TSAM Notes

liljag18

November 2020

Will be on the test: the practical reality of:
TCP: stream protocol
UDP: unreliable datagram protocol.

IETF - Internet Engineering Task Force
IAB - Internet Architecture Board
ISOC - Internet Society
RFC - Request for Comment
ISO - International Organization for Standardization
TCP - Transmission Control Protocol
UDP - User Datagram Protocol
AM - Amplitude Modulation
FM - Frequency Modulation
SNR - Signal to Noise ratio
CSMA - Carrier Sense Multiple Access
PDU - Protocol Data Unit
ATM - Asynchronous Transfer Mode
SYN - synchronous idle
STX - Start Text
ETX - End Text
EOT - End of Transmission
CRC - cyclic redundancy check
ACK - acknowledgment
NACK - negative acknowledgment
PPP - Point to Point
ISP - Internet Service Provider
MAP - Medium Access Protocols
CSMA - Carrier Sense Multiple Access
ROM - Read Only Memory
OUI - Organizationally Unique Identifier
ARP Address Resolution Protocol
MAC Medium Access Control
NIC Network Interface Controller
TDM Time Division Multiplexing
FDMA Frequency Division Multiple Access

LAN Local Area Network
WAN Wide Area Network
FDDI Fiber Distributed Data Interface
RTT - Round Trip Times
QOS - Quality of Service
IHL - Header Length
DSCP - Diff Serv Code Point
NAT - Network Address Translation
NP - Nondeterministic Polynomial
TURN - Traversal Using Relays around NAT
ICE - Interactive Connectivity Establishment
STUN - Session Traversal Utilities for NAT
ICMP - Internet Control Message Protocol
HID - Human Interface Device Profile
SPP: Serial Port Profile
L2CAP: Logical link control and adaption protocol
RFCOMM: Radio frequency communication (RS-232 emulation)
BNEP: Bluetooth network encapsulation protocol (personal area networking PAN), bound to L2CAP
TCP: Telephony Control Protocol
AVCTP: Audio/video data transport protocol - audio/video
Host Controller Interface (HCI)

1 Lecture 1: Intro and Logistics

- **ping** - evaluate a machine's availability/accessibility
- **ifconfig/ipconfig/ip** - find ip address
- **nslookup/dig/host** - DNS lookup
- **ip route show/ netstat -r** - display host's local IP forwarding table
- **netcat (nc)** - create TCP or UDP connections and send lines of text back and forth
- **Wireshark** - packet capture + analysis tool

1.1 1.15 IETF and OSI

- **IETF** - Internet Engineering Task Force
- **IAB** - Internet Architecture Board, **ISOC** - Internet Society
- Request for Comment (**RFC**) documents - contain all formal internet standards

- **ISO** - International Organization for Standardization
- TCP implementation should follow a general principle of robustness: conservative in what you do, liberal in what you accept from others
  - Liberal in what you accept, conservative in what you send

- **Broadcast network** - synchronization (everyone guaranteed to get some information)
- network theory based on static networks
- real networks depend on time which is hard to represent in theories
- Real time "t"
  - **Communication latency** - time to send and deliver a message
  - **Computation latency** - time to process a message
- **Buffers**
  - messages have to be repeatedly stored and re-transmitted
- Practical Development + Information Theory + Mathematical Graph Theory
- Data systems: **Packet switch** (like the postal service)
  - Break up the system - don't know exactly where the packet is going, just how to get it there
  - Better than a circuit switch (used in phones), makes transferring packets more adaptable
- **TCP/IP model** - more common
  - Application - Transport - Internet - Link
  - Networks built on networks, layers may repeat
- High level application needed to know about the physical layer
- **Computer communication**
  - Need sender and receiver
  - Sender needs to know where receiver is (IP + port for internet)
  - Receiver has to be able to accept incoming connections
  - IP + port wrapped up into sockets

1.2 Client

- Initiates connection to server
- Interfaces directly to the user
- Communicates with the server
- 1:1 (N:1)(C:S)

1.3 Server

- Waits for connections from the client
- Provides services to the client
- Communicates with the client
- 1:N

1.4 Peer-to-peer

- Does both (N:N)
- Some or all peers take on server role
- Organized topologies tend to emerge at scale
- Software needs to be both Client and Server

- **Hierarchical Topology**
  - client1 - server - client2
- **port** - number of the house, **IP address**- street address

2 Lecture 2: Network Programming

- **Topologies**
  - Strictly Hierarchical
  - Full mesh - limited scaling
  - Partial mesh - scalable with restrictions
  - Grow - scaling
- Real Time

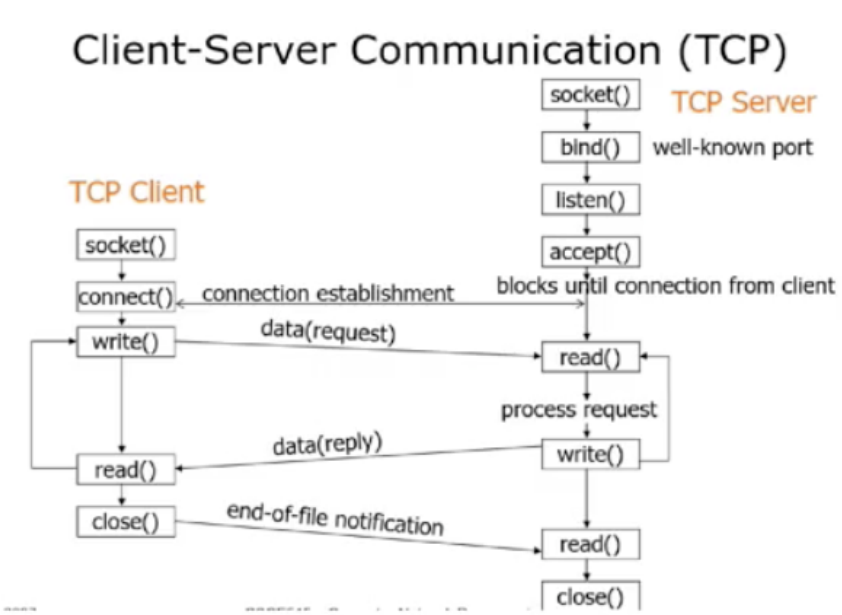


Figure 1: Client Server communication

- Operates in a loop
- Debugging - **Heizenbugs** - timing has changed due to some code (including simple print statements) making bugs appear and disappear
- Dump logs instead over **UDP** (User Datagram Protocol) to another computer or use Wireshark

2.1 Statistical Multiplexing (Sharing)

- Servers can only receive one message at a time, computers just do this fast
- **Stateless:** state contained in the message
- **Statefull:** Client and Server maintain local information
- Finite state machines - alright to use for simple things, not complex

- usr/bin - home for most standard operating systems

- usr/local -> man -k socket — less -> man socket -> man 4 inet / man 4 inet6
- usr/include - locate socket.h

2.2 Non-Blocking Socket

- Server listening for data from the other end
- How long should they wait?
- Blocking (TCP default)
  - Server waits until it receives at least one byte
- in non blocking the server checks for data continuously but does not stop
- Sends error if read() -> buffer is empty or if send() -> buffer is full

3 Lecture 3: Physical Layer

- **OSI Layers:** Applications - Presentation - Session - Transport - Network - Data Link - Physical
- We use **TCP/IP** - layers can't know what's going on underneath them so in theory they don't care (especially internet and application layers) - don't have to change the layer above if you change the protocol in the layer below
  - Layers -> Protocols
  - Application -> HTTP SMTP RTP DNS
  - Transport -> TCP UDP
  - Internet -> IP ICMP
  - Link -> DSL SONET 802.11 Ethernet
- **Information Theory**
  - How you quantify, store, and communicate information
  - Need to consider amount of **UNIQUE** information
  - Delicate balance between total quantity of info and how much is actually shared
  - Television and radio have a low amount of information being transmitted since it is all the same

- Internet - people can break up into smaller groups
- You can never have all the things that you want at the same time due to restrictions
- How do we most efficiently transmit and receive information? - Encoding and decoding, detecting correct transmissions errors

3.1 Long Distance Communication

- Electronics use modulation to send a signal over wire and radio
  - **Amplitude Modulation (AM)** - change amplitude (size of signal goes up and down)
  - **Frequency Modulation (FM)** - used for longer distance communication and changes the frequency (size of signal goes side to side), can transmit very fast, used in fiber optics
- Over distance signal becomes distorted
  - Signals get quieter over longer distances
  - Noise is introduced by transmission/reception, need to reproduce the signal and have many switches to clean up the message and get rid of noise
  - Amplification amplifies noise until noise dominates
  - Keep in mind the **Signal to Noise ratio (SNR)**

3.2 Nyquist Limit

- Applies generally to any oscillating signal/system
- Must have 2 full cycles to be able to accurately reconstruct - aim to have 3 or 4 for practicality
- Range of human hearing is 20HZ to 20kHz
- CD's are sampled at 44100 samples per second
- Nyquist rate - upper bound on thee symbol rate across a bandwidth limited channel

3.3 Fisher Consensus

"Impossible to guarantee that any synchronously connected set of nodes (computers) can ever agree on even a single bit value."

- Can't solve it, have to either work around it, or live with it
- Synchronize - i.e. have a single point of failure which can easily get overloaded which negates what you were trying to do by distributing the program (to avoid such overloading)
- Live with it, catch errors when they happen
- For many applications, as long as you're aware of Fisher, it's not an issue. Note the tendency to drive distributed programmers mad... like gophers

3.4 Key Concepts

- The unit of information is a bit
- Information is a measure of different data
  - Send to 100 people the same message = 1 piece of information
  - Send 1 person 100 messages = 100 pieces of information
  - Broadcast TV/Radio - good because everybody listening is synchronized, bad because huge drop in the potential information shared
  - Synchronizing, in networking terms, is expensive - costs information capacity, often compute time (processes/machines have to wait for everything to report in)
- Bandwidth is either:
  - **Signal Processing** - the difference between the highest and lowest frequency in a signal (Hertz)
  - **Data Communications** - Maximum rate of data transfer across a given path (bits/sec) - bits in data communication, bytes in data storage, careful about that

- **Shannon Limit on Information in a single channel**

$$C = B * \log_2(1 + S/N)$$

  - C = Maximum Channel Capacity

  - B = Bandwidth in Hz
  - S = Signal power over bandwidth (Watts)
  - N = average power of the noise
- More errors on channel the lower the bandwidth - fiber optic communication better because it is almost always error free unlike copper and WiFi
- **Store and forward** - allowing more information to move forward and spread the information to a wider audience (it's caching on the internet, kinda cheating but allowed), backbone of the internet
- In movie streaming you aren't streaming straight from the source in America but rather from some place closer that has it on hand
- Caching locally prevents the internet from needing to resend the package which saves time
- Web browser keeps a local copy of the pages you have visited in a cache and just uses that copy when you go back
- Cannot transmit endless amounts of information

3.5 Physical Layer

- Copper drops - high error rate, cables sometimes longer than they should be, packets get dropped and need to be resent, bandwidth drops as a result
- Fiber optic cables made of glass and glass breaks, be careful
- Wireless blocks - only sends a message to one at a time, just really fast, in a malfunction many packets are dropped, broadcast protocol so limited amount of people allowed to listen
- Satellites have really long delays
- "Last mile" problem - physical cable to your house or whatever is the majority of the cost in a network

3.6 Copper

- Dominated early networking - dedicated links for data communication or carried over phone lines (modems)
- Relatively low capacity, lots of errors
- Network was the bottleneck
- Network transmission protocols favoured computational solutions

over transmission solutions (error detection and correction over retransmission), everything possible was done to preserve bandwidth, and this heritage still influences protocols

- Copper still widely in use in other countries
- Issues:
  - Cross talk - electrical interference between two copper cables close together without casing
  - Limited length (100 m max)
  - Requires repeaters
  - High error rate
  - Cheaper

Fiber (practically error free) and wireless (high error rates but wider reach) are more broadly used in countries like Iceland that have developed further when it comes to technology. Therefore we cant remove all error catching.

3.7 Fiber-Optic Networking

- If we were designing the lower level protocol stacks today, with only fibre in mind, we would do it differently
- Error rate went down
- Much faster
- Longer range
- Advantages: thinner, higher capacity, less signal degradation with distance, fewer repeaters, lower power
- Disadvantages:
- Most extraordinary increase in speed and efficiency in human history (greater than the increase in computational power in computers)
- Network is no longer the overhead, less bottleneck
- 1 second to transmit 10 Gb from Barcelona to New York City (size of 32 volumes of Encyclopedia Britannica

- NEVER LOOK AT OR TOUCH THE END OF THE FIBER-OPTIC CABLE - has a lazer, shine against a wall to see if it is switched off
- Most of the fiber-optic cable you can see is the wrapping protecting the core which is made of glass
- High infrared wavelength
- Has a delay, not instantaneous
- Light moving through a glass core, speed depends on refractive index of glass core and wavelength of light being used
- $Speedoflightincable(latency) = \frac{Speedoflight}{Refractiveindexofcable}$
- Developments - Hollow core fibers (may not be practical but can get close to the speed of light) and multiple wavelengths on one wire, more colors, can send more
- Connect right colors together, if it is difficult then it is wrong

Advantages	Disadvantages
Thinner	Ends need regular inspection and cleaning (need training)
Higher capacity	Not easy to join together (need to splice together)
Less signal degradation with distance	More expensive
Fewer repeaters	Fragile (don't bend too much)
Lower power	Copper is more available

3.8 Wireless

- Different protocols in the physical and data link layers than copper and fiber use
- General principles for layered design, possible to plug and play protocols between layers
- Ability to adapt to widely varying behaviors
- Carries its own physical layer protocols - **Carrier Sense Multiple Access (CSMA)**
- Broadcast medium
  - Send same information to all nodes, lower info capacity than point-to-point network
  - Nothing stops a broadcast being performed on a point-to-point network
  - Disadvantages: High loss due to interference, frequency use has to be carefully regulated, higher delays + jitter, lower security (more encrypted now)

3.9 Time to transfer 100MB file over 10Mbps link

Sample Problem

- UNITS: 10 MBps (megaBYTES -> disk drive in bytes) == 80 Mbps (megaBITS -> network in bits)
- Network overhead - packet sizes etc (read Q carefully to see what assumptions you can make)
- Assume no network overhead in this problem
- 100MB -> 800 Mb, at 10Mbps it takes 80 seconds
- 10 GB -> 80 Gb -> 80.000 Mb, at 10Mbps it takes 8.000 seconds or roughly 2 hours and 16 minutes

Metric Units

Exp.	Explicit	Prefix
10^3	1,000	Kilo
10^6	1,000,000	Mega
10^9	1,000,000,000	Giga
10^{12}	1,000,000,000,000	Tera
10^{15}	1,000,000,000,000,000	Peta
10^{18}	1,000,000,000,000,000,000	Exa
10^{21}	1,000,000,000,000,000,000,000	Zetta
10^{24}	1,000,000,000,000,000,000,000,000	Yotta

4 Lecture 4: Datalink Layer

4.1 Protocols and Layers

- Each layer tries to pretend the layer below isn't there, works most of the time
- Software pretends some layers aren't there and gets a virtual link between the two (Layer 3)
- Upper layers will recover
  - Protocol stack builds upwards in complexity and services offered
  - Reliability, data order -> higher level services
  - Upper layers should not be aware of lower level details, interface between layers similar to a contract, lower layers provide specified services to upper layer
  - Lower layer in particular should be plug and play -> different datalink protocols depending on medium (copper, wireless, fiber, etc)
  - Humans are the top layer
- Design choices (Tradeoffs, can't have both)

  - Fixed packet size vs variable packet sizes
  - Synchronized (send packets at fixed times) vs unsynchronized
  - Error correction (extra info needed in packet) vs retransmission of packets
  - Acknowledge packet vs don't acknowledge
  - Connectionless (message sent without prior arrangement) vs connection oriented (connection setup and then message sent like with a phone)
- Three variants
  - Ethernet -> unacknowledged connectionless service (usually only locally and with a wire)
  - WiFi -> Acknowledged connectionless service
  - TCP -> Acknowledged connection-oriented service
- **Protocol Data Unit (PDU)** - a message from one side to another
  - Physical layer -> Bit
  - Data link Layer -> Frame
  - Network Layer -> Packet
  - Transport Layer -> Segment (TCP) Datagram (UDP)

4.2 Physical Signaling Sublayer

- Data link functions
  - Provide well-defined interface to network layer
  - Deal with transmission errors
  - Regulate flow of data so that slow receivers aren't swamped by fast senders

4.2.1 Interface to Network Layer

- Perform character encoding, transmission, reception, and decoding
- Determine start and end of packets (framing)
- Simple transmission error handling
- Error recovery (may be delegated to or repeated by a higher level, datalink layer not required to guarantee data integrity, "Upper Layer will recover" or die trying)

4.2.2 Synchronous vs Unsynchronyzed Communication

- Recurring problem/solution approach to datacomms
- Synchronous
  - Synchronized by some kind of external clock
  - Receiver expects frames to arrive at fixed frequency
  - Uses clock to read frames from the wire
  - Problems include drift, limit to actually being able to be instantaneous, Einstein and Fisher problems
  - Good for local communication
- Asynchronous Communication
  - Not synchronized
  - Frames have to have start/end frame markers or known frame length (1 start bit, 8 data bits, 1 stop bit)
  - Incurs a high overhead
  - Might have trouble finding start and stop markers, can see data just can't read it

4.2.3 How to find the frame on receiver end - Solutions

- Transmit length of frame
  - Header field includes length of the frame
  - Header is of fixed length (in turn restricts the length of length field and in turn restricts the packet size)
  - Difficult to recover from errors in length count
  - How do you find the next header?
- Fixed frame length
  - Can make a software's life easier but not necessarily efficient
  - Fragmentation occurs if message is longer than the frame -> no optimal frame length, inefficient, however queuing problems are easier and reduces jitter (deviation from expected reception time)
  - **ATM (Asynchronous Transfer Mode)** cell size was 53 bytes

- Flag bytes with byte stuffing
  - Character based frame signalling
  - Main framing method 1960-1975
  - SYN synchronous idle
  - STX Start Text
  - ETX End Text
  - EOT End of Transmission
  - CRC cyclic redundancy check
  - Used to flag communication so cannot appear in data
  - If flag byte occurs in data -> use byte stuffing -> If sender sees a FLAG in data inject an ESC byte, receiver sees only one FLAG (genuine end of frame), otherwise remove duplicate (Same for ESC, just add another ESC)
  - FLAG errors: if FLAG and ESC bytes get corrupted (all framing techniques are sensitive to error) then may either detect premature end of frame, entire frame is lost, Checksum will fail at receiver, find next frame by looking for FLAG, need to detect hole in data and either recover the lost frame or get it resent
- Flag bits with bit stuffing
  - Bit oriented framing
  - Byte stuffing imposes a particular character format
  - Bit stuffing just uses a pattern of bits to signal start/end of frame (6 bits in a row 01111110)
  - Constant flags or 1's considered idle
  - Inserts 0 after any 5 bits in data, receiver similarly removes to avoid misinterpret
  - Advantage: Can vary length of flag, longer flag will reduce need for stuffing, short packets use a short flag -> less overhead, can combine with other techniques for efficiency/safety

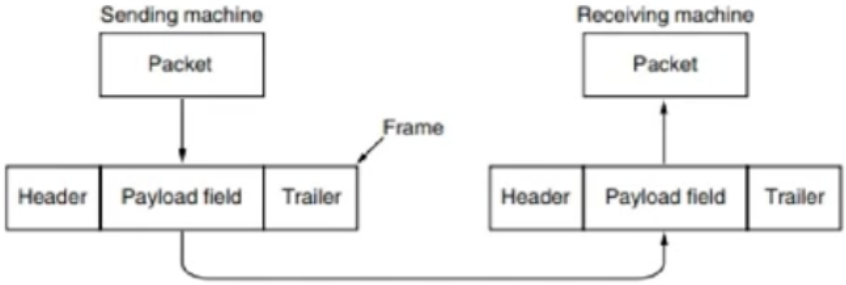


Figure 3-1. Relationship between packets and frames.

