Developed and Presented By Dr. Mehrdad Sepehri Sharbaf
CSUDH
Computer Science Department

http://csc.csudh.edu/

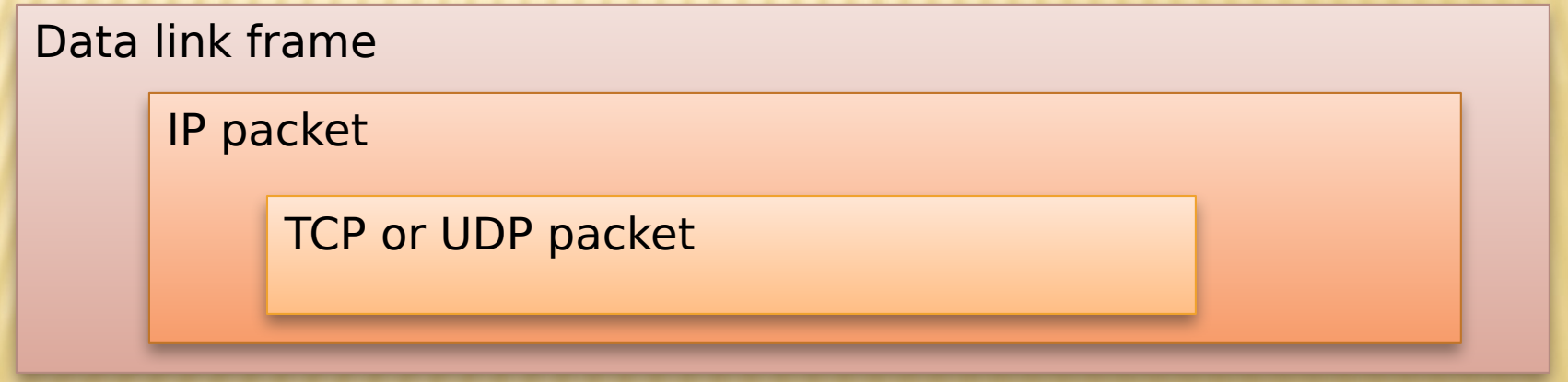
The some of the materials are excerpted from Michael T. Goodrich & Roberto Tamassia,'s Book, and Ross Anderson's Book

ATTACK ON NETWORK AND DEFENSE

NETWORKS: IP AND TCP

INTERNET PROTOCOL

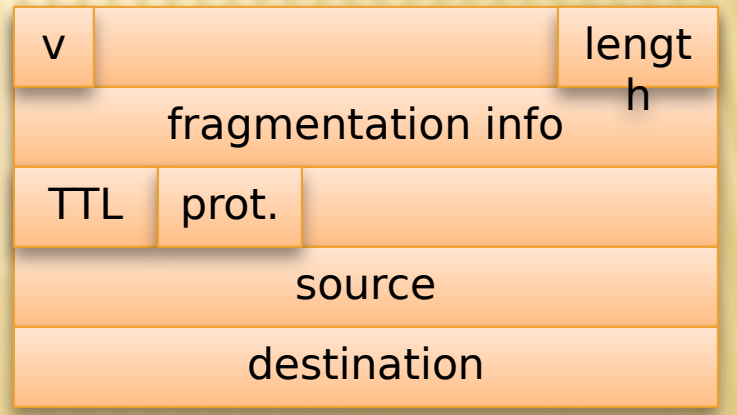
- Connectionless
  - Each packet is transported independently from other packets
- Unreliable
  - Delivery on a best effort basis
  - No acknowledgments
  - Packets may be lost, reordered, corrupted, or duplicated
- IP packets
  - Encapsulate TCP and UDP packets
  - Encapsulated into link-layer frames



IP ADDRESSES AND PACKETS

- IP addresses
  - IPv4: 32-bit addresses
  - IPv6: 128-bit addresses
- Address subdivided into network, subnet, and host
  - E.g., 128.148.32.110
- Broadcast addresses
  - E.g., 128.148.32.255
- Private networks
  - not routed outside of a LAN
  - 10.0.0.0/8
  - 172.16.0.0/12
  - 192.168.0.0/16

- IP header includes
  - Source address
  - Destination address
  - Packet length (up to 64KB)
  - Time to live (up to 255)
  - IP protocol version
  - Fragmentation information
  - Transport layer protocol information (e.g., TCP)



IP ADDRESS SPACE AND ICANN

- Hosts on the internet must have unique IP addresses
- Internet Corporation for Assigned Names and Numbers
  - International nonprofit organization
  - Incorporated in the US
  - Allocates IP address space
  - Manages top-level domains
- Historical bias in favor of US corporations and nonprofit organizations

- Examples

003/8 May 94 General Electric
009/8 Aug 92 IBM
012/8 Jun 95 AT&T Bell Labs
013/8 Sep 91 Xerox Corporation
015/8 Jul 94 Hewlett-Packard
017/8 Jul 92 Apple Computer
018/8 Jan 94 MIT
019/8 May 95 Ford Motor
040/8 Jun 94 Eli Lily
043/8 Jan 91 Japan Inet
044/8 Jul 92 Amateur Radio Digital
047/8 Jan 91 Bell-Northern Res.
048/8 May 95 Prudential Securities
054/8 Mar 92 Merck
055/8 Apr 95 Boeing
056/8 Jun 94 U.S. Postal Service

A TYPICAL UNIVERSITY'S IP SPACE

- Most universities separate their network connecting dorms and the network connecting offices and academic buildings
- Dorms
  - Class B network 138.16.0.0/16 (64K addresses)
- Academic buildings and offices
  - Class B network 128.148.0.0/16 (64K addresses)
- CS department
  - Several class C (/24) networks, each with 254 addresses

IP ROUTING

- A router bridges two or more networks
  - Operates at the network layer
  - Maintains tables to forward packets to the appropriate network
  - Forwarding decisions based solely on the destination address
- Routing table
  - Maps ranges of addresses to LANs or other gateway routers

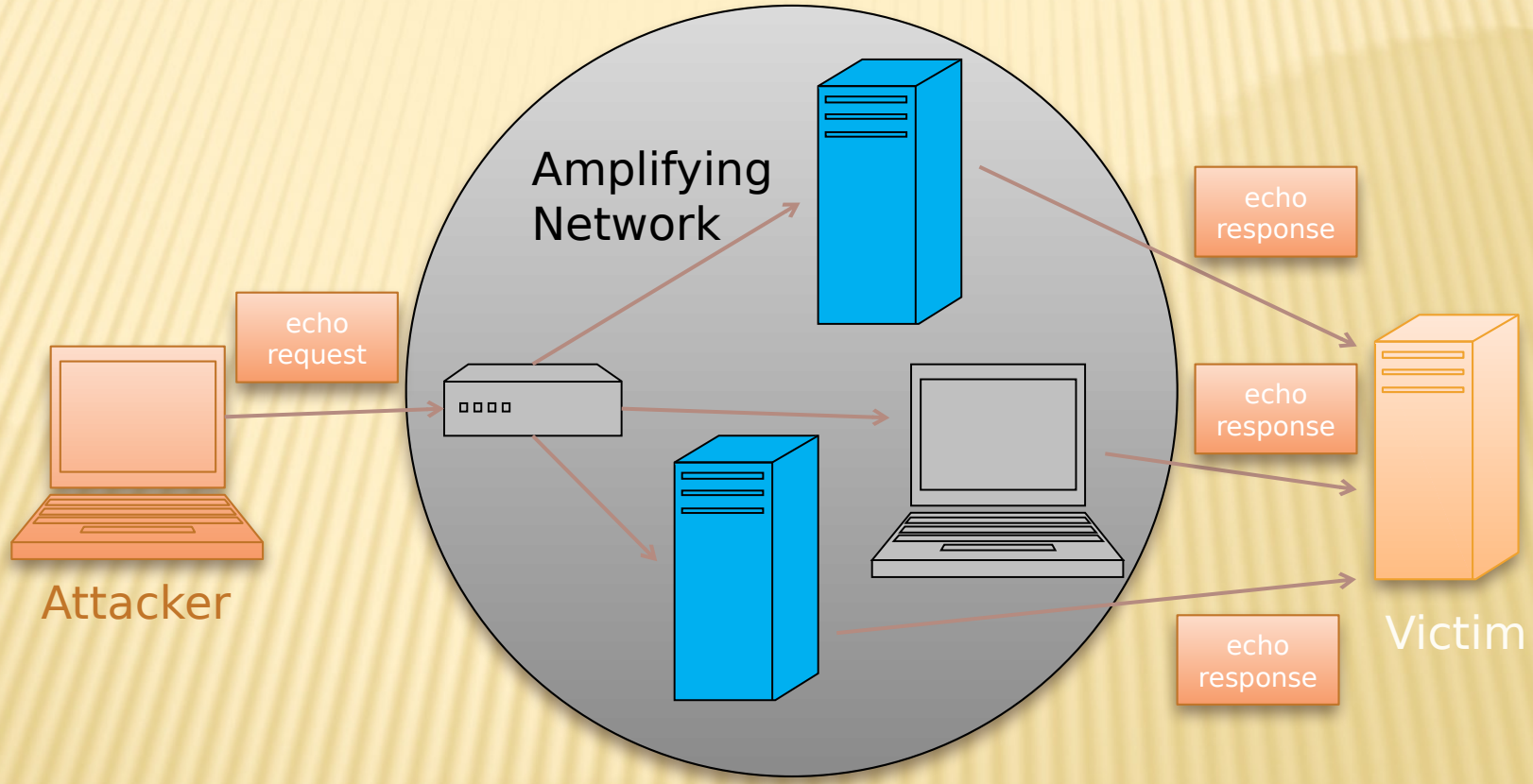
INTERNET CONTROL MESSAGE PROTOCOL (ICMP)

- Internet Control Message Protocol (ICMP)
  - Used for network testing and debugging
  - Simple messages encapsulated in single IP packets
  - Considered a network layer protocol
- Tools based on ICMP
  - Ping: sends series of echo request messages and provides statistics on roundtrip times and packet loss
  - Traceroute: sends series ICMP packets with increasing TTL value to discover routes

ICMP ATTACKS

- Ping of death
  - ICMP specifies messages must fit a single IP packet (64KB)
  - Send a ping packet that exceeds maximum size using IP fragmentation
  - Reassembled packet caused several operating systems to crash due to a buffer overflow
- Smurf
  - Ping a broadcast address using a spoofed source address

SMURF ATTACK



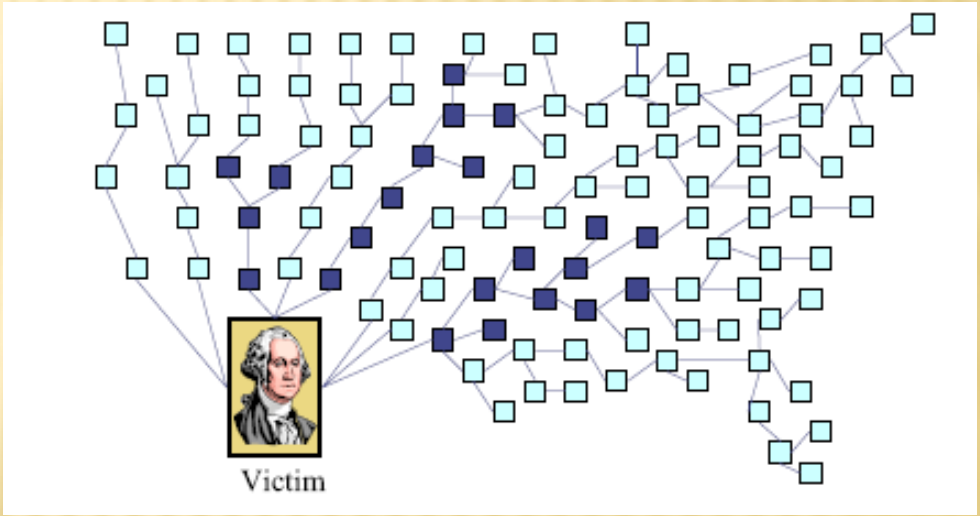
IP VULNERABILITIES

- Unencrypted transmission
  - Eavesdropping possible at any intermediate host during routing
- No source authentication
  - Sender can spoof source address, making it difficult to trace packet back to attacker
- No integrity checking
  - Entire packet, header and payload, can be modified while en route to destination, enabling content forgeries, redirections, and man-in-the-middle attacks
- No bandwidth constraints
  - Large number of packets can be injected into network to launch a denial-of-service attack
  - Broadcast addresses provide additional leverage

DENIAL OF SERVICE ATTACK

- Send large number of packets to host providing service
  - Slows down or crashes host
  - Often executed by botnet
- Attack propagation
  - Starts at zombies
  - Travels through tree of internet routers rooted
  - Ends at victim
- IP source spoofing
  - Hides attacker
  - Scatters return traffic from victim

Source:
M.T. Goodrich, Probabalistic Packet Marking for Large-Scale IP Traceback, IEEE/ACM Transactions on Networking 16:1, 2008.



IP TRACEBACK

- Problem
  - How to identify leaves of DoS propagation tree
  - Routers next to attacker
- Issues
  - There are more than 2M internet routers
  - Attacker can spoof source address
  - Attacker knows that traceback is being performed
- Approaches
  - Filtering and tracing (immediate reaction)
  - Messaging (additional traffic)
  - Logging (additional storage)
  - Probabilistic marking

PROBABILISTIC PACKET MARKING

- Method
  - Random injection of information into packet header
  - Changes seldom used bits
  - Forward routing information to victim
  - Redundancy to survive packet losses
- Benefits
  - No additional traffic
  - No router storage
  - No packet size increase
  - Can be performed online or offline

TRANSMISSION CONTROL PROTOCOL

- TCP is a transport layer protocol guaranteeing reliable data transfer, in-order delivery of messages and the ability to distinguish data for multiple concurrent applications on the same host
- Most popular application protocols, including WWW, FTP and SSH are built on top of TCP
- TCP takes a stream of 8-bit byte data, packages it into appropriately sized segment and calls on IP to transmit these packets
- Delivery order is maintained by marking each packet with a sequence number
- Every time TCP receives a packet, it sends out an ACK to indicate successful receipt of the packet.
- TCP generally checks data transmitted by comparing a checksum of the data with a checksum encoded in the packet

PORTS

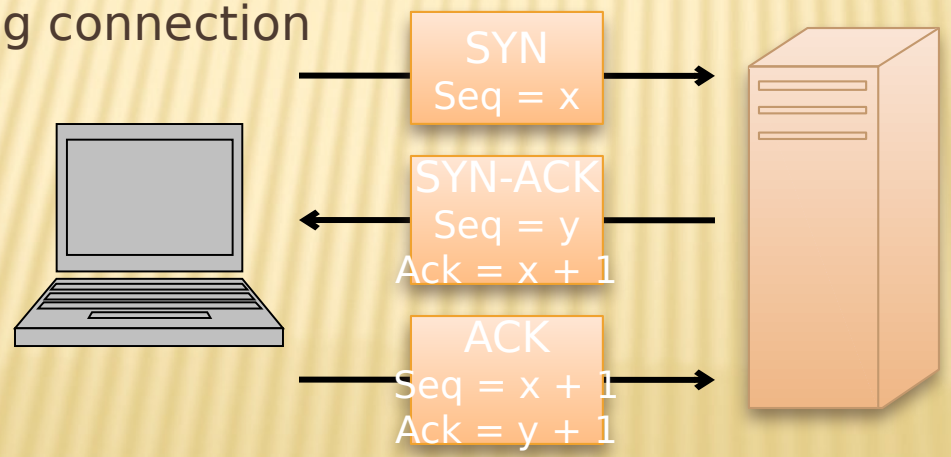
- TCP supports multiple concurrent applications on the same server
- Accomplishes this by having ports, 16 bit numbers identifying where data is directed
- The TCP header includes space for both a source and a destination port, thus allowing TCP to route all data
- In most cases, both TCP and UDP use the same port numbers for the same applications
- Ports 0 through 1023 are reserved for use by known protocols.
- Ports 1024 through 49151 are known as user ports, and should be used by most user programs for listening to connections and the like
- Ports 49152 through 65535 are private ports used for dynamic allocation by socket libraries

TCP PACKET FORMAT

Bit Offset	0-3	4-7	8-15	16-18	19-31
0	**Source Port**			**Destination Port**	
32	**Sequence Number**				
64	**Acknowledgment Number**				
96	Offset	Reserved	Flags	**Window Size**	
128	Checksum			Urgent Pointer	
160	Options				
>= 160	Payload				

ESTABLISHING TCP CONNECTIONS

- TCP connections are established through a three way handshake.
- The server generally has a passive listener, waiting for a connection request
- The client requests a connection by sending out a SYN packet
- The server responds by sending a SYN/ACK packet, indicating an acknowledgment for the connection
- The client responds by sending an ACK to the server thus establishing connection



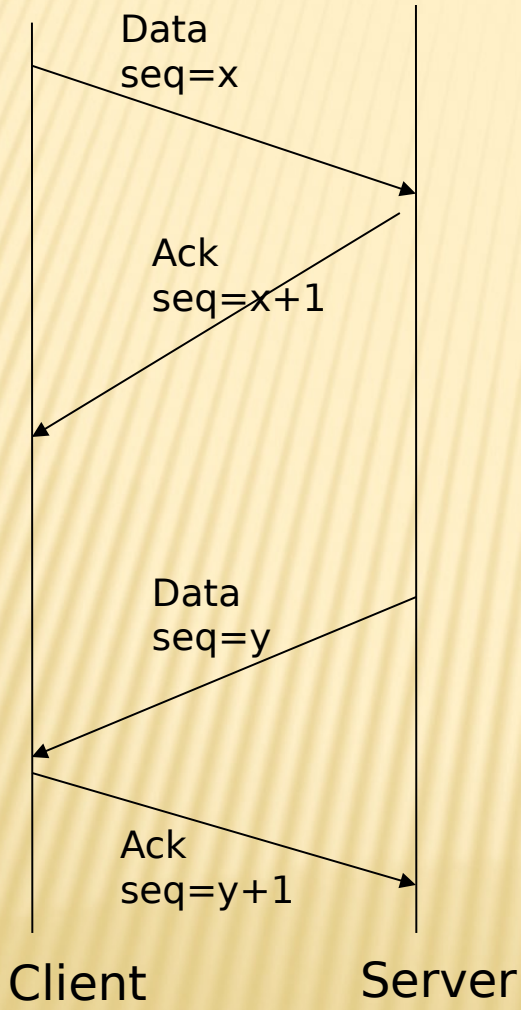
SYN FLOOD

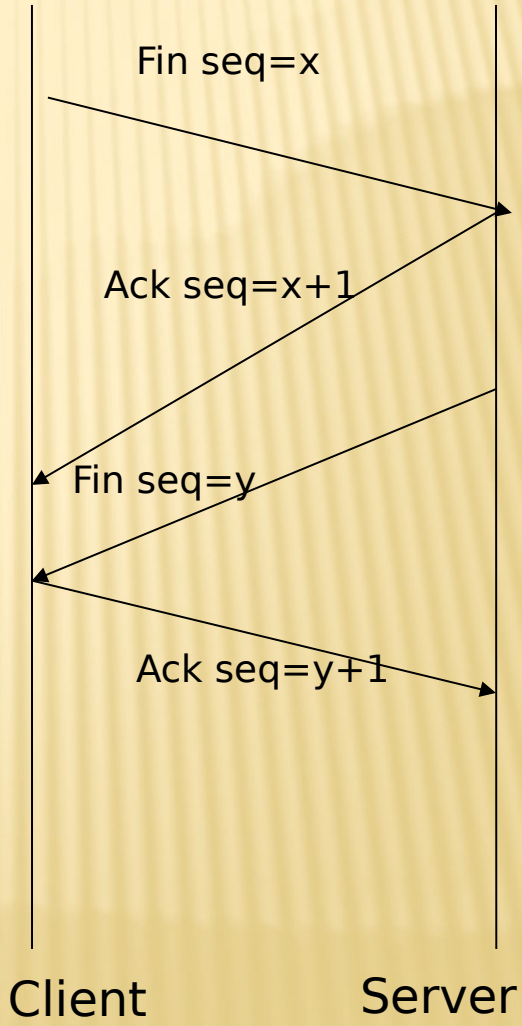
- Typically DOS attack, though can be combined with other attack such as TCP hijacking
- Rely on sending TCP connection requests faster than the server can process them
- Attacker creates a large number of packets with spoofed source addresses and setting the SYN flag on these
- The server responds with a SYN/ACK for which it never gets a response (waits for about 3 minutes each)
- Eventually the server stops accepting connection requests, thus triggering a denial of service.
- Can be solved in multiple ways
- One of the common way to do this is to use SYN cookies

TCP DATA TRANSFER

- During connection initialization using the three way handshake, initial sequence numbers are exchanged
- The TCP header includes a 16 bit checksum of the data and parts of the header, including the source and destination
- Acknowledgment or lack thereof is used by TCP to keep track of network congestion and control flow and such
- TCP connections are cleanly terminated with a 4-way handshake
  - The client which wishes to terminate the connection sends a FIN message to the other client
  - The other client responds by sending an ACK
  - The other client sends a FIN
  - The original client now sends an ACK, and the connection is terminated

TCP DATA TRANSFER AND TEARDOWN





TCP CONGESTION CONTROL

- During the mid-80s it was discovered that uncontrolled TCP messages were causing large scale network congestion
- TCP responded to congestion by retransmitting lost packets, thus making the problem was worse
- What is predominantly used today is a system where ACKs are used to determine the maximum number of packets which should be sent out
- Most TCP congestion avoidance algorithms, avoid congestion by modifying a congestion window (cwnd) as more cumulative ACKs are received
- Lost packets are taken to be a sign of network congestion
- TCP begins with an extremely low cwnd and rapidly increases the value of this variable to reach bottleneck capacity
- At this point it shifts to a collision detection algorithm which slowly probes the network for additional bandwidth
- TCP congestion control is a good idea in general but allows for certain attacks.

