CSE 127 Computer Security

Stefan Savage, Spring 2019, Lecture 13

Network Security I

Objectives

- Understand
  - Architecture of the Internet protocol suite (TCP/IP)
    - CSE123 in 20mins
  - Common weaknesses in networking protocols
  - Available mitigations and their limitations

Review: Internet Protocol Suite

- Application Layer
  - Examples: SMTP, FTP, SSH, HTTP, etc.
- Transport Layer: Port-addressed host-to-host communications (on LAN or WAN).
  - User Datagram Protocol (UDP): single packet transmission with no reliability or ordering mechanisms.
  - Transmission Control Protocol (TCP): connection establishment, reliable transmission, and flow-control.
- Internet Layer (IP): Fragmentation, reassembly, and end-to-end (across network boundaries) routing of data packets.
  - Provides a uniform interface that hides the underlying network topology.
- Link Layer: Transmission of data frames within a local network (without intervening routers).
  - Example: Ethernet
- Physical Layer: Transmission of raw bits (rather than logical data packets) over a physical data link connecting network nodes.
  - Example: 100BASE-T
  - [Technically not part of the Internet Protocol Model, but is still there]

Review: Internet Protocol Suite

Internet protocol suite

Application layer

BGP · DHCP · DNS · FTP · HTTP · IMAP · LDAP · MGCP · MQTT · NNTP · NTP · POP · ONC/RPC · RTP · RTSP · RIP · SIP · SMTP · SNMP · SSH · Telnet · TLS/SSL · XMPP · *more...*

Transport layer

TCP · UDP · DCCP · SCTP · RSVP · *more...*

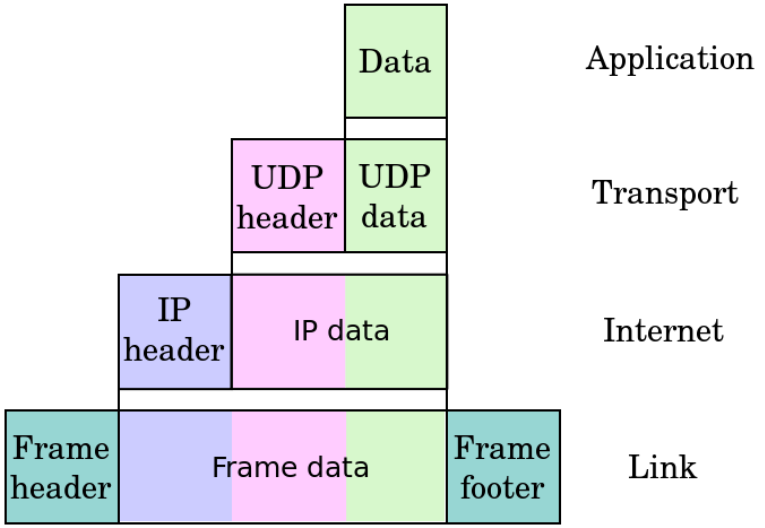
Internet layer

IP (IPv4 · IPv6) · ICMP · ICMPv6 · ECN · IGMP · IPsec · *more...*

Link layer

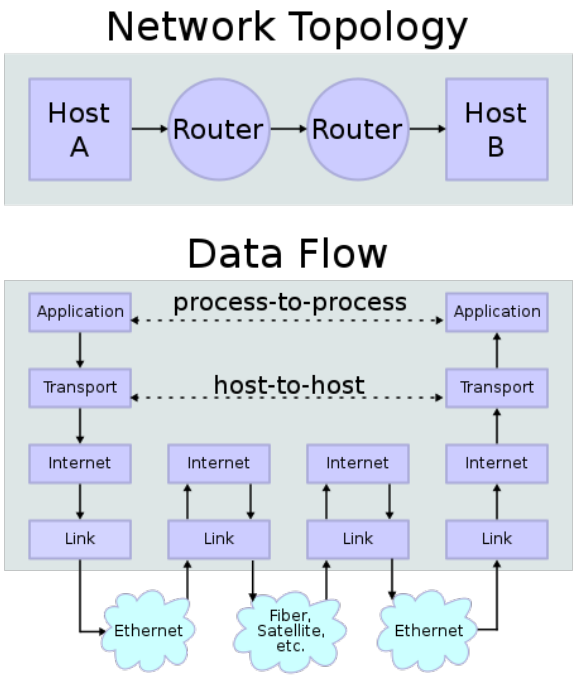
ARP · NDP · OSPF · Tunnels (L2TP) · PPP · MAC (Ethernet · DSL · ISDN · FDDI) · *more...*

V · T · E



https://en.wikipedia.org/wiki/Internet_protocol_suite

Review: Internet Protocol Suite



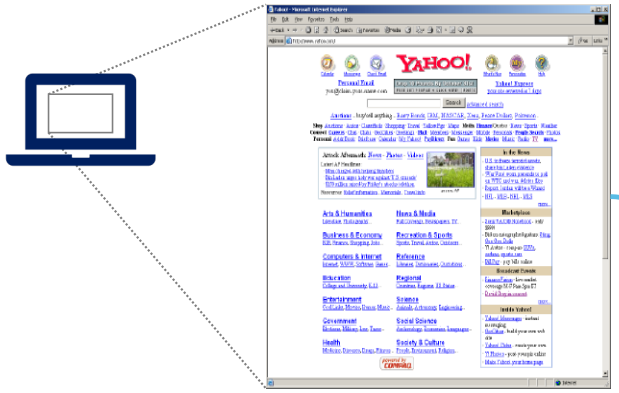
https://en.wikipedia.org/wiki/Internet_protocol_suite

TCP/IP Protocol Stack by Example

- ROUGHLY, what happens when I click on a URL while UCSD's network?

Application Layer (HTTP)

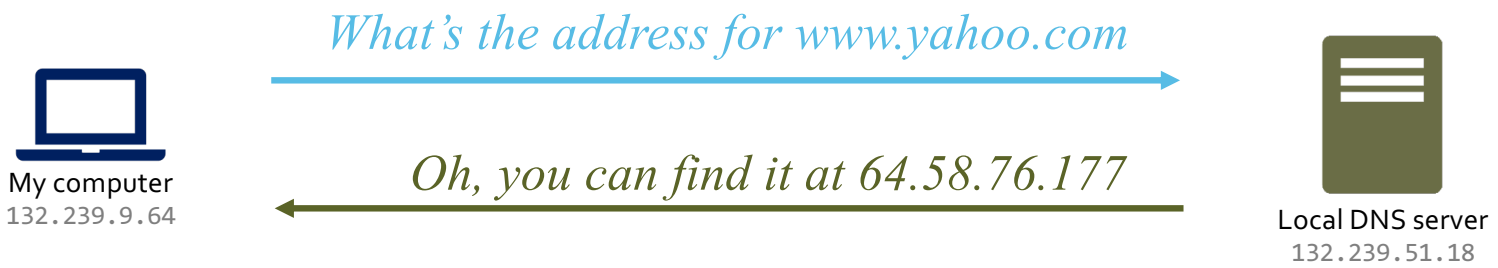
- Turn click into HTTP GET request



```
GET http://www.yahoo.com/r/mp HTTP/1.1
Host: www.yahoo.com
Connection:keep-alive
…
```

Application Layer (Name Resolution)

- Where is www.yahoo.com?

