

CSC458/2209 PA1

Simple Router

Based on slides by: Antonin & Yinan Liu

Jun Lin Chen

Get the up-to-date version from
<https://www.cs.toronto.edu/~jlchen/csc458/pa1.pdf>



Asking Question

- Please ask questions on Piazza.
Try not to give away any homework hints.
I will try my best to be responsive.

Overview

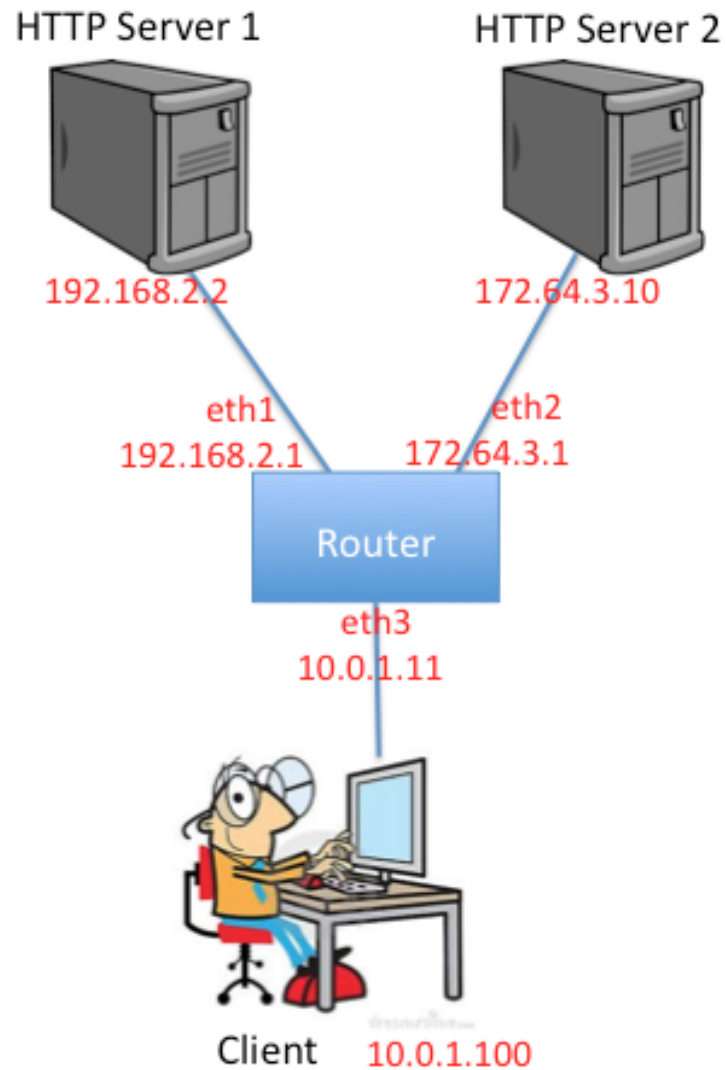
- You are going to write a “simplified” router
 - Given a static network topology
 - Given a static routing table
 - **You are responsible for** writing the logic to handle incoming Ethernet frames (ICMP, ARP, IP....):
 - Forward it
 - Generate ICMP messages
 - Drop it
 - And more ...

But how to do it???

- Where will my routing logic run?
- Where will the traffic come from?
- How will I test my code?