OPTIMISTIC ACK ATTACK

- An optimistic ACK attack takes advantage of the TCP congestion control
- It begins with a client sending out ACKs for data segments it hasn’t yet received
- This flood of optimistic ACKs makes the servers TCP stack believe that there is a large amount of bandwidth available and thus increase cwnd
- This leads to the attacker providing more optimistic ACKs, and eventually bandwidth use beyond what the server has available
- This can also be played out across multiple servers, with enough congestion that a certain section of the network is no longer reachable
- There are no practical solutions to this problem

SESSION HIJACKING

- Also commonly known as TCP Session Hijacking
- A security attack over a protected network
- Attempt to take control of a network session
- Sessions are server keeping state of a client’s connection
- Servers need to keep track of messages sent between client and the server and their respective actions
- Most networks follow the TCP/IP protocol
- IP Spoofing is one type of hijacking on large network

IP SPOOFING

- IP Spoofing is an attempt by an intruder to send packets from one IP address that appear to originate at another
- If the server thinks it is receiving messages from the real source after authenticating a session, it could inadvertently behave maliciously
- There are two basic forms of IP Spoofing
  - Blind Spoofing
    - Attack from any source
  - Non-Blind Spoofing
    - Attack from the same subnet

BLIND IP SPOOFING

- The TCP/IP protocol requires that “acknowledgement” numbers be sent across sessions
- Makes sure that the client is getting the server’s packets and vice versa
- Need to have the right sequence of acknowledgment numbers to spoof an IP identity

NON-BLIND IP SPOOFING

- IP Spoofing without inherently knowing the acknowledgment sequence pattern
  - Done on the same subnet
  - Use a packet sniffer to analyze the sequence pattern
    - Packet sniffers intercept network packets
    - Eventually decodes and analyzes the packets sent across the network
    - Determine the acknowledgment sequence pattern from the packets
    - Send messages to server with actual client's IP address and with validly sequenced acknowledgment number

PACKET SNIFFERS

- Packet sniffers “read” information traversing a network
  - Packet sniffers intercept network packets, possibly using ARP cache poisoning
  - Can be used as legitimate tools to analyze a network
    - Monitor network usage
    - Filter network traffic
    - Analyze network problems
  - Can also be used maliciously
    - Steal information (i.e. passwords, conversations, etc.)
    - Analyze network information to prepare an attack
- Packet sniffers can be either software or hardware based
  - Sniffers are dependent on network setup

DETECTING SNIFFERS

- Sniffers are almost always passive
  - They simply collect data
  - They do not attempt “entry” to “steal” data
- This can make them extremely hard to detect
- Most detection methods require suspicion that sniffing is occurring
  - Then some sort of “ping” of the sniffer is necessary
  - It should be a broadcast that will cause a response only from a sniffer
- Another solution on switched hubs is ARP watch
  - An ARP watch monitors the ARP cache for duplicate entries of a machine
  - If such duplicates appear, raise an alarm
  - Problem: false alarms
    - Specifically, DHCP networks can have multiple entires for a single machine

STOPPING PACKET SNIFFING

- The best way is to encrypt packets securely
  - Sniffers can capture the packets, but they are meaningless
    - Capturing a packet is useless if it just reads as garbage
  - SSH is also a much more secure method of connection
    - Private/Public key pairs makes sniffing virtually useless
  - On switched networks, almost all attacks will be via ARP spoofing
    - Add machines to a permanent store in the cache
    - This store cannot be modified via a broadcast reply
    - Thus, a sniffer cannot redirect an address to itself
- The best security is to not let them in in the first place
  - Sniffers need to be on your subnet in a switched hub in the first place
  - All sniffers need to somehow access root at some point to start themselves up

PORT KNOCKING

- Broadly port knocking is the act of attempting to make connections to blocked ports in a certain order in an attempt to open a port
- Port knocking is fairly secure against brute force attacks since there are 65536^k combinations, where k is the number of ports knocked
- Port knocking however if very susceptible to replay attacks. Someone can theoretically record port knocking attempts and repeat those to get the same open port again
- One good way of protecting against replay attacks would be a time dependent knock sequence.

USER DATAGRAM PROTOCOL

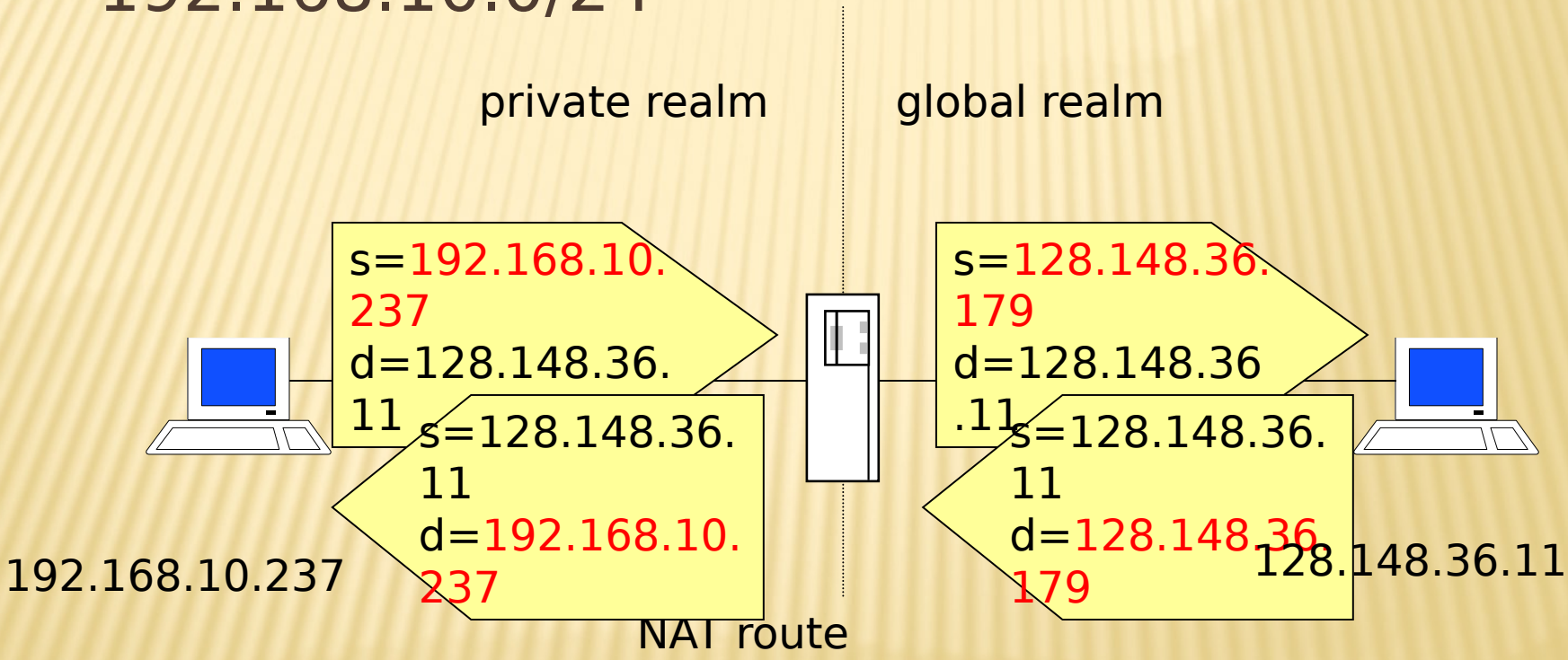
- UDP is a stateless, unreliable datagram protocol built on top of IP, that is it lies on level 4
- It does not provide delivery guarantees, or acknowledgments, but is significantly faster
- Can however distinguish data for multiple concurrent applications on a single host.
- A lack of reliability implies applications using UDP must be ready to accept a fair amount of error packages and data loss. Some application level protocols such as TFTP build reliability on top of UDP.
  - Most applications used on UDP will suffer if they have reliability. VoIP, Streaming Video and Streaming Audio all use UDP.
- UDP does not come with built in congestion protection, so while UDP does not suffer from the problems associated with optimistic ACK, there are cases where high rate UDP network access will cause congestion.

NETWORK ADDRESS TRANSLATION

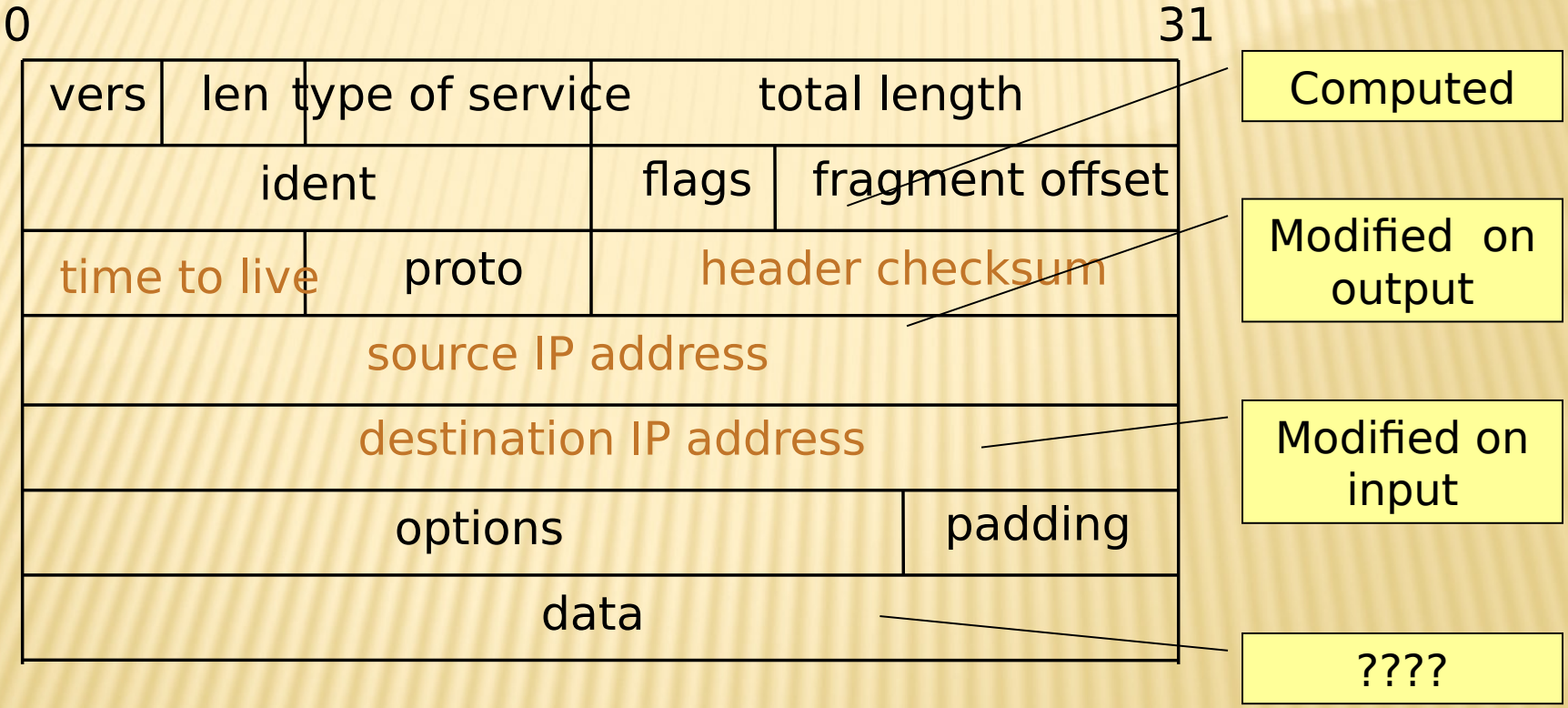
- Introduced in the early 90s to alleviate IPv4 address space congestion
- Relies on translating addresses in an internal network, to an external address that is used for communication to and from the outside world
- NAT is usually implemented by placing a router in between the internal private network and the public network.
- Saves IP address space since not every terminal needs a globally unique IP address, only an organizationally unique one
- While NAT should really be transparent to all high level services, this is sadly not true because a lot of high level communication uses things on IP

TRANSLATION

- Router has a pool of private addresses 192.168.10.0/24



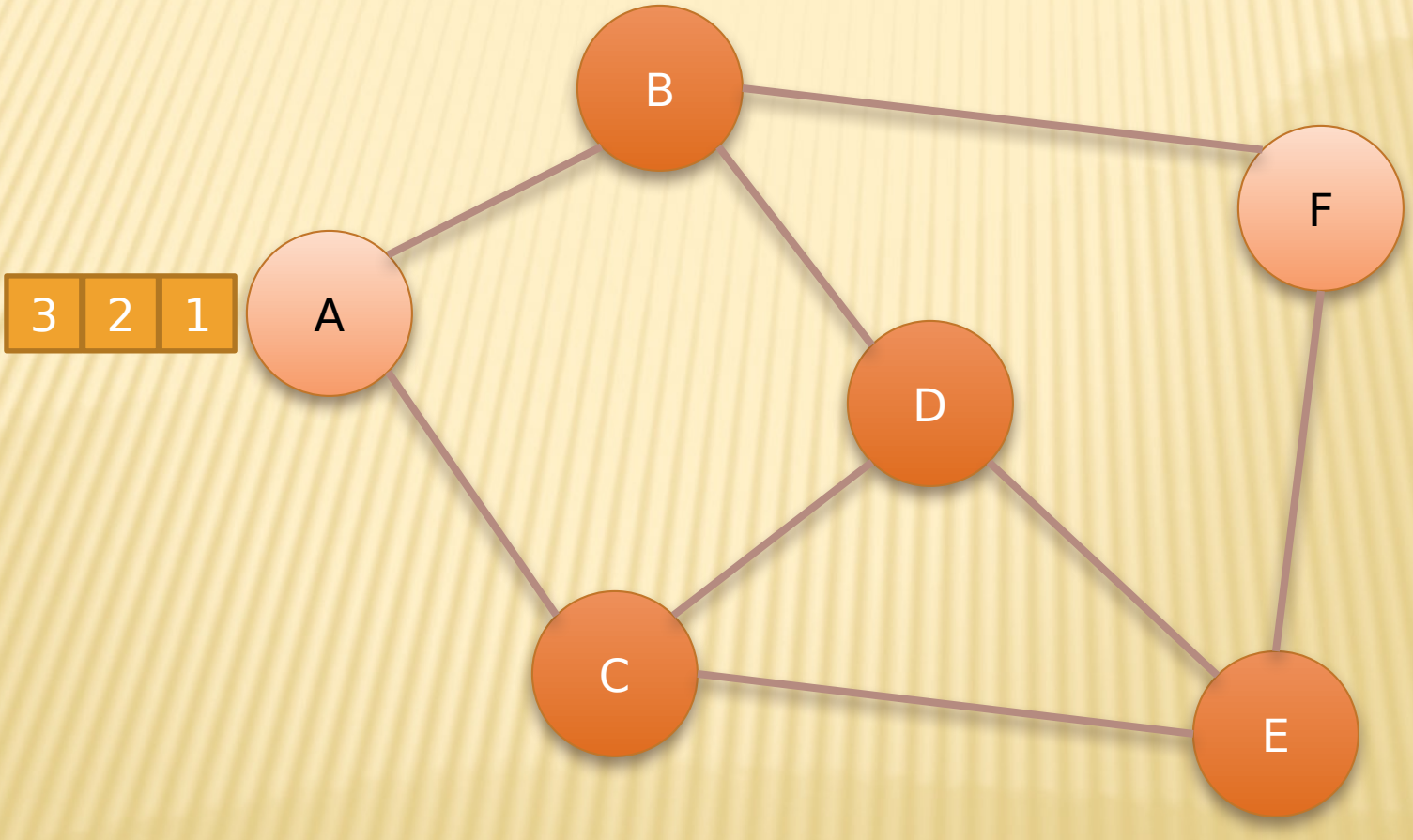
IP PACKET MODIFICATIONS



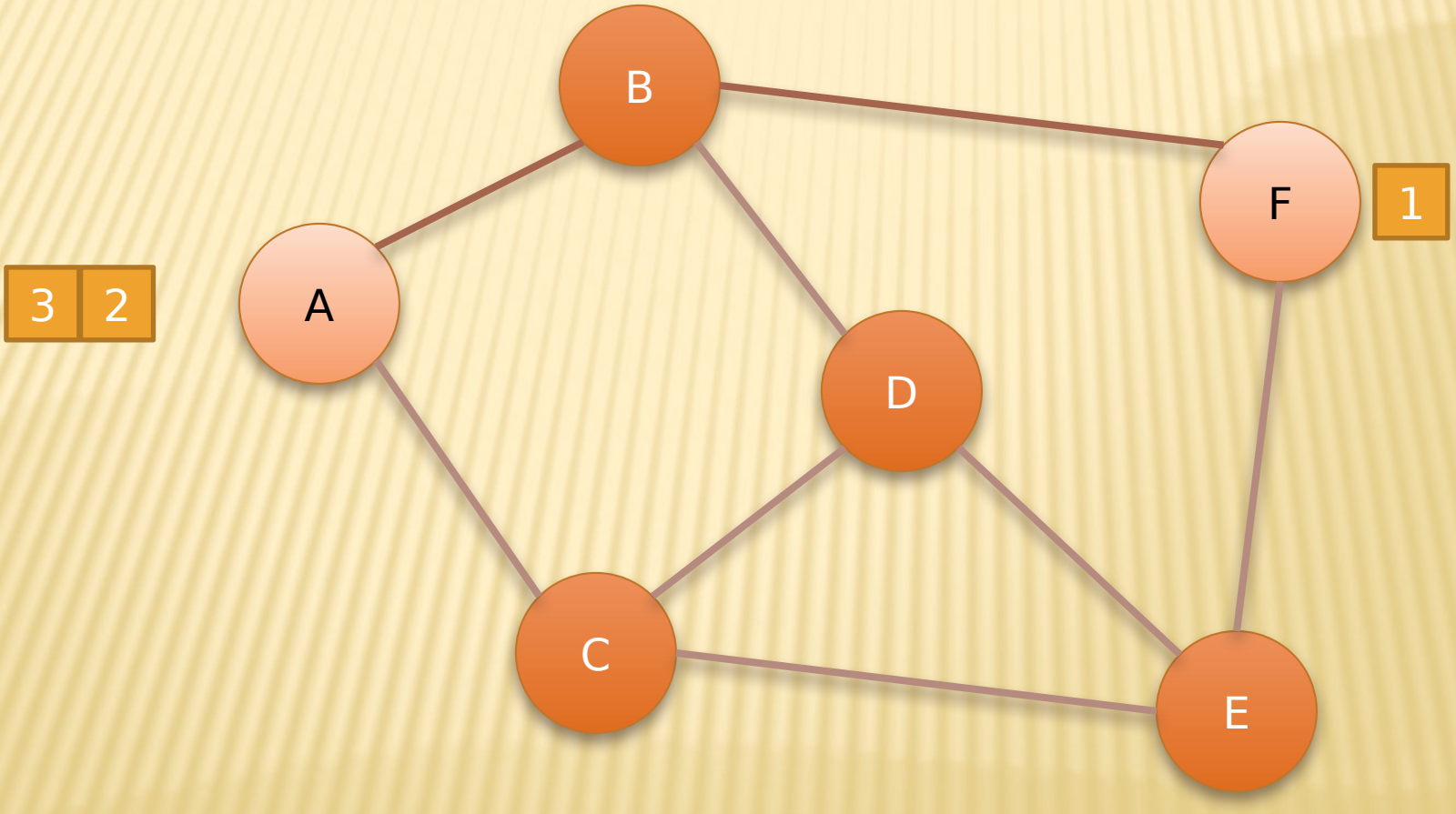
COMPUTER NETWORKS

- Circuit switching
  - Legacy phone network
  - Single route through sequence of hardware devices established when two nodes start communication
  - Data sent along route
  - Route maintained until communication ends
- Packet switching
  - Internet
  - Data split into packets
  - Packets transported independently through network
  - Each packet handled on a best efforts basis
  - Packets may follow different routes