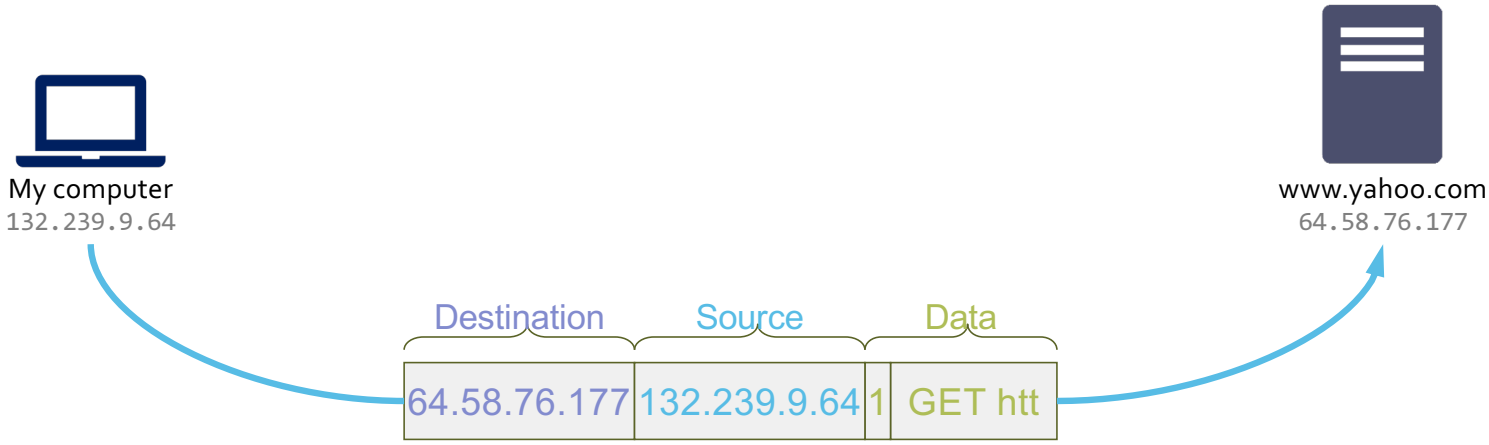


Transport Layer (TCP)

- Break message into packets (TCP segments)
- Should be delivered reliably & in-order

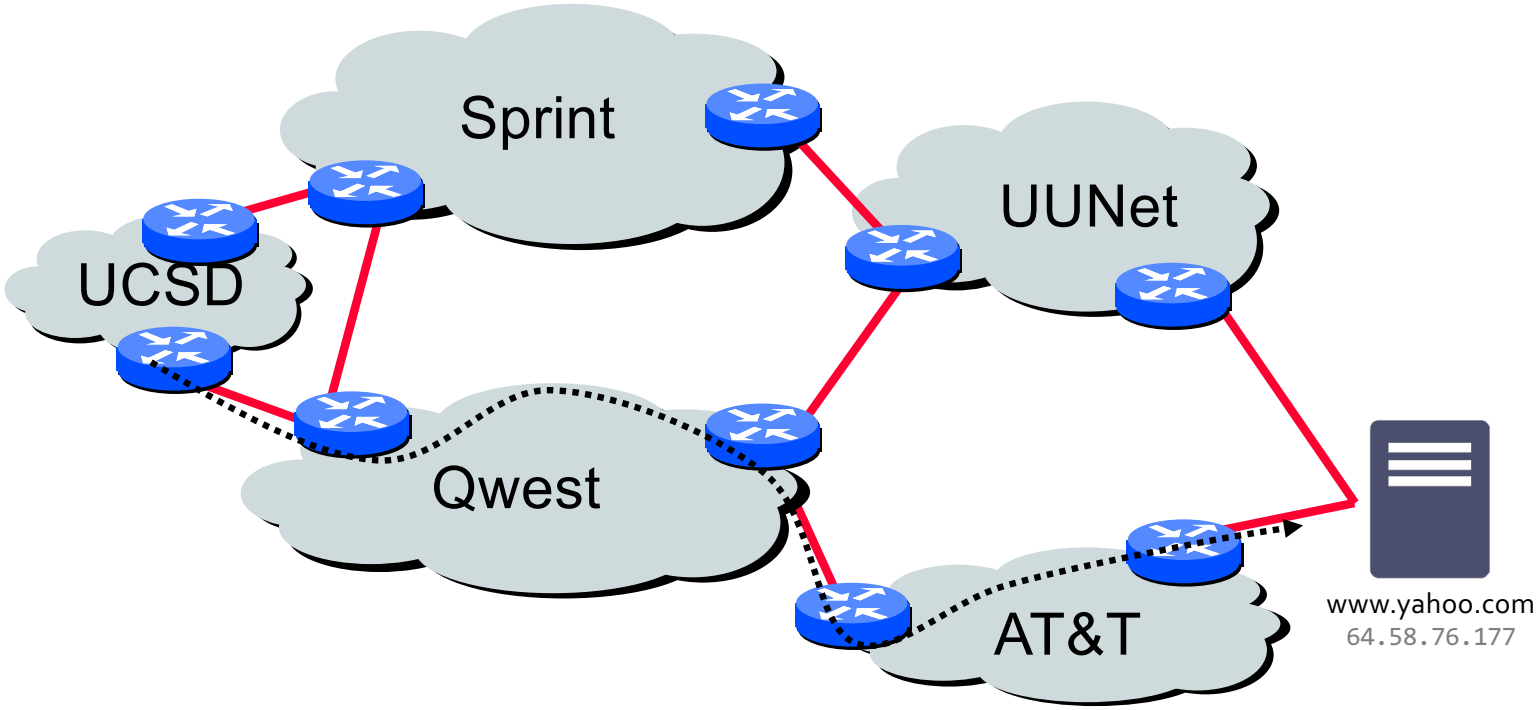
Network layer: IP Addressing

- Address each packet so it can traverse network and arrive at host
- Addresses are generally globally unique



