Announcements

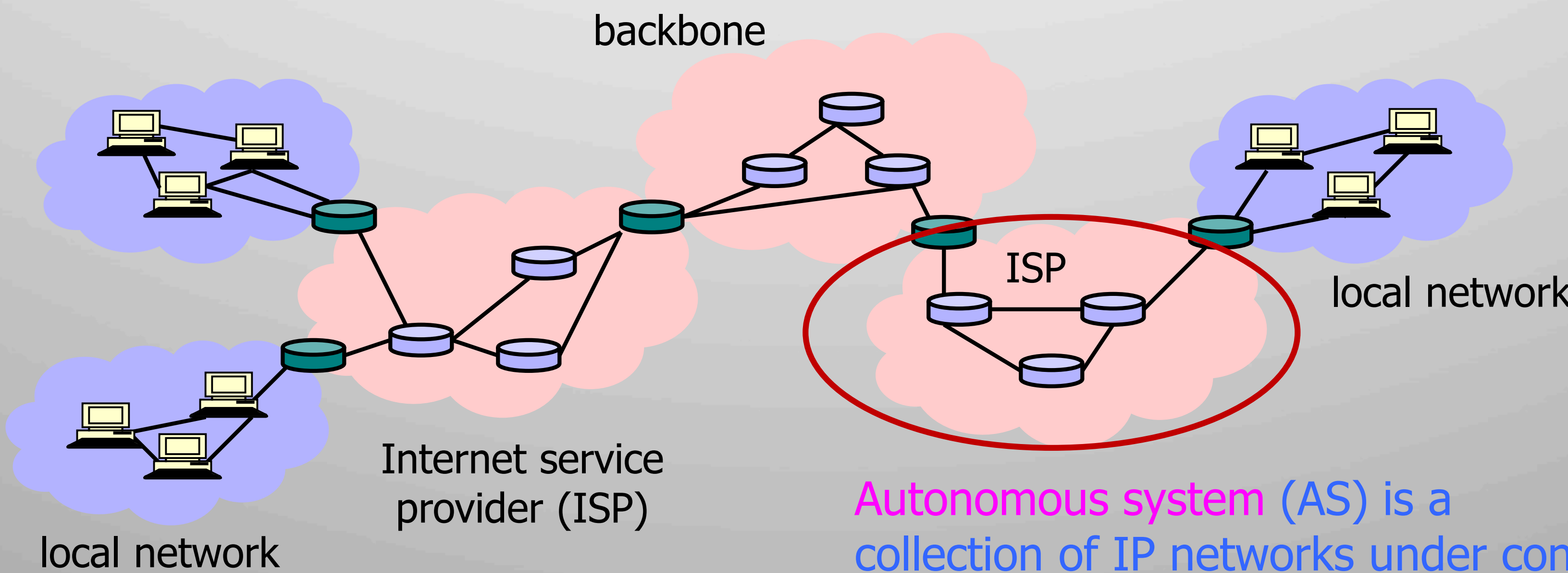
- This Thursday: ***Last lecture!***
  - Special Lecture on ***Smart Transportation Security***
  - Recent advance in security issues of self-driving cars and smart traffic light --- *one of the most disruptive tech today, impacting the **safety** for all of us*
  - Attention: **It's within the scope of final exam**
- **Final exam: *12/12, 1:30-3:30 PM***
  - Bring ***your photo ID*** with you

Lecture 16
CS 134

Network/Internet Threats/Attacks

[lecture slides are adapted from previous slides by Prof. Gene Tsudik]

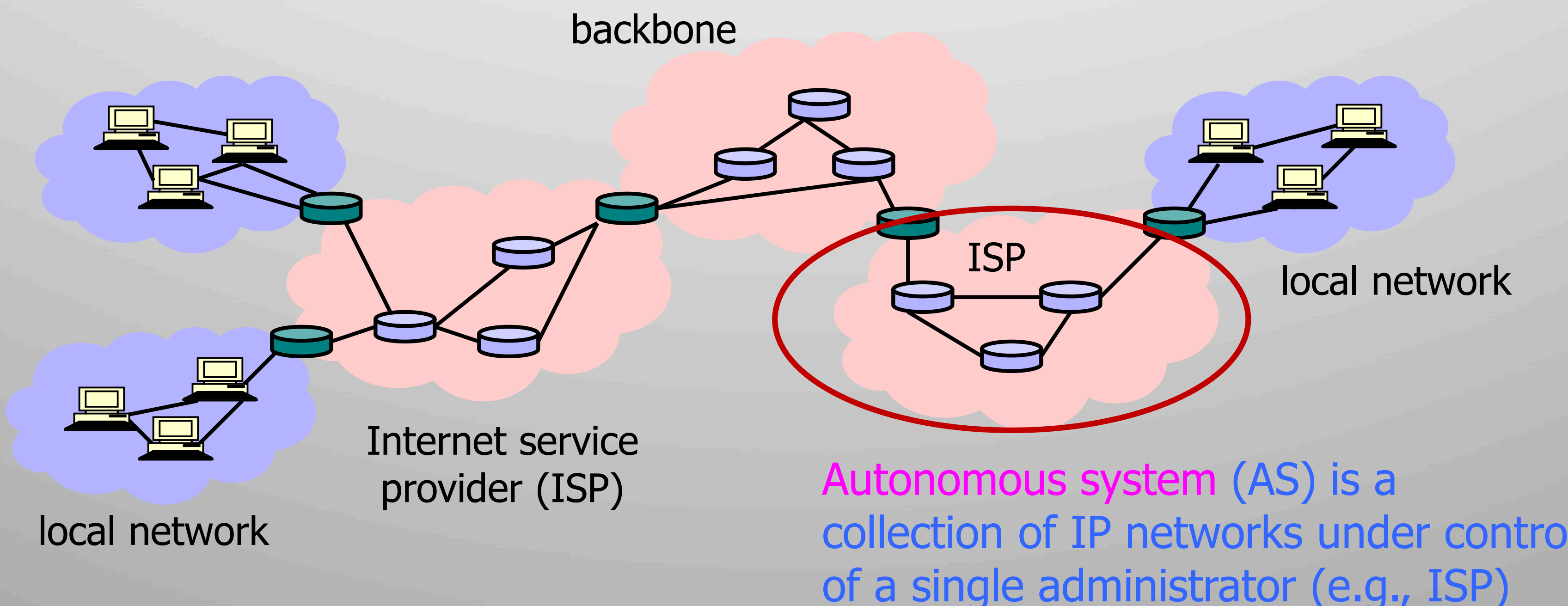
Internet Structure



Autonomous system (AS) is a collection of IP networks under control of a single administrator (e.g., ISP)