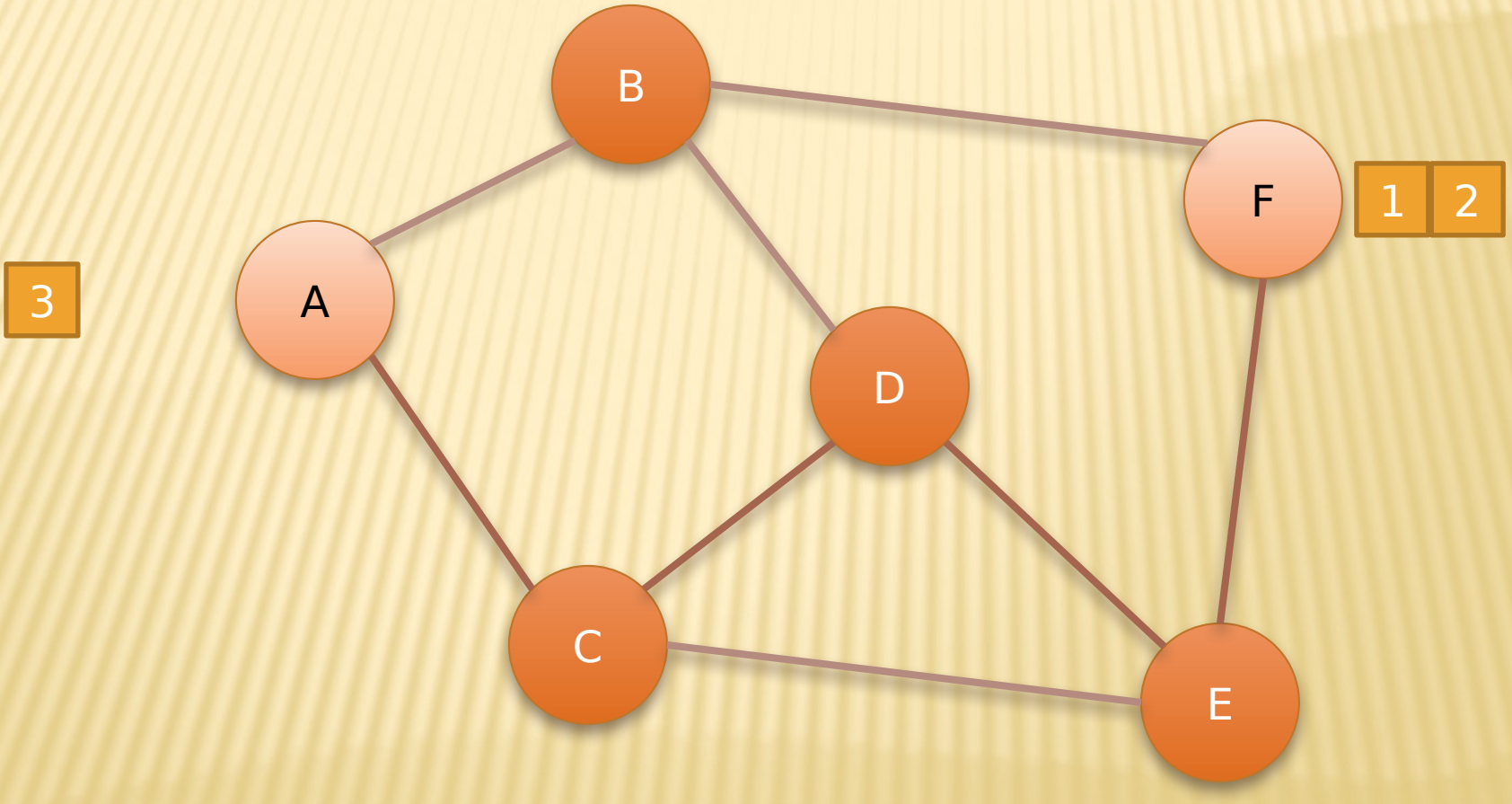
PACKET SWITCHING



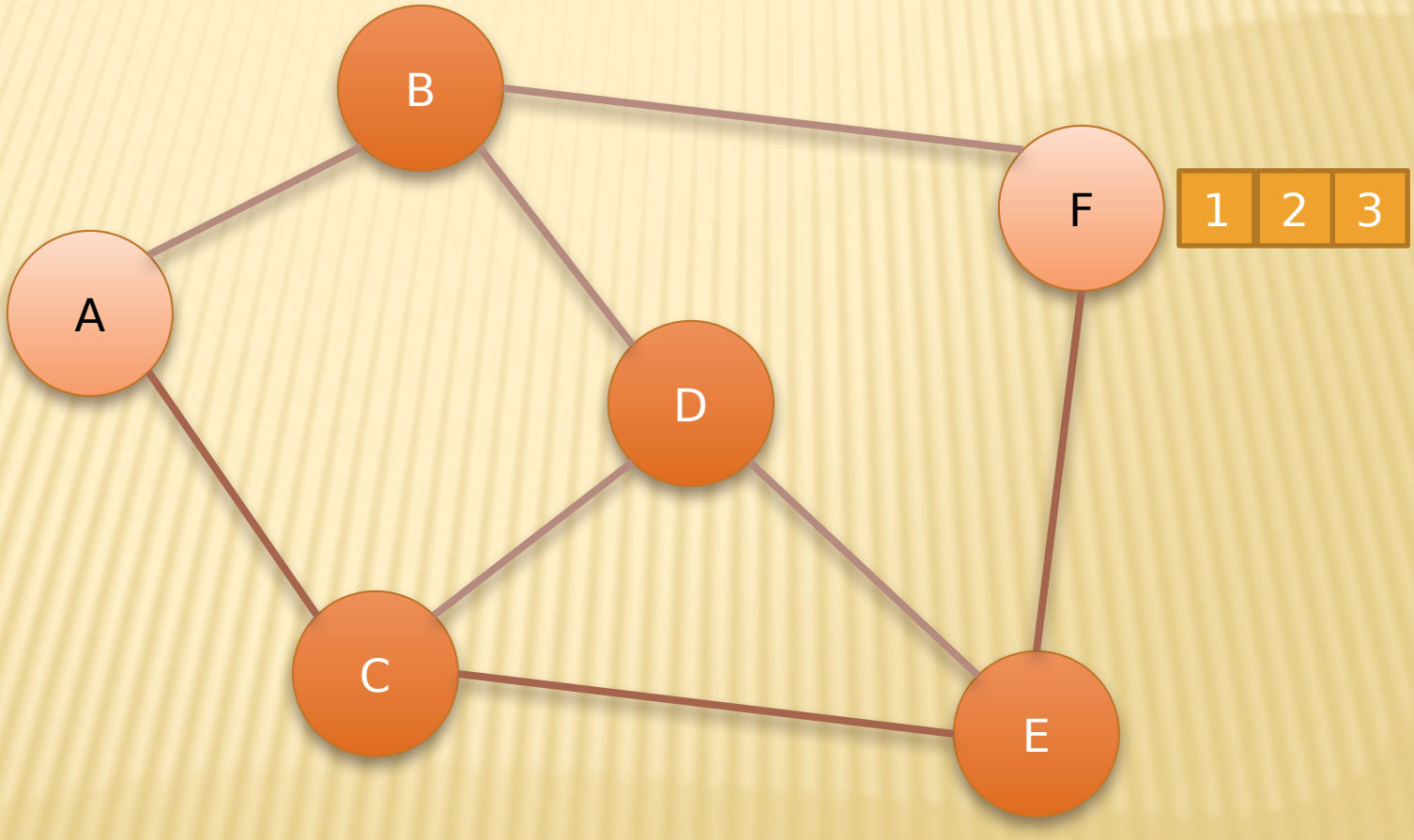
PACKET SWITCHING



PACKET SWITCHING



PACKET SWITCHING

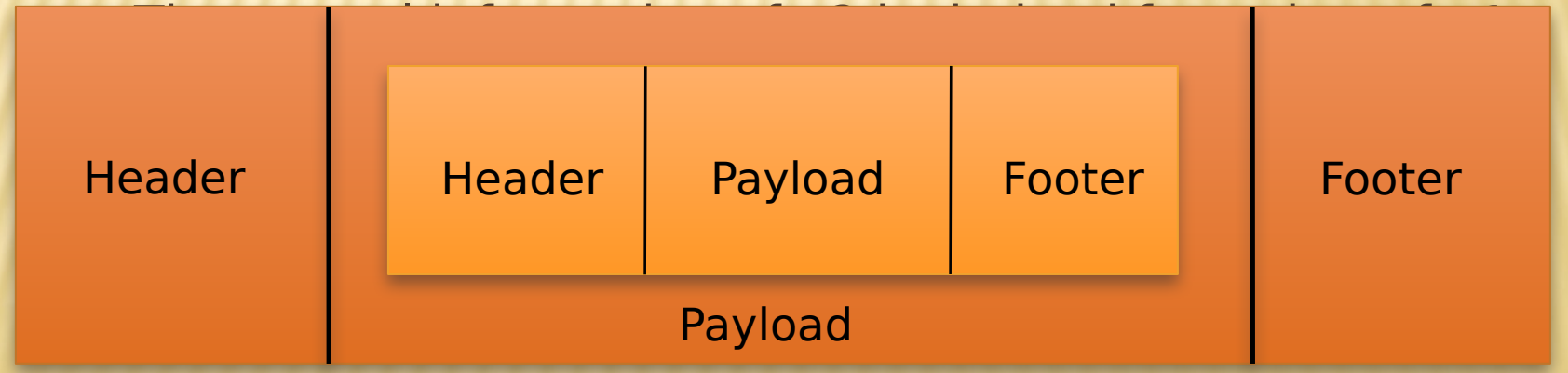


PROTOCOLS

- A protocol defines the rules for communication between computers
- Protocols are broadly classified as connectionless and connection oriented
- Connectionless protocol
  - Sends data out as soon as there is enough data to be transmitted
  - E.g., user datagram protocol (UDP)
- Connection-oriented protocol
  - Provides a reliable connection stream between two nodes
  - Consists of set up, transmission, and tear down phases
  - Creates virtual circuit-switched network
  - E.g., transmission control protocol (TCP)

ENCAPSULATION

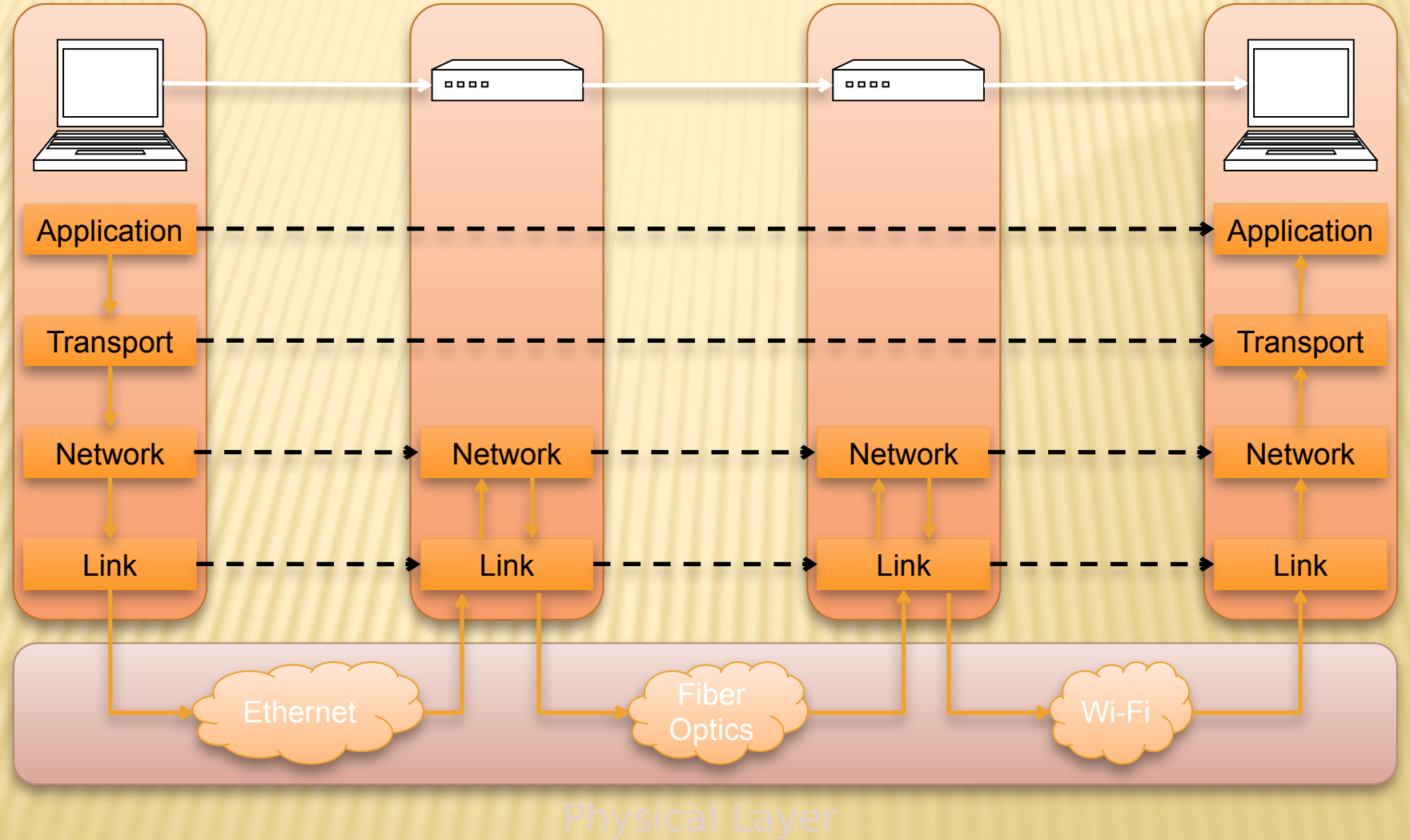
- A packet typically consists of
  - Control information for addressing the packet: header and footer
  - Data: payload
- A network protocol N1 can use the services of another network protocol N2
  - A packet p1 of N1 is encapsulated into a packet p2 of N2
  - The payload of p2 is p1



NETWORK LAYERS

- Network models typically use a stack of layers
  - Higher layers use the services of lower layers via encapsulation
  - A layer can be implemented in hardware or software
  - The bottommost layer must be in hardware
- A network device may implement several layers
- A communication channel between two nodes is established for each layer
  - Actual channel at the bottom layer
  - Virtual channel at higher layers

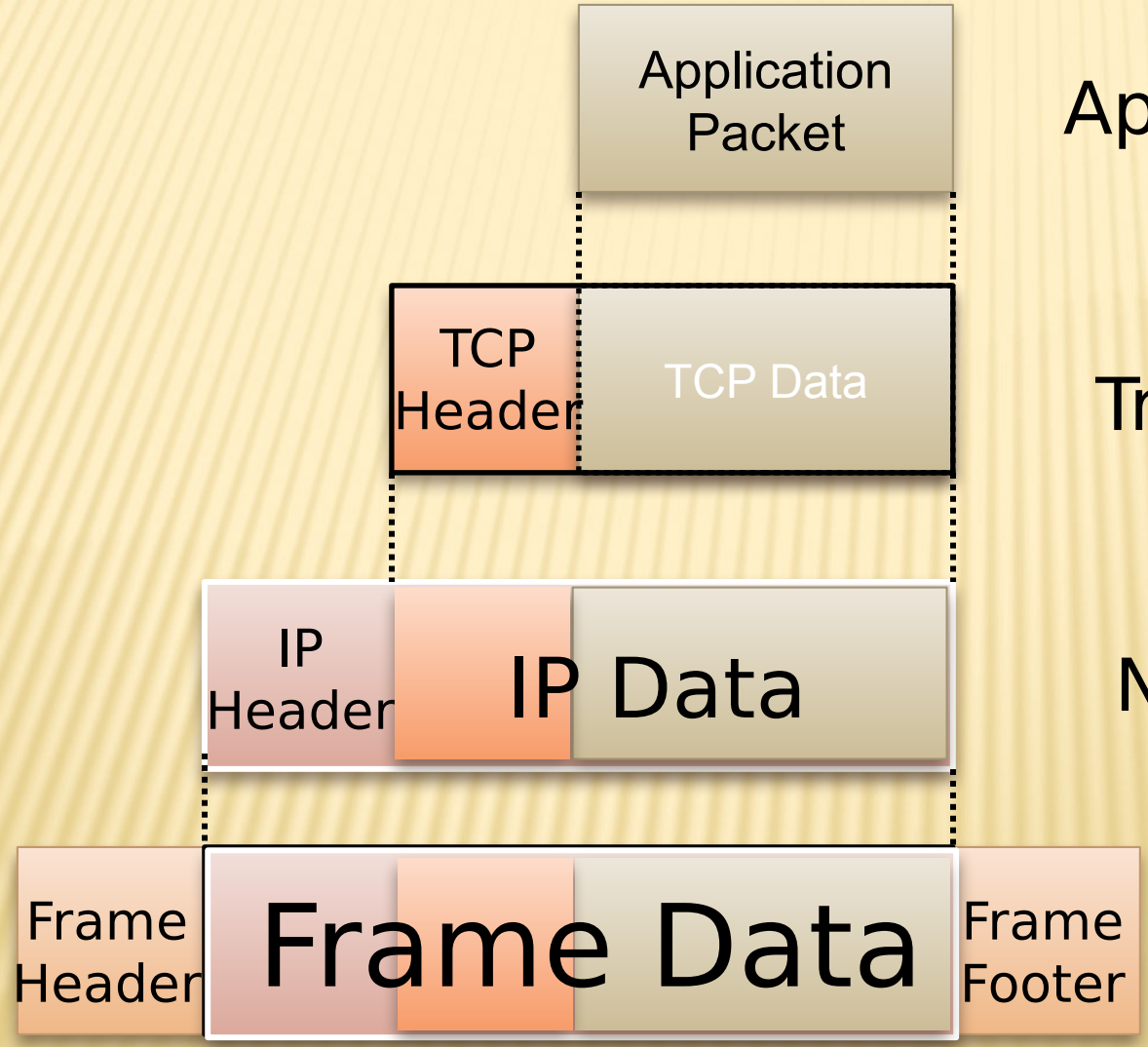
INTERNET LAYERS



INTERMEDIATE LAYERS

- Link layer
  - Local area network: Ethernet, WiFi, optical fiber
  - 48-bit media access control (MAC) addresses
  - Packets called frames
- Network layer
  - Internet-wide communication
  - Best efforts
  - 32-bit internet protocol (IP) addresses in IPv4
  - 128-bit IP addresses in IPv6
- Transport layer
  - 16-bit addresses (ports) for classes of applications
  - Connection-oriented transmission layer protocol (TCP)
  - Connectionless user datagram protocol (UDP)

