CSC458 PA1
Simple Router

Ehsan Etesami
ehsan.etesami@utoronto.ca

Thanks to David, Parsa and Farid

Fall 2025

Department of Computer Science
University of Toronto

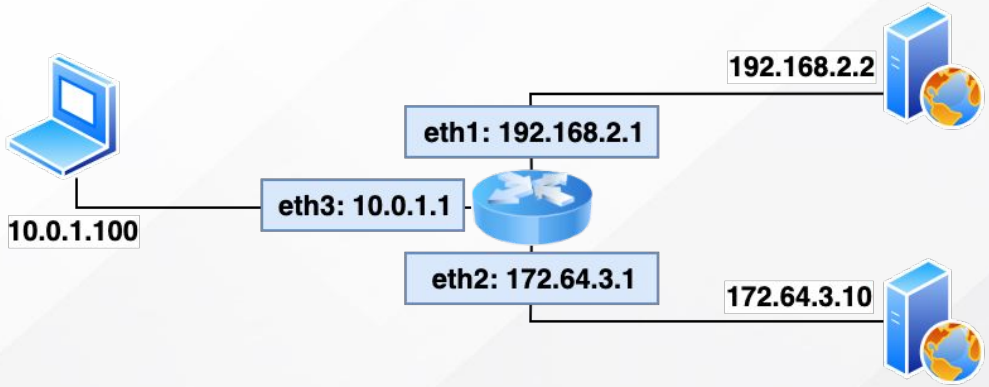
PA1 Objective

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 ...



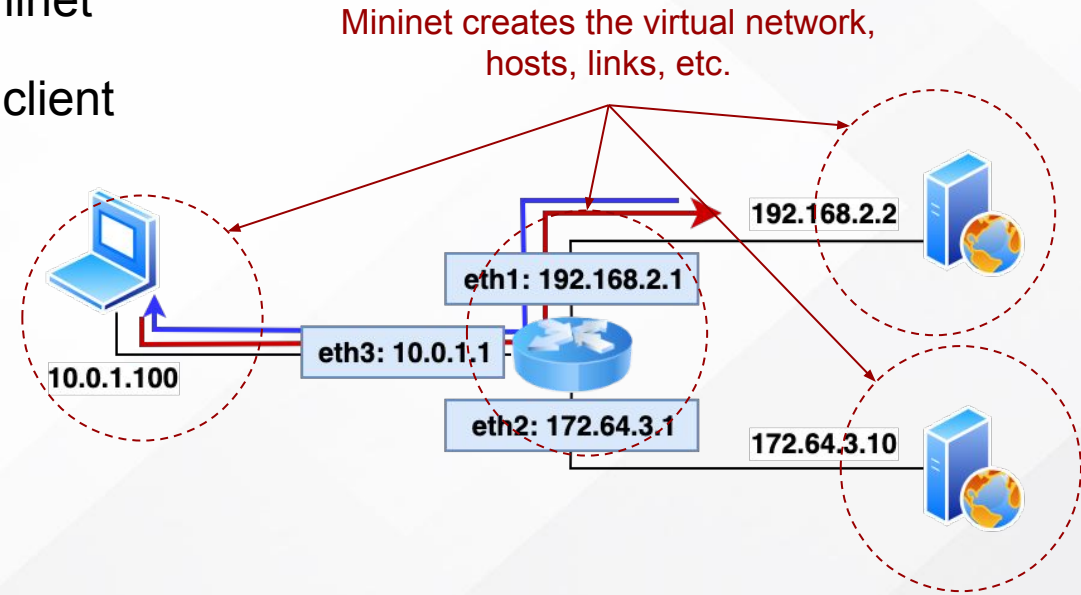
Setup Overview

No hardware router

Network topology emulated with Mininet[1]

