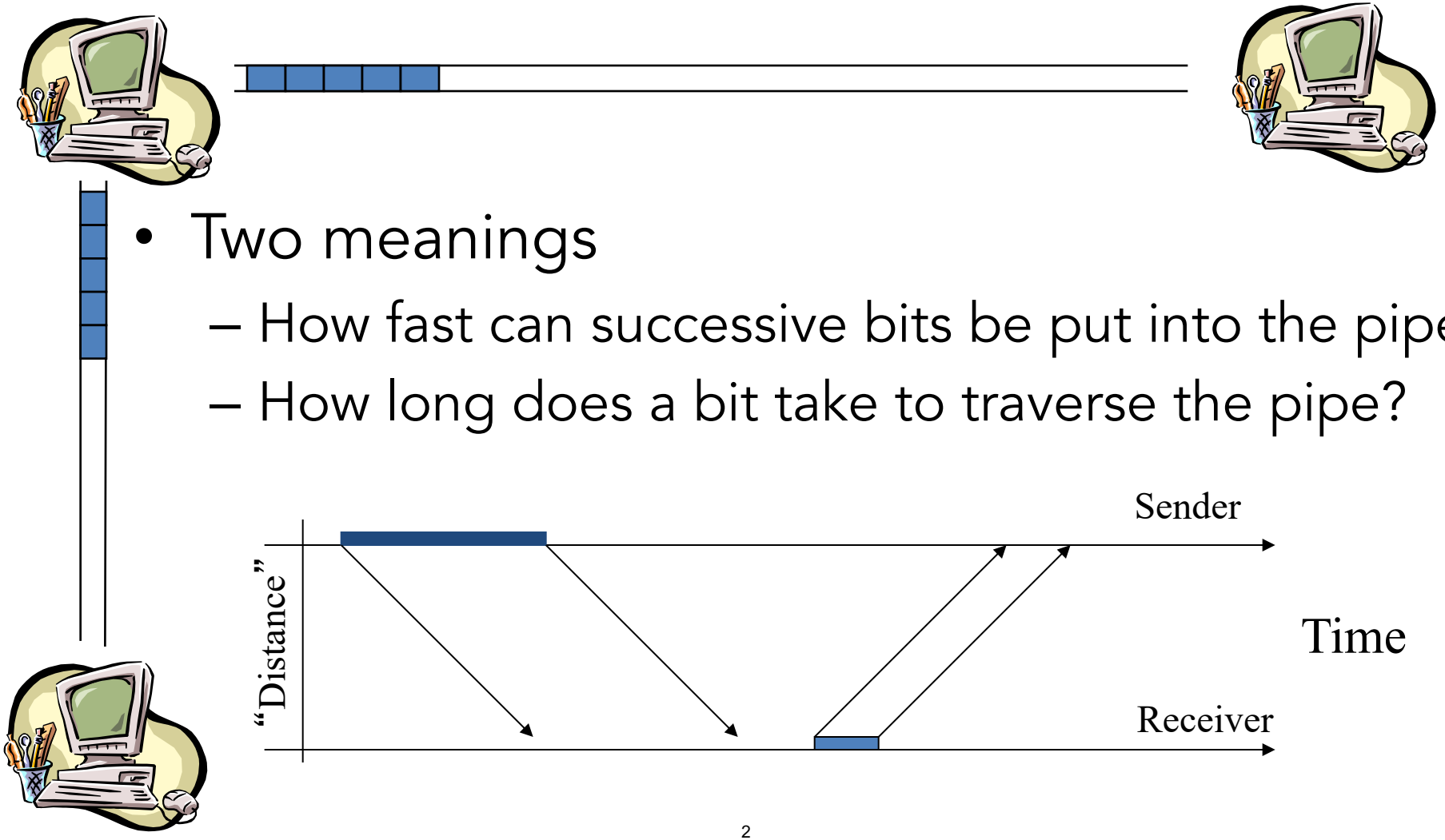
# CSC/ECE-570: Computer Networks

-- Network Simulation

Dr. Yuchen Liu

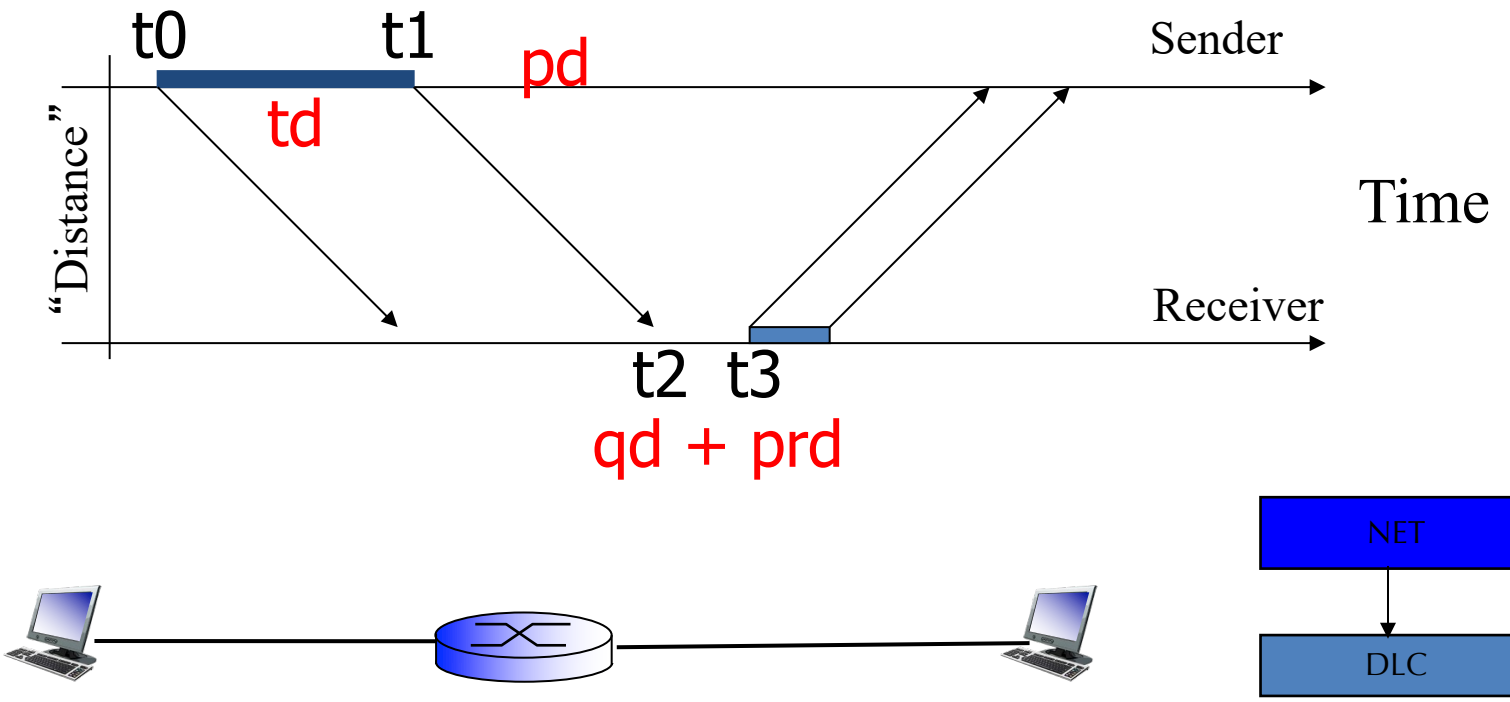
https://www.csc.ncsu.edu/people/yliu322

# Delay



- Two meanings
  - How fast can successive bits be put into the pipe?
  - How long does a bit take to traverse the pipe?

# Examples

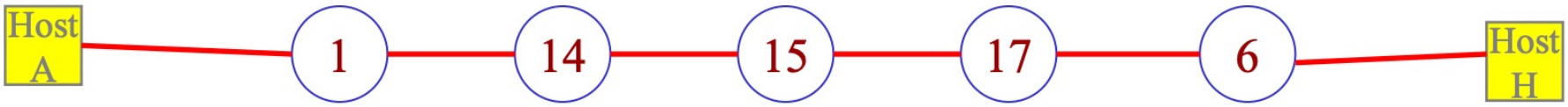



Processing delay: memory write/read 1) Delay from NET to shared memory;
2) Delay from shared memory to DLC.