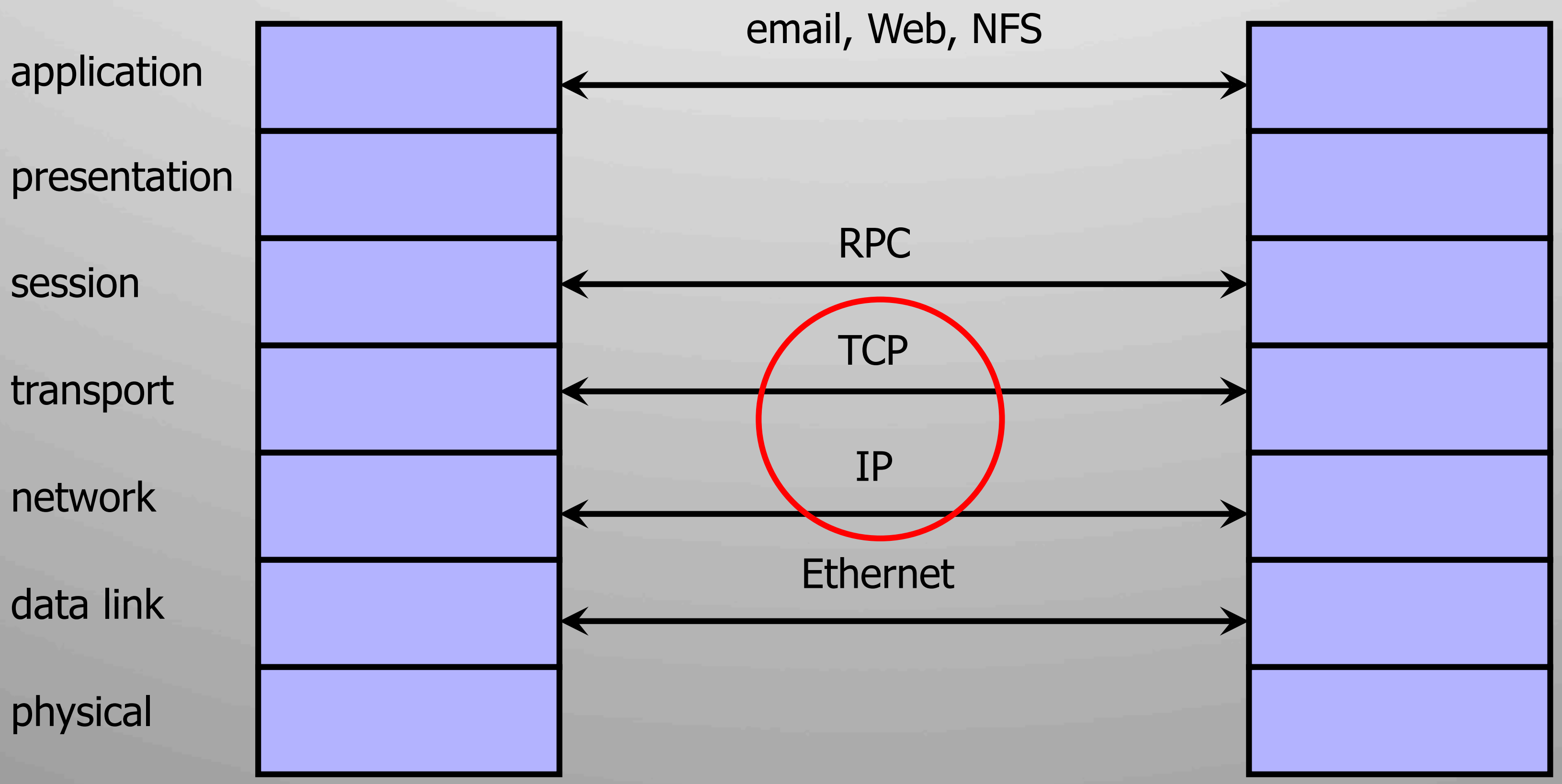
- TCP/IP for packet routing and connections
- Border Gateway Protocol (BGP) for <u>external</u> route discovery
- Domain Name System (DNS) for IP address discovery