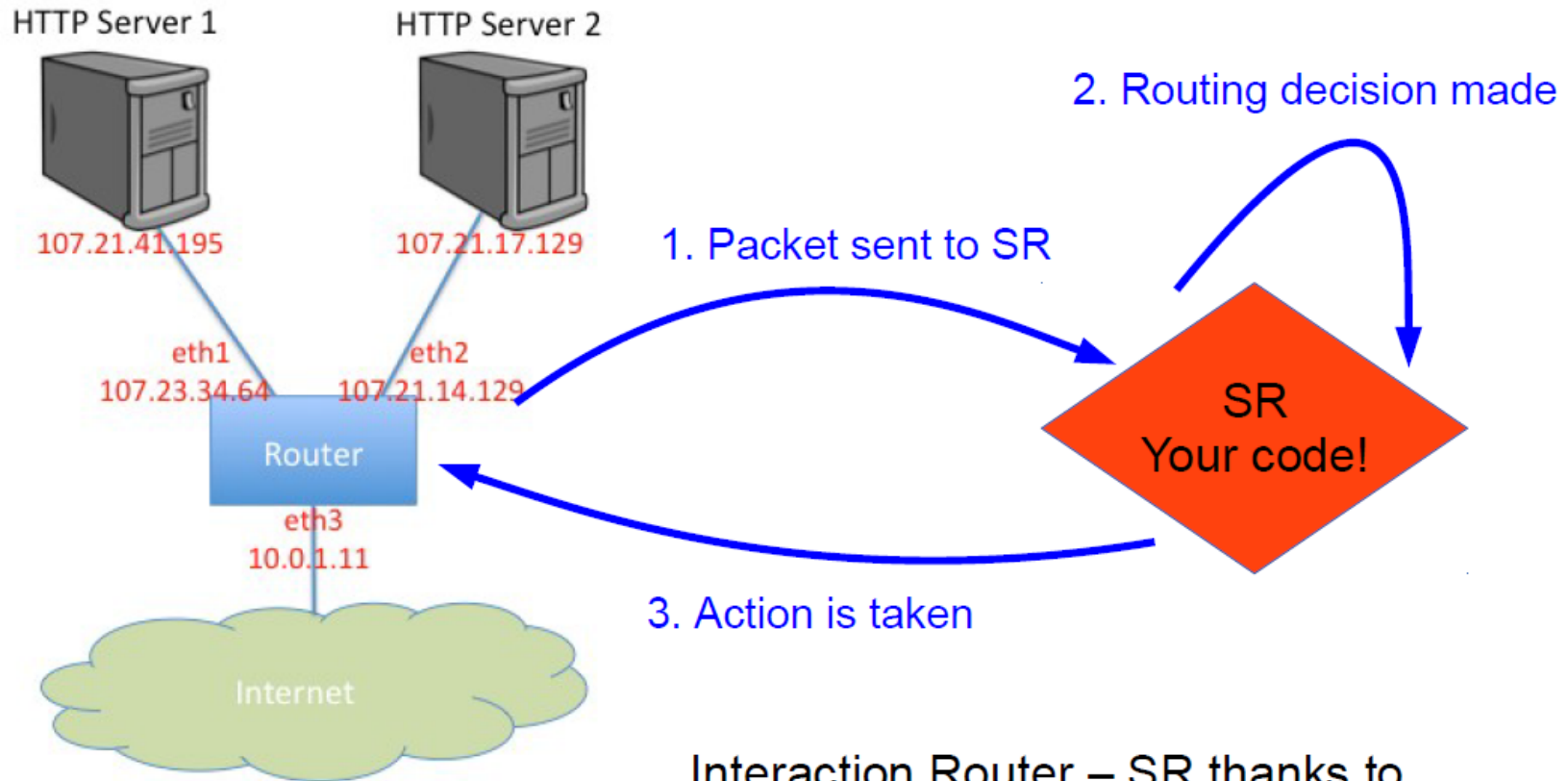
- No hardware router
- Network topology emulated with Mininet:
your router connects 2 servers to a client
- Your router will handle real traffic
- The topology is emulated on CDF machines!
So please test it in our labs

