

# Recitation 1

Alireza Naghizadeh

# Click Router

# Router

A packet-processing network consists of routers and hosts.

Hosts use packets as a means to an end; they are mostly concerned with providing communication abstractions to applications.

Routers, however, are pure packet processing applications, e.g., they provide an abstraction for applications.

They are interested only in packets, which they route from place to place based on packet header information.

There are many other packet processing applications such as firewalls, NATs, packet balancers etc.

# What is a 'Software Router'?

Router in a PC

Program that receives, processes and forwards packets to the next node

# Click Modular Router

Extensible toolkit for writing packet processors

PhD thesis Dr. Eddie Kohler (MIT) Architecture centered on elements:

Small building blocks

- Perform simple operations e.g. decrease TTL

- Add/remove components e.g. connect queues

The whole Click router itself is a Linux kernel module!

- Directed graphs of elements

- Elements are written in C++

# Why Click?

## Modular framework

Packets are intercepted with Click Toolkit

e.g. `FromDevice(eth0) -> Queue() -> ToDevice(eth1);`

## Building blocks:

Provides a number of prewritten routing elements like Queues, Shapers, packet senders etc.

You can write your own elements!

## High level programming interface

Hides Kernel intricacies

Easy to install, modify and use

# Prerequisites

Just a few requirements

- C++ programming

- Language syntax

- Object oriented programming

Compile a program

Linux command line

- Basic knowledge (e.g. run programs)

- Basic networking tools (e.g. ifconfig)

Networks Theory

# Intro to Our VM

Download from:

<http://www.winlab.rutgers.edu/comnet2/Projects/downloads/comnetsII.ova>

VM OVS format is compatible with most virtual environments

Suggestion: Virtual Box from Oracle. It is free, multiplatform

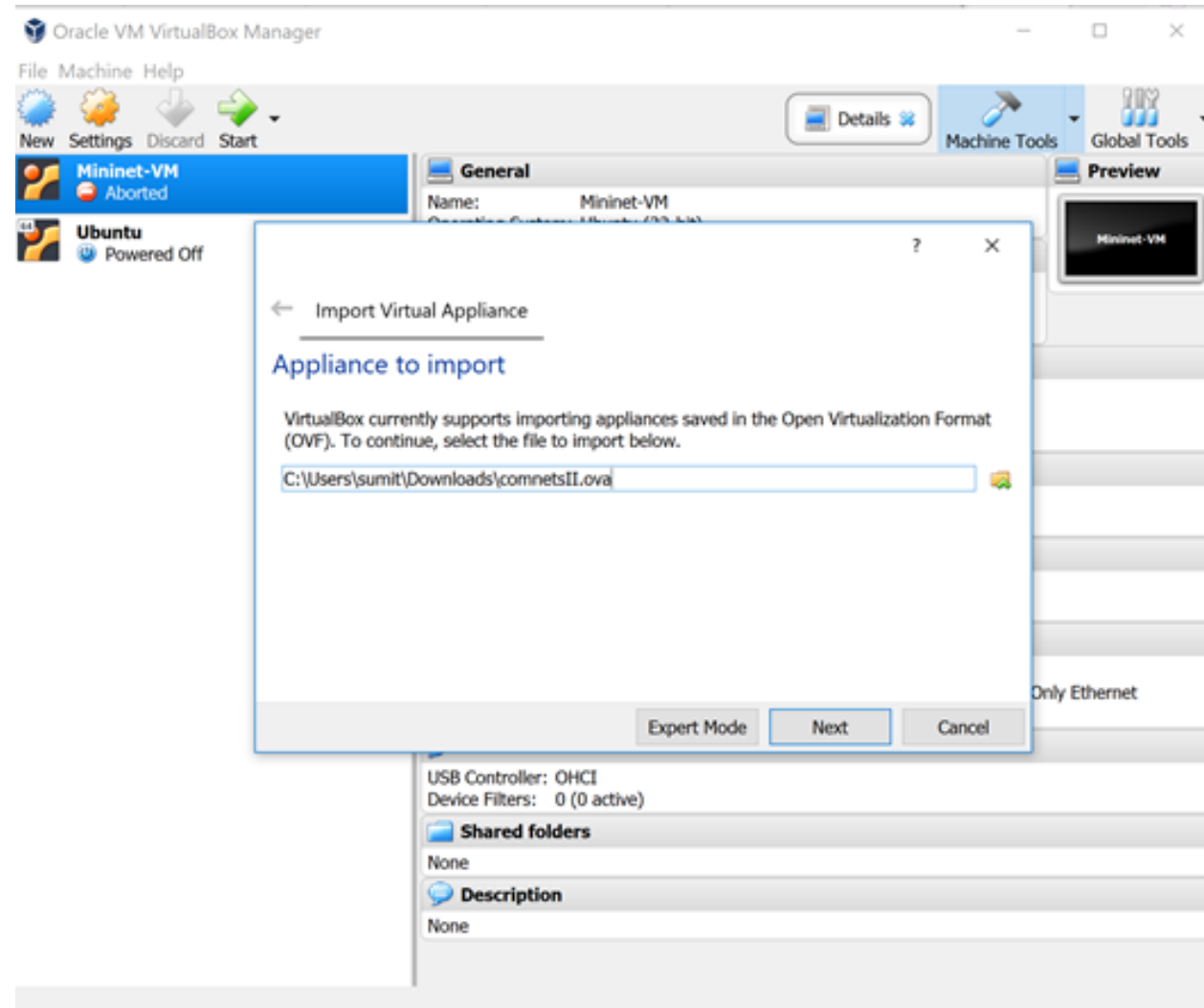
<https://www.virtualbox.org/wiki/Downloads>