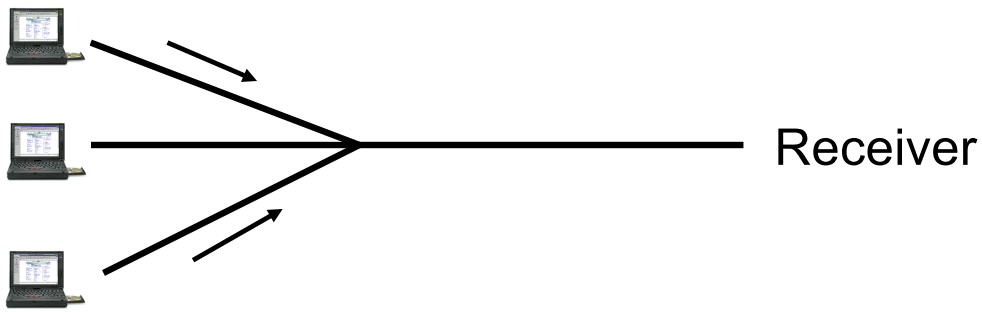
Network Layer: IP Routing

- Each router forwards packet towards destination



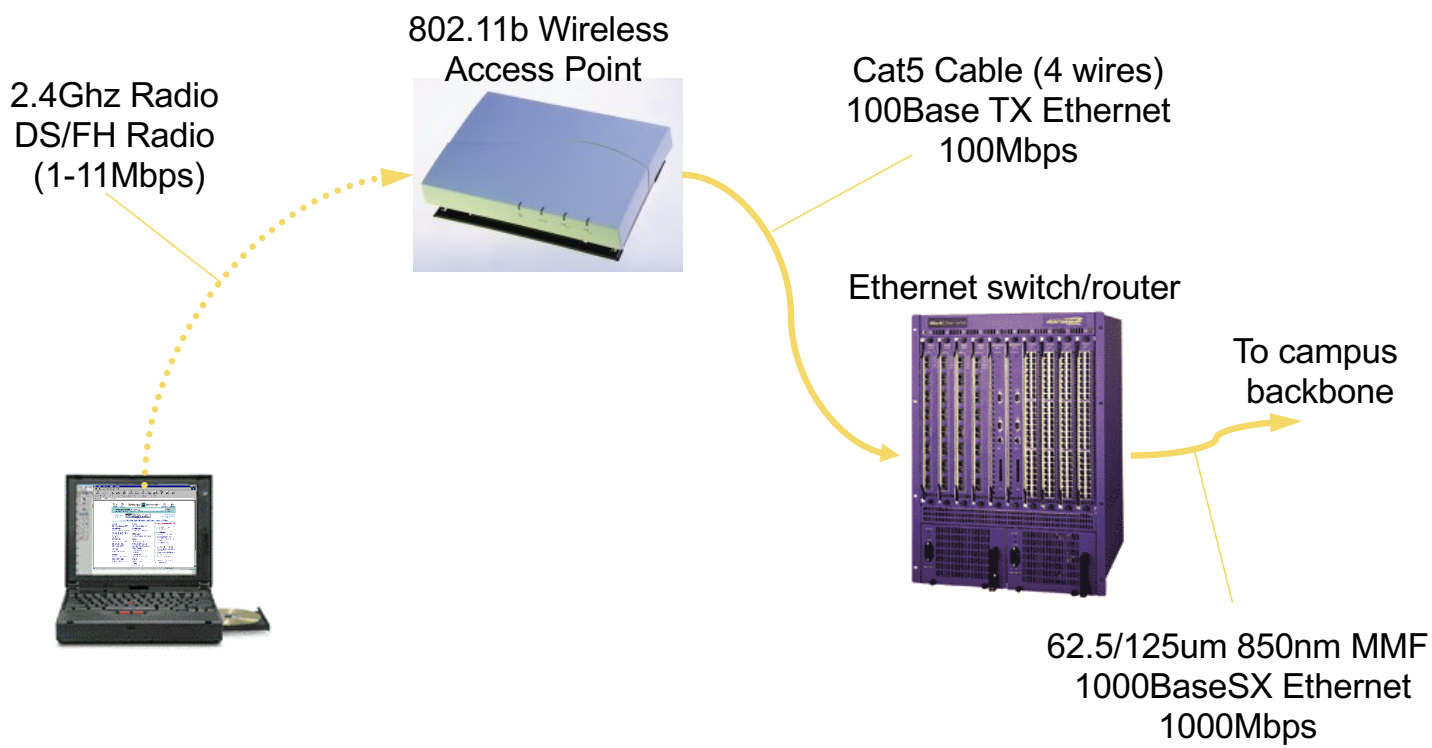
Datalink layer (Ethernet)

- Break message into frames
- Media Access Control (MAC)
  - Can I send now? Can I send now?



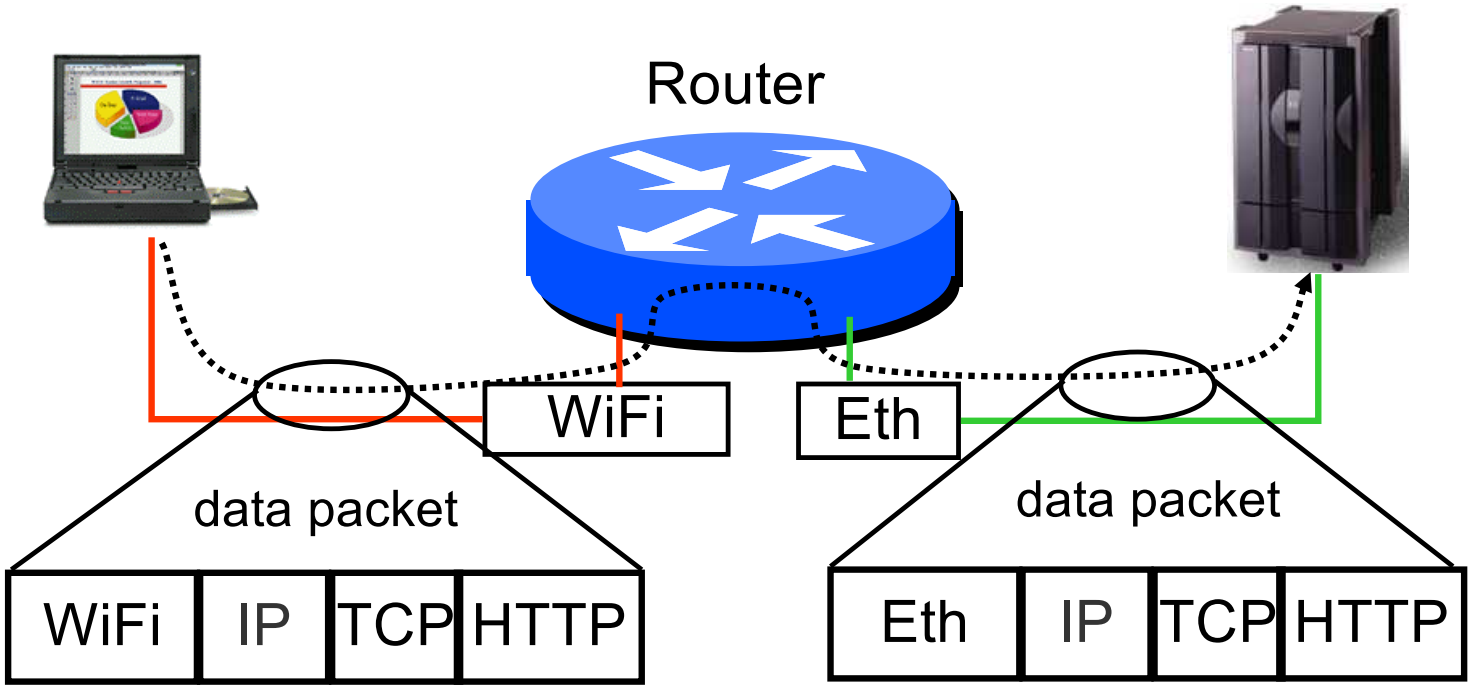
- Send frame

Physical layer

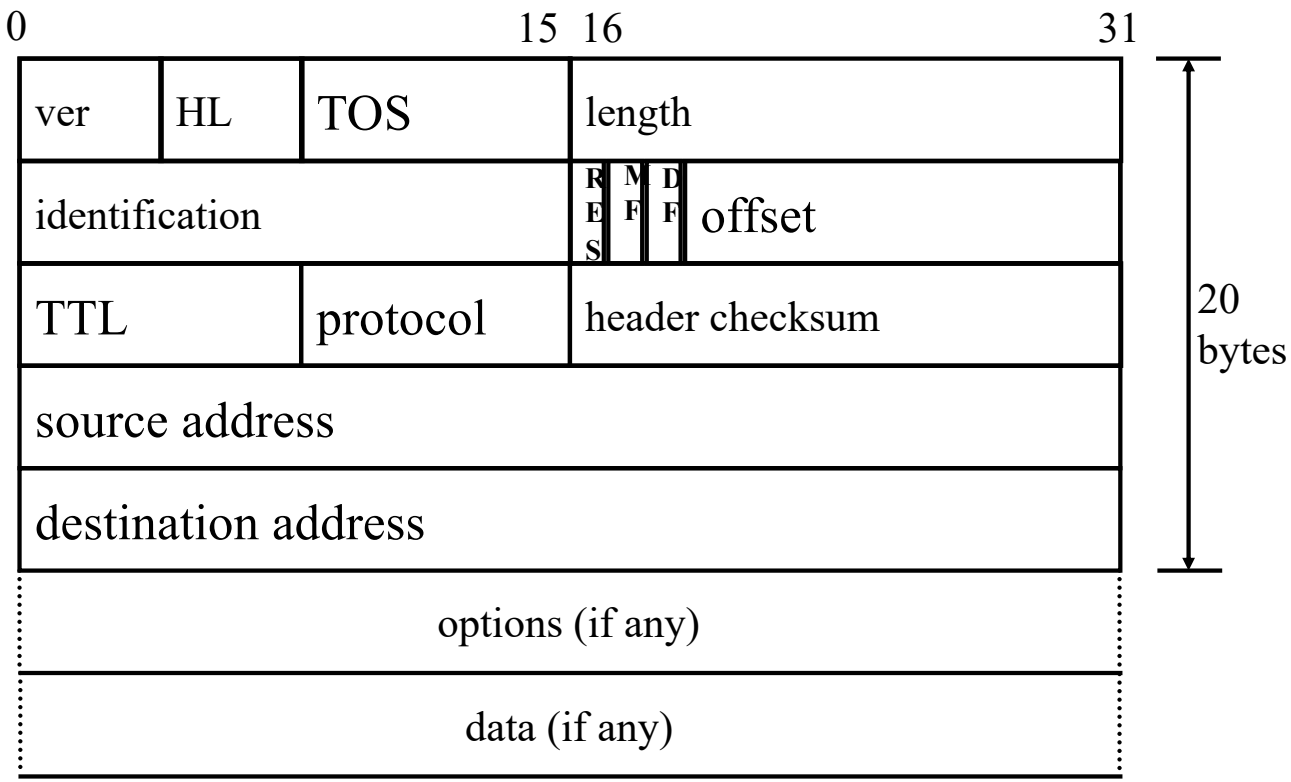


How IP works

Separate physical networks communicate to form a *single* logical network (IP addrs globally unique)