Internet Structure

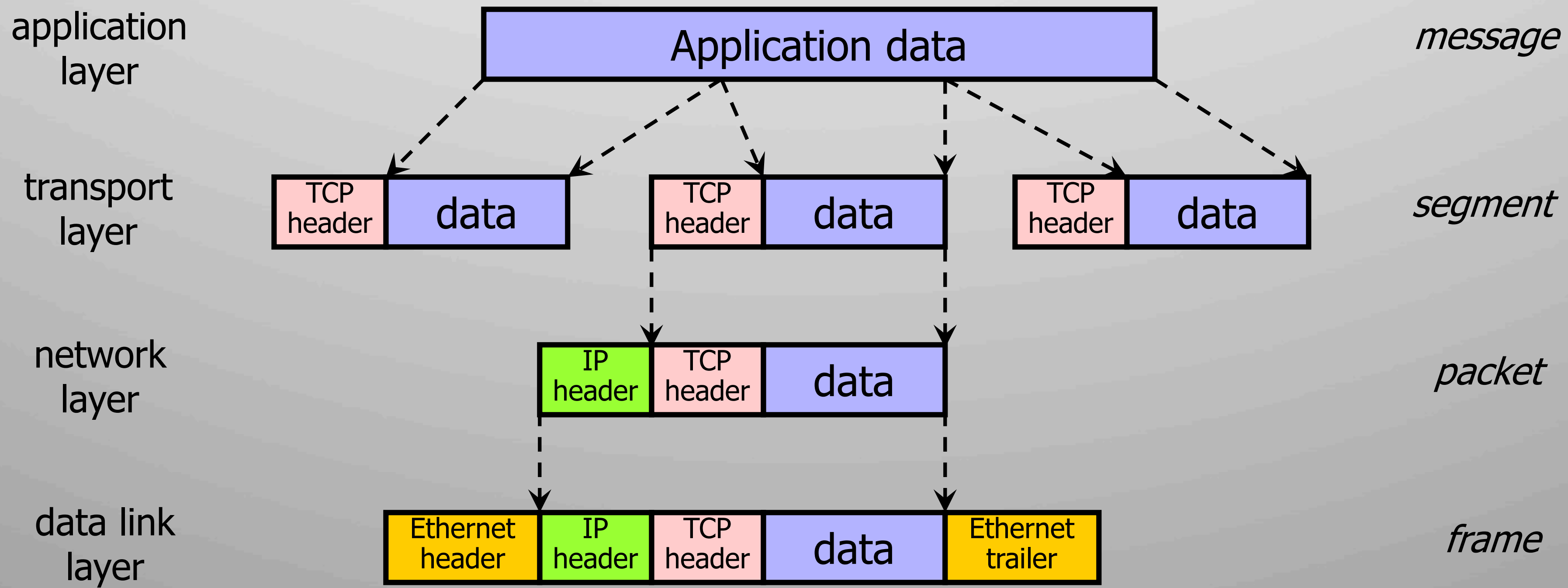


- **TCP/IP for packet routing and connections**
- Border Gateway Protocol (BGP) for <u>external</u> route discovery
- **Domain Name System (DNS) for IP address discovery**

OSI Protocol Stack



Data Formats

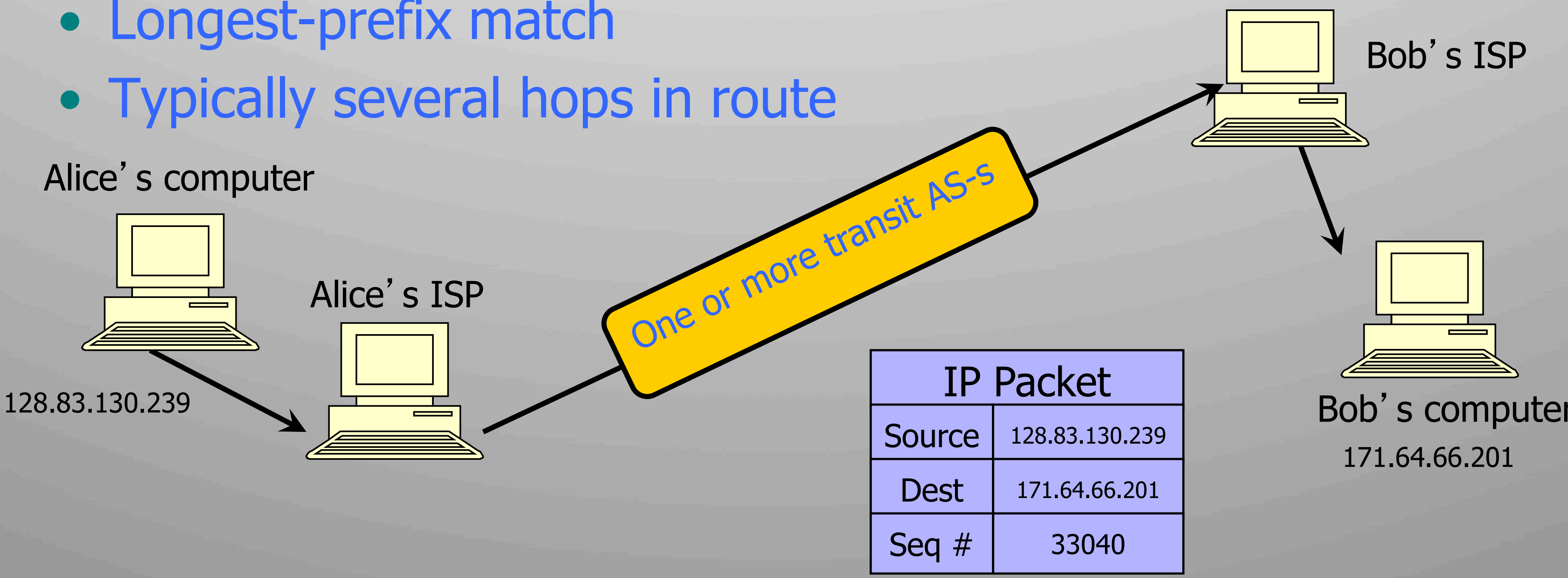


TCP (Transmission Control Protocol)

- Sender: break data into segments
  - Sequence number is attached to every packet
- Receiver: reassemble segments
  - Acknowledge receipt; lost packets are re-sent
- Connection state maintained by both sides

IP (Internet Protocol)

- Connectionless
  - Unreliable, “best-effort” protocol
- Uses addresses (and prefixes thereof) used for routing
  - Longest-prefix match
  - Typically several hops in route



IP Packet	
Source	128.83.130.239
Dest	171.64.66.201
Seq #	33040

ICMP (Control Message Protocol)