The following instructions are based on Virtual Box



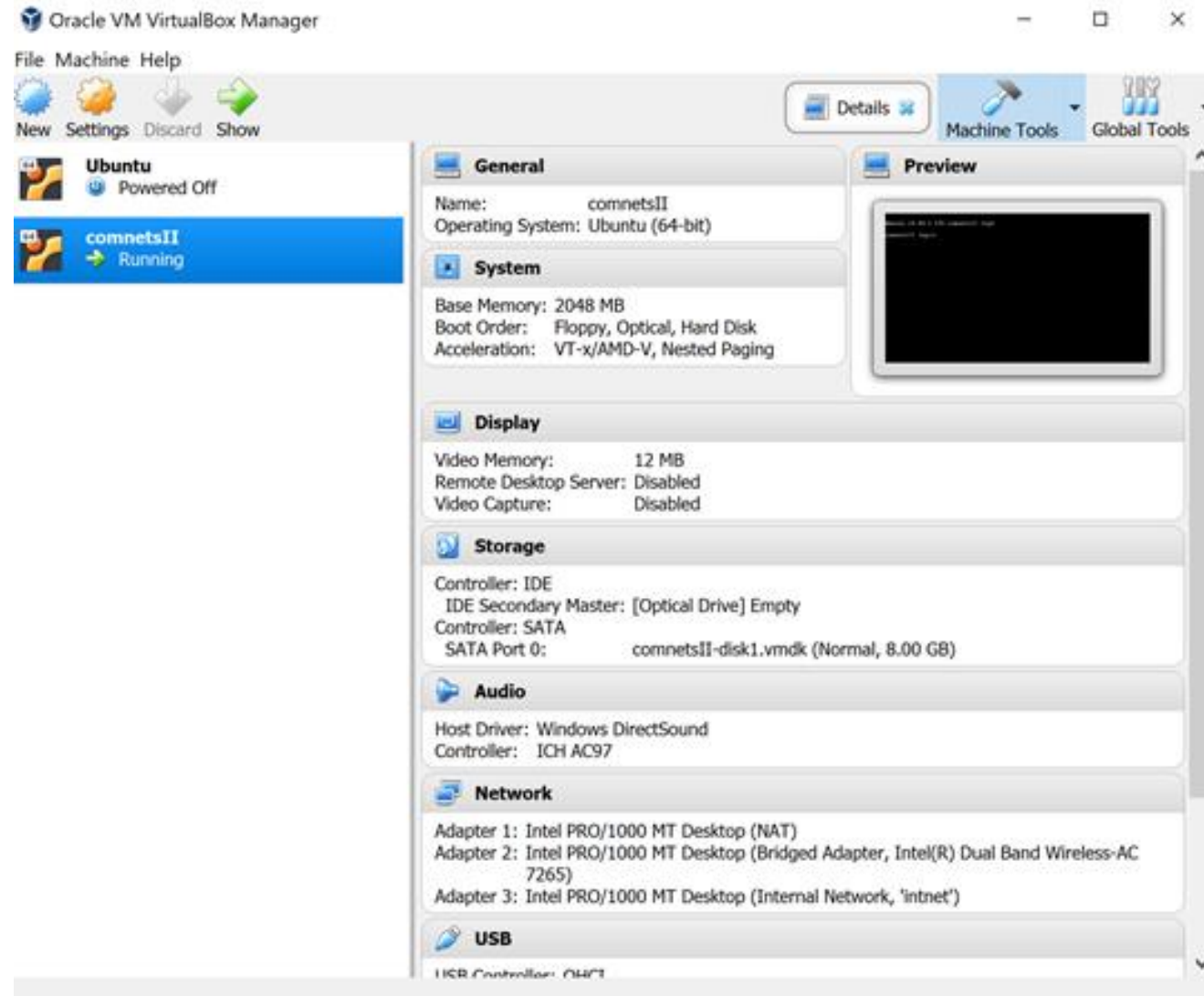
# Loading the VM into VBox

File -> Import Appliance



# SSH into the VM

Start the VM



## SSH into the VM

Right click on the vm, go to settings and then Network. We can create three types of networks as follows:

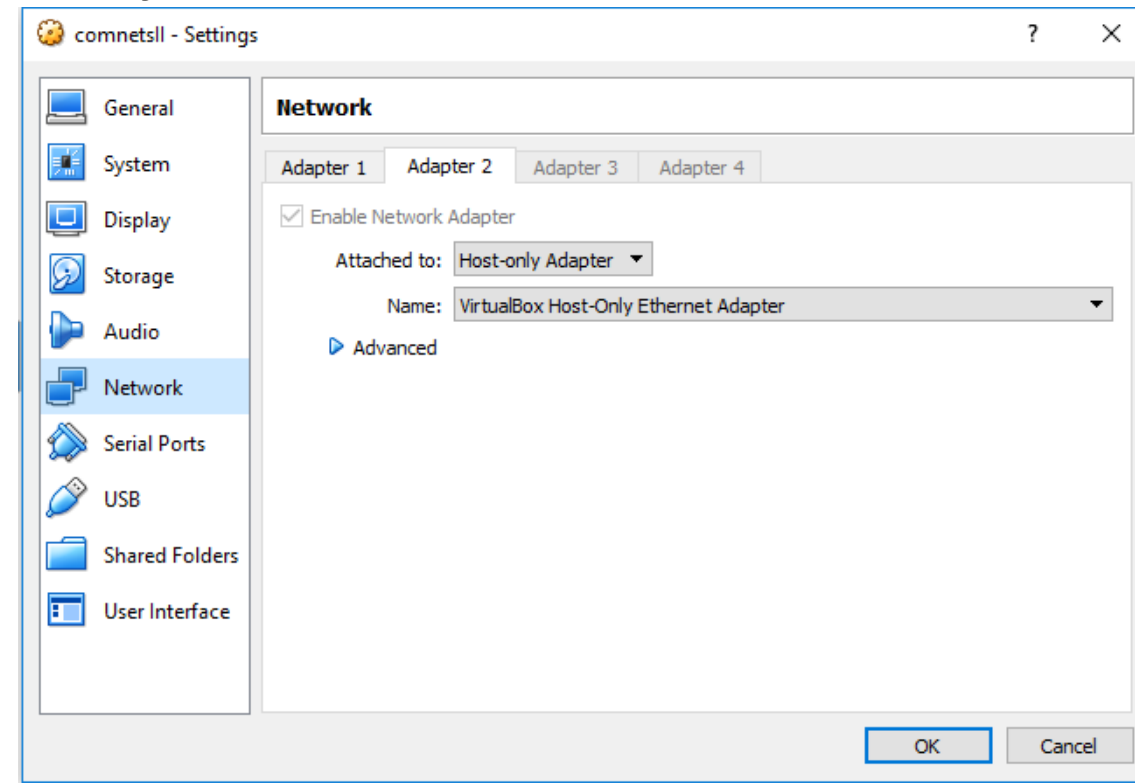
**Bridged Networking:** Bridged networking connects a virtual machine to a network by using the network adapter on the host system. If the host system is on a network, bridged networking is often the easiest way to give the virtual machine access to that network.

