Computer Network Design

Network Layer II

Yalda Edalat – Spring 23

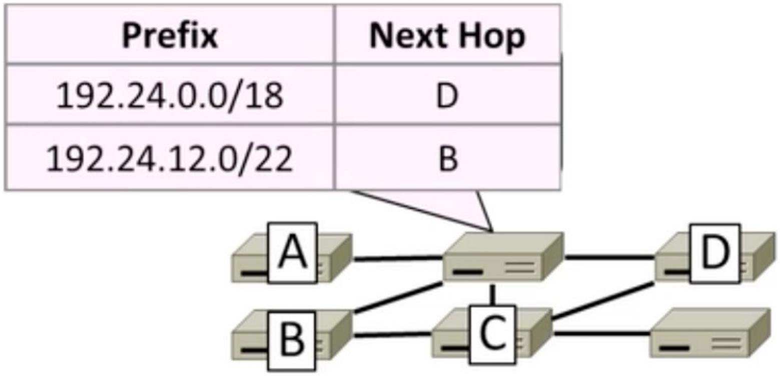
Review

- Which devices are on same networks?
- A: 192.168.137.44/21
- B: 192.168.141.12/21
- C: 192.168.145.3/21

- A: 11000000.10101000.10001001.00101100
- B: 11000000.10101000.10001101.00001100
- C: 11000000.10101000.10010001.00000011

IP Forwarding

- IP addresses on one network belong to the same prefix
- Node uses a table that lists the next hop for IP prefixes



Prefix	Next Hop
192.24.0.0/18	D
192.24.12.0/22	B

Longest Matching Prefix

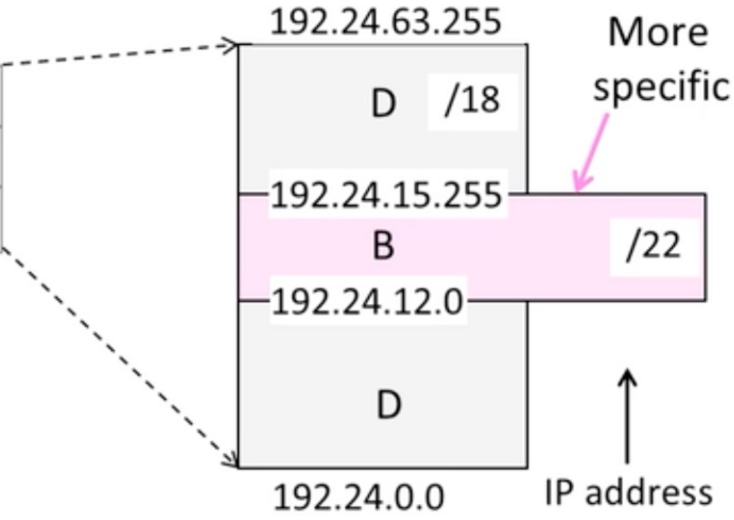
- Prefixes in the table might overlap!

- Longest matching prefix forwarding rule:
  - For each packet, find the longest prefix that contains the destination address, i.e., the most specific entry
  - Forward the packet to the next hop router for that prefix

Longest Matching Prefix (2)

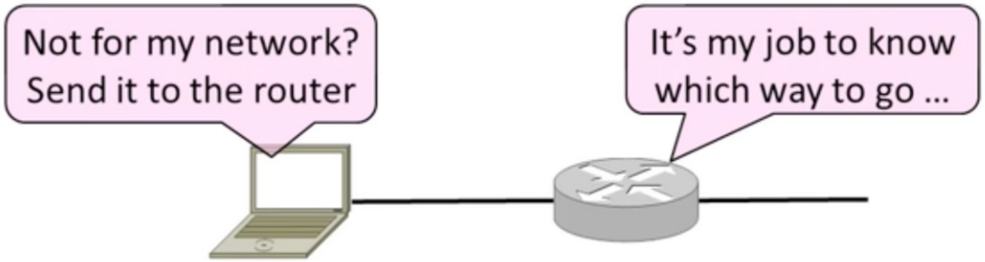
Prefix	Next Hop
192.24.0.0/18	D
192.24.12.0/22	B

192.24.6.0 →
192.24.14.32 →
192.24.54.0 →



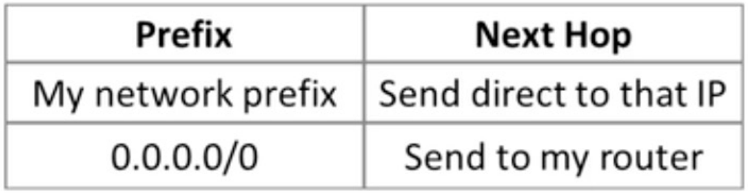
Host/Router Distinction

- In the Internet:
  - Routers do the routing, know which way to all destinations
  - Hosts send remote traffic (out of prefix) to nearest router


Not for my network? Send it to the router
It's my job to know which way to go ...

Host Forwarding Table

- Give using longest matching prefix
  - 0.0.0.0/0 is a default route that catches all IP addresses



Prefix	Next Hop
My network prefix	Send direct to that IP
0.0.0.0/0	Send to my router

Example

- Source: 192.168.22.78/24
- Destination: 192.168.23.71/24

- What is next hop? Is it 192.168.23.71? Or default router of source?

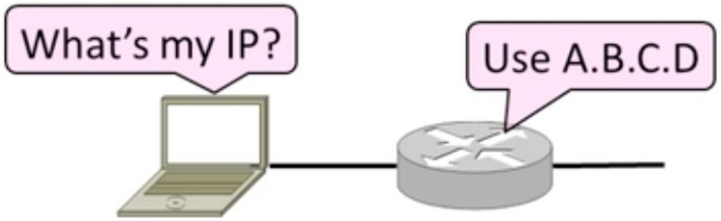
Getting IP Addresses

- Problem:
  - A node wakes up for the first time …
  - What is its IP address? What is the IP address of its router? Etc.
  - At least Ethernet address is on NIC

Getting IP Addresses (2)

1. Manual configuration (old days)
   - Can’t be factory set, depends on use

2. A protocol for automatically configuring addresses (DHCP)
   - Shifts burden from users to IT folk



DHCP

- DHCP (Dynamic Host Configuration protocol) is widely used

- It leases IP address to nodes
- Provide other parameters too
    - Network prefix
    - Address of local router (default router)
    - DNS server, time server, etc.

