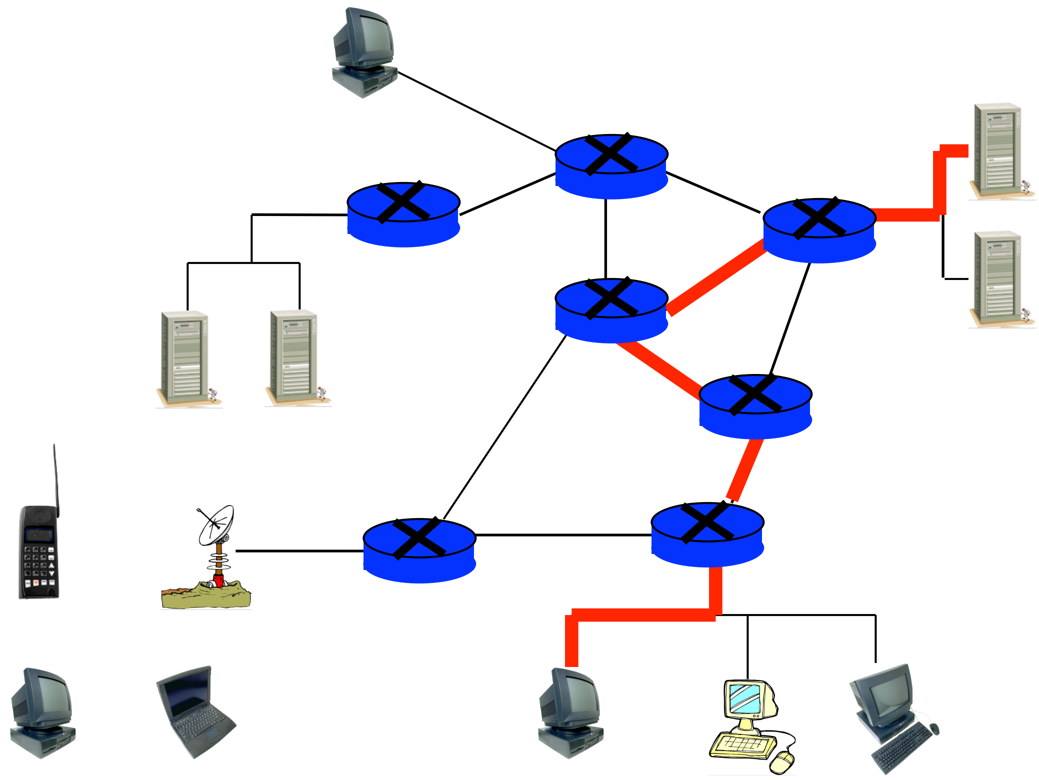
Networking Basics

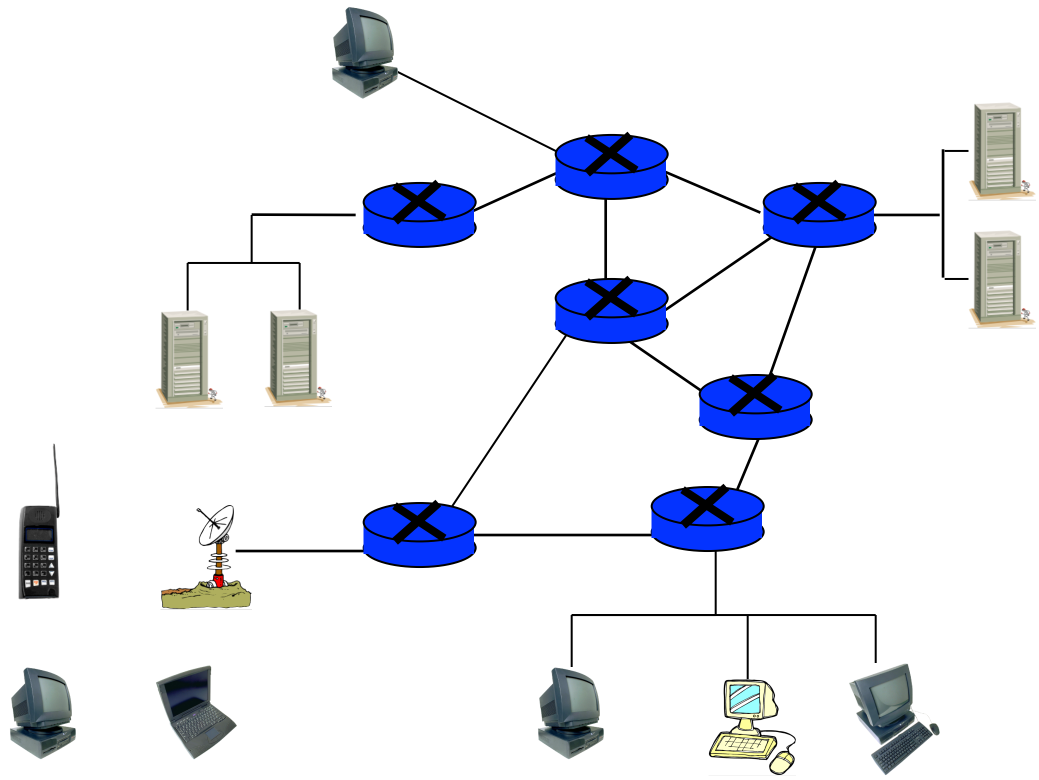
Network

- Includes
  - Computers
  - Servers
  - Routers
  - Wireless devices
  - Etc.
- Purpose is to transmit data