4.3 Error Detection and Correction

- Single Bit Errors -> One bit changed by excited particle or something
- Errors rarely evenly distributed in data even though math depends on this
- Burt Errors -> on WiFi, sequence of errors, lot of errors together

4.3.1 Error detection and correction

- Parity Bit/Check Bit
  - Add parity bit to the frame
  - Count number of bits in a frame (1 bits ODD -> parity bit 1, 1 bits EVEN -> parity bit 0, Number of 1 bits always even)
  - Original ASCII was 7 bit characters, 8th bit was parity check bit -> later dropped
- Error correction
  - Add redundant info to frame, parity bit not enough
  - Allows errors to be detected and corrected
  - More info == longer frame, increase probability of error, transmission cost, and cost of processing at receiver, but avoids cost of retransmitting the frame
- Hamming codes -> Sending multiple slightly wrong versions of the same message -> add extra parity bits to the power of 2 ELEPHANT
- Increases size of message and proportion of redundancy

- Detect n bits of error, correct n-1 bits
- Can never guarantee all errors will be detected, can guarantee lower bound, cost of all this is significant overhead
- Cycle Redundancy Check
  - More complex form of error detection
  - Based on theory of cyclic error-correcting codes
  - Takes advantage of known additional information at each end
  - Divide string(frame) by generator polynomial G(x), add remainder R(x) to end of frame. Receiver divides frame by G(x), no remainder = error
- Receiver sends back acknowledgment frames to sender (ACK and NACK (error in frame))
- Timer on acknowledgment
- Multiple copies of frame -> frame header includes sequence number

- Protocols -> LAN, PPP, respond to requests from the network layer, issues requests to physical layer
- Flow Control - One sending faster than can be received
  - Negotiate fixed speed - used in lower protocol levels
  - Feedback-based - receiver tells sender to pause
- Addressing not necessarily needed at data link layer when physical wire or local, WiFi does need addressing at this layer

4.4 Connecting more than one Machine

- Statistical multiplexing
  - Multiple hosts can share connection -> Provided they don't all try to to all use it simultaneously -> Vodafone does this, not everyone is using the same thing at the same time, traffic is monitored
  - Too much traffic will cause blocking, flow control helps prevent this
  - Relies on being able to queue or delay packets
- Time division multiplexing
  - Each has their own time slot
  - Inefficient, unused time slots

5 Lecture 5: Local Area Networking

5.1 Statistical Multiplexing

- Multiple Access Protocol or Time Division Multiplexing
- Fundamental Feature in Communication Networks
- Assumption: Multiple hosts can share a connection -> provided they don't try to use it simultaneously
- Banks and parking lots make use of this idea
- Allows bandwidth to be divided arbitrarily amongst hosts
- Statistical bc depends on -> number of hosts, amount of traffic, frequency of traffic, capacity of channel

5.1.1 Example: Network Provisioning

Given a set link capacity how many users can be provisioned and maintain performance 'guarantees'? (oversubscription, no absolute answer)

- 100 Gbps link, each customer sold 1Gbps
- 100% capacity = 100 customers
  - 4:1 -> oversubscription would be 400 customers (each customer using on average 25% of their bandwidth
  - 2:1 -> oversubscription would be 200 customers (each customer using on average 50% of their bandwidth, twice as expensive from ISP perspective
  - Issues: bursts, heavy sustained use
  - Commercial (business hour) use, vs household (evening and weekend)
  - Dedicated bandwidth if customers ask for it

- Kurose: an ideal multiple access protocol
  - Decentralized algorithm for how you can all share this broadcast channel, no central coordination that is efficient
  - Assumes nodes arrange to transmit co-operatively and don't block each other
  - Not likely to happen

5.2 Local Area Networking

- Local Area Network
  - Primitive, used in olden days, similar to home network, network printer, fridge, laptops, etc
  - Because they are all connected to one another, they can use broadcast by doing address discovery since they are all on the same network, works until you have more than one printer or the such
- Network Hub
  - Simple connection devices
  - Connects several devices onto one link
  - Broadcasts received packets to all connected devices
  - Bandwidth split between all connected devices
  - Phased out in favor of switches
- Network Switch
  - Smarter than Hub
  - Unmanaged - work out bugs itself, can be controlled and customized
  - Determine which packet goes to which computer, doesn't send it to all like the hub does
  - Handles a LAN
- Routers
  - Forward packets between local networks
  - Connect campus/home network to the internet
- Home Router
  - Connection between home network and internet
  - Network Address translation to internet address
- Backbone Router
  - ISP owns, provides the internet
  - Huge capacity, small boxes

5.3 Medium Access Protocols

MAC (Medium Access Protocols) - control access to single shared broadcast channel, range of protocols and approaches.

- Interference: Two or more hosts broadcast simultaneously
- Collision: Node receives two or more signals at same time

5.3.1 Three classes

- Channel Partitioning
  - Fixed division of channel into pieces
  - Each node allotted exclusive use of specific piece of channel
  - Time Division Multiplexing (TDM), Frequency Division (FDMA)
- Random Access
  - Host uses channel ad hoc
  - Protocol provides recovery mechanisms for collisions
  - Engineered to reduce frequency of broadcast
  - ALOHA, S-ALOHA, CSMA, carrier sensing (for wire), CSMA/CD (ethernet), CSMA/CA (802.11)
- Turn Taking
  - Nodes take turns, but negotiations take time
  - Some form of polling from central controller
  - Token passing
  - Bluetooth, FDDI, token ring
  - host that is allowed to talk has the token, guaranteed share, time limit

Random Access Protocols - Node transmits when it has a packet to send, collision can occur, algorithms to recover from collisions

5.4 ALOHAnet

5.4.1 ALOHAnet

- Additive Links On-line Hawaii Area
- Used to link schools in Hawaii

5.4.2 ALOHA Protocol

- Radio Station would send transmission when it wanted - receiver would send ACK, if not received, collision assumed
- Pick random back off time, then retransmit
- Stations don't check if another station is ready for transmitting -> solved by assigning transmission times and using fixed frame size = slotted ALOHA

- Master clock for time slot synchronization
- Stations must not start a broadcast within 1 frame of each other
- Node chooses random wait time before retransmitting, progressively longer but still random

5.4.3 CSMA: Carrier Sense Multiple Access

- Listen before transmission
- Only send if channel is free
- Reduces collisions a little
- But if more than 1 node is waiting collision is likely when they both start
- wait and listen a further random time before starting

5.4.4 CSMA/CD :: Collision Detection

- CSMA: carrier sensing applied with detection of collisions
- On detecting collision, node aborts -> collision detection easy in wired LAN (compare received/transmitted signal strength), not so much in wireless (Local transmission overwhelms received)
- reduces channel loss

5.4.5 MAC Protocols: Taking controlled turns

Master and slave dynamic, concerns: polling overhead, latency, single point of failure (master)

5.4.6 Token Ring

- Control "token" passed from one node to another sequentially
- Data -> Node waits until it has token then sends data, sends token to next node when done
- No data -> Node immediately sends token to next node
- Issues: overhead of token packets, higher latency, single point of failure (node with token)

5.5 Ethernet II Protocol

5.5.1 Ethernet II Addressing

- Communicating hosts must be able to uniquely identify each other -> Guarantee Uniqueness - Single point of allocation otherwise you get Fisher problem
- Ideally addressing helps with routing
- MAC protocols require identification when on LAN
- As hosts are sharing a medium this must be unique

5.5.2 MAC Addresses

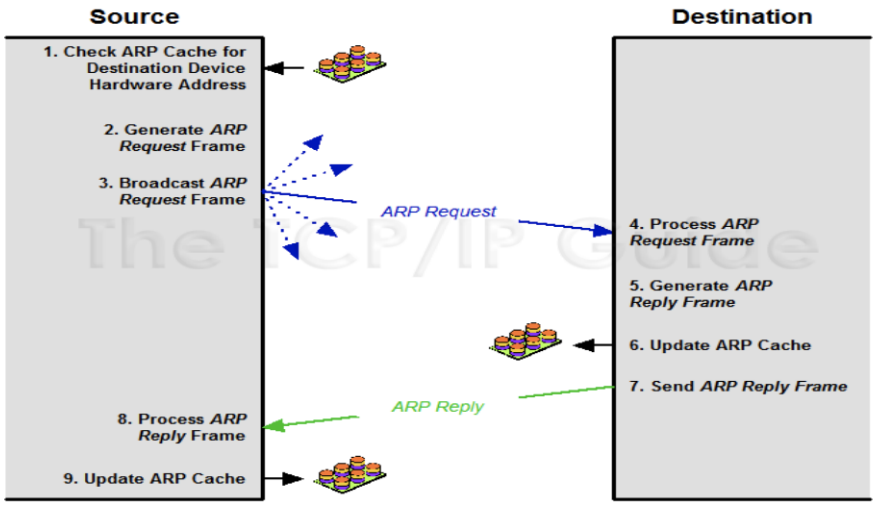
- Media Access Control (MAC)
- Unique Identifier assigned to a Network Interface Controller(NIC) (Hardware address, Ethernet hardware address, Physical address)
- In theory globally unique
- Usually burnt into Read Only Memory (ROM)
- OUI - Organizationally Unique Identifier

Mac ⇒ EUI-64: Original 48-bit address allocation estimated to last until 2080, renamed to Extended Unique Identifier.
First three octets identify manufacturer.

5.5.3 Unicast vs Multicast

Unicast: Frame is intended for just the addressed interface (NIC)
Unicast Flood: Switch (network equipment) forwards frame to all known hosts, used if switch doesn't recognize the address
Multicast: Frame is forwarded to all local hosts - they can ignore it
Broadcast: FF-FF-FF-FF-FF-FF - sent to all hosts for processing

ARP Protocol



5.5.4 Address Resolution Protocol (ARP)

arp -v
ARP Broadcasts: Local hosts and switches will typically cache ARP messages, ARP table has mapping of MAC addresses to IP addresses (dynamically or statically(administrator defines mapping manually) done), broadcast to all hosts used if target host's address not in cache
Reverse ARP (RARP): Client broadcasts request for an IP address
ARP is a necessary protocol to link WAN's and LAN's, regarded as a level 2 and 3 protocol, acts like a bridge

6 Lecture 6: Internetworking

6.1 Scaling and Topology - ELEPHANT

Topology, Latency, and size (number of nodes) affect scaling

Group size: Maximum no. of single hop connections that a given node can support, depends on Bandwidth, Application requirements, CPU, Larger network considerations

Latency:

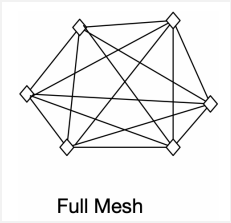
- Communication Latency: time taken to send a message between two nodes
- Processing Latency: time taken to process a message by the recipient

- **Long communication** latencies favour hierarchical networks
- **Short communication** latencies favour distributed ones.
- Communication relies on a send-receive protocol

6.1.1 Topologies

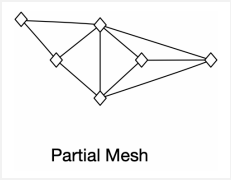
Full mesh

- Information capacity scales as L(L-1)
- Best topology for exchanging information for small groups
- Size is bounded by the group size
- Worst topology for control
- Vulnerable to broadcast storms



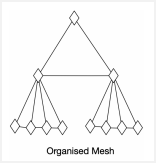
Partial mesh

- Cannot guarantee consensus
- Depending on error rate, voting algorithms can provide strong guarantees
- Vulnerable to broadcast storms with increasing N



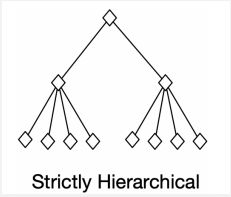
Organized Mesh

- Only topology that can scale to large N and grow capacity
- Similar issues to other mesh topologies



Strictly Hierarchical

- Best topology for control purposes e.g. botnets
- Worst topology for exchanging information
- Robust to broadcast storms with increasing N
- Required if consensus must be guaranteed.



Trade-offs

Hierarchical	Mesh
Relatively Simple	Complicated
Single Point of Control	No single point of control
Synchronization is what it does best	Impossible to guarantee consensus
Fragile - single point of failure	Robust - hard to bring down all nodes
Low information capacity	High information capacity
Cannot scale to large N (nodes)	Scales to large N
Good for high latency communication	Poor performance
Vulnerable to broadcast storms	Very vulnerable to broadcast storms

6.2 Network Services

6.2.1 Guarantees a Network should offer

- Guaranteed Delivery -> all packets sent will eventually arrive at destination, tricky if forced to take same path and that fails
- In order packet delivery -> Packets arrive in the order they are sent, tricky if they are allowed to take different paths
- Guaranteed Delivery within specified time -> Packets will also arrive within a specified time, tricky if there are lots of other packets (congestion)
- Guaranteed Bandwidth -> Sending host is guaranteed a specified bit rate to the destination, inefficient if not fully utilized
- Security -> No eavesdropping or diversion to different hosts

Essentially store and forward packet switching using LAN and WAN

Local Area Networks (LAN)	Wide Area Networks (WAN)
Local to an office, floor, building, campus	WAN's connect LAN's together
Different degrees and different hardware	Allow traffic from one LAN to be sent/received to another
Scaling up requires linking LAN's together	Problem of WAN's is to be able to operate at scale
Conceptually still under same "network" or control	Longer Round Trip Times, different information capacity domain

Longest Religious War Ever in Computer Networking

Underlying Issues both sides are trying to solve are known generally as Quality of Service (QOS)

Circuit Switching	Packet Switching
Connection Oriented	Connectionless
Dedicated circuit between two ends	Packets sent through network on arbitrary routes
Championed by Phone Companies	Championed by DARPA and University Development
Allows packet delivery rate and bandwidth to be guaranteed	No guarantees of delivery, rate, or bandwidth, "Best Effort"

6.2.2 Wide Area Network Architectures

- X.25
  - Extensive error detection and correction
  - Guaranteed delivery in order
- ATM (Telco/PTT)
  - Dedicated circuit between two ends
  - Guaranteed delivery
  - Bounded delay
  - Guaranteed bandwidth
- Internet

- Connectionless
- Best of effort service
- No guarantees on anything

6.2.3 Fundamental Problems

- Addressing - how hosts are identified to each other
- Routing - how packets of data "know" which address to go to
- Fragmentation - how are large packets that are broken up put back together
- Reliability - how errors are handled
- Order - how data is delivered in the order sent
- Time - how long do we allow for end to end delivery before network layer can give up
- Scale - number of nodes network can accommodate and how much traffic they can generate

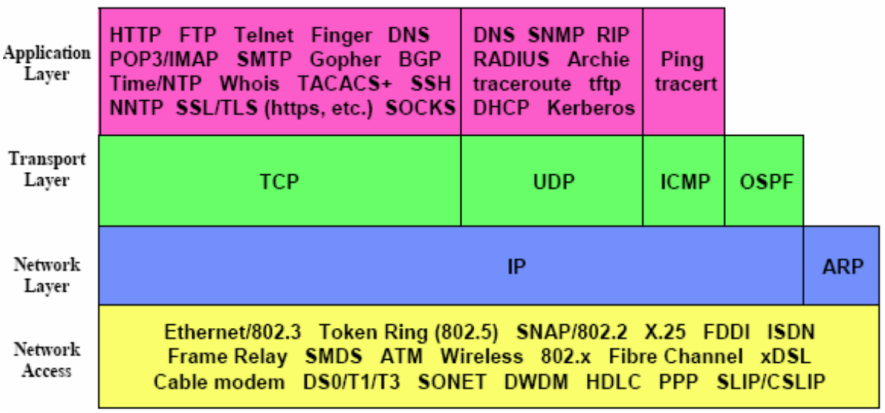
Unreliable - packets of data can be arbitrarily dropped (IP and UDP)
Reliable - all packets are delivered or connection is dropped
Reliable and in order - all packets sent are delivered in the order which they are sent or the connection is dropped (TCP)

6.2.4 RFC 1958: Architectural Principles of the Internet

- Connectivity is its own reward
- End-to-End functions require End-to-End protocols
- Experience trumps theory
- All designs must scale
- Be strict when sending and tolerant in receiving
- Circular dependencies must be avoided
- Modularity is important. Keep things separate whenever possible

- Keep it simple stupid (KISS) - avoid options and parameters wherever possible

6.2.5 Internet Protocol Stack



IP Datagram -> basic network transfer unit, <header>:<payload>, unreliable and in big-endian order (network order, little-endian is Intel order)
12345678
Big endian: 12 34 56 78
Little endian: 78 56 34 12
IP Fields:

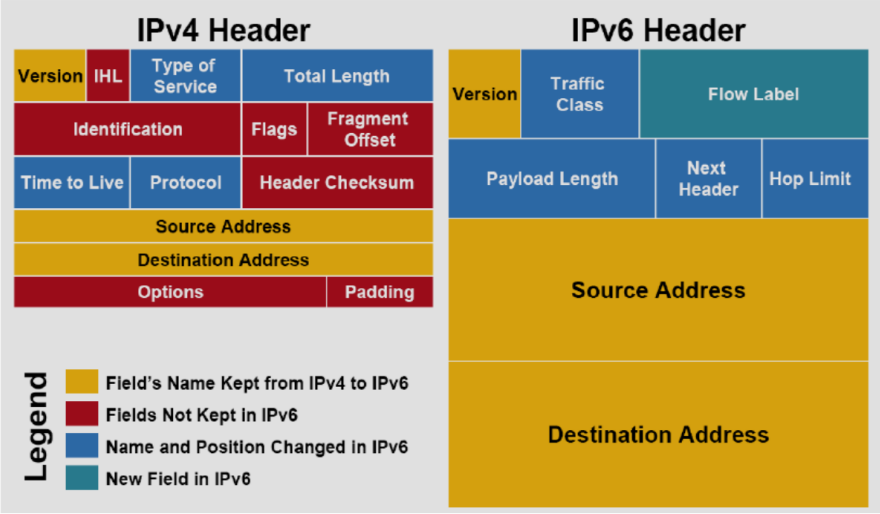
- Version: IPv4 or IPv6
- IHL (Header length): options field in header is not mandatory
- Type of Service: Indicate type of traffic (Diff Serv Code Point (DSCP), has never really been used)
- Total length: length of entire datagram (max 65535 bytes)
- ID: used to identify datagram fragments
- Flags: whether datagram can be fragmented, fragmentation control
- Frag offset (used in fragment reassembly)
- Time to Live: each router will decrement this field by 1
- Protocol: Layer 4 protocol sending IP packet (UDP, TCP, ICMP, etc)
- Header Checksum: on header only
- IP OPtions (debugging (Record route) and research)

Fragmentation:

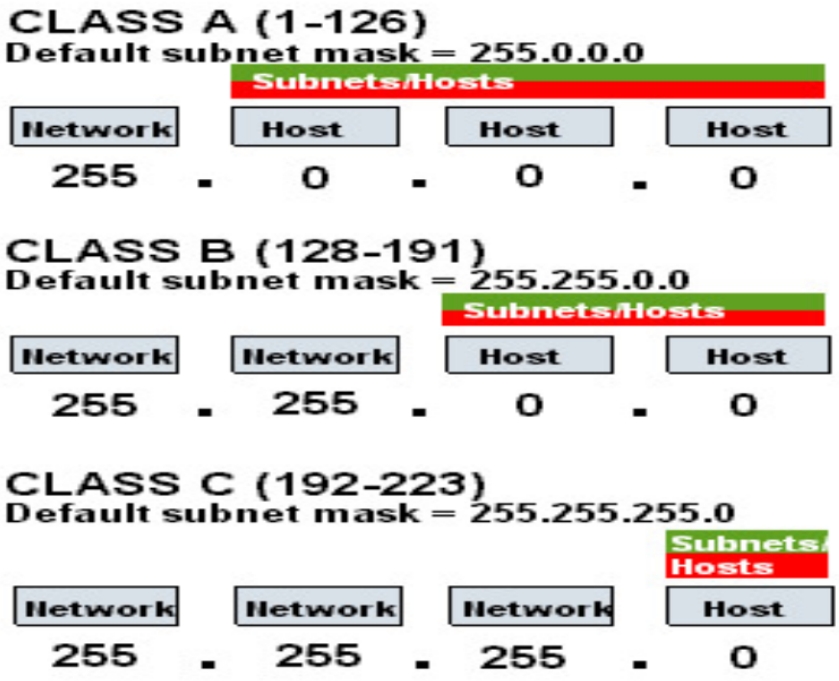
- Within the network there are no guarantees
- Datagrams can be arbitrarily broken up and reassembled
- MTU: Maximum Transmission Unit for link: can be less than IP datagram size, routers must then break datagram into fragments
- IP packets can be lost: need to know offset, sequence, last fragment

6.3 Addressing

IPv4 dominates internet addressing



6.4 IPv4



Subnet Addressing - subnet mask ELEPHANT LOOK AT ASSIGNMENT 2 AGAIN

- Subnetowrks are a logical division of the IP network address space
- use / to provide shorthand reference
- 198.0.1.130/24 -> 24 bits allocated to network prefix, remaining 8 bits are the host address (start counting from the left side)
- Subnet mask 255.255.255.0 -> masks off network part of address to leave host's space

6.4.1 Example: How many addresses in the following ranges?

Answer: $2^{32-mask}$

1. 10.0.0/8 -> $2^{32-8} = 16777216$
2. 192.168.0.0/16 -> $2^{32-16} = 65536$
3. 255.255.255.255/32 -> $2^{32-32} = 1$
4. 224.0.0.0/4 -> $2^{32-32} = 268435456$

Classless Interdomain Routing (CIDR) replaced classful which was talked about above to slow down growth of internet routing tables and slow axhuastion of IPv4 addresses. Network could be assigned on any bit boundary.

Autonomous System (AS) - Entity that controls an Internet Routing Policy

NAT: Network Address Translation

- Having a class A or B address allows all local hosts to have an internet presence
- Problem: Not enough IP addresses for everybody, not everybody wants one either for security reasons
- NAT remaps one IP Address space to another: rewrites source IP address on outgoing traffic, remembers mapping of IP's network behind it, can "hide" an entire network behind one IP address
- Impacts incoming connectivity: NAT device can easily track outgoing traffic and remap, much more difficult to match incoming traffic to destination
- Limited by port range (16 bits approx 60000 connections)

7 Lecture 7: The Internet

7.1 Subnetting (redux)

Subnetting notation is used to describe the network, in particular for routing (later). It is a way of collapsing the size of the routing tables needed to direct packets through the internet. Subnets are not used in the actual addressing of packets.

7.1.1 Addressing

- Addressing and Routing are reated NP (nondeterministic polynomial) complete problems
- Addressing: How do addresses get assigned, is there some kind of central allocation, and if not how are they guaranteed unique
- Routing: Is there some kind of central route finder (no that's worse than NP complete), if not how is routing responsibility distributed
- Both problems operate within the FIsher consensur issues
- Delegate and Distribute authority

7.1.2 Post Office Addressing

- The address encodes the route
- New addresses are added at the lowest part of the address possible
- Works similar to a post office

7.1.3 Key concepts

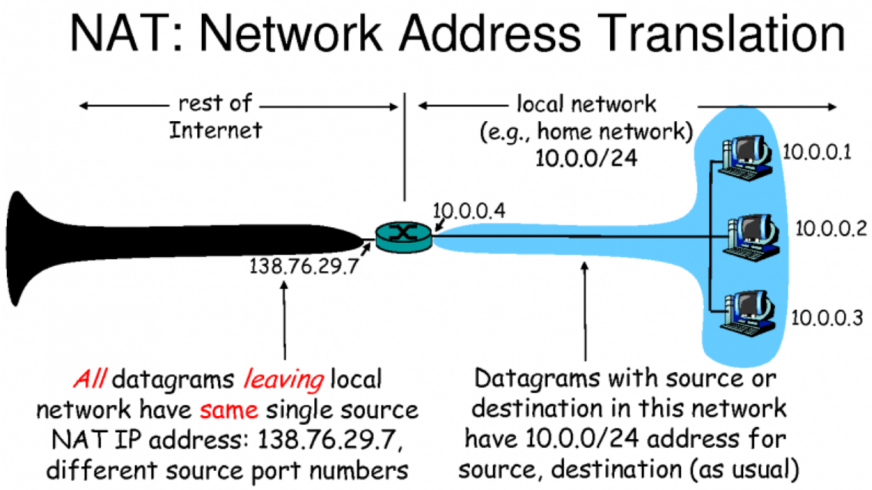
- Address allocation is decentralized
- Each country/city/street can allocate its own addresses independently
- Easy to insert Countries/Cities/Streets so long as their name is unique at their level
- Each delivery hub only needs to know how to send to the next hub
- IP addressing works exactly the same way
- MAC address is similar - delegated to manufactures within their range - LAN's then rely on broadcast messages to fill in address tables

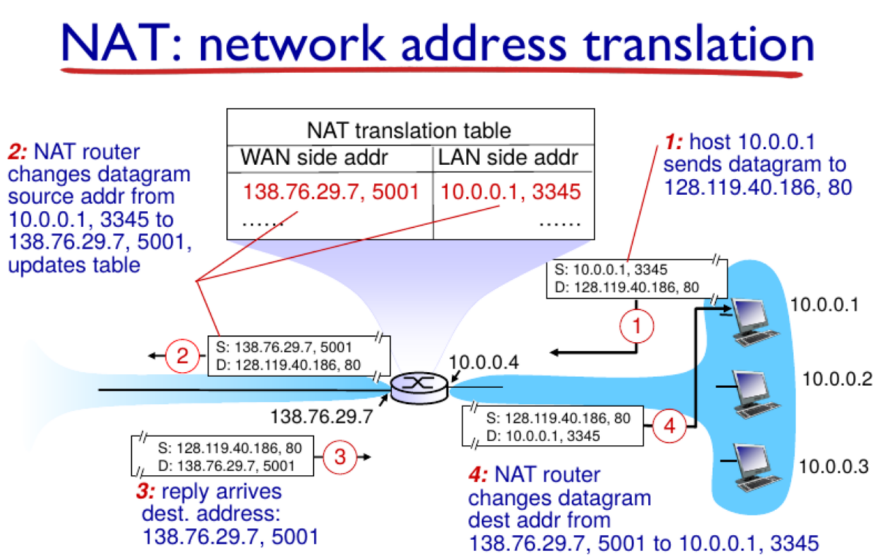
7.2 Network Address Translation

This is the reason why the internet has been able to scale past the IPv4 network address range - you can have lots of addresses behind one IPv4 address with translation. Many devices can be connected and have their own ip address but then the router will assign them all one public IP address.

Creates problems for

- Any kind of peer to peer application - generally isn't possible to directly connect to a server behind a NAT
- Anybody who wants to host an Internet application on their home network - ISP agreements may also prevent this due to bandwidth restrictions
- Distributed applications (Multi-user games, phone/chat applications)
- This was not the design intent for the internet





Kurose and Ross: Network Layer

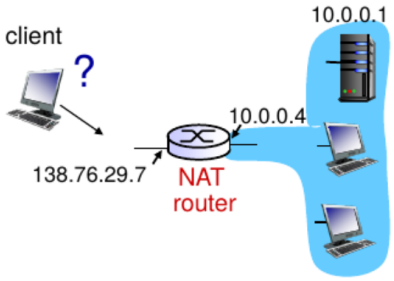
7.2.1 NAT Traversal Techniques

- NAT violates principle that addressing is no longer end to end - issue is incoming connections, which computer is their destination
- Number of ways to solve this:
  - TURN: Traversal Using Relays around NAT
  - NAT hole punching

  - STUN: Session Traversal Utilities for NAT - package of methods to circumvent NAT
  - ICE: Interactive Connectivity Establishment
  - UPnP: Protocol to allow requests to router to open port
  - NAT-PMP, now PCP
- Note: NAT Traversal techniques often bypass security policies

NAT traversal problem

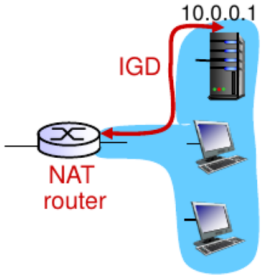
- client wants to connect to server with address 10.0.0.1
  - server address 10.0.0.1 local to LAN (client can't use it as destination addr)
  - only one externally visible NATed address: 138.76.29.7
- *solution 1:* statically configure NAT to forward incoming connection requests at given port to server
  - e.g., (123.76.29.7, port 2500) always forwarded to 10.0.0.1 port 25000

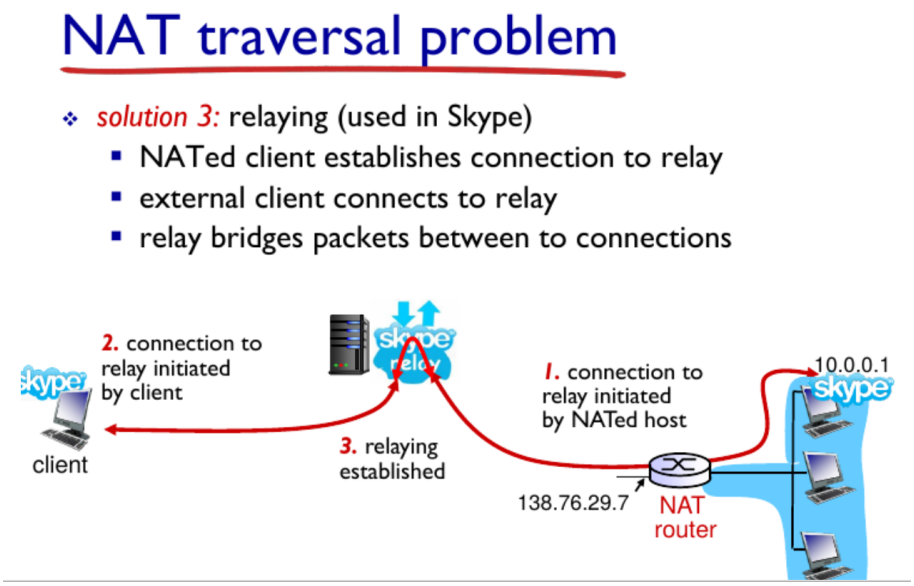


NAT traversal problem

- *solution 2:* Universal Plug and Play (UPnP) Internet Gateway Device (IGD) Protocol. Allows NATed host to:
  - learn public IP address (138.76.29.7)
  - add/remove port mappings (with lease times)

  i.e., automate static NAT port map configuration





Hole Punching

- Behavior of many NAT boxes and their port allocation schemes are fairly predictable
- Assuming two hosts know each other's NAT IP address - can try and guess the port being used by the NAT, somewhat dubious technique
- TCP hole punching - both hosts reuse the same local endpoint to try to connect simultaneously, violates TCP standard, relies on SYN packet getting through on one side

7.3 IPv6

- Simplified header format: fixed length 40 bytes, no fragmentation allowed, no checksum, next header - upper layer protocol being carried or Options
- ICMPv6 - additional messages, "Packet Too Big"
- Backwards compatibility: Tunneling, IPv4 routers carry IPv6 as a payload in an IPv4 datagram

IPv6 Addressing

IPv6 Address Representation

- 128 bits in length and written as a string of hexadecimal values
- In IPv6, 4 bits represents a single hexadecimal digit, 32 hexadecimal values = IPv6 address

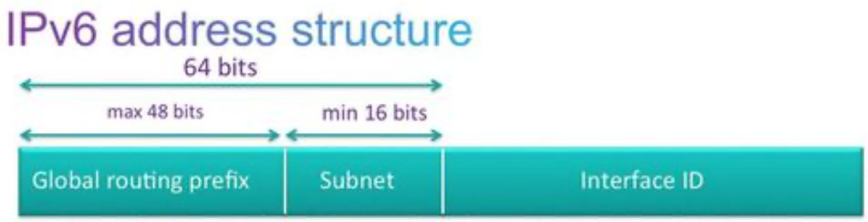
```
2001:0DB8:0000:1111:0000:0000:0000:0200
FE80:0000:0000:0000:0123:4567:89AB:CDEF
```

- Hextet used to refer to a segment of 16 bits or four hexadecimals
- Can be written in either lowercase or uppercase

Rules for displaying

Rule 1: Omit leading 0's

Rule 2: Omit all 0 segments - double colon can replace a series of 1+ 0000 segments but only once



- The RIRs get /12
- The ISPs get /32
- Organisations get /48 from the ISP
- The next 16 bits can be subnetted to obtain a maximum of 2^16 different subnets
- The last 64 bits are used for the host portion

7.3.1 Consequences

- The original decentralized network design was biased towards central control
- NAT is necessary to allow computers to access the internet

- doesn't allow them to readily access each other - peer to peer communication is hard, need a central point to coordinate
- client-server applications, heavily hierarchical - controlled by a central party
- FB chat, Reddit, Skype, SMS, Google, etc
- Peer to peer applications possible but operate at a disadvantage

7.4 Error Signaling: ICMP

7.4.1 ICMP: Internet Control Message Protocol

- Used to send error messages
- Used by traceroute and ping
- Starts after the IP header
- Maximum length is 576 bytes
- Data contains the IP header of the packer that caused the error message plus at least the first 8 bytes of its data
- Has been used to create covert communication channels (ICMP tunnels)

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo and echo reply	Check if a machine is alive
Timestamp request/reply	Same as Echo, but with timestamp
Router advertisement/solicitation	Find a nearby router

7.5 Packet Programming

Useful commands: "locate <*file*>" and "find .-name *file* -print

man 7 udp -> Gives Linux Programmer's Manual

Order of eval -> +-, <<>>, & (Bitwise AND), — (Bitwise OR), ∧ (Bitwise exclusive OR)

7.5.1 Potential Pitfalls

- Target of cast is a pointer
- Casting must have exactly the right lengths

- Debugging -> print out actual bytes (printf("%x", buffer[0]) or use Wireshark to cross check it was actually send
- buffer has space allocated
- Correct length of buffer is being passed to recvfrom

More stuff about project on the slide

8 Lecture 8: TCP/IP

8.1 Introduction

8.1.1 TCP/IP

- Implemented using IP datagram packets
- Two way connection: Sender - Receiver
- Connection required
- Guarantees to application: reliabe, in-order packet delivery, or else **your connection gets dropped**
- By convention the client initiates teh connection, server receives

Generally more efficient to send a stream of segments and not wait for each one ot be individually acknowledged before sending the next. It's really a question though of where to have the buffers and how to manage them. There will always need to be some kind of buffer somewhere.