Emulated Topology



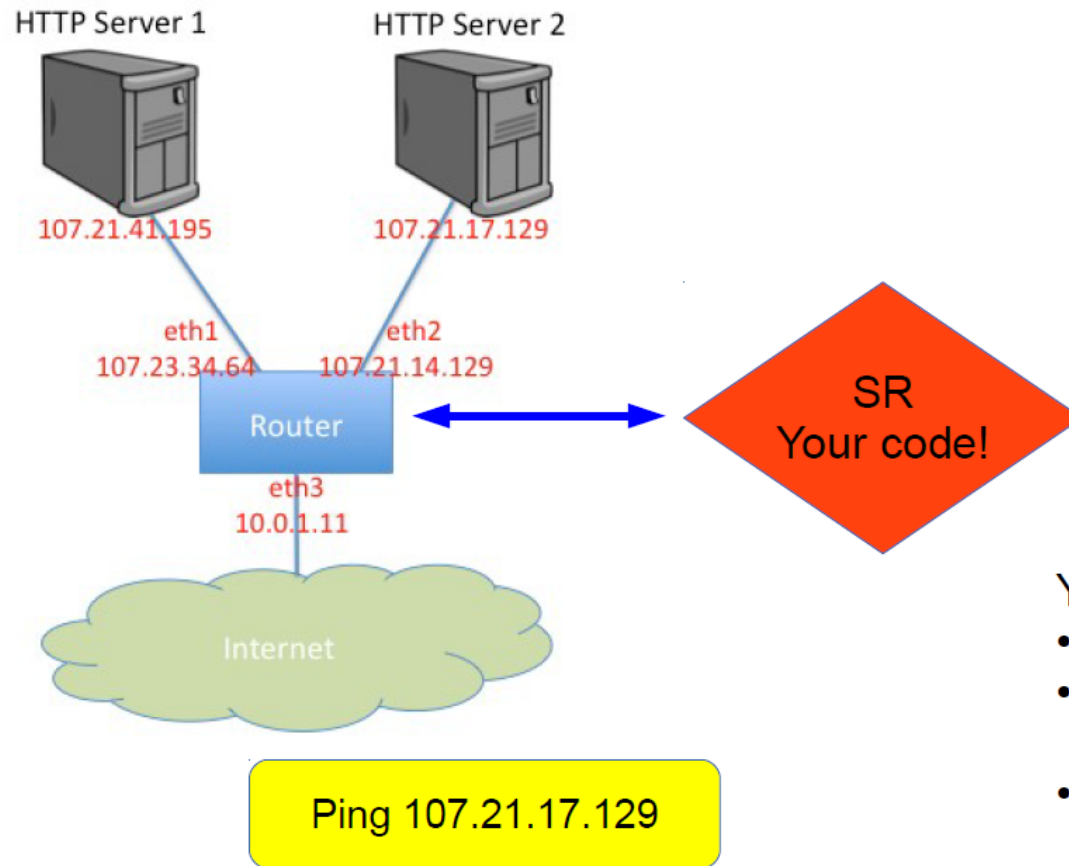
Topology for Simple Router

Emulated Topology