# Reducing Delays: To Reduce –

- Propagation delay
  - Get the ends closer together.
  - Usually not at luxury of network designer
- Transmission delay
  - Use faster transmission equipment
  - Has to be at both ends of link, high link capacity
  - Might need to upgrade cable (or medium)
- Queuing delay
  - Use faster memory for buffers, faster buses, higher clock-speed
  - Use fewer stages of buffering
  - Reduce buffer size (but probability of loss increases)
- Processing delay
  - Use faster computing equipment
  - Do more of the processing in dedicated hardware rather than in custom software on general purpose hardware

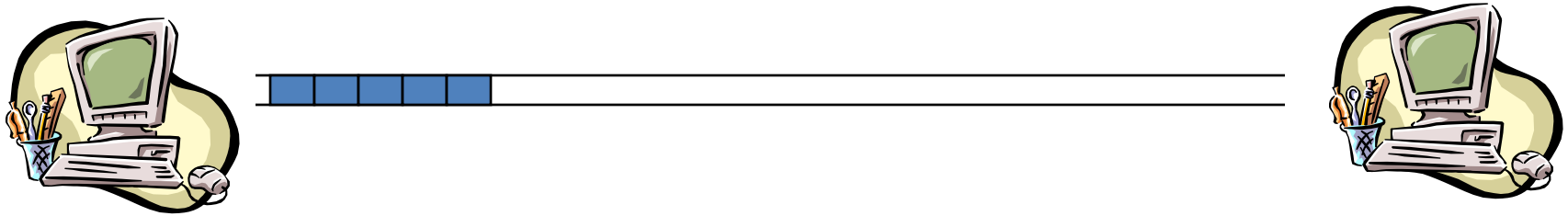
# Question




Assume: All links: 2.5 km; C = 100Mbps; propagation speed = 200m/microsec.
queuing delay = processing delay =0; packet size = 1000 bytes

What is the round-trip latency?

# Throughput



- Total number of bits transferred, over given time
  - Related obviously to the transmission delay
  - Throughput sometimes referred to as “bandwidth”
  - “Bandwidth-delay product (BDP)” - filling the pipe
- Conceptually simply – need to be able to measure
  - *ping* can measure round-trip delay
  - Various tools such as *Iperf* for throughput

# Network Simulation

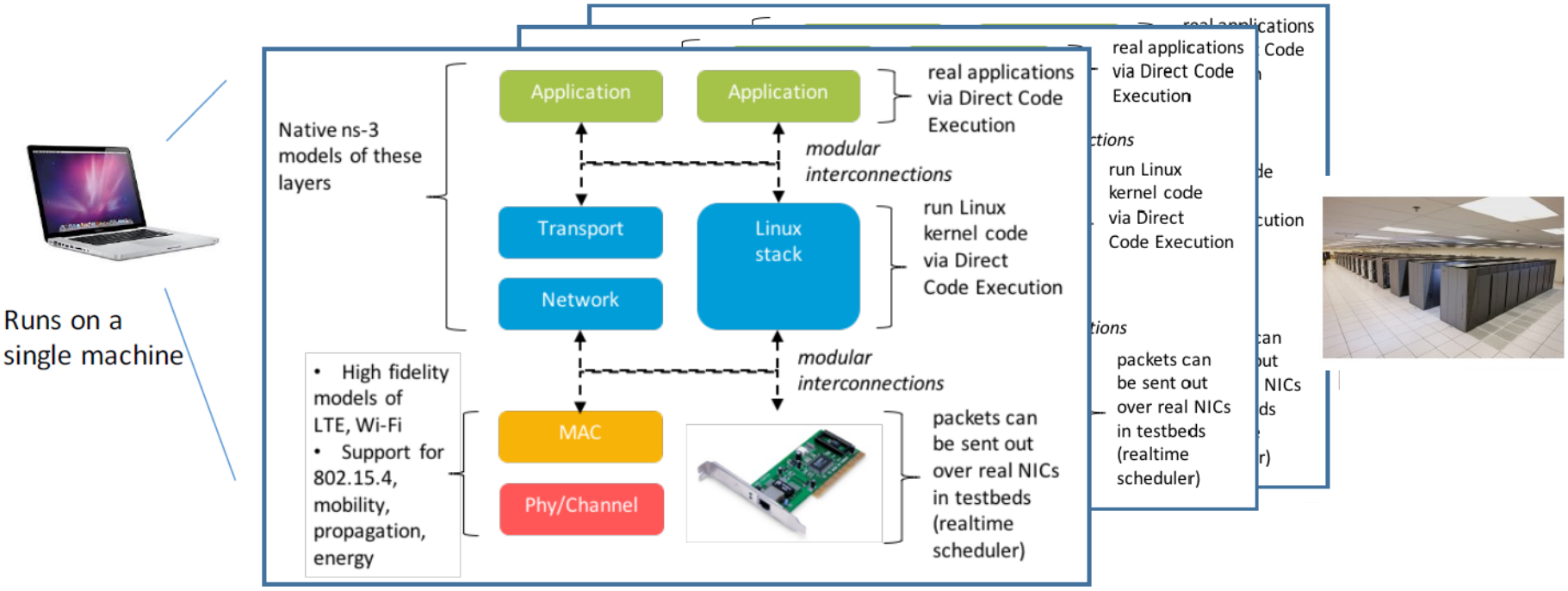
# Why Simulation?

