Introduction to Information Security
14-741/18-631 Fall 2021
Unit 4: Lecture 2:
TCP/IP Vulnerabilities

Limin Jia liminjia@andrew

This lecture's agenda

- **Outline**
  - Brief networking review
  - TCP attacks
  - Routing attacks
  - Vulnerable applications
  - Defenses
- **Objective**
  - Examine some of the key vulnerabilities in the widely used TCP/IP and related protocols

Origins of the Internet

- **U.S. military**
  - (D)ARPAnet
  - Most research done in the mid-60s through late 70s
- **Objective**
  - Have a network capable of withstand massive failures
  - e.g., nuclear war
- **TCP/IP designed with reliability and fault-tolerance in mind**
- **Does this automatically provide security?**

TCP/IP

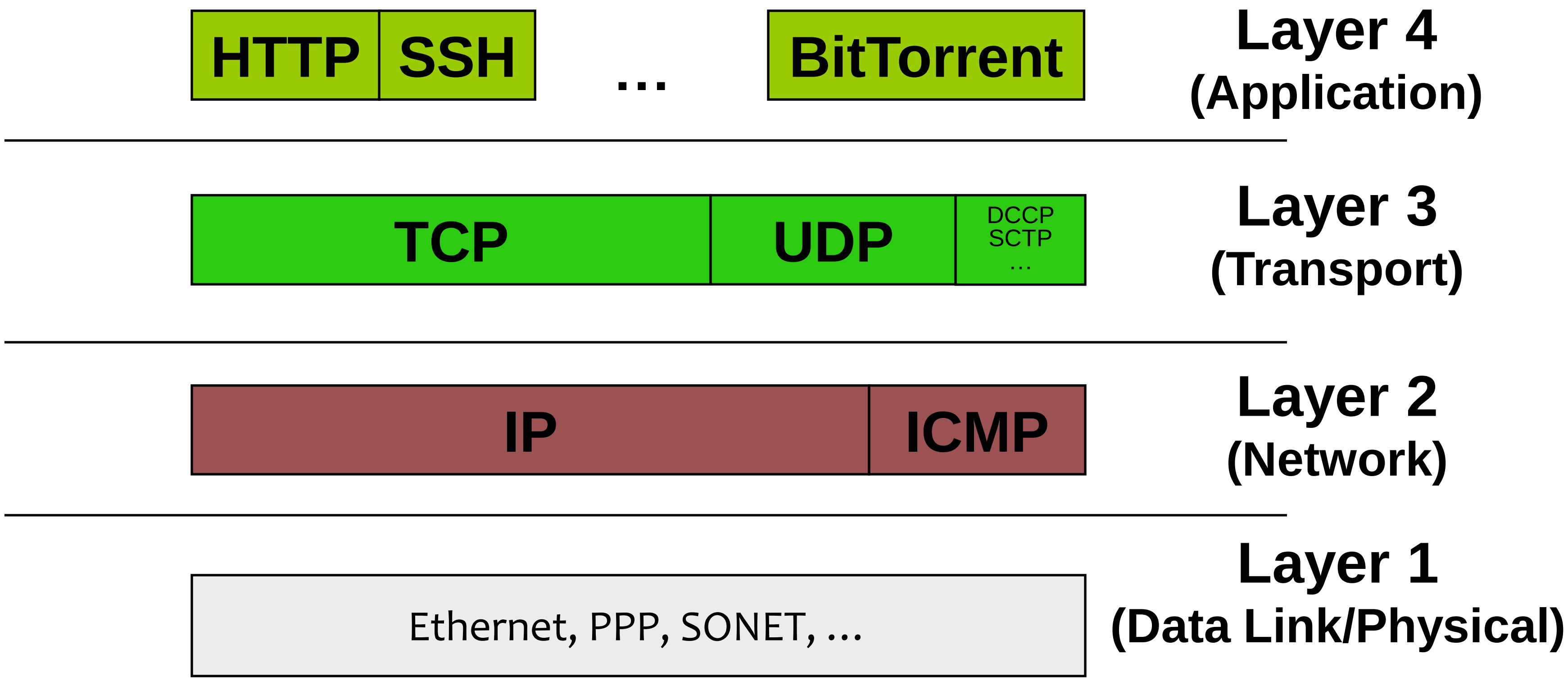
- **TCP/IP protocol suite overwhelmingly used on the Internet**

- **Layered architecture**
  - Different from 7-layer OSI model
  - Essentially no session or presentation layers
  - Physical and data link layer basically merged

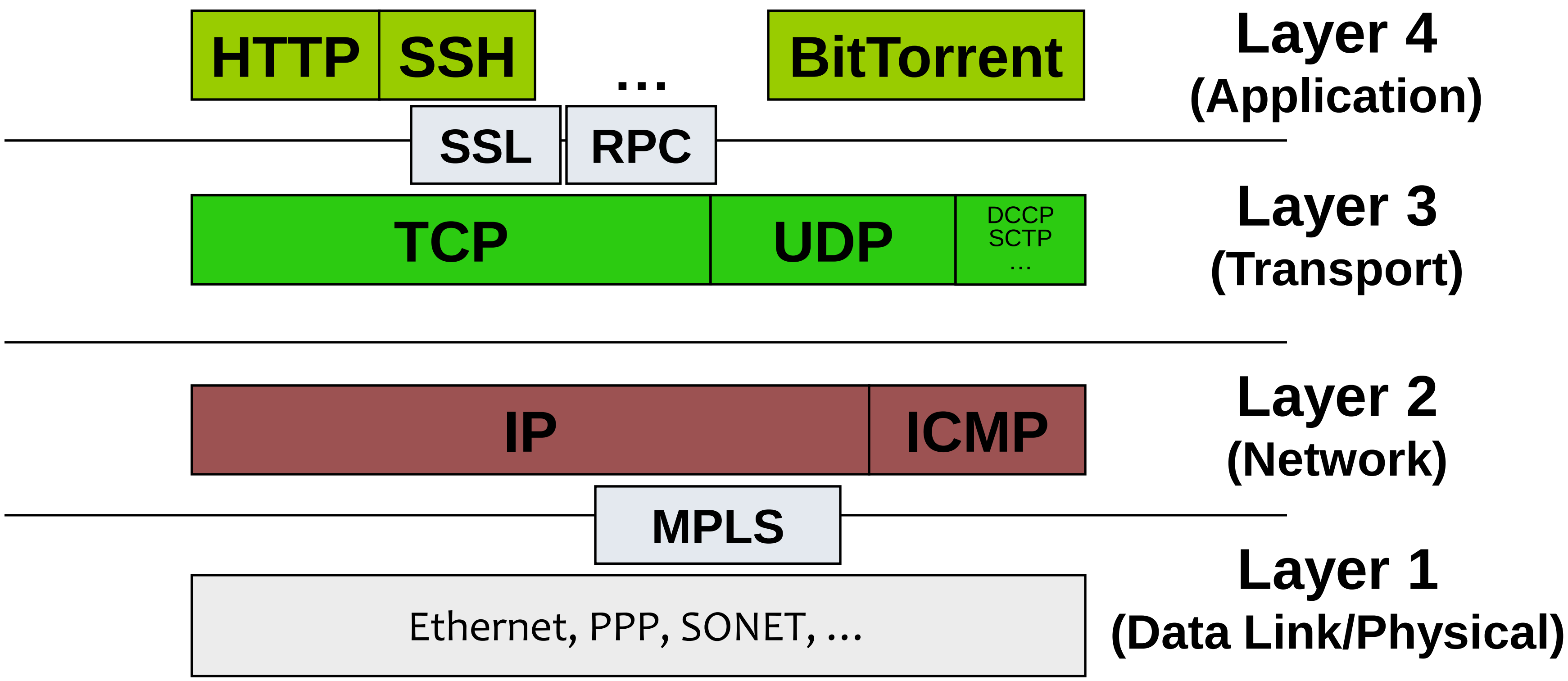
Layered architecture

- **Each layer provides service to layer above and relies on services from layer below**
  - Why?
- **Different functions at each layer**
  - Data link: physical (local) connectivity
  - Network: end-to-end reachability
    - Relies on physical connectivity
  - Transport: reliability, ordering, …
    - Relies on end-to-end reachability
- **In general, layer n only talks with layers n-1 and n+1**
  - Exceptions (aberrations?): HTTP over IP, QUIC,…