IPv4 Packet Header

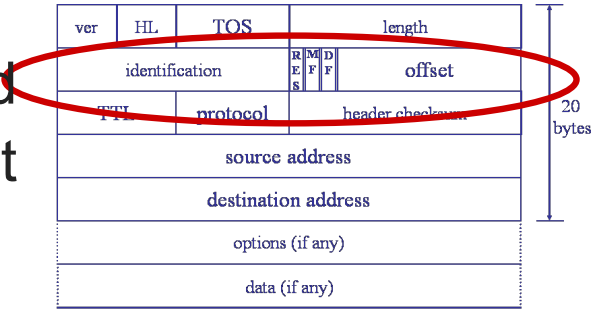


IP Protocol Functions (Summary)

- Routing
  - IP host knows location of local router (gateway)
  - IP gateway must know route to other networks
    - Packets usually take multiple hops to get to destination
  - Addresses are globally meaningful
    - 32 bits (IPv4), address separated into network part and host part
- Error reporting
  - Send Internet Control Message Protocol (ICMP) packet to source if there is a problem
- Fragmentation and reassembly
  - If max-packet-size less than the user-data-size
- TTL field: decremented after every hop
  - Packet dropped if TTL=0. Prevents infinite loops.

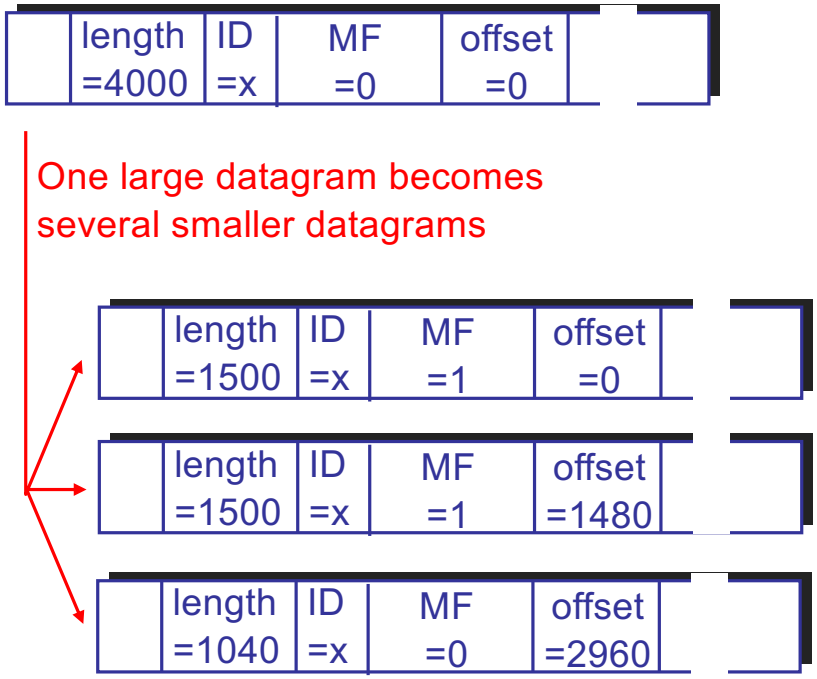
Fragmentation

- Sender writes unique value in *identification* field
- If router fragments packet it copies id into each fragment



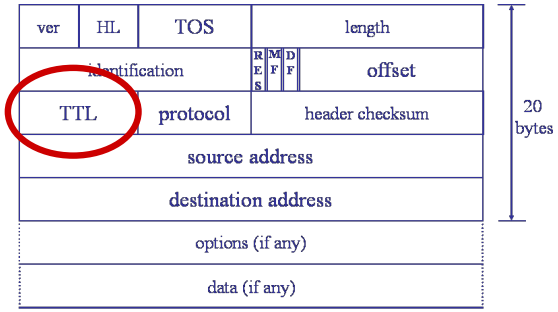
- *Offset* field indicates position of fragment in bytes (offset 0 is first)
  - *MoreFragments* flag indicates that this isn't the last fragment
  - *DontFragment* flag tells gateway not to fragment
- All routers must support 576 byte packets (MTU)

IP Fragmentation and Reassembly



TTL (Time-to-Live)

- How many more routers can this packet pass through?
  - Designed to limit packet from looping forever
- Each router decrements TTL field
- If TTL is 0 then router discards packet

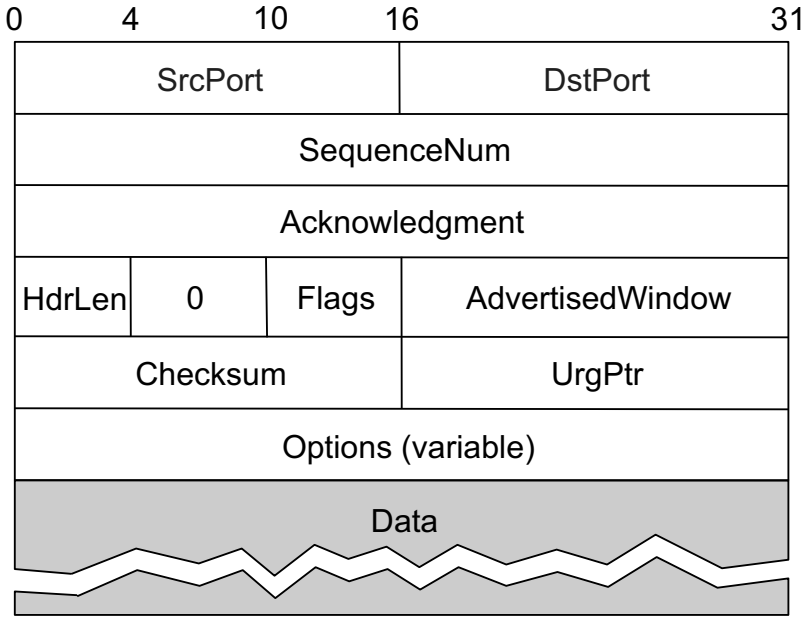


TCP Primer

- TCP provides reliable, ordered delivery of bytes
- Establishes a stateful bi-directional session between two IP:port endpoints
- Each side maintains:
  - Sequence number: sequence base + count of bytes sent
  - Acknowledgement number: acknowledgement base + count of bytes received
- Special packet flags
  - SYN: I want to start a connection
  - FIN: I want to shut down a connection
  - RST: We are killing this connection now

