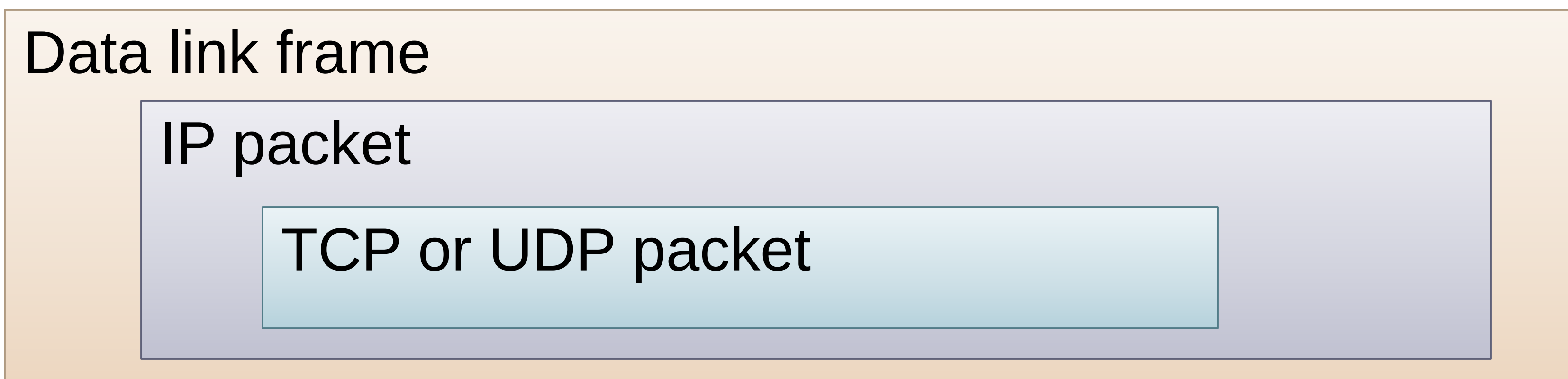
Networks: IP and TCP

Internet Protocol

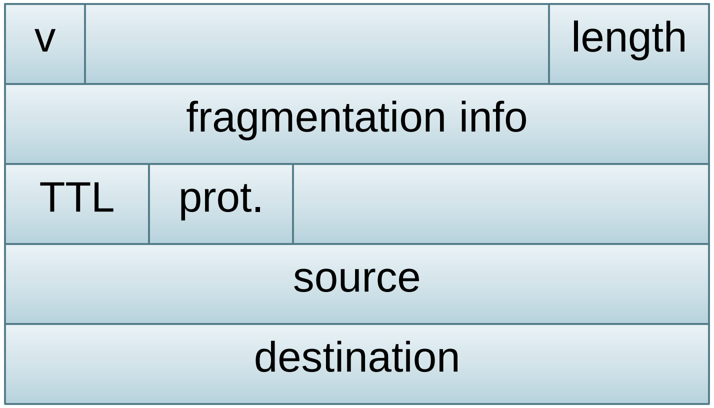
- **Connectionless**
  - Each packet is transported independently from other packets
- **Unreliable**
  - Delivery on a best effort basis
  - No acknowledgments
  - Packets may be lost, reordered, corrupted, or duplicated
- **IP packets**
  - Encapsulate TCP and UDP packets
  - Encapsulated into link-layer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

Classful addressing definition

Class	Leading bits	Size of *network number* bit field	Size of *rest* bit field	Number of networks	Addresses per network	Total addresses in class	Start address	End address	Default subnet mask in dot-decimal notation	CIDR notation
Class A	0	8	24	128 (2^7)	16,777,216 (2^{24})	2,147,483,648 (2^{31})	*0.0.0.0*	*127.0.0.0*[a]	*255.0.0.0*	*/8*
Class B	10	16	16	16,384 (2^{14})	65,536 (2^{16})	1,073,741,824 (2^{30})	*128.0.0.0*	*191.255.0.0*	*255.255.0.0*	*/16*
Class C	110	24	8	2,097,152 (2^{21})	256 (2^8)	536,870,912 (2^{29})	*192.0.0.0*	*223.255.255.0*	*255.255.255.0*	*/24*
Class D (multicast)	1110	not defined	not defined	not defined	not defined	268,435,456 (2^{28})	*224.0.0.0*	*239.255.255.255*	not defined	not defined
Class E (reserved)	1111	not defined	not defined	not defined	not defined	268,435,456 (2^{28})	*240.0.0.0*	*255.255.255.255*	not defined	not defined