- DHCP is a client-server application
    - Uses UDP ports 67, 68

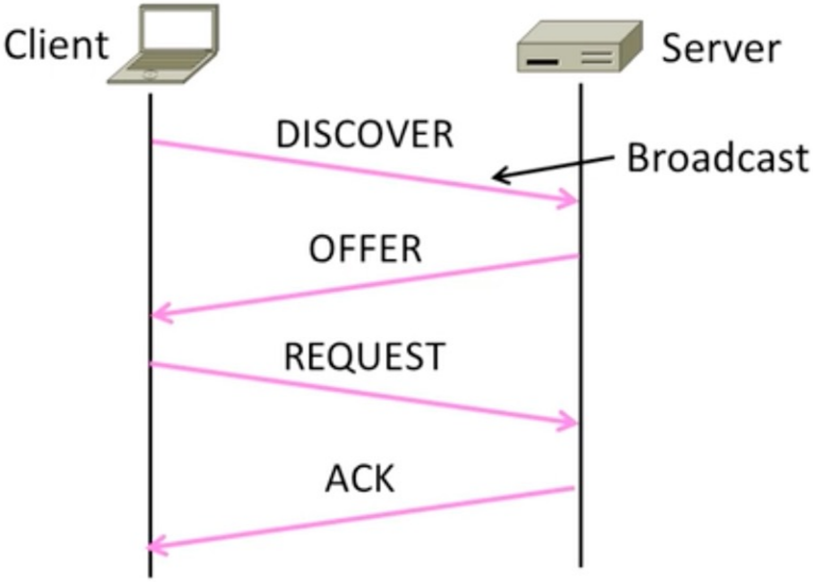
DHCP Addressing

- Bootstrap issue:
  - How does node send a message to DHCP server before it is configured?

- Answer:
  - Node sends broadcast messages that delivered to all nodes on the network
  - Broadcast address is all 1s (IP: 255.255.255.255) (Ethernet: ff:ff:ff:ff:ff:ff)

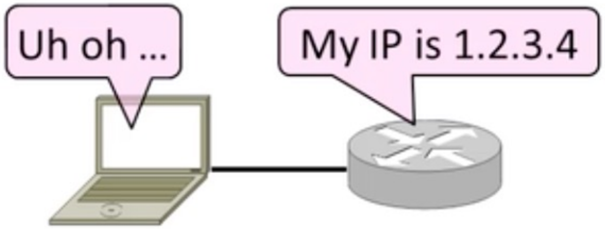
DHCP Messages

- To renew an existing lease, an abbreviated sequence is used:
  - REQUEST, followed by ACK



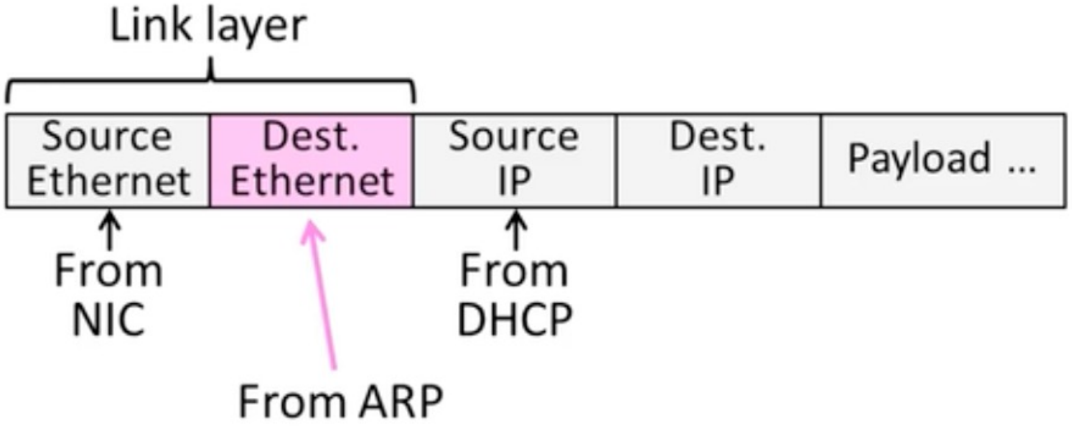
Sending an IP Packet

- Problem:
  - A node needs link layer addresses to send a frame over the local link
  - How does it get the destination link address from a destination IP address?



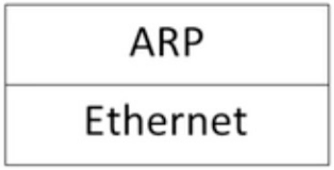
ARP (Address Resolution Protocol)

- Node uses to map a local IP address to its link layer addresses



ARP Protocol Stack

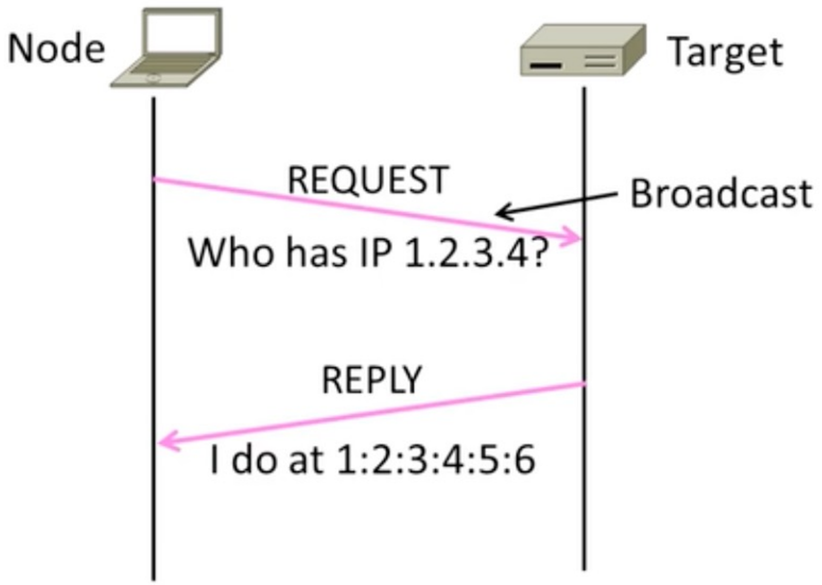
- ARP sits right on top of link layer
  - No servers, just asks node with target IP to identify itself
  - Uses broadcast to reach all nodes



ARP

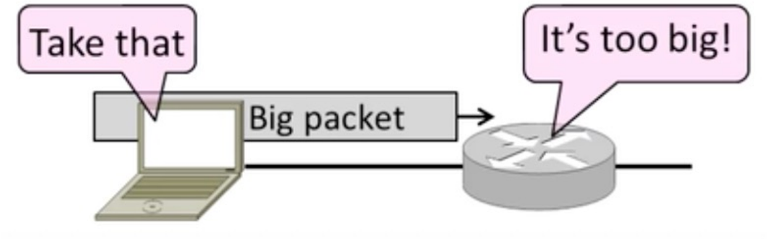
Ethernet

ARP Messages



Packet Size Problem

- Different networks have different maximum packet sizes (MTU)
- How do we connect networks with different maximum packet sizes?
  - Need to split up packets or discover the largest size to use



- Prefer large packets for efficiency
  - But what size is too large?
  - Difficult because node doesn't know complete network path