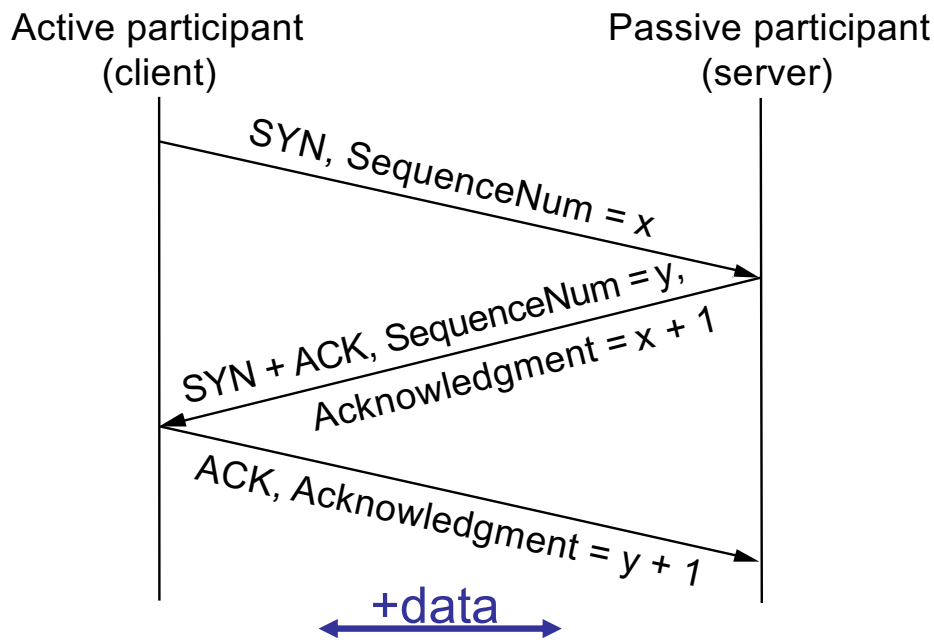
TCP Header Format

- **Ports** plus IP addresses identify a connection



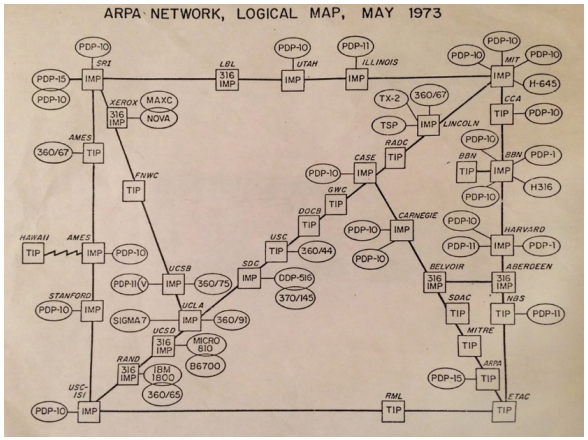
Connection Setup: Agree on initial Sequence #'s

- Three-way handshake



TCP/IP Security (1970's)

- Original TCP/IP design: Trusted network and hosts
  - Administered by mutually trusted parties
- End-to-end Principle
  - Intelligence is at the edges
  - Network is simple
    - Optimized for speed and simplicity, maintains no state
- Robustness Principle
  - "*In general, an implementation must be conservative in its sending behavior, and liberal in its receiving behavior. That is, it must be careful to send well-formed datagrams, but must accept any datagram that it can interpret (e.g., not object to technical errors where the meaning is still clear).*"
    - https://www.ietf.org/rfc/rfc0791.txt



https://en.wikipedia.org/wiki/ARPANET

TCP/IP Security (1980's)

- Wait … what if we can't trust everyone?
  - "*When describing such attacks, our basic assumption is that the attacker has more or less complete control over some machine connected to the Internet. This may be due to flaws in that machine's own protection mechanisms, or it may be because that machine is a microcomputer, and inherently unprotected. Indeed,* ***the attacker may even be a rogue system administrator***."
    - *Security Problems in the TCP/IP Protocol Suite*, Steve Bellovin 1989
- Can't trust the hosts
  - Compromised hosts
  - Untrusted insiders on internal networks
  - Anyone can connect to public Internet [next decade]
- But network is still trusted

TCP/IP Security (today)

- Can't trust the network either
  - Network equipment can be compromised
  - Untrusted network operators
  - Anyone can access the physical channel of wireless networks

TCP/IP Security

- Built-in trust assumptions:
  - Network protocols used only as intended
    - Correct packet headers
    - Consideration of others' resources
      - Rate limiting of costly operations
  - Hosts controlled by trusted administrators
    - Random people can't get onto the network
    - Correct information reported by hosts
    - Protocols implemented correctly

Attacker Models

- Man in the middle: can see, block, and modify traffic
  - Attacker controls wifi access point
- Passive: Eavesdrop on traffic
  - Attacker has passive tap or recorded traces
- Off-path: attacker can inject traffic into network
  - Anyone with access to network

No Confidentiality

- Who can see the packets you send?
  - Network (routers, switches, access points, etc.)
  - Unprotected WiFi network: everyone within range
  - WPA2 Personal (PSK): everyone on same network
  - Non-switched Ethernet: everyone on same network
  - Switched Ethernet: maybe everyone on same network

No Authentication

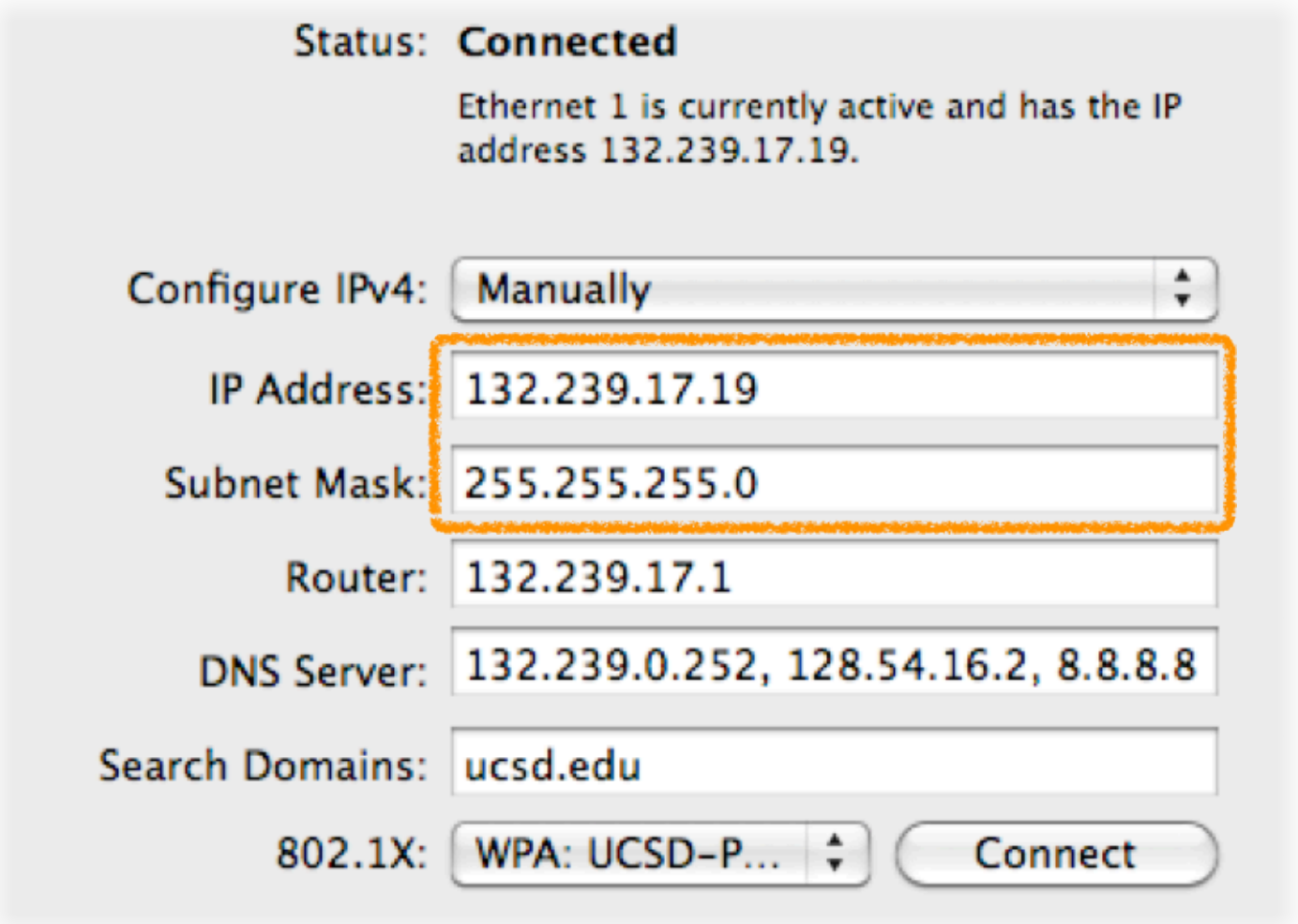
- TCP/IP offers no authentication of packets
  - Source address in IP header set by sender
- Attacker with direct access to network (including MitM) can spoof source address
  - Spoof: forge, set to arbitrary value
- Connectionless protocols (UDP) especially vulnerable
  Why?