8.1.2 Buffering at Endpoints

- Kernel/OS is responsible for segment reassemby (TCP layer)
  - NIC is responsible for handling Ethernet frames correctly
  - Allows TCP/IP to run over different low level protocols at each end
  - Specialized NIC's are starting to become availabe that will do this
- Traffic stream reassembled in correct order
- Out of order packets arriving too early queued
- Lost/Delayed packets requested for retransmission

Sliding Window Protocols

- Sender keeps a buffer with sent segments

- Receiver keeps a similar buffer
- Sender doesn't delete the segment it is sent until it receives acknowledgment from the receiver
- When it does, it advances its pointer in the window to the next unacknowledged segment

Reassembly

- Reassembly begins with each segment having a sequence number - hosts know the order segments were sent in
- Host and Client also use timeouts - give up on waiting for segments
- Timeout too short: premature timeout and unnecessary retransmission
- Timout too long: slow reaction to lost packets
- Depends on Round Trip Time and on the medium being used for transmission and how far away the hosts are
- Implies Host and Client have to measuer RTT

TCP/IP originally used as a file transfer protocol because connection is seen by application as a single stream of data. Socket communication is two way.

S.listen() for connections -> C.connect() -> S.accept() connection -> send file-C.send() -> Receive data -> connection C.close() -> accept C.close() or can S.close() at S end.

8.2 How does it work?

Segment is the name for packet at the TCP level because to the user a TCP connection is a single stream of data.

8.2.1 TCP Flags

- URG - Urgent data, prioritize
- ACK - Acknowledgement
- PSH - Send data to application immediately
- RST - Reset
- SYN - Establish connection (first packet)
- FIN - Terminate connection
- ECN - Echo (SYN flag = 1 - peer is ECN capable, else if SYN flag = 0 - Network congestion notification)
- CWR - Congestion Window Reduced
- NS - ECN-nonce

8.2.2 TCP Options

- Maximum Segment Size (MSS)
- Window Scale
- SACK (Selective Acknowledgement) Permitted
- SACK (Selective Acknowledgement)
- Timestamps
- Often used experimentally

8.2.3 TCP Segment Numbers

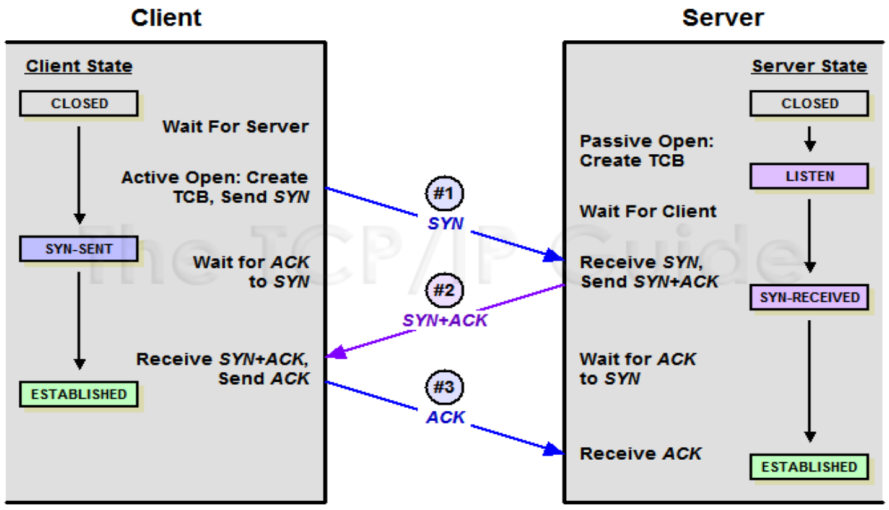
- Each TCP segment carries a sequence number and acknowledgement number
- TCP segments are carried using IP: unreliable, connectionless
- Withing network TCP segments are invisible (in theory at least)
- Server and Client hosts have to handle consequences of segments - arriving out of order, being delayed, being dropped
- Maintain local state to do so

8.2.4 How do they do this?

- Sequence number: number of byte in stream of first byte in segments' data
- Acknowledgement: sequence number of next byte expected
- Both sides maintain separate counters for sent and received traffic
- ISN: Initial Sequence Number
- Wireshark displays relative sequence number - rebased to 0

No length field in TCP header, length is left as implementation dependent by TCP standard, consequently theoretical limit on TCP packet size is IP length

8.3 TCP Protocol Connection Establishment



8.3.1 Denial of Service: SYN Flooding

- Denial of Service attack - objective is to take down hosts
- Operates by causing servers to use up their resources, in particular memory
- Server has to remember state of connecting clients - each connection consumes a certain amount of memory, server will eventually timeout
- Typically organized using botnets - Distributed Denial of Service (DDoS)

8.3.2 Countermeasures

- Increase TCP Backlog
- Reduce the SYN-RECEIVED timer
- SYN Caches - reduce size of state stored by server
- SYN Cookies - reduce state storage to 0
  - Moves state of connection into Acknowledgment field
  - Transmission Control Block (TCB) - server computes hash of basic TCB, Sends this Acknowledgment field in SYN+ACK, client returns this when completing connection, more info being stored in TCP Timestamp to preserve high-performance options
  - Alters TCP synchronization procedures from RFC 793
- Development of stateless connect protocols

8.4 Timers, and Delayed and Missing Packets

8.4.1 TCP Round Trip Time (RTT)

- TCP timers have to adjust to RTT on connection
- Endpoints measure RTT from packets
- RTT is variable
- Adjust timeout appropriately
- $EstimatedRTT(t) = (1-\alpha) * EstimatedRTT(t-1) + \alpha * SampleRTT(t)$

8.4.2 What happens on Timeout

- Depends on state client/server are in
- During data transfer - packets will be retransmitted
- Retransmit segment(s) that caused timeout
- Restart timer

Packet loss due to lost data vs premature timeout

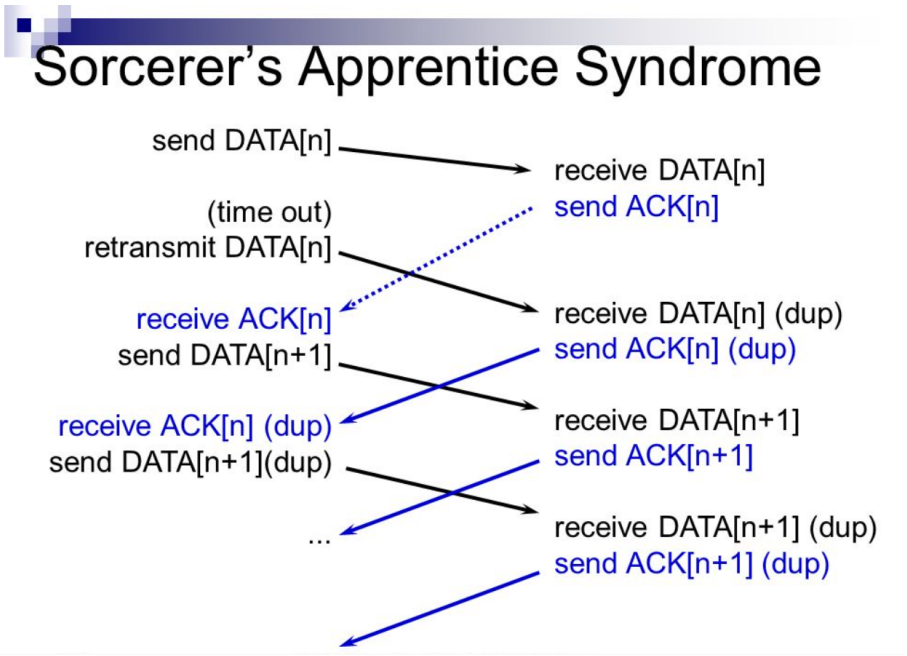
8.4.3 Packet Duplication

- TCP sockets typically transmit sequences of packets
- Don't wait for each packet to be acknowledged first
- Selective Acknowledgment (SACK) - allows missing packets to be requested, appear as "duplicate acknowledgments" - Wireshark

8.5 TFTP: "Sorcerer's Apprentice Syndrome"

8.5.1 Stop and Wait

- Client sends Data
- Server sends ACK
- Client sends more Data
- What could go wrong
  - ACK lost so client retransmit
  - Packet arrives a little later - triggered two ACK's, receipt of those ACK's triggered two more packets



9 Lecture 9: TCP/IP and UDP

9.1 Internet Control Protocols

9.1.1 Design Decisions in the Early Internet

- Tended to be very prosaic - limited funding, getting it working at all was favored over complexity
- Security was not considered - other systems at the time had a lot of security, internet was a closed research network (misbehaving networks/hosts could be thrown off), tradition continued for several decades
- Frequent congestion collapses were a feature - in a congestion collapse network becomes unusable, positive feedback loop (retransmission of dropped packets increases load, causes more packets to be lost)
- First intervention was to make TCP/IP reliably unreliable - end computers dropped connection if too many delayed/lost packets

Internet Control Protocols - conceptually at the same level as IP

- DHCP: Dynamic Host Control Protocol
  - Host Address assignment

- ICMP: Internet Control Message Protocol
  - As used by ping
  - Encapsulated in IP packets
  - Increasingly being blocked by routers/firewalls
  - Can be used to scan ports, probe for hosts
- ARP Address Resolution Protocol
  - Replaced by NDP (Neighbor Discovery Protocol) in IPv6

ARP (Address Resolution Protocol) - bridges between IP addresses (Internet, dynamic/static assignment) and MAC addresses (local network, static assignment)

9.1.2 ARP (In)Security

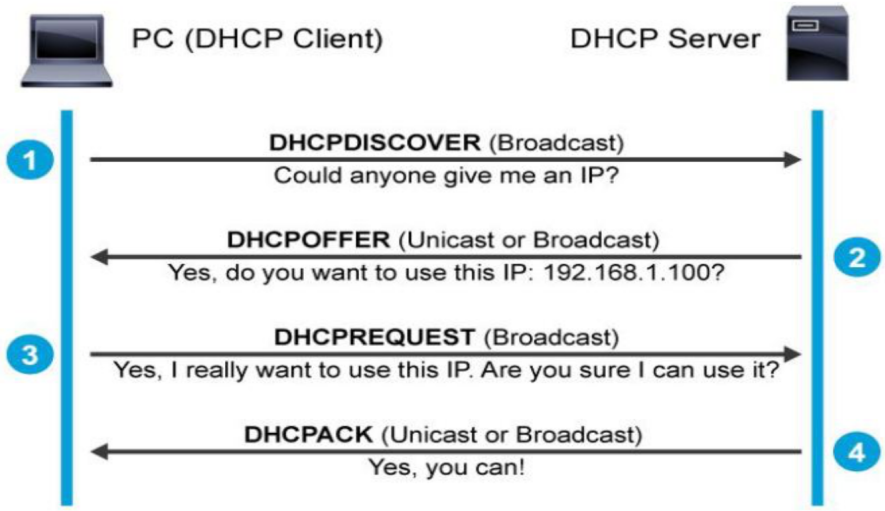
- ARP Spoofing/Cache Poisoning/ARP poison routing
  - Attempt to associate attacker's MAC address with target host IP address
  - Attacker's machine will now receive target's traffic
  - aka "Man in the Middle" attack
  - There is no authentication in ARP protocol itself
  - Requires direct access to the local network
- Legitimate use is seamless network redundancy
  - i.e. Replace a lower layer server without disrupting existing traffic
  - Also for monitoring traffic for debugging
- Defense
  - Cross check IP and MAC assignments
  - Kernel software to reject uncertified ARP responses
  - Restrict access to local network to known MAC addresses

9.2 DHCP: Dynamic Host Configuration Protocol

9.2.1 DHCP protocol

- Dynamically assigns IP address and other network configuration to a host

- Clients ease DHCP addresses for configured time
- Requires configured DHCP server
- Connectionless, UDP protocol



Booting from DHCP

Can configure DHCP sp that client is also sent a file, aka BOOTP server, TFTP(trivial file transfer protocol) - no user authentication or error recovery on large file transfer cause it uses UDP

9.2.2 UDP vs TCP

UDP

- Connectionless protocol
  - Checksums for data integrity
  - Port number addressing
- No end to end guarantee
  - UDP datagrams can be delivered in any order
  - UDP datagrams can be dropped at any point
  - Lack of overhead means UDP should be delivered faster
- Fairly simple protocol
- Preferred for trouble shooting, logging, congestion or non-critical traffic etc

TCP

- Network Guarantees
  - All packets will be delivered
  - In the order they were sent
  - eg. file transmission
  - Or else connection will be dropped
  - No guarantees about how long this will take

9.3 Network Congestion Collapse Issues

9.3.1 TCP Design

- Intended originally for slow file transfer (E-mail, Usenet, IRC, FTP(File transfer protocol))
- No WiFi or Cellular Internet - often much higher delays than wired connections, higher errors, more prone to larger dropouts, requires more buffering end to end
- Hosts relatively homogeneous
- Requires large buffers end to end - large buffers can potentially increase overall delay
- There is no centralized way to control overall network load because the Internet was designed to be completely decentralized
- Large enough networks can always be overloaded
- Congestion handling had to be built into its protocol - TCP connections will attempt to perform end to end flow control

9.3.2 Goodput vs Overload

- Goodput is the useful user traffic transmitted by a network
- In any kind of congestion collapse the network is still carrying its full load of traffic, its just that the traffic representing user data has been driven out by administrative and other recovery traffic such as retransmits
- Matters because future of network applications isn't simply point to point networking, distributed application congestion is also a thing

9.3.3 Original Idea with TCP and UDP

- TCP for applications that were not time sensitive
  - Needed reliable connections
  - Network handles the nasty error correction/detection/reordering
  - If traffic is dropped end nodes will retransmit
  - Temporary congestion, end nodes can recover
  - Sustained congestion, TCP will self throttle (slow down), eventually connection will be dropped
- UDP for applications that were time sensitive
  - Transmission is faster - less reliable
  - Because packets are time sensitive, no point in delivering them too late
  - Would be nice to prioritize it, shouldn't be that much of it anyway

TCP buffers: each node on the route maintains its own set of buffers for its traffic and next hop destinations. This allows potentially for some prioritization decisions to be made about which segments get forwarded and when
TCP starvation: People are choosing unreliable traffic (UDP) over reliable traffic (TCP)

9.4 TCP Error Handling

Protocol Error User side: can't connect, noticeably unreliable (connection keeps dropping), not fast enough, high variance in packet arrival times (real time applications)
Protocol Error Network side: Missing packets, delayed packets, timeouts being too long or too short when reacting to errors, network congestion due to

too much traffic, behavior of protocol triggering network congestion, especially tragedy of the commons type errors
Flow control can help (management of data flows)

9.4.1 TCP Flow Control - Window Sizes

- Managed using Window Sizes
  - Receiver advertises a Window size to sender
  - Sender cannot send more unacknowledged bytes than this
- In TCP Header, max size is 65535 bytes
  - Actual size is arbitrary - can be anything < 65535
  - Not big enough for high speed links when they were introduced
- Options - TCP Window Scaling
  - Multiplier on header window size
  - Window size is right shifted by value in header options - size of window cannot exceed max sequence number
  - Advertised during setup
  - Must be supported by both sides, otherwise ignored
  - "Long Fat Networks" LFN - large bandwidth delay, i.e. modern fiber optic networks
- Sender window size is controlled by *receiver's* window value
- Window size is used for end to end signaling as a crude (on/off) way to control flow

Bandwidth-Delay Product
Capacity of available network path: $Capacity(bits) = bandwidth(bits/s) * RTT(s)$
For example: 1Gbps domestic Ethernet, 50ms RTT
$Bandwidth * RTT = 10^9 b/s * 10^{-3} * 50ms = 10^6 * 50 = 6.25 MB/connection$ (divided by 8 in the end to convert from Mb to MB

9.4.2 Zero Window

- Means the TCP receive buffer is full
- Receiver can send an ACK with an offered window of 0, later when the

buffer has cleared, it sends a window update with a non-zero offered window size

- This is very crude - it is possible for the other end to infer from this they should slow down but it would be much better if it was a signal of the rate the receiver could handle

9.4.3 keepalive

- Endpoints with low traffic can periodically exchange "keepalive" packets
- Stops connection from being automatically torn down due to no traffic
- Under linux this is enabled using a sysctl (kernel) control

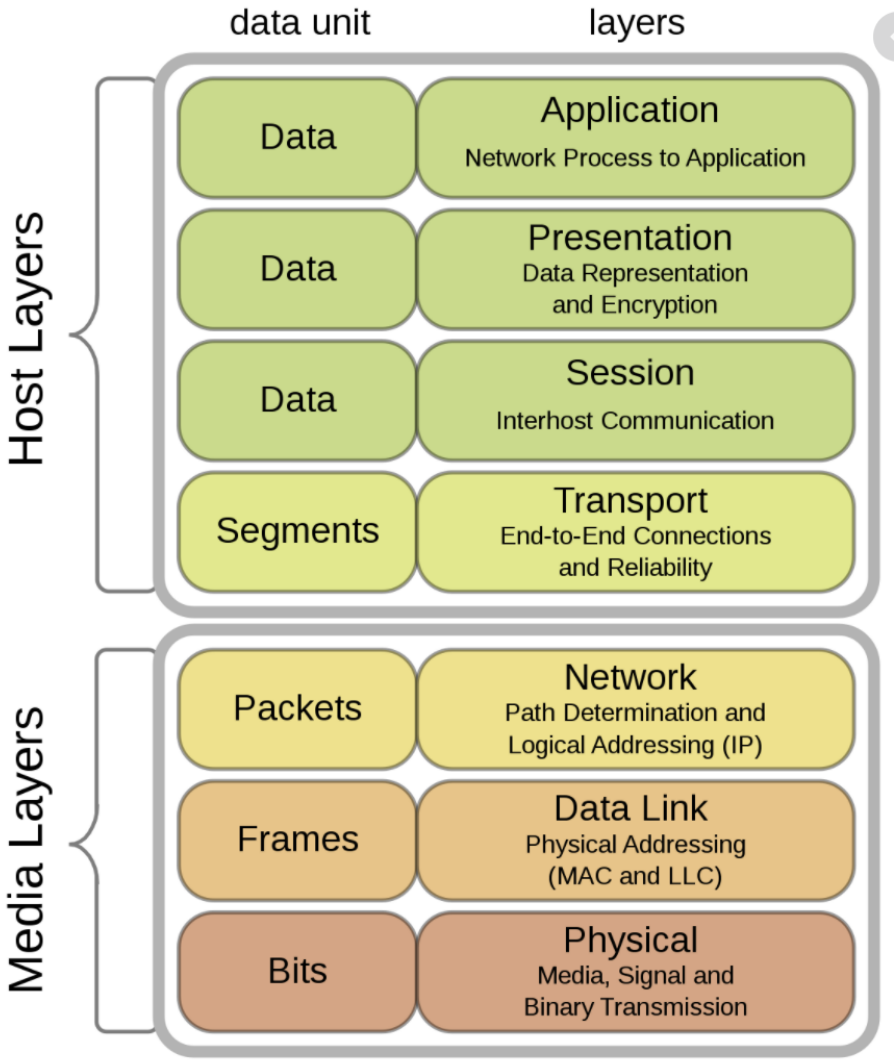
9.4.4 When writing network programs..