TCP/IP protocol stack

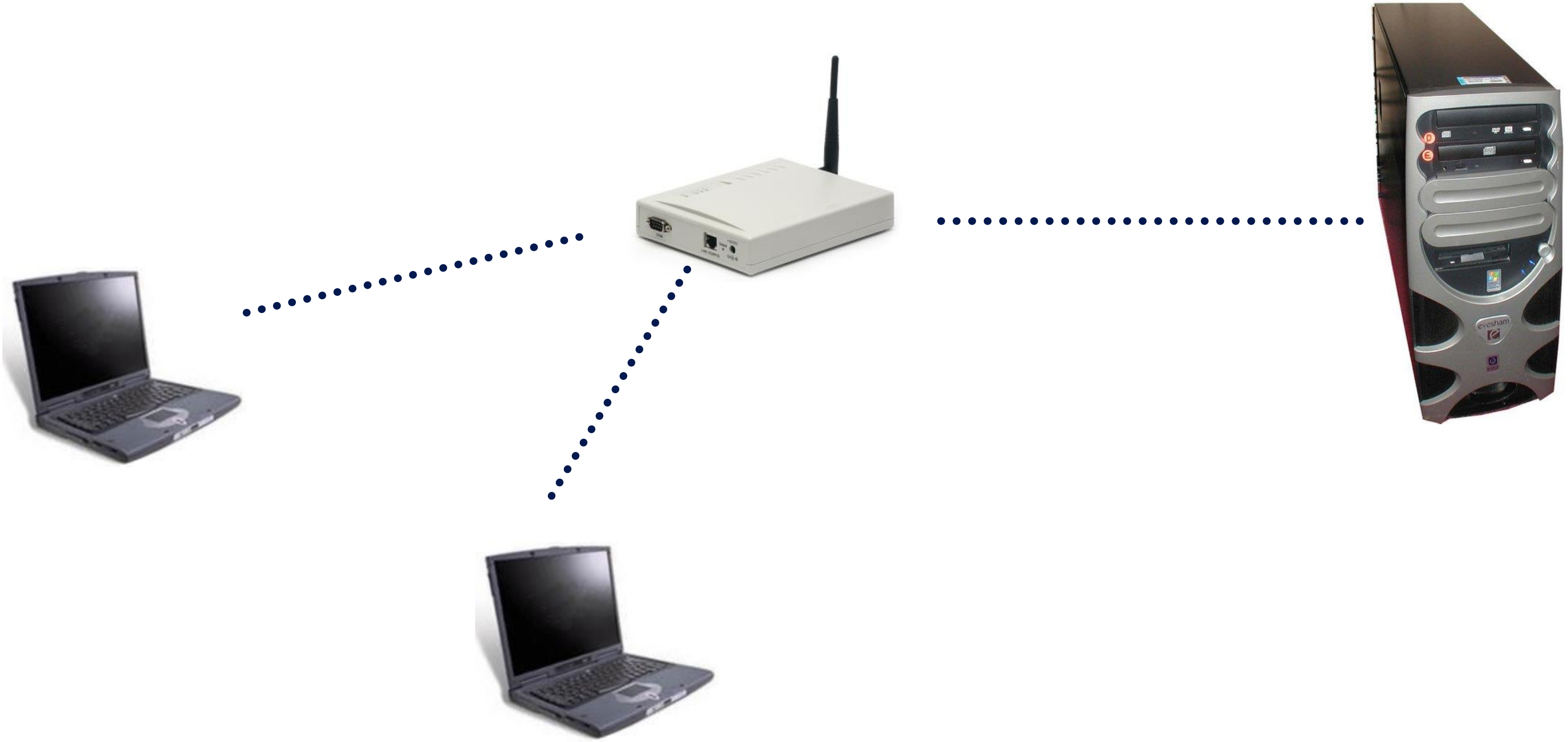


TCP/IP protocol stack



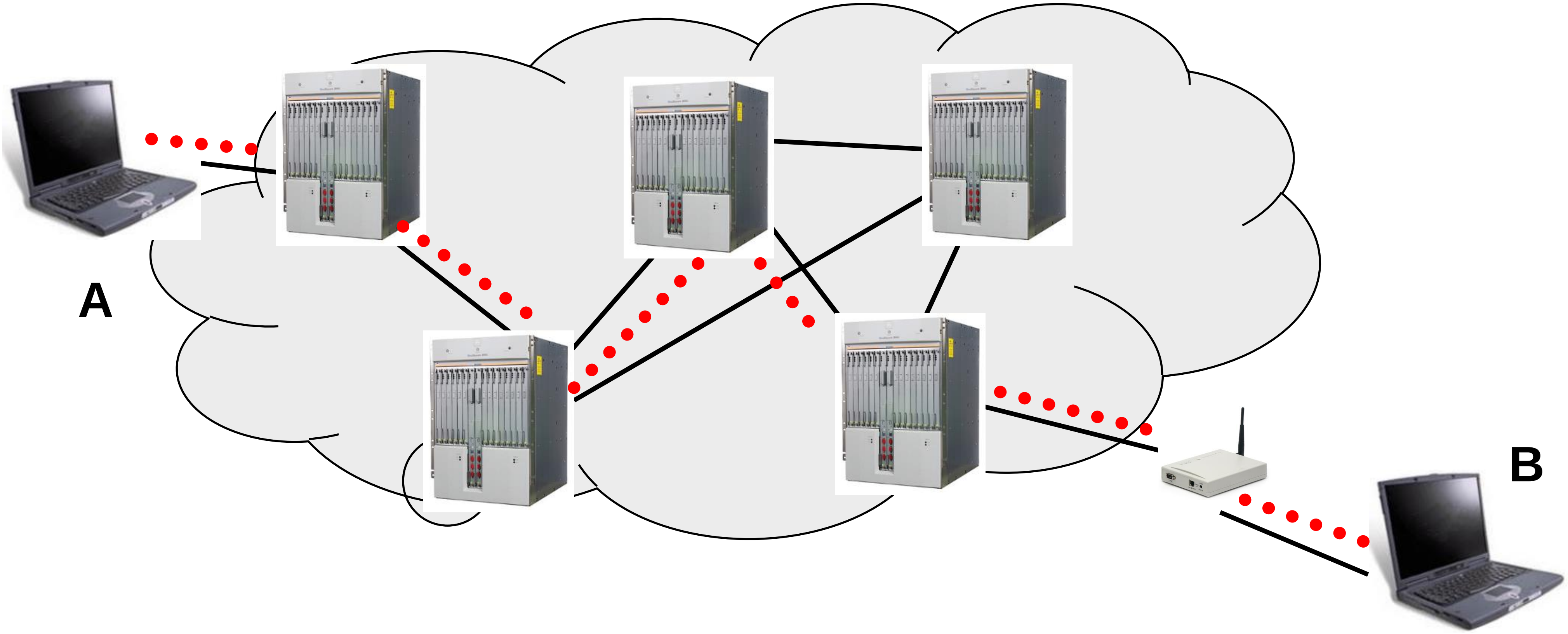
Data link layer: Local connectivity

- Example: 802.11b



Network layer: Global connectivity

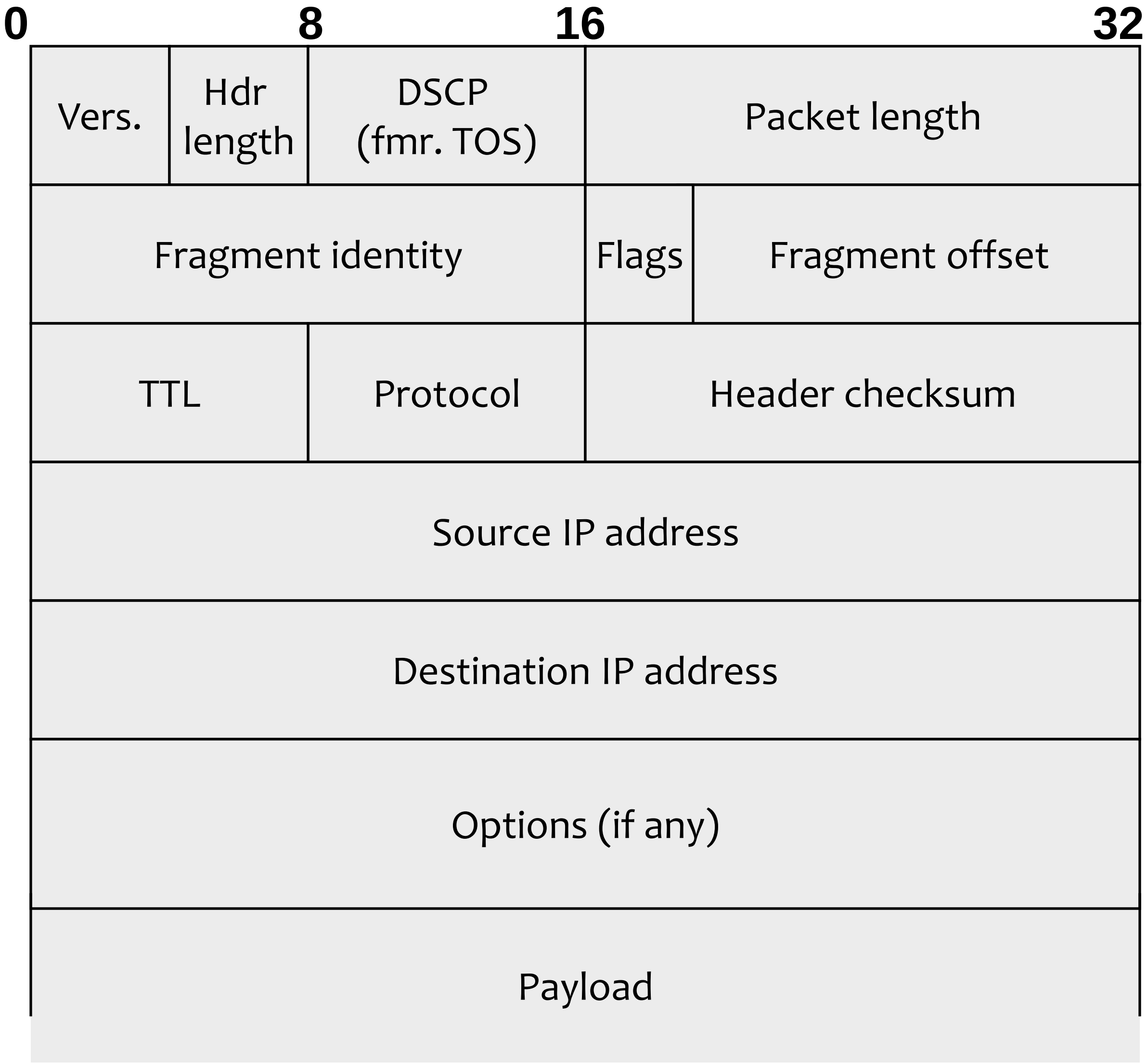
- **Routing:** How does the packet get from A to B?
- **Fragmentation**
  - Making sure each local medium (e.g., 802.11b, Ethernet, SONET) can accommodate all packets
  - Find smallest MTU and break up application layer packets accordingly



More on routing

- **Forwarding tables at each router populated by routing protocols**
  - RIP, OSPF, BGP, …
- **Originally (a long time ago), manually updated**
  - Humans still (too much?) in the loop
  - BGP misconfigurations are a serious problem
- **Routing protocols update tables based on cost of each route**
  - Exchange tables with neighbors, or everyone (depends on protocol)
  - Use neighbor leading to the shortest path

IP packet format



Transport layer: Reliability

- **Connection-oriented service**
  - Network layer only provides connectionless service
- **Provides resilience to**
  - Data corruption
    - Payload checksum
  - Data loss
    - Sequence numbers, timeouts + retransmits
  - Out-of-order delivery
    - Sequence numbers
  - Congestion
    - Flow control, AIMD
- **Wait… What about UDP?**
  - Simple (de)multiplexing primitive over IP