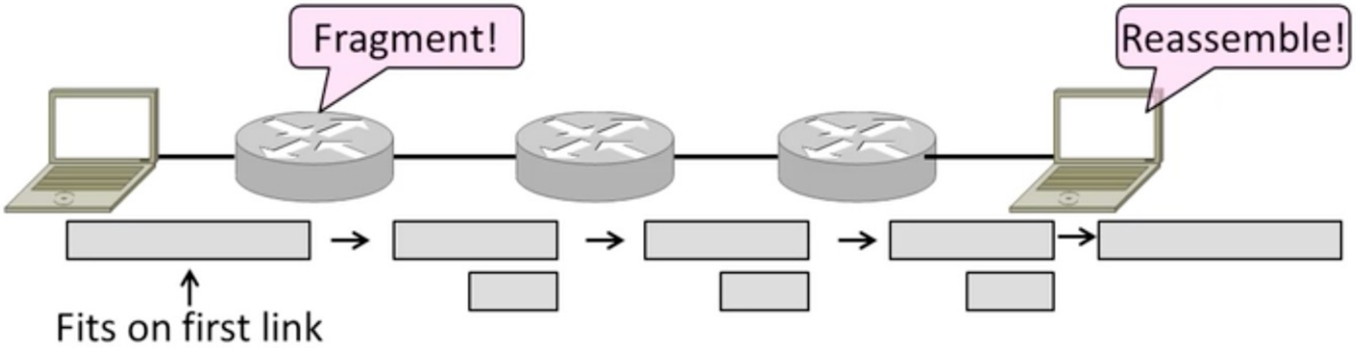
Packet Size Solution

- Fragmentation
  - Split up large packets in the network if they are too large to send
  - Classic method, dated

- Discovery
  - Find the largest packet that fits on the network path and use it
  - IP uses today instead of fragmentation

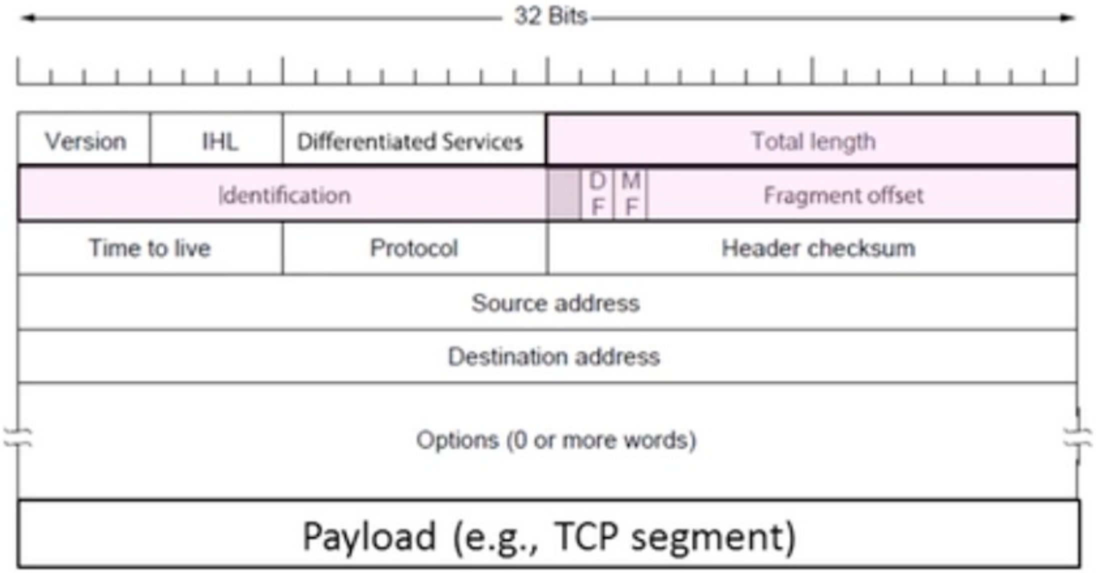
IPv4 Fragmentation

- Routers fragment packets that are too large to forward
- Receiving host reassembles to reduce load on routers



IPv4 Fragmentation Fields

- Header fields used to handle packet size differences
    - Identification, Fragment offset, MF/DF control bits

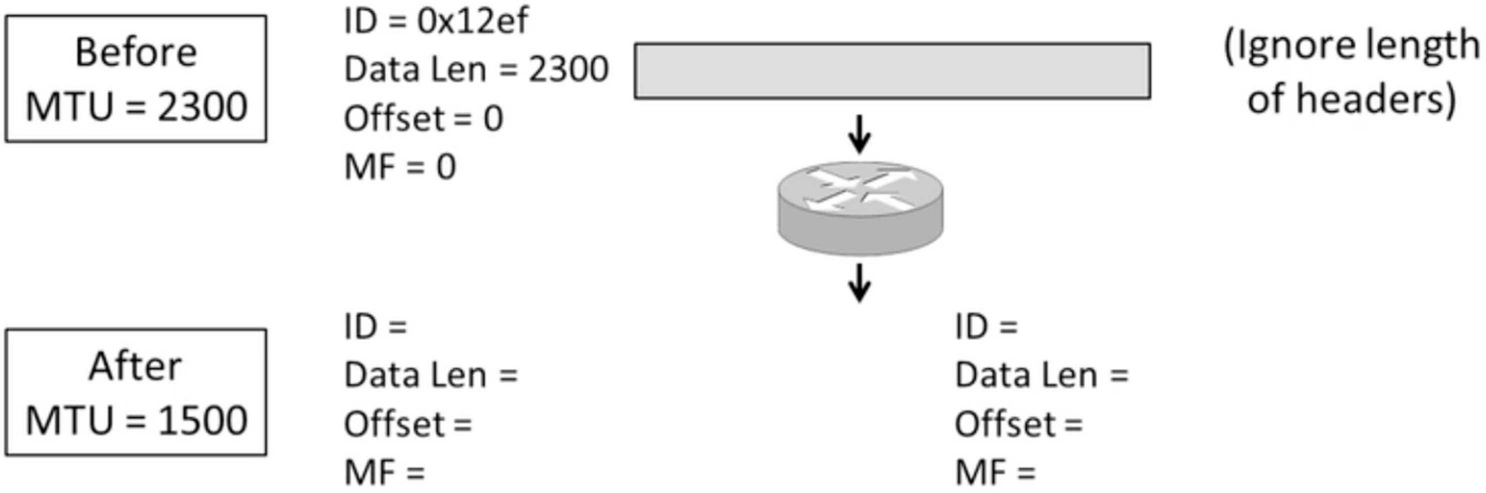


IPv4 Fragmentation Procedure

- Routers split a packet that is too large:
  - Typically break into large pieces
  - Copy IP header to pieces
  - Adjust length on pieces
  - Set offset to indicate position
  - Set MF (More Fragments on all pieces except last

- Receiving hosts reassembles the pieces:
  - Identification field links pieces together, MF tells if it has all pieces

IPv4 Fragmentation (2)



