CS 118 Discussion Week 8: Intra-Domain Routing and the Link Layer

Slides by Eric Newberry, UCLA

Questions

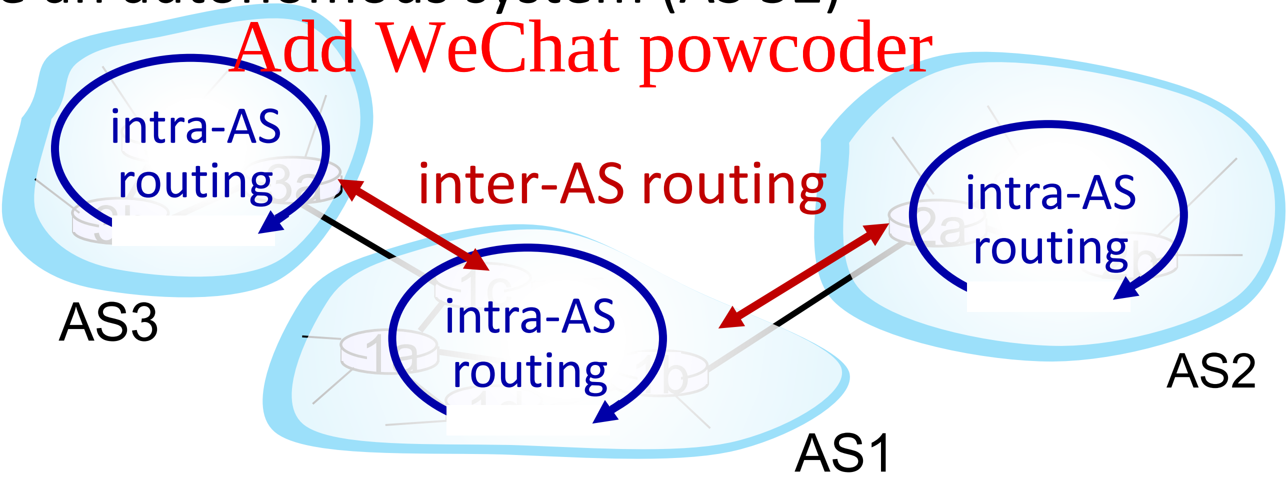
- Any questions from last week’s material or Project 2?

Making Routing Scalable

- Routing protocols spread information about how to reach destinations throughout the network
- What are some limitations of the routing protocols we discussed last time?

Autonomous Systems (ASes)

- Split Internet up into regions under one administrative control – each one is an "autonomous system" (AS)
  - Can be thought of as a domain, as in DNS
  - Assigned unique number by a central authority
  - UCLA would be an autonomous system (AS 52)



