

Computer Networks

A Simple Router Implementation

Tutorial #5
Fall 2025



Simple Router using Mininet

Preliminars

- You will need Mininet + OpenFlow + POX
 - Use the latest VM image that can be found
<https://cmsweb.utsc.utoronto.ca/marcelo-ponce/CSCD58/Mininet-VM-SR.zip>
 - It also includes the `cs144_lab3` git-repository.
 - You still need to checkout the “standalone” branch,
i.e.
`git checkout --track origin/master/standalone`

Overview -- Goals

- Implement/write a “simplified” router
 - Given a static network topology
 - Given a static routing table
 - **You are responsible for** writing the logic to handle incoming *Ethernet frames* (ICMP, ARP, IP....):
 - Forward it
 - Generate ICMP messages
 - Drop it
 - and more...

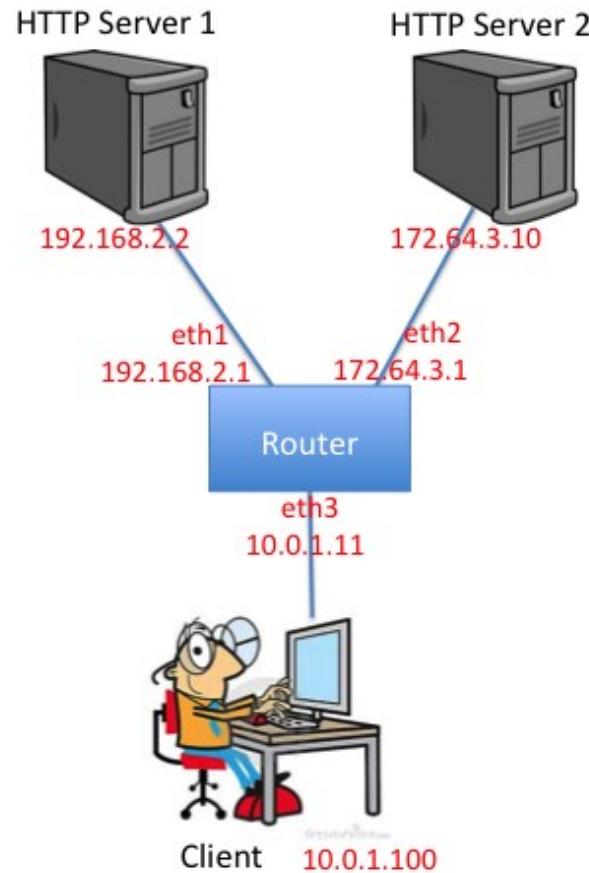


Emulated Topology

- No hardware router
- Network topology emulated with Mininet:
- your router connects 2 servers to a client
 - Your router will handle real traffic



