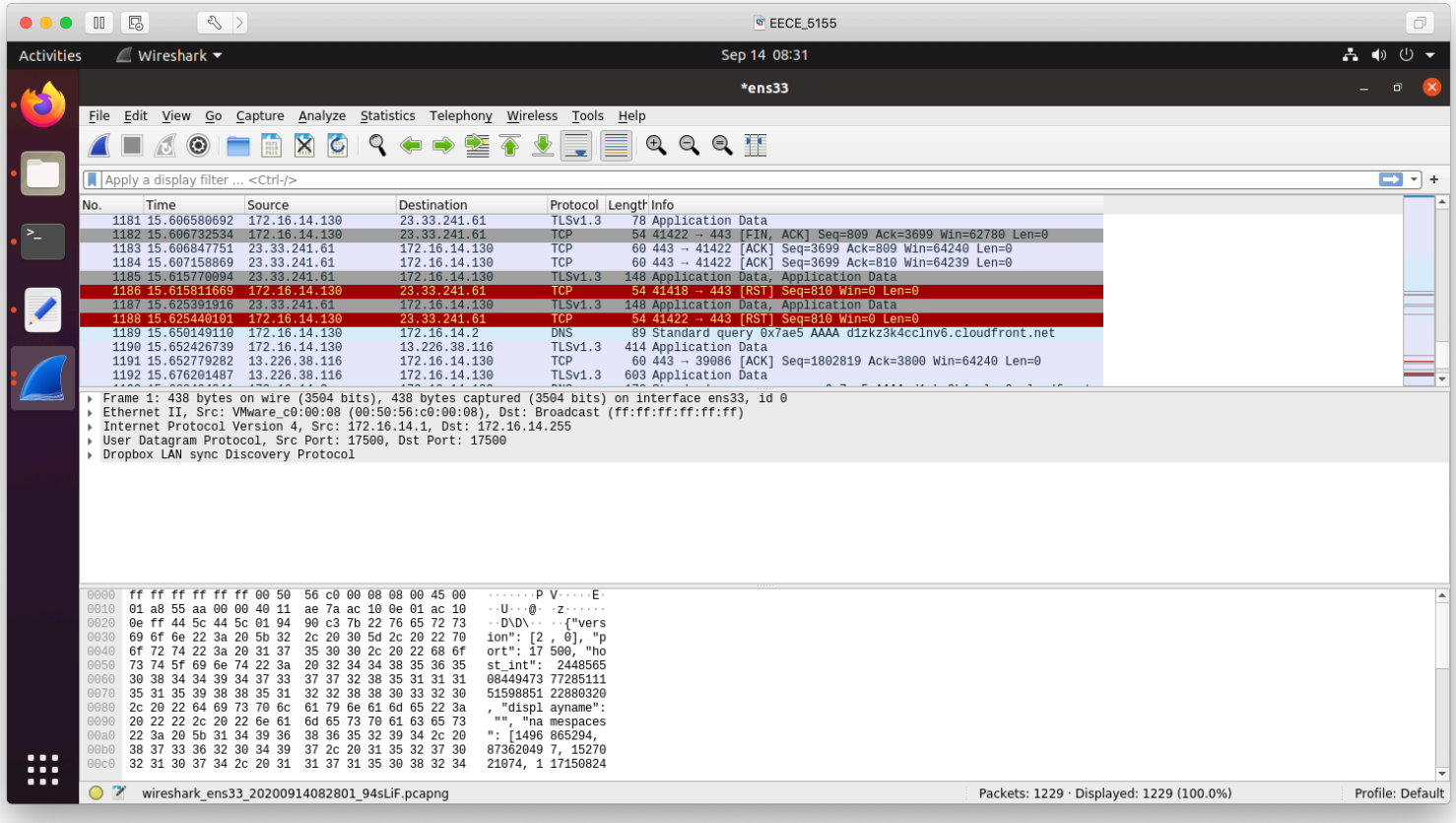
- We can modify the first script first.cc to generate a trace file in pcap format
  - We should add
    - `pointToPoint.EnablePcapAll ("myfirst");`
  - Before
    - `Simulator::Run ();`

- Now, in the current folder, two pcap files have been generated (one for each node):
  - How can we easily see the contents of those files?