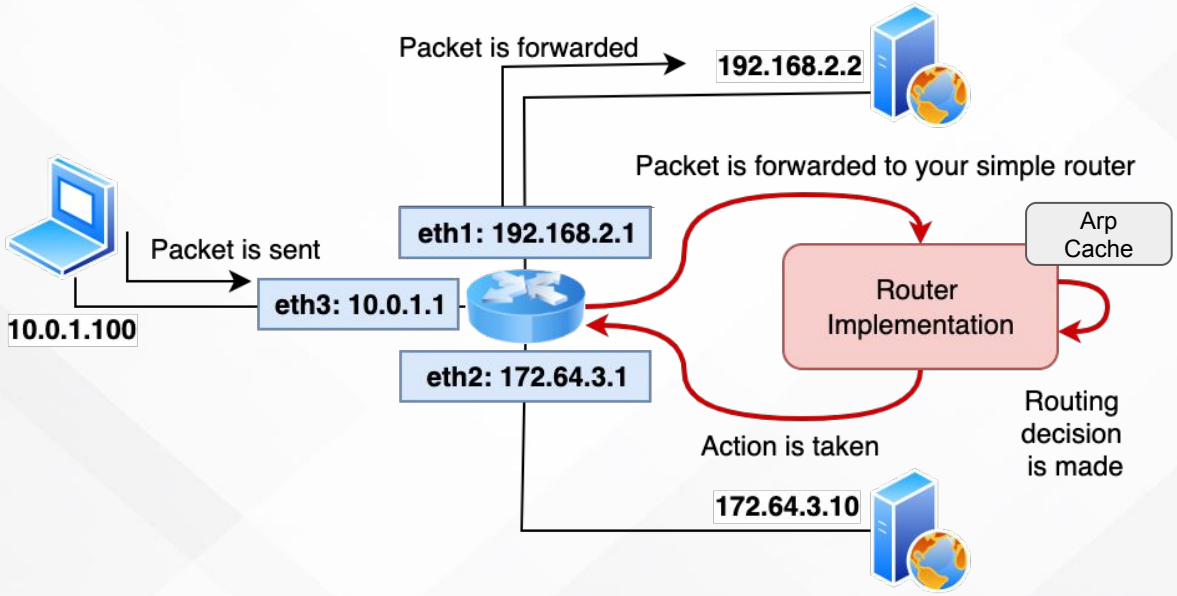
Your router connects 2 servers to a client

Your router will handle real traffic



1. https://mininet.org/

Emulated Topology



Routing Decision:

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

What Your Routing Logic Must Do

Receive and parse raw Ethernet frames

Route **Ethernet frames** between client and servers

Perform **IPv4 forwarding** (**TTL decrement**, **checksum update**, **Longest Prefix Match**)

Handle **ARP requests/replies**, **maintain ARP cache** (15s timeout, retries once/sec up to 5 times)

Queue packets **while resolving ARP**

Handle **TCP/UDP packets** sent to **router interfaces**

Generate **ICMP messages**:

- Echo replies
- Destination unreachable
- TTL exceeded

↪ ***See handout for full requirements***

Expected User Behavior

Ping any **router interface** from client

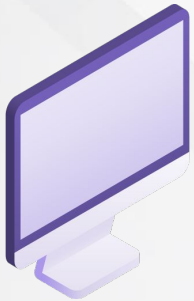
Tracepath/traceroute to and through the router

Ping servers (and **traceroute** to them)

HTTP download (**wget/curl)** from app servers

Correct **ICMP responses** when appropriate (ping to router, port unreachable, etc.)