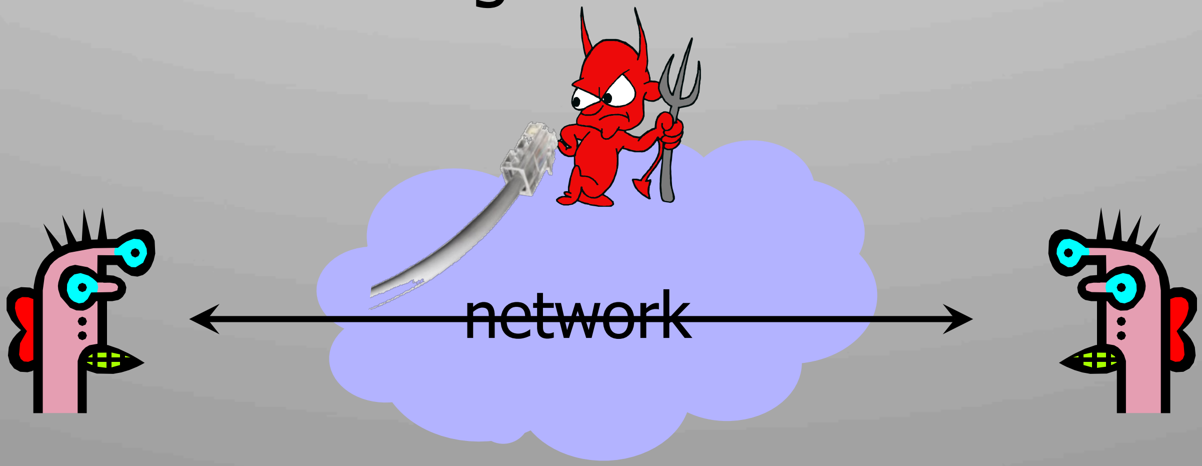
- Provides feedback about network operation
  - Out-of-band (control) messages carried in IP packets
  - Error reporting, congestion control, reachability, etc.
- Example messages:
  - Destination unreachable
  - Time exceeded
  - Parameter problem
  - Redirect to better gateway
  - Reachability test (echo / echo reply)
  - Timestamp request / reply

Security Issues in TCP/IP

- Network packets pass by and thru untrusted hosts
  - Eavesdropping (packet sniffing)
- IP addresses are public
  - E.g., Ping-of-Death, Smurf attacks
- TCP connection requires state
  - SYN flooding
- TCP state easy to guess
  - TCP spoofing and connection hijacking

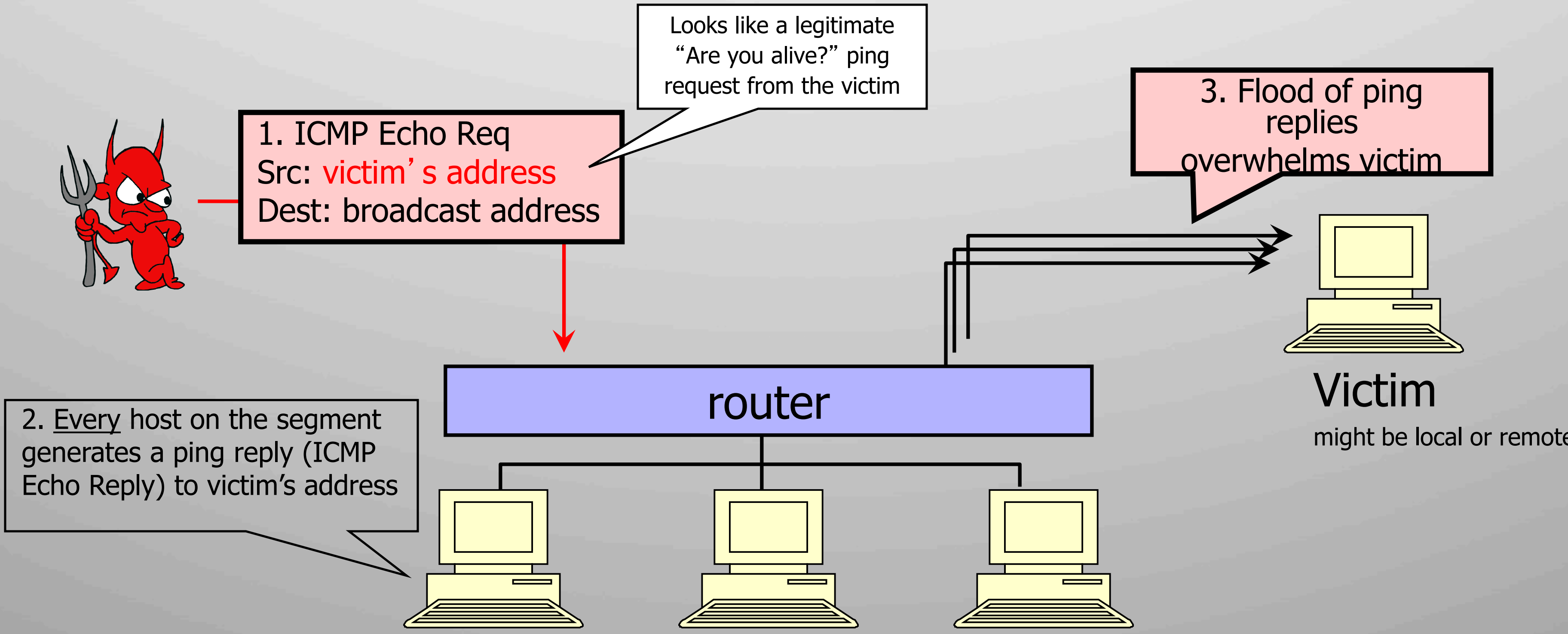
Packet Sniffing

- Many old applications send data unencrypted
  - Plain ftp, telnet send passwords in the clear (as opposed to sftp and ssh)
- Network Interface Card (NIC), e.g., Ethernet device, in “promiscuous mode” can read all data on its broadcast segment



Solution: encryption (e.g., IPsec), improved routing

"Smurf" Attack



Solution: reject external packets to broadcast addresses

"Ping of Death"