Emulated Topology



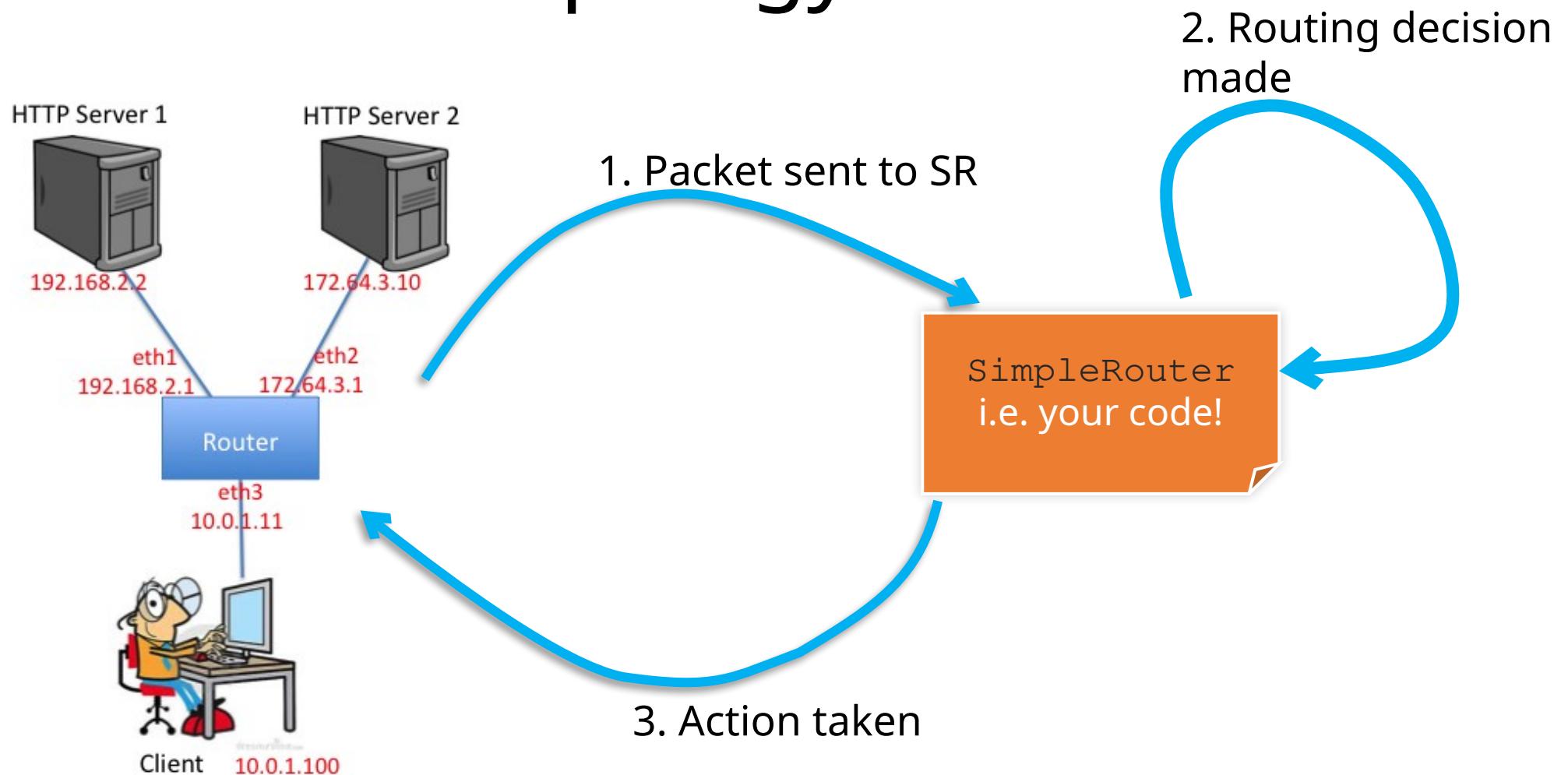
```
# IP_CONFIG
```

```
server1 192.168.2.2
server2 172.64.3.10
client 10.0.1.100
sw0-eth1 192.168.2.1
sw0-eth2 172.64.3.1
sw0-eth3 10.0.1.1
```