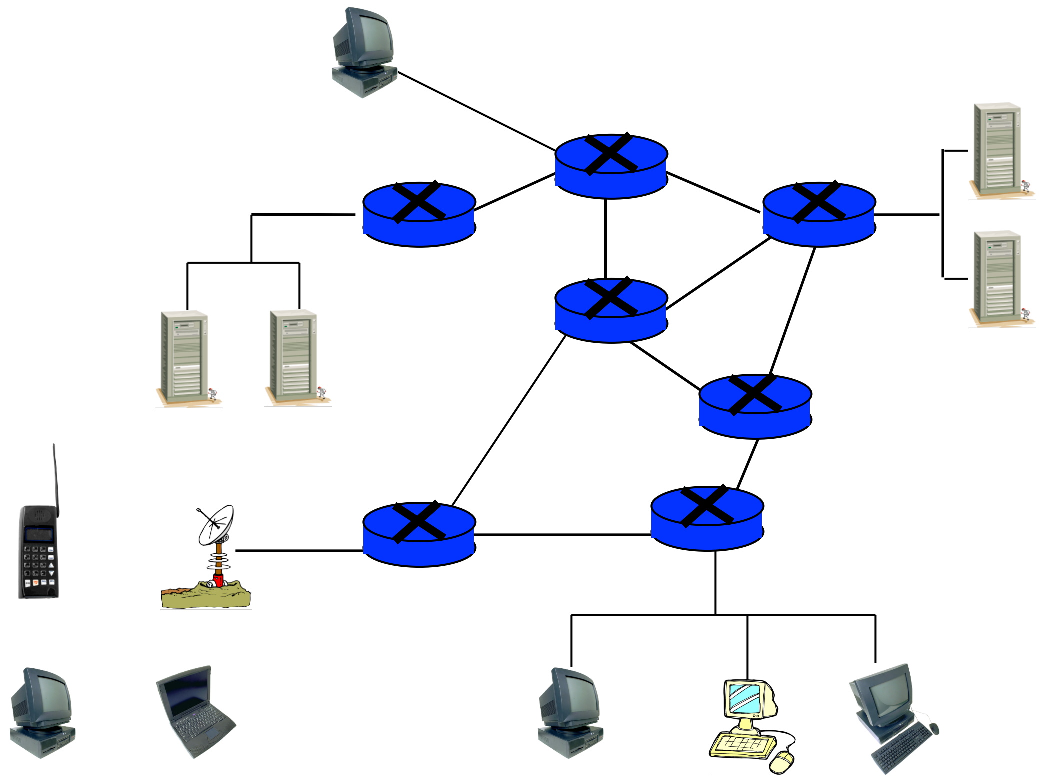
Network Edge

- Network **edge** includes
- Hosts
  - Computers
  - Laptops
  - Servers
  - Cell phones
  - Etc., etc.



Network Core

- Network **core** consists of
  - Interconnected mesh of routers
- Purpose is to move data from host to host



Packet Switched Network

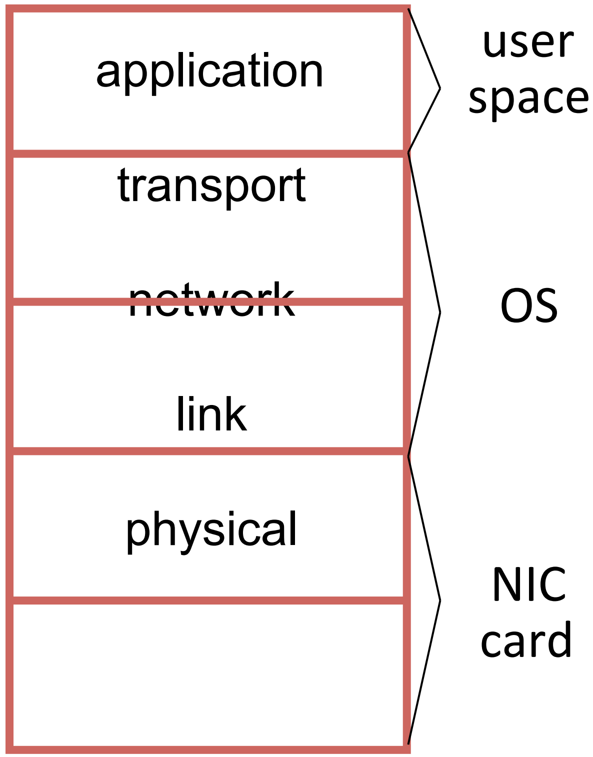
- Telephone network is/was **circuit switched**
  - For each call, a dedicated circuit established
  - Dedicated bandwidth
- Modern data networks are **packet switched**
  - Data is chopped up into discrete packets
  - Packets are transmitted independently
  - No dedicated circuit is established
  - More efficient bandwidth usage
  - But more complex than circuit switched

Network Protocols

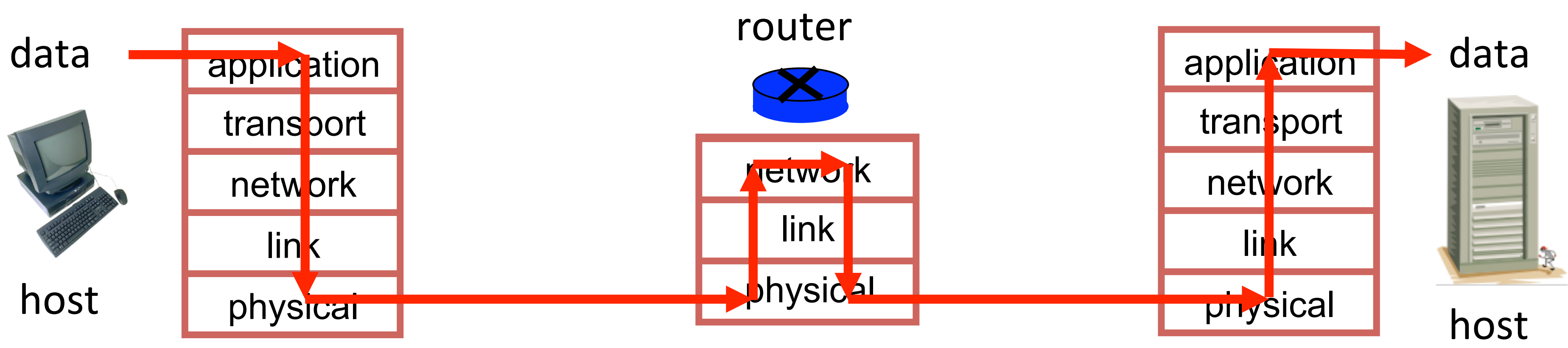
- Study of networking focused on **protocols**
- Networking protocols precisely specify "communication rules"
- Details are given in **RFC**s
  - RFC is essentially an Internet standard
- **Stateless** protocols don't remember
- **Stateful** protocols do remember
- Many security problems related to "state"
  - E.g., DoS is a problem with stateful protocols