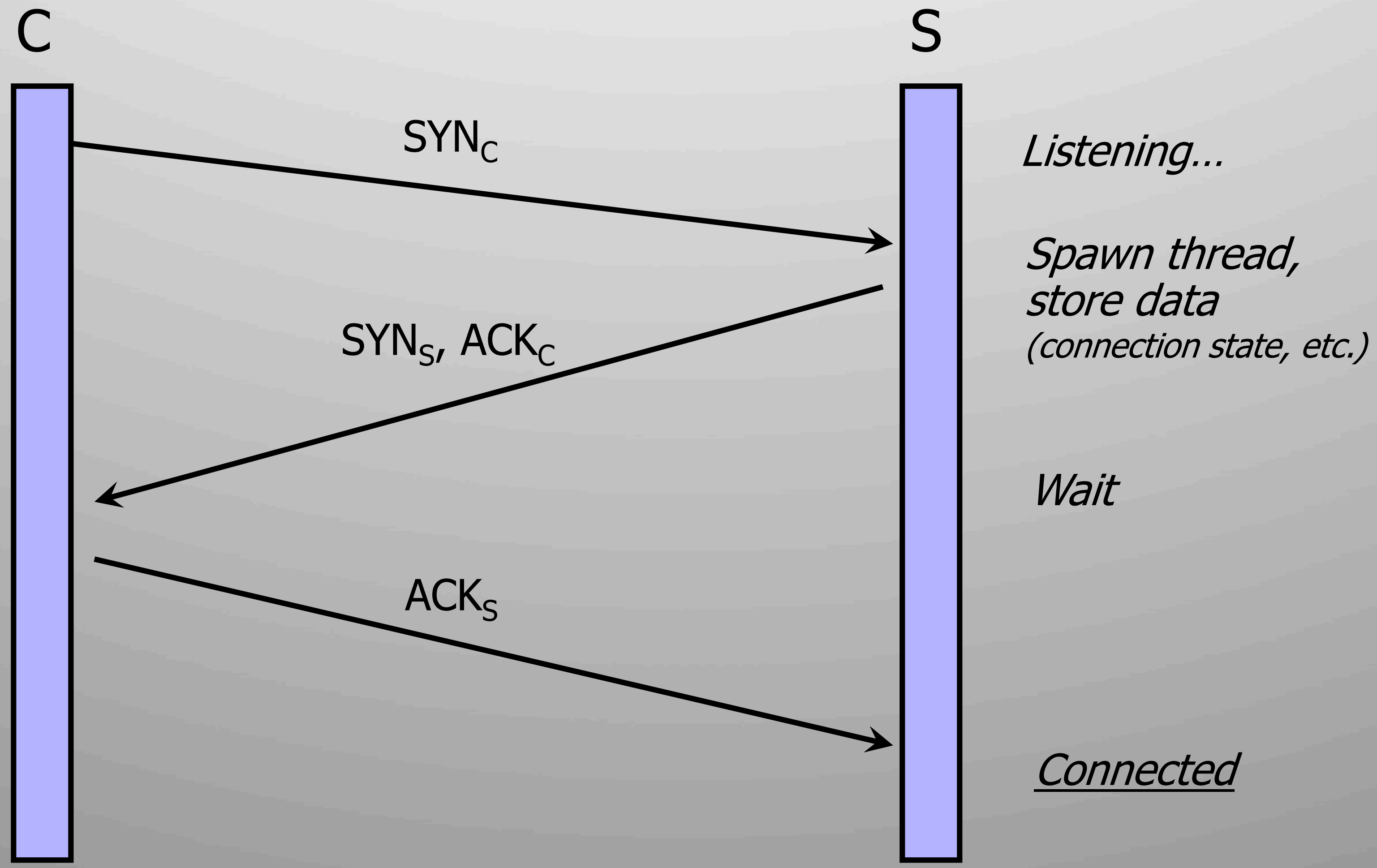
- When an old Windows machine receives an ICMP packet with payload over 64K, it crashes and/or reboots
  - Programming error in older versions of Windows
  - Packets of this length are illegal, so programmers of old Windows code did not account for them