Interaction Router – SR thanks to
POX and Openflow

Emulated Topology



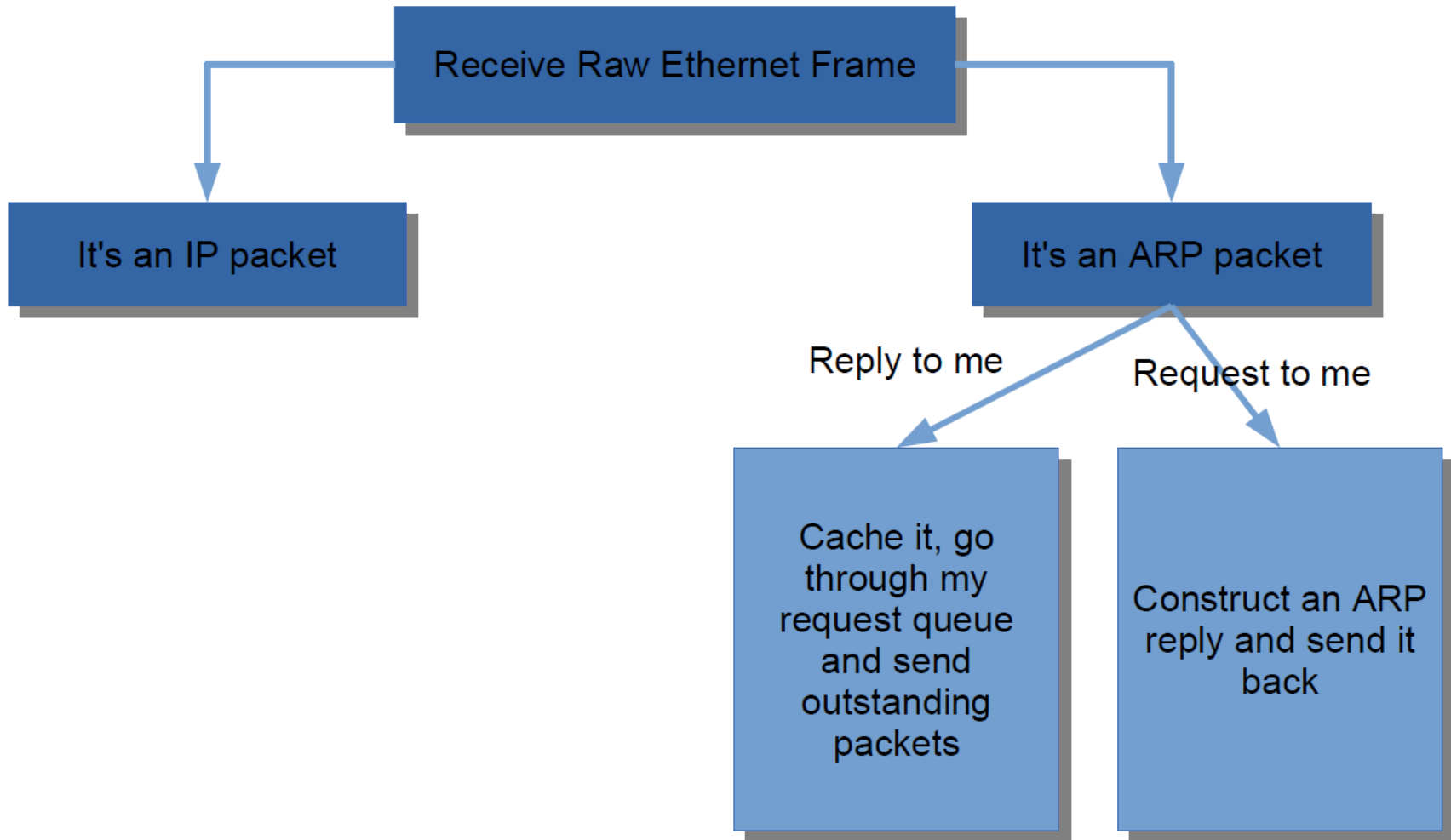
Your routing decision:

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

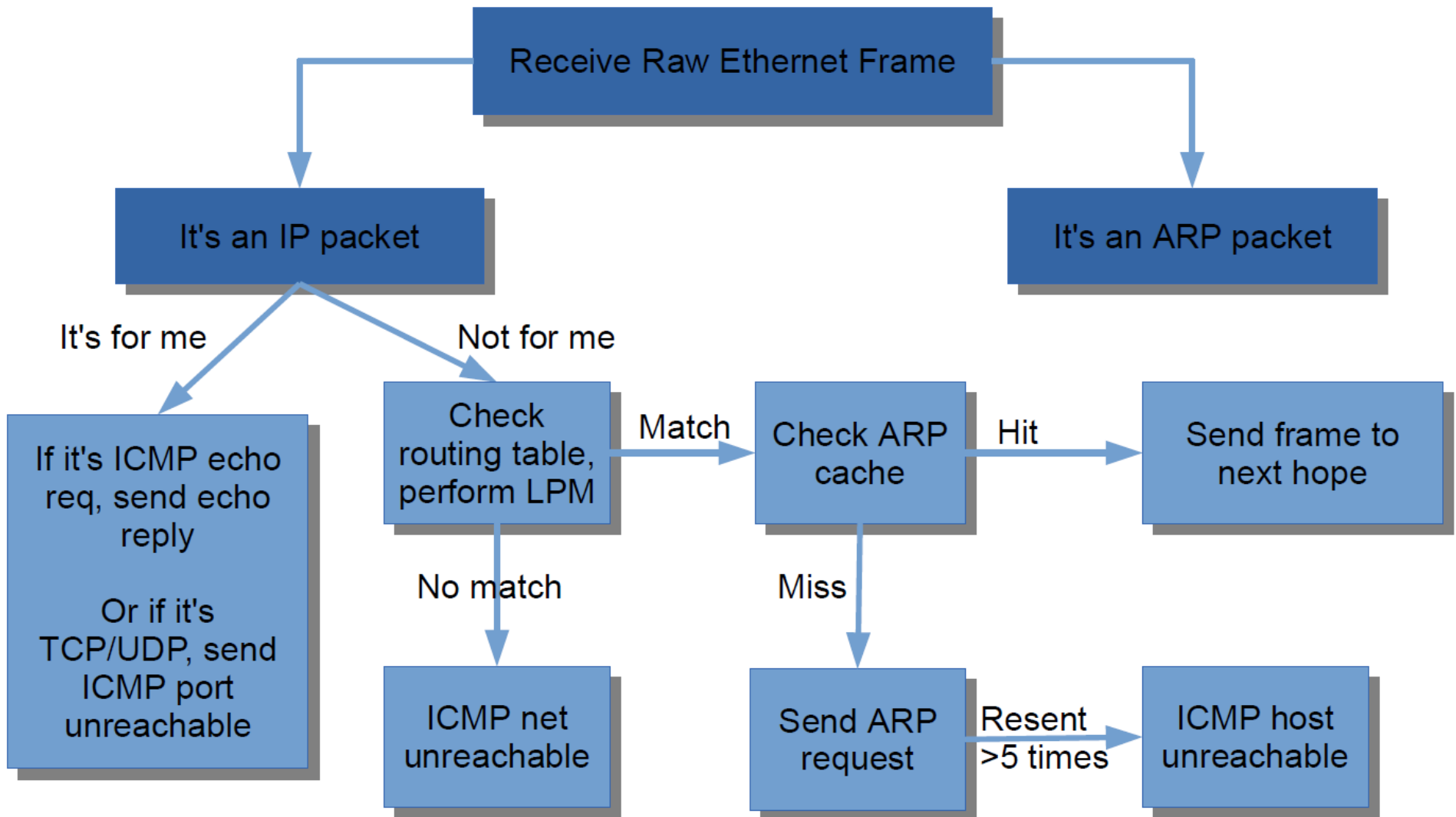
What your routing logic needs to do?

- 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
- See course webpage for full requirements:
<http://www.cs.toronto.edu/~yganjali/courses/csc458/page-3/simple-router/>

A rough flow chart



A rough flow chart



A rough flow chart

- Many things missing from this chart
 - Checksums, TTLs
- Read the instructions carefully
- 500+ lines of code, so **start early**
- Submission:
 - 8 time per day
 - Last 7 days before the final submission
 - The Markus server would give some feedback
- Final submission: **Oct. 11st at 5pm**

How to test your code?

- 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!

Some advice

- Be thorough in your testing
 - Do not hesitate to change the routing table
 - What about an incorrect routing table?
 - Be careful when implementing Longest Prefix Match
 - Don't get mixed up with **endianness**:
 - Linux is little endian
 - Network is big endian
 - Try to put the calls to **hton()**, **ntoh()** in a single place
 - Write good quality code
 - Do not hardcode constants, avoid code duplication
- Coding Guidelines
- <https://web.stanford.edu/class/cs244a/CS244aCodingGuidelines.html>

Things that may be useful

- 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
- Pizza

Start reading!

<http://www.cs.toronto.edu/~yganjali/courses/csc458/page-3/simple-router/>