Introduction to IP & Mac Addresses



IP address can be changed

IP Address: 10.0.0.1

MAC Address: 00:1B:44:11:3A:B7

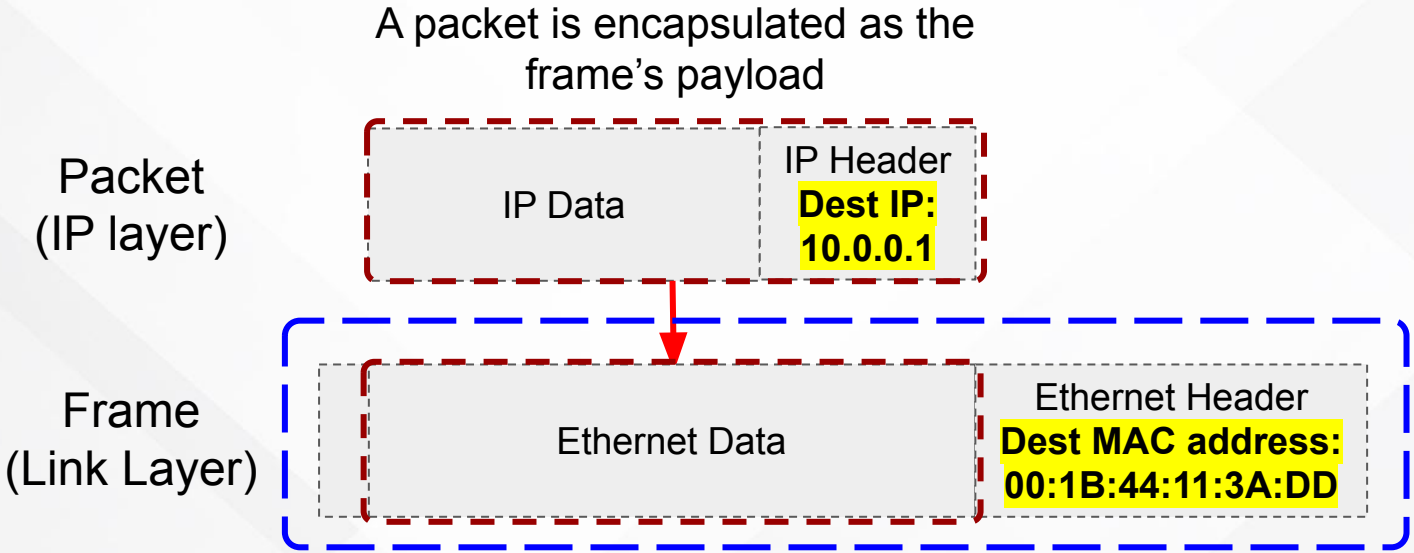
unique to every network card or interface

When data is sent over a network, it uses the IP address to find the correct destination.

Within the local network, it needs the MAC address to actually deliver the data to the right device.