- Investigate the new network behavior
- Predict/estimate the network performance
  - verify the design, modeling, protocols
  - another tool to study advanced computer networks

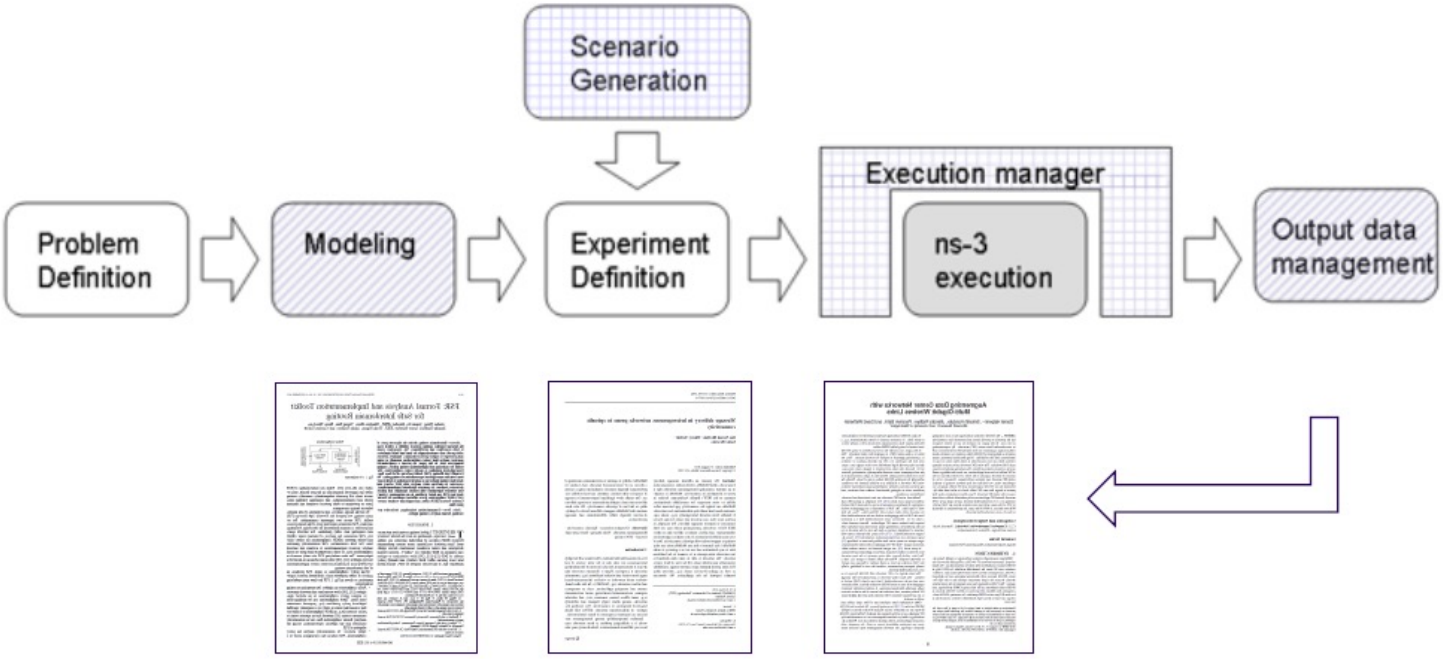
- Experiments vs. Simulations
  - commercial devices are not always feasible
  - easily adjust network configuration parameters
  - but accuracy and high-fidelity should be considered