INTERNET PACKET ENCAPSULATION



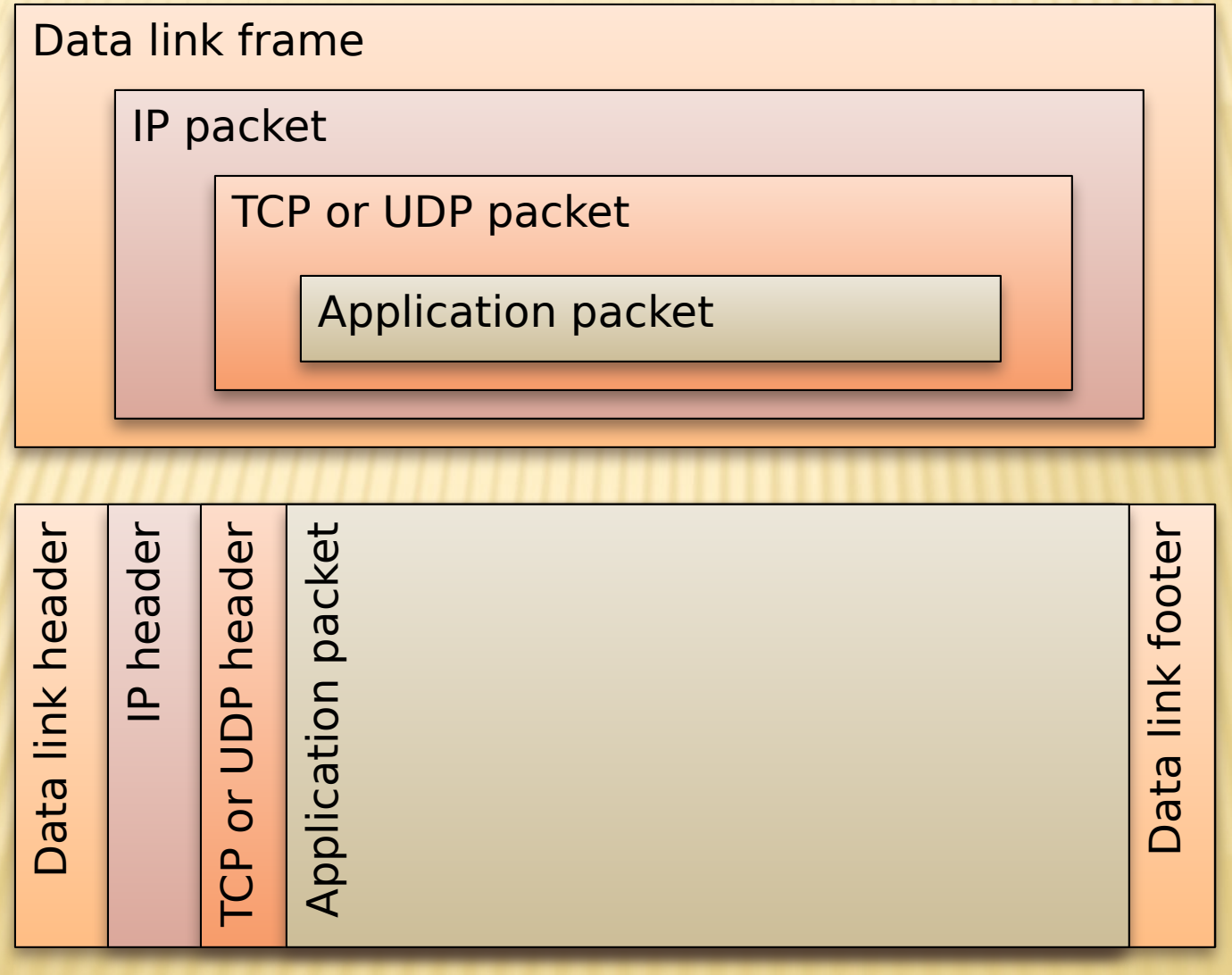
Application Layer

Transport Layer

Network Layer

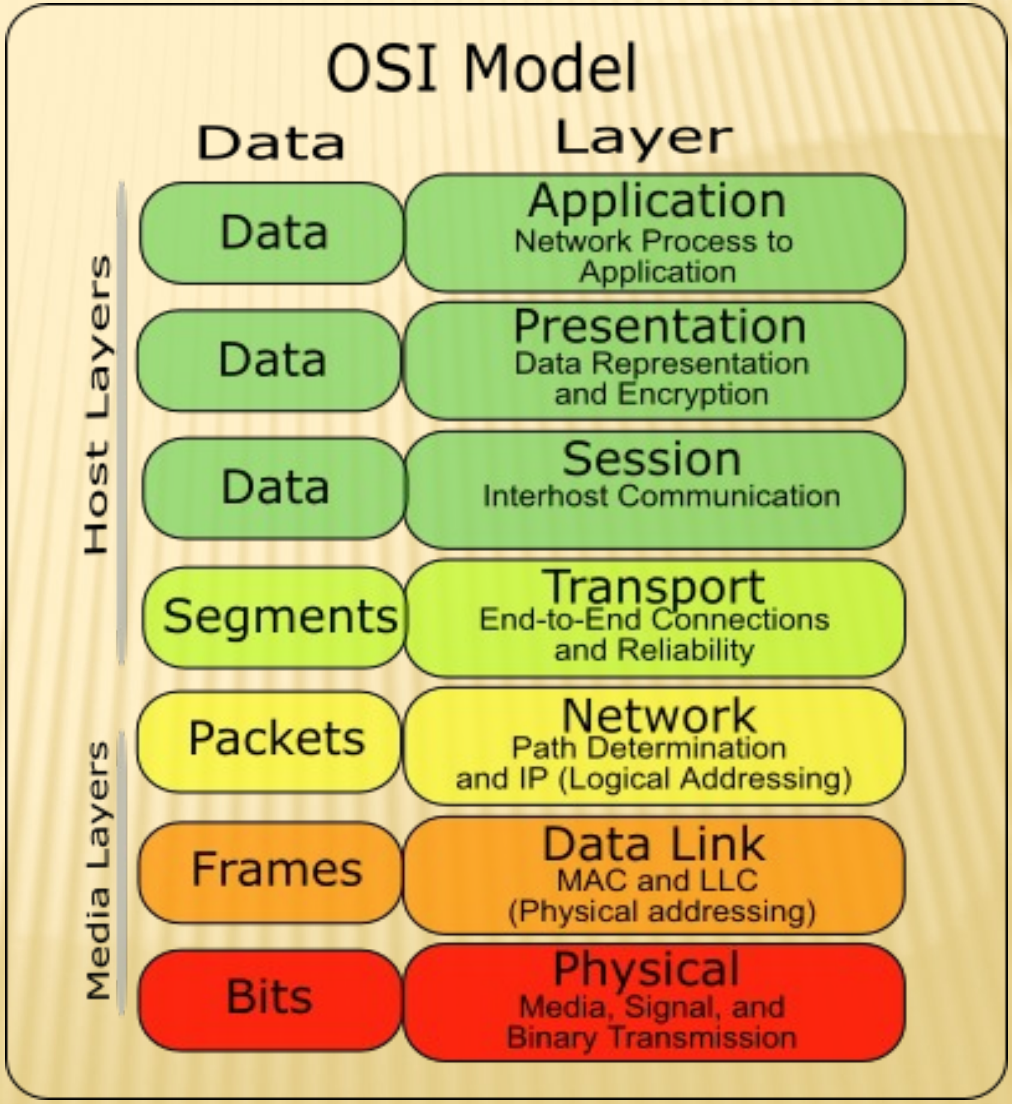
Link Layer

INTERNET PACKET ENCAPSULATION



THE OSI MODEL

- The OSI (Open System Interconnect) Reference Model is a network model consisting of seven layers
- Created in 1983, OSI is promoted by the International Standard Organization (ISO)



NETWORK INTERFACES

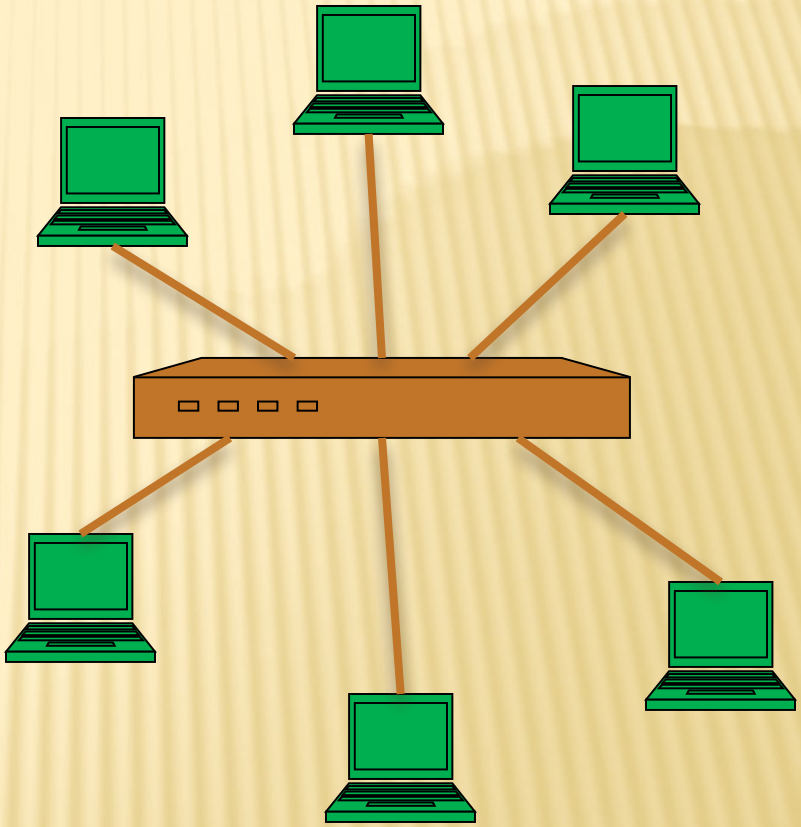
- Network interface: device connecting a computer to a network
  - Ethernet card
  - WiFi adapter
- A computer may have multiple network interfaces
- Packets transmitted between network interfaces
- Most local area networks, (including Ethernet and WiFi) broadcast frames
- In regular mode, each network interface gets the frames intended for it
- Traffic sniffing can be accomplished by configuring the network interface to read all frames (promiscuous mode)

MAC ADDRESSES

- Most network interfaces come with a predefined MAC address
- A MAC address is a 48-bit number usually represented in hex
  - E.g., 00-1A-92-D4-BF-86
- The first three octets of any MAC address are IEEE-assigned Organizationally Unique Identifiers
  - E.g., Cisco 00-1A-A1, D-Link 00-1B-11, ASUSTek 00-1A-92
- The next three can be assigned by organizations as they please, with uniqueness being the only constraint
- Organizations can utilize MAC addresses to identify computers on their network
- MAC address can be reconfigured by network interface driver software

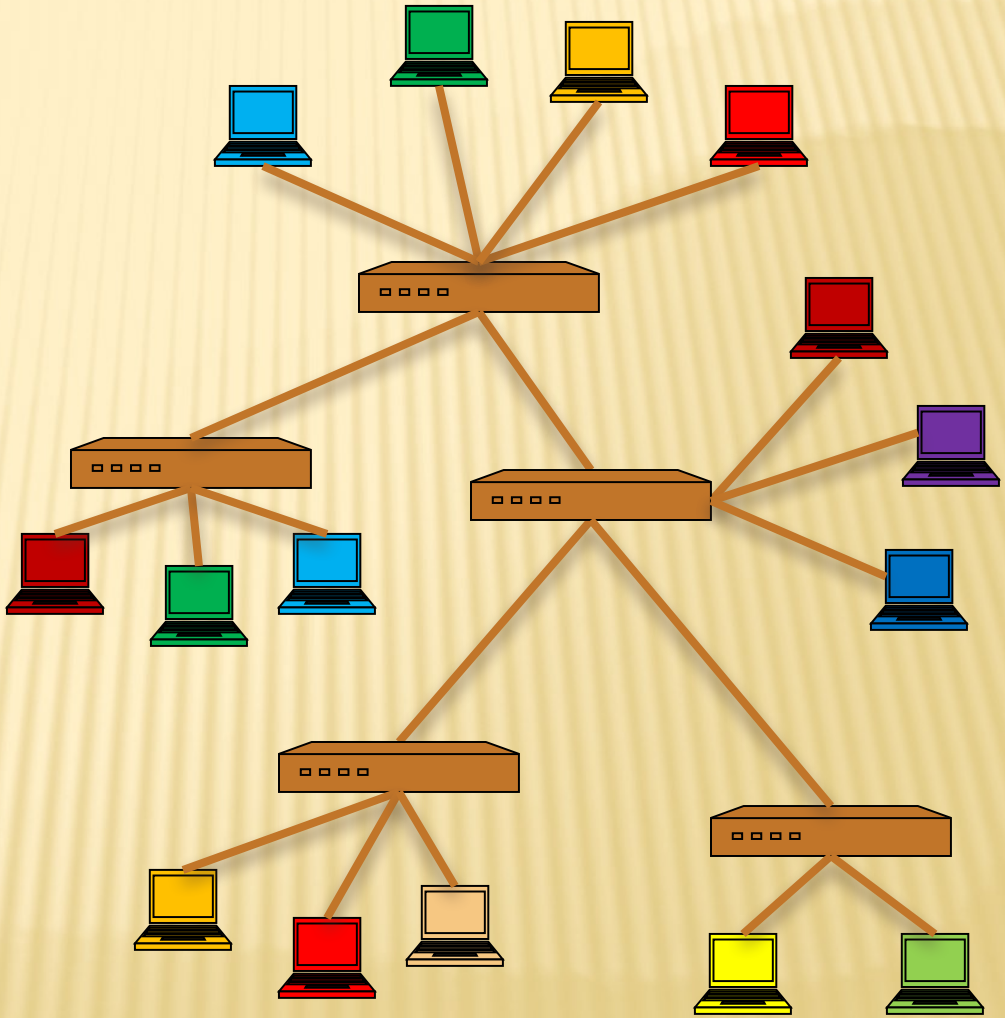
SWITCH

- A switch is a common network device
  - Operates at the link layer
  - Has multiple ports, each connected to a computer
- Operation of a switch
  - Learn the MAC address of each computer connected to it
  - Forward frames only to the destination computer



COMBINING SWITCHES

- Switches can be arranged into a tree
- Each port learns the MAC addresses of the machines in the segment (subtree) connected to it
- Fragments to unknown MAC addresses are broadcast
- Frames to MAC addresses in the same segment as the sender are ignored



MAC ADDRESS FILTERING

- A switch can be configured to provide service only to machines with specific MAC addresses
- Allowed MAC addresses need to be registered with a network administrator
- A MAC spoofing attack impersonates another machine
  - Find out MAC address of target machine
  - Reconfigure MAC address of rogue machine
  - Turn off or unplug target machine
- Countermeasures
  - Block port of switch when machine is turned off or unplugged
  - Disable duplicate MAC addresses

VIEWING AND CHANGING MAC ADDRESSES

- Viewing the MAC addresses of the interfaces of a machine
  - Linux: ifconfig
  - Windows: ipconfig /all
- Changing a MAC address in Linux
  - Stop the networking service: /etc/init.d/network stop
  - Change the MAC address: ifconfig eth0 hw ether <MAC-address>
  - Start the networking service: /etc/init.d/network start
- Changing a MAC address in Windows
  - Open the Network Connections applet
  - Access the properties for the network interface
  - Click “Configure …”
  - In the advanced tab, change the network address to the desired value
- Changing a MAC address requires administrator privileges

ARP

- The address resolution protocol (ARP) connects the network layer to the data layer by converting IP addresses to MAC addresses
- ARP works by broadcasting requests and caching responses for future use
- The protocol begins with a computer broadcasting a message of the form

  who has <IP address1> tell <IP address2>

- When the machine with <IP address1> or an ARP server receives this message, its broadcasts the response

  <IP address1> is <MAC address>

- The requestor's IP address <IP address2> is contained in the link header
- The Linux and Windows command arp - a displays the ARP table

```
Internet Address      Physical Address      Type
128.148.31.1          00-00-0c-07-ac-00     dynamic
128.148.31.15         00-0c-76-b2-d7-1d     dynamic
128.148.31.71         00-0c-76-b2-d0-d2     dynamic
128.148.31.75         00-0c-76-b2-d7-1d     dynamic
128.148.31.102        00-22-0c-a3-e4-00     dynamic
128.148.31.137        00-1d-92-b6-f1-a9     dynamic
```

ARP SPOOFING

- The ARP table is updated whenever an ARP response is received
- Requests are not tracked
- ARP announcements are not authenticated
- Machines trust each other
- A rogue machine can spoof other machines

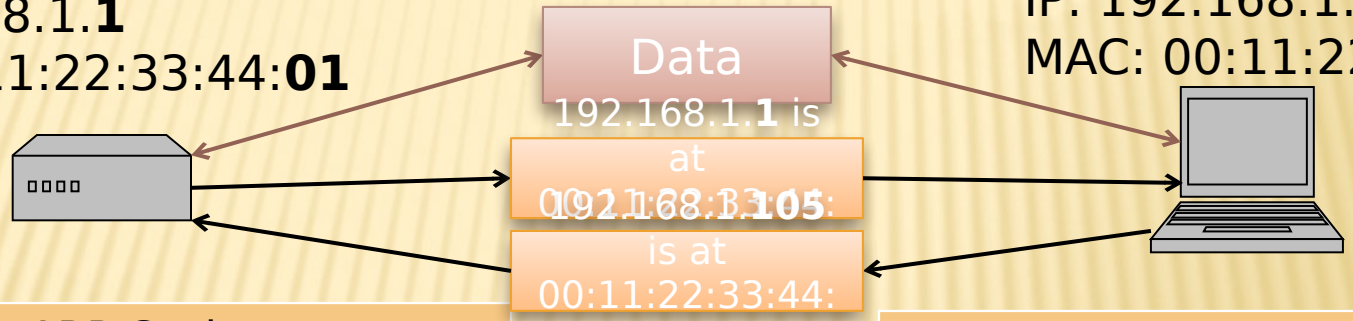
ARP POISONING (ARP SPOOFING)

- According to the standard, almost all ARP implementations are stateless
- An arp cache updates every time that it receives an arp reply… even if it did not send any arp request!
- It is possible to “poison” an arp cache by sending gratuitous arp replies
- Using static entries solves the problem but it is almost impossible to manage!

ARP CACHES

IP: 192.168.1.**1**
MAC: 00:11:22:33:44:**01**

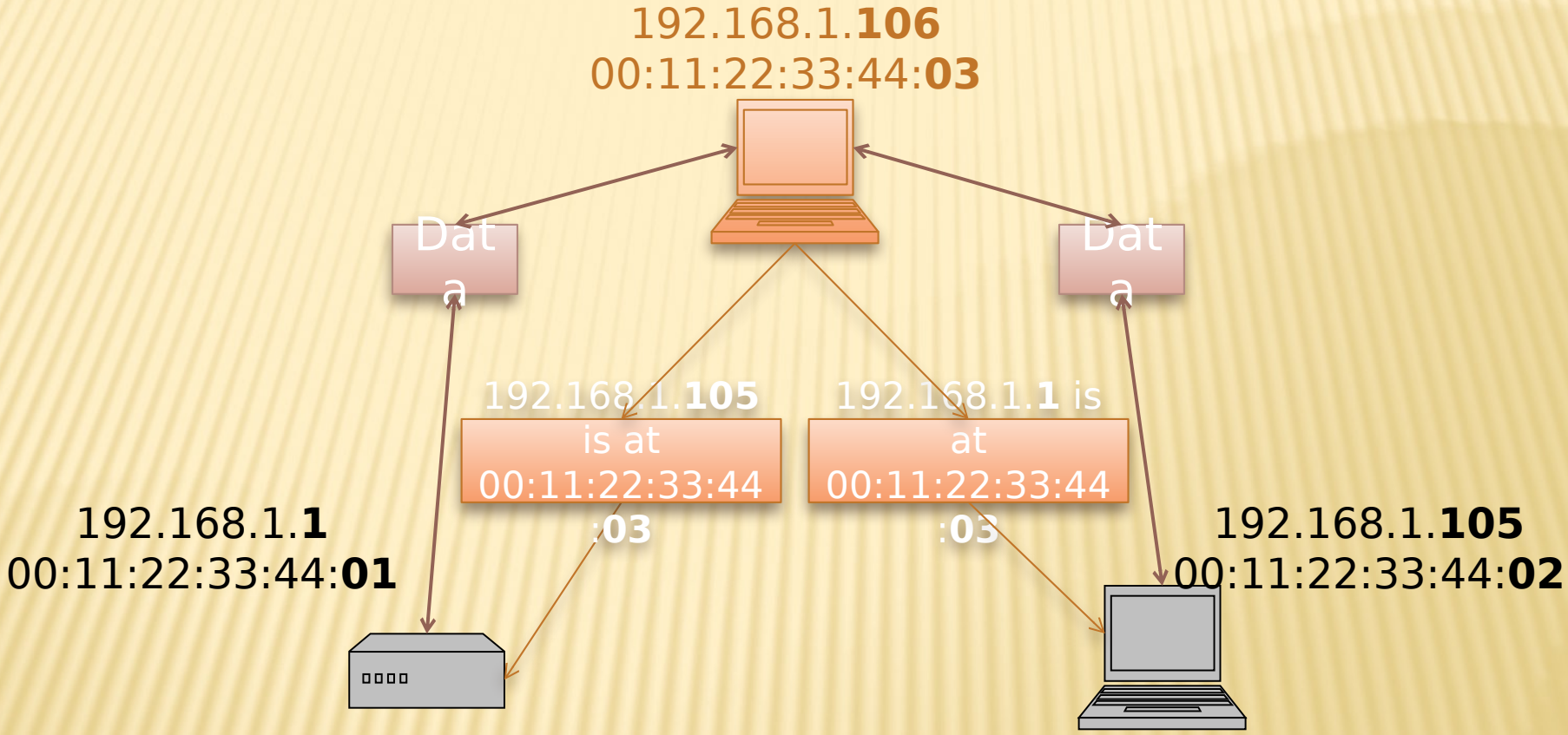
IP: 192.168.1.**105**
MAC: 00:11:22:33:44:**02**



ARP Cache	
192.168.1.**1** **05**	00:11:22:33:44:**0** **2**

ARP Cache	
192.168.1.**1**	00:11:22:33:44:**0** **1**

POISONED ARP CACHES



Poisoned ARP Cache	
192.168.1.**1** **05**	00:11:22:33:44:**0** **3**

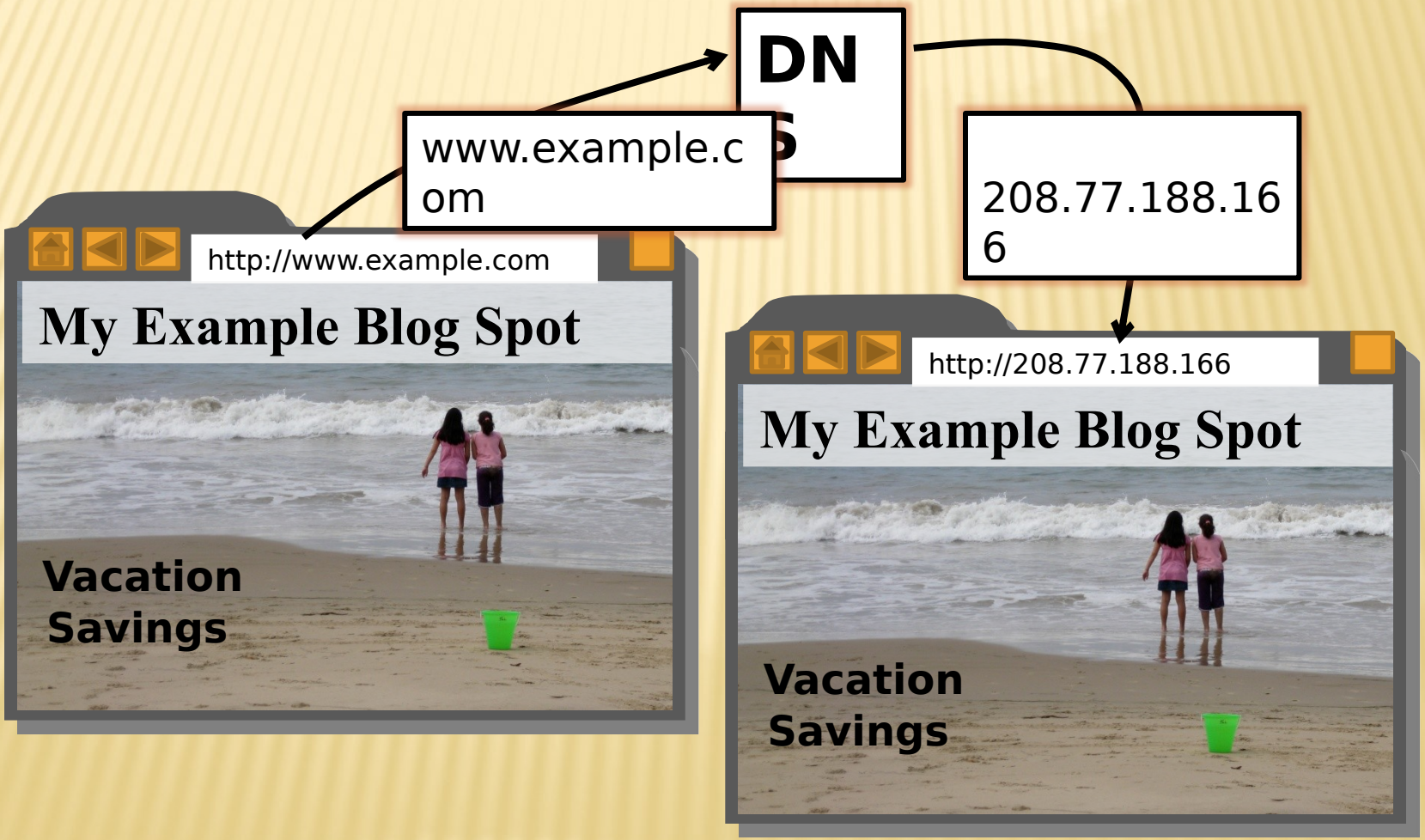
Poisoned ARP Cache	
192.168.1.**1**	00:11:22:33:44: **03**

ROOT CAUSE AND DEFENSE

- The ARP spoofing is derived from the lack of identity verification in the Internet's underlying mechanisms.
- Defense:
  - Checking for multiple occurrences of the same MAC address on the LAN.
  - Manually specify a router's ARP cache to assign certain MAC addresses to specify IP addresses. Requires to adjust the cache are ignored.