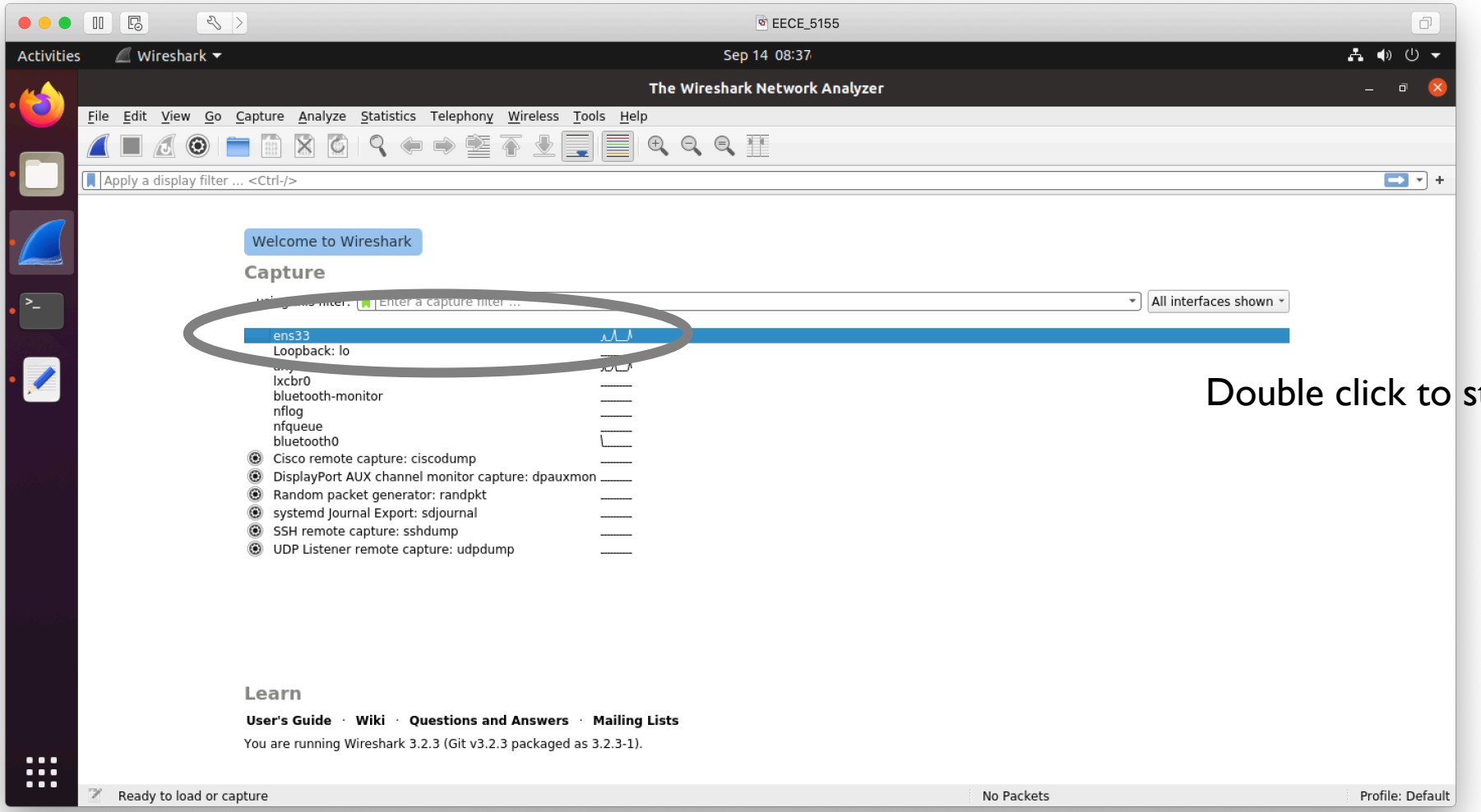
Wireshark

Institute for the Wireless Internet of Things at Northeastern

- An open-source packet analyzer:
  - To capture packets in real time (for which you will need root privileges)
  - To analyze existing packet captures