# What is ns-3?

- ns-3 is a leading open source, packet-level network simulator oriented towards network research interaction with testbeds

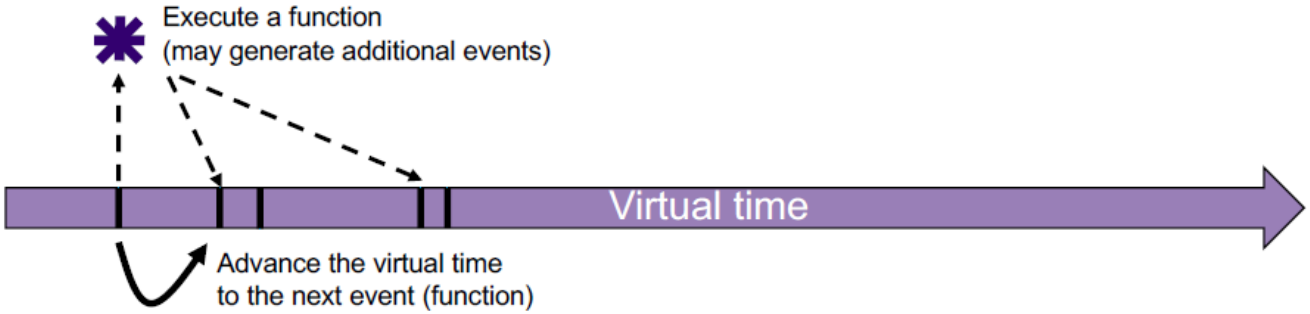


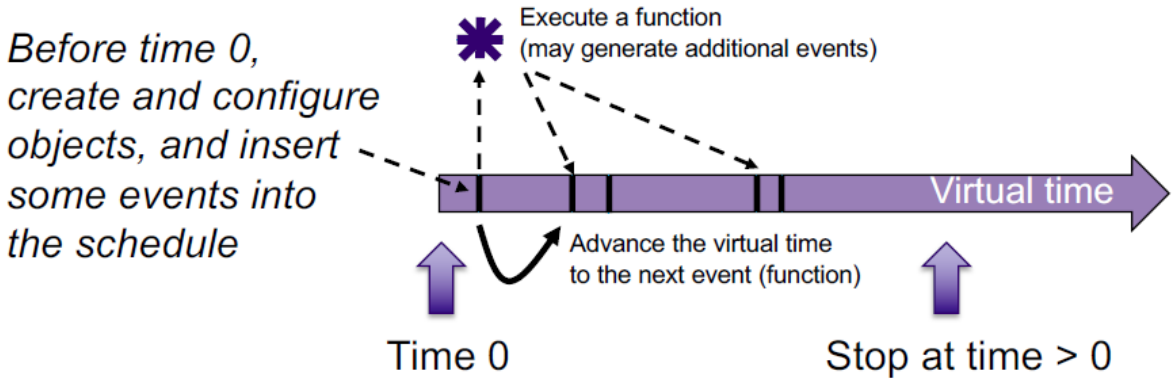
# The research workflow



# Simulation Details

- Discrete-event simulation basics
  - virtual time; events; scheduler; simulator (Run & Stop)



# ns-3 Details

- Written primarily in C++ (Python)
- Current version: ns-3.42
  - https://www.nsnam.org/
- Advanced features
  - wired network, wireless network, cellular, Wi-Fi, ML …
- ns-3 Appstore
  - https://apps.nsnam.org/
- Visualization – NetSimulyzer
  - https://apps.nsnam.org/app/netsimulyzer/

# Tutorial on ns-3

- Installing ns-3
  - On Linux and macOS, install natively
  - On Windows, use VirtualBox to run Ubuntu
- Dependencies: Install required packages
  - Follow https://www.nsnam.org/wiki/Installation#Ubuntu.2FDebian.2FMint
- Installation: Using ./ns3 -- a build automation tool

  ./ns3 configure --enable-tests --enable-examples

  ./ns3 build
  - Follow https://www.nsnam.org/wiki/Installation
- Validation: Run "./test.py" to validate install

# Documentation

- https://www.nsnam.org/releases/ns-3-37/documentation/

- For information on models use the Model Library
- For help on specific modules (APIs), use the ns-3 Doxygen documentation