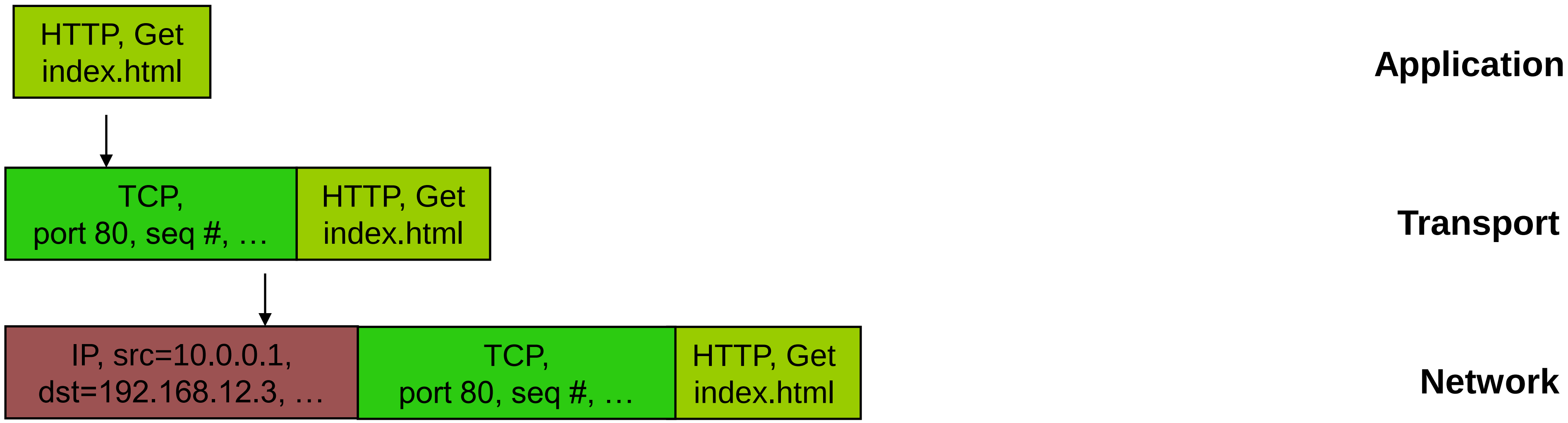
Layer encapsulation example

HTTP, Get index.html

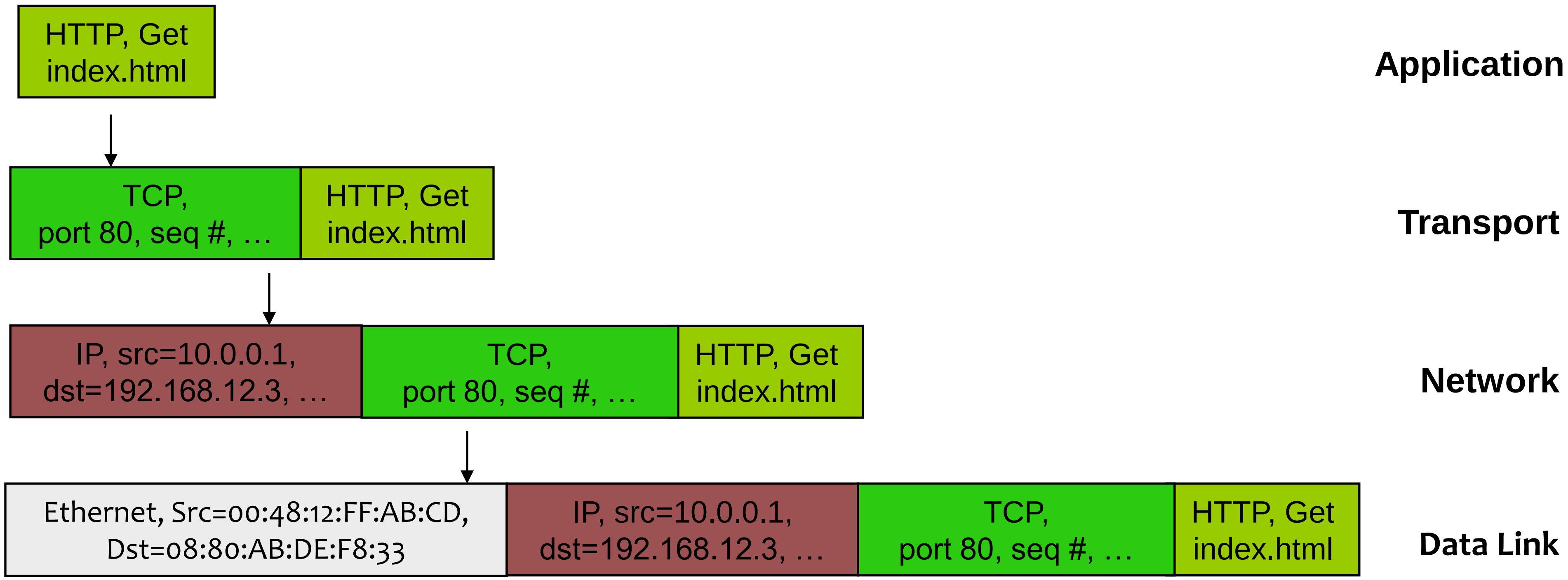
Application

Layer encapsulation example

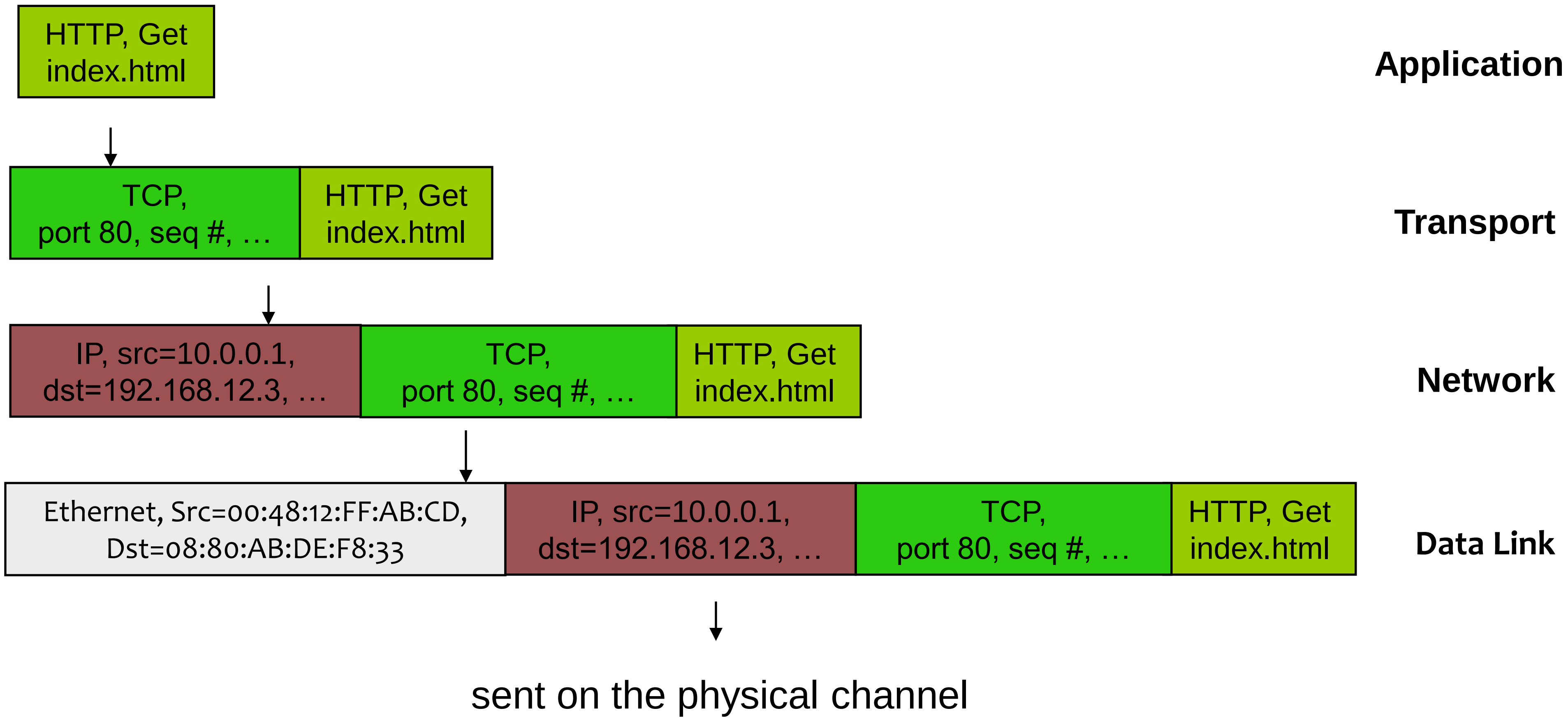
Layer encapsulation example



Layer encapsulation example



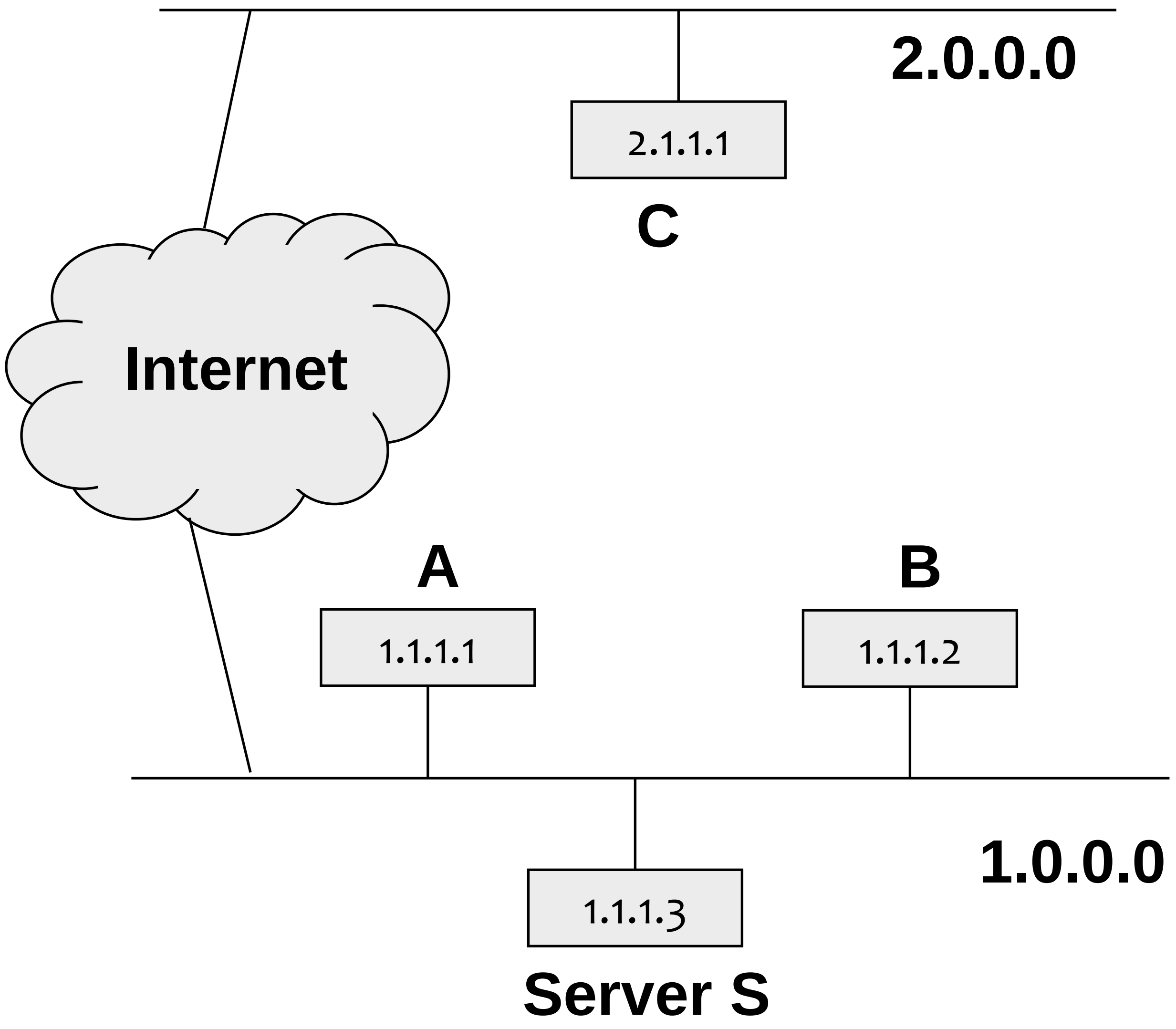
Layer encapsulation example



Naming/Identification

- **IP addresses vs. readable host names**
  - Humans are not good at remembering numbers, use E.g., www.cmu.edu
  - Globally unique (but can correspond to multiple hosts, e.g., www.google.com)
- **Naming system translates readable host names to physical address**
  - DNS translates name to IP address (e.g.,128.2.42.52)
  - Address reflects location in the network
- **A Common authentication problem**
  - IP-based authentication/identification can be defeated by IP address spoofing