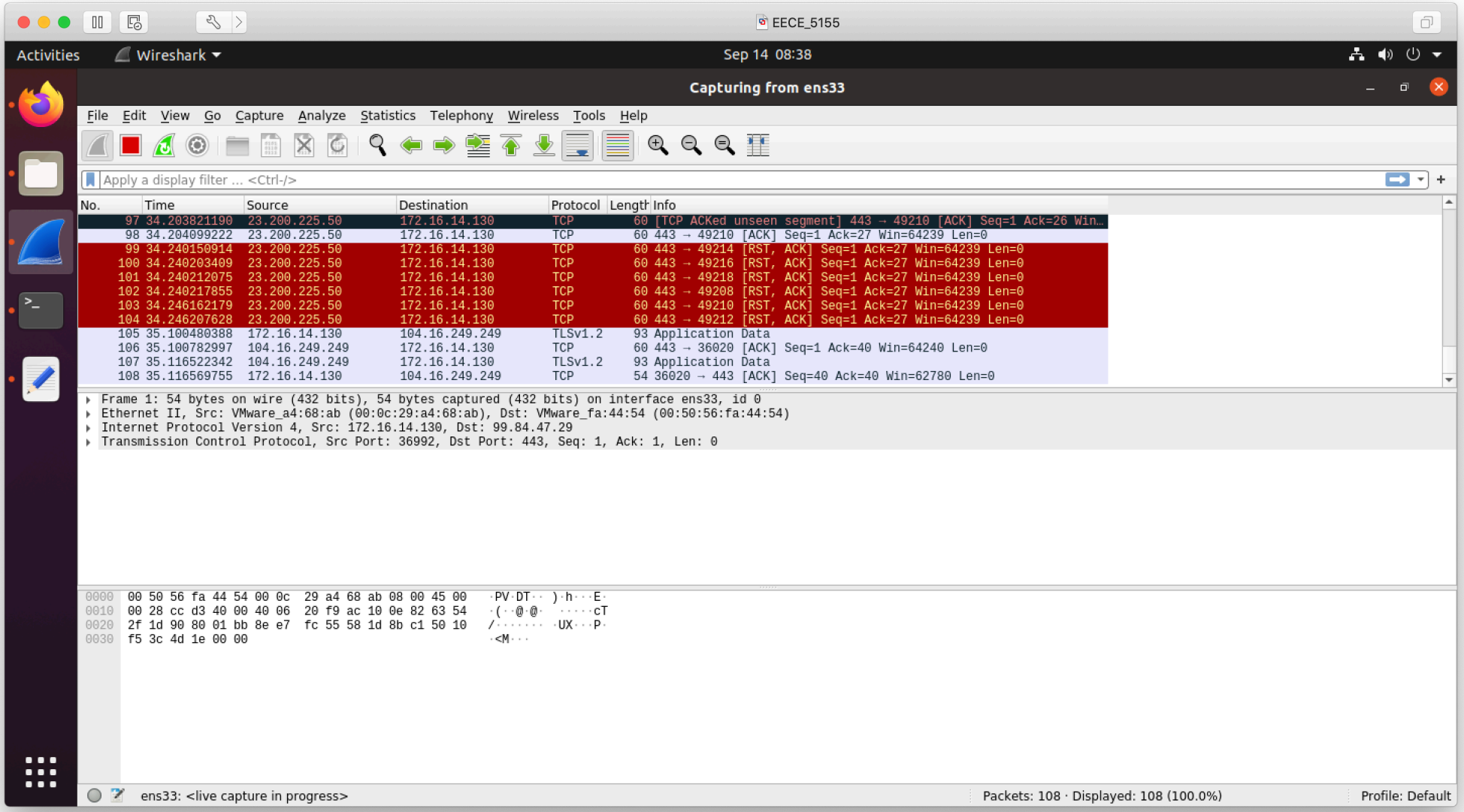
Live Traffic Capture



- Go to terminal and type `sudo wireshark &`

Double click to start capturing!