Link Layer (MAC)

- Physical channel is often shared by multiple hosts on the local network.
  - Examples: open WiFi, non-switched Ethernet
- Link layer controls access to the physical medium.
  - Also known as the Media Access Control (MAC) layer.
  - Another acronym collision[3]
- How to make sure each host only gets frames addressed to it?
- Each host is responsible for picking up frames addressed to it and ignoring the others.
  - Honor system!
- Filtering typically happens on the network card (or equivalent).
  - Only frames addressed to this host are parsed and passed on to the layer above.
- Many support "promiscuous" mode – all frames are picked up.

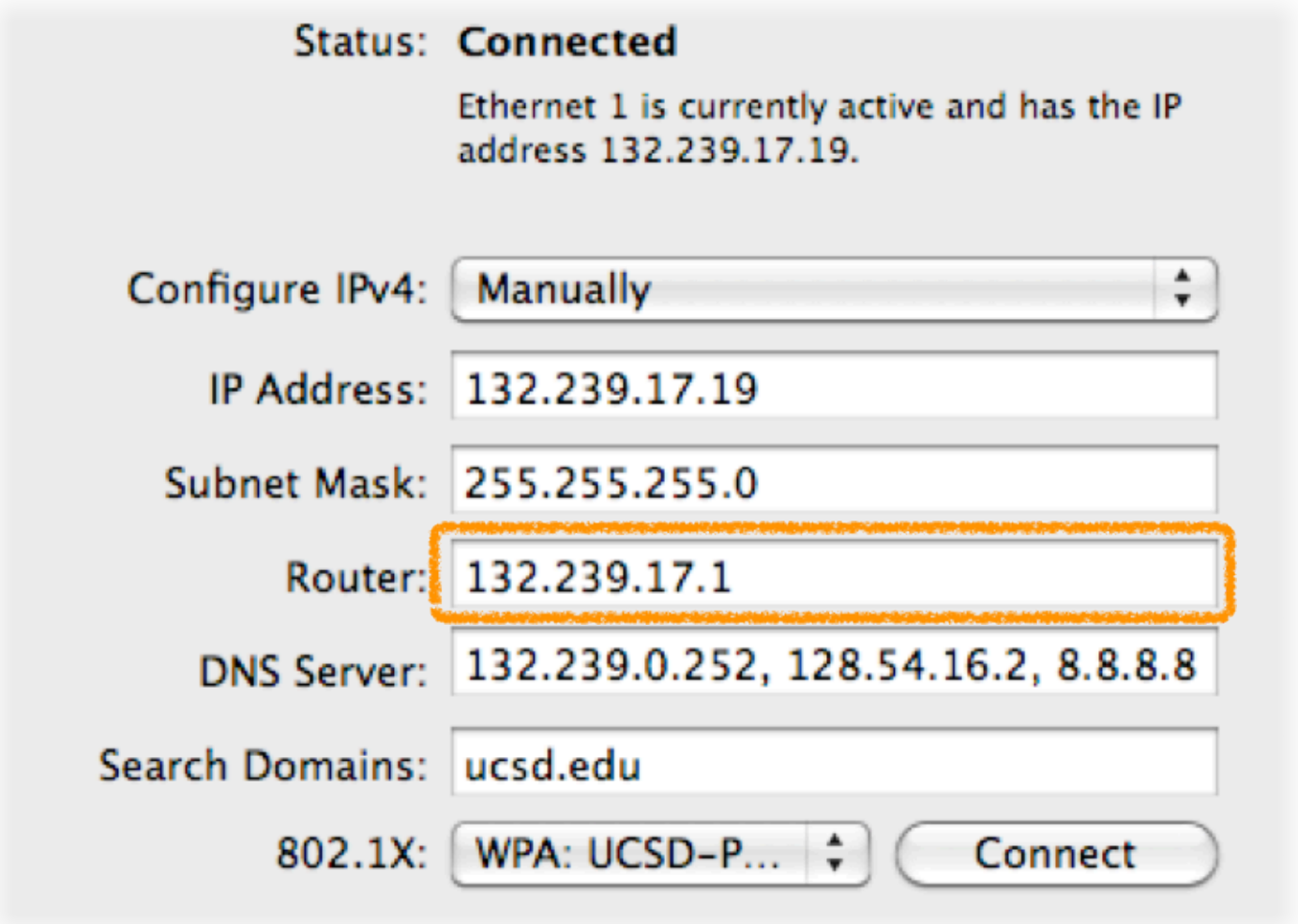
Network Routing

- Say I want to send packet to 8.8.8.8 …
- Step 1: Is host on local network?
  - Check subnet masks of local networks


Status: Connected
Ethernet 1 is currently active and has the IP address 132.239.17.19.
Configure IPv4: Manually
IP Address: 132.239.17.19
Subnet Mask: 255.255.255.0
Router: 132.239.17.1
DNS Server: 132.239.0.252, 128.54.16.2, 8.8.8.8
Search Domains: ucsd.edu
802.1X: WPA: UCSD-P...
Connect

Network Routing

- Say I want to send packet to 8.8.8.8 …
- Step 1: Is host on local network?
  - Local: send directly
  - Not local: send via default gateway

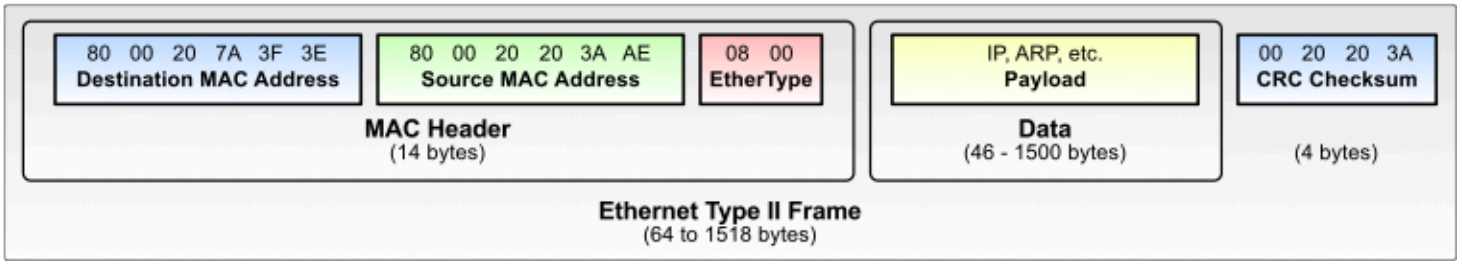

Status: Connected
Ethernet 1 is currently active and has the IP address 132.239.17.19.
Configure IPv4: Manually
IP Address: 132.239.17.19
Subnet Mask: 255.255.255.0
Router: 132.239.17.1
DNS Server: 132.239.0.252, 128.54.16.2, 8.8.8.8
Search Domains: ucsd.edu
802.1X: WPA: UCSD-P...
Connect

Network Routing

- Say I want to send packet to 8.8.8.8 …
- Step 1: Is host on local network?
  - Local: send directly
  - Not local: send via default gateway
- Step 2: Create IP packet
- Step 3: Create and send link layer (e.g. Ethernet) frame

Network Routing

- Ethernet frame:

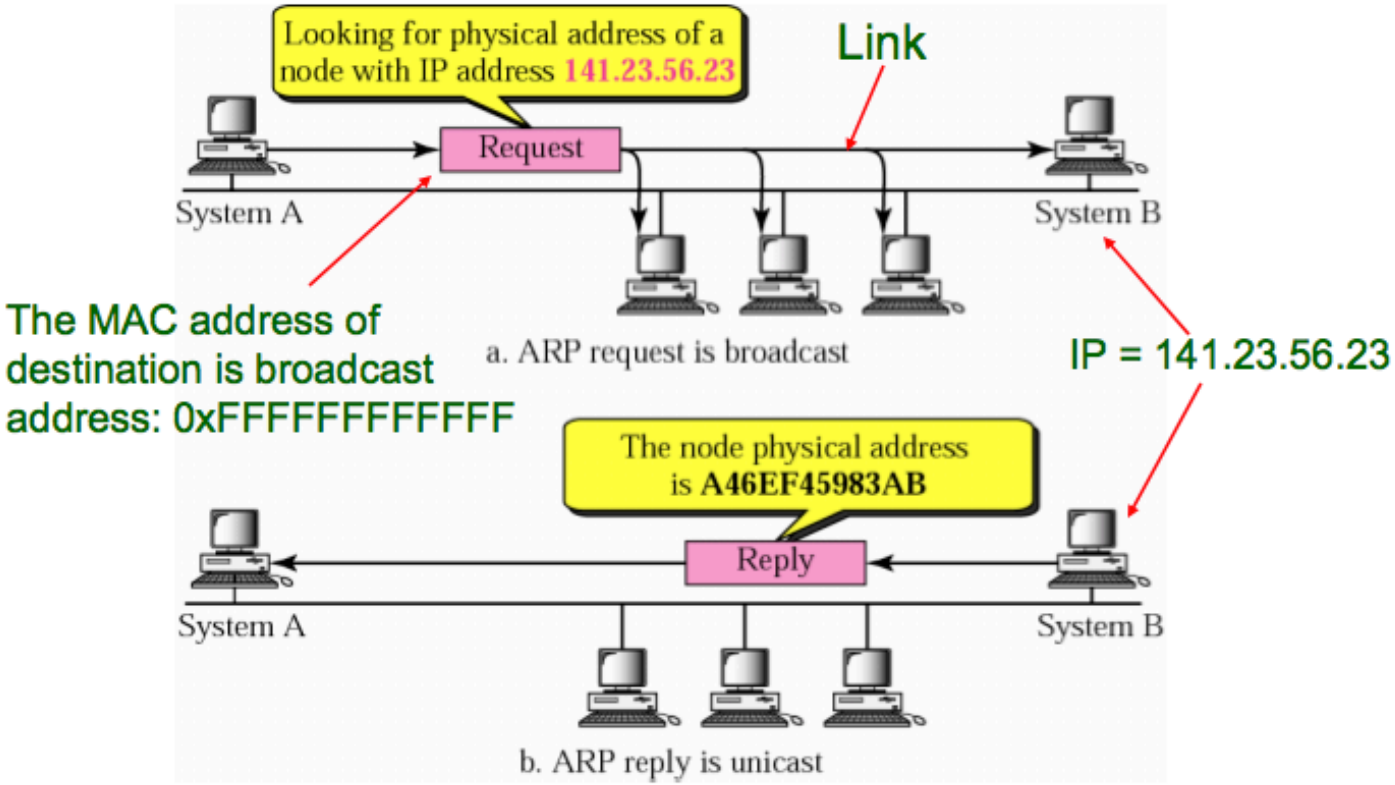


- Host needs to fill in Ethernet destination address
  - MAC address of host on local network
  - MAC address of gateway for host not on local network
- How to find Ethernet address from an IP address?

Address Resolution Protocol (ARP)

- Address Resolution Protocol (ARP)
  - used to query hosts on local network to get link-layer address for an IP address
- Protocol sketch
  - Alice (looking for Bob's IP) broadcasts an ARP request:
    - "What is the MAC address of 10.0.0.3?"
  - Bob sees broadcast (asking about his IP address) and replies:
    - "The MAC address of 10.0.0.3 is 01:02:03:04:05:06."
  - Alice sends IP packet for 10.0.0.3 in an Ethernet frame to 01:02:03:04:05:06.

Address Resolution Protocol (ARP)



©The McGraw-Hill Companies, Inc., 2000

Address Resolution Protocol (ARP)

- ARP messages are Ethernet frames
- ARP requests are broadcast (on the local subnet)
- Anyone can send an ARP reply

ARP Spoofing

- Since
  - ARP requests are broadcast (on the local subnet)
  - Anyone can send an ARP reply
- Attacker on the network can impersonate any other host
  - Who has the MAC address for IP address 8.8.8.8?
  - Uh… I do… send your traffic to me…
- Mitigation
  - Fixed ARP tables
    - Impractical for all but small fixed networks
  - Port binding on switch
    - Restrict MAC and IP addresses allowed to a single port on a switch at a time
  - Higher level host authentication
    - E.g. SSH or TLS

Problems With Addressing

- This problem repeats at every protocol layer:
  - Source needs to send something to destination.
  - How to know which address corresponds to name?
    - Domain name to IP address
    - IP routing
    - IP address to Ethernet address
    - …

SECTIONS

Chicago Tribune

SUBSCRIBE SALE: 10¢ a week

LOG IN

SUNDAY MAY 27, 2018

SPORTS BREAKING MOST POPULAR OPINION ENTERTAINMENT BUSINESS

94°

Change-of-address scam moved UPS corporate headquarters to tiny Rogers Park apartment, feds say

A Rogers Park man is under investigation after postal inspectors say they uncovered a scheme to scam Atlanta-based UPS out of mail and checks through the use of a change-of-address form that redirected the mail to the man's Ashland Avenue apartment on the Far North Side. (Nancy Stone / Chicago Tribune)

By **Jason Meisner** · **Contact Reporter**
Chicago Tribune

APRIL 23, 2018, 5:00 AM

Network Routing

- Say I want to send packet to 8.8.8.8 …
- Step 1: Is host on local network?
  - Local: send directly
  - Not local: send via default gateway
- Step 2: Create IP packet
- Step 3: Create and send link layer (e.g. Ethernet) frame
- Step 4: Gateway picks next router in path and forwards the IP packet
  - Repeat until destination is reached
  - How to know which router to forward to next?

Border Gateway Protocol (BGP)

- Border Gateway Protocol (BGP) is used to manage IP routing information between networks on the Internet
- Each BGP node maintains connections to a set of trusted neighbors
  - Connections between neighbors may be (weakly) authenticated
- Neighbors share routing information
- No authorization
  - Malicious (or malfunctioning) BGP nodes may provide incorrect routing information that redirects IP traffic

BGP Hijacking

- 2008 Pakistan tried to block YouTube within the country
  - Pakistan Telecom claimed ownership of YouTube's IP block via BGP
  - BGP nodes forwarded this routing information
  - YouTube is "sinkholed" globally
  - https://www.ripe.net/publications/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study
- 2018 MyEtherWallet.com compromised, $100,000's reported stolen
  - Attackers used BGP hijacking to claim ownership of a chunk of Amazon Route 53 (DNS) addresses
  - Used hijacked DNS traffic to direct MyEtherWallet.com-bound traffic to attackers' servers in Russia
  - https://dyn.com/blog/bgp-hijack-of-amazon-dns-to-steal-crypto-currency/

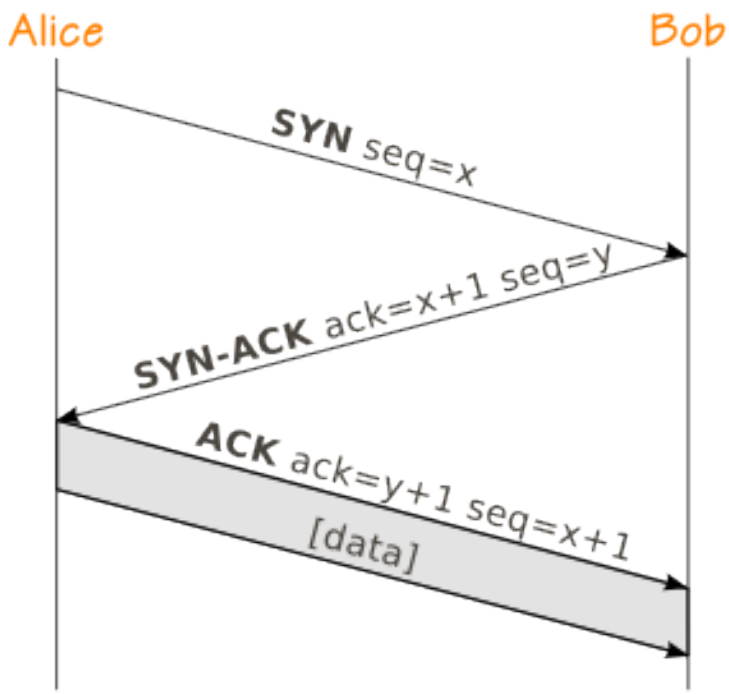
IP Spoofing Attacks

- There is no authentication in Link or Internet layers
- Even if routing is correct, Eve can still spoof Alice's IP address
  - Eve can send IP packets claiming to be form Alice
  - Eve may not be able to receive IP packets addressed to Alice
- UDP: trivial
  - Stateless protocol, each datagram is independent of others
- TCP: mode complicated, but still possible
  - Two endpoints maintain a shared state
  - Attacker must be able to guess it