Protocol Stack

- Application layer protocols
  - HTTP, FTP, SMTP, etc.
- Transport layer protocols
  - TCP, UDP
- Network layer protocols
  - IP, routing protocols
- Link layer protocols
  - Ethernet, PPP
- Physical layer



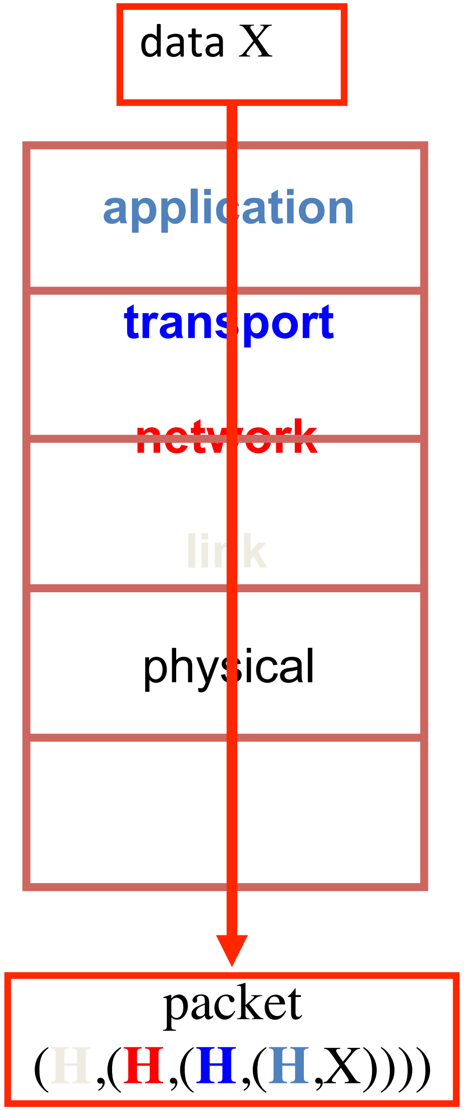
Layering in Action



- At source, data goes “down” the protocol stack
- Each router processes packet “up” to network layer
  - That’s where routing info lives
- Router then passes packet down the protocol stack
- Destination processes up to application layer
  - That’s where the data lives

Encapsulation

- X = application data at source
- As X goes down protocol stack, each layer adds header information:
  - Application layer: (**H**, X)
  - Transport layer: (**H**, (**H**, X))
  - Network layer: (**H**, (**H**, (**H**, X)))
  - Link layer: (**H**, (**H**, (**H**, (**H**, X))))
- Header has info required by layer
- Note that app data is on the inside



Application Layer

- Applications
  - Web browsing, email, P2P, etc.
  - Running on hosts
  - Hosts want network to be transparent
- Application layer protocols
  - HTTP, SMTP, IMAP, Gnutella, etc.
- Protocol is only one part of an application
  - For example, HTTP only a part of web browsing

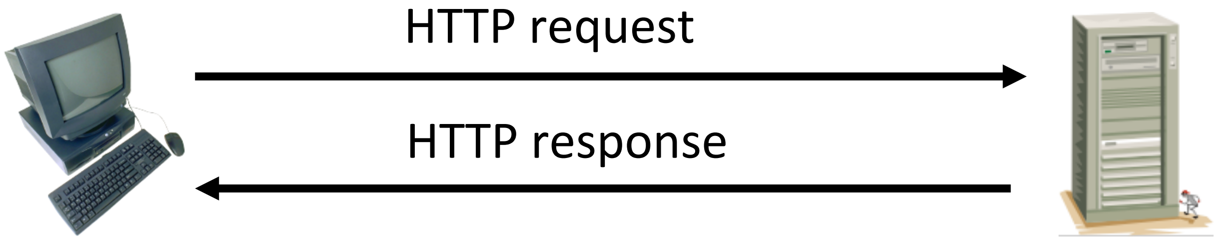
Client-Server Model

- **Client**
  – "speaks first"
- **Server**
  – tries to respond to request
- Hosts are clients and/or servers
- Example: Web browsing
  – You are the client (request web page)
  – Web server is the server

