RUTGERS
THE STATE UNIVERSITY
OF NEW JERSEY

Project 2

Alireza Naghizadeh

Project 2

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.

Click Routers: Main Concepts

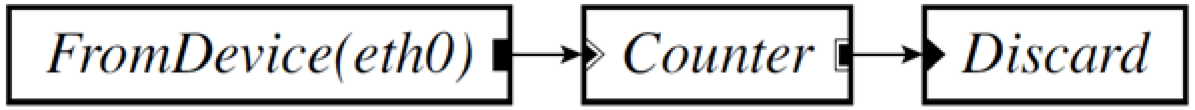
- Elements
- Ports
- Packets
- Configuration
- More…

Click Routers: Main Concepts

Router: Elements connected by edges

Output ports to input ports

Describes possible packet flows through directed graphs

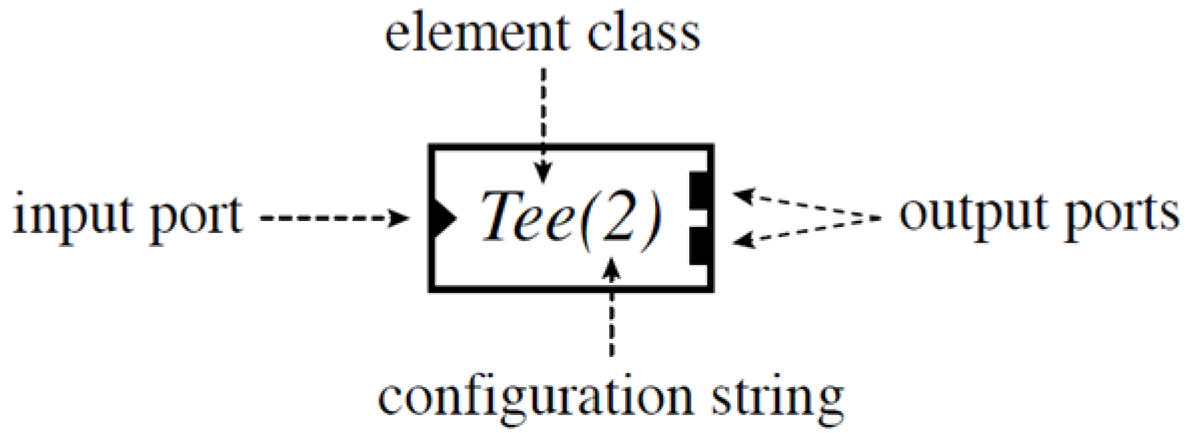


Elements

Most important user-visible abstraction in Click

Elements (they are C++ classes)