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

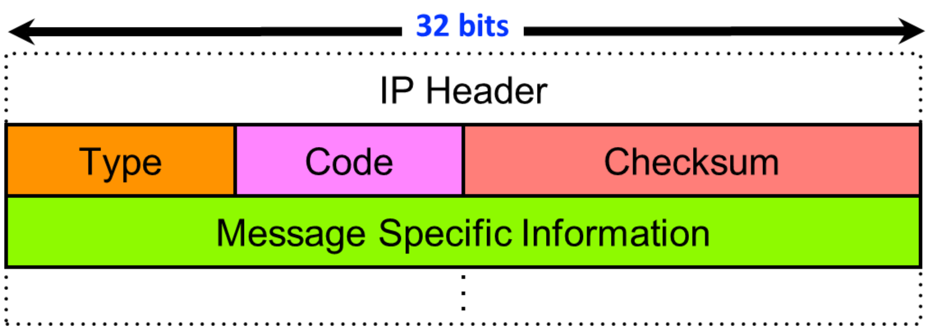
- **A router bridges two or more networks**
  - Operates at the network layer
  - Maintains tables to forward packets to the appropriate network
  - Forwarding decisions based solely on the destination address
- **Routing table**
  - Maps ranges of addresses to LANs or other gateway routers

Internet Routes

- **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 Format

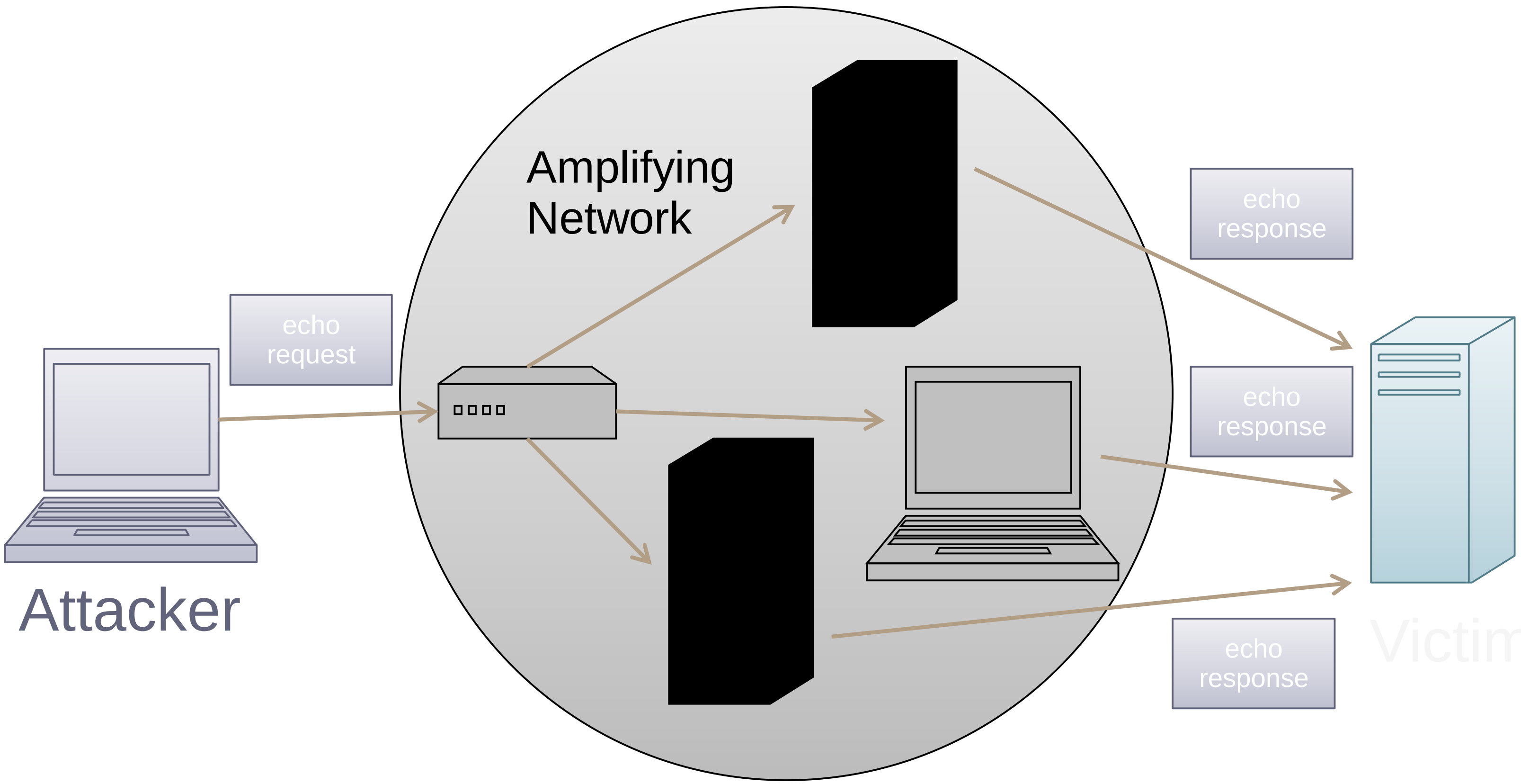
Type	Code	Description
0	0	Echo reply (used in ping)
3	0	Destination network unreachable
3	1	Destination host unreachable
3	2	Destination protocol unreachable
3	3	Destination port unreachable
3	4	Fragmentation required
3	6	Destination network unknown
3	7	Destination host unknown
8	0	Echo request (used in ping)
9	0	Router advertisement
10	0	Router discovery/selection/solicitation
11	0	TTL expired in transit