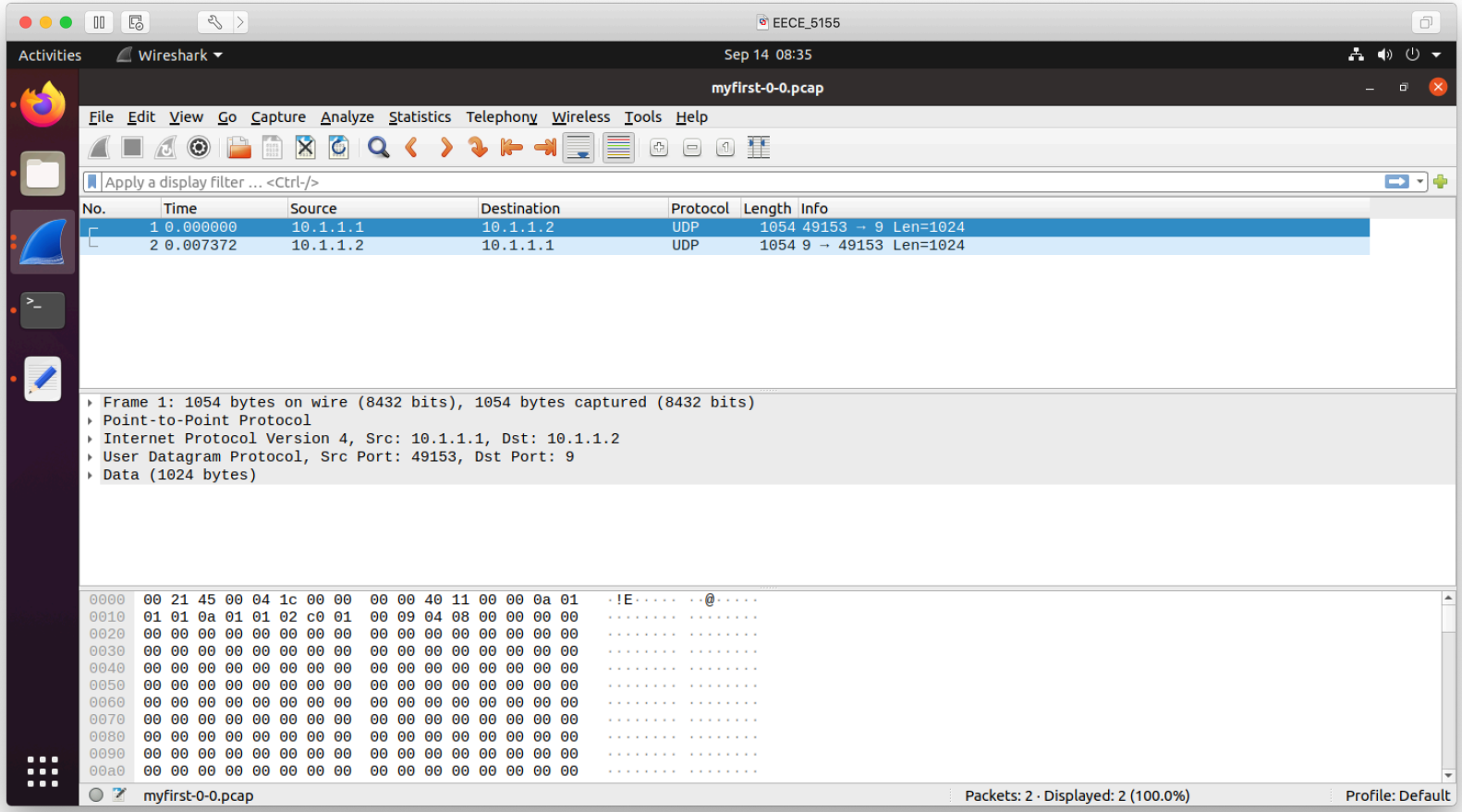
Live Traffic Capture



Going back to our example: myfirst-1-0.pcap

Institute for the Wireless Internet of Things at Northeastern

- Double click on the pcap file generated after running the ns-3 simulation
  - (most likely inside source/ns-3.31/)



Explore the contents of the file

First Laboratory Assignment

Institute for the Wireless Internet of Things at Northeastern

- Will be available on Canvas by the weekend
- Before that:
  - Follow the instructions in these slides to get your system working
  - Play with Wireshark and explore the amount of information it collects
    - Be curious!!!