Network security issues



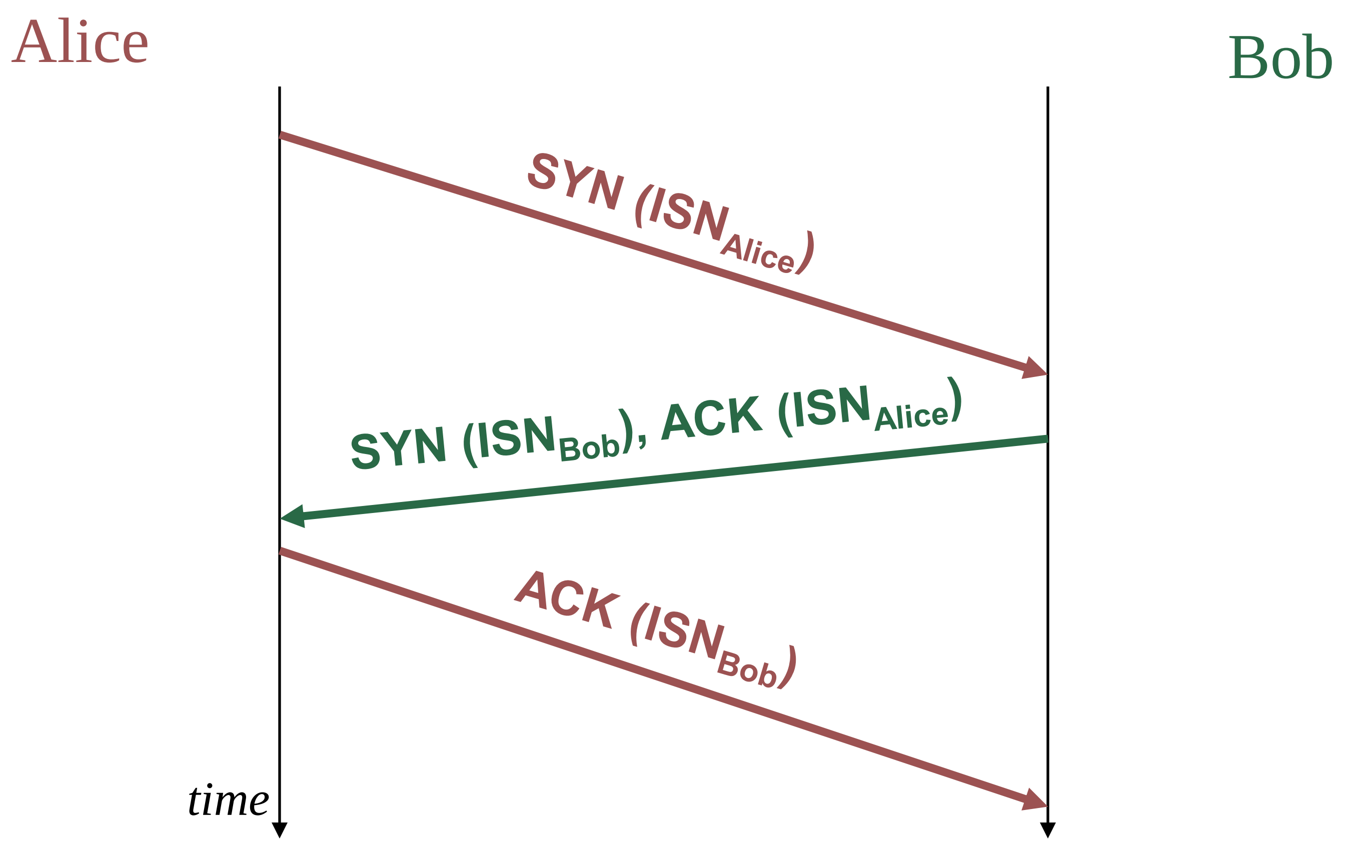
- Message authentication?
- How can S know that the packet really originated from A?
- Can B impersonate A to S?
- Can C impersonate A to S?

- Shared networks are easy to eavesdrop on
  - E.g., just set interface in promiscuous mode
    tcpdump, ethereal
  - B can easily eavesdrop on the packet A sent to S
- Note: eavesdropping or injection of arbitrary packets easy in today's networks

Security problems in the TCP/IP protocol suite

- **By Steven M. Bellovin, 1989**
- **Classic network security paper**
- **Wakeup call for networking researchers**
- **Paper lists many security vulnerabilities**
- **Attacks studied in this lecture**
  - IP-based authentication
  - TCP level attacks
  - Routing attacks
  - Application-level attacks
- **More next time, and even more next semester**

Review: TCP 3-way handshake

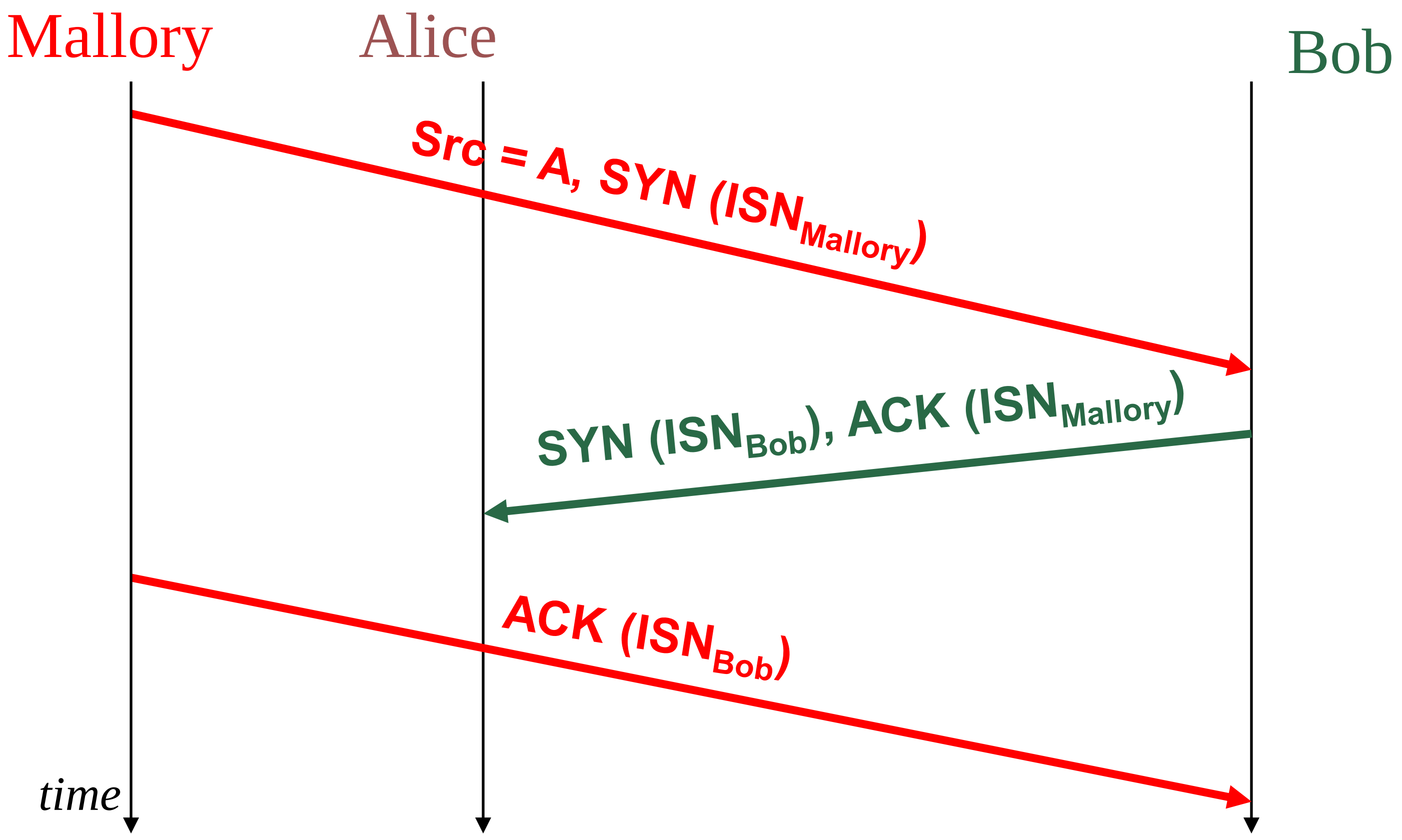


(Data exchange)