Element instances: C++ objects

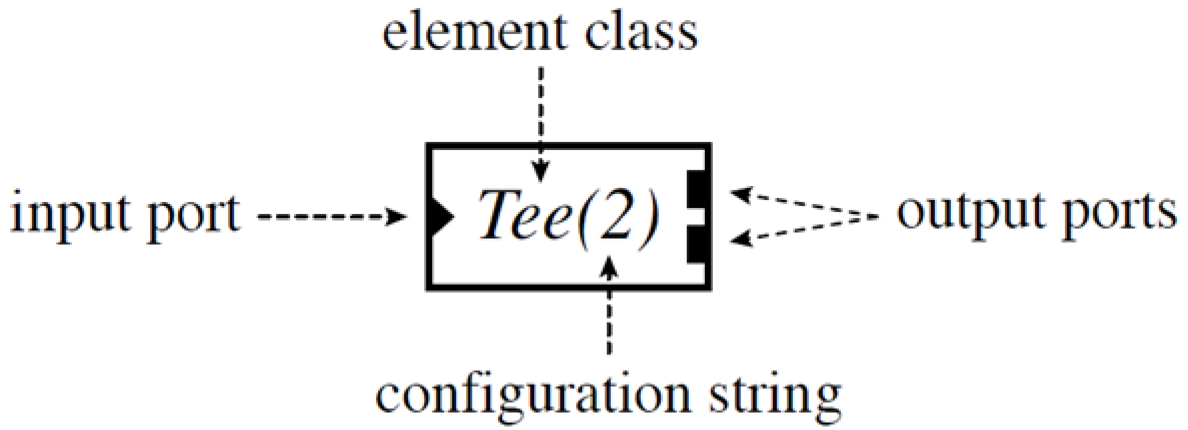


Elements

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

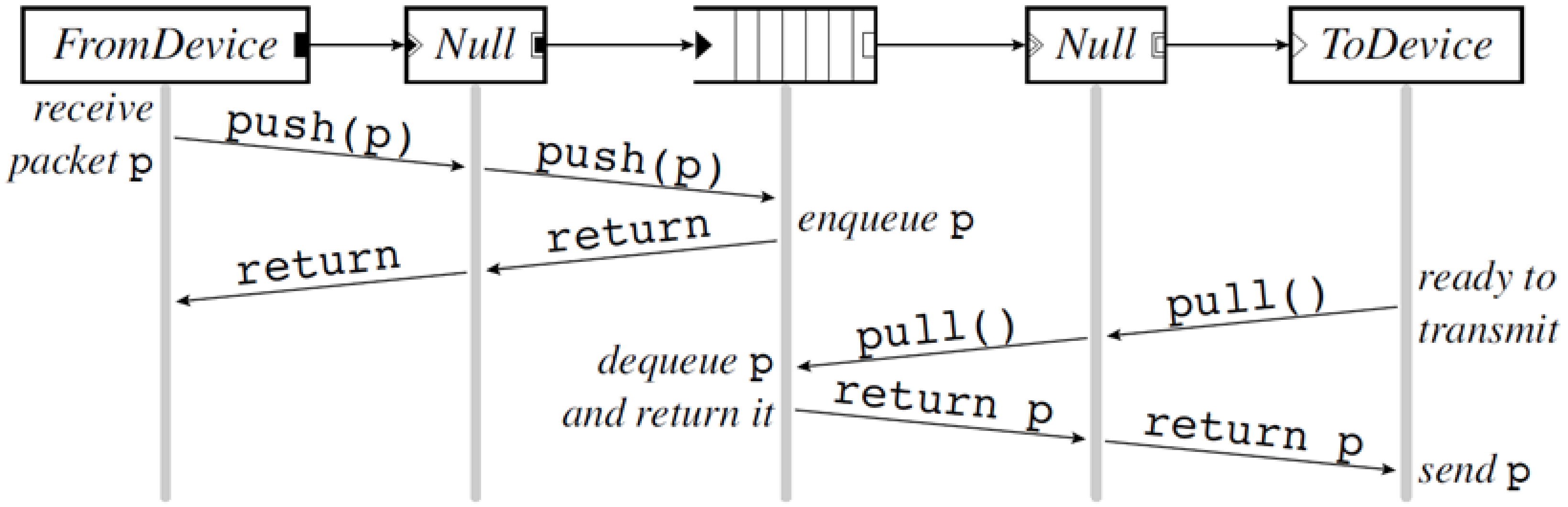
Pull port:

- Empty square or triangle
- Destination initiates packet transfer: Used with polling, scheduling etc.

Agnostic port:

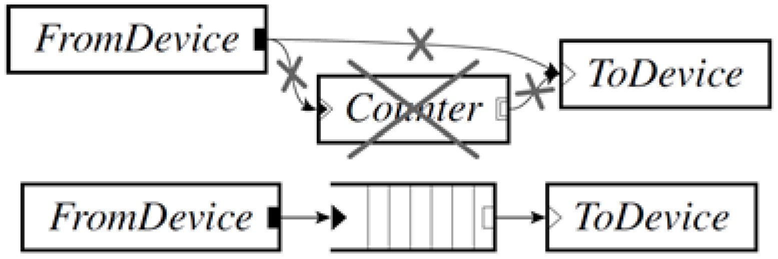
- Square-in-square or triangle-in-triangle
- Becomes push or pull (inner square/triangle filled or empty)

Ports


FromDevice
Null
Null
ToDevice
receive packet p
push(p)
push(p)
enqueue p
return
return
ready to transmit
pull()
pull()
dequeue p and return it
return p
return p
send p

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



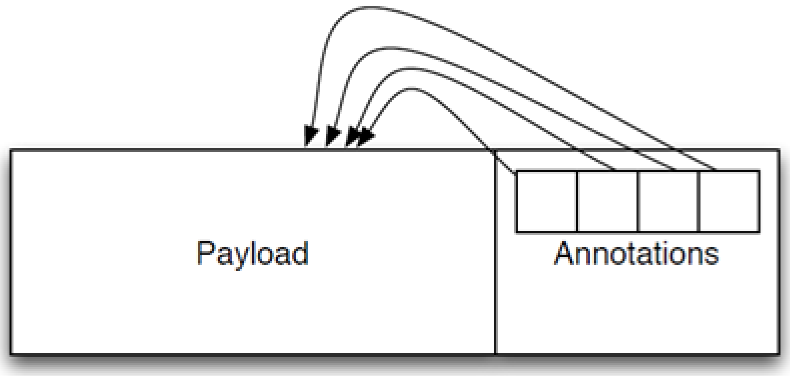
Packets

Packet consists of payload and annotations:

Payload: is the data

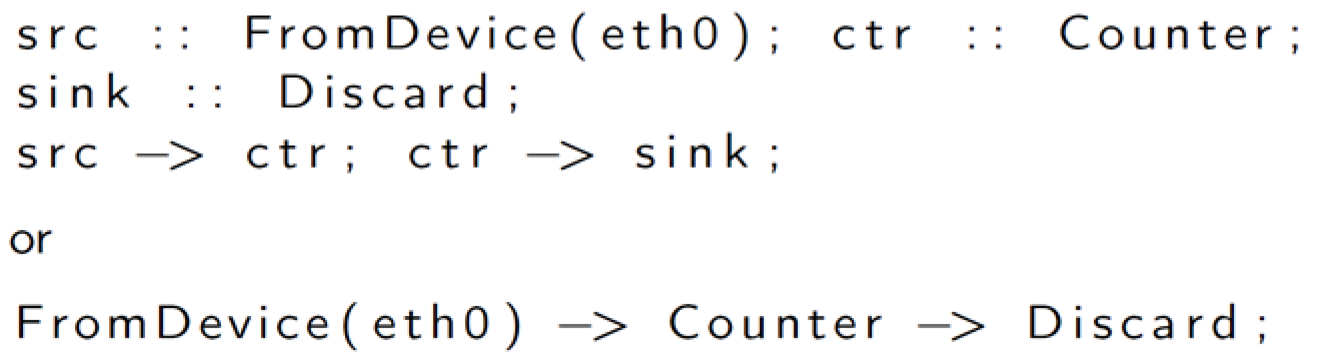
Annotations: parts give the router enough information to peel the layers off, most of them are fixed and should be determined by standard protocols.

- Destination IP address
- Paint
- Fix IP Source flag
- etc.



Intro to Configurations

Text files describing the Click graph:
- Generally has .click filetype extension
- Elements with their configurations
- Connections between elements
- Flexible syntax



```
src :: FromDevice(eth0); ctr :: Counter;
sink :: Discard;
src -> ctr; ctr -> sink;
```

or

```
FromDevice(eth0) -> Counter -> Discard;
```

Intro to Configurations

Identified by number (0,1,..)

Input port: -> [nr1]Element ->

Output port: -> Element[nr2] ->

Both: ->[nr1]Element[nr2]->

In case we have only one port, the number can be omitted

```
mypackets::IPClassifier(dst host $myaddr,-);
FromDevice(eth0) -> mypackets;
mypackets[0]-> Print(mine)->[0]Discard;
mypackets[1]-> Print("the others") -> Discard;
```

Element Configuration Example

```
SimpleElement(“data”)

SimpleElement(“data”,ACTIVE false)

SimpleElement(“moredata”,800)

SimpleElement(“data”,800,DATASIZE 67,SOURCE 1.2.3.4)
```

Interaction Between Multiple Instances: Interfaces Use

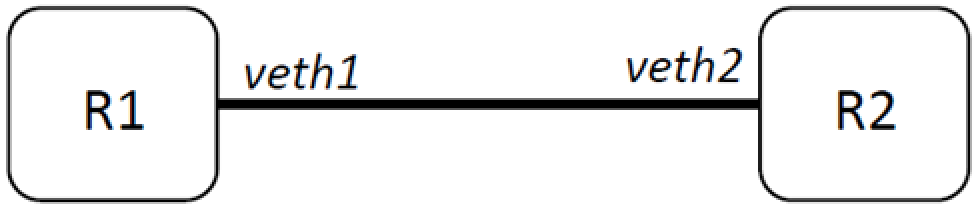
Click provides elements used to interact with the system network interfaces

- FromDevice: read packets from the device
  - Element with one outgoing push port
- ToDevice: write packets into the device
  - Element with one incoming pull port
- Multiple click instances can interact using these elements
- Which interfaces to use?
  - We will provide scripts to simplify your life
  - Feel free to explore what these scripts do

Project Part 1:

Use provided script to create 2 virtual interfaces (check tools directory)

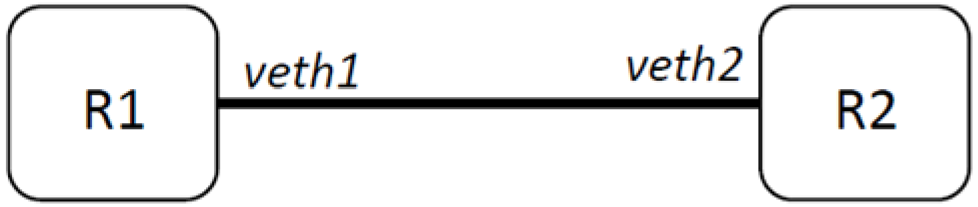
- Run: $ sudo createNet1
  - The script will create the virtual interfaces veth1 and veth2
- Run: $ifconfig or $ip a
  - You should see a list of available interfaces with their IP and MAC addresses
- Obtained Topology



RUTGERS

Project Part 1:

With the given topology



1- One click instance: generates packets and transmits them into the device

2- Second click instance: reads the packets and prints them

Hints:

Ethernet and ip encapsulation?
Encapsulation elements are available and ready to be used
Did you try $sudo?
Start receiver click first!

RUTGERS

Project Part 2:

For Part1 we only implemented a generator and a sink

Normally a router processes packets and forwards them.

We want to experiment on multi-directional communications.

Same as Part 2, but the message has to be echoed back to the origin router.

If you need to recreate the network, use the same script from part1.

Final Notes

Use Piazza for questions, to make things in order add your questions in hw2 section.

We will go through these slides on Wed.

I will have an office hour next week Monday 4:30 to 5:30, Room 103C, Tillet Hall

Submission deadline: 7/20 via Sakai

Submission should contain only the click configuration files. If you want to include additional information, write a README file.

Do not include the whole click resources!