Domain Name System

- The domain name system (DNS) is an application-layer protocol for mapping domain names to IP addresses



Domain Name System

- DNS provides a distributed database over the internet that stores various resource records, including:
  - Address (A) record: IP address associated with a host name
  - Mail exchange(MX) record: mail server of a domain
  - Name server (NS) record: authoritative server for a domain

For example, if example.com wishes to sub-delegate "john.example.com." to John who works at Example, inc., lines like this can be added to the example.com zone file:

```
john.example.com. NS ns1.john.example.com.
john.example.com. NS ns2.john.example.com.
# It's important to provide "glue"; in other words, let the world know
# the IPs for these name servers.
ns1.john.example.com. 10.9.8.7
ns2.john.example.com. 10.5.77.65
```

John, who is running is own nameservers with the IPs 10.9.8.7 and 10.5.77.65 then has a zone file for john.example.com. that looks something like this:

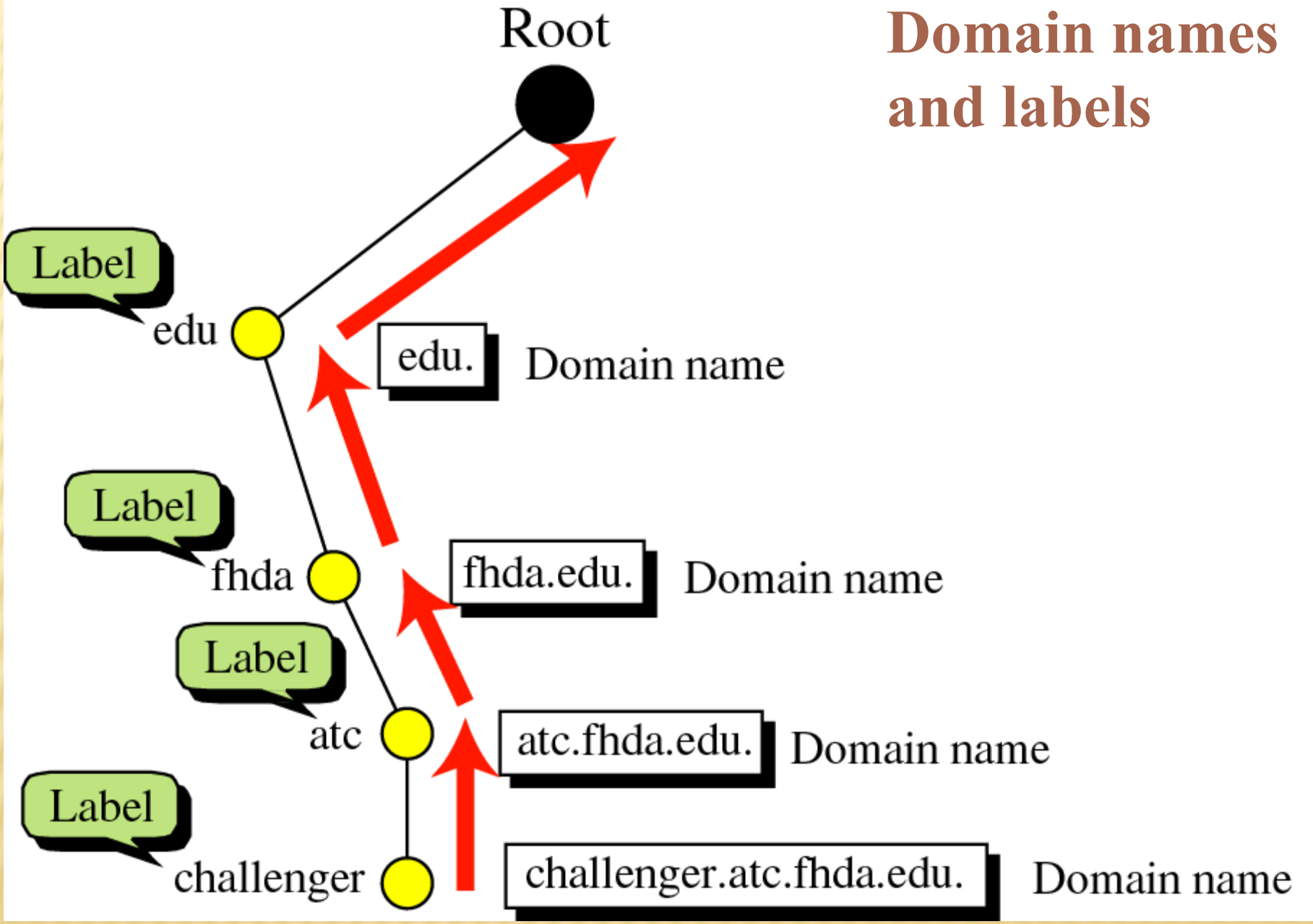
```
# It is best if the NS records for a subzone agree with the delegation
# records above
john.example.com. NS ns1.john.example.com.
john.example.com. NS ns2.john.example.com.

ns1.john.example.com. 10.9.8.7
ns2.john.example.com. 10.5.77.65

# Now that that is out of the way, here is the rest of the zone
john.example.com. 10.9.8.7
www.john.example.com. 10.5.77.65
john.example.com. MX 10 mail.john.example.com.
mail.john.example.com. 10.9.8.7
```

Example DNS entries from http://www.maradns.org/tutorial/recordtypes.html

Name Servers

- Domain names:
  - Two or more labels, separated by dots (e.g., cs166.net)
  - Rightmost label is the top-level domain (TLD)
- Hierarchy of authoritative name servers
  - Information about root domain
  - Information about its subdomains (A records) or references to other name servers (NS records)
- The authoritative name server hierarchy matches the domain hierarchy: root servers point to DNS servers for TLDs, etc.
- Root servers, and servers for TLDs change infrequently
- DNS servers refer to other DNS servers by name, not by IP: sometimes must bootstrap by providing an IP along with a name, called a glue record

Domain names and labels



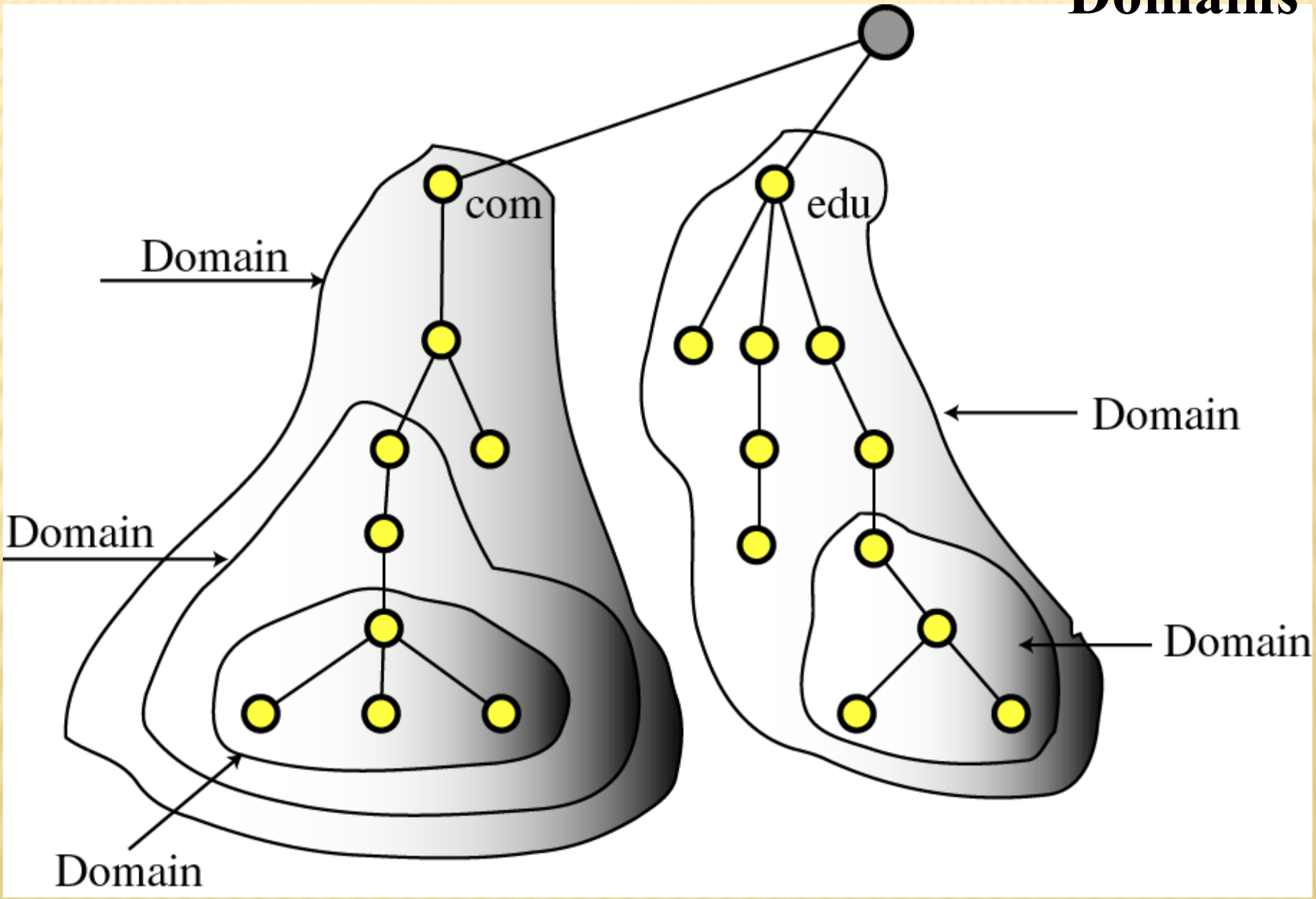
Namespace Management

- ICANN: Internet Corporation for Assigned Names and Numbers
- ICANN has the overall responsibility for managing DNS. It controls the root domain, delegating control over each top-level domain to a domain name registry
- Along with a small set of general TLDs, every country has its own TLD -- (cTLDS) – controlled by the government.
- ICANN is the governing body for all general TLDs
- Until 1999 all .com, .net and .org registries were handled by Network Solutions Incorporated.
- After November, 1999, ICANN and NSI had to allow for a shared registration system and there are currently over 500 registrars in the market
- Also since 1999, ICANN has created additional gTLDs including some which are sponsored by consortiums or groups of companies.

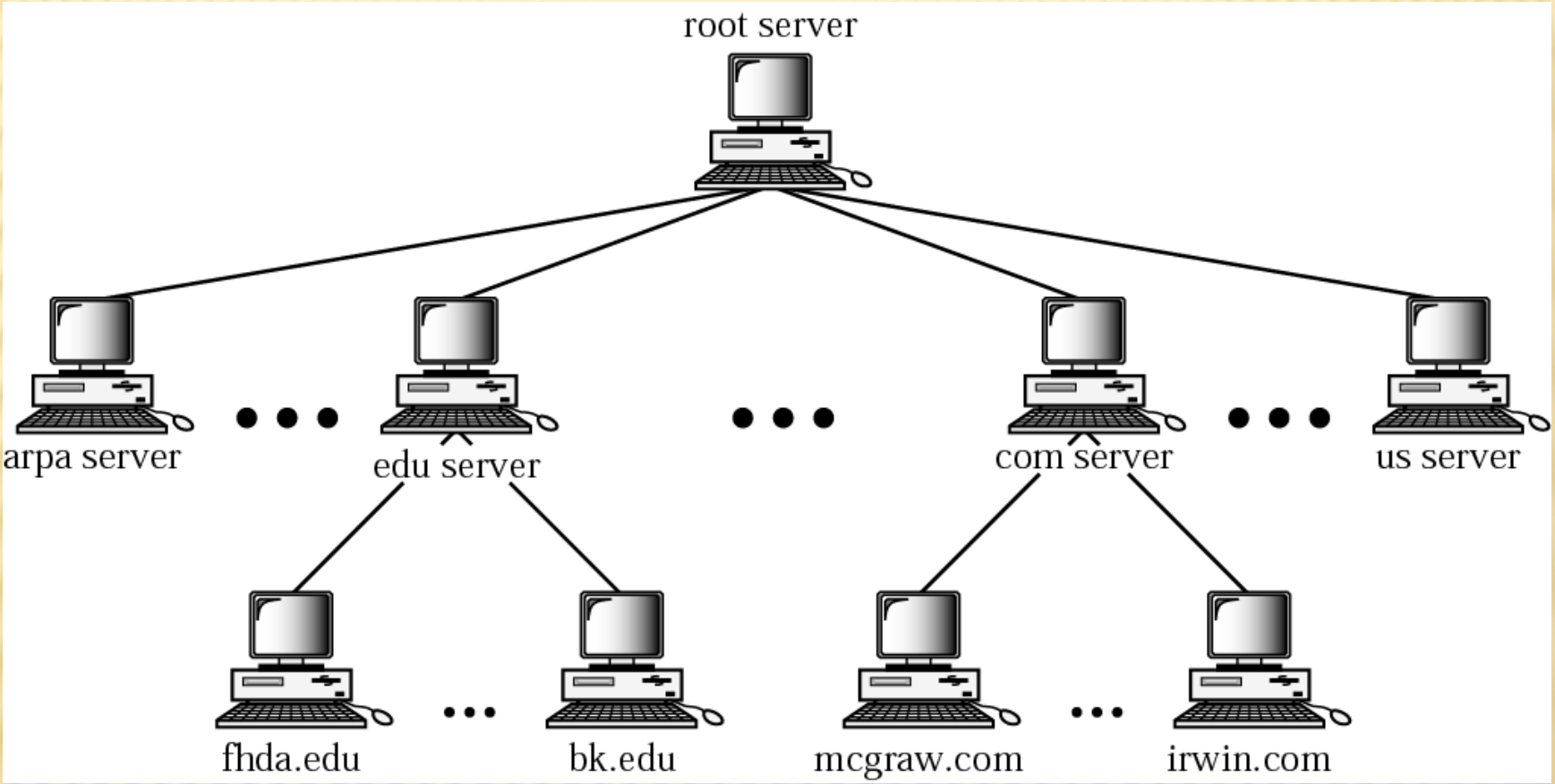
TOP LEVEL DOMAINS

- Started in 1984
- Originally supposed to be named by function
  - .com for commercial websites, .mil for military
- Eventually agreed upon unrestricted TLDs for .com, .net, .org, .info
- In 1994 started allowing country TLDs such as .it, .us
- Tried to move back to hierarchy of purpose in 2000 with creation of .aero, .museum, etc