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



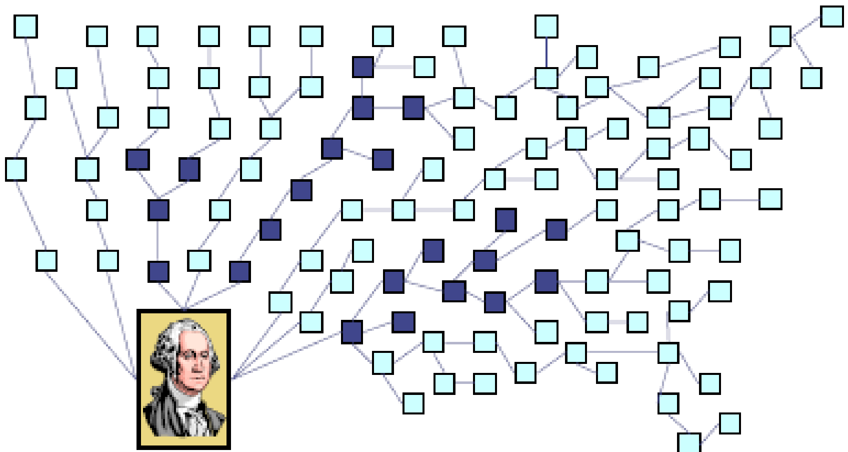
IP Vulnerabilities

- **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

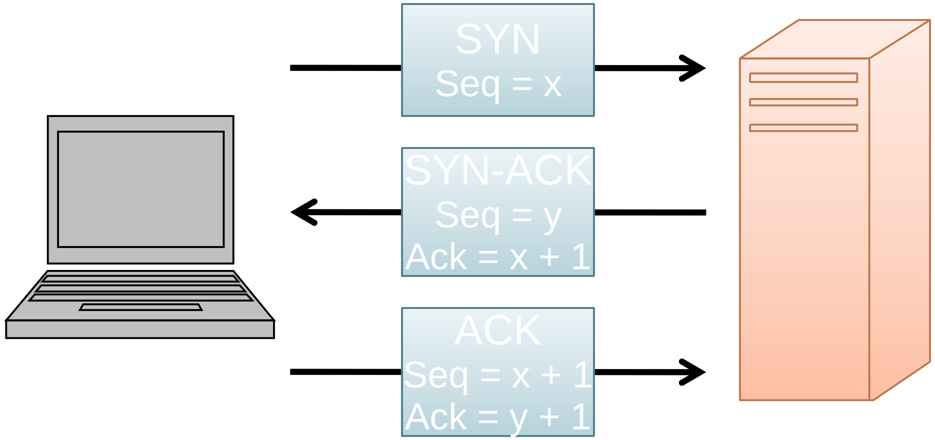
- **TCP** supports multiple concurrent applications on the same server
- Accomplishes this by having ports, 16 bit numbers identifying where data is directed
- The **TCP** header includes space for both a source and a destination port, thus allowing **TCP** to route all data
- In most cases, both **TCP** 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	Reserv ed	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**



TCP 3-way Handshake Protocol

- **When the server receives the initial SYN packet, it uses TCB (Transmission Control Block) to store the information about the connection.**
- **This is called half-open connection as only client-server connection is confirmed.**
- **The server stores the TCB in a queue that is only for the half-open connection.**