**NAT Networking:** With NAT, a virtual machine does not have its own IP address on the external network. Instead, a separate private network is set up on the host system. In the default configuration, a virtual machine gets an address on this private network from the virtual DHCP server. The virtual machine and the host system share a single network identity that is not visible on the external network.

# SSH into the VM

**Host-Only Networking:** Host-only networking creates a network that is completely contained within the host computer. Host-only networking provides a network connection between the virtual machine and the host system by using a virtual network adapter that is visible on the host operating system.

Make sure that you have two adapter for NAT and Host-Only

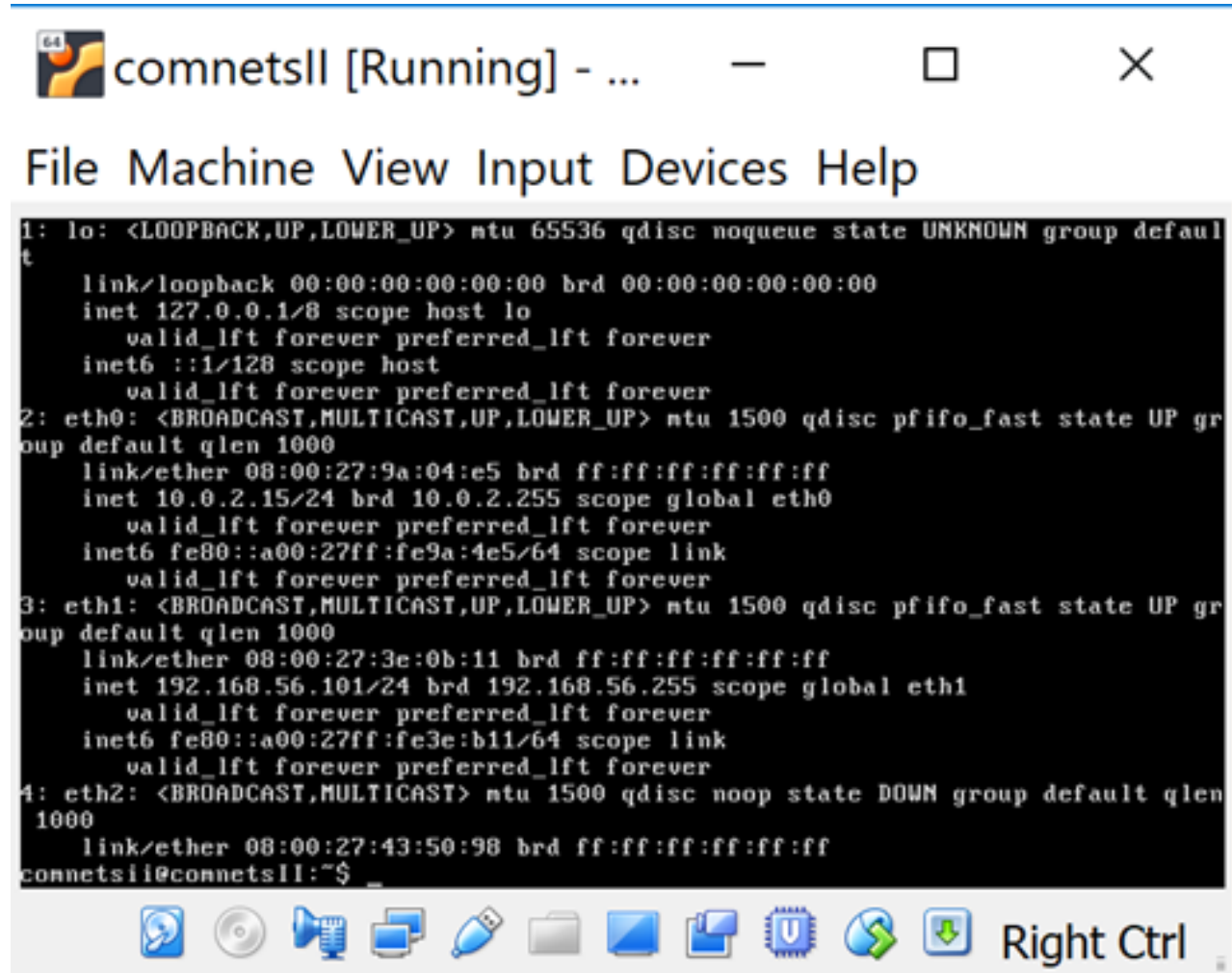


# SSH into the VM

Login: username=comnetsii  
Passowrd=comnetsii

# SSH into the VM

```
$sudo ifconfig eth1 up
$sudo ifconfig eth1 192.168.56.101
```



```
comnetsII [Running] - ...
File Machine View Input Devices Help

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:9a:04:e5 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe9a:4e5/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:3e:0b:11 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe3e:b11/64 scope link
        valid_lft forever preferred_lft forever
4: eth2: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 08:00:27:43:50:98 brd ff:ff:ff:ff:ff:ff
comnetsII@comnetsII:~$
```

# SSH into the VM

You can now ssh into the VM (while it is running), from your host OS

On Unix machines:

```
$ssh comnetsii@192.168.56.101
```

On windows:

Use tools such as MobaXterm or Putty

# Available Resources

In the home folder you will find the following resources:

- click: click sources and resources

- examples: small set of examples that can be used as reference

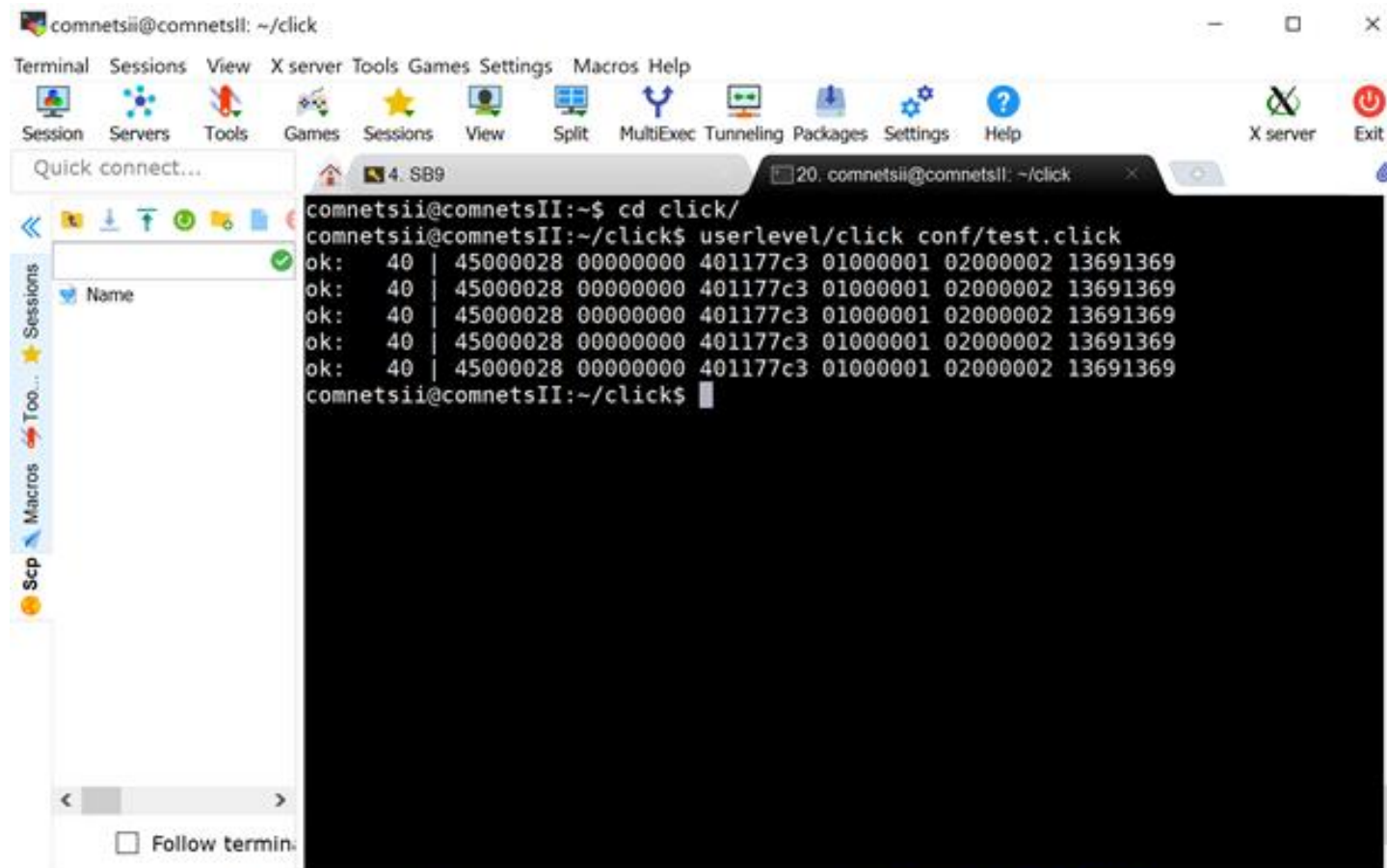
- elements: a few elements are provided to be used in future excercises

- tools: script utilities used to model our virtual environment (more on this later..)



# How to Run a Click Instance

Run you first click instance (inside the click folder):  
\$userlevel/click conf/test.click



The screenshot shows a MobaXterm terminal window with the following content:

```
comnetsii@comnetsII: ~/click
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
4. SB9 20. comnetsii@comnetsII: ~/click
comnetsii@comnetsII:~$ cd click/
comnetsii@comnetsII:~/click$ userlevel/click conf/test.click
ok: 40 | 45000028 00000000 401177c3 01000001 02000002 13691369
ok: 40 | 45000028 00000000 401177c3 01000001 02000002 13691369
ok: 40 | 45000028 00000000 401177c3 01000001 02000002 13691369
ok: 40 | 45000028 00000000 401177c3 01000001 02000002 13691369
ok: 40 | 45000028 00000000 401177c3 01000001 02000002 13691369
comnetsii@comnetsII:~/click$
```

# Click Routers: Main Concepts

Elements

Ports

Packets

Configuration

More...