- Assuming you control both ends - the OS can often help out
- Consider application requirements very carefully
- Talk to Network Admins and other experts

10 Lecture 10: Advanced TCP/IP

Any network must solve how to start, tear down connection and not let connection bring down its network (Avoid Congestion Collapse)

Network Stack



10.1 Piggyback ACKs

Network overhead (Administrative overhead, Badput): this just gets worse and worse as we go up network stack into application layer

Header

TCP Packet Overhead: IP + TCP header = 20 + 20 = 40 bytes

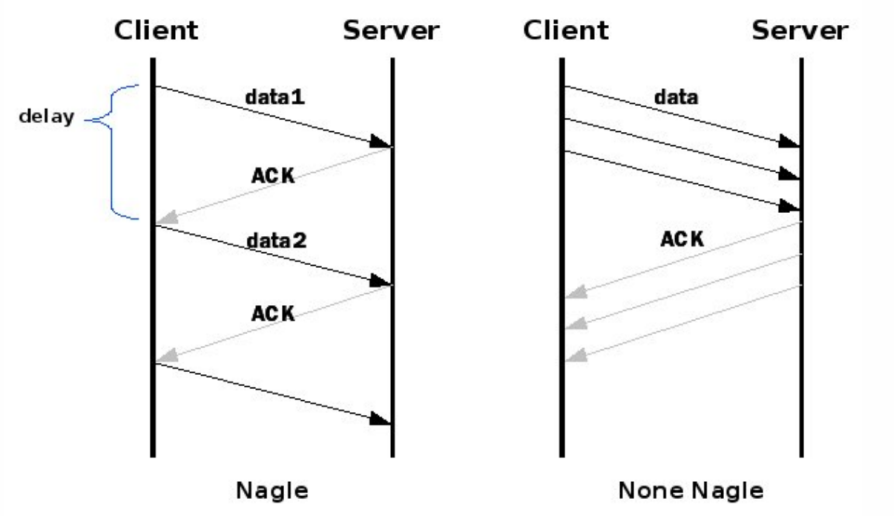
UDP Packet Overhead: IP header + UDP header = 20 + 8 = 28 bytes

Ethernet framing and data is extra, mobile data charged by the byte

On overhead basis it is much more efficient to send large packets so long as they don't get lost too often, can become extremely significant very quickly.

10.1.1 Delayed Acknowledgments (Piggyback ACK's)

- A host may delay sending an ACK response by up to 500ms
- But with a stream of full-size segments it must acknowledge every second segment
- The ACK's can then be placed on data traffic going the other way (potential to save 40+ bytes every time this is done
- Potentially improves performance/traffic with small packets
- Can create **"Silly Window Syndrome"**
  - Caused by poorly implemented TCP flow control
  - Can arise if sending application program creates data slowly, receiving application program consumes data slowly, or both
  - Can happen easily if application programmers take the network for granted
  - Once a connection gets into this state it can be hard to recover from
  - Solution: Force receiver to wait until it has a reasonable amount of buffer space (wait until buffer is half empty or until it can handle the starting Max Segment Size)
  - Goal of solution: Sender doesn't send small segments and receiver doesn't ask for them
- Interacts badly with Nagle's Algorithm
  - Attempts to limit number of small packets/connection
  - Typically on by default
  - As long as there is a sent packet outstanding - sender will buffer data arriving to be sent
  - Data is then sent when segment amount of data has been received and when ACK arrives for previous segment



10.2 Nagle's Algorithm Issues

- Should be disabled if working with real time traffic
- eg. Interactive games, lots of small real time update packets being sent
- Can be problematic with timeouts if using aggressive (relatively short) timeouts, connection can be dropped while Nagle is delaying traffic, servers may do this if handling lots of connections (don't want stale connections around longer than necessary)
- Can create a deadlock with delayed ACKs (piggy back acks) - receiver waits for data to piggyback ack on, sender waits for an ack to send data
- Manages to be what you don't want to happen on time critical real time applications

10.2.1 ACK starvation

- Asymmetric traffic - large flows in one direction, small or no traffic coming back
- ACK starvation can occur if no traffic to piggy back on
- Typically some kind of bug in underlying network software
- Manifests intermittently because not seen with normal traffic

10.3 TCP Timers

10.3.1 TCP Timer Management

- Retransmission (RTO) - waiting for a segment ACK
- Persistence - used to periodically send window probes, send packet with no data and ACK bit on, reply is also no data with ACK bit on
- Keepalive - length of time without data before server tears down connection
- Time-Waited - connection termination

10.4 Congestion Control with Timers

10.4.1 Van Jacobson Slow Start Algorithm

Goal is kind of fun, provide a method over the entire network, for each host to automatically figure out the highest throughput it can get for each application, without any outside intervention by the programmers or network administrators. We may not like the way these algorithms impact us locally but we'd like it even less if they weren't there.
Congestion window: Sender's window size can be altered by network congestion (effectively overrides user settings), sender has 2 window sizes (receiver's advertised window size and congestion window size CWND), actual size is minimum of the two.
TCP Congestion Control: Additive Increase, Multiplicative Decrease: Sender slowly increases transmission rate (window size) until it reaches limit or segment losses occur. Effectively probes for network congestion and reacts when it finds it, believed that this can result in large scale wave effects in network traffic

10.4.2 Slow start

- At start of connection: CWND = MSS (Max Segment Size)
- For each ACKed segment CWND += 1 MSS until half the allowed window size reached, after this it is increased sublinearly: CWND += 1/segment ACK until threshold is reached or ACK timeout occurs
- Initially quick growth - increases exponentially for half the threshold size
- Slow start required for all TCP connections
- Based on traffic analysis and control theory

Congestion

- ACK Timeout is treated by TCP as a congestion symptom

- Note, this can occur anywhere in the route
- When it occurs: reduce threshold, threshold value is set to 1/2 the last CWND where CWND = 1MSS (Max segment size
- Rinse, repeat, recycle

Issues

- Assumes unacknowledged segments are due to network congestion - can also be lost due to poor data link layers (wireless and cellular networks)
- Performs badly with short lived connections because it doesn't have time to converge, can't always hold connections open (persistence)

1988 TCP Tahoe:: Fast Retransmit and Fast Recovery

- If TCP detects an out of order segment
  - Generate an immediate ACK (duplicate ACK)
  - Inform sender what segment number is expected
- If 3 or more consecutive duplicate ACK's received:
  - Retransmit immediately - **Fast Retransmit**
  - Reset threshold to half last CWND
  - Each new ACK increases CWND by $MSS/CWND$
- i.e. slow start remains, small change to congestion handling arithmetic

1990 TCP Reno (Van Jacobson)

- Added fast recovery
- At congestion (lost ACK)
  - Save half CWND as threshold **and** as new CWND
  - i.e. skip slow start
  - Once threshold is reached, CWND += 1 RTT

Common complaint is TCP gives artificially low throughputs, Ignore slow start, assume always have data to send

You could get better throughput with multiple TCP if OS threshold is incorrectly set too low, limit on connection will be less than actual bandwidth and limit is on each connection.

10.4.3 Random Early Detection (RED)

- Network components buffer as much as they can
- Drops packets that come in and can't be buffered - also known as "Tail Drop", biased against bursty traffic
- Issues: Unfair allocation of buffer space to flows, TCP Global Synchronization (connections hod traffic back/send it simultaneously)
- Solution: Randomly drop packets, fairer: more packets a host sends, more likely to be dropped

10.4.4 Explicit Congestion Notification (ECN)

- Must be negotiated by all participants
- Works in conjunction with AQM (Active Queue Management)
- Router explicitly marks packets to signal congestion
- ECN compliant senders initiate congestion avoidance
- Packets from non-ECN flows are dropped (RED)

10.4.5 Active Queue Management(AQM)

- Detect early signs of congestion
- Use ECN to try and maintain a "good" average queue size
- If managing lots of connections (router) - randomly notify
- How it works: Packet's aren't (shouldn't be) dropped ECN notification is sent instead, With RED, ECN sent instead of packet drop, If it works, sender adapts as if a packet was dropped, Changes Congestion Window etc.
- Next problem: Low-rate denial of service attacks, RED algorithms notably vulnerable (Attack causes oscillating TCP queue size, RED chips in to make things worse), Robust Random Early Detection (RRED) (Attempt to filter out attack packets, Assume well behaved hosts will back off, Preemptively drop packets from hosts that don't)

10.4.6 TCP Cubic

- Designed for high bandwidth/high latency networks
- Window is a cubic function of time since last congestion event
- Window size depends on previous congestion event
- Does not rely on ACK receipt (Windows size quickly grows to previous limit, Plateaus - grows very slowly probing for more bandwidth, Grows quickly if it finds some)
- Window growth is independent of RTT - Fairer to further away streams

11 Lecture 11: Routing

11.1 Routing Overview

11.1.1 Bridging vs Routing

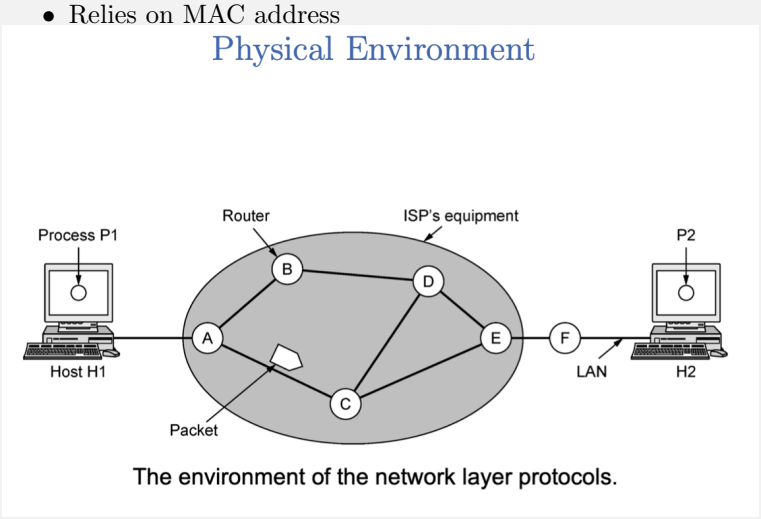
Determined by the path taken between two points.

Bridging

- Creating a simple aggregate network from multiple network segments
- eg. Connecting two LAN segments
- Relies on MAC address

Routing

- Routing is the process of selecting a path for traffic in a network OR between multiple networks

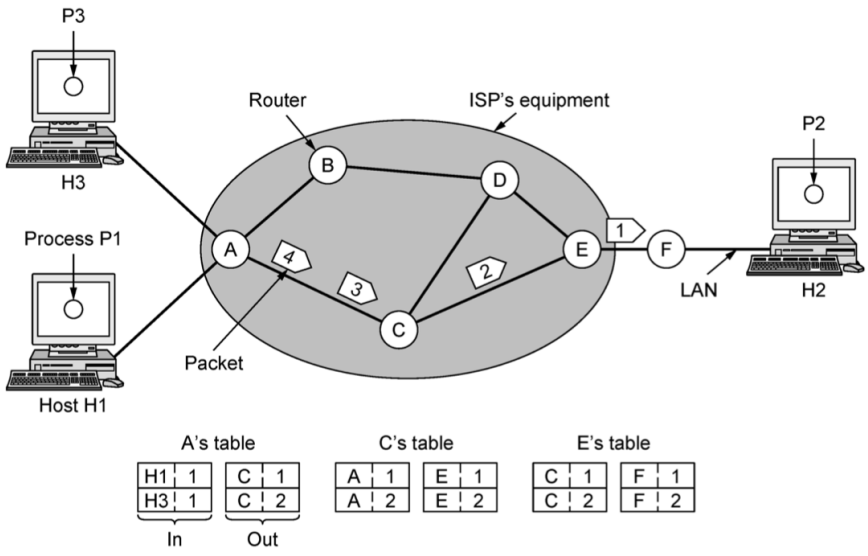


The environment of the network layer protocols.

11.1.2 Routing

- Datagram routing: route chosen on packet by packet basis - in practice there is less variability than theorized
- Virtual Circuit Routing: route chosen per distinct connection (like phone calls, once set up all traffic on circuit takes the same path)
- Static routing: route chosen as a prearranged, hard defined, set of relationships
- Good routing depends on distance (length of the link), speed, delay on the link, congestion, monetary cost, peering agreements, and load considerations. Classic example of a real time constrained information transfer problem
- If each router in the network has to exchange full routing information with every other router in order to determine network paths, then the resulting traffic load shoots up to infinity extremely quickly
- Routing has to work across the Internet between network providers and within their networks (in practice another acronym soup of protocols

Virtual Circuit



Routing within a virtual-circuit network.

The perspective users have of the "Internet" is not the reality. Users should see a seamless internet, but in fact it is a loose federation of autonomous systems

- locally controlled networks, that are joined together by routing algorithms.

11.1.3 Routing Alphabet Soup 1

- Interior Gateway Protocol (IGP)
  - Protocol used to exchange routing information within an AS
- Border Gateway Protocol (BGP)
  - Exterior Gateway Protocol (EGP), used to exchange routing and reachability information between AS's
- Interior Border Gateway Protocol (iBGP)
  - Full mesh protocol used within AS

11.2 Routing, Forwarding, and Bridging: Interior Gateway Protocols

Routing Protocols and Algorithms - Routing Algorithms are used to make decisions about the best path to use, routing protocols implement routing algorithms

11.2.1 Routing Table, Routing Information Base(RIB)

- Table stored in router or networked computer
- Lists routes to known network destinations - may include metrics/weights for those routes (eg. distances)
- Effectively contains information about the local network topology
- Goal of Routing protocols is to construct Routing Tables

11.2.2 Forwarding Table, aka Content-addressable memory(CAM) table

- Used to relay packets between connected network segments
- Optimized for fast lookup of destination addresses
- Precise role depends on network location
- Maps input network interface to correct output network interface
- Forwarding decision also depends on type of forwarding
  - **Unicast:** 1 - 1 (source : destination)
  - **Multicast:** 1 - N (source : destinations in multicast group)
  - **Broadcast:** 1 - All (source : all hosts on next segment)

11.3 Bridging

11.3.1 Bridges

- Fairly dumb devices that allow LANs to be inter-connected - one of the ways VPN's can be configured on your local host, when this is the case ifconfig only shows one ip address
- Relay on forwarding tables - no routing
- Broadcasts used to find unknown hosts
- Used to connect LAN segments to each other
- Connect LAN's to routers
- Peripheral devices, used as cheaper alternatives

11.3.2 Routing Protocols

- Interior Gateway Protocols (IGP)
  - Open Shortest Path First (OSPF)
  - Routing Information Protocol (RIP)
  - Intermediate System to Intermediate System (IS-IS)
  - Enhanced Interior Gateway Protocol (EIGRP)
- Exterior Gateway Protocols
  - Exterior Gateway Protocol (EGP)
  - Border Gateway Protocol (BGP)

Routing Algorithm Goals: Reliable, Fair, Optimal (often in conflict with fairness), Maximize total network throughput, Minimize total packet delay (latency), stable - must converge to some kind of solution (equilibrium - things aren't changing a whole lot, does not mean an optimal solution has been found).

11.3.3 Directed Graphs (digraphs)

- Digraph is a finite nonempty set of nodes N and a set of ordered node pairs A called directed arcs
- Data networks are best represented with digraphs, although typically links tend to be bidirectional
- Directed (all lines are arrows) Acyclic (No cycles (loops) in the graph) Graph (DAG): If a sink tree includes all possible paths it becomes a DAG, used to solve shortest path problem

11.3.4 Shortest Path Routing

- Links between nodes are allocated a cost: actual cost to AS and link length, delay, congestion
- Costs may change with time
- Length of route is sum of all costs
- Shortest path == path with minimum length
- Algorithms: Bellman-Ford (Centralized and distributed, distance vector routing), Dijkstra's algorithm (Link state routing)

11.4 Distance Vector Routing

Each router maintains a table (vector) - best known distance to each destination, which output link to use to get there
Router is assumed to know distance - can be measured or set (eg. use router ping to measure RTT and delay or examine output queues to that link (congestion))
Distance vector algorithm: Rather than calculating the entire networks table, each node is continuously reacting to updates from its neighbors, updating its own distance table and rebroadcasting it.

11.4.1 Routing Information Protocol (RIP)

- Automatic routing protocol used in early internet
- Limit of 15 hops (to prevent routing loops
- RIPv1 routers broadcast routing table updates every 30s
- Poor scalability (15 hop max)
- Poor convergence - Bellman-Ford is being continuously recalculated
- Special Features:
  - Variety of methods used to prevent known routing problems
  - Split-Horizon route advertisement to prevent routing loops and forbid router from advertising a route back on the interface that provided it
  - Route Poisoning - sends updates with an infinity metric, indicates that a previously advertised route has gone down
  - Holddowns - prevent delayed update messages changing metric
  - Used UDP

11.5 Link-State Routing

11.5.1 Idea

- Each Router starts up and performs the following:
  1. Discover neighbors and learn their network addresses
  2. Measure the delay or cost to each of its neighbors (eg. RTT)
  3. Construct a routing announcement with this information
  4. Send this to all other routers (not just neighbors, flooding)
  5. Compute shortest path to every other router
- This distributes (over time) complete information on network to all routers

Dijkstra's Algorithm is centralized (single node gets topology info and computes routes and routes then broadcast to the rest of the network) and distributed (each node i broadcasts d_ij to all j its neighbors who in turn flood their neighbors, all nodes then calculate shortest paths - Open Shortest Path First (OSPF))

Distance Vector vs Link State Algorithms

Distance Vector	Link State
Uses hop count for metric	Uses Shortest Path
Network information at 1 hop remove	Gets entire network topology
Relies on timed/periodic updates	Event triggered updates
Slow convergence	Faster convergence
Susceptible to routing loops	Less susceptible

11.5.2 Hierarchical Routing

- Implicit assumptions so far:
  - Routers are identical in size/capacity
  - Network is evenly distributed
  - Everybody has the same routing constraints
- Different routing decisions (peering arrangements - will carry each other's traffic)

Autonomous System Number (ASN): AS have their own ID number

allocated by IANA

12 Lecture 12: Routing Protocols

12.1 Review

12.1.1 Path determination

- Packet routing problem believed to be NP-hard
- NP-complete Routing problems:
  - Path-constrained, path-optimization - select shortest path meeting specified constraints
  - Multi-path-constrained routing - find multiple paths meeting constraint, issue for high performance/traffic environments
  - These are being studied for more elaborate routing constraint schemes

12.1.2 Several Different Algorithm Families in Use

- Key issue is how table updates are distributed
  - Flooding - broadcast to all, scaling issues, typically used on local campuses
  - Peer update (propagation delay issues) - updates to directly connected/trusted peers
- Neither approach is guaranteed to be secure

Organized mesh topologies allow arbitrary scaling, with the assumption that significant amounts of traffic can be retained in local groups of communicating nodes.