TCP initial sequence number (ISN) prediction attack

- **Objective of the attack: Impersonation of a communication participant**
  - E.g., Alice may talk to Bob
  - Mallory can pose as Alice,
    even though Mallory cannot intercept traffic between Alice and Bob
- **Risk: severe, particularly when authentication is based on IP address**
  - e.g., rsh: arbitrary execution of commands possible

Abusing the TCP 3-way handshake



(Mallory can send commands to Bob as Alice)

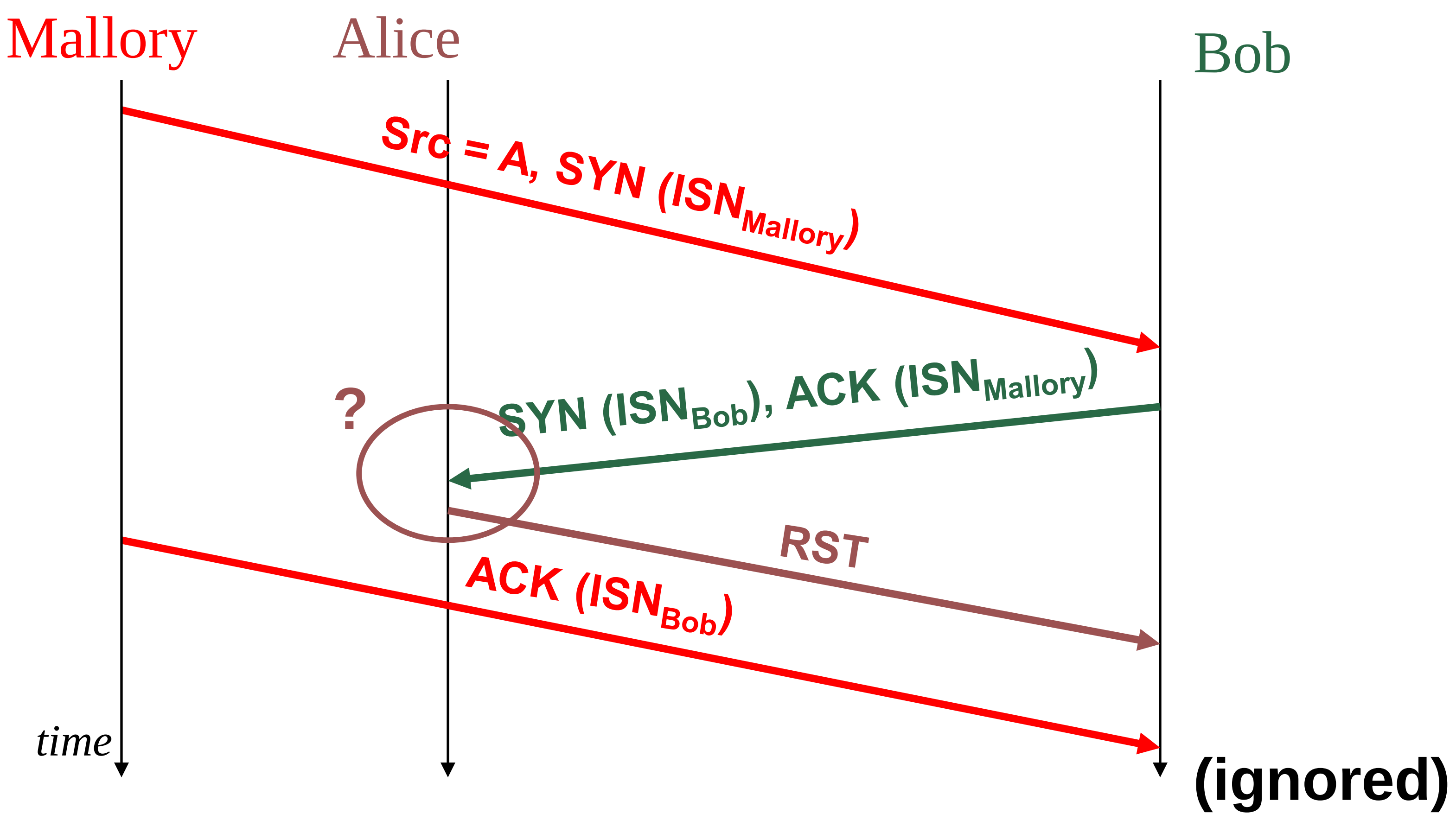
Predicting ISN_{Bob}

- **For the attack to be successful, Mallory needs to be able to predict ISN_{Bob}**
  - Easy on a LAN (just eavesdrop)
  - But not much harder on a WAN…
- **Flawed TCP ISN choices**
  - Always start at same ISN
  - After each connection, ISN++
  - ISN = (c1+c2*(time in ms)) mod 2^{32}

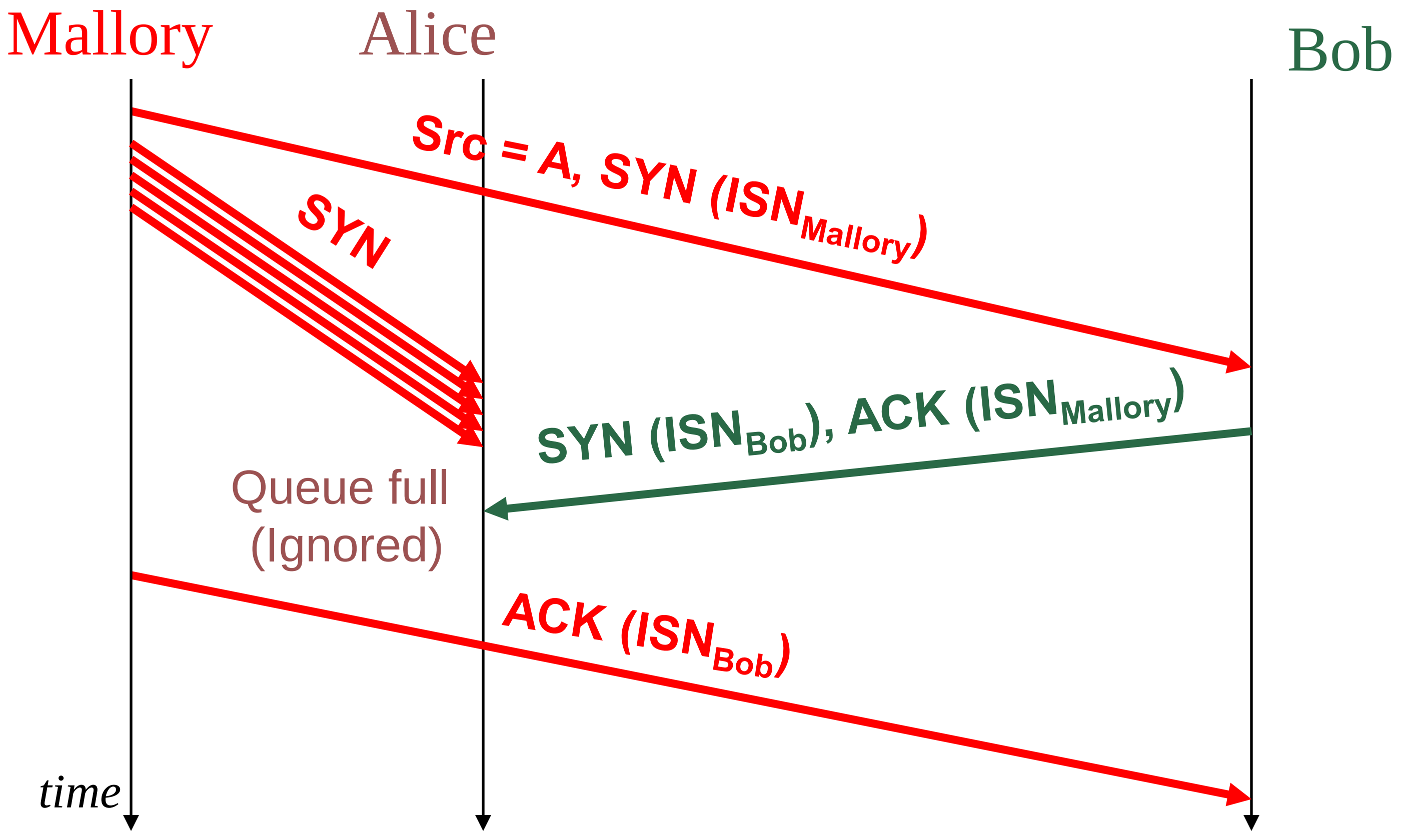
Better ISN choices

- **ISN is should be unpredictable!**
- **ISN = rand() function of C library?**
  - How random is rand()?
  - Especially at boot time?
- **current ISN = H(previous ISN)?**
- **ISN = AES_K(counter++)?**

Doesn’t Alice reset the connection?