Domains

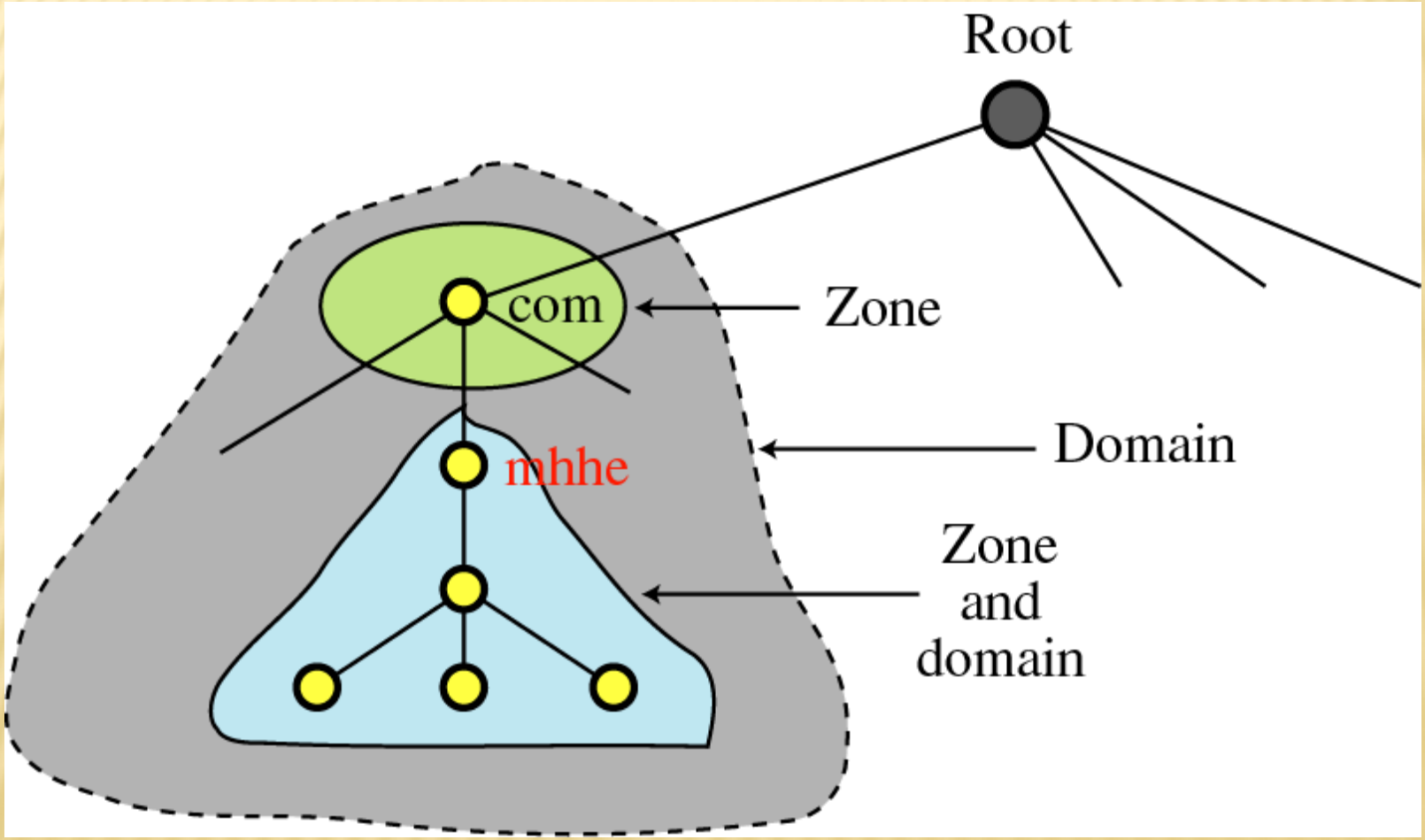


Hierarchy of name servers



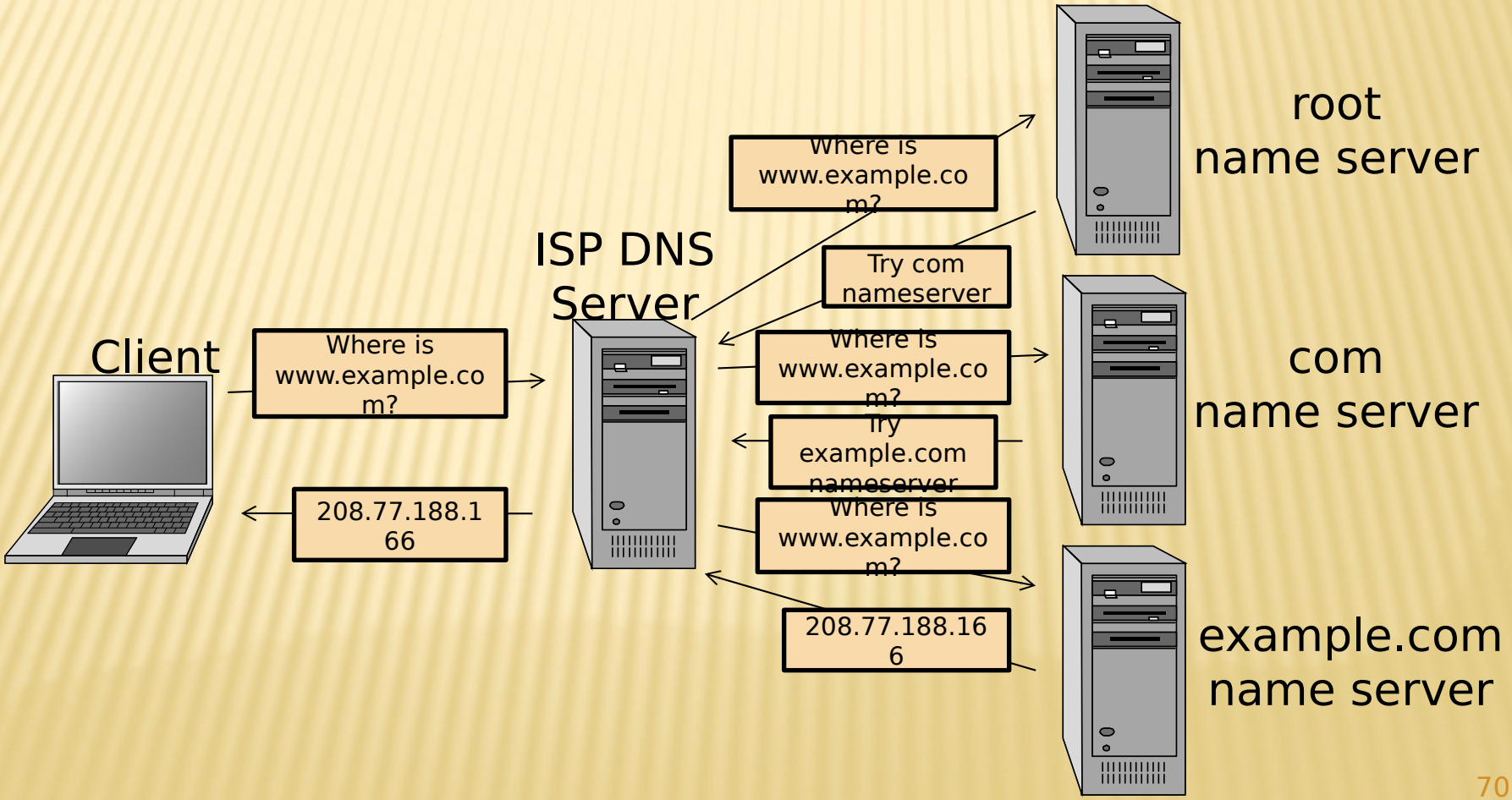
Zones and domains

- Zone: collection of connected nodes with the same authoritative DNS server

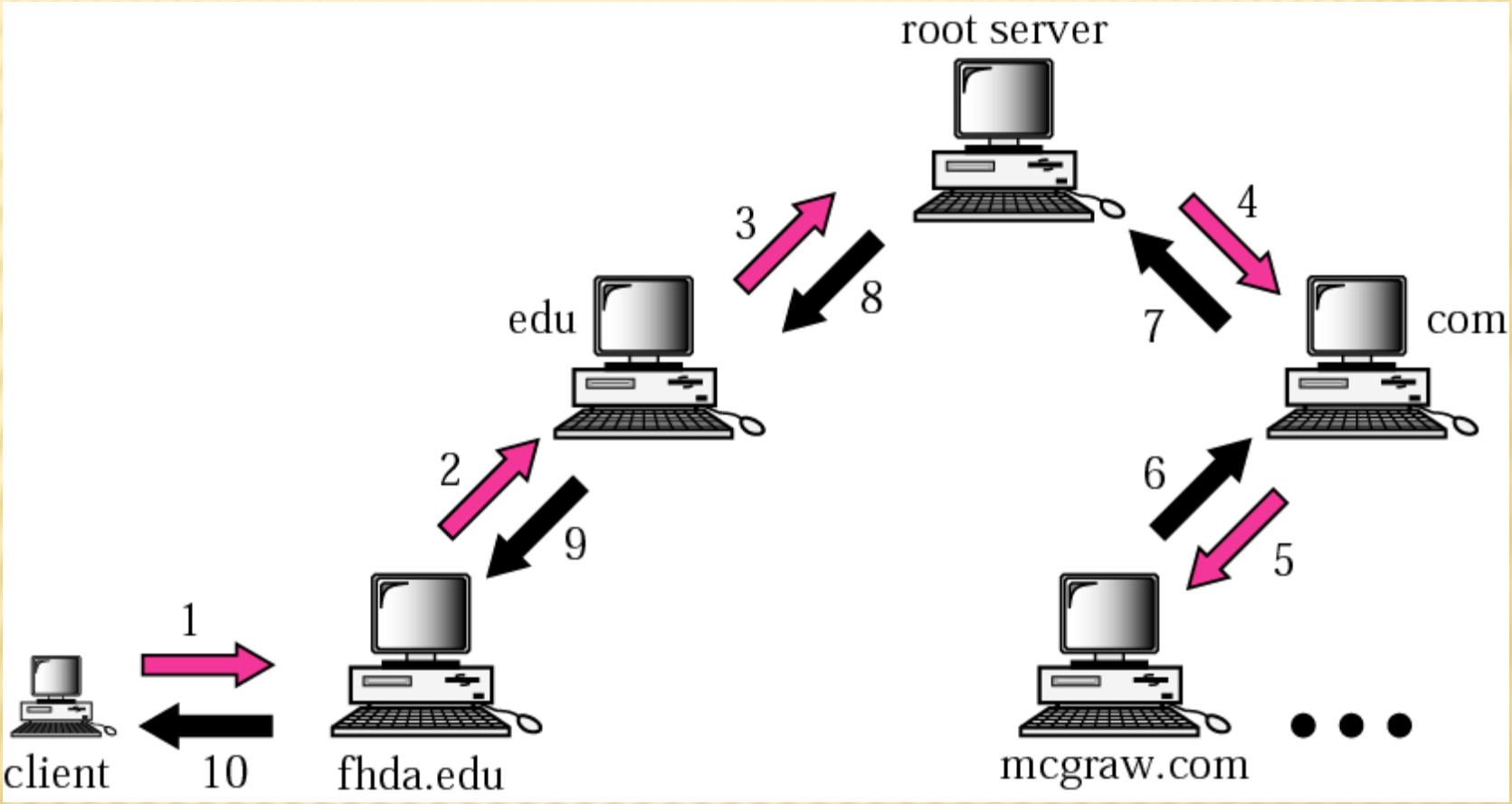


Name Resolution

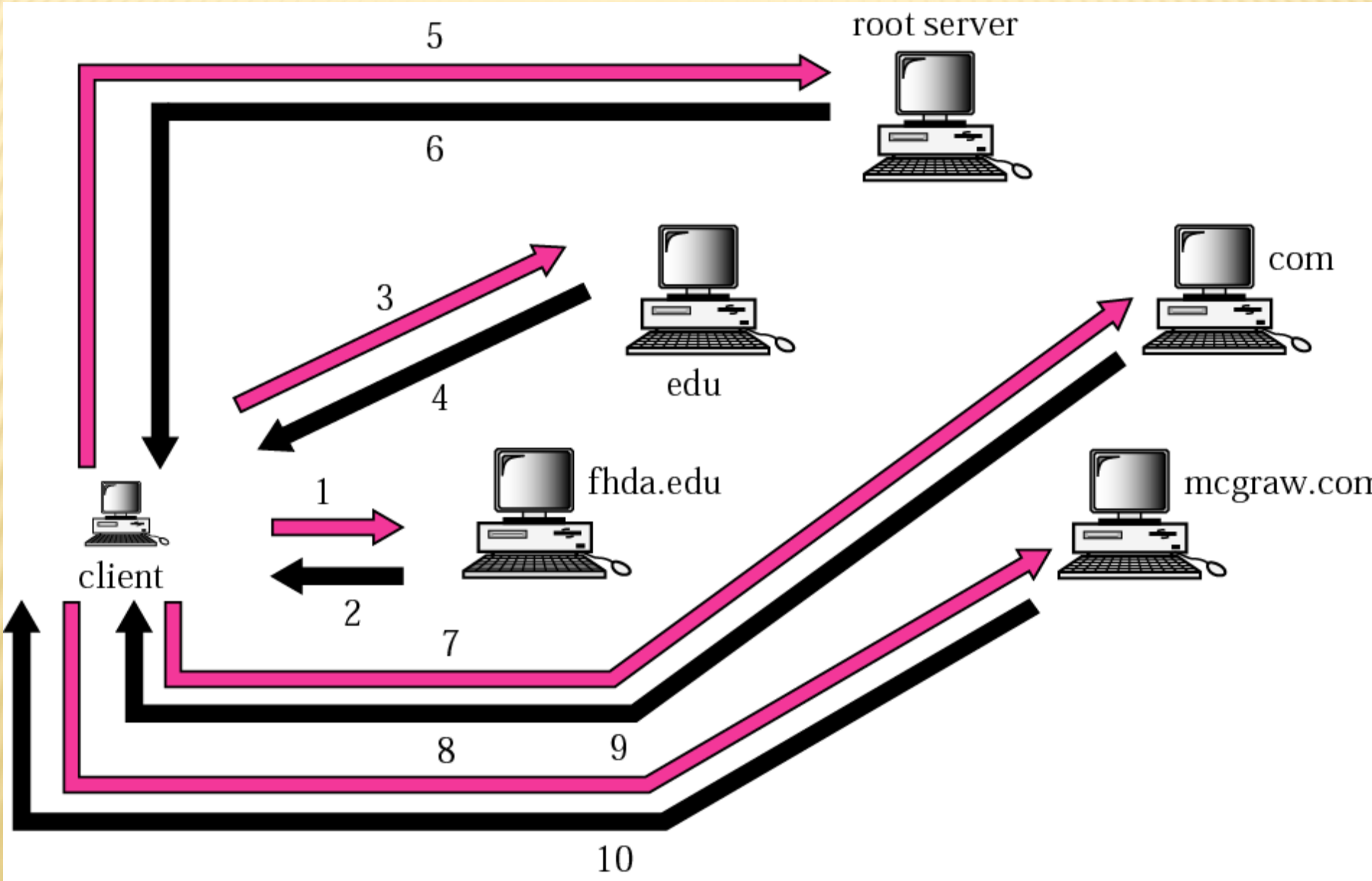
- Resolution method when answer not in cache:



Recursive resolution



Iterative resolution



AUTHORITATIVE NAME SERVERS

- Control distributed among authoritative name servers (ANSs)
  - Responsible for specific domains
  - Can designate other ANS for subdomains
- ANS can be master or slave
  - Master contains original zone table
  - Slaves are replicas, automatically updating
- Makes DNS fault tolerant, automatically distributes load
- ANS must be installed as a NS in parents' zone

DYNAMIC RESOLUTION

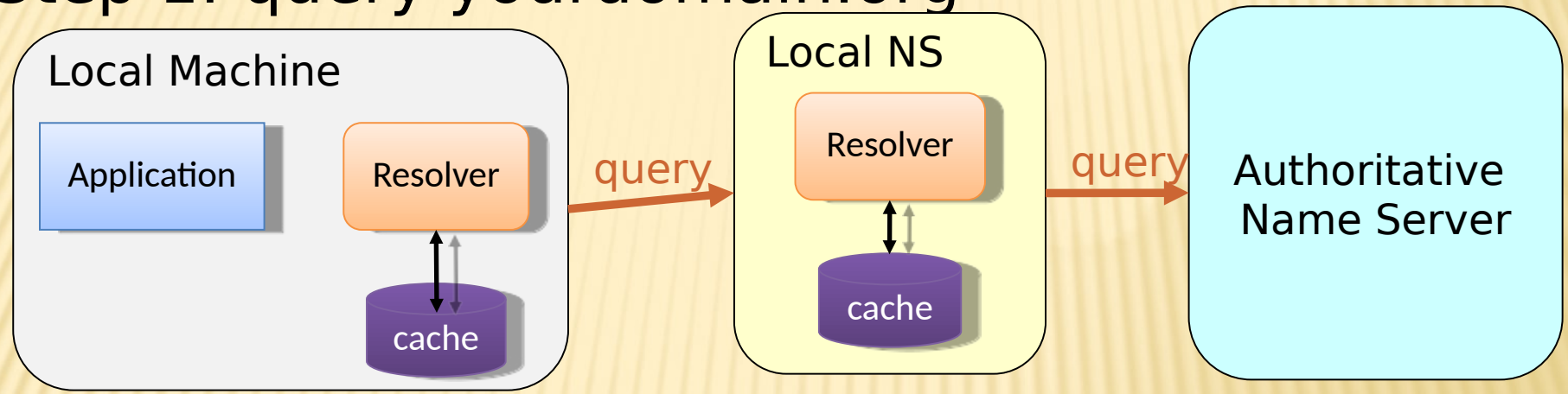
- Many large providers have more than one authoritative name server for a domain
- Problem: need to locate the instance of domain geographically closest to user
- Proposed solution: include first 3 octets of requester's IP in recursive requests to allow better service
- Content distribution networks

DNS Caching

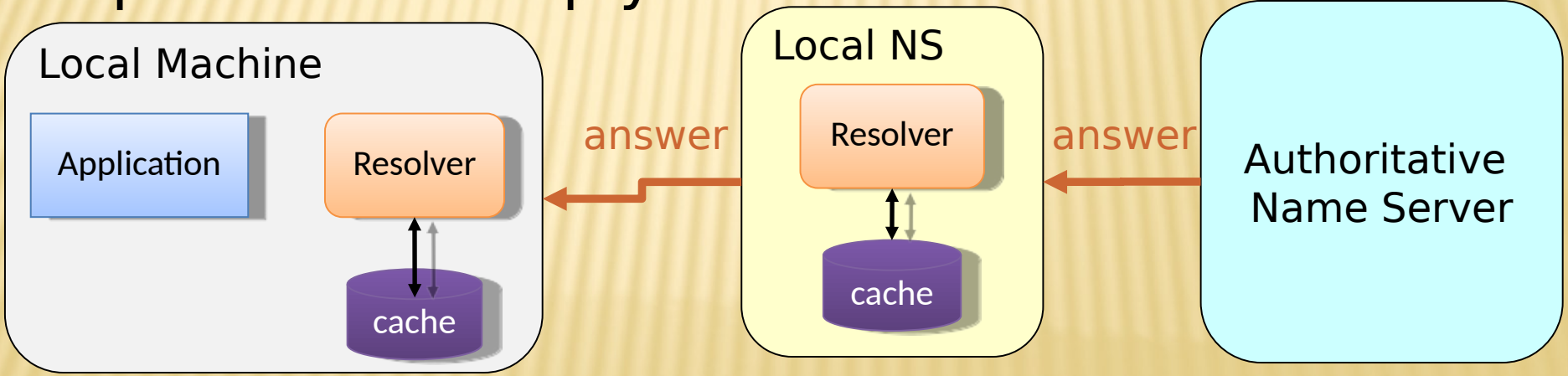
- There would be too much network traffic if a path in the DNS tree would be traversed for each query
  - Root zone would be rapidly overloaded
- DNS servers cache results for a specified amount of time
  - Specified by ANS **reply's time-to-live field**
- Operating systems and browsers also maintain resolvers and DNS caches
  - View in Windows with command ipconfig /displaydns
  - Associated privacy issues
- DNS queries are typically issued over UDP on port 53
  - 16-bit request identifier in payload

DNS CACHING

Step 1: query yourdomain.org

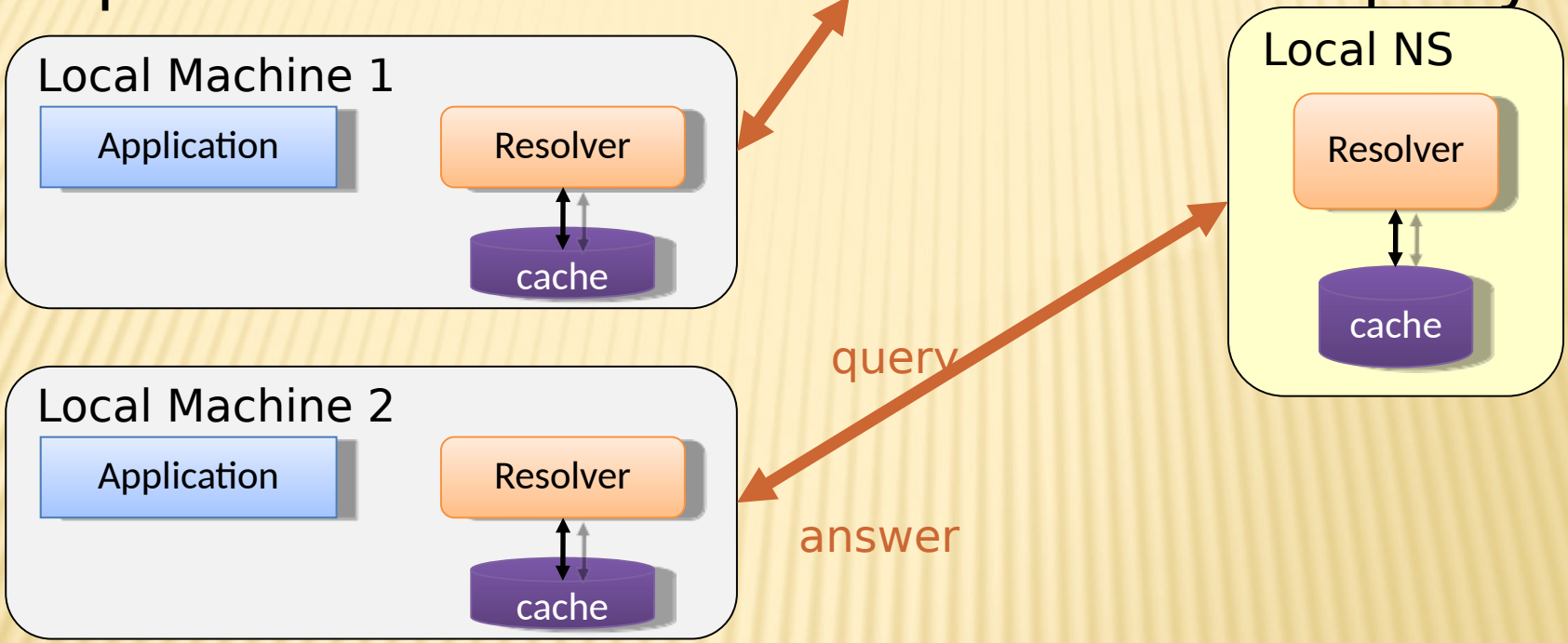


Step 2: receive reply and cache at local NS and h



DNS CACHING (CON'D)

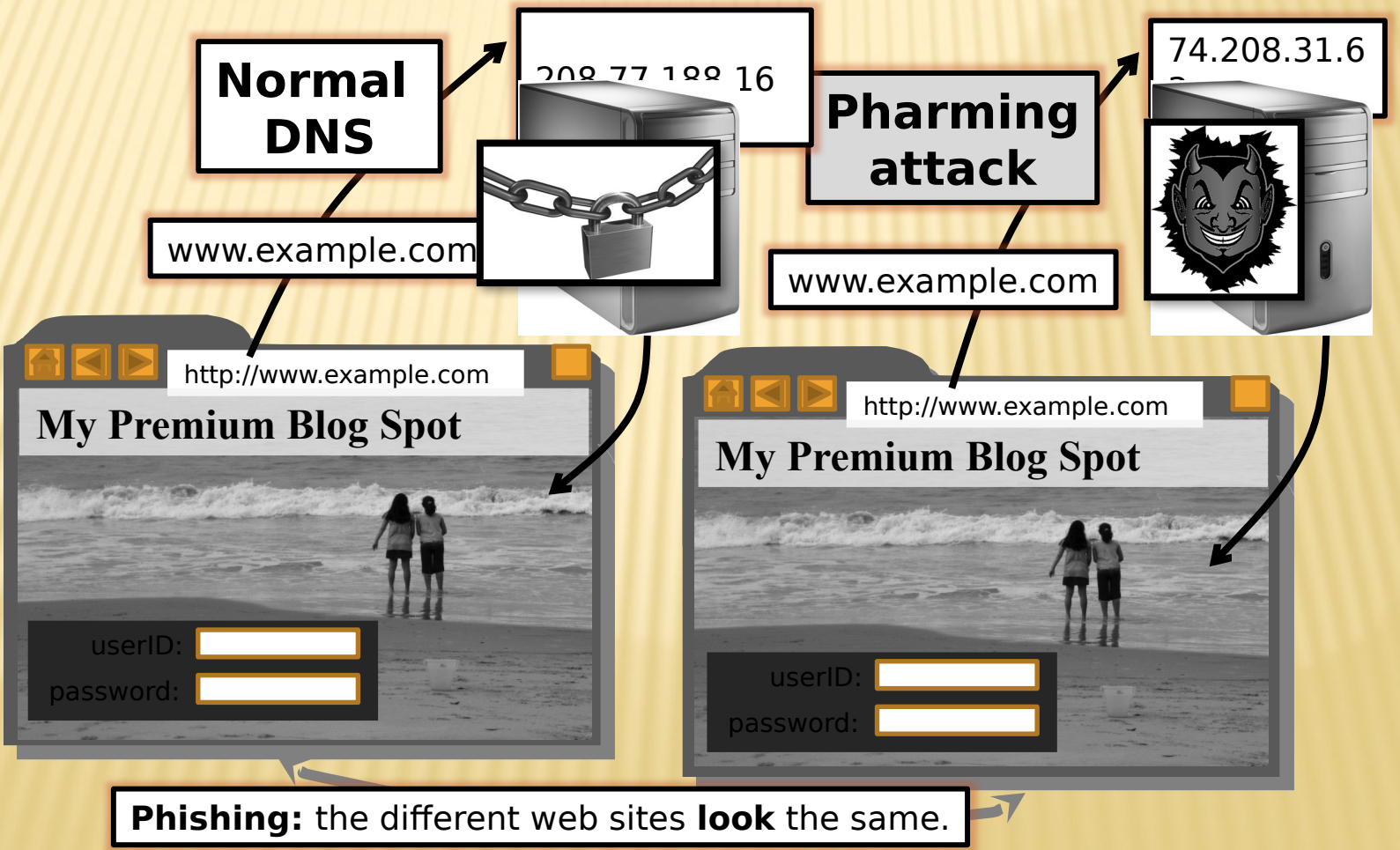
Step 3: use cached results rather than querying th