Peer-to-Peer Model

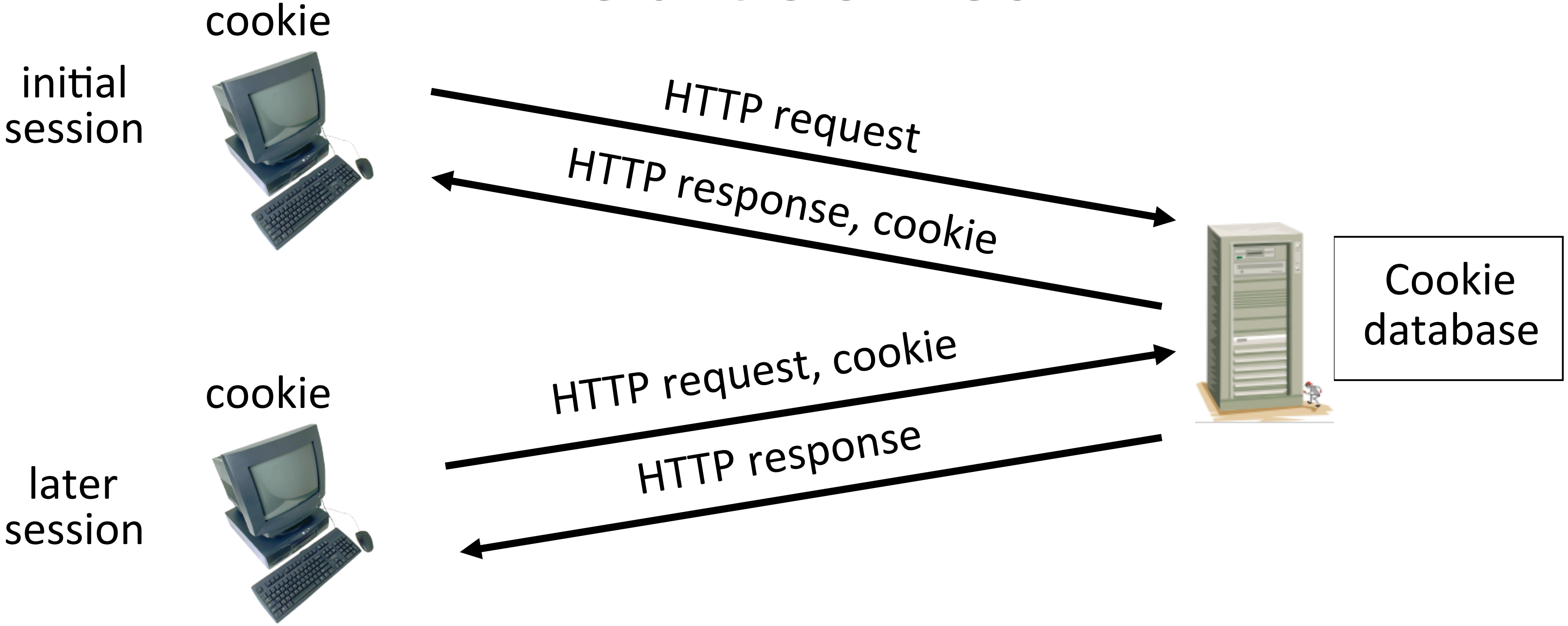
- Hosts act as clients and servers
- For example, when sharing music
  - You are client when requesting a file
  - You are a server when someone downloads a file from you
- In P2P, how does client find server?
  - Many different P2P models for this

HTTP Example



- HTTP --- **H**yper**T**ext **T**ransfer **P**rotocol
- Client (you) requests a web page
- Server responds to your request

Web Cookies



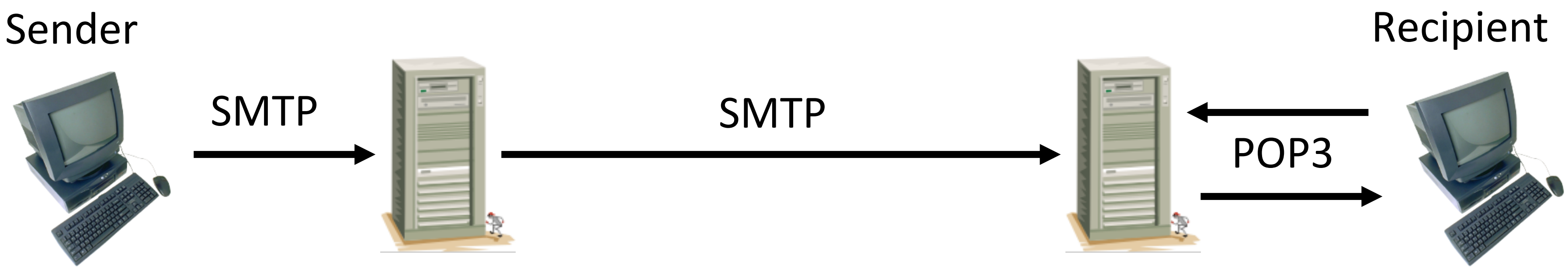
- HTTP is stateless — cookies used to add state
- Initially, cookie sent from server to browser
- Browser manages cookie, sends it to server
- Server looks in cookie database to “remember” you

Web Cookies

- Web cookies used for…
  - Shopping carts
  - Recommendations, etc., etc.
  - A very, very weak form of authentication
- Privacy concerns
  - Web site can learn a lot about you
  - Multiple web sites could learn even more

SMTP

- SMTP used to send email from sender to recipient's mail server
- Then use POP3, IMAP or HTTP (Web mail) to get messages from server
- As with many application protocols, SMTP commands are human readable



Spoofed email with SMTP

User types the red lines:

```
> telnet eniac.cs.sjsu.edu 25
220 eniac.sjsu.edu
HELO ca.gov
250  Hello ca.gov, pleased to meet you
MAIL FROM: <arnold@ca.gov>
250 arnold@ca.gov... Sender ok
RCPT TO: <stamp@cs.sjsu.edu>
250 stamp@cs.sjsu.edu ... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
It is my pleasure to inform you that you
are terminated
 .
250 Message accepted for delivery
QUIT
221 eniac.sjsu.edu closing connection
```

Application Layer

- DNS --- Domain Name Service
  - Convert human-friendly names such as www.google.com into 32-bit IP address
  - A distributed hierarchical database
- Only 13 "root" DNS server clusters
  - Almost a single point of failure for Internet
  - Attacks on root servers have succeeded
  - But, attacks have not lasted long enough