Solution: patch OS, filter out ICMP packets

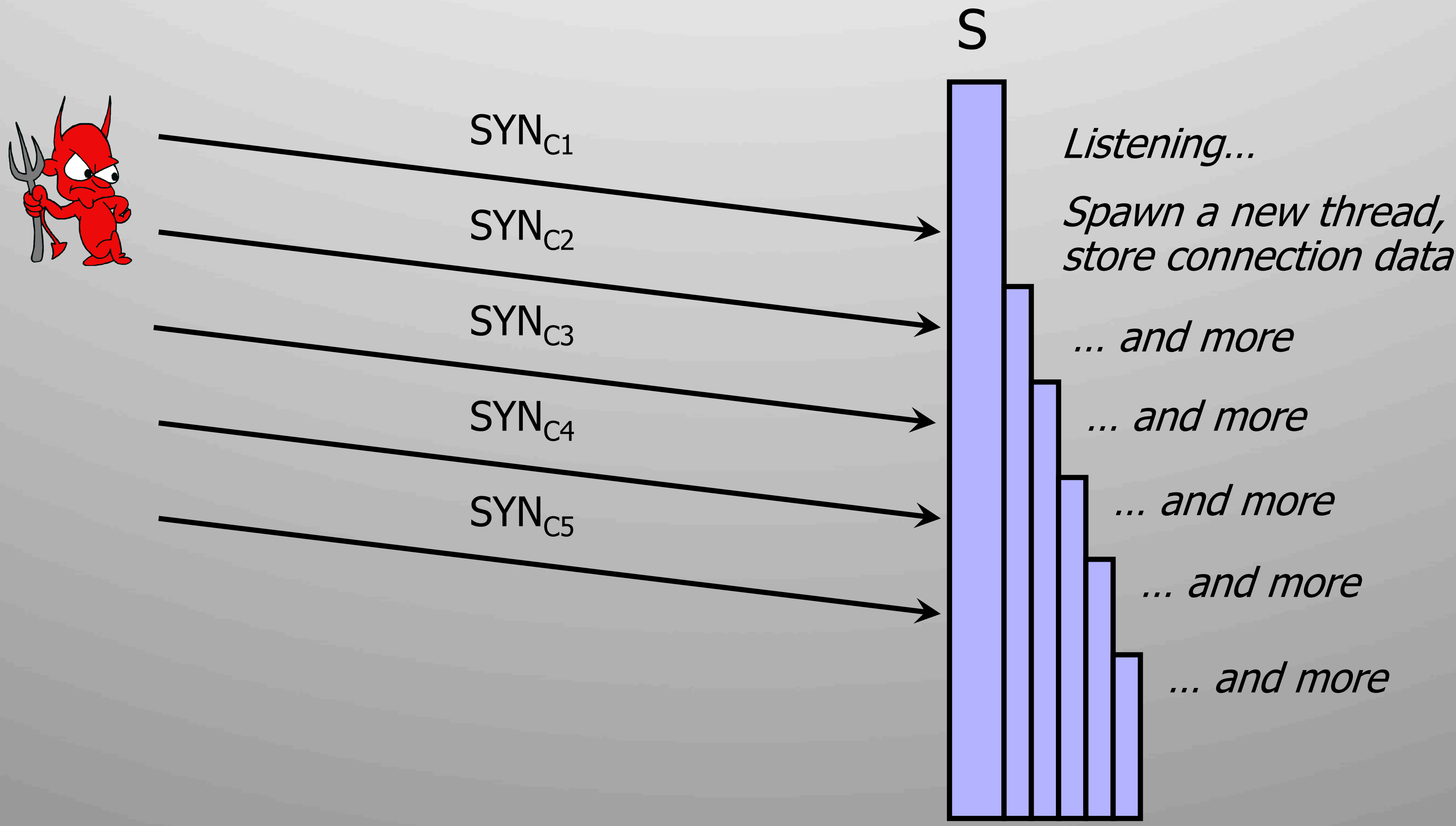
Security Issues in TCP/IP

- Network packets pass by and thru untrusted hosts
  - Eavesdropping (packet sniffing)
- IP addresses are public
  - E.g., Ping-of-Death, Smurf attacks
- **TCP connection requires state**
  - **SYN flooding**
- TCP state easy to guess
  - TCP spoofing and connection hijacking

TCP Handshake Reminder



SYN Flooding Attack



SYN Flooding Explained

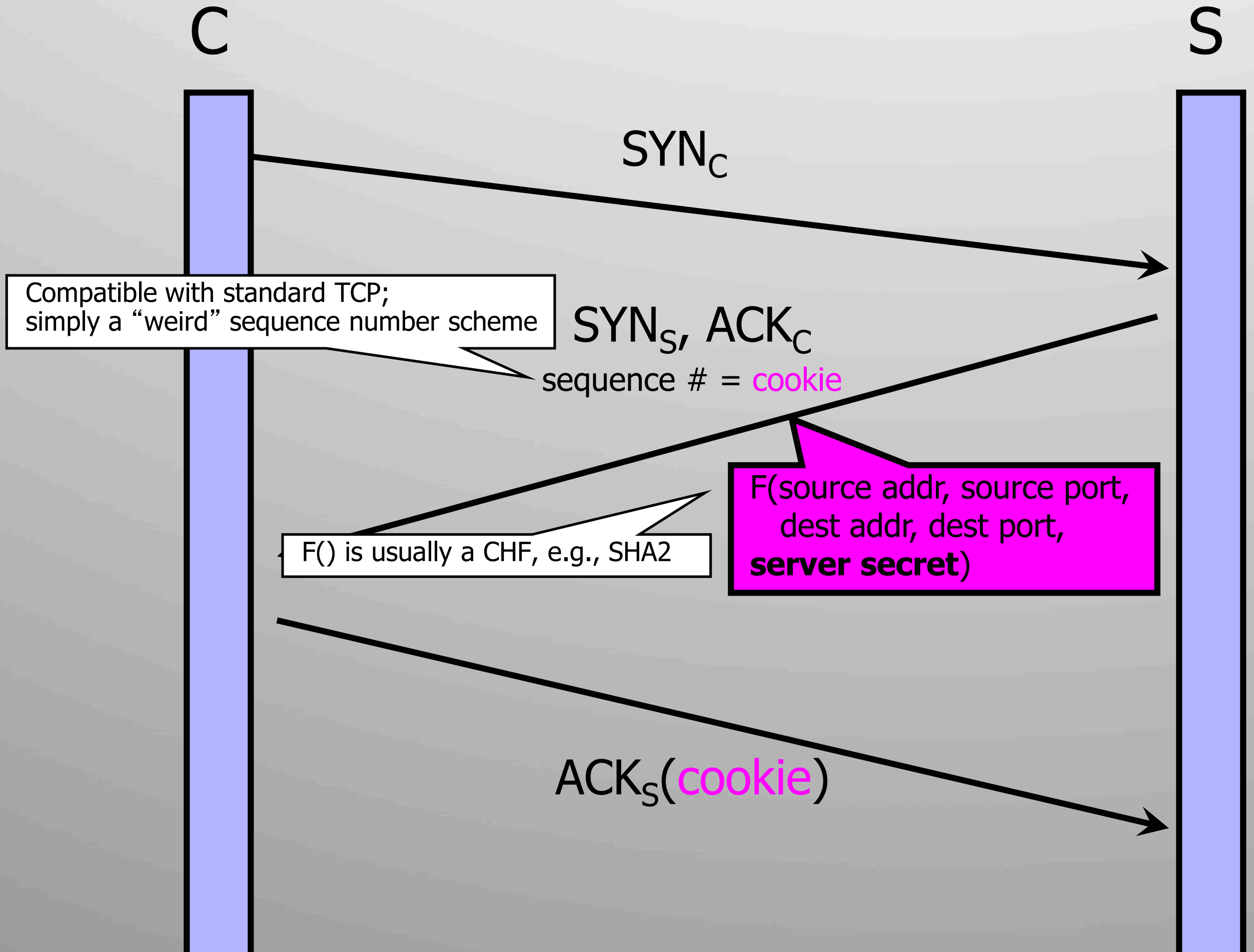
- Attacker sends many connection requests (SYNs) with **spoofed source (IP) addresses**
- Victim allocates resources for each request
  - New thread, connection state maintained until timeout
  - Fixed bound on half-open connections
- Once server resources are exhausted, requests from legitimate clients are denied
- This is a classic DoS attack example: ASYMMETRY!!!
  - Common pattern: it costs nothing to TCP client to send a connection request, but TCP server must spawn a thread for each request
  - **Other examples of this behavior?**
    - **TLS/SSL server public key decryption**

Preventing Denial of Service

- DoS is caused by asymmetric state allocation
  - If server opens new state for each connection attempt, attacker can initiate many connections from bogus or forged IP addresses
- Cookies allow server to remain stateless until client produces:
  - Server state (IP addresses and ports) stored in a cookie and originally sent to client
- When client responds, cookie is verified

SYN Cookies

[Bernstein and Schenk]



- Cookie must be fresh, and unforgeable
- Client should not be able to invert a cookie (why?)