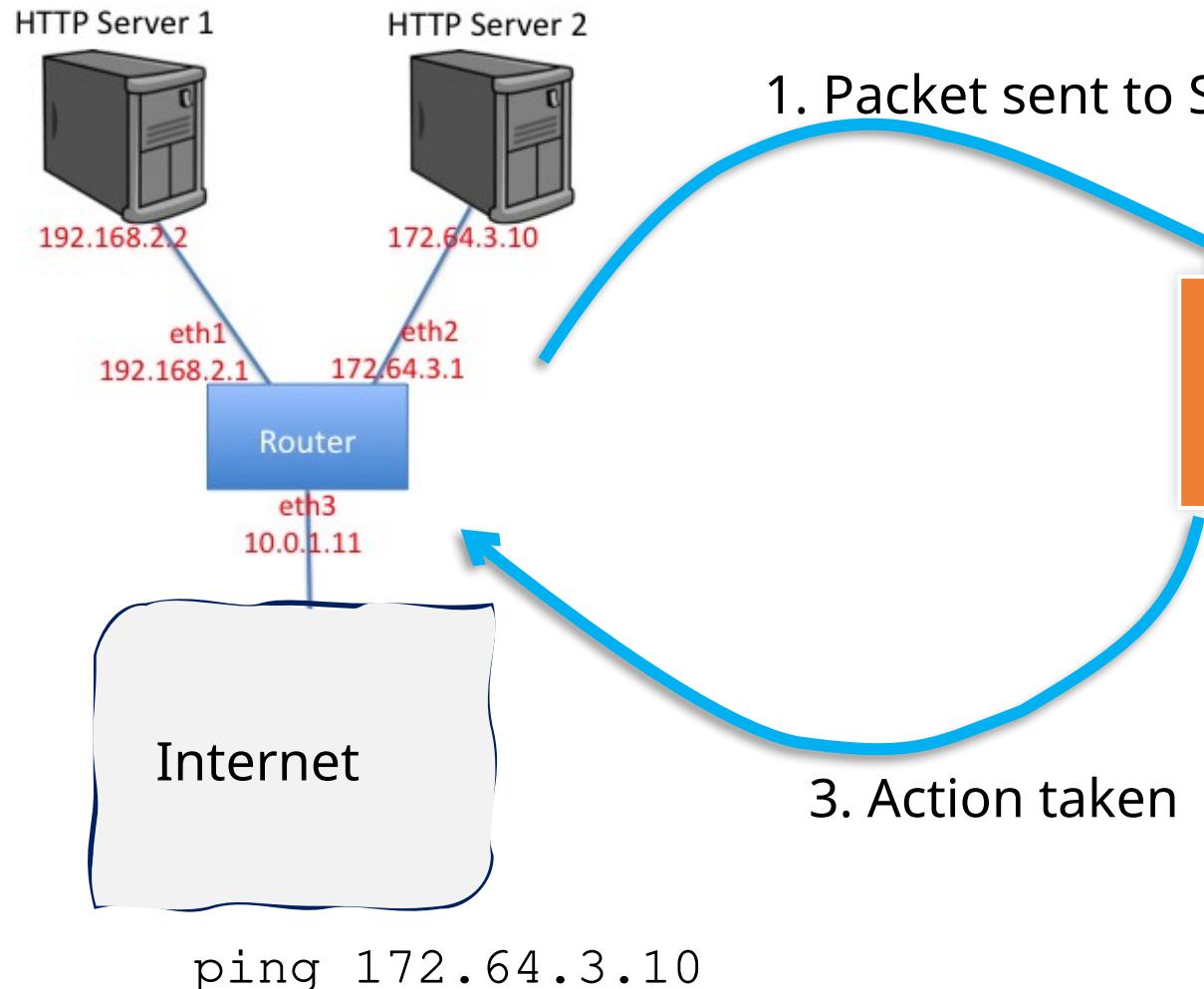
```
# Routing table
```

```
10.0.1.100 10.0.1.100 255.255.255.255 eth3
192.168.2.2 192.168.2.2 255.255.255.255 eth1
172.64.3.10 172.64.3.10 255.255.255.255 eth2
```