# Click Routers: Main Concepts

Router: **Elements** connected by **edges**

Output ports to input ports

Describes possible **packet flows** through directed graphs

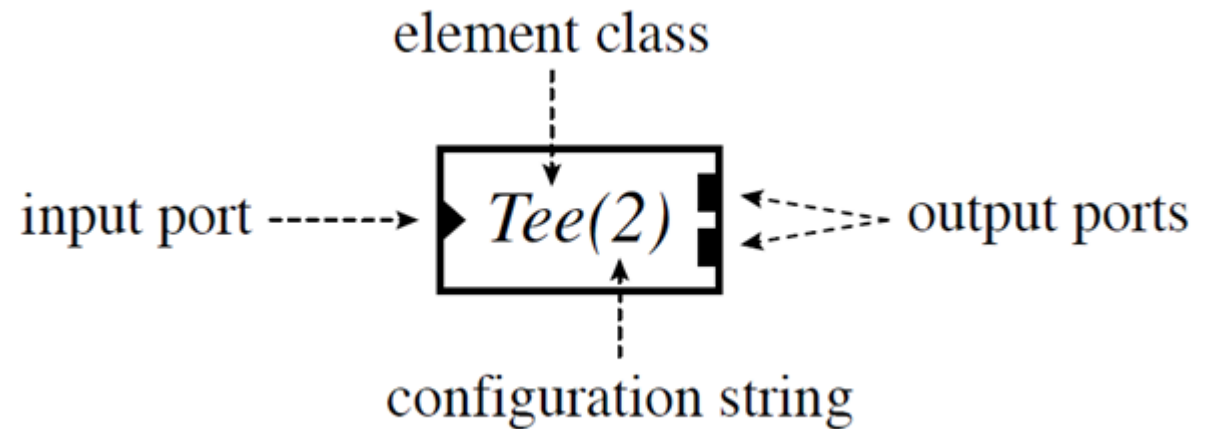


# Elements

Most important user-visible abstraction in Click

Elements (they are C++ classes)

Element instances: C++ objects

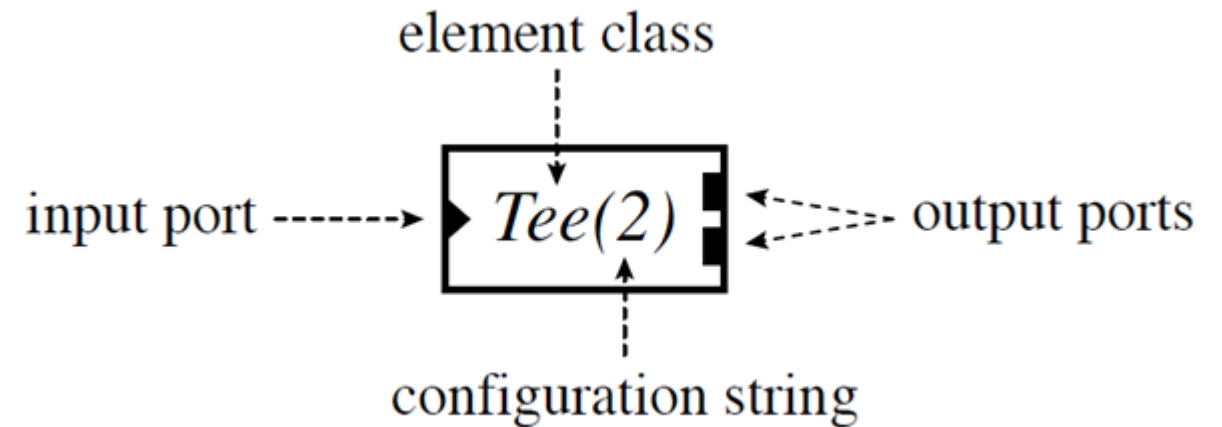


# Elements

Input port(s): Interface where packets arrive, triangles

Output port(s): Interface where packets leave

Inside: packet processing!