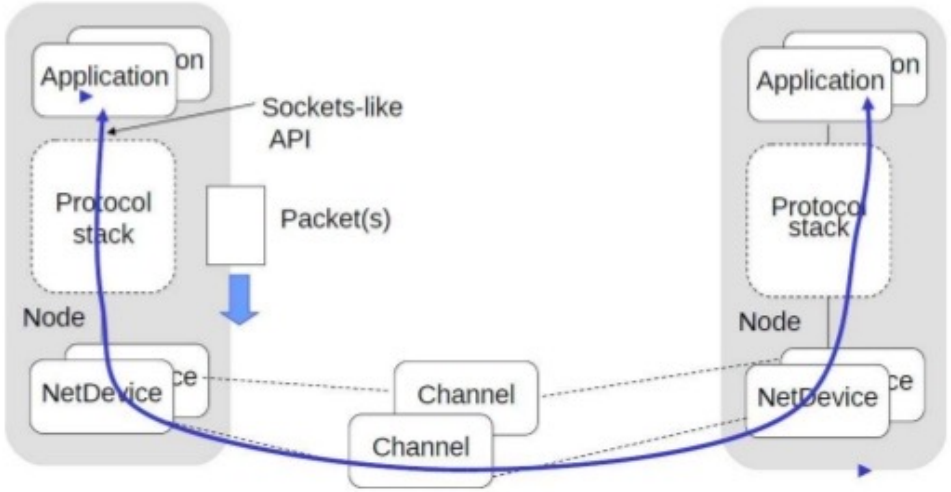
# Script

- Run the script
  - Start with an example script from 'examples/'
  - Copy the "*.cc" file into the folder named "scratch"
    (All simulations must be run from the scratch folder)
  - Run using ./ns3 run "<programName>"
    (Omit ".cc" from the program's name)

- See https://www.nsnam.org/support/faq/

# Simulate Networks

- Node: Basic computing device over which functionalities can be added (protocol stacks, applications, etc)
- Channel: A model of the communication subnetwork connecting the nodes (eg: WifiChannel, p2p)
- Net device: A model of the Network Interface Card (NIC)
  - Hardware and software
  - Enables a "Node" to communicate through a "Channel" (eg: WifiNetDevice)
- Application: Generates the activities on the node to be simulated (eg: a UDP server)



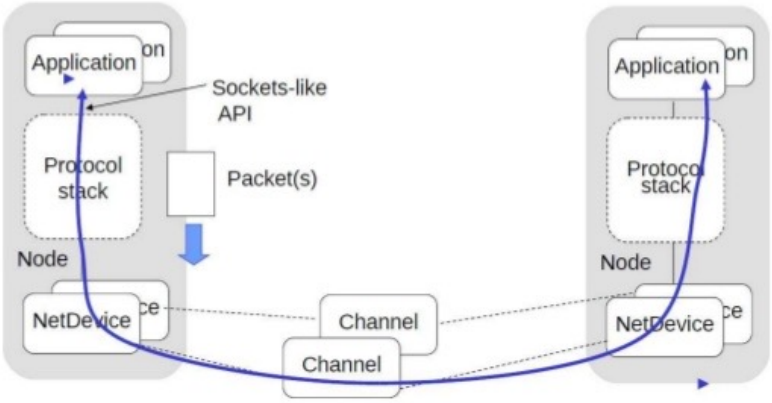
# Setting up a Simulation

To create and simulate a functioning network in ns-3, we should create/install:

- All nodes, topology
- Net Devices for each node
- Channels
- Protocol stacks on each node
- Applications on each node

In addition, we should

- Connect channels to Net devices,
- Configure each Net device, channel, application, protocol stack, node,
- Etc.