TCP Connection Spoofing

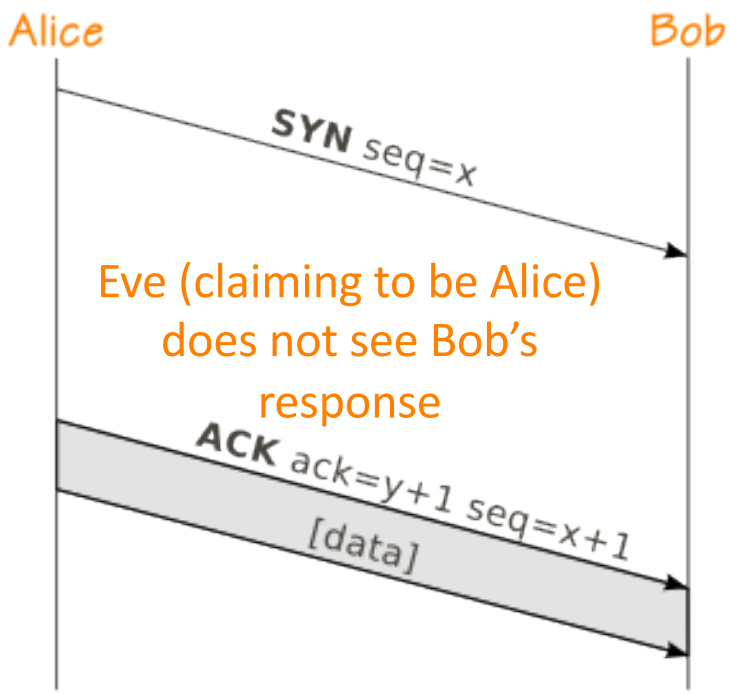
- Eve needs to complete the TCP three-way handshake between "Alice" and Bob
- Eve can't see traffic between Alice and Bob
  - "TCP off-path attack"
- Eve needs to guess initial sequence number y in order to correctly ACK Bob's SYN

Three-Way Handshake



source: Wikipedia

Three-Way Handshake



source: Wikipedia

TCP Connection Spoofing

- The sequence number field is 32 bits
- Early implementations just incremented a global counter used to initialize sequence numbers for TCP connections
  - RFC 793 requires counter incrementing every 4 μs (250 kHz)
  - Early BSD kernels incremented by a large constant every second
- Later pseudo-random number generators were used
  - PRNGs were still global, weaknesses allowed guessing

Example Denial-of-Service (DoS) vulnerability with TCP spoofing [Watson'04]

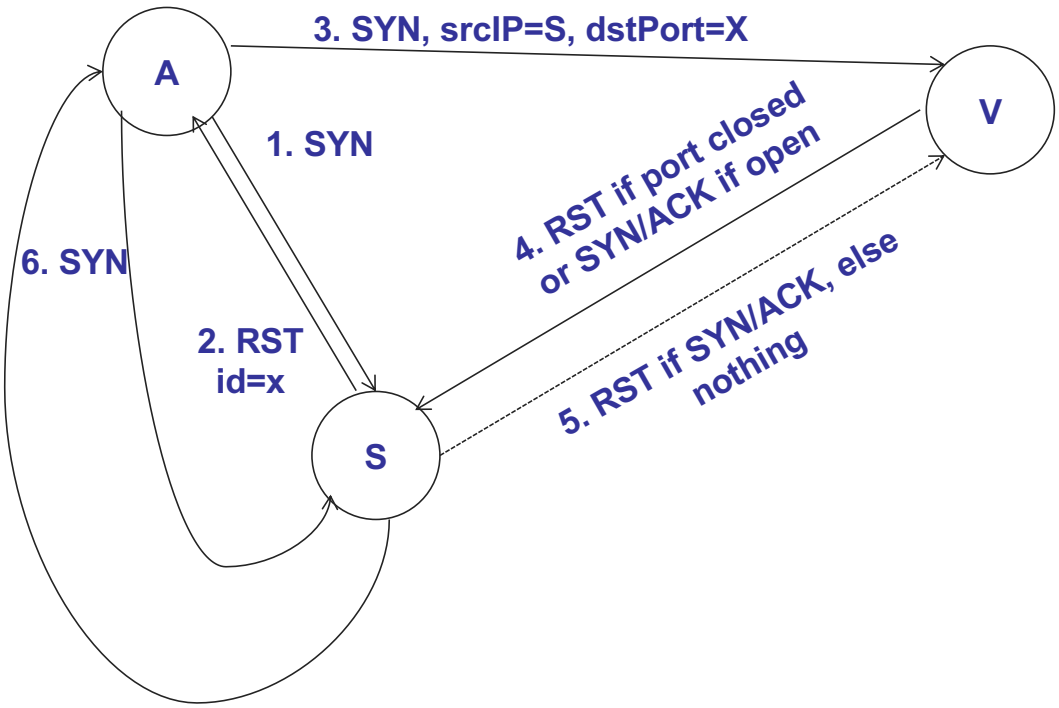
- Suppose attacker can guess seq. number for an existing connection:
  - Attacker can send Reset (RST) packet to close connection. Results in DoS.
  - Naively, success prob. is $1/2^{32}$ (32-bit seq. #'s).
  - Most systems allow for a large window of acceptable seq. #'s
    - Much higher success probability.

- Attack is most effective against long lived connections (expensive to set up again; BGP)

Blind port scanning

- Similar issue: IP identification field
  - Hosts typically increment by one after each packet to ensure id field is unique (recall: for fragmentation)
  - If you receive a pkt from host A at time t1 with id =10, and another packet at time t2 with id=12, you can infer… that host A sent another packet somewhere

Blind port scanning



7. Response from S…. What is value of id?
x+1 or x+2?

DDoS attacks w/spoofing

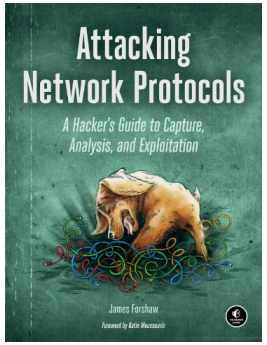
- Send millions of packets at some victim with forged source address (e.g., random source Ips)
  - Hard to block
  - Hard to identify where they are coming from

- Reflector attacks
  - Identify online services that generate large response packets in response to small requests (e.g., DNS, Memcached, NTP)
  - Send requests to such services with a source address of the victim
  - Can hugely amplify bandwidth of attacker

- Partial fix
  - Source address validation (don't let people spoof… but must impl on all networks)
  - Edge routers filter outbound packets with source addresses that could not be generated from within their network (i.e., UCSD won't forward packets without UCSD source address)

Additional Resources

- *Wireshark*
  - https://www.wireshark.org/
- *Attacking Network Protocols*
  - By James Forshaw
  - https://nostarch.com/networkprotocols



Review

- TCP/IP protocol stack was not designed for modern threat models
  - No authentication, no integrity, no confidentiality
  - Protocols vulnerable to manipulation

- Must rely on application-layer security mechanisms

Reminder

- No class on Tuesday

Next Lecture…

Network Security II