12.1.3 Within an Autonomous System (eg. ISP, RU, etc.)

- AS can choose its own interior routing scheme or schemes
- Responsible for all interior routing, routing to and from other AS
- Four types:
  - Multihoned: maintains connections to more than 1 AS
  - Stub: connects to only one AS
  - Transit: Allows connections through itself to other AS's
  - Internet Exchange Point (IX or IXP): Physical infrastructure - Used by ISP's or Content Delivery Networks to exchange traffic

12.2 OSPF

12.2.1 Open Shortest Path First (OSPF)

- "Open": publicly available
- Widely used in enterprise networks - companies/corporate
- Uses Link State algorithm - maintains map of network topology at each node, computers routes using Dijkstra's algorithms
- "Advertisements" - announcements of table changes - sent directly over IP
- All OSPF messages are authenticated (prevent malicious spoofing)
- Multiple same-cost paths allowed
- "Type of Service" (TOS): Supposed to provide different routes per type of traffic, in practice never implemented by major equipment manufacturers
- Supports unicast (1:1) and multicast (1:n): multicast is implemented using a group scheme, hosts join a multicast group, routers automatically distribute traffic to group members
- Supports a two-layer hierarchy (Hierarchical OSPF)

12.2.2 Hierarchical OSPF

- Two levels: local area and backbone
- Each node knows: local area topology and shortest path to other local area
- Area border routers: summarize distances to nets in own area, advertise to other area border routers
- Backbone routers: route to other backbone routers
- Boundary Routers connect to other AS's

12.2.3 Link State Routing

1. Each Router when it starts up, examines its links
2. Measures a "cost" of the links for each neighbor
3. delay - ICMP echo packet, link bandwidth, actual cost
4. Constructs a packet of all information that it knows

5. Sends the packet to all other routers in the network (flood)
6. Processes updates from network and sends updates to network

12.3 IS-IS

12.3.1 Intermediate System to Intermediate System (IS-IS)

- Interior Gateway Protocol - used by transit and core networks
- Link State Protocol
- Uses Dijkstra algorithm to compute best path
- Similar convergence properties
- Operates directly above level 2 - protocol neutral - can support IPv6 without modification

12.4 IS-IS vs OSPF

- OSPF uses IP Protocol 89 as transport (Data Link Header — IP Header — OSPF Header — OSPF Data)
- IS-IS is directly encapsulated in Layer 2 (Data Link Header — IS-IS Header — IS-IS Data)
- IS-IS more stable version of OSPF in early 1990's
- Migration to IPv6 is easier with IS-IS

12.5 Border Gateway Protocol: BGP

"All an intradomain protocol has to do is move packets as efficiently as possible from the source to the destination. It does not have to worry about politics"
Tannenbaum

We do need to think about politics.

12.5.1 BGP

- BGP neighbors must be manually configured
- Single TCP connection between routers
- All BGP messages are sent as unicast (1:1)

- BGP routers may only belong to a single AS
- Once BGP peering established, KEEPALIVE messages sent every 60s (default)
- UPDATE messages contain Network Layer Reachability Information (NLRI)

12.5.2 BGP Examples

1. Do not carry commercial traffic on the educational network
2. Never send traffic from the pentagon on a route through Iraq
3. Use Telia Sonera instead of Verizon because it is cheaper
4. Don't use ATT in Australia because performance is poor
5. Traffic starting or ending at Apple should not transit Google

12.5.3 BGP Preconditions

1. Next-hop IP Address of path is reachable
2. Local AS number is not part of the AS_PATH - loop prevention
3. BGP synchronization enabled - candidate prefix is in IGP routing table
4. BGP prefix is not dampened

12.5.4 BGP Prefix Dampening

- Used to reduce propagation of unstable routes
- Prefix "flaps" - moved into dampening state history - assigned penalty of 1000
- Each flap incurs further 1000 penalty
- If it reaches suppress-limit (typically 2000) won't be advertsed
- Penalty reduced by half by the half-life timer once it stabilizes

12.5.5 BGP Decision Process (first 6 criteria of 12)

1. WEIGHT: at Local router, used to prefer one of multiple uplinks
2. LOCAL-PREF: at Local ASN - prefer one of several routers
3. LOCALLY GENERATED: local routes preferred over external
4. AS-PATH length: preference to shortest AS_PATHS
5. ORIGIN: Prefer 0-IGP over 1-EGP
6. MED: preference is given to lowest MED metric - "Cold-potato" routing or best-exit routing, most networks use hot-potato - get rid of foreign traffic, Multi-Exit Discriminator, lowest MED preferred (only held at adjacent routers)

12.5.6 Interior Border Gateway Protocol (iBGP)

- Used with transit AS's
- BGP router must add its own AS number when forwarding to another AS to prevent routing loops from occurring, BGP will drop a router if it sees its own ASN in the AS_PATH list
- If router forwards BGP within its own AS - will see its own ASN - Internal BGP used in this case

12.5.7 Border Gateway Protocol

- The glue that holds the Internet together
- Routing protocol used to interconnect Autonomous Systems
- BGP allows each AS to:
  - Obtain reachability information from neighboring AS's
  - Propagate this information to all AS-internal routers
  - Determine good routes to other networks (based on reachability and policy)
  - Advertise its own existence
- Path Vector routing protocol
  - Only installs "best path" into the routing table
  - Only announces "best path" to other BGP peers
  - "Best path" can be manually controlled
- Uses TCP as transport protocol

12.5.8 Path Vector Routing Protocols

- Case of distance-vector protocols: use distance between themselves and destination as metric (eg. RIP), Best Path Selection Algorithm (Bellman-Ford algorithm is similar), routers do not know the entire network topology, they know the locally best output link for a given IP range
- Router appends its identifier to current path in updates
- Allows loops to be avoided

Hot Potato Routing: Suppose there are two or more best inter-routes then choose route with closest NEXT-HOP

Entry gets into forwarding table by: Router becomes aware of prefix via BGP route advertisements from other routers, determine router output port for prefix (use BGP route selection to find best inter-AS route, use OSPF to find best intra-AS route leading to best inter-AS route, router identifies router port for that best route), enter prefix-port entry in forwarding table

13 Lecture 13: BGP Security and DNS

13.1 Cryptographic Signing

13.1.1 Idea

- Function which maps arbitrary size data (input) to fixed length output
- Designed to be a 1-way function - infeasible to reverse (invert)
- Five important properties (for the function):
  1. Deterministic: Same message - same hash
  2. Computable (quick to generate)
  3. Infeasible to reverse it
  4. Small changes to message create an uncorrelated hash message (otherwise would be possible to reverse)
  5. Infeasible to get 2 messages with same hash value
- Very large input bit stream can be reduced to a small signature file which can be used for verification (or in conjunction with a private key for authentication)

13.1.2 Applications

- Used to verify file integrity (is copy the unchanged original?)
- Also used to identify the file (file "signature", eg. git software archive)
- Digital Signatures - guarantee of authenticity of message
- Password verification - store cryptographic has of password, avoid storing password in cleartext or even encrypted, original password cannot be determined from stored hash
- Proof of work - Bitcoin and clock ledger technologies

13.1.3 Message-digest algorithm: MD5

- Widely used hash function/algorithm - returns 128-bit hash for message
- Used to provide guarantee original file is same copy
- Was widely used as a cryptographic hash function
- Examples: Provide 1-way hash for a password, BGP spoof protection, Lightweight Directory Access Protocol (LDAP)

13.1.4 "Nothing up my Sleeve" Numbers

- Cryptographic algorithms often need some form of random initialization (eg. values of π, e, irrational roots)
- Data Encryption Standard (DES) S-Box constants - No explanation by NSA at time, later found to protect against differential cryptanalysis
- MD5 uses sine() derived constants
- P Curve constants
  - Coefficients in curves generated by hashing specified seeds
  - Under suspicion since 1999
  - May give predictable values

MD5: An example of what can go wrong.

- Designed in 1991, as secure replacement for MD4 (Rivest)
- "pseudo-collision" found in MD5 compression function
- 1996 Dobbertin found collision in compression algorithm
  - Recommendation switch to SHA-1
- March 2004 MD5CRK distributed birthday attack announced
  - Collisions announced in 17 August 2004 (Wang, et. al)
- 1 March 2005, Lenstra, Wang, De Weger - construction of two X.509 certificates with different public keys
- 2008 - collision attack sued to fake SSL certificate validity
- December 2010 Single Block 512-bit collision published

SHA-256, SHA-512 current replacements to MD5, susceptible to length extension attacks

MD5 is horribly insecure and should never be used

13.2 BGPSEC

13.2.1 BGP Recap

- BGP allows routers to create a distributed routing table of IP Prefixes
- Allows AS's to discover routes
- Routing is based on local criteria, not necessarily efficient criteria - price, route length, lots of flexibility
- BGP lacks basic authentication mechanisms (old protocol)

13.2.2 Key BGP BCPs (Best Common Practices)

- Blind Attacker
  - RFC2827 - Even without broad adoption you can prevent people from spoofing your ranges, and thus all TCP attacks
  - BGP ACLs - Don't let invalid BGP packets on the wire
- Non-Blind Attacker

  - L2 best practices - Stop sniffing, hijacking
  - MD5 - adds additional pain to the attacker
  - Ingress/Egress prefix filtering - limits damage in case of compromise (update flooding)

- Compromised Router
  - Ingress/Egress prefix filtering - limits extent of damage a compromised router can cause (update flooding)

13.3 Secure Border Gateway Protocol(s)

"The nice thing about standards is that you have so many to choose from."
Andrew Tannenbaum, 2nd edition Computer Networks

13.3.1 BGPSEC

- RPKI provides certification of number holdings in registry
- Route Origin Authorization (ROA) - digital signed authority for route advertisements, contains address prefix and range of allowed sizes, originating ASN
- Receiving AS doesn't know the route path is valid, no way to validate that the authorized AS is providing a valid path
- RPKI can be circumvented (eg. route leaks and path shortening)

13.4 Domain Name System (DNS)

13.4.1 RFC 882, RFC883, RFC1034, RFC1035, RFC1123 and RFC2181

- Decentralized hierarchical naming system
- Distributed database
- Maps user friendly computer names and url's to computer IP addresses
- Critically: can map more than one computer to the same name: Aliasing - allows load distribution across multiple hosts
- Reverse mapping IP address $\Rightarrow$ name
- Relies on local servers and caching for performance
- 13 Authorities provide 1058 Top Level Domains (TLD), eg. .is, .com, .tv etc.

13.4.2 Name Hierarchy

- Unique domain suffix is assigned by Authority
- Domain administrator has complete control over the domain
- No limit on number of subdomains or number of levels
- computer.site.division.company.com
- computer.site.subdivision.division.company.com
- Domains within an organization do not have to be uniform in number of subdomains or levels

13.4.3 Local DNS Server

- Not strictly part of the hierarchy
- Each AS, ISP, company, etc. has one or more (typically at least 2) - Default Name Server
- Local hosts query is first sent to local DNS
  - Maintains local cache of known name-IP address translations
  - Acts as proxy, forwards query into public DNS hierarchy
  - Caches may be out of date for some time

13.4.4 There are 5 Addressing Methods

- Unicast addressing: 1 $\Rightarrow$ 1 eg. TCP/IP, UDP
- Broadcast: 1 $\Rightarrow$ All (local networks
- Multicast: 1 $\Rightarrow$ Many (Address can specify subset of nodes to receive)
- Anycast: 1 to 1 from a set based on distance
- Geocast: 1 to many based on physical networks (Mobile, Ad hoc, networking)

13.4.5 BIND: Berkeley Internet Name Domain

Local Machine Resolver $\Rightarrow$ Local DNS Caching Resolve $\Rightarrow$ Regional Server Authoritative Server

- Primary, Secondary and Tertiary name servers can be configured
- When any DNS cannot resolve a request it bounces up to a root-server
- Propagation of changes can take several hours

- Resolvers use UDP for single name requests
- Resolves may also fill in a partial name and query
- TCP for groups of names
- Recursive Query - refer query up the hierarchy
- Iterative Query - reply or give me next server in hierarchy
- Each entry has a time to live (TTL)
- Can also specify type of service (eg. MX - mail)

13.4.6 Attacks

- DOS(Denial of Service) - direct attack on DNS server
- DNS Amplification/Reflection Attack
- DNS Hijacking
- Cache Poisoning
- DNS Tunneling (UDP port 53)
- DNS Server performance attacks (Slow Drip, SYN floods, NXDomain)

Open DNS Resolvers must be closed

13.4.7 Mitigations

- DNSSEC (estimated 7% deployment circa 2016)
- Company level: separate servers for internal and external resolution
- Internal server inside the company network
- Monitor DNS servers carefully, (real time load, etc.)

13.5 DNSSEC

- Adds cryptographic key protection to DNS server records
- Provides origin authentication and integrity assurance
- Signed origin attestation chain (PKI) starting at root

14 Lecture 14: TCP/IP in the Application Layer

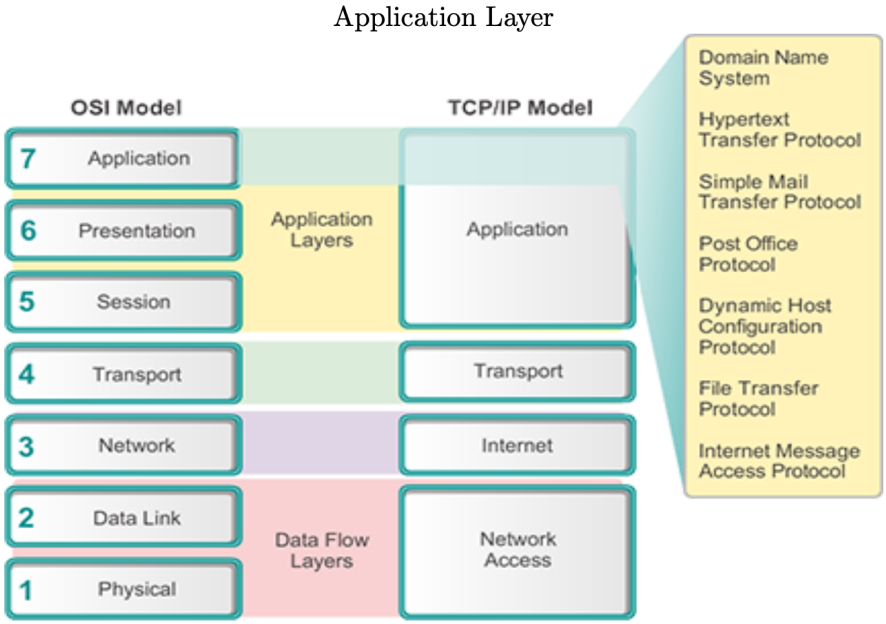
14.1 Self-Similarity

Fractals - class of formula that is infinitely recursive, at any level of detail things look the same.

14.1.1 Problems with Current Mathematical Models

- Poisson Process:
  - On a short time scale looks very bursty
  - Longer time scale flattens to white noise
  - Telephone traffic
- Self-Similar (fractal) process:
  - Always looks bursty
  - Looks the same at any time scale
  - Internet (Ethernet) traffic
  - Hard to predict
- Consequences:
  - Traffic has similar statistical properties at a range of timescales: ms, secs, mins, hrs, days
  - Merging of traffic (as in a statistical multiplexer) does NOT result in smoothing of traffic
  - Bursty data Streams -> Aggregation -> Bursty Aggregate Streams

14.2 Application Requirements



14.2.1 What transport services do Applications need?

- Data Integrity
  - Some apps (file transfer, web transactions) require 100% reliable data transfer
  - Other apps can tolerate some loss
- Throughput
  - Some apps (multimedia) require minimum amount of throughput to be effective
  - Other apps (elastic apps) make use of whatever throughput they get
- Timing
  - Some apps (internet telephony, interactive games) require low delay to be effective
- Security
  - Encryption, data integrity ...

Transport Requirements

application	data loss	throughput	time sensitive
file transfer	no loss	elastic	no
e-mail	no loss	elastic	no
real-time audio/video	loss-tolerant	audio: 5 kbit/s–1 Mbit/s, video 10 kbit/s–5 Mbit/s	yes, 100 ms
stored audio/video	loss-tolerant	same as above	yes, few seconds
interactive games	loss-tolerant	few kbit/s up	yes, 100 ms
text messaging	no loss	elastic	yes and no

14.2.2 Hidden Issues with higher level protocols

- Header Overhead
  - Every protocol layer brings with it its own header packet
  - After a few layers these start to add up - reduce goodput
- Application writers just want to send/receive data
  - Don't want to have to deal with the details
  - Prefer remote procedure call or similar mechanism
  - For real time applications in particular this is very difficult
- Who are you? Where are you? Where is my printer?
  - Identity - connecting to the right machine
  - Roaming - mobile devices in particular

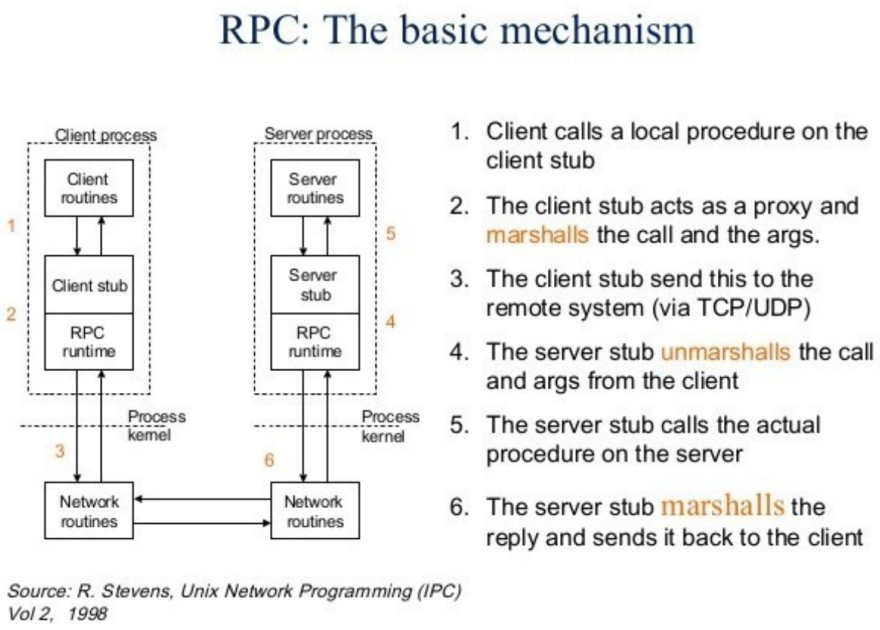
- Some form of microservice architecture is inevitable for applications that want to scale, especially if they can take advantage of high speed network connections
- However, too high a fragmentation can be just as a big a problem as too small (or monolithic architecture)
- i.e. always analyze application requirements carefully, wrt scaling and Fisher Consensus requirements

14.3 Remote Procedure Calls

RPC - Remote Procedure Call
RMI - Remote Method Invocation

14.3.1 Common Application Tasks

- Define protocol between nodes
  - For Client/Server - duplicate code
  - Tedious and error prone
- Define interface for client and server - Sending/Receiving methods for each task
- Handling network issues - Delays, Timeouts, Process Crashes
- RPC goal: reduce complexity, avoid code duplication
- Problems: error handling and network latency



Source: R. Stevens, Unix Network Programming (IPC) Vol 2, 1998

gRPC is a language agnostic, high-performance Remote Procedure Call (RPC) framework.