# ns-3 Helpers

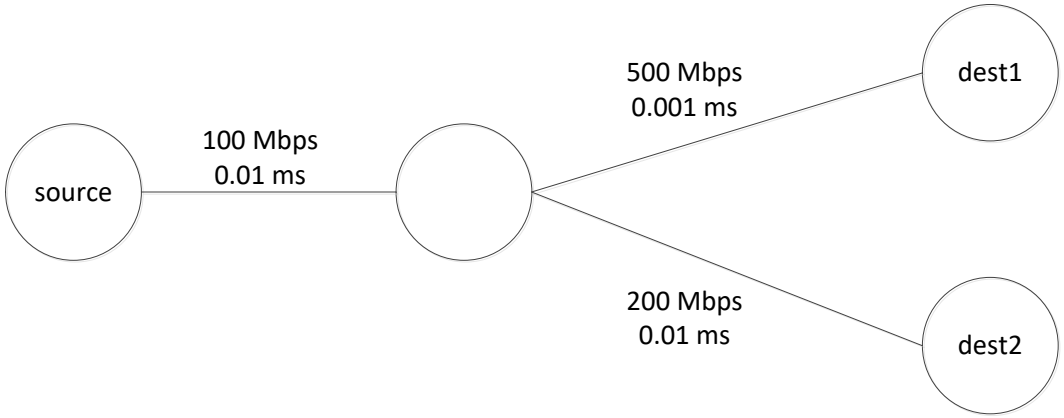
- NodeContainer: Create and manage the nodes
- PointToPointHelper: Create and manage P2P channel between two nodes
- CsmaHelper:  Create and manage CSMA channels
- YansWifiChannelHelper: Create and manage Wi-Fi channels
- NetDeviceContainer: Create and manage net devices
- InternetStackHelper: Create and manage Internet (TCP/IP) stack
- IPv4AdressHelper: assign the  IP addresses

…

# Example – Networking

- Run the script "first.cc"
  - Start with an example script from 'examples/'
  - Copy the "*.cc" file into the folder named "scratch"
    (All simulations must be run from the scratch folder)
  - Run using ./ns3 run "myfirst"
    (Omit ".cc" from the program's name)

# Example – More Links




```
NetDeviceContainer devices;
devices = pointToPoint.Install (P2Pnodes.Get(0), P2Pnodes.Get(1));
```

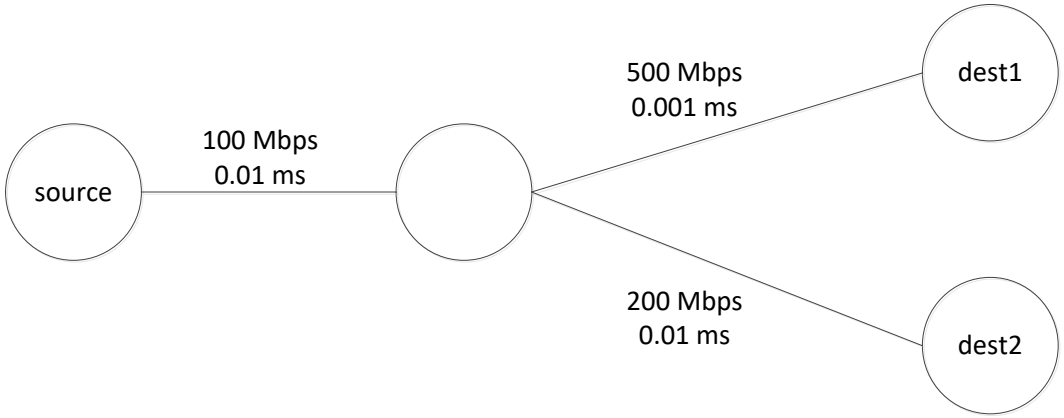
NodeDeviceContainer device1, device2, device3;

device1 = p2p1.install (P2Pnodes.Get(0), P2Pnodes.Get(1));

device2 = p2p2.install (P2Pnodes.Get(1), P2Pnodes.Get(2));

…

# Example – More Links




```
Ipv4AddressHelper address;
address.SetBase ("10.1.1.0", "255.255.255.0");

Ipv4InterfaceContainer interfaces = address.Assign (devices);
```

Ipv4Interface interface1; interface2, interface3;
interface 1 = address.assign (device1);

address.newNetwork ();