Source: Kurose & Ross, 8th Edition Slides

Intra-AS Routing

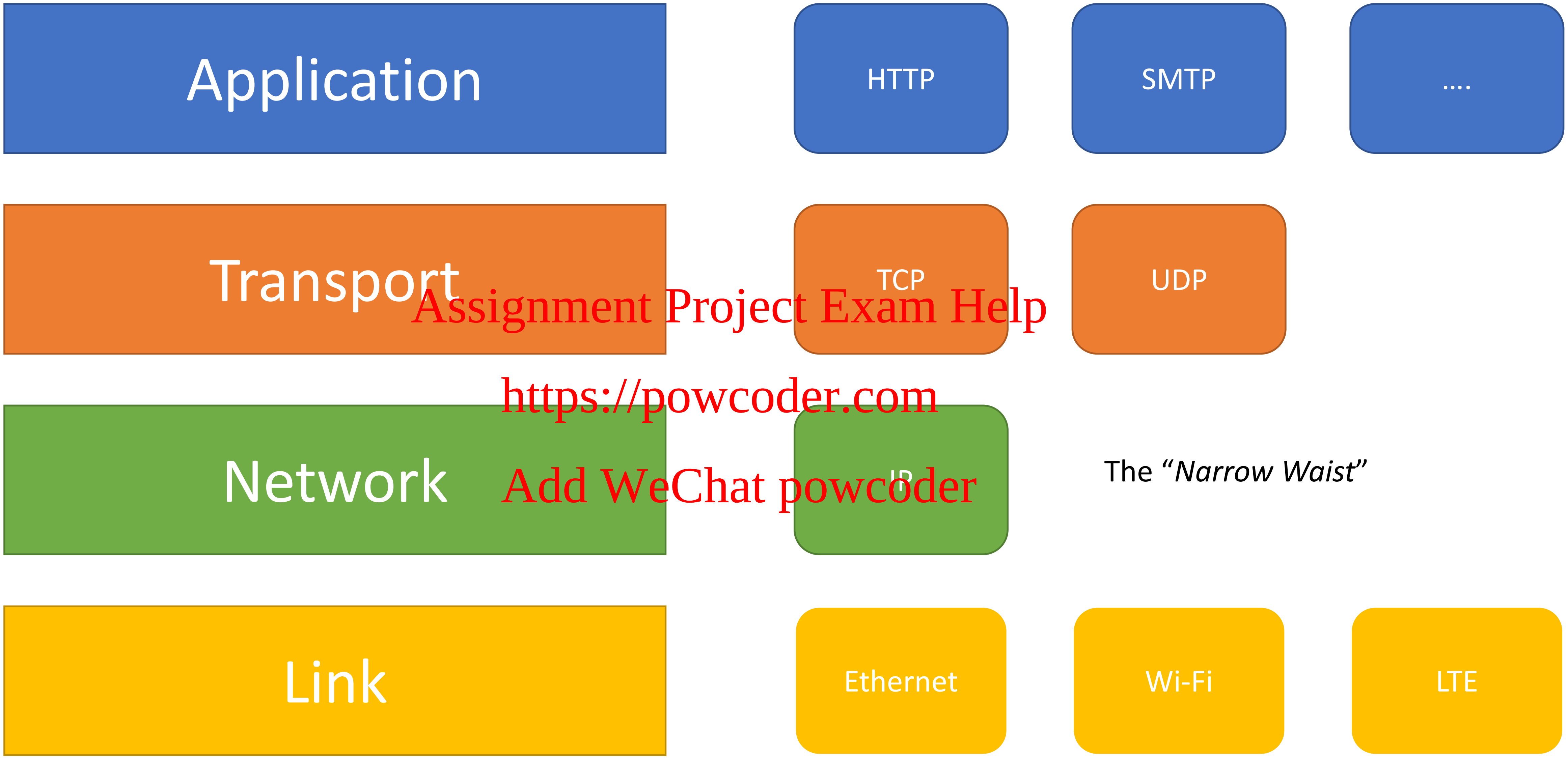
- Problem: How do we route inside an AS?
- Solution: Use link-state or distance-vector (like before)
  - Perhaps with some optimizations
- Examples of intra-AS routing protocols:
  - Routing Information Protocol (RIP) – distance-vector, mostly no longer used
  - Enhance Interior Gateway Routing Protocol (EIGRP)
    - Formerly proprietary distance-vector protocol
  - Open Shortest Path First (OSPF)
    - Basically just like the link-state protocol we learned last week, but with added security, multiple possible link metrics (how costs/weights are determined)
    - Can be hierarchical, limiting advertisement flooding to “areas” and having “border routers” that provide connectivity between areas

Inter-AS Routing: BGP

- Border Gateway Protocol (BGP)
- Route for "policy" (economics) instead of lowest cost
  - E.g., prefer customer routes more because they make money vs using links where we are customer and have to pay for traffic
- "Path-vector" routing – send AS path to reach destination to neighbors
  - E.g., [AS 52, AS 3, AS 32578, AS 673, AS 7933]
  - Can also detect loops if we see same AS multiple times in path!

Internet Control Message Protocol

- Special protocol used by IP hosts to communicate network diagnostic information
- Ping uses ICMP ("Echo" results in an "Echo Reply")
- Traceroute uses ICMP
  - Essentially pings with increasing time-to-live (TTL), resulting in a "TTL expired" message back from each hop


Application
HTTP
SMTP
....
Transport
TCP
UDP
Network
IP
The “*Narrow Waist*”
Link
Ethernet
Wi-Fi
LTE

The Link Layer

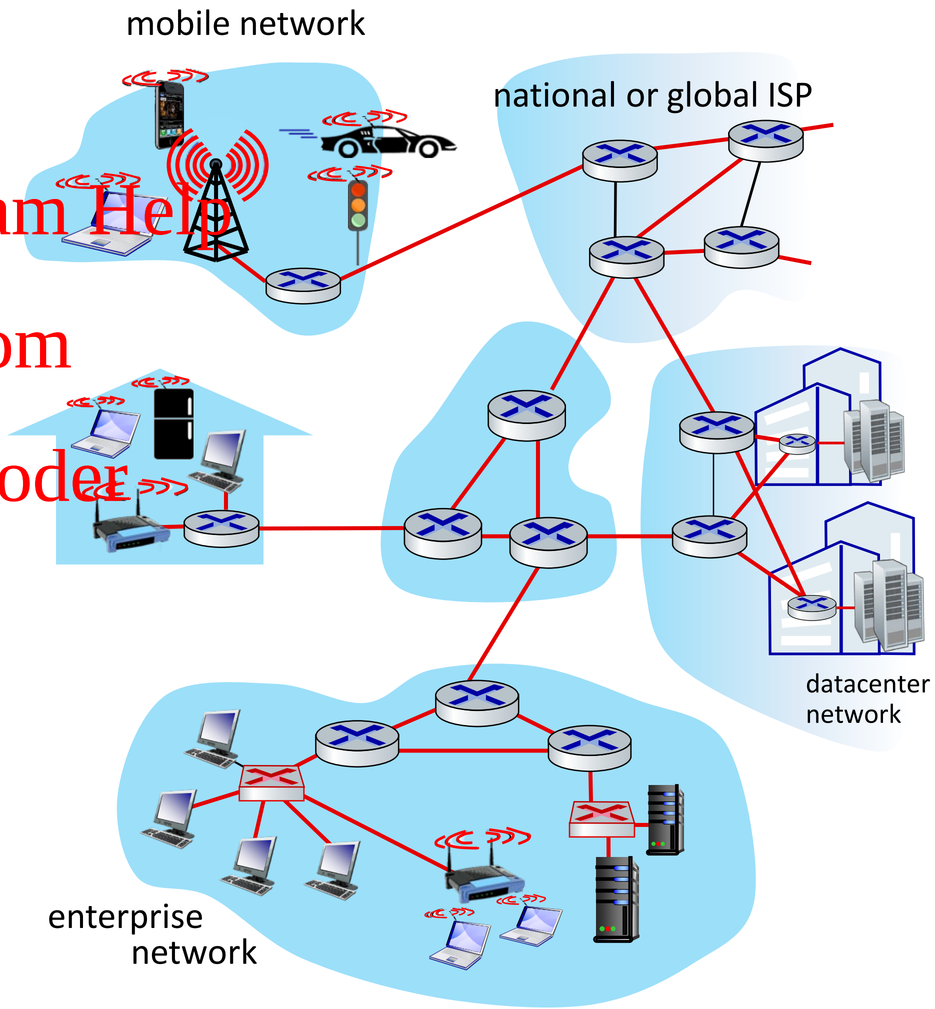
- How do we send packets from one host to another over some medium
- Medium: e.g., copper wire, the air, fiber optics
- Why is it separate from the Network layer?

Network Layer vs. Link Layer

- We need an identical protocol that runs on all hosts – the IP protocol
  - One protocol to tie them all together – the “narrow waist”
- However, we need to communicate between *physically adjacent* hosts over various types of links
- Don’t want to have to change the network layer protocol on all hosts when adding a new type of link

Link Layer Protocols

- Ethernet
- Wi-Fi
- LTE (4G)
- 5G
- Satellite links



Source: Kurose & Ross, 8th Edition Slides

Terminology

- What do we call units of data sent over each layer?
- Transport – Segment
- Network –Datagrams (Packets)
- Link – Frames

Link Layer Services

- Framing
  - Split datagrams up into “frames”
  - Need to split to fit in MTU of link – “Maximum Transmission Unit”
- Link access (sometimes)
  - If physical link is shared by multiple hosts (i.e., can overhear transmissions not intended for us), need to make sure don’t transmit over others
  - Identify sender and receiver (on Ethernet, “MAC” addresses)
- Reliable delivery (sometimes)
  - Need to make sure packets actually get to other end of link
  - Needed on links with high probably of corruption or loss (e.g., Wi-Fi)
  - Why would be have reliability if TCP already provides end-to-end reliability?

Link Layer Services

- Flow Control
  - Don't send too quickly and overwhelm remote host

- Error detection and (optionally) correction
  - Use checksums to make sure packets that get to other end of link without being corrupted by noise, etc.
  - Optionally, receiver can correct single or few bit errors with "error correcting codes"

- Duplex
  - Half-duplex – only one host can transmit on link at once
  - Full-duplex – all hosts can transmit on link at once

Cyclic Redundancy Checks

100000

$$1001\overline{)101110000}$$

1001

00101

000

1010

1001

00110

000

1100

1001

0101

1010

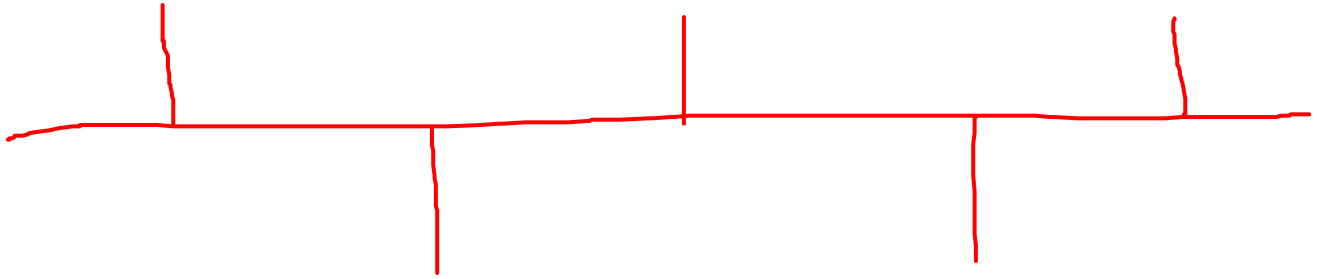
1001

0011

CRC = 011

Broadcast Links vs. Point-to-Point

- Point-to-point is where two hosts, and only two hosts, are directly connected by a link
  - E.g., Modern Ethernet
- Broadcast mediums have multiple hosts sharing the same medium that can overhear each other's transmissions, even if not intended for them
  - E.g., Wi-Fi, cellular networks, classical Ethernet (that nobody uses anymore)



The Downsides of Broadcasting

- What if multiple hosts transmit at once?
- Collision! The bits get garbled and nobody gets understood ☹
- Have to devise a way to share the medium without talking over each other → "Media Access Control" (MAC)
- In practice, two main types of approaches:
  - Everyone transmits at specified and different times, frequencies, etc.
    - E.g., time-division multiple access (TDMA), frequency-division multiple access (FDMA)
    - No collisions, but less efficient use of bandwidth – what if division goes unused?
  - Random access protocols
    - E.g., everyone tries to transmit whenever – allows collisions, but *recovers* from them

Ethernet Media Access Control: CSMA/CD

- Listen to see if anyone currently transmitting. If so, wait and try again
- If nobody transmitting, begin transmitting but listen for other signals on the wire
- If detect another transmission on top of yours, back off for a random period of time and try again
  - If keeps happening, keep expanding the range of time your timer can last for
  - Also send a special “jam” signal to alert everyone on the link about collision
    - Must be a minimum of 64 bytes to ensure all hosts on the entire span of link know

Modern Ethernet: Point-to-Point

- As link speeds increase, collisions become more costly to handle
  - Need greater jam packet sizes to ensure all nodes know at low latencies
- Moreover, can still only do half-duplex
- Therefore, modern Ethernet uses only point-to-point links between devices
- Wi-Fi protocol is based largely on Ethernet but still uses shared medium (the air), so uses a related mechanism called CSMA/CA
  - Will cover this later when we get to wireless networks

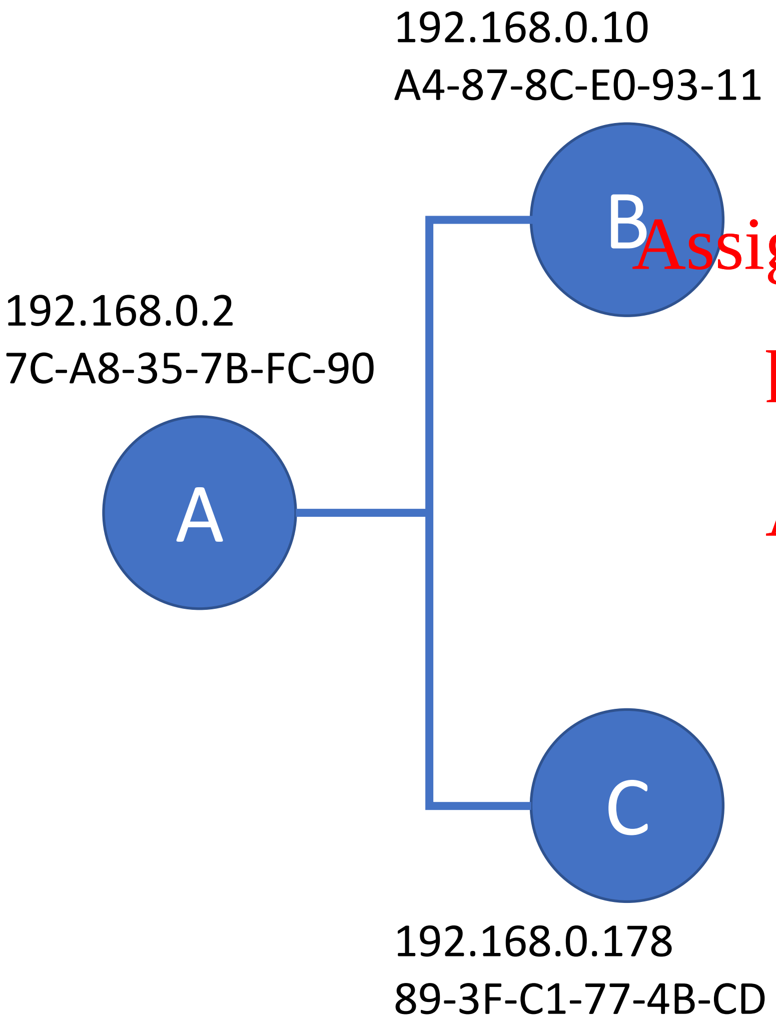
Link-Layer Identifiers

- IP addresses uniquely identify hosts at network layer
- But we also need to uniquely identify hosts at link layer
  - Many link-layer protocols predate IP and used different identifiers
  - Different requirements – IP intended to bridge links of different types
- For Ethernet, use globally-unique 48-bit identifiers called MAC addresses
  - Represent in hexadecimal, e.g., AB-12-CD-34-EF-56
  - Usually "burned into" physical interface cards at time of manufacture

Address Resolution Protocol

- Great, we have MAC addresses! But how do we find out who has what MAC address?
  - Enter the Address Resolution Protocol (ARP)
  - Maps IP addresses to MAC addresses on a local link
- Super simple!
  - Don't know mapping of IP on same link → MAC? Ask "who has IP address x.y.z.a?"
  - Host responds "I have IP address x.y.z.a and my MAC address is AA-BB-CC-DD-EE-11"
  - Store this mapping for future use (expire at some point in case mapping changes)

ARP in Action



A: Who has C (192.168.0.178)?

C: I have ______________ at MAC address 89:3F:C1:77:4B:CD

A's ARP Table

IP Address	MAC Address
. . . .178	89: . .

Router Project Notes

Forwarding process outline – handlePacket()

- Receive Ethernet frame in SimpleRouter::handlePacket()
- Check if frame destined for us or for FF:FF:FF:FF:FF:FF – if not, drop
- Check if EtherType field is IP – if not, return
- Validate IP header checksum – if invalid, return
- Decrement TTL by 1 – if now == 0, return
- Recompute IP header checksum and insert into header
- Check if IP packet destined to a local interface
  - If so…
    - Check if protocol field in IP header == ICMP – if not, return
    - Respond to ping packet
  - If not…
    - Perform next hop lookup and send packet

Next hop lookup and send

- Check routing table for longest-prefix matching next hop entry for destination IP address
  - Includes “next hop” IP address, information about interface to send on
- Set Ethernet frame source address to MAC address of interface to send on
- Check if existing ARP table entry for MAC address of “next hop”
  - If so, set as Ethernet frame destination address and send packet on next hop interface with sendPacket()
  - If not, queue packet for ARP request with m_arp.queuePacket()
    - Our code will handle actually sending out any queued packets when ARP response received

Responding to a ping packet

- Check if ICMP code field == Echo Request – if not, return
- Change ICMP code field to Echo
- Recompute ICMP checksum (note that covers from start of ICMP header to end of packet buffer)
- Swap IP source and destination addresses
- Set IP TTL to 64
- Recompute IP checksum and insert in packet
- Perform longest-prefix routing and send as if a normal packet