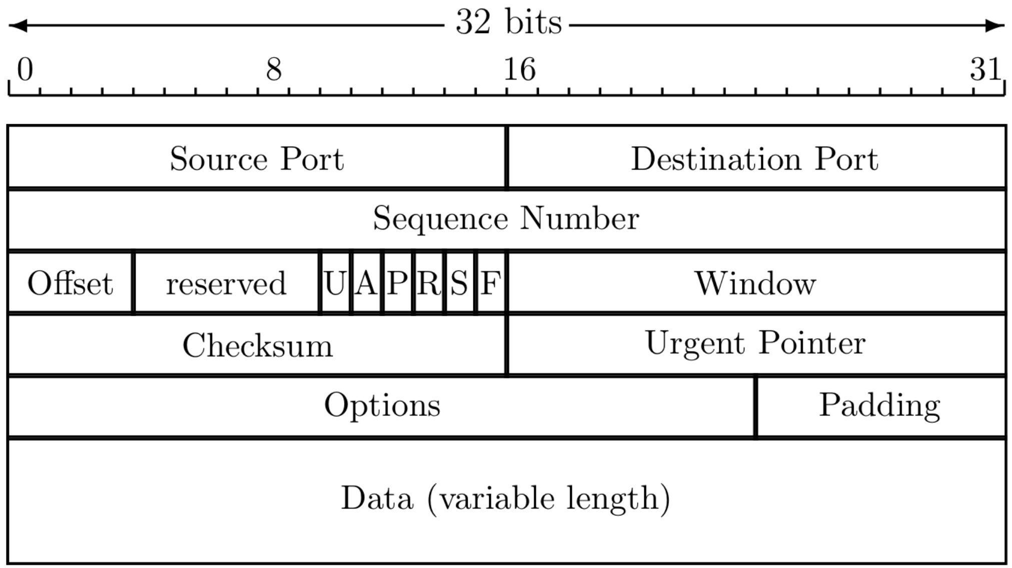
Transport Layer

- The network layer offers unreliable, “best effort” delivery of packets
- Any improved service must be provided by the hosts
- Transport layer: two protocols of interest
  - TCP — more service, more overhead
  - UDP — less service, less overhead
- TCP and UDP runs on hosts, not routers

TCP

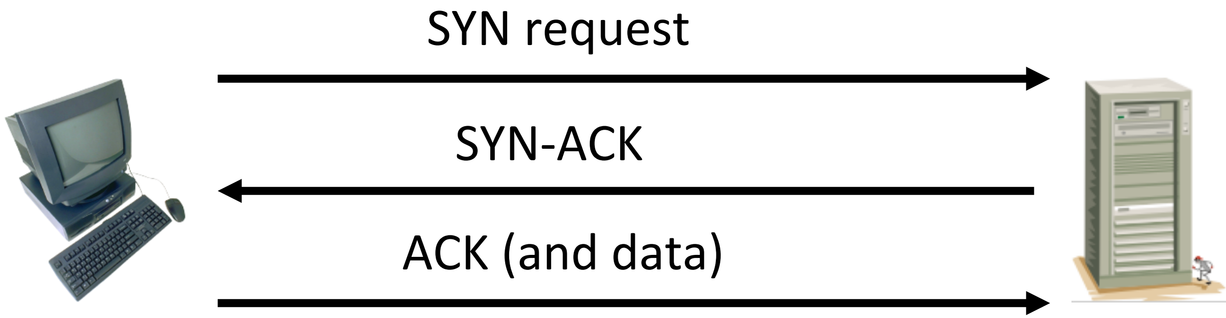
- TCP assures that packets…
  - Arrive at destination
  - Are processed in order
  - Are not sent too fast for receiver: **flow control**
- TCP also provides…
  - Network-wide **congestion control**
- TCP is **connection-oriented**
  - TCP contacts server before sending data
  - Orderly setup and take down of “connection”
  - No true connection, only a logical connection

TCP Header



- Source and destination port
- Sequence number
- Flags (ACK, SYN, RST, etc.)
- Usually 20 bytes (if no options)

TCP Three-Way Handshake



- **SYN**: synchronization requested
- **SYN-ACK**: acknowledge SYN request
- **ACK**: acknowledge msg 2 and send data
- Then TCP "connection" established
  - Connection terminated by FIN or RST

Denial of Service Attack

- The TCP 3-way handshake makes denial of service (DoS) attacks possible
- Whenever SYN packet is received, server must remember “half-open” connection
  - Remembering consumes resources
  - Too many half-open connections and server’s resources will be exhausted, and then…
  - …server can’t respond to legitimate connections

UDP

- UDP is minimalist, “no frills” service
  - No assurance that packets arrive
  - No assurance packets are in order, etc., etc.
- Why does UDP exist?
  - More efficient (smaller header)
  - No flow control to slow down sender
  - No congestion control to slow down sender
- Packets sent too fast, they will be dropped
  - Either at intermediate router or at destination
  - But in some apps this is OK (audio/video)

Network Layer

- Core of network/Internet
  - Interconnected mesh of routers
- Purpose of network layer
  - Route packets through this mesh
- Network layer protocol is known as **IP**
  - Follows a **best effort** approach
- IP runs in every host and every router
- Routers also run routing protocols
  - Used to determine the path to send packets
  - Routing protocols: RIP, OSPF, BGP, …

IP Addresses

- **IP address** is 32 bits
- Every host has an IP address
- Not enough IP addresses!
  - Lots of tricks used to extend address space
- IP addresses given in dotted decimal notation
  - For example: 195.72.180.27
  - Each number is between 0 and 255
- Usually, host's IP address can change

Socket

- Each host has a 32 bit IP address
- But many processes on one host
  - You can browse web, send email at same time
- How to distinguish processes on a host?
- Each process has a 16 bit **port number**
  - Port numbers < 1024 are "well-known" ports (HTTP is port 80, POP3 is port 110, etc.)
  - Port numbers above 1024 are dynamic (as needed)
- IP address and port number define a **socket**
  - Socket uniquely identifies process, Internet-wide