# Ports

Push port:

- Filled square or triangle

- Source initiates packet transfer: event based packet flow

Pull port:

- Empty square or triangle

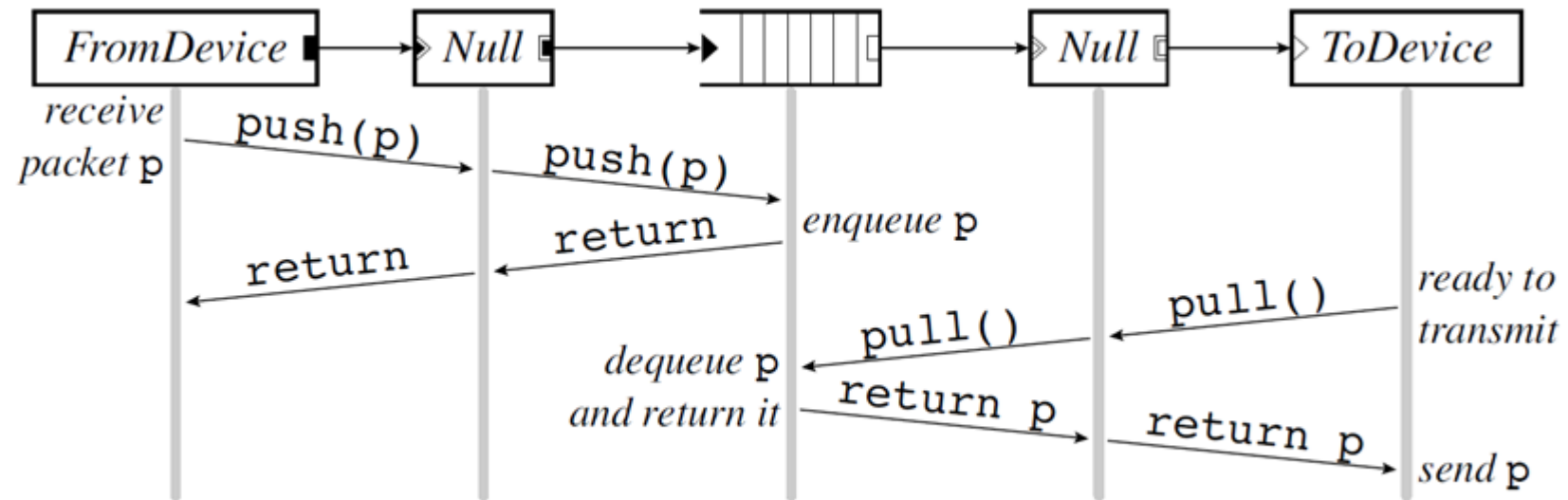
- Destination initiates packet transfer: Used with polling, scheduling etc.

Agnostic port:

- Square-in-square or triangle-in-triangle

- Becomes push or pull (inner square/triangle filled or empty)

# Ports



# Push-Pull Violations

## Push port

has to be connected to push or agnostic port

Conversion from push to pull with push-to-pull element

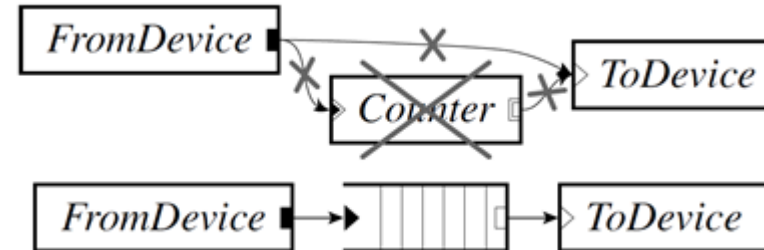
E.g. queue

## Pull port

Has to be connected to pull or agnostic port

Conversion from pull to push with pull-to-push element

E.g. unqueue





# Project 1 Goals

Get familiar with Click's environment

Get familiar with our virtualized environment

Practice with existing elements and create first running configuration

Use existing interfaces to communicate between multiple click instances

# The project

1. Single Click instance
2. Create a new packet with payload “hello”
3. Print its content to terminal

Drop the packet

Hints:

Only use of existing elements

Configuration should use 3 elements

Click provides a collection of “source” elements

use these urls for reference information:

<https://github.com/kohler/click/wiki/Elements>

<https://web.archive.org/web/20170720053721/http://www.read.cs.ucla.edu/click/elements/>