interface 2 = address.assign (device2);
…

# Example – Networking

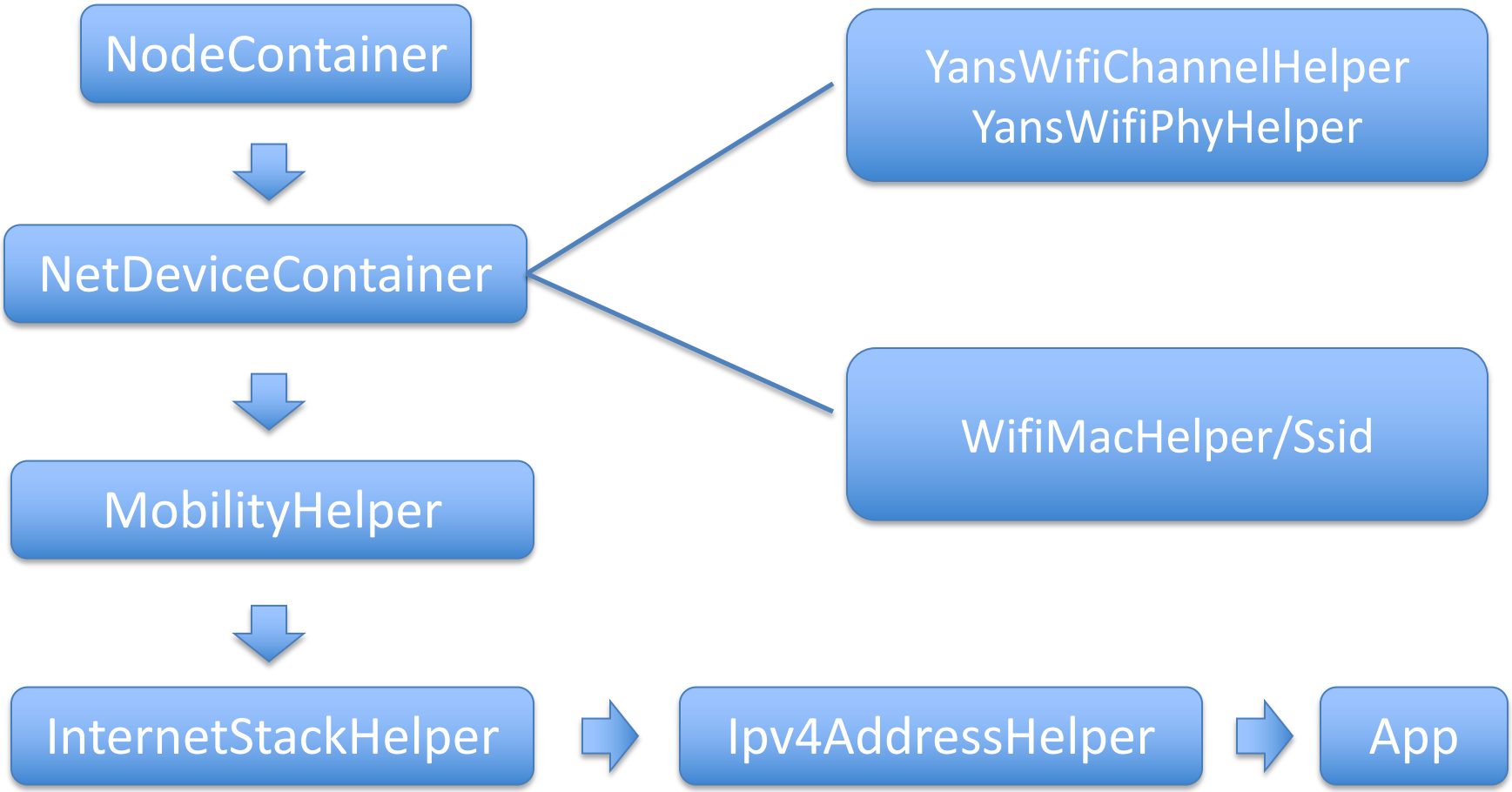
- Run the script "third.cc"
  - Start with an example script from 'examples/'
  - Copy the "*.cc" file into the folder named "scratch"
    (All simulations must be run from the scratch folder)
  - Run using ./ns3 run "third"
    (Omit ".cc" from the program's name)

- subnet example

  third.cc

  command-line parameters (e.g., --nCsma=6)

# Wi-Fi Setting



# What if Python?

https://github.com/Gabrielcarvfer/ns3_for_education

Colab

# Resources

For troubleshooting and tutorials

- The official ns-3 tutorial: https://www.nsnam.org/docs/tutorial/html/index.html
- Tutorial on C++: http://www.cplusplus.com/doc/tutorial/
- ns-3 Google groups: https://groups.google.com/g/ns-3-users
- ns-3 documentation: https://www.nsnam.org/doxygen/index.html
- ns-3 Wiki page: https://www.nsnam.org/wiki/
- ns-3 support page: https://www.nsnam.org/support/
- ns-3 App Store: https://apps.nsnam.org/

# Next

- Physical Layers