Network Address Translation

- Network Address Translation (**NAT**)
- Used to extend IP address space
- Use one IP address, different port numbers, for multiple hosts
  - "Translates" outside packet (based on port number) to IP for inside host

NAT-less Example

Web
server

IP: 12.0.0.1
Port: 80

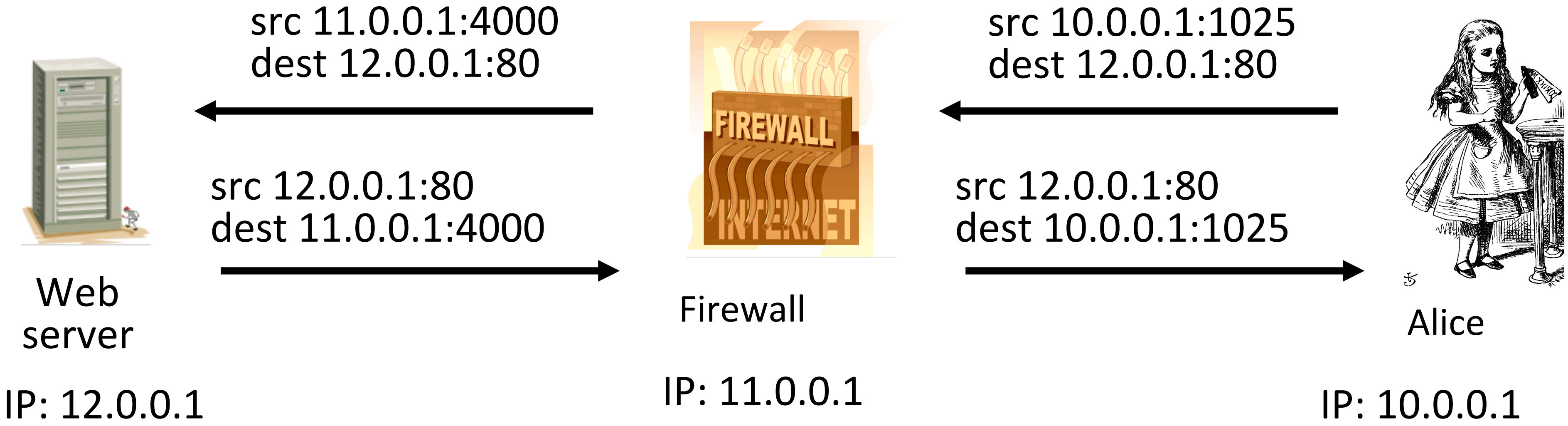
source 11.0.0.1:1025
destination 12.0.0.1:80