Step 4: Evict cache entries upon ttl expiration

PHARMING: DNS HIJACKING

- Changing IP associated with a server maliciously:



DNS Cache Poisoning

- Basic idea: give DNS servers false records and get it cached
- DNS uses a 16-bit request identifier to pair queries with answers
- Cache may be poisoned when a name server:
  - Disregards identifiers
  - Has predictable ids
  - Accepts unsolicited DNS records

DNS Cache Poisoning Prevention

- Use random identifiers for queries
- Always check identifiers
- Port randomization for DNS requests
- Deploy DNSSEC
  - Challenging because it is still being deployed and requires reciprocity

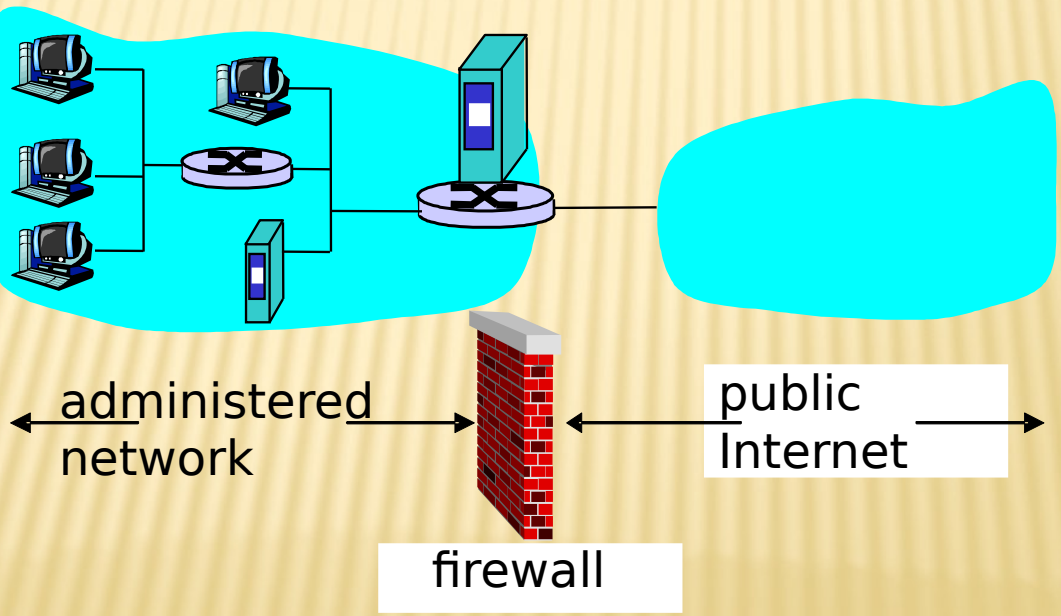
DNSSEC

- Guarantees:
  - Authenticity of DNS answer origin
  - Integrity of reply
  - Authenticity of denial of existence
- Accomplishes this by signing DNS replies at each step of the way
- Uses public-key cryptography to sign responses
- Typically use trust anchors, entries in the OS to bootstrap the process

FIREWALLS

firewall

isolates organization's internal net from larger Internet, allowing some packets to pass, blocking others.



FIREWALLS: WHY

prevent denial of service attacks:

- SYN flooding: attacker establishes many bogus TCP connections, no resources left for “real” connections

prevent illegal modification/access of internal data.

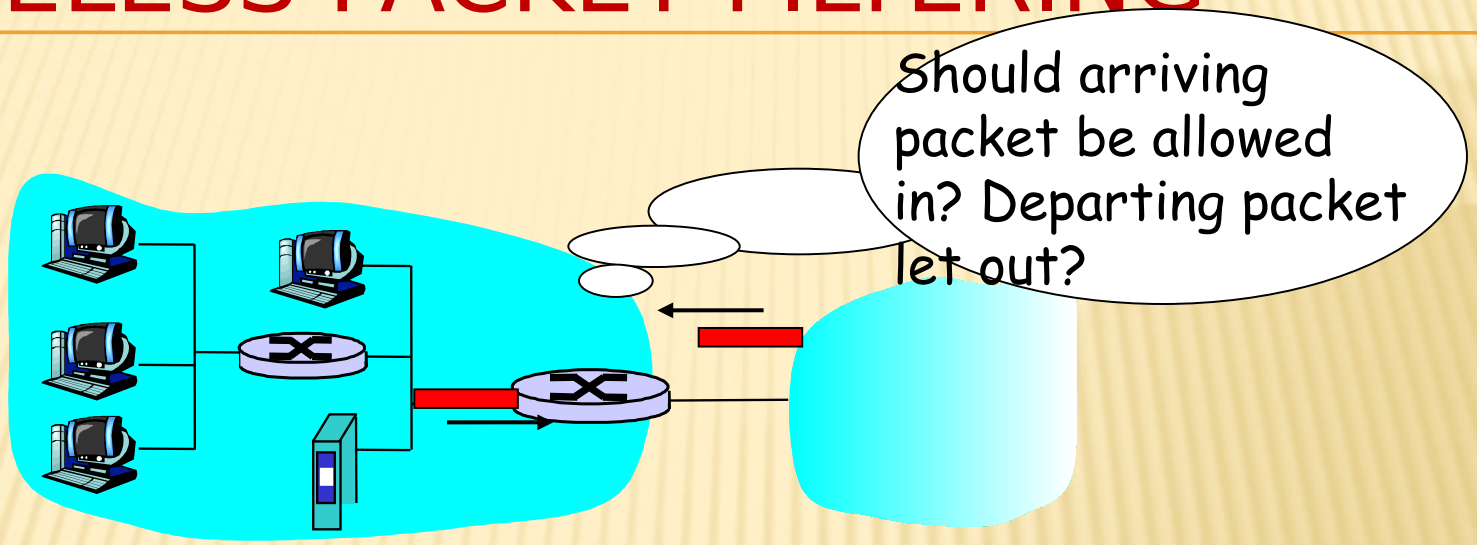
- e.g., attacker replaces CIA’s homepage with something else

allow only authorized access to inside network (set of authenticated users/hosts)

three types of firewalls:

- stateless packet filters
- stateful packet filters
- application gateways

STATELESS PACKET FILTERING



- internal network connected to Internet via router firewall
- router filters packet-by-packet, decision to forward/drop packet based on:
  - source IP address, destination IP address
  - TCP/UDP source and destination port numbers
  - ICMP message type
  - TCP SYN and ACK bits

STATELESS PACKET FILTERING: EXAMPLE

- example 1: block incoming and outgoing datagrams with IP protocol field = 17 and with either source or dest port = 23.
  - all incoming, outgoing UDP flows and telnet connections are blocked.
- example 2: Block inbound TCP segments with ACK=0.
  - prevents external clients from making TCP connections with internal clients, but allows internal clients to connect to outside.

STATELESS PACKET FILTERING: MORE EXAMPLES

Policy	Firewall Setting
No outside Web access.	Drop all outgoing packets to any IP address, port 80
No incoming TCP connections, except those for institution's public Web server only.	Drop all incoming TCP SYN packets to any IP except 130.207.244.203, port 80
Prevent Web-radios from eating up the available bandwidth.	Drop all incoming UDP packets - except DNS and router broadcasts.
Prevent your network from being used for a smurf DoS attack.	Drop all ICMP packets going to a "broadcast" address (eg 130.207.255.255).
Prevent your network from being tracerouted	Drop all outgoing ICMP TTL expired traffic

ACCESS CONTROL LISTS

- *ACL:* table of rules, applied top to bottom to incoming packets: (action, condition) pairs

action	source address	dest address	protocol	source port	dest port	flag bit
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80 (web)	any
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53 (DNS)	---
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----
deny	all	all	all	all	all	all

STATEFUL PACKET FILTERING

- stateless packet filter: heavy handed tool
  - admits packets that "make no sense," e.g., dest port = 80, ACK bit set, even though no TCP connection established:

action	source address	dest address	protocol	source port	dest port	flag bit
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK

- *stateful packet filter:* track status of every TCP connection
  - track connection setup (SYN), teardown (FIN): can determine whether incoming, outgoing packets "makes sense"
  - timeout inactive connections at firewall: no

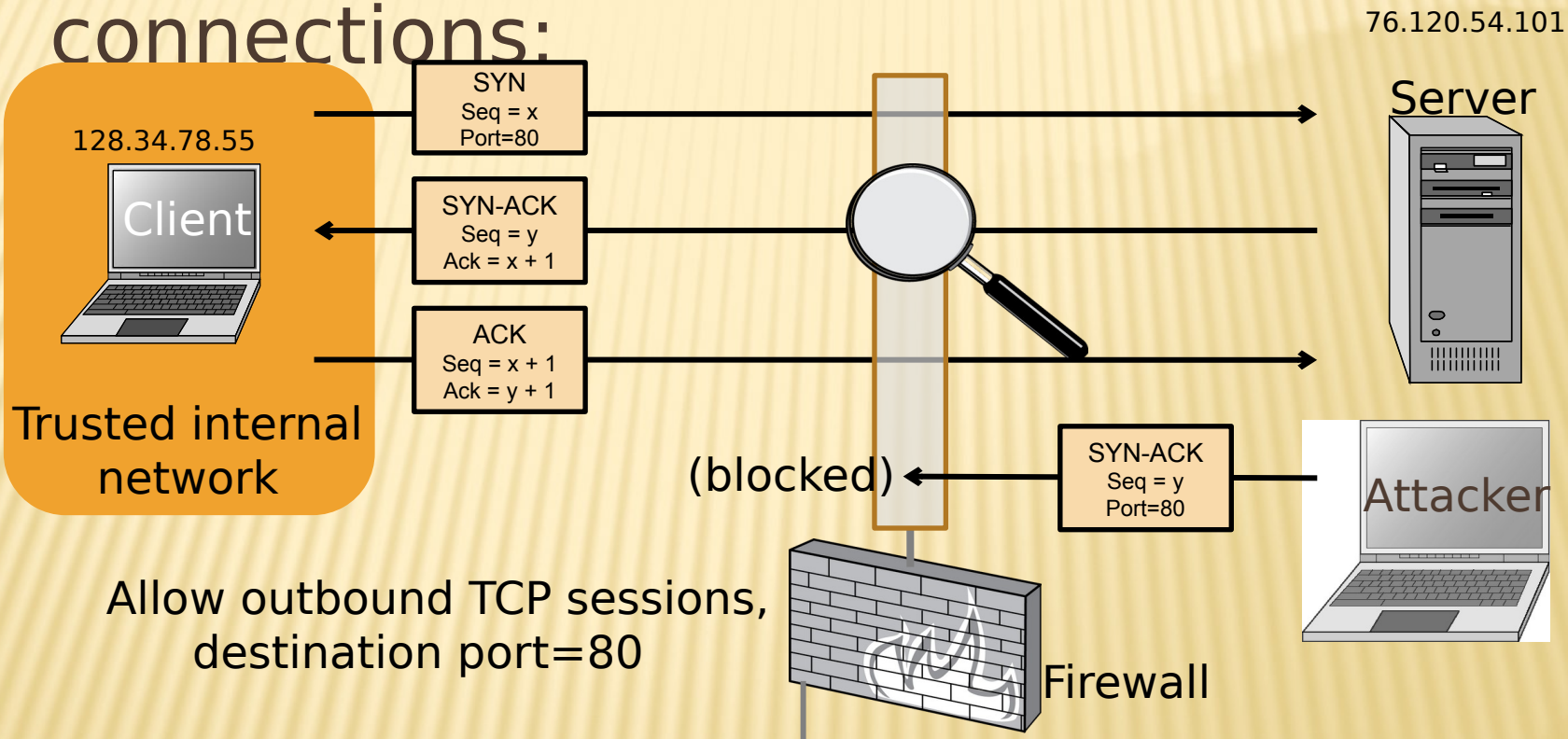
STATEFUL PACKET FILTERING

- ACL augmented to indicate need to check connection state table before admitting packet

action	source address	dest address	proto	source port	dest port	flag bit	check conxion
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any	
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK	X
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---	
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----	X
deny	all	all	all	all	all	all	

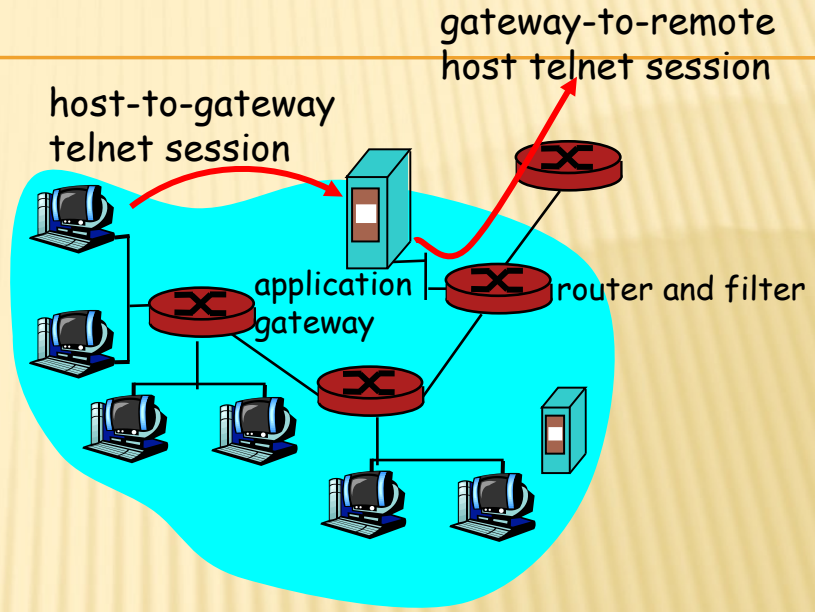
STATEFULL FIREWALL EXAMPLE

- Allow only requested TCP connections:



APPLICATION GATEWAYS

- filters packets on application data as well as on IP/TCP/UDP fields.
- example: allow select internal users to telnet outside.



1. require all telnet users to telnet through gateway.
2. for authorized users, gateway sets up telnet connection to dest host. Gateway relays data between 2 connections
3. router filter blocks all telnet connections not originating from gateway.

LIMITATIONS OF FIREWALLS AND GATEWAYS

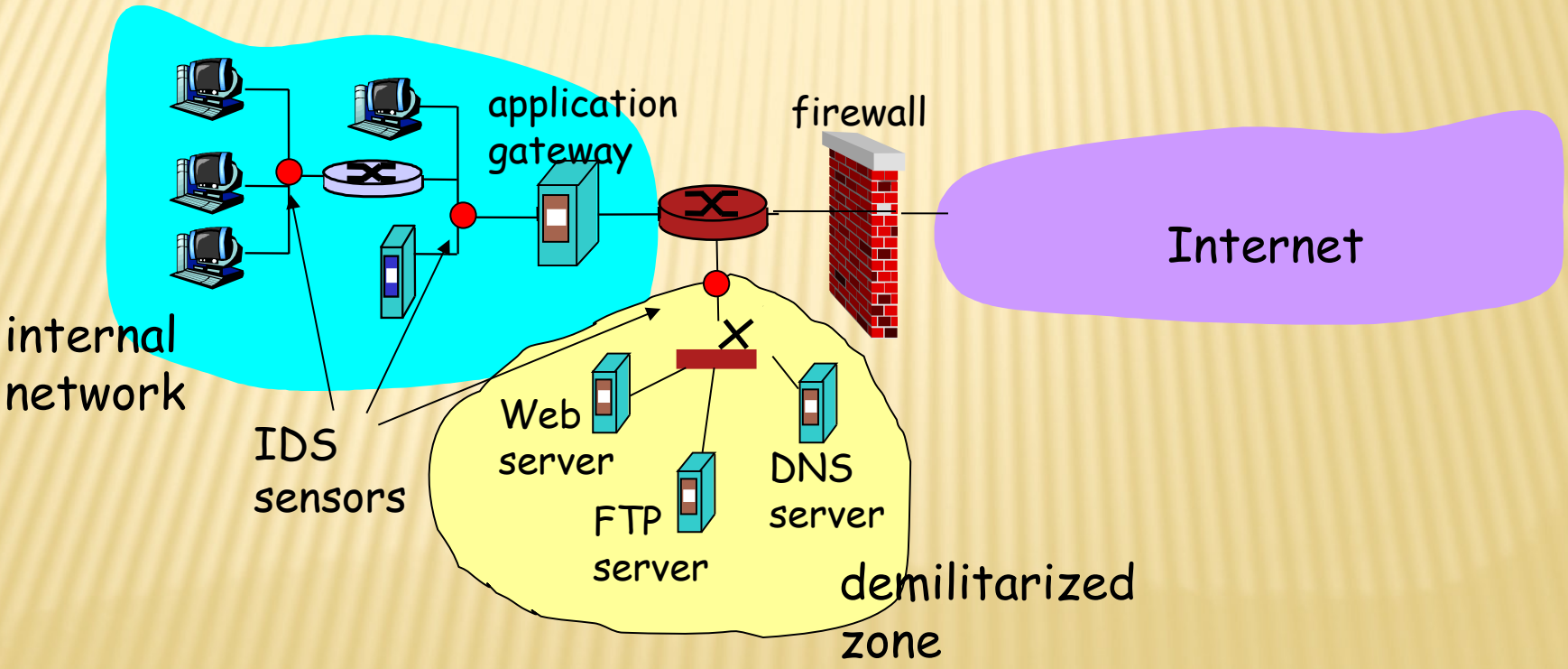
- IP spoofing: router can't know if data "really" comes from claimed source
- if multiple app's. need special treatment, each has own app. gateway.
- client software must know how to contact gateway.
  - e.g., must set IP address of proxy in Web browser
- filters often use all or nothing policy for UDP.
- tradeoff: degree of communication with outside world, level of security
- many highly protected sites still suffer from attacks.

INTRUSION DETECTION SYSTEMS

- packet filtering:
  - operates on TCP/IP headers only
  - no correlation check among sessions
- *IDS: intrusion detection system*
  - *deep packet inspection:* look at packet contents (e.g., check character strings in packet against database of known virus, attack strings)
  - examine correlation among multiple packets
    - port scanning
    - network mapping
    - DoS attack

INTRUSION DETECTION SYSTEMS

- multiple IDSs: different types of checking at different locations



IP SECURITY (IPSEC)

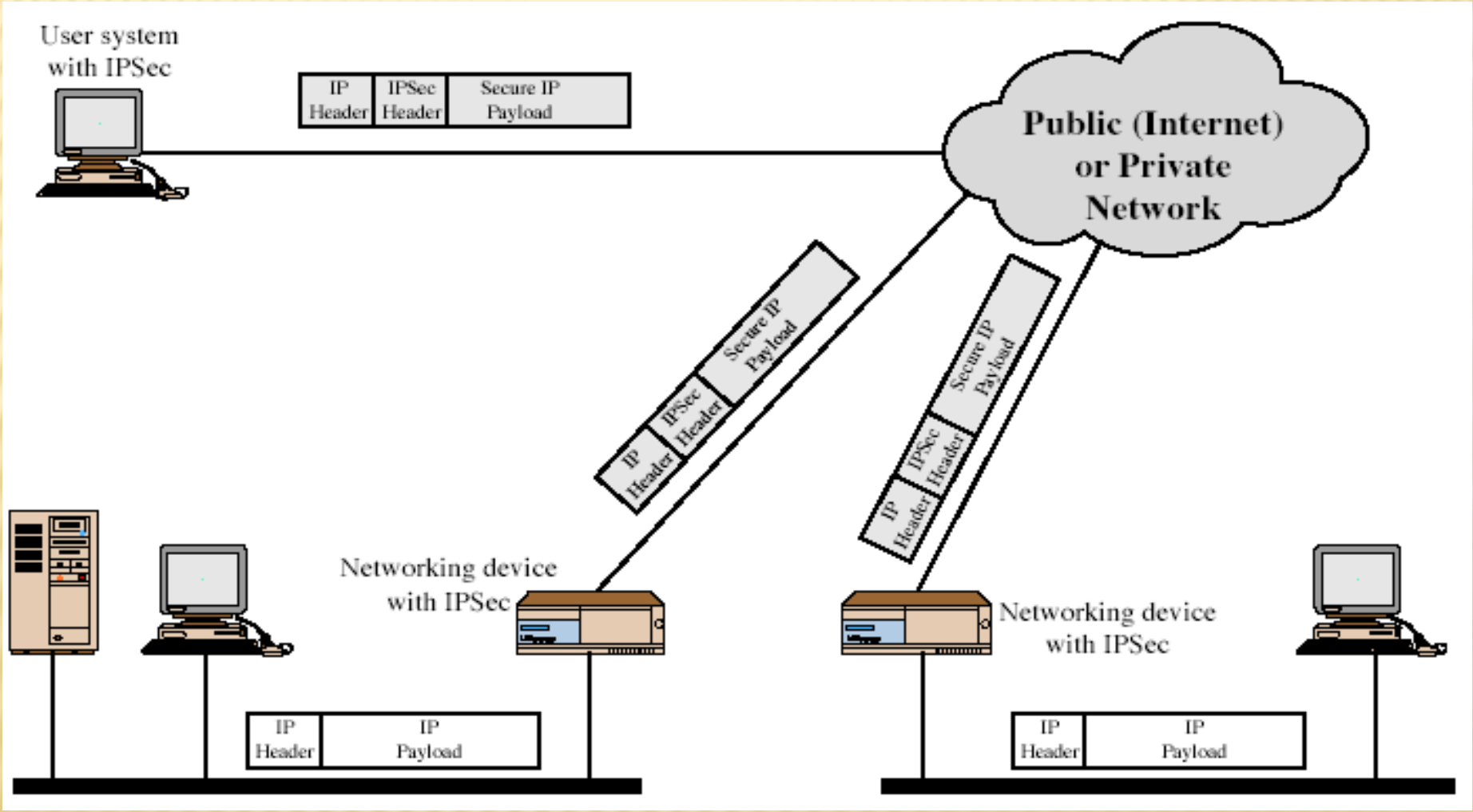
- Suite of protocols from Internet Engineering Task Force (IETF) providing encryption and authentication at the IP layer
  - Arose from needs identified in RFC 1636
  - Specifications in:
    - RFC 2401: Security architecture
    - RFC 2402: Authentication
    - RFC 2406: Encryption
    - RFC 2408: Key management
- Objective is to encrypt and/or authenticate **all** traffic at the IP level.