Introduction to ARP

ARP: Address Resolution Protocol



ARP Table

IP Address	MAC Address	TTL
10.0.0.1	00:1B:44:11:3A:DD	15000

Key Protocols to Know

Ethernet: src/dst MACs; forwarding by changing dst MAC to next-hop

IPv4: header lengths, TTL, checksum

ICMP: echo request/reply + error types and codes

ARP: request broadcast, reply unicast; mapping IP ↔ MAC; caching & timeouts

ICMP Messages to Generate

Echo reply (type 0): reply to ping to our interface

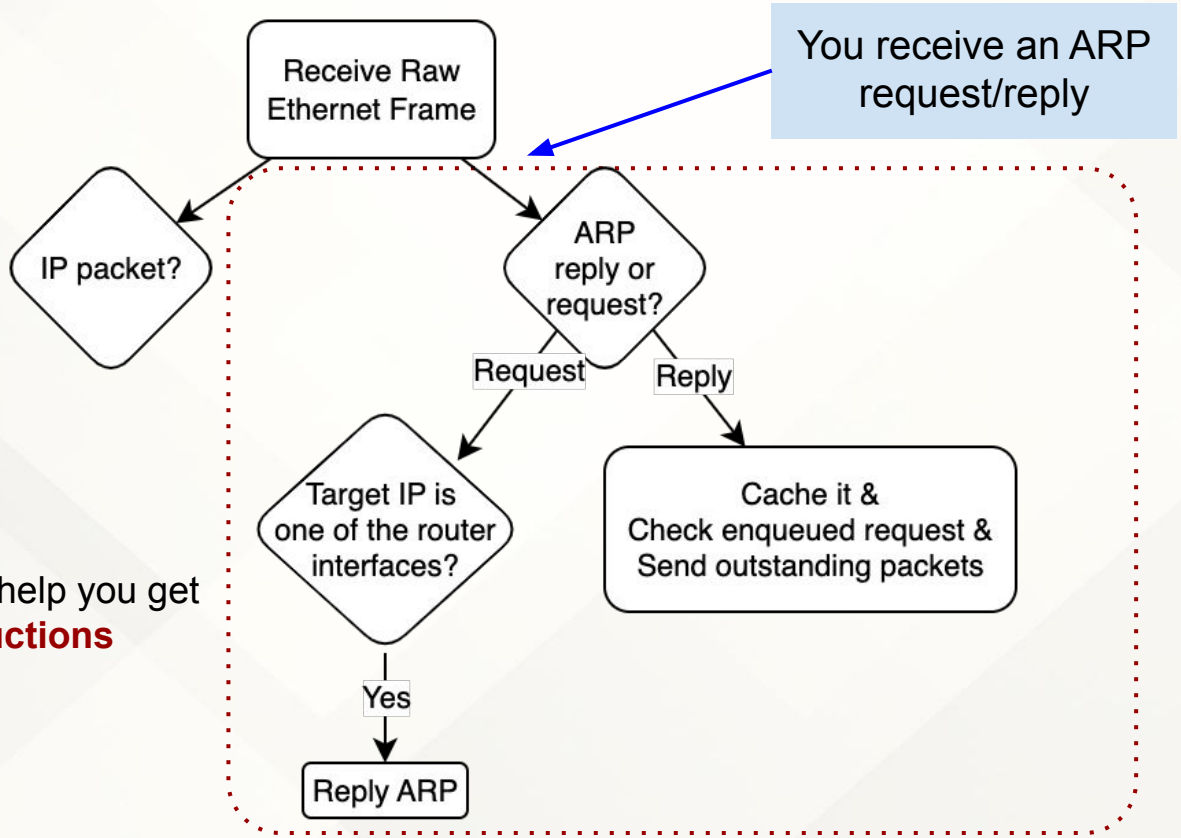
Destination net unreachable (type 3, code 0): no route found