source 12.0.0.1:80
destination 11.0.0.1:1025

Alice

IP: 11.0.0.1
Port: 1025

NAT Example

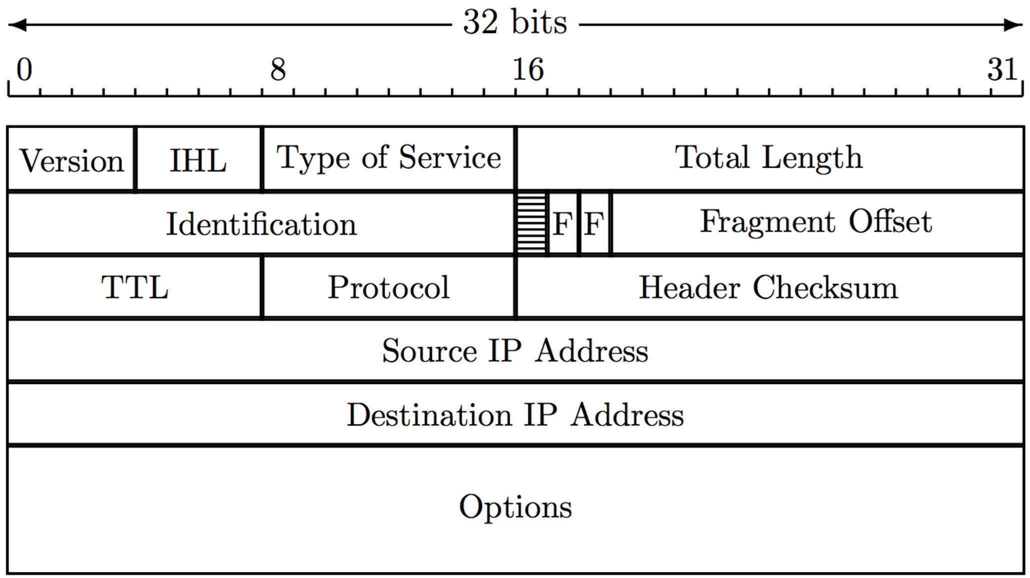


NAT Table	
4000	10.0.0.1:1025

NAT: The Last Word

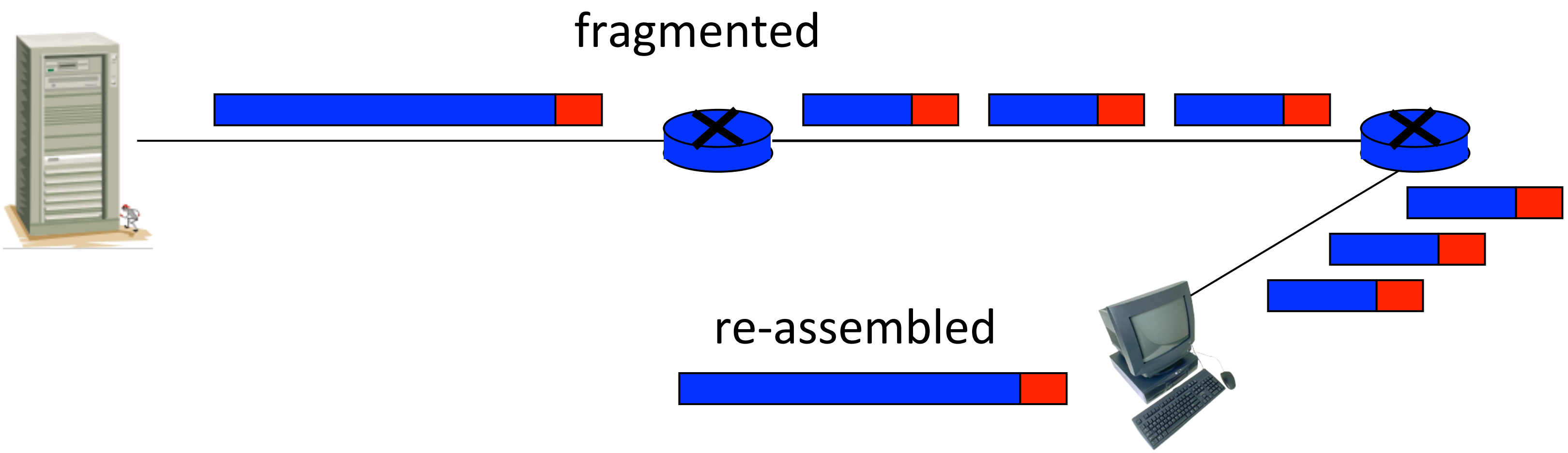
- Advantage(s)?
  - Extends IP address space
  - One (or a few) IP address(es) can be shared by many users
- Disadvantage(s)?
  - Makes end-to-end security difficult
  - Might make IPSec less effective (IPSec discussed in Chapter 10)

IP Header



- IP header used by routers
  - Note source and destination IP addresses
- Time to live (TTL) limits number of "hops"
  - So packets can't circulate forever
- Fragmentation information (see next slide)

IP Fragmentation



- Each link limits maximum size of packets
- If packet is too big, router fragments it
- Re-assembly occurs at destination

IP Fragmentation

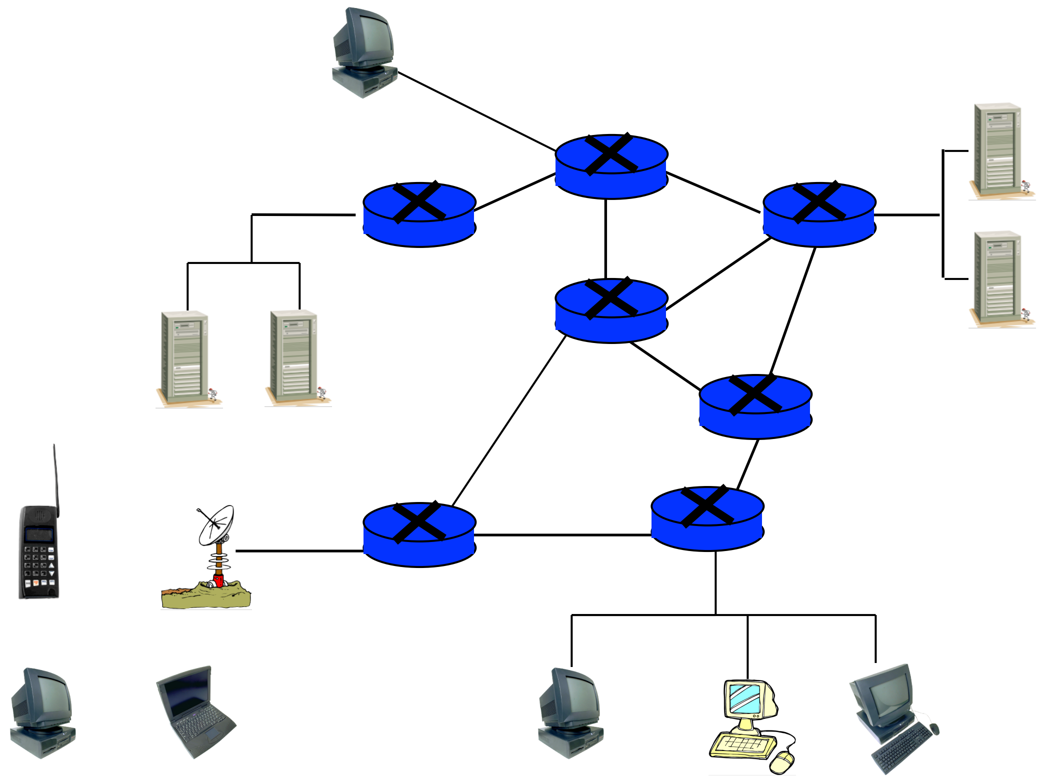
- One packet becomes multiple packets
- Packets reassembled at **destination**
  - Prevents multiple fragmentation/re-assemble
- Fragmentation is a security issue…
  - Fragments may obscure real purpose of packet
  - Fragments can overlap when re-assembled
  - Must re-assemble packet to fully understand it
  - Lots of work for firewalls, for example

IPv6

- Current version of IP is IPv4
- IPv6 is a "new-and-improved" version
- IPv6 is "bigger and better" than IPv4
  - ***Bigger*** addresses: 128 bits
  - ***Better*** security: IPSec
- How to migrate from IPv4 to IPv6?
  - Unfortunately, nobody has a good answer…
- So IPv6 has not taken hold (yet?)