Before
MTU = 2300

ID = 0x12ef
Data Len = 2300
Offset = 0
MF = 0

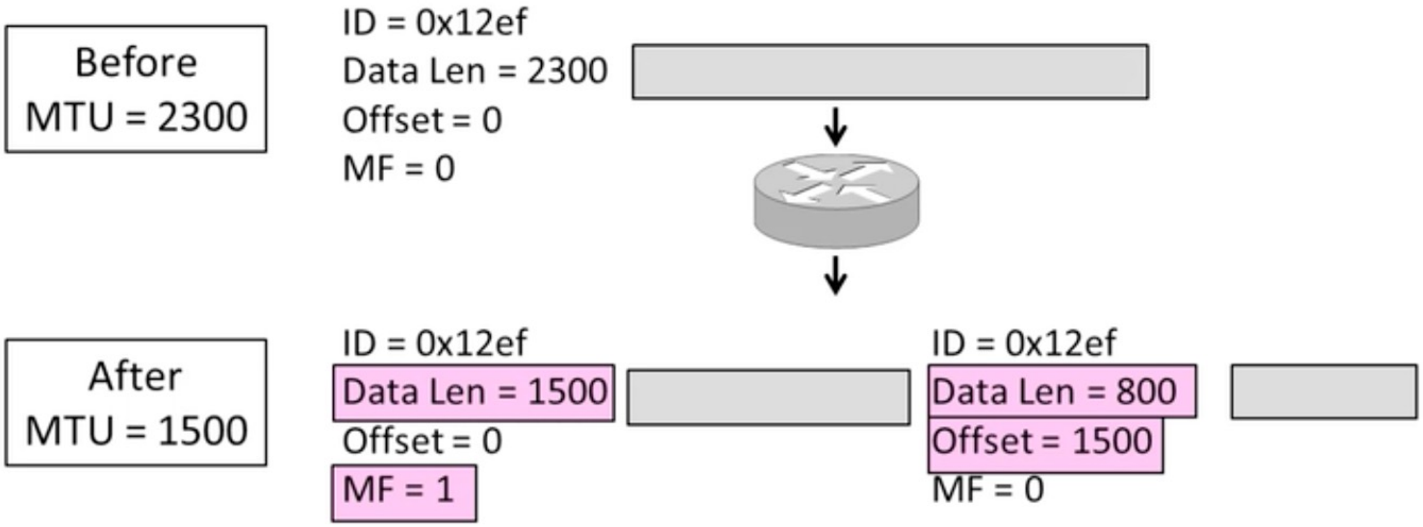
(Ignore length of headers)

After
MTU = 1500

ID =
Data Len =
Offset =
MF =

ID =
Data Len =
Offset =
MF =

IPv4 Fragmentation (3)



Before
MTU = 2300

ID = 0x12ef
Data Len = 2300
Offset = 0
MF = 0

After
MTU = 1500

ID = 0x12ef
Data Len = 1500
Offset = 0
MF = 1

ID = 0x12ef
Data Len = 800
Offset = 1500
MF = 0

IPv4 Fragmentation (4)

- It works!
  - Allow repeated fragmentation

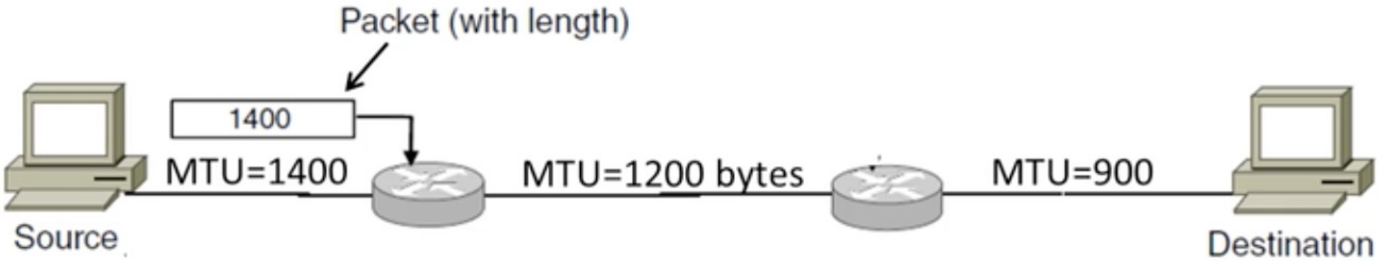
- But fragmentation is undesirable
  - More work for routers, hosts
  - Tends to magnify loss rate
  - Security vulnerabilities too

Path MTU Discovery

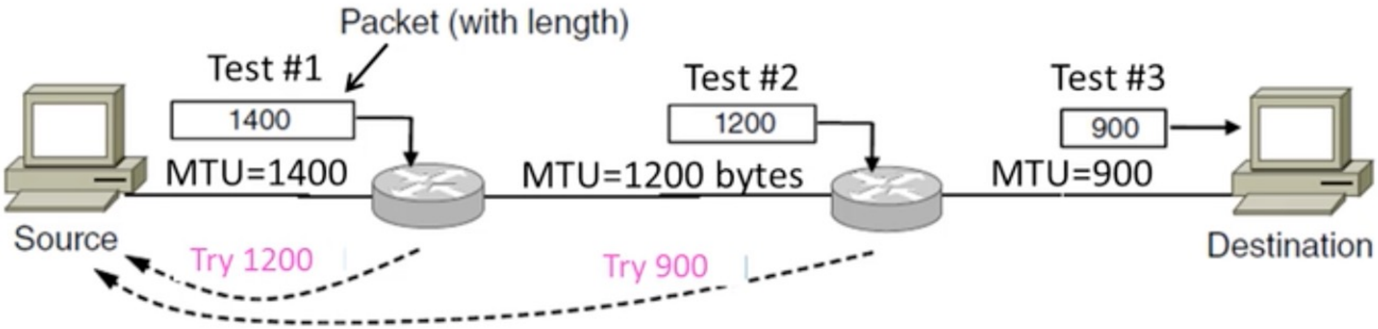
- Discover the MTU that will fit
  - So we can avoid fragmentation
  - The method in use today

- Host tests path with large packet
  - Routers provide feedback if too large; They tell host what size would have fit

Path MTU Discovery (2)



Path MTU Discovery (3)



Path MTU Discovery (4)

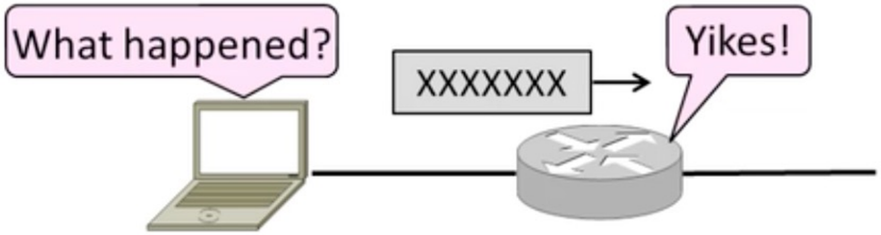
- Process may seem involved
  - But usually quick to find right size

- Path MTU depends on the path and so can change over time
  - Search is ongoing

- Implemented with ICMP
  - Set DF (Don’t Fragment) bit in IP header to get feedback messages

Error Handling

- What happens when something goes wrong during forwarding?
  - Need to be able to find the problem