- **After the server gets ACK packet, it will take this TCB out of the queue and store in a different place.**
- **If ACK doesn’t arrive, the server will resend SYN+ACK packet. The TCB will eventually be discarded after a certain time period.**

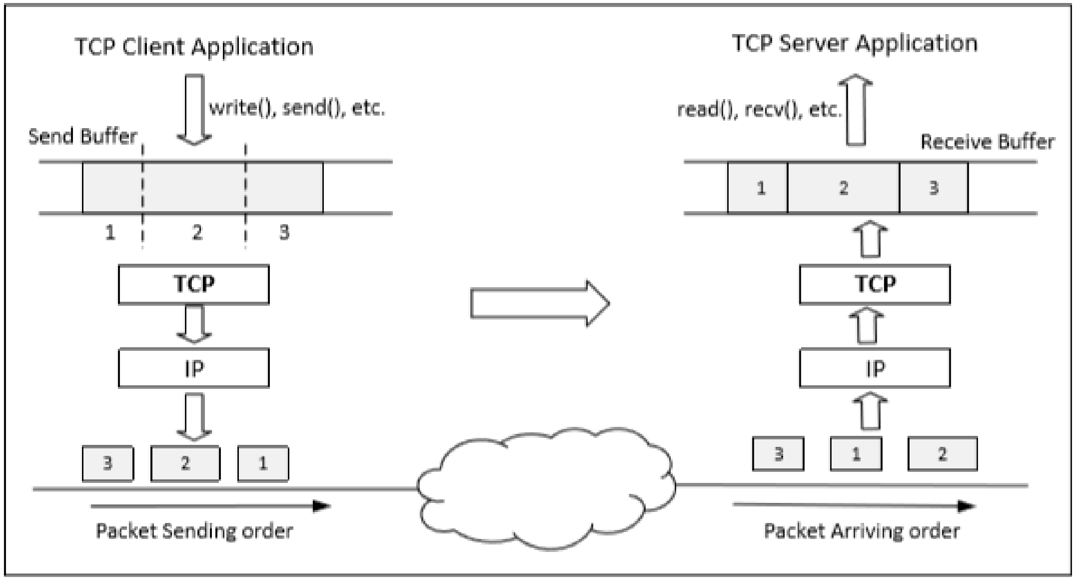
SYN Flood

- **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**

Countermeasures: SYN Cookies

- **After a server receives a SYN packet, it calculates a keyed hash (H) from the information in the packet using a secret key that is only known to the server.**
- **This hash (H) is sent to the client as the initial sequence number from the server. H is called SYN cookie.**
- **The server will not store the half-open connection in its queue.**
- **If the client is an attacker, H will not reach the attacker.**
- **If the client is not an attacker, it sends H+1 in the acknowledgement field.**
- **The server checks if the number in the acknowledgement field is valid or not by recalculating the cookie.**