Link Layer

- Link layer sends packet from one node to next
- Links can be different
  - Wired
  - Wireless
  - Ethernet
  - Point-to-point…



Link Layer

- On host, implemented in adapter: Network Interface Card (NIC)
  - Ethernet card, wireless 802.11 card, etc.
  - NIC is “semi-autonomous” device
- NIC is (mostly) out of host’s control
  - Implements both link and physical layers

Ethernet

- Ethernet is a **multiple access** protocol
- Many hosts access a shared media
  - On a local area network, or LAN
- With multiple access, packets can "collide"
  - Data is corrupted and packets must be resent
- How to efficiently deal with collisions in distributed environment?
  - Many possibilities, but ethernet is most popular
- We won't discuss details here…

Link Layer Addressing

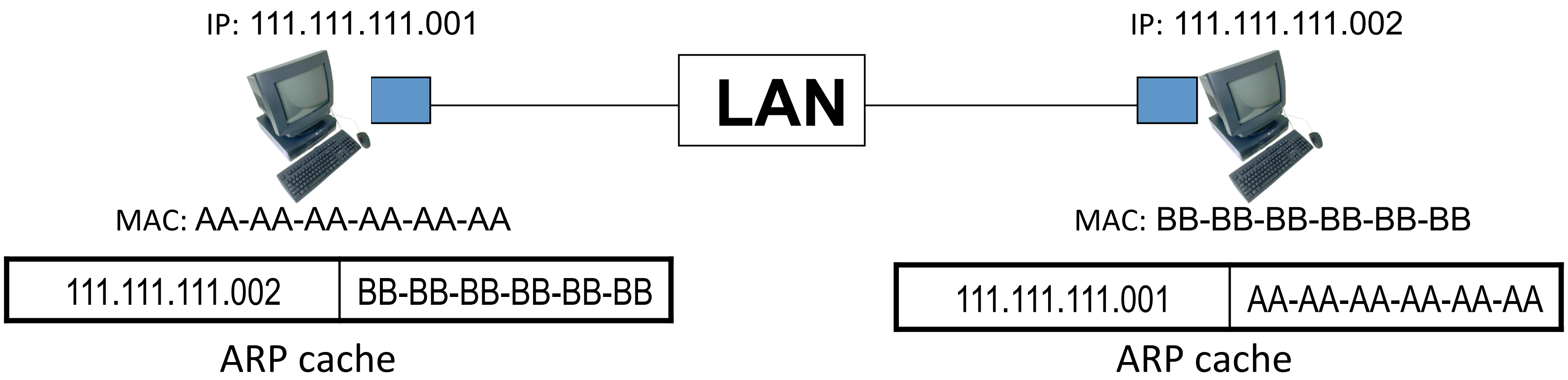
- IP addresses live at network layer
- Link layer also requires addresses (why?)
  - **MAC address** (LAN address, physical address)
- MAC address
  - 48 bits, globally unique
  - Used to forward packets over one link
- Analogy…
  - IP address is like your home address
  - MAC address is like a social security number

ARP

- Address Resolution Protocol (ARP)
- Used by link layer — given IP address, find corresponding MAC address
- Each host has ARP table, or **ARP cache**
  - Generated automatically
  - Entries expire after some time (about 20 min)
  - ARP used to find ARP table entries

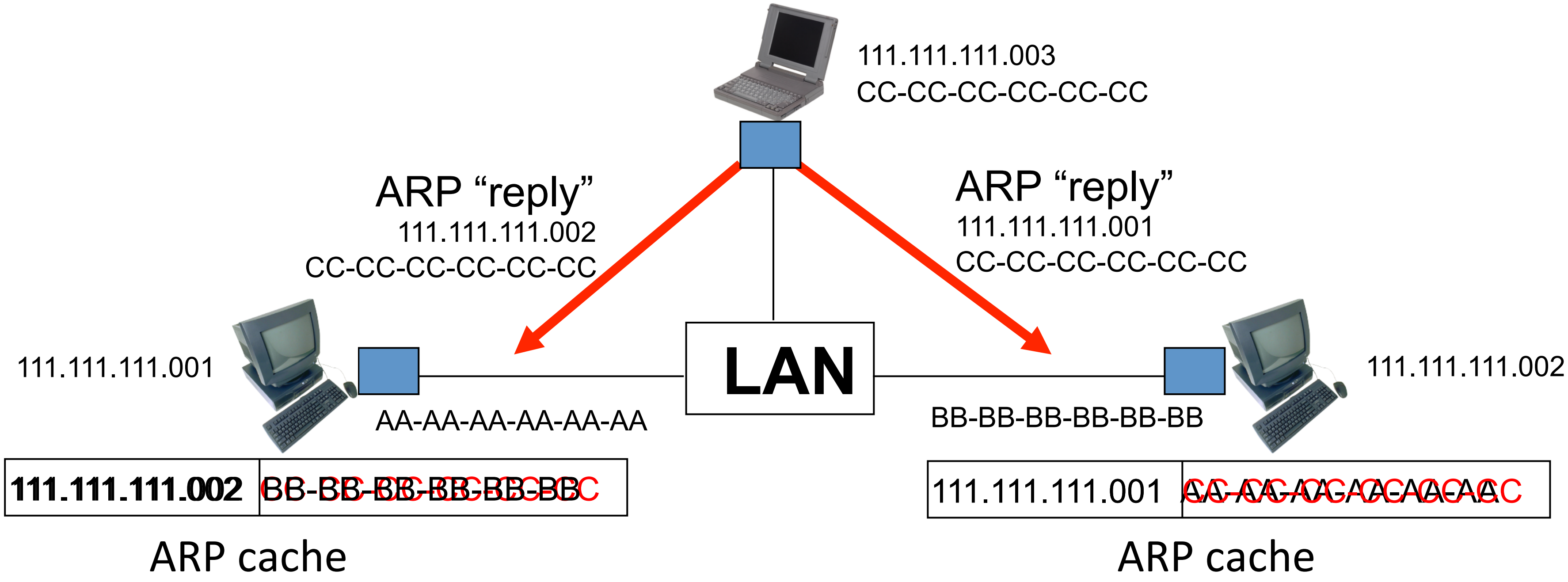
ARP

- ARP is ***stateless***
- ARP sends **request** and receives ARP **reply**
- Replies used to fill ARP cache



ARP Cache Poisoning

- ARP is stateless, so…
- Accepts **“reply”**, even if no **request** sent



- Host CC-CC-CC-CC-CC-CC is man-in-the-middle