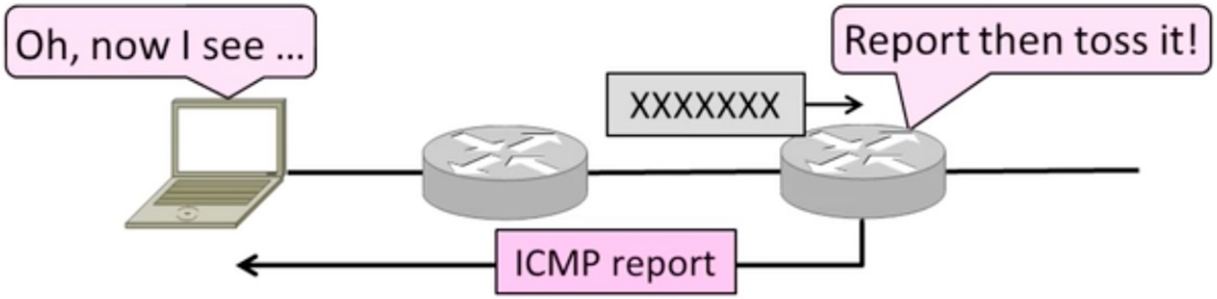
Internet Control Message Protocol

- ICMP is a companion protocol to IP
    - They are implemented together
    - Sits on top of IP (IP protocol=1)

- Provides error report and testing
    - Error is at router while forwarding
    - Also, testing that hosts can use

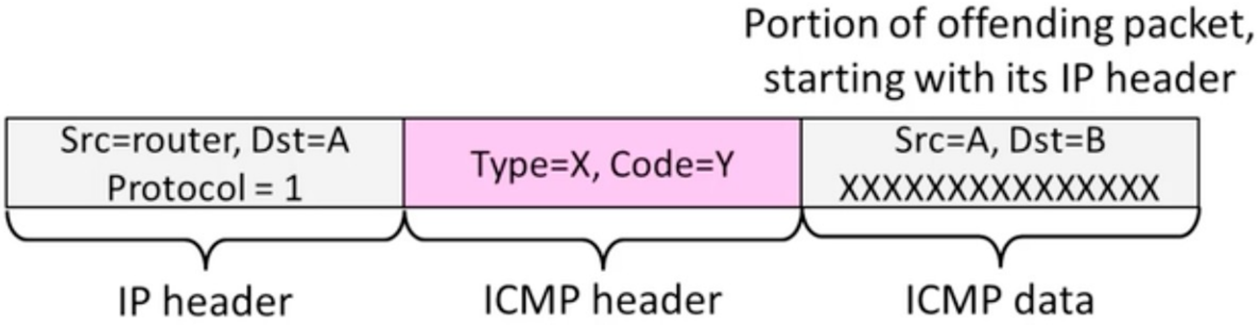
ICMP Errors

- When router encounters an error while forwarding:
  - It sends an ICMP error report back to the IP source address
  - It discards the problematic packet; host needs to rectify

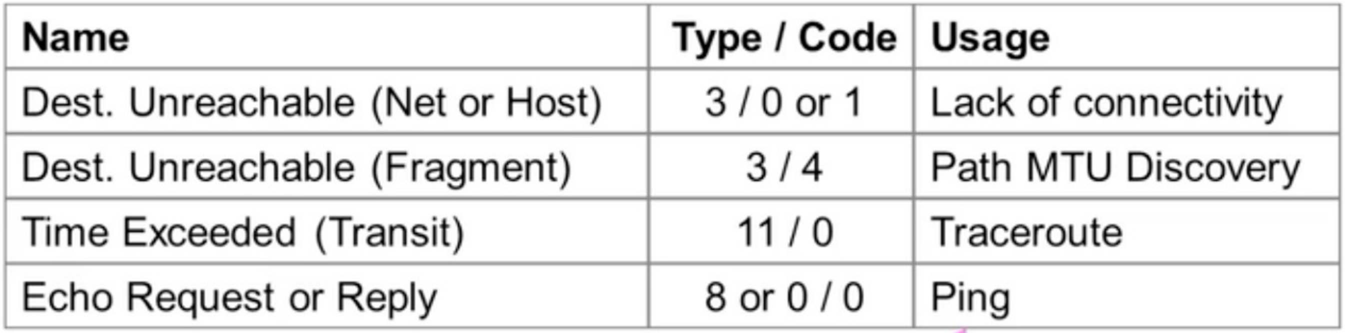


ICMP Message Format

- Each ICMP message has a type, code and checksum
- Often carry the start of the offending packet as payload
- Each message is carried in an IP packet



Example ICMP Message

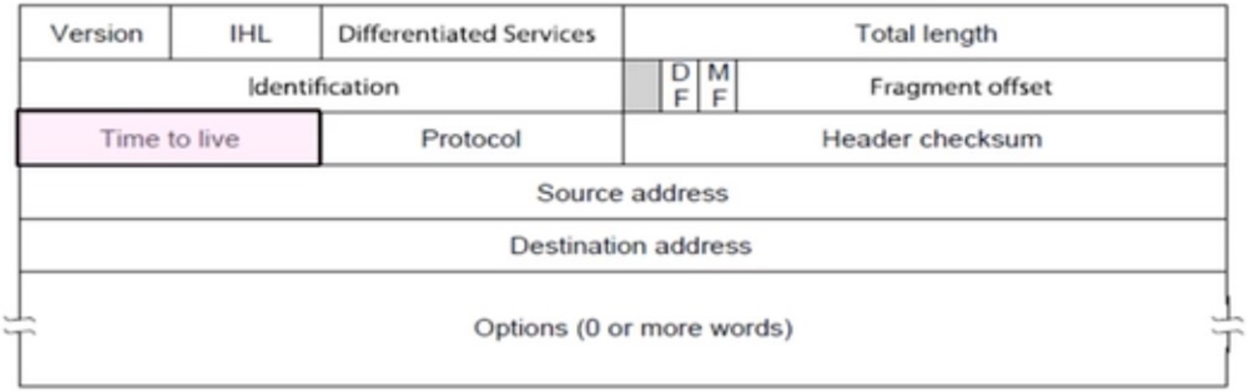


Name	Type / Code	Usage
Dest. Unreachable (Net or Host)	3 / 0 or 1	Lack of connectivity
Dest. Unreachable (Fragment)	3 / 4	Path MTU Discovery
Time Exceeded (Transit)	11 / 0	Traceroute
Echo Request or Reply	8 or 0 / 0	Ping

Testing, not a forwarding error: Host sends Echo Request, and destination responds with an Echo Reply