TCP Data Transmission

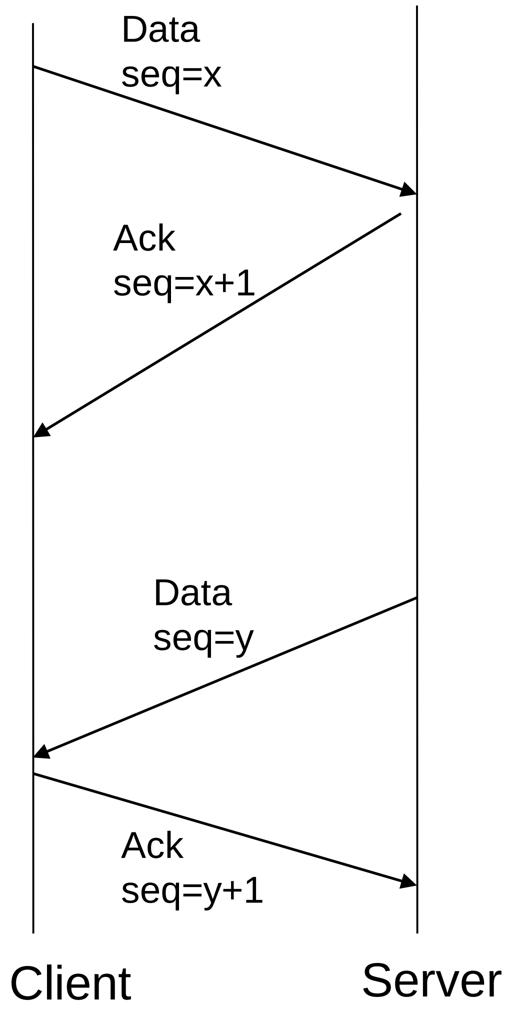


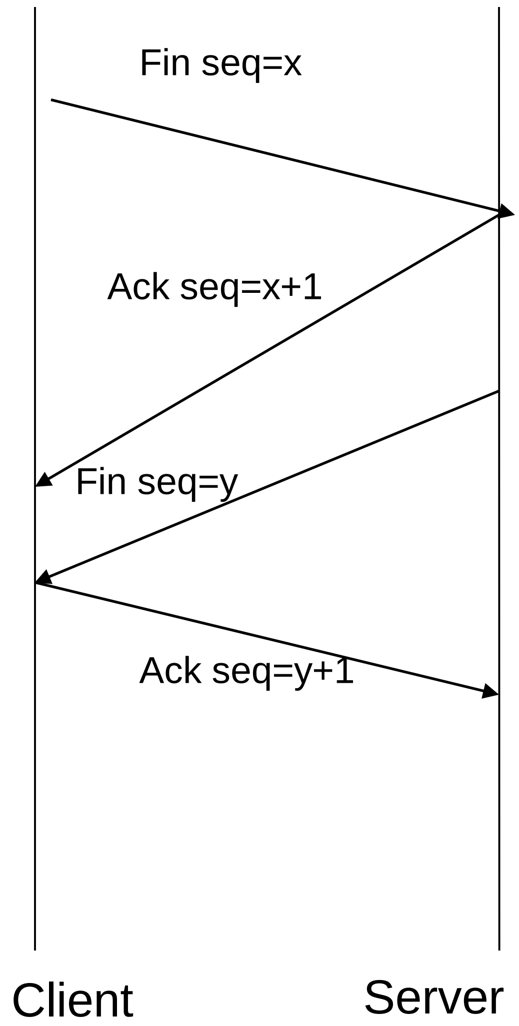
- Once a connection is established, OS allocates two buffers at each end, one for sending data (send buffer) and receiving buffer (receive buffer).
- When an application needs to send data out, it places data into the TCP send buffer.