SR Emulated Topology



SR Emulated Topology



2. Routing decision made

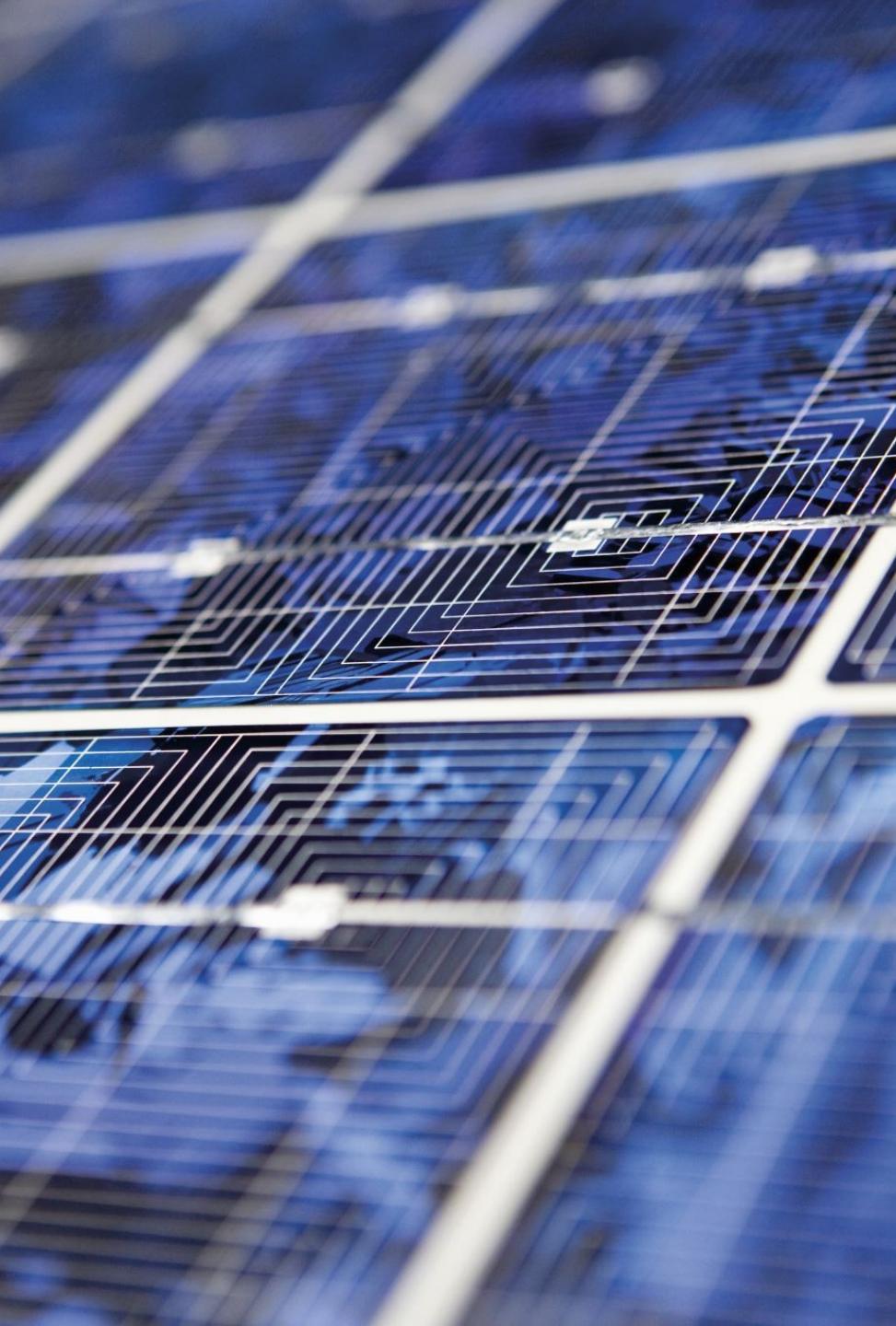
1. Packet sent to SR

3. Action taken

SimpleRouter
i.e. your code!

Routing decision:

- Look at the routing table
- Figure out to which interface to forward the packet
- Make necessary changes to the packet



Routing Logic...