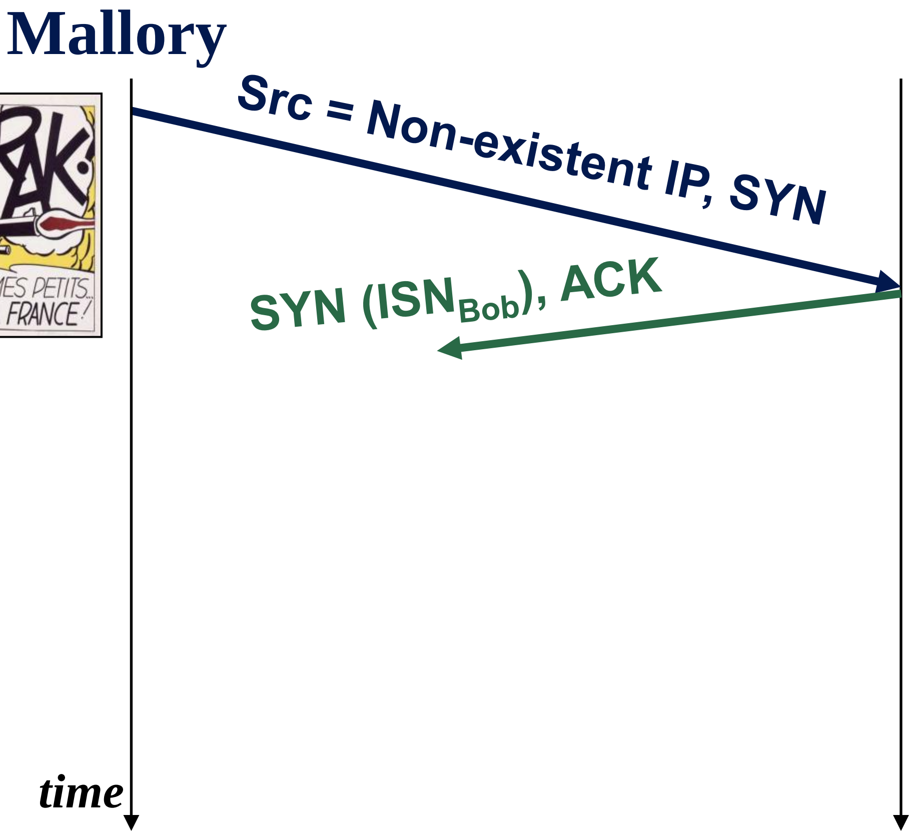


Abusing the TCP 3-way handshake (revised)



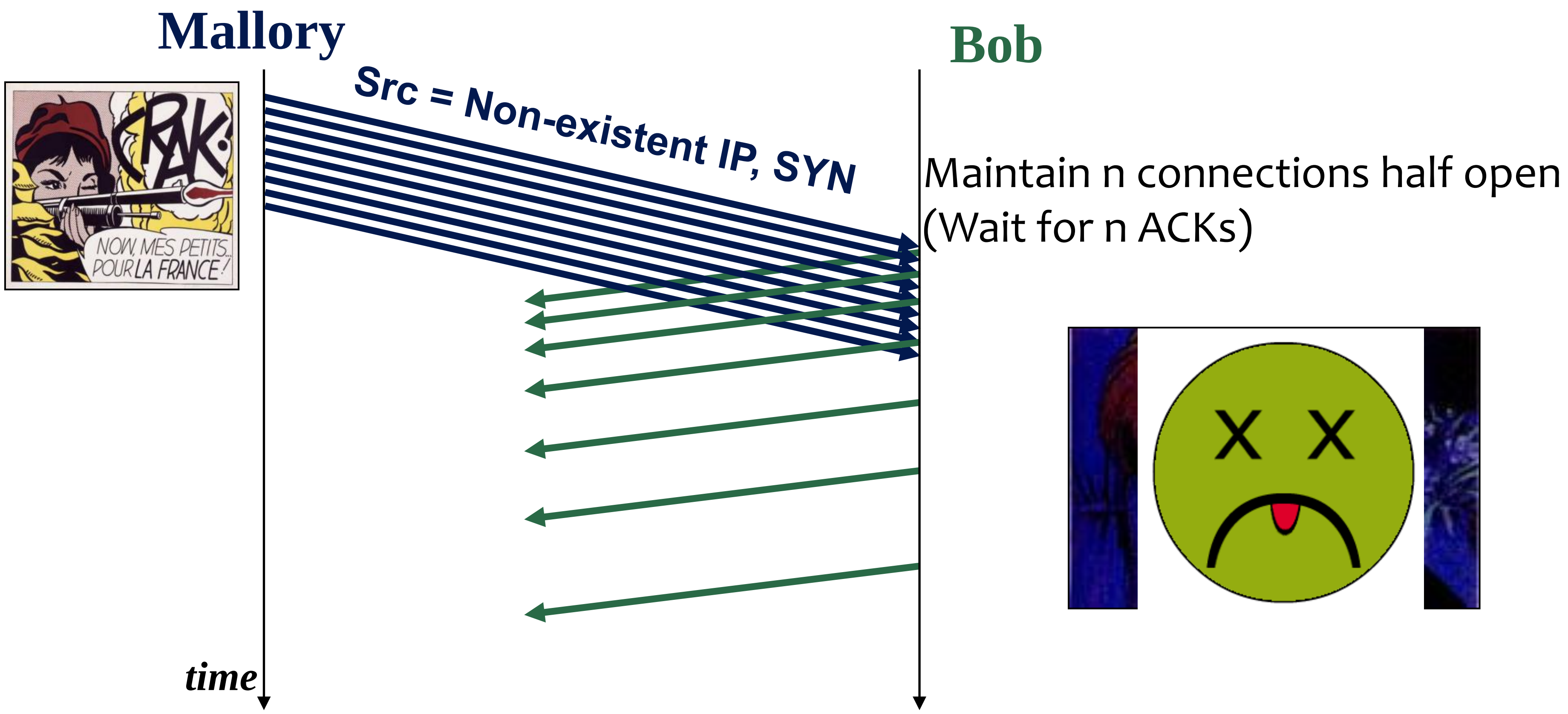
(Mallory can send commands to Bob as Alice)

TCP SYN flood



Maintain 1 connection half open
(Wait for ACK)

TCP SYN flood



SYN flood details

- **Why does server exhaust resources?**
  - Need to store requests for 511 seconds
  - Server has finite-size queue for incomplete connections, usually 1024 entries
- **Memory is cheap, why not store all requests?**
  - 736 bytes per connection…
- **Why store any information at all?**
  - If SYN ACK dropped by network, server re-sends SYN ACK until timeout or client sends ACK, otherwise legitimate clients will wait
  - TCP options (performance enhancements) need to be stored, otherwise slow connection