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 entries 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

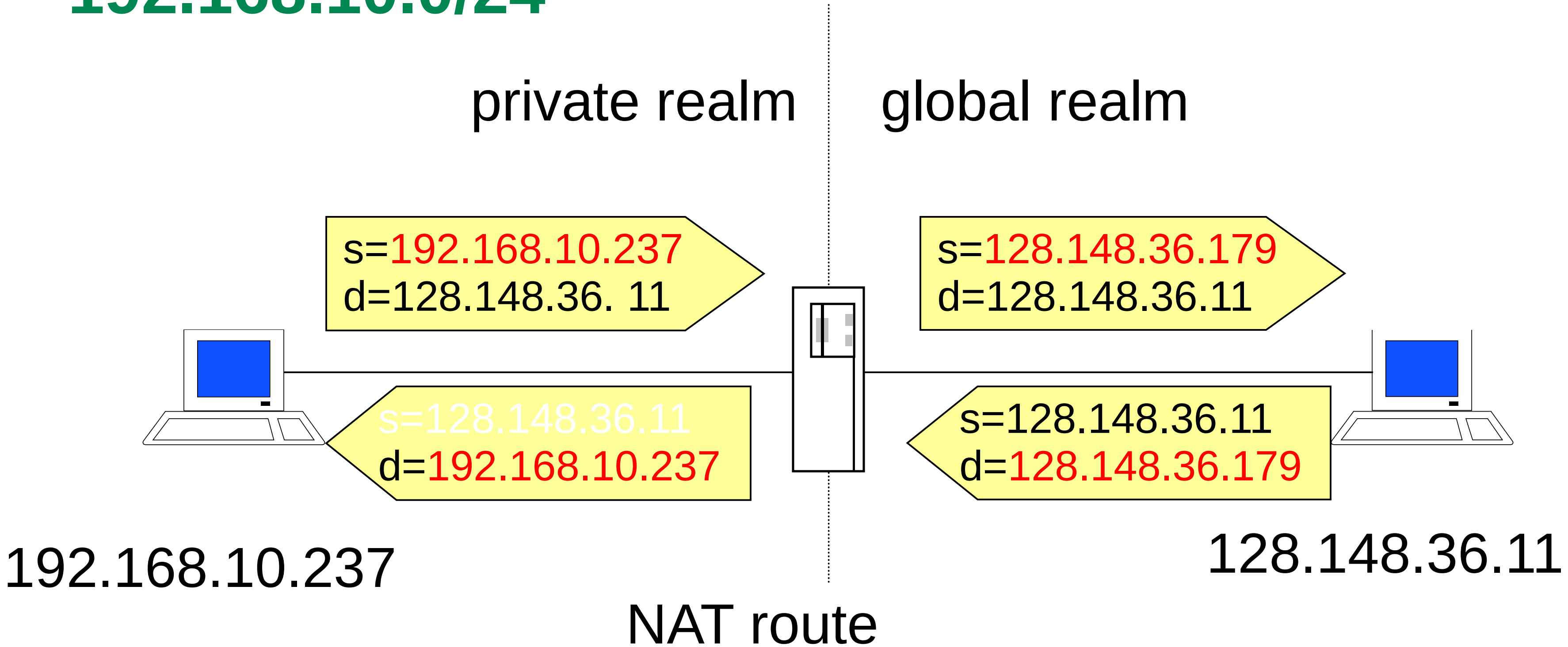
- **UDP is a stateless, unreliable datagram protocol built on top of IP, that is it lies on level 4**
- **It does not provide delivery guarantees, or acknowledgments, but is significantly faster**
- **Can however distinguish data for multiple concurrent applications on a single host.**
- **A lack of 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