Recompute cookie, compare with the one received, only establish connection if they match

More info: http://cr.yp.to/syncookies.html

Note: each TCP packet carries a 32-bit seq numbers

Anti-Spoofing Cookies: Basic Pattern

- Client sends request (message #1) to server
- Typical protocol:
  - Server sets up connection, responds with message #2
  - Client may complete session or not (potential DoS)
- Cookie version:
  - Server responds with hashed connection data instead of message #2
    - Does not spawn any threads, does not allocate resources!
  - Client confirms by returning cookie (with other fields)
    - If source IP address is bogus, attacker can't confirm
    - WHY?

Passive Defense: Random Deletion

Table of half-open connections on server

121.17.182.45
231.202.1.16
121.100.20.14
5.17.95.155

- If SYN queue is full, delete random entry
  - Legitimate connections have a chance to complete
  - Fake addresses will be eventually deleted. WHY?
- Easy to implement

TCP Connection Spoofing

- Each TCP connection has associated "state" info:
  - Sequence number, port number, src IP, dst IP, etc.
- TCP state is easy to guess
  - Port numbers are standard, seq numbers are *predictable*
- Can inject packets into existing connections
  - If attacker knows initial sequence number and amount of traffic, can guess current number
  - Guessing a 32-bit seq number is not practical, BUT…
  - Most systems accept a *large window* of sequence numbers (to handle massive packet losses, e.g., in shaky wireless networks)
  - Send a flood of packets with likely sequence numbers

DoS by Connection Reset

- If attacker can guess the current sequence number for an existing connection, can send a **reset** packet to close it (RST flag=1 in TCP header)
- Especially effective against long-lived connections
  - For example, background system services such as push notification

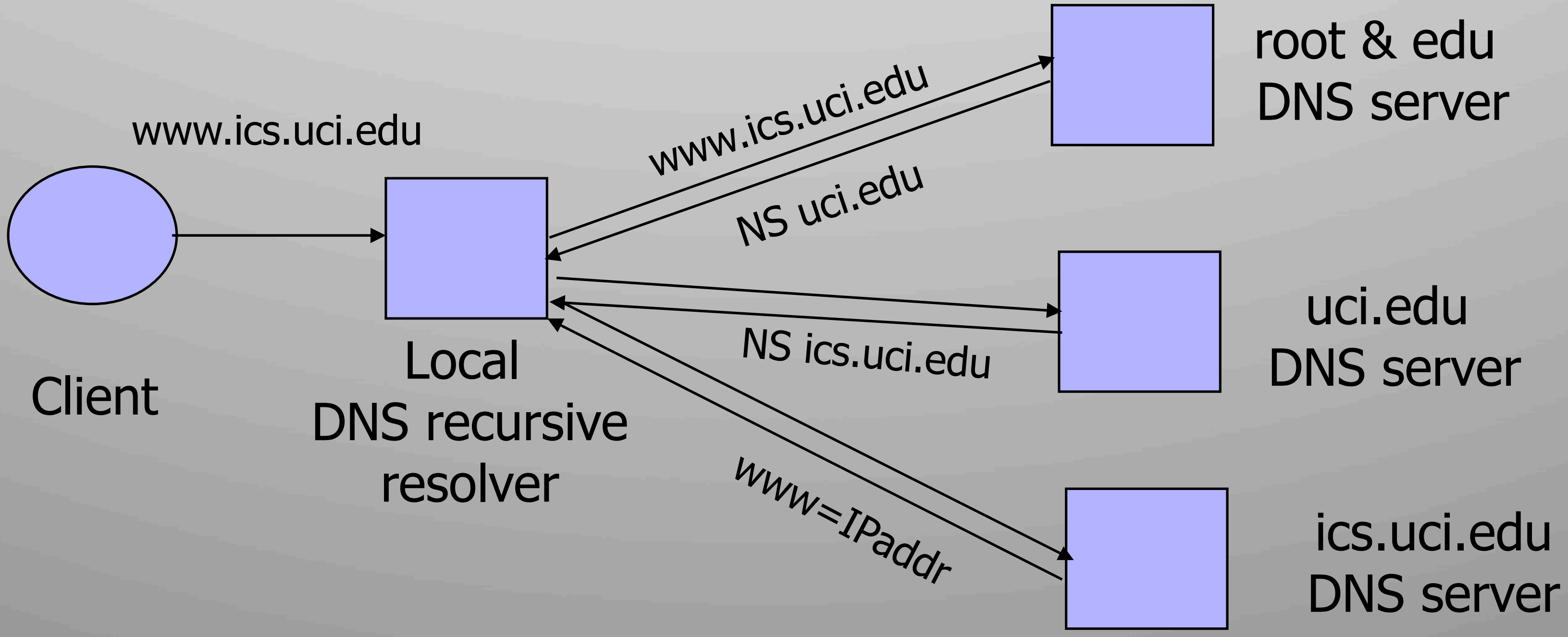
Countermeasures

- Above transport layer: Kerberos
  - Provides authentication, protects against application-layer spoofing
  - Does <u>not</u> protect against connection hijacking
- Above network layer: SSL/TLS and SSH
  - Protects against connection hijacking and injected data
  - Does <u>not</u> protect against DoS by spoofed packets
- Network (IP) layer: IPsec
  - Protects against hijacking, injection, DoS using connection resets, IP address spoofing
  - But muddled/poor key management…