Destination host unreachable (type 3, code 1): 5 ARP requests timed out

Port unreachable (type 3, code 3): TCP/UDP to router interface

Time exceeded (type 11, code 0): TTL expired / dropped during forwarding

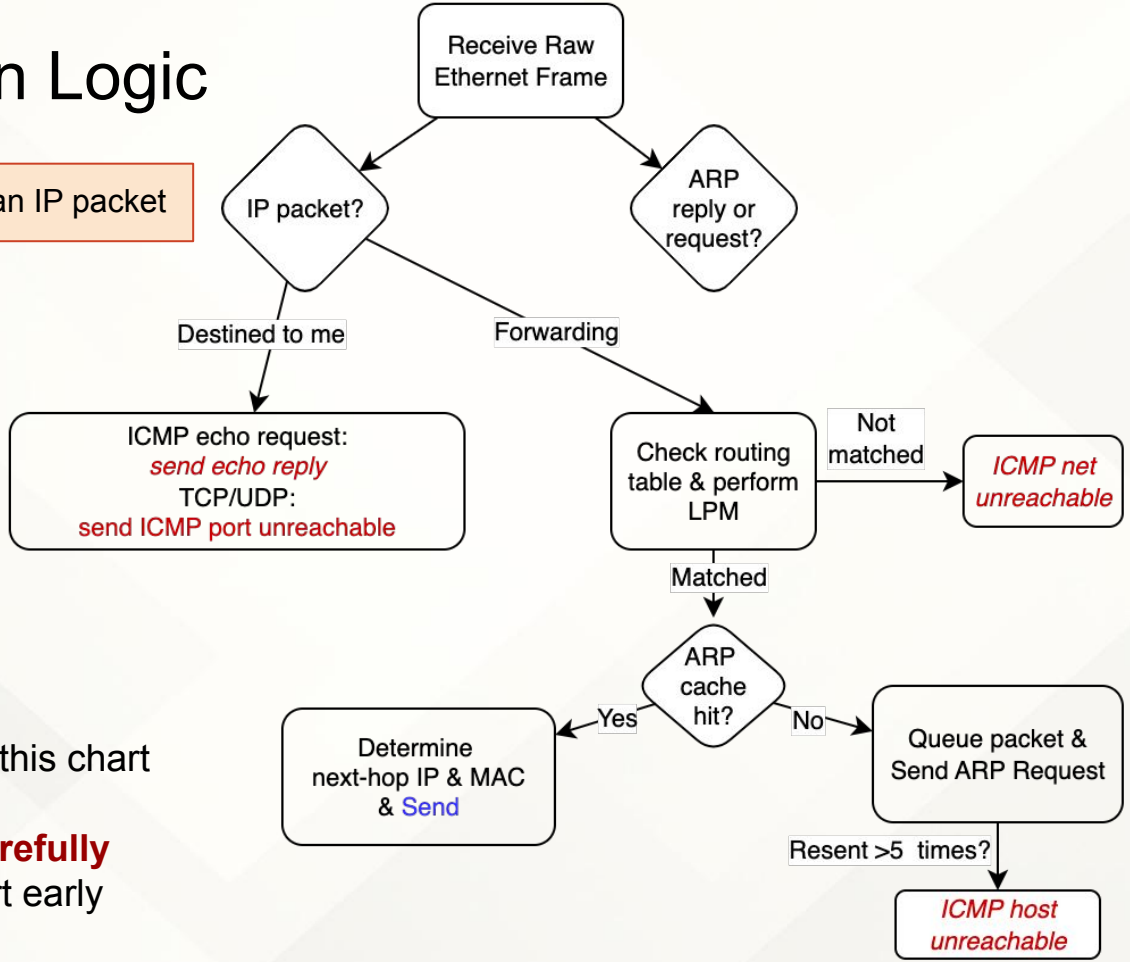
Router Decision Logic



↳ Use this chart only to help you get started. **Read the instructions carefully!**

Router Decision Logic

You receive an IP packet



- Many things missing from this chart
  Checksums, TTLs
- **Read the instructions carefully**
- 500+ lines of code, so start early

Files & Code Structure (where to implement)

`sr_router.c / sr_router.h` — main packet handler: sr_handlepacket()

`sr_arpcache.c / sr_arpcache.h` — ARP cache, request queue; implement `sr_arpcache_sweepreqs()`

`sr_vns_comm.c – sr_send_packet()` used to send raw packets

`sr_protocol.h` — packet header structs (Ethernet/IP/ICMP/ARP)

`sr_if.c / sr_rt.c` — helpers for interfaces & routing table (rtable read)

sr_utils.c — provided helpers & printing utilities

Automated Tester

Public tests (50%); core forwarding & ARP behavior:

⇉ ARP reply, ARP expiration, ICMP echo, ICMP forward, TCP forward

Private tests (40%); robustness and hidden corner cases:

⇉ additional hidden cases

Style/documentation (10%): code quality, comments, README

Deliverable & Submission

Package contents of router directory into pa1.tar.gz

Use provided **Makefile compress rule**: `cd router/ ; make compress`

Ensure make produces **executable** file

Submit pa1.tar.gz to Markus

If adding files, extend the tar command in Makefile compress rule

Recommendations and Tips

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

Tools and Best Practices

Mininet console, which supports:

- ⇉ `tcpdump, ping, traceroute`

Debug functions in `sr_utils.c`

- ⇉ `print_hdrs()`
- ⇉ `print_addr_ip_int()`

GDB/Valgrind

PA1: Environment Setup

On Intel/AMD computers:

- Use a VM image in VirtualBox

On ARM MacBooks and Macs:

- Install the UTM virtual machine software
- Use the provided ARM64 GNU/Linux virtual machine image