Traceroute

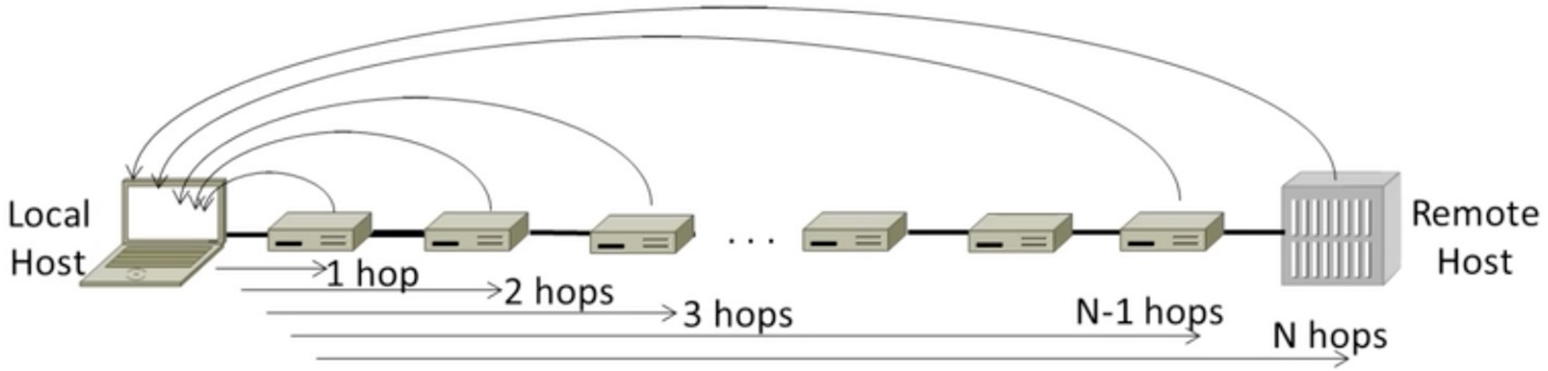
- IP header contains TTL (Time To Live) field
  - Decremented every router hop, with ICMP error if it hits zero
  - Protects against forwarding loops



<table>
<tr><td>Version</td><td>IHL</td><td>Differentiated Services</td><td colspan="4">Total length</td></tr>
<tr><td colspan="3">Identification</td><td></td><td>D
F</td><td>M
F</td><td>Fragment offset</td></tr>
<tr><td colspan="2">Time to live</td><td>Protocol</td><td colspan="4">Header checksum</td></tr>
<tr><td colspan="7">Source address</td></tr>
<tr><td colspan="7">Destination address</td></tr>
<tr><td colspan="7">Options (0 or more words)</td></tr>
</table>

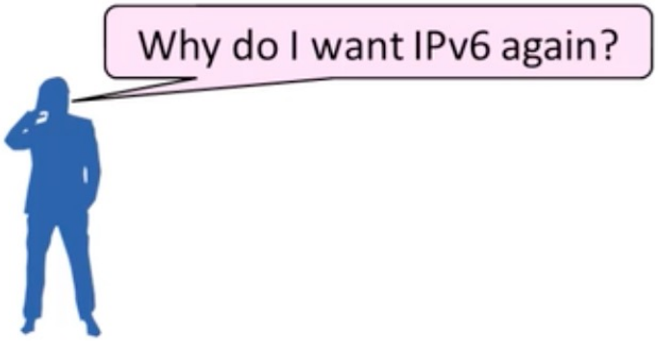
Traceroute (2)

- Traceroute repurposes TTL and ICMP functionality
  - Sends probe packets increasing TTL starting from 1
  - ICMP errors identify routers on the path



Internet Growth

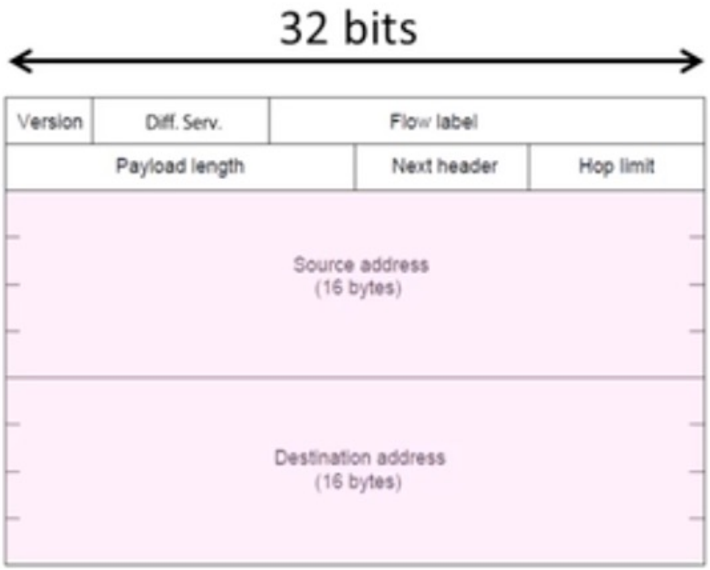
- At least a billion Internet hosts and growing …
    - And we’re using 32-bit addresses!
- IP version 6, the future of IPv4 that is now (still) being deployed



IPv6

- Feature large addresses
  - 128 bits, most of header

- New notation
  - 8 groups of 4 hex digits (16 bits)
  - Omit leading zeros, groups of zeros



  - EX: 2001:0db8:0000:0000:0000:ff00:0042:8329
  - ->2001:db8::ff00:42:8329

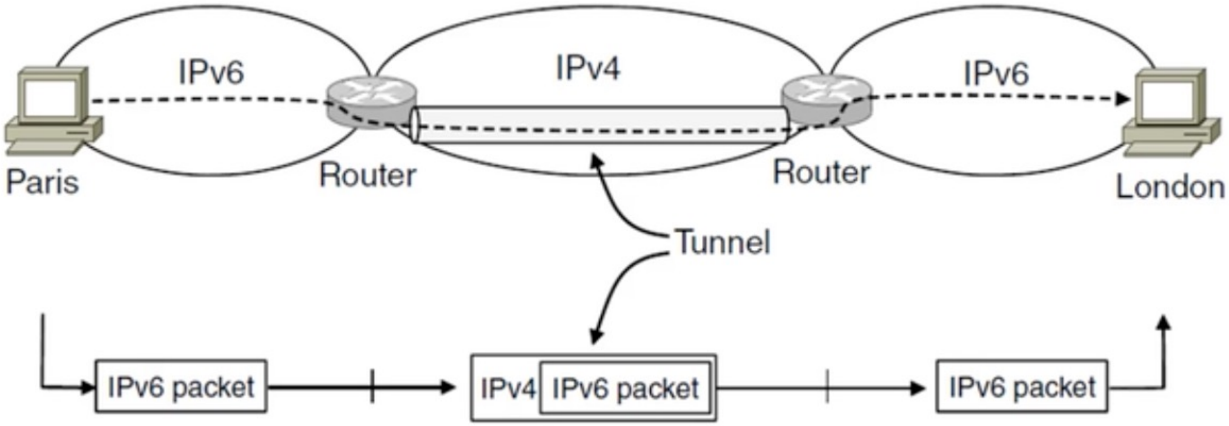
IPv6 Transition

- The big problem:
    - How to deploy IPv6?
    - Fundamentally incompatible with IPv6?