Route *Ethernet frames* between the client and the HTTP servers

Handle ARP request and replies

- Maintain an ARP cache

Handle traceroutes

- Generate TTL Exceeds Message

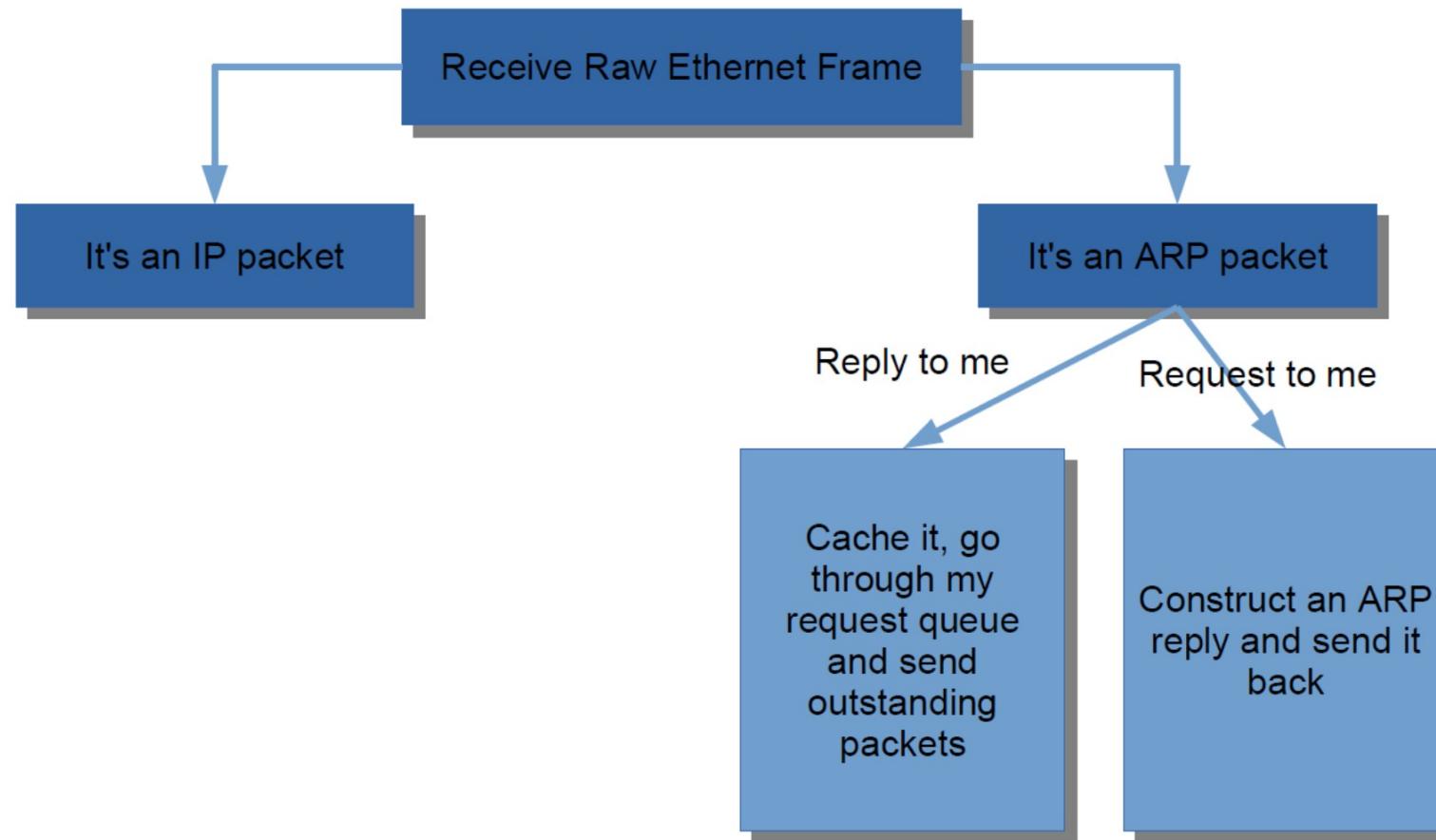
Handle TCP/UDP packets sent to one of the routers' interfaces