The main benefits of gRPC are:

- Modern high-performance lightweight RPC framework.
- Contract-first API development, using Protocol Buffers by default, allowing for language agnostic implementations.
- Tooling available for many languages to generate strongly-typed servers and clients.
- Supports client, server, and bi-directional streaming calls.
- Reduced network usage with Protobuf binary serialization.

These benefits make gRPC ideal for:

- Lightweight microservices where efficiency is critical.
- Polyglot systems where multiple languages are required for development.
- Point-to-point real-time services that need to handle streaming requests or responses.

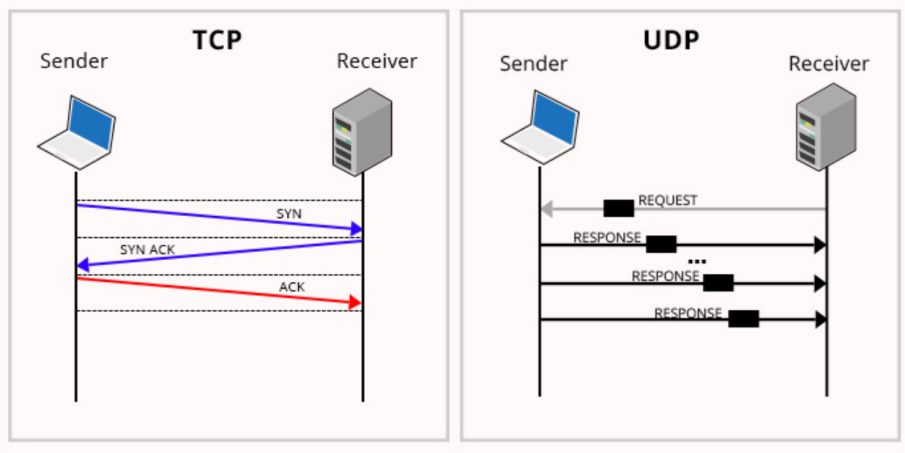
14.4 TCP vs UDP

TCP

- Connect-orientated - State of connection can be examined, Three-way handshake (3 packet) overhead
- Reliable - or connection is dropped
- Data arrives in order sent
- Header size (overhead) is 20 bytes
- Data arrives as continuous stream
- Adapts to congestion (flow control)

UDP

- Connectionless protocol - State must be communicated
- Unreliable
- No guarantees about order of delivery
- No acknowledgments to data
- Header size (overhead) is 8 bytes
- Data is delivered in discrete packets as sent
- No flow control
- May be faster (less overhead)



14.5 TCP vs UDP

TCP

- Long, continuous, reliable flows of data
- Acknowledgments are required by data/application
- Maximize throughput vs congestion
- High error rate on connection
- Ideal Application: file transfer

UDP

- Small packets of data (less header overhead)
- Time sensitive data
- Very low (eg. fiber optic cables) error rate - If it arrives too late, application doesn't need it or don't care about order of delivery
- Ideal Applications: Real time audio streaming, logging

14.5.1 UDP Applications

- Real time critical data - data is useless and can be discarded if it arrives late, no need to retransmit or acknowledge
- Examples:
  - Voice Over IP (VOIP) - Telephony over the internet, late audio packets are useless
  - Live Video Streaming - Dropouts result in loss of picture/sound, annoying but can recover to current picture
  - Remote Logging - to prevent log messages being a source of congestion, otherwise positive feedback issues

14.5.2 TCP or UDP? Practice ELEPHANT WATCH

- What is the Network scope of your application?
  - eg. Local network only, Corporate Network, Internet, Financial?
- Does transport traffic need to be encrypted?
  - How the encryption algorithm works has to be included.
- Which computers does your application run on?
  - Workstation, Laptop, Cell phone, Server, bespoke?
  - What resources do they have?
- How much control do you have over the Network?
  - Home Network - NAT box, ISP provider
  - Company Network - Firewall, ISP provider, internal network
  - ISP Provider - Control over AS domain, peering arrangements
  - Data center - control equipment, but not applications

14.6 Newer Protocols

14.6.1 QUIC (Quick UDP Internet Connections) and TOU

- Proposed to provide end to end encryption
- Includes encryption of protocol headers - Idea is to decouple slow TCP development, Allow end applications to deploy their own protocols, Avoid problems with "middle boxes" - firewalls
- Considerably complicates Network Troubleshooting
- "Middle boxes" are deployed for a reason - Removes control from local network owners

14.6.2 Valid application issues with TCP

- Users very small packets (overhead)
- Low and infrequent data rates - connect overhead
- Doesn't need order guarantees
- Doesn't need reliability guarantees - But may need occasional acknowledgments
- Problems around slow start - Bad performance with wireless networks, Also poor for short-lived connections

14.6.3 Known Issues with TCP (solution/mitigation

- TCP Head of Line Blocking (virtual output queues)
- Cost of 3-way opening handshake (TCP Fast Open)
- Slow start and small initial windows (initCWND10)
- Packet loss causes large backoff (TCP cubic)
- Limit size of send buffer (TCP NOTSENT LOWAT socket option)
- NAT timeout and IP roaming (IPv6?)
- TCP Buffer Bloat (Router upgrades)
- Primary Issue: Slow to roll out over entire internet

14.7 Invalid TCP issues: Dodging Congestion Management

- Increase throughput by not using end-to-end flow control - flow control performed on individual TCP connections so it is possible in some circumstances to improve on throughput by having multiple connections simultaneously and manage the required multiplexing at the application
- Circumvent slow start
- Rise of "selfish UDP" protocols - main core routers reprogrammed to preferentially drop UDP, previously UDP had been prioritized
- Also led to rise of multiple simultaneous TCP connections - UDP was prioritized because it was small, delay sensitive traffic but as it grew TCP starvation began to occur as UDP packets crowded TCP out on the core routers

14.7.1 Multipath TCP (MPTCP)

- Standard TCP builds connection between interfaces
- MPTCP Core idea: use multiple paths between hosts seamlessly
- Examples: WiFi and Cellular links on phone or WiFi and Ethernet on Workstations
- Uses options fields of TCP header - cannot rely on ports, NAT may rewrite, must embed info in protocol

14.7.2 Reliable UDP (RUDP)

- Added to UDP - Acknowledge of received packets, windowing and flow control, retransmission of lost packets, over buffering (faster than real-time streaming)
- Beloved by Game Developers
- TCP Issue: Head of Line blocking - delay to stream if packet is lost, at least RTT*2 because all packets are delivered in order
- eg. Series of packets with bullet info - one bullet can be "lost", others would still hit, insensitive to small amounts of out of order delivery

14.7.3 RTP: Real-time Transport Protocol

- Designed for video and real time streaming
- Used in conjunction with RTCP (Real-time Transport Control Protocol)
- Runs over UDP, widely used for VOIP and Video Streaming
- Supports Jitter compensation, detection of packet loss and out of order delivery, multiple destinations via IP Multicast
- Requires Intelligence in application to handle loss and order issues

14.7.4 Data Center TCP (DTCP)

- Data centers are interesting places - Can assume all routers, switches etc. under single control
- Issues with very short term bursts of traffic - Typically due to MapReduce clusters, All clients reporting back at same time, Over by the time standard congestion mechanisms take effect.
- Modifies Explicit Congestion Notification (ECN) - Estimates fraction of bytes encountering congestion - Modifies TCP congestion window based on that.

14.7.5 Stream Control Transmission Protocol (SCTP)

- Originally a Telephony oriented protocol
- Combines features of TCP and UDP
- Multihoming (different IPs) but does not load share on them - Fault tolerance only
- Supports multiple data streams
- Avoids head of line blocking

- Provides automatic Heartbeat control
- Very little actual deployment - WebRTC

14.7.6 Unity Networking (Games)

- Real-Time Transport Layer - optimized UDP based protocol, multi-channel design, supports quality of service levels/channel, client-server or peer-to-peer
- Also provide Internet Services - servers in their cloud to provide contact points (matchmaking, relay server (NAT hole punching))

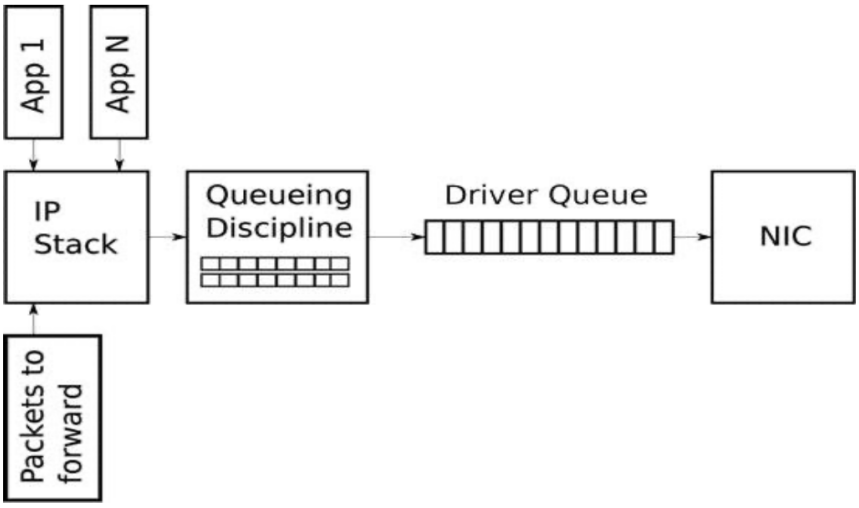
14.7.7 High Level API

- Message handlers
- Distributed object management
- State synchronization
- Network Identity
- Network Proximity Checking
- Host migration
- Provides rudimentary game security
- Supports local and server authority where the object is synchronized

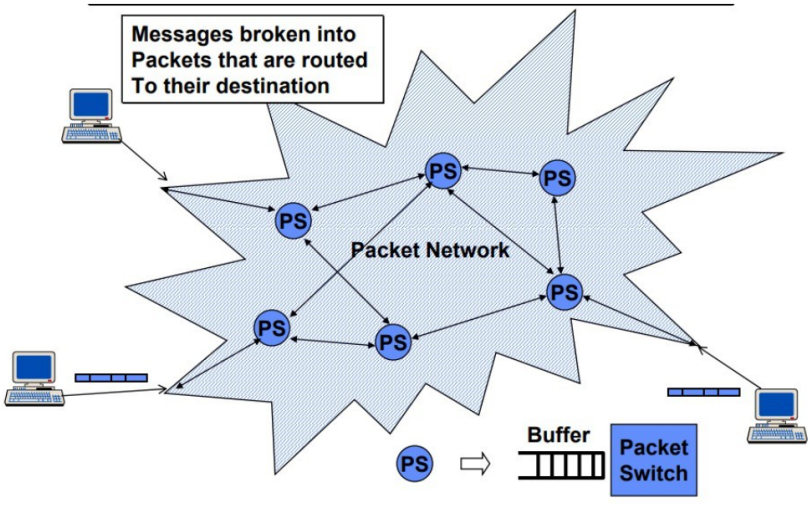
15 Lecture 15: Queuing

15.1 Buffers and Networks

Queuing in the Linux Network Stack



Internet



It is mathematically necessary to have spare capacity in order to prevent congestion

15.1.1 Problems and solutions

- Buffers are necessary but also problematic - increasing buffer size reduces packet loss, increases overall latency, decreases TCP's ability to detect and react to congestion
- Eliminate buffers as much as possible - central routers use switched backplanes, minimize copying packets from buffer to buffer
- Manage queues carefully - engineering application buffers for desired characteristics
- Mismatches often occur - Windows and Linux allow kernel buffer sizes to be altered

Queues occur when short term demand for service exceeds capacity (i.e. when the arrival rate or the rate for a system is variable. Secret of queuing theory: must have significant excess capacity to avoid queues.

15.1.2 Strengths and Weaknesses

- Queuing models are based on simplifications and approximations of reality
- But results are often useful as a back of the envelope calculation - pick most conservative assumptions, round up or down generously, treat as estimate/order of magnitude
- Now possible to calculate for some dynamic systems
- Can also be simulated
- Provide useful bounds

15.2 Queuing Models

Arrival rate $\lambda >$ Service rate μ :: Trouble!
Arrival rate $\lambda <$ Service rate μ :: May be in trouble
Arrival rate $\lambda <>$ Service rate μ :: Probably in trouble

15.2.1 Kendall's Notation - A/S/C/K

- A - Arrival rate/process (λ)
- S - Departure rate/service time (μ)

  - M: Markov (exponential distribution)
  - D: Deterministic
  - G: General (arbitrary distribution)
- C - Number of parallel servers in the system
- K - Maximum size of queue (may be assumed to be ∞
- FIFO/FCFS First in First Out

15.2.2 Little's Law

$L = \lambda.W$
L: Long term average number of customers in a stationary system
λ: Long term effective arrival rate
W: Average wait time

15.3 Example

15.3.1 M/M/1 Queue

Airport runway for arrivals only - Arriving aircraft join a single queue for the runway, Assume exponentially distributed service time, $\mu = 27$ services/hour, Arrivals with a rate $\lambda = 20$ arrivals/hour.

1. $W = \frac{1}{\mu-\lambda} = \frac{1}{27-20} = 1/7hour = 8.6minutes$
2. $L = \lambda.W = \frac{\lambda}{\mu-\lambda} = \frac{20}{27-20} = 2.9aircrafts$
3. $W_q = W - \frac{1}{\mu} = \frac{1}{\mu-\lambda} - \frac{1}{\mu} = \frac{1}{27-20} - \frac{1}{27} \approx 6.4minutes$
4. $L_q = \lambda.W_q = \frac{\lambda^2}{\mu(\mu-\lambda)} = \frac{20^2}{27(27-20)} \approx 2.1aircraft$

Increase arrival rate to $\lambda = 25$ arrivals/hour

1. $W = \frac{1}{\mu-\lambda} = \frac{1}{27-25} = 1/2hour = 30minutes$
2. $L = \lambda.W = \frac{\lambda}{\mu-\lambda} = \frac{25}{27-25} = 12.5aircrafts$
3. $W_q = W - \frac{1}{\mu} = \frac{1}{\mu-\lambda} - \frac{1}{\mu} = \frac{1}{27-25} - \frac{1}{27} \approx 27.8minutes$
4. $L_q = \lambda.W_q = \frac{\lambda^2}{\mu(\mu-\lambda)} = \frac{25^2}{27(27-25)} \approx 11.6aircraft$

15.4 Bufferbloat

Internet Buffers are typically invisible until congestion hits, for a typical end device buffering is being performed by the application, kernel, Network Interface Card, Router/LAN/WiFi, Core Router(s).

15.4.1 Mitigations

- "Tail Drop" - wait until buffer is full, drop new packets - problem: several seconds or more before congestion is signaled
- RED: Random Early Detection/Discard - as queue grows probabilistically drop incoming packets before they enter the queue, not widely deployed
- AQM - Active Queue Management: probabilistically drop packets as queues grow, cake algorithm now widely deployed on routers
- ECN - Explicit Congestion Notification

16 Lecture 16: Quality of Service

16.1 Introduction

16.1.1 What is Quality of Service

- Ability to provide different priority to different applications
- Goal is to provide guarantees of performance/traffic type
- Takes several different forms: Resource reservation (dedicated bandwidth), traffic shaping/policing, scheduling, congestion avoidance
- The only way to guarantee QoS is to drastically over-provision the network

16.1.2 Quality of Service (QoS)

- QoS needs depend on position in network
- Internet providers: manage peering agreements, connectivity between AS's, manage customer relationship (bandwidth customer is using vs paying for)
- Companies: control over internal network, internal video conferencing, company services, can always rent dedicated connections
- End users: control over their connection/network, prioritize certain kinds of traffic, bandwidth allocation

16.2 QoS Types

16.2.1 Individual Traffic Flows: Four Primary Requirements

- Bandwidth
- Delay - Latency, causes:
  - Speed of light is 300000 km/s

  - Number of hops (routers/other devices) - each hop is a CPU delay and potential queue
  - Serialization (function bandwidth) - how long it takes to put the packet onto the wire, dominates short distances
- Jitter - variation in arrival times, causes:
  - Congestion
  - Errors/Lost packets (retransmission delay)
  - Fragmentation - data being sent in too large chunks
- Loss - dropped/discarded packets, causes:
  - Transmission media - Copper, wireless
  - Congestion Delays
  - Bandwidth (insufficient)

Traffic Type Requirements

Application	Bandwidth	Delay	Jitter	Loss
Email	Low	Low	Low	Medium
File sharing	High	Low	Low	Medium
Web access	Medium	Medium	Low	Medium
Remote login	Low	Medium	Medium	Medium
Audio on demand	Low	Low	High	Low
Video on demand	High	Low	High	Low
Telephony	Low	High	High	Low
Videoconferencing	High	High	High	Low

16.2.2 ATM (Asynchronous Transfer Mode)

- 1990's circuit and packet switching technology - Heavily influenced by phone system requirements, i.e. continuous long duration realtime audio
- Offered multiple QoS Types
- 53 byte cell size
- Allocated bandwidth for different traffic types

- Extremely complex to configure

Different categories of QoS guarantees (from ATM) constant bit rate (CBR), real-time Variable bit rate (rtVBR) (compressed video conferencing), non real time Variable Bit Rate (nrt-VBR), UBR - unspecified Bit Rate (models Internet's best effort service), ABR - Available Bit Rate (network switches calculate available bandwidth).