IP SECURITY ISSUES

- Eavesdropping
- Modification of packets in transit
- Identity spoofing (forged source IP addresses)
- Denial of service

- Many solutions are application-specific
  - TLS for Web, S/MIME for email, SSH for remote login
- IPSec aims to provide a framework of open standards for secure communications over IP
  - Protect every protocol running on top of IPv4 and IPv6

TYPICAL USAGE



IPSEC SERVICES

- Data origin authentication
- Confidentiality
- Connectionless and partial sequence integrity
  - Connectionless = integrity for a single IP packet
  - Partial sequence integrity = prevent packet replay
- Limited traffic flow confidentiality
  - Eavesdropper cannot determine who is talking
- These services are transparent to applications above transport (TCP/UDP) layer

Major IPSec Components

- Security Association (SA) Database
  - Each SA refers to all the security parameters of one communication direction
  - For two-way communications, at least two SAs are needed.
- Two Protocols
  - AH – Authentication Header
  - ESP – Encapsulating Security Payload
    1. Encryption only
    2. Encryption with authentication
- Two Encapsulation modes
    1. Transport mode
    2. Tunnel mode

USES OF IPSEC

- **Virtual Private Network (VPN) establishment**
  - For connecting remote offices and users using public Internet
- **Low-cost remote access**
  - e.g. teleworker gains secure access to company network via local call to ISP
- **Extranet connectivity**
  - Secure communication with partners, suppliers, etc.

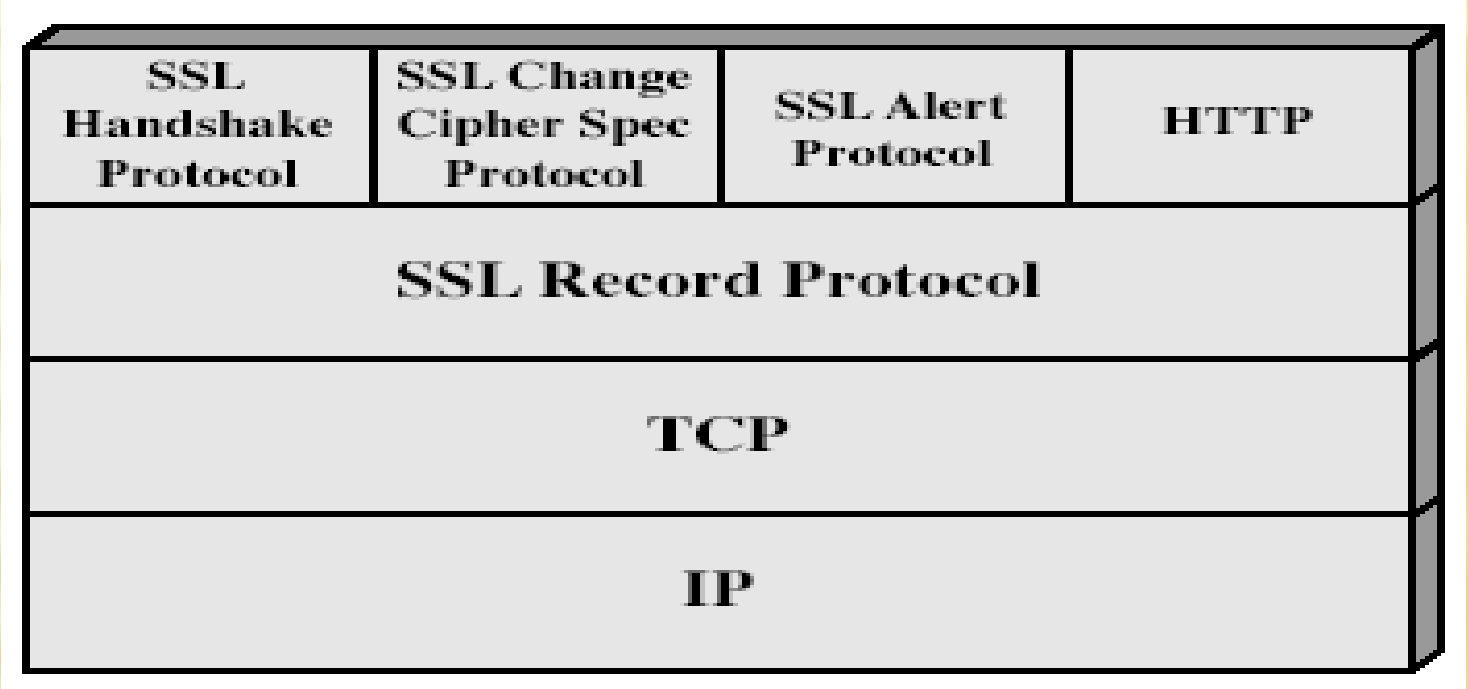
WEB SECURITY

- Web now widely used by business, government, individuals
- but Internet & Web are vulnerable
- have a variety of threats
  - integrity
  - confidentiality
  - denial of service
  - authentication
- need added security mechanisms

SSL (SECURE SOCKET LAYER)

- transport layer security service
- originally developed by Netscape
- version 3 designed with public input
- subsequently became Internet standard known as TLS (Transport Layer Security)
- uses TCP to provide a reliable end-to-end service
- SSL has two layers of protocols

SSL ARCHITECTURE



SSL ARCHITECTURE

- **SSL session**
  - an association between client & server
  - created by the Handshake Protocol
  - define a set of cryptographic parameters
  - may be shared by multiple SSL connections
- **SSL connection**
  - a transient, peer-to-peer, communications link
  - associated with 1 SSL session

SSL RECORD PROTOCOL

- **confidentiality**
  - using symmetric encryption with a shared secret key defined by Handshake Protocol
  - IDEA, RC2-40, DES-40, DES, 3DES, RC4-40, RC4-128
  - message is compressed before encryption
- **message integrity**
  - using a MAC with shared secret key
  - similar to HMAC but with different padding

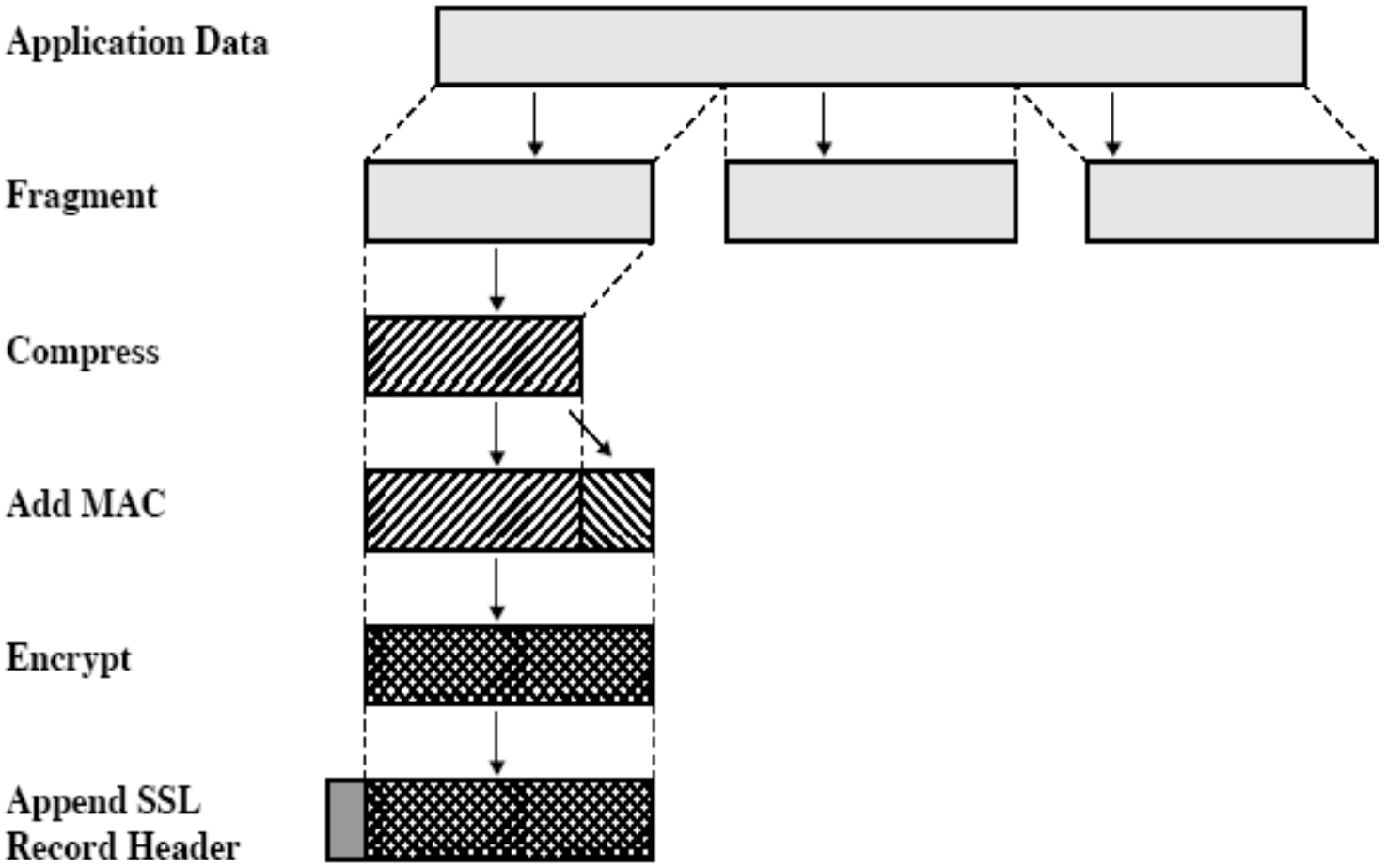


Figure 17.3 SSL Record Protocol Operation

SSL CHANGE CIPHER SPEC PROTOCOL

- one of 3 SSL specific protocols which use the SSL Record protocol
- a single message
- causes pending state to become current
- hence updating the cipher suite in use

TLS (TRANSPORT LAYER SECURITY)

- IETF standard RFC 2246 similar to SSLv3
- with minor differences
  - in record format version number
  - uses HMAC for MAC
  - a pseudo-random function expands secrets
  - has additional alert codes
  - some changes in supported ciphers
  - changes in certificate negotiations
  - changes in use of padding

IEEE 802.11 SECURITY

- *war-driving:* drive around Bay area, see what 802.11 networks available?
  - More than 9000 accessible from public roadways
  - 85% use no encryption/authentication
  - packet-sniffing and various attacks easy!
- securing 802.11
  - encryption, authentication
  - first attempt at 802.11 security: Wired Equivalent Privacy (WEP): a failure
  - current attempt: 802.11i

WIRED EQUIVALENT PRIVACY (WEP):

- authentication
  - host requests authentication from access point
  - access point sends 128 bit nonce
  - host encrypts nonce using shared symmetric key
  - access point decrypts nonce, authenticates host
- no key distribution mechanism
- authentication: knowing the shared key is enough

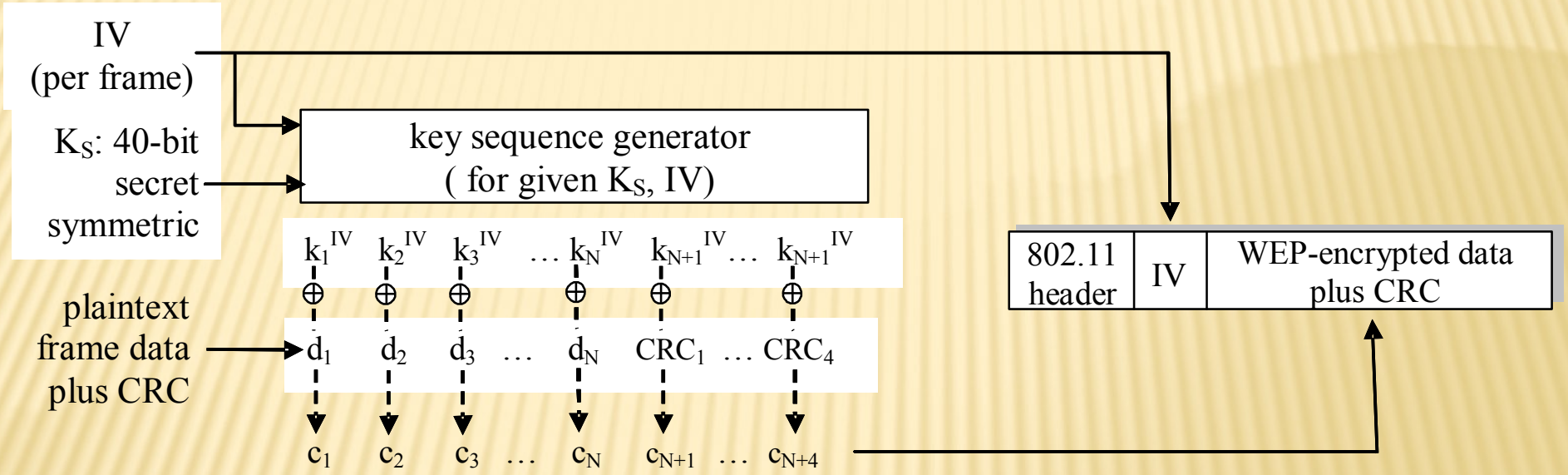
WEP DATA ENCRYPTION

- host/AP share 40 bit symmetric key (semi-permanent)
- host appends 24-bit initialization vector (IV) to create 64-bit key
- 64 bit key used to generate stream of keys, k_i^{IV}
- k_i^{IV} used to encrypt ith byte, d_i, in frame:

$$c_i = d_i \text{ XOR } k_i^{IV}$$

- IV and encrypted bytes, c_i sent in frame

802.11 WEP ENCRYPTION



Sender-side WEP encryption :ol

BREAKING 802.11 WEP ENCRYPTION

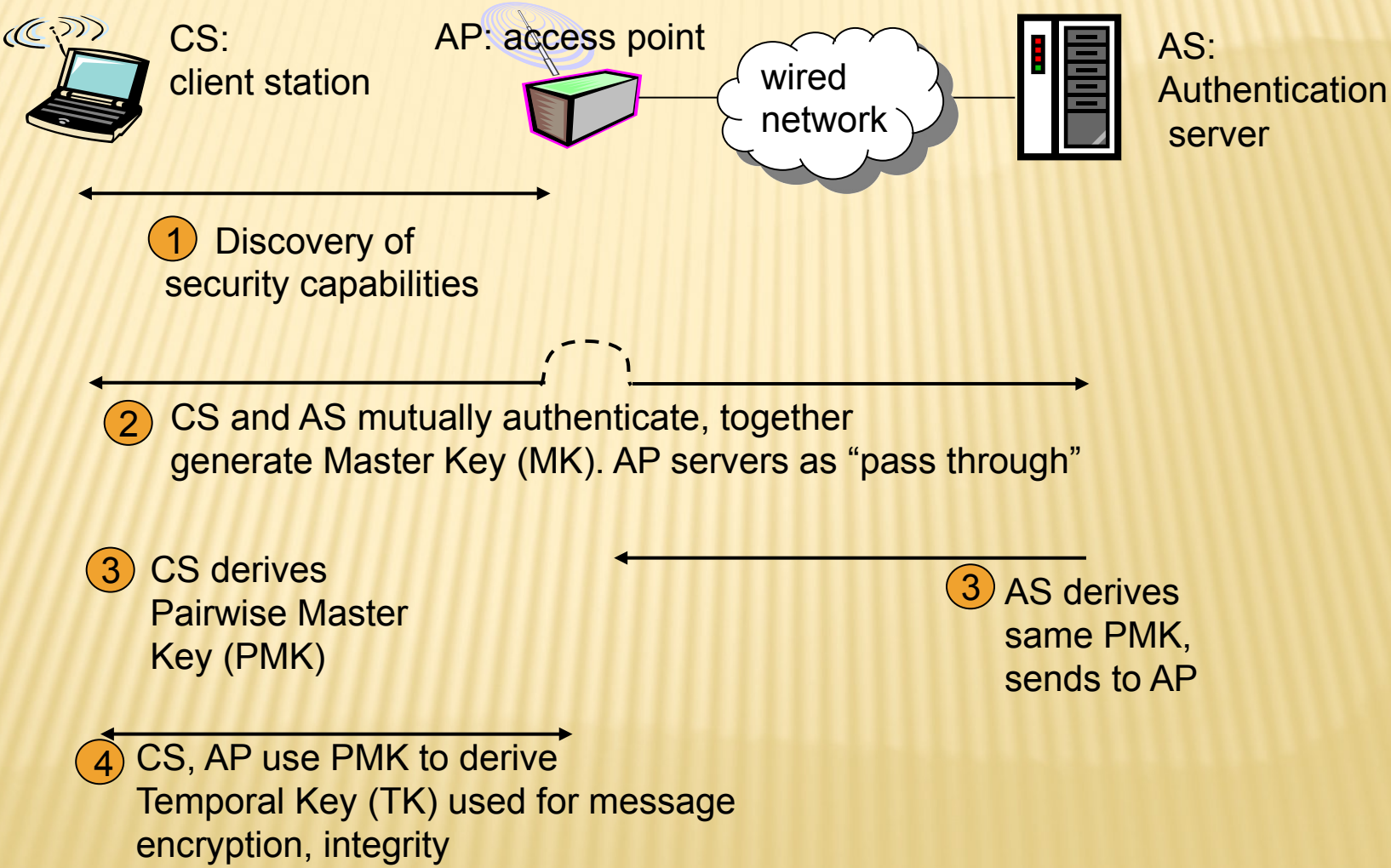
security hole:

- 24-bit IV, one IV per frame, -> IV's eventually reused
- IV transmitted in plaintext -> IV reuse detected
- attack:
  - Trudy causes Alice to encrypt known plaintext d_1 d_2 d_3 d_4 …
  - Trudy sees: $c_i = d_i$ XOR k_i^{IV}
  - Trudy knows c_i d_i, so can compute k_i^{IV}
  - Trudy knows encrypting key sequence k_1^{IV} k_2^{IV} k_3^{IV} …
  - Next time IV is used, Trudy can decrypt!

802.11I: IMPROVED SECURITY

- numerous (stronger) forms of encryption possible
- provides key distribution
- uses authentication server separate from access point

802.11I: FOUR PHASES OF OPERATION



VIRUSES, WORMS, TROJANS, ROOTKITS

- Malware can be classified into several categories, depending on propagation and concealment
- Propagation
  - Virus: human-assisted propagation (e.g., open email attachment)
  - Worm: automatic propagation without human assistance
- Concealment
  - Rootkit: modifies operating system to hide its existence
  - Trojan: provides desirable functionality but hides malicious operation
- Various types of payloads, ranging from annoyance to crime

INSIDER ATTACKS

- An **insider attack** is a security breach that is caused or facilitated by someone who is a part of the very organization that controls or builds the asset that should be protected.
- In the case of malware, an insider attack refers to a security hole that is created in a software system by one of its programmers.

DEFENSES AGAINST INSIDER ATTACKS

- Avoid single points of failure.
- Use code walk-throughs.
- Use archiving and reporting tools.
- Limit authority and permissions.
- Physically secure critical systems.
- Monitor employee behavior.
- Control software installations.