DNS: Domain Name Service

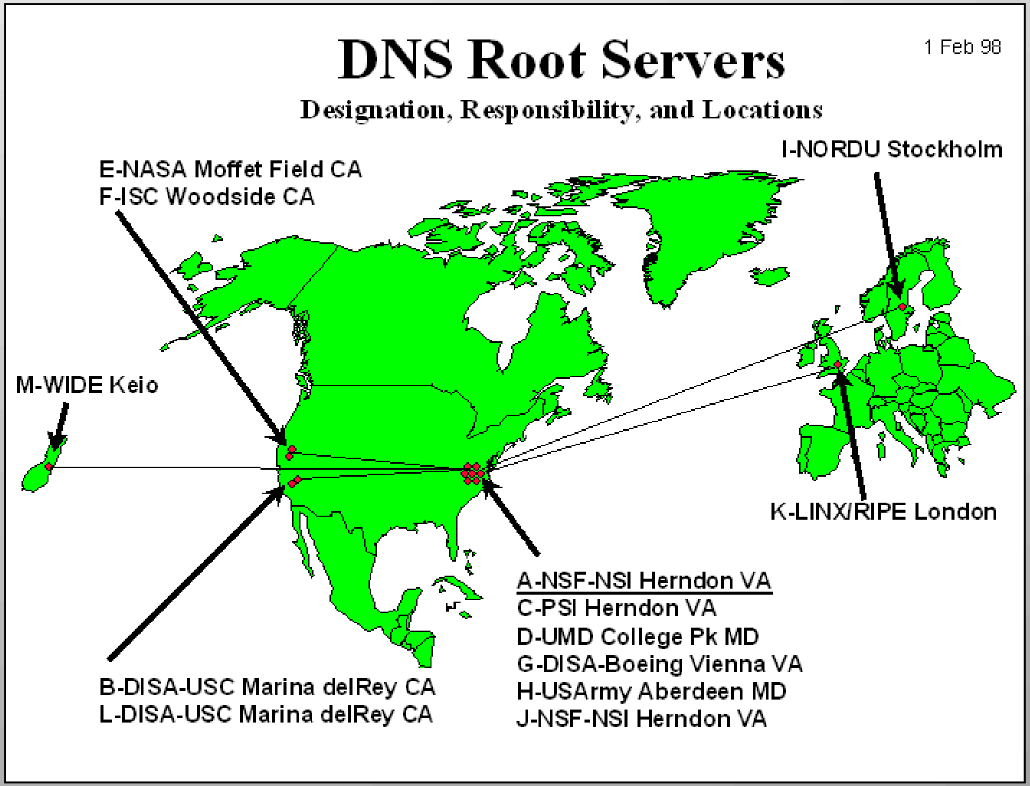
DNS maps symbolic names to numeric IP addresses

(for example, www.uci.edu ↔ 128.195.188.233)



DNS Root Name Servers

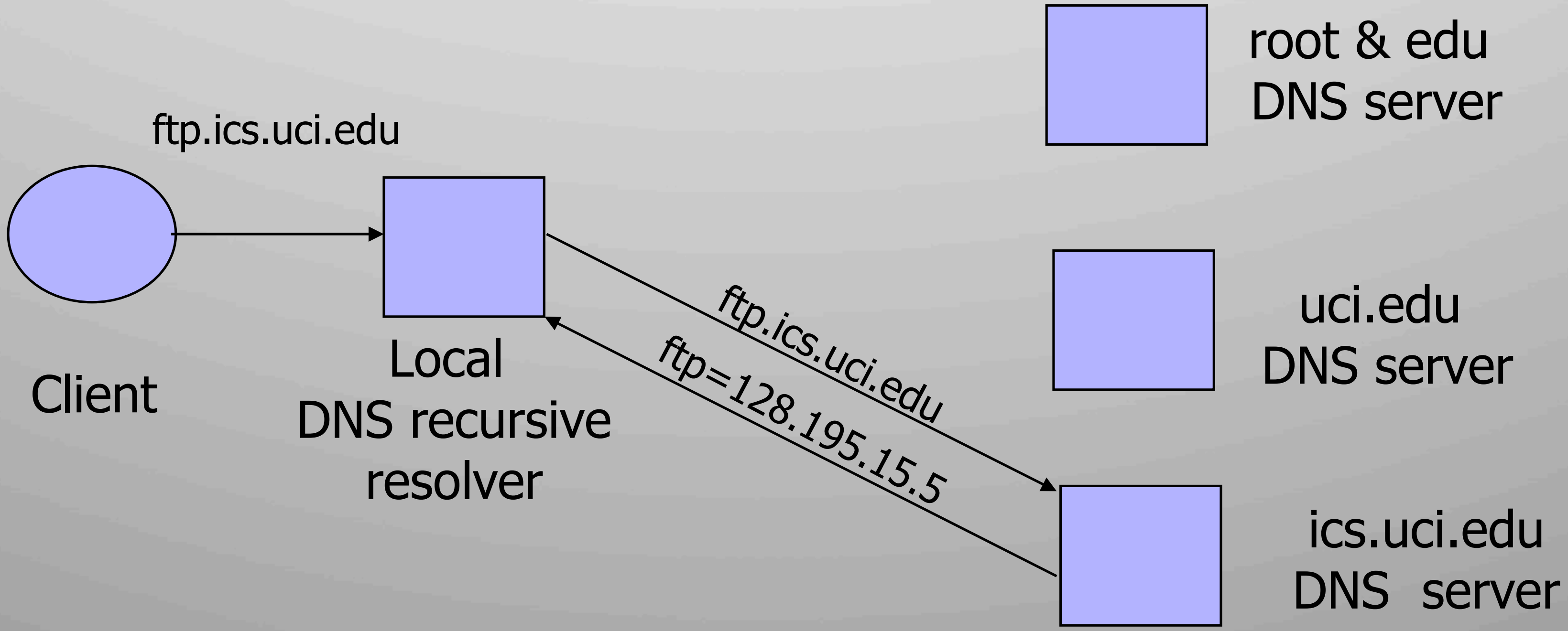
- Root name servers for top-level domains
- Authoritative name servers for subdomains
- Local name resolvers contact authoritative servers when they do not know a name



DNS Caching

- DNS responses are cached:
  - Quick response for repeated queries
  - Other queries may reuse some parts of lookup
    - NS records for domains
- DNS negative queries are cached
  - Don't have to repeat past mistakes, e.g., typos
- Cached data periodically times out
  - Lifetime (TTL) of data controlled by owner of data
  - TTL passed with every record

Cached Lookup Example



DNS “Authentication”