16.2.3 QoS Issues

- QoS is not required when there is adequate bandwidth
- By implication - QoS always involves decisions that penalize some traffic
- Applying QoS may well make situation worse by increasing overheads - reserved bandwidth (less efficient use), prioritized traffic (traffic starvation), TCP congestion behavior (dropped packets)
- Some problems can be solved with buffers - constant bit rate vs 1-2 minute delay buffer

16.3 QoS Algorithms

16.3.1 Burstiness

- Tendency of packet traffic to arrive in clusters or bursts
- Caused by buffering/queue behavior en route and changes in bandwidth along the route
- Adverse effect on queuing system behavior
- Exacerbating packet losses can push network into congestion

16.3.2 Leaky Bucket

Leaky bucket parameters - bucket size (set to max size of burst that can be tolerated, overflows are typically discarded), drain rate (set to the average rate of the incoming flow)

- Single-server queuing system with constant service rate (M/D/1)
- Can be set to send a max number of packets (bucket size)
- Bounds traffic rather than exactly control
- Trade off: token bucket size against bandwidth share
- Bigger bucket - larger (worse-case) delays, more jitter
- More bandwidth - less efficient

(r, T) Traffic Shaping - enforces traffic transmission rate, breaks flow into fixed size chunks, excess traffic is delayed or dropped (also known as throttling but if flow is insufficient may be padded out).

16.4 Scheduling Strategies

16.4.1 Packet Scheduling

- Requirements to offer a performance guarantee - must be able to reserve sufficient resources, packets can take any route
- Packet Scheduling Algorithms, reserve: Bandwidth, Buffer space, processing/CPU

16.4.2 FIFO - First in First Out

- Drop new packets when full - tail drop
- Heavy flows may starve other traffic
- Simplest - less CPU
- Rarely used as primary scheduling algorithm

16.4.3 Priority Queuing

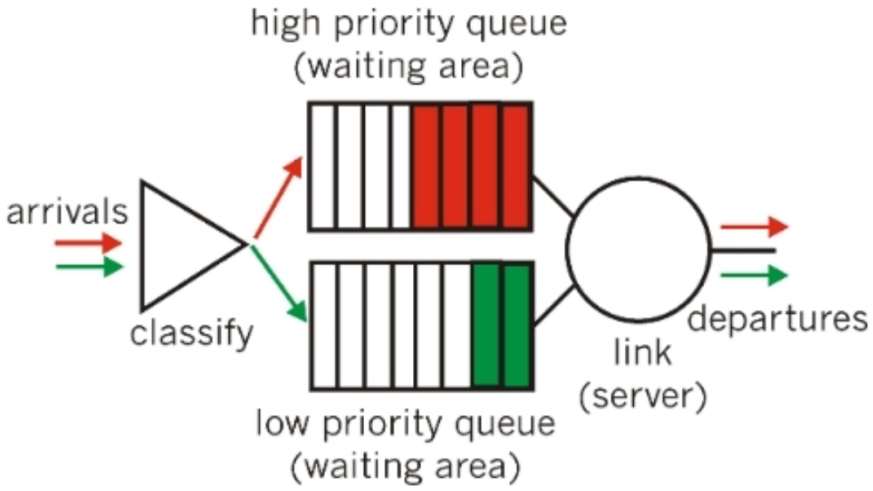


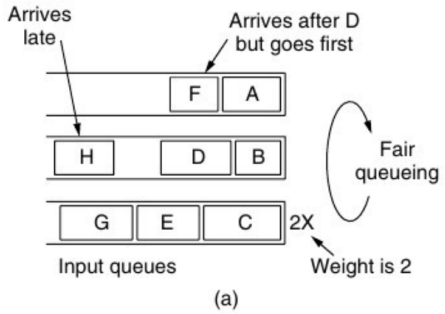
Figure 6.6-3: Priority queuing model

16.4.4 Round Robin (aka Fair Queuing)

Figure 5-30. Round-robin fair queueing.

- Separate buffers for each connection
- Router sends packet from each buffer in turn
- Imposes constant traffic rate on all flows
- Prefers larger packets

16.4.5 Weighted Fair Queuing



Packet	Arrival time	Length	Finish time	Output order
A	0	8	8	1
B	5	6	11	3
C	5	10	10	2
D	8	9	20	7
E	8	8	14	4
F	10	6	16	5
G	11	10	19	6
H	20	8	28	8

(b)

Figure 5-31. (a) Weighted Fair Queueing. (b) Finishing times for the packets.

16.4.6 Priority Queuing

- Traffic is tagged with a priority - aggregates
- High priority traffic is sent first
- Low priority traffic may be completely starved - often combined with some kind of guarantee for low priority

16.4.7 Deficit Round Robin

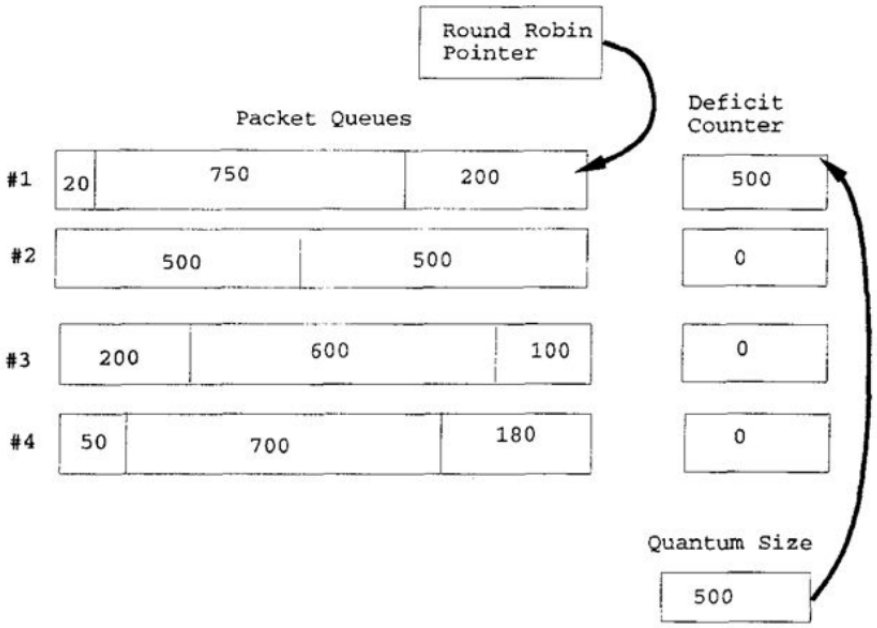


Fig. 2. Deficit round-robin: At the start, all the DC variables are initialized to zero. The round-robin pointer points to the top of the active list. When the first queue is serviced, the Q value of 500 is added to the DC value. The remainder after servicing the queue is left in the DC variable.

16.4.8 Weighted Random Early Detection (WRED)

- Congestion avoidance queuing discipline
  1. Packet is queued if queue size is below threshold
  2. Probabilistically dropped or queued if between
  3. Dropped if queue size greater than max
- Interacts with TCP/IP end point congestion management - Dropped packets signal endpoints to back off traffic rate

16.4.9 Fair Queuing and Bufferbloat

- Queuing strategies intersect with TCP Flow Control - Buffer management - TCP Reno and Cubic effectively grab queue capacity
- FIFO interacts with buffer management - Increases latency and jitter in congestion, Use of Random Early Detect (RED)

- Fair queuing strategies would change behavior of algorithms
- Nobody controls all the pieces of the puzzle

17 Lecture 17: Peer To Peer

17.1 Overlay Networks

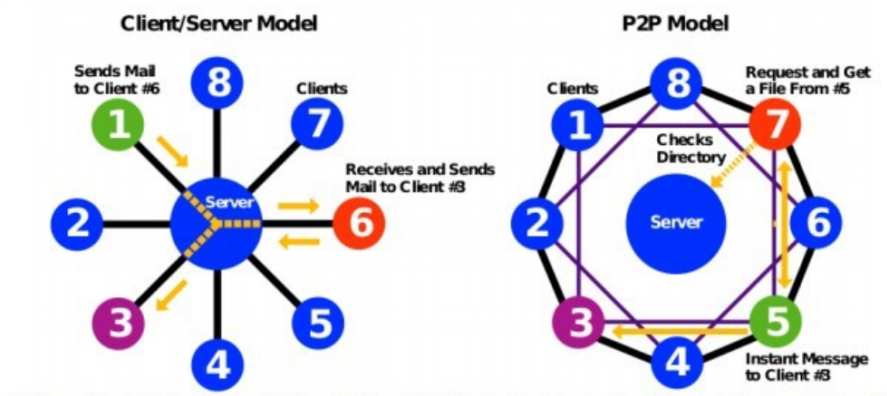
17.1.1 P2P Overlay

- TCP IP Connection to Neighbors
- Periodic ping - KEEPALIVE - solves TCP/IP disconnect timeout issues
- Can disconnect/connect "at will"
- Application has to handle P2P stability/churn, connectivity, application routing

17.2 History

Networking in theory: P2P leverages the end nodes capabilities, escapes from the "dumb terminal" paradigm and in particular allows end nodes to communicate with each other.
In practice more complex - status of intelligent browsers vs servers.

Client/Server vs P2P (simplified)



[David Barkai et al. An Introduction to Peer-to-Peer Computing]

17.2.1 P2P "Principle" - Core Concepts

- Self-organizing - no central management

- Resource sharing by end devices (i.e. files, CPU, data)
- Peers all have same capabilities and are more or less equal
- Large numbers of peers in network
- Based on voluntary collaboration (eg. Wikipedia)

17.2.2 P2P in Principle vs in Practice

Principle

- Scales as L $\sqrt{N}$ where L is number of link connections
- L typically a function of bandwidth and CPU capacity
- Significant issues with Fisher Consensus

Practice

- For sufficiently large L and sufficiently small N scaling issues may not be apparent
- Otherwise need to self-organize into group of groups structures
- Or increase L, reduce N, accept limit on scaling

17.2.3 Structured vs Unstructured P2P Networks

Structured

- Have a defined topology
- Peers and objects in network have identifiers
- Distributed indexing provides object location

Unstructured

- Objects have no special identifier
- Location of objects is not known apriori
- Each Peer responsible for its own objects
- Some form of search protocol used to locate

17.2.4 Technological Impediments

- NAT
  - Due to IPv4 restrictions
  - Blocks direct access to local hosts
  - eg. video conferencing at home
  - Skype - P2P network, used connected hosts to relay NATted hosts
- Firewalls
  - Block incoming connections
  - Port 80 sometimes used to circumvent

- Asymmetric Bandwidth
  - P2P creates demand for uplink as well as downlink bandwidth
  - ISP's over provision uplink capacity in particular
  - Due to use of caching to minimize downlink usage

17.2.5 Technological Challenges

- Linking hosts together scalably - random association may link widely separated hosts, true scaling requires some of link related localism
- Hosts must be able to join and leave ad hoc - network churn can be very destabilizing
- Locating objects/hosts/facilities within P2P system - same issues as in Internet, addressing and routing
- Tragedy of the Commons Issues: Avoiding leeching, distributing load equitably, file sharing - retaining complete copies across network, finding and keeping rare items

17.3 Addressing and Routing

17.3.1 Centralization is Easy

- Single point to find things - Database or similar
- Can partition data across multiple databases - each DB has its own slice of the entire dataset, front end to direct requests to correct server, "Divide and Conquer"
- No Consensus Issues
- Poor Fault tolerance - can tackle with replicating servers, single point of attack for adversary (copyright holders and musicians)

17.3.2 Distribution is Hard

- Replication and fault tolerance usually a given
- Scaling usually not a problem - organized distribution topology can be designed or will emerge
- Finding thins is hard - peers have to have ways to query other peers efficiently, risk of causing broadcast storms
- Any kind of guarantee is difficult - Fisher consensus

17.3.3 Distributed Hash Table Algorithms(DHT) 1997

- Design Goals - Index and distribute entire dataset in network, scalable, handle node churn (rapid joining and leaving of network)
- Examples (original 4): Content Addressable Network, Chord, Pastry. Tapestry

17.3.4 Overlay Networks

- DHT's used to create an overlay network for the P2P "cloud"
- Node has to be able to efficiently join and leave network
- Each node uses the algorithm provided by the DHT
- Picks neighbors and consequently creates a topology
- Based on some idea of "closeness"
- Hash keys are created to identify similar objects - Distributed among nodes, Used to more or less efficiently route to desired destination

17.4 DHT Based P2P Algorithms

17.4.1 Common approach

1. Assign random (160-bit) ID to each node
2. Define a metric topology on the 160-bit numbers i.e the space of keys and node IDs
3. Each node keeps contact info to O(logn) other nodes
4. Provide a lookup algorithm which finds the node whose ID is closest to a given key - need a metric that identifies closest node uniquely
5. Store and retrieve a key/value pair at the node whose ID is closest to the key

17.4.2 Content Addressable Networks (CAN)

- Search space: d-dimensional coordinate space (on a d-torus)
- Each node owns a distinct zone in the space
- Each node keeps links to the nodes responsible for zones adjacent to its zone (in the search space) – ~2d on avg
- Each key hashes to a point in the space

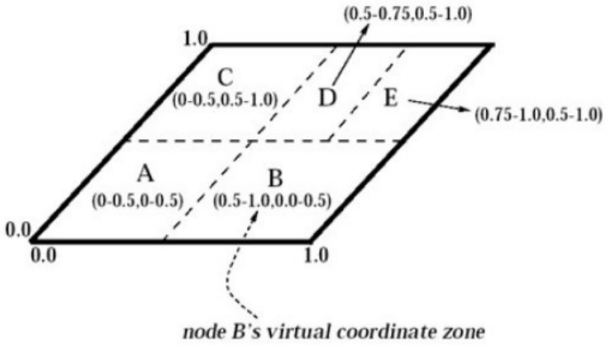


Figure 1: *Example 2-d space with 5 nodes*

* Figure from "A Scalable Content-Addressable Network", S. Ratnasamy et al., In Proceedings of ACM SIGCOMM 2001.

Node Joining: Space is partitioned over a gird, first node to join owns gird, second node splits with first, third node is attached to Node 1 or Node 2 at random (takes half their space, rinse, repeat, recycle), on leaving a node's space is merged with adjacent node

17.4.3 Chord

- P2P DHT with O(log n) search time
- Each node has a key, which is the hash of the node's identifier - eg. Host IP Address (reusing existing structure)
- Each data item has a key which is the hash of it's file identifier
- Any hash function can be used - paper used SHA-1 - SHA-1 gives a 160-bit hash value, N = 2^{160}
- N hash values are conceptually arranged in a circle, modulo N
- Each node is found at the location on the ring equal to it's Key (hash)
- Each data item is stored is the smallest value >= to the data key (modulo N)
- Each node keeps local "finger table" - Stores node keys, K+1, K+2, K+4, K+8 (K is node's key)
- To find a key, find closest node in table without being > key - Forward query to that node

- Results in a deterministic way to find location based on data

17.4.4 Node Joining and Leaving

- When a node joins, it must create its own finger table and other nodes must adjust theirs
- Original paper did not describe how this is done
- Requires some kind of bootstrapping node
  - Typically implemented to use a node in the ring
  - Still requires a way to find that node
  - Reddit may be necessary, but is not considered sufficient

17.4.5 Pastry - read more in slide packet

- P2P structure based on Plaxton routing
- Developed at Microsoft Research and University of Washington
- Data items have unique 128-bit IDs $(0..2^{128} - 1)$
- For routing, treated as sequences of digits in base 2b - Typically b = 4
- Once again, IDs arranged as a circle
- Node IDs randomly generated as node join
- Routing is based on numeric closeness of IDs - Node will choose node in its table with longest match

Kademlia - more in slide packet

17.4.6 Data Structures

- Contact - a pair of node ID and IP:UDP_port
- k-bucket - a container for no more than k contacts, operations place contact and remove contact
- Routing table - operations place contact and remove contact, constrained tree of k-buckets, each bucket responsible for a range of the node ID space

17.4.7 Joining, Leaving, and Refresh

- Join
  - Needs IP of node already in the Network
  - Computes random ID until it finds unused one - Can persist previous ID, But no long-term memory in the network
  - Borrow some contacts from node already present
  - Find self - populates other k-buckets with this node's ID
- Leave - No action - picked up at refresh or timeout
- Hourly k-bucket refreshes if necessary

17.5 Futures

17.5.1 P2P Networks: Current State

- A lot of different P2P protocols - Mostly DHT derived in some form
- Each is standalone - no interworking, not even gateways
- i.e. throwback to beginning on Internet
- Scaling, Routing and Reliability continue to be research issues
- Lot of work still to be done

18 Lecture 18: Network Security

If builders built buildings the way programmers wrote programs, then the first woodpecker that came along would destroy civilization.
Gerald Weinberg (circa 1980)

18.1 Attack Surface

Attack Surface: Attack surface of a software environment or system is the total of different points (attack vectors) where an unauthorized user can try to attack the software or system.

18.2 Defense Against the Dark Arts

18.2.1 Backups

1. Protect against accidental data loss
2. Protect against modification

3. Assist in identifying incidents
4. Regular backup to unattached media
5. Regular offsite backups
6. Distribute among computers

18.2.2 Physical

1. Are servers in physically secure areas?
2. UPS (Uninterruptible Power Supply) Tested?
3. Fire? Earthquakes?
4. Is there a complete contingent backup site plan?
5. When was it last tested?
6. Is paper waste being securely shredded?
7. Disposal of old hardware?

18.2.3 Defense Strategies

Defense in Depth

- Physical Access
- Network
- Hosts or Operating System
- Applications

Defense in Breadth

- Split up areas and protect separately
- Whitelist and Blacklist
- Identify high priority targets
- Physically disable problematic interfaces eg. USB
- Do accounting really need to browse the Web?