Defense against SYN flood: TCP SYN cookie

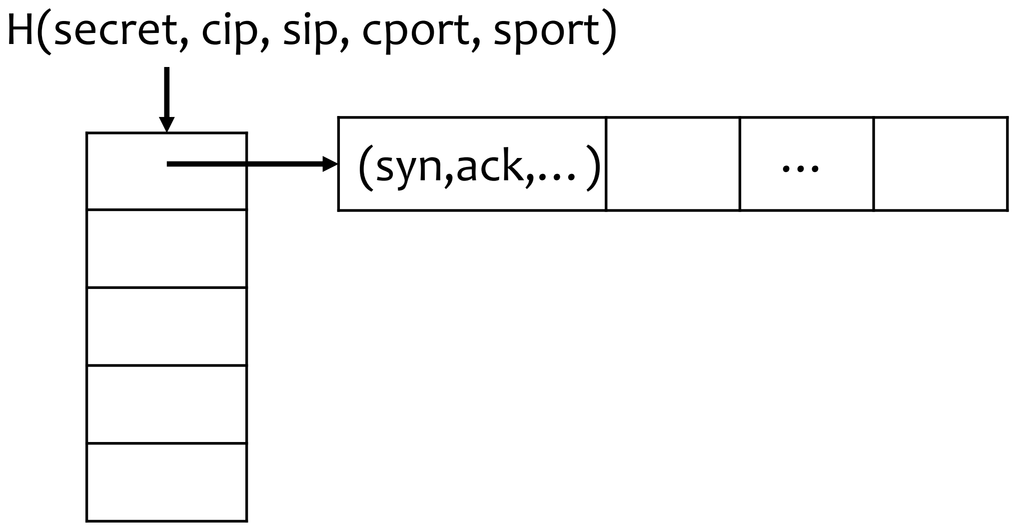
- **Approach**
  - Server computes server ISN (SYN cookie) based on connection state, reply to client, server keeps no state after SYN
  - Client sends ACK with cookie
  - Server verifies cookie, if correct, allocates connection state, reconstruct the connection state (with some information loss such as special TCP options)
- **What about: Cookie = H(SIP, CIP, Sport, Cport)?**
- **Better: Cookie = H(SIP, CIP, Sport, Cport, skey)**

Question

- **Connection state is much larger than incomplete (half-open) connection state**
- **Why does client not follow up with ACK packet in TCP SYN flooding?**
  - No IP source address spoofing
    - Server can easily identify attacker, since it has thousands of open connections
    - Server can block all connections from attacker IP
  - IP source address spoofing
    - Attacker does not receive SYN ACK, does not know server ISN, cannot send ACK

TCP SYN cache

- **Best strategy: keep efficient SYN cache for incomplete connections, only if too many requests send SYN cookies**
- **SYN cache design**
  - Efficient hash table
  - Each hash table entry has linked list with all incomplete connections
  - Limit number of entries on linked list per bucket
  - Compute hash with a secret



Results

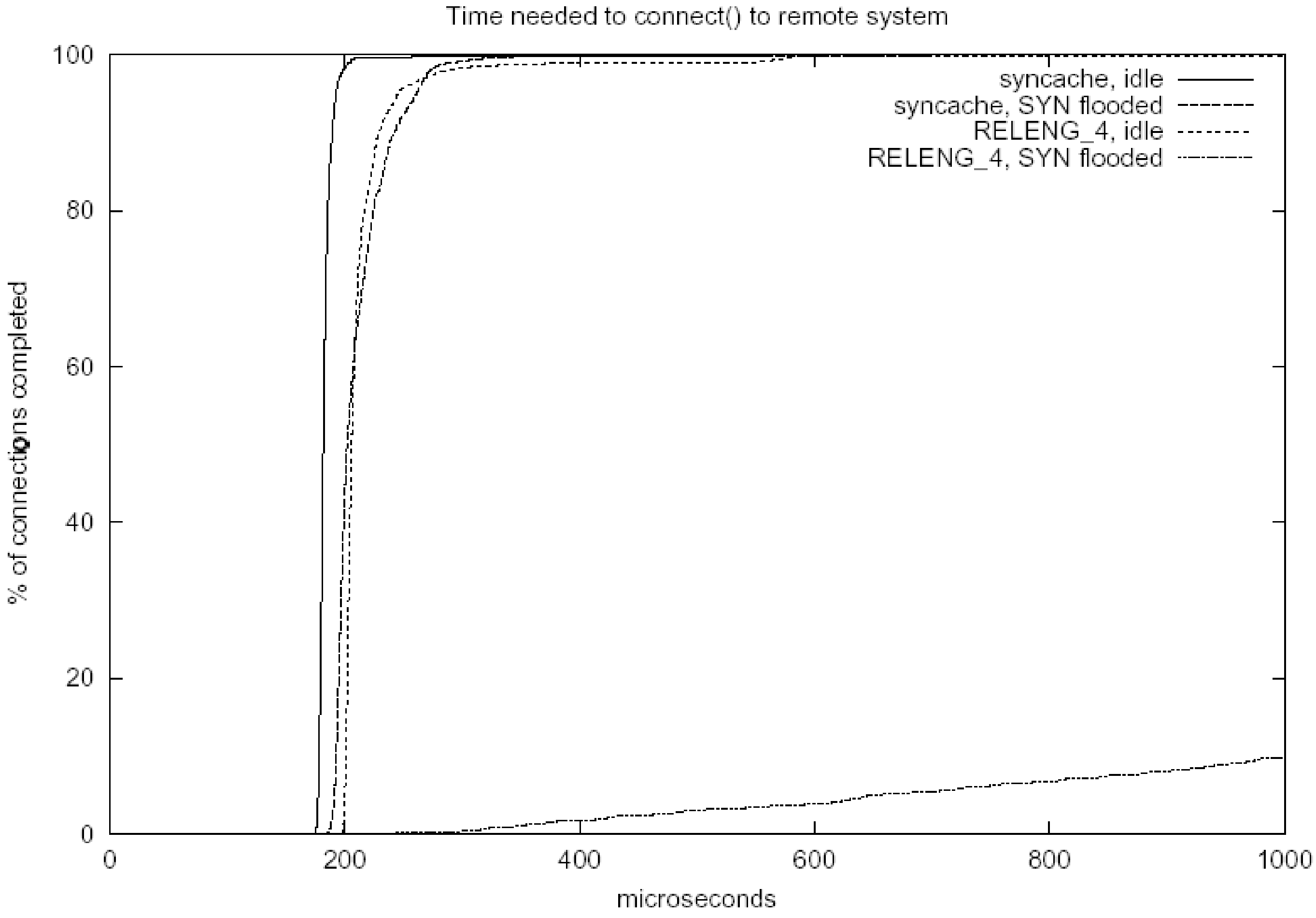


Figure 3: Time needed to connect() to remote system.

Tampering routing tables

- Routing tables updated from information received from topological neighbors
- Checks are non-existent or casual (IP address-based authentication)
- Possibility to route all traffic through a malicious host

Defending against routing attacks

- **Authentication of all packets**
  - is difficult for broadcast protocols (e.g., RIP)
- **Do not accept new routes to your own networks**

Application problems

- **Finger service gives out information useful for password cracking**
- **User name and password are often sent in the clear**
  - POP email retrieval protocol
  - FTP
  - Telnet
- **TFTP causes many weaknesses**
  - e.g., home routers, remote booting
- **DNS insecurity**

Defenses

- Authenticated communication prevents packet injection attacks
- Encrypted communication protects clear text
- IPsec, SSH, TLS/SSL provide secrecy and authentication

1989… That’s old, isn’t it?

- **Aren’t most of the attacks listed in the paper defeated by now?**
  - You wish…
    - A number of network appliances (cheap, or less cheap) still have predictable TCP ISNs
    - Some servers still run unencrypted POP/SMTP/IMAP or services like fingerd, *r*-services (rarer, but still)
  - Even for attacks that have been addressed, we need to learn from our mistakes
    - Source routing is evil, we disable it
    - Great, but then we advocate MPLS, which may use source routing, overlay source routing, …

Take away slide

- **TCP/IP remarkable at fault-tolerance…**
- **… but not built with malicious attackers in mind**
  - To their credit, the designers of TCP/IP **did** foresee most of these problems
  - But crypto-based solutions couldn't easily make it into open standards back then
- **Be careful with what you assume from a protocol**
  - TCP and IP **do not provide any authentication**
    - An IP address is just that: an address, **not** an identity
  - Host-based authentication is an heresy
  - Need additional mechanisms (Kerberos, …)
- **Relatively easy to abuse application-layer services**
  - Information services (SNMP, …) are covert channels leaking information