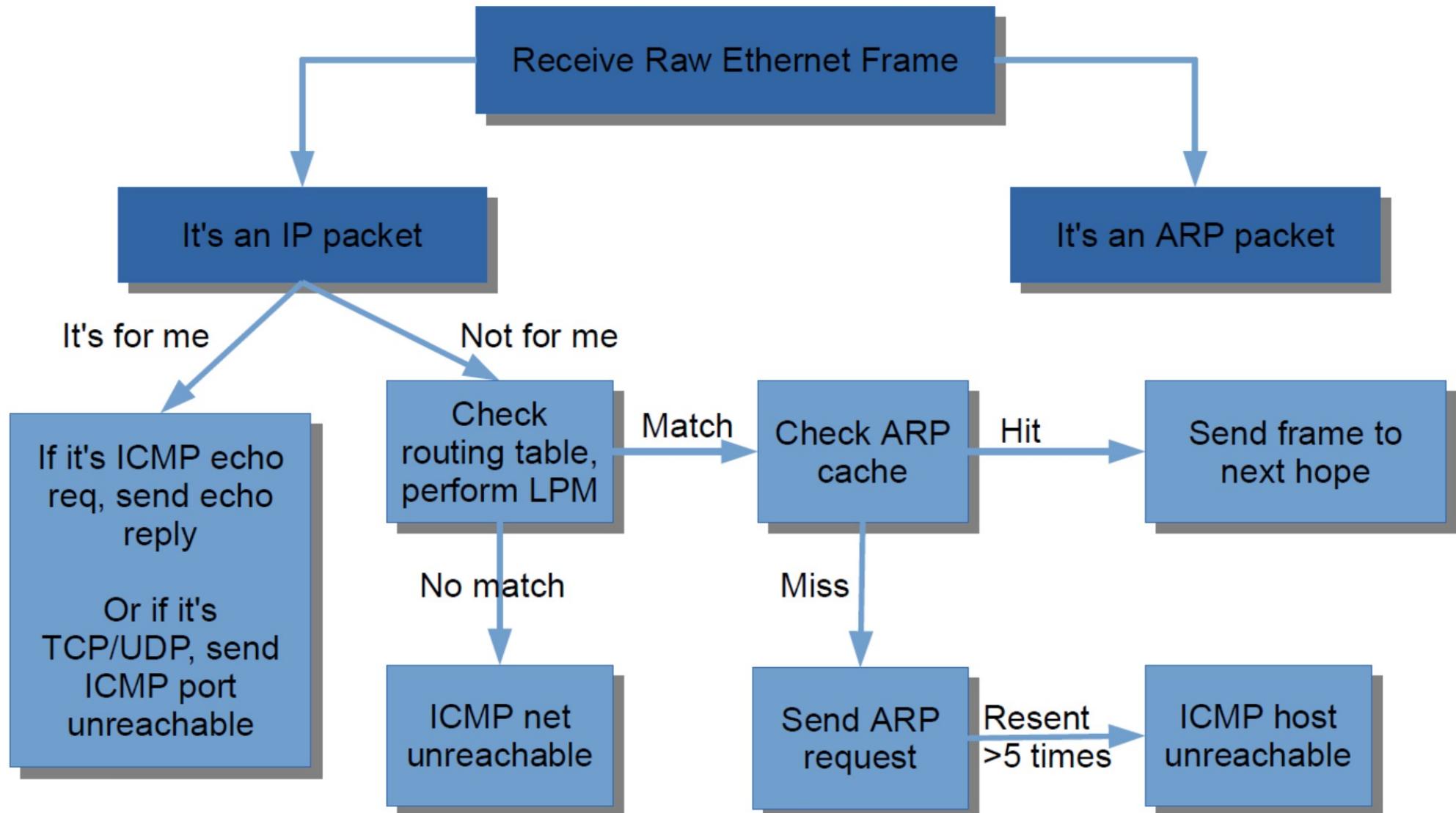
Generate ICMP Port Unreachable

Respond to ICMP echo requests

More details on the assignment handout...

A rough flow chart...





Several things still missing from this chart, e.g. Checksums, TTLs, ...

Testing your implementation

- Test connectivity with `ping` from a server or the client
- Traceroute will not work well outside of Mininet:
 - Use Mininet CLI

```
mininet> server1 traceroute -n server2
```
- HTTP requests with `wget`, `curl`, `lynx`
- Don't forget to test "error" cases!

Things that may be useful...

- Don't get mixed up with **endianess**:
 - Linux is little endian
 - Network is big endian
 - Try to put the calls to `hton()`, `ntoh()` in a single place
- Mininet console, which supports: `tcpdump`, `ping`, `traceroute`
(`apt-get install traceroute` on instance)
- Debug functions in `sr_utils.c`
`print_hdrs()` `print_addr_ip_int()`
- GDB/Valgrind
- Tutorials Point:
http://www.tutorialspoint.com/unix_sockets/index.htm