- Dozens of approaches proposed
    - Dual stack (speak IPv4 and IPv6)
    - Translators (convert packets)
    - Tunnels (carry IPv6 over IPv4)

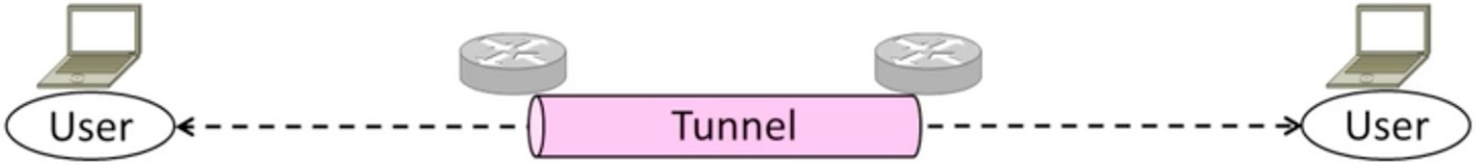
Tunneling

- Native IPv6 islands connected via IPv4
  - Tunnel carries IPv6 packets across IPv4 network



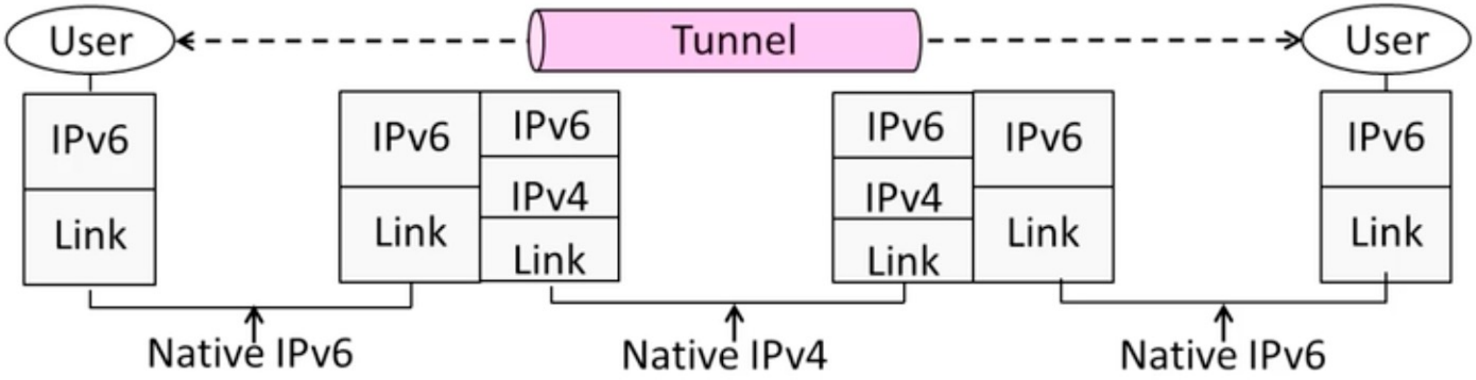
Tunneling (2)

- Tunnel acts as a single link across IPv4 network



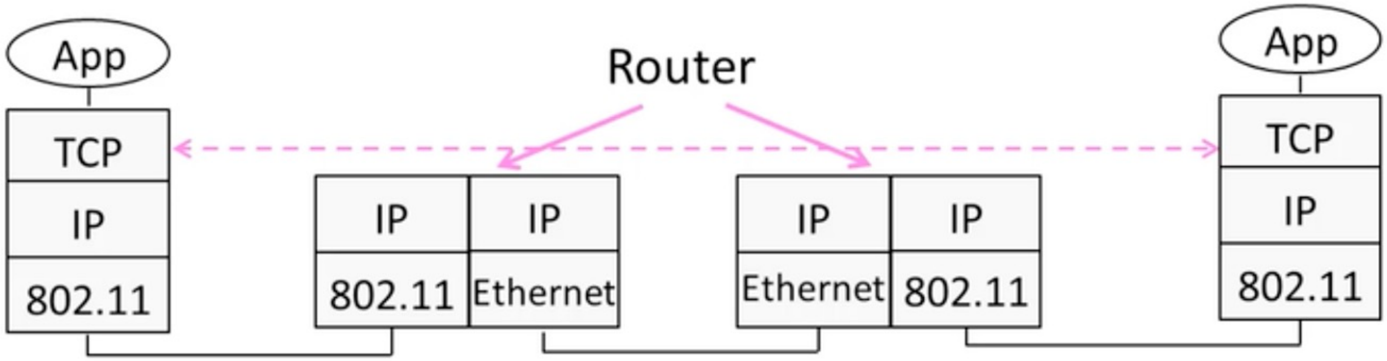
Tunneling (3)

- Tunnel acts as a single link across IPv4 network



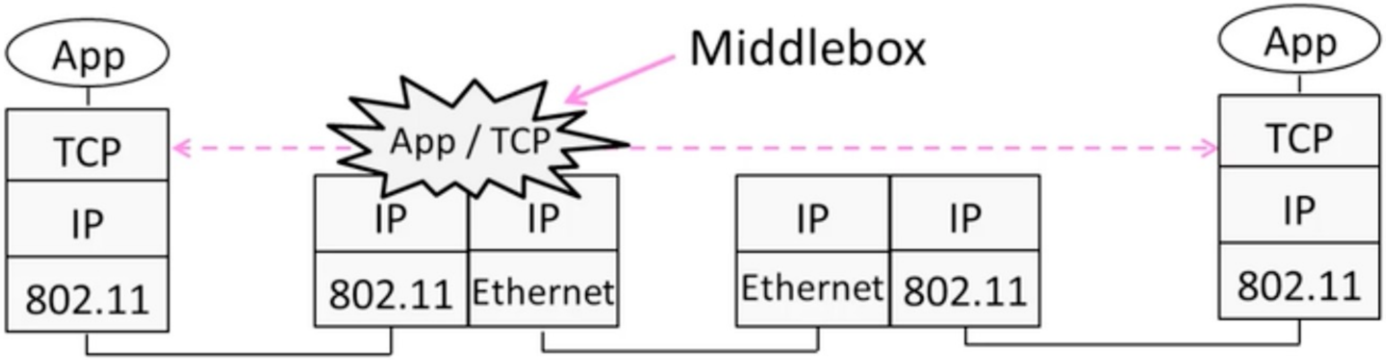
Layering Review

- Remember how layering is meant to work?
  - "Routers don't look beyond the IP header." Well …



Middleboxes

- Sit "inside the network" but perform "more than IP" processing on packets to add new functionality
  - NAT box, Firewall / Intrusion Detection System



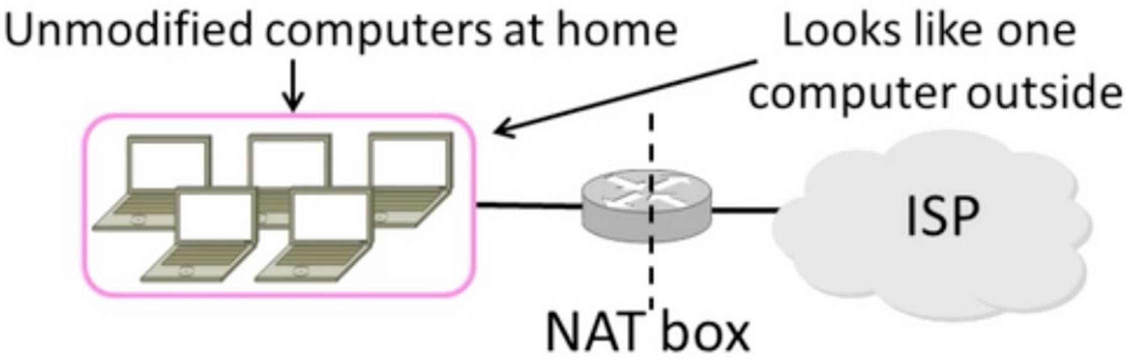
What is NAT (Network Address Translation)

- NAT box connects an internal network to an external network
  - Many internal hosts are connected using few external addresses
  - Middlebox that “translates addresses”

- Motivated by IP address scarcity
  - Controversial at first, now accepted

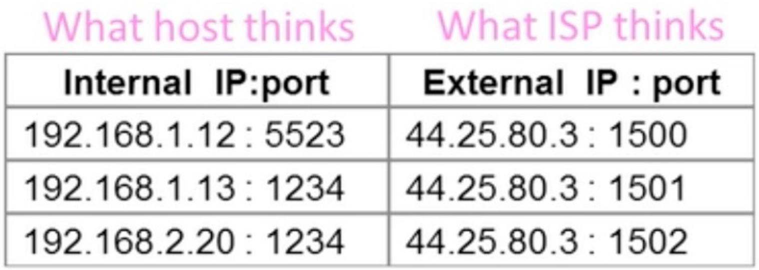
NAT (2)

- Common scenario:
  - Home computers use “private” IP addresses
  - NAT (in AP/firewall) connects home to ISP using a single external IP address



How NAT Works?

- Keeps an internal/external table
  - Typically uses IP address + TCP port
  - This is address and port translation

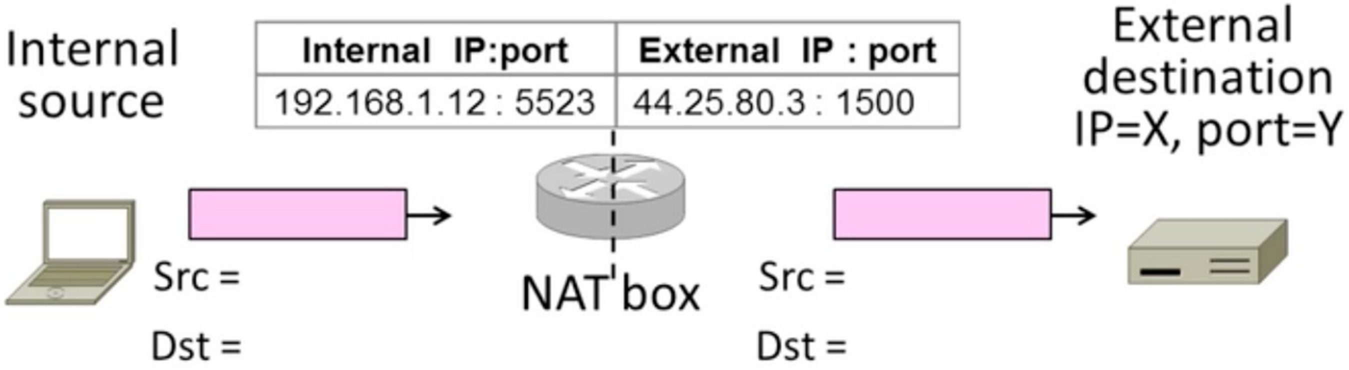


What host thinks	What ISP thinks
Internal IP:port	**External IP : port**
192.168.1.12 : 5523	44.25.80.3 : 1500
192.168.1.13 : 1234	44.25.80.3 : 1501
192.168.2.20 : 1234	44.25.80.3 : 1502

- Need ports to make mapping 1-1 since there are fewer external IPs

How NAT Works? (2)

- Internal -> External:
  - Look up and rewire source IP/port



Internal source

Internal IP:port	External IP : port
192.168.1.12 : 5523	44.25.80.3 : 1500

External destination IP=X, port=Y

Src =

Dst =

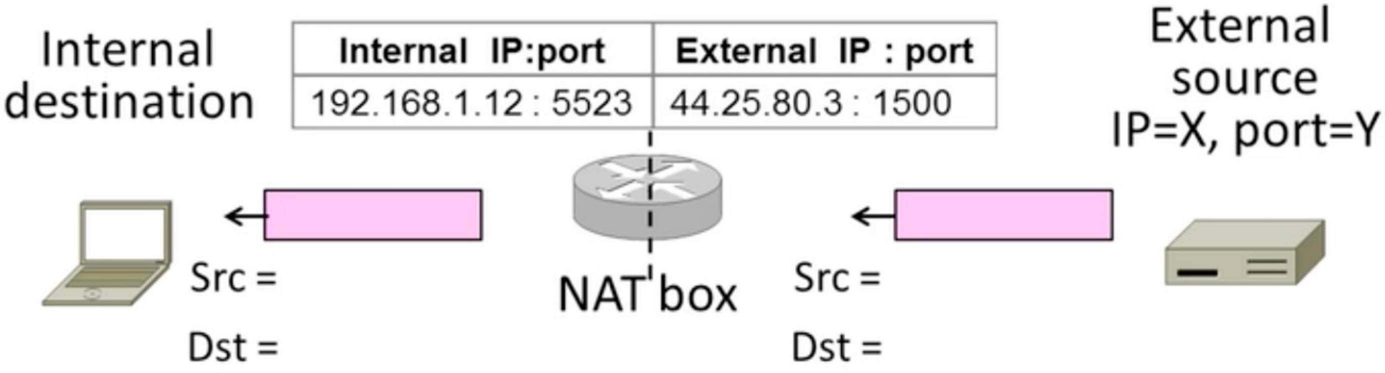
NAT box

Src =

Dst =

How NAT Works? (3)

- External -> Internal:
  - Look up and rewire destination IP/port



Internal IP:port	External IP : port
192.168.1.12 : 5523	44.25.80.3 : 1500

NAT Downsides

- Connectivity has been broken!
  - Can only send incoming packets after an outgoing connection is set up
  - Difficult to run servers or peer-to-peer apps (Skype) at home
- Doesn’t work so well when there are no connections (UDP apps)
- Breaks apps that unwisely expose their IP addresses (FTP)

NAT Upsides

- Relieves much IP address pressure
  - Many home hosts behind NATs
- Easy to deploy
  - Rapidly, and by you alone
- Useful functionality
  - Firewall, helps with privacy

To do

- Quiz next week
- Research reference due March 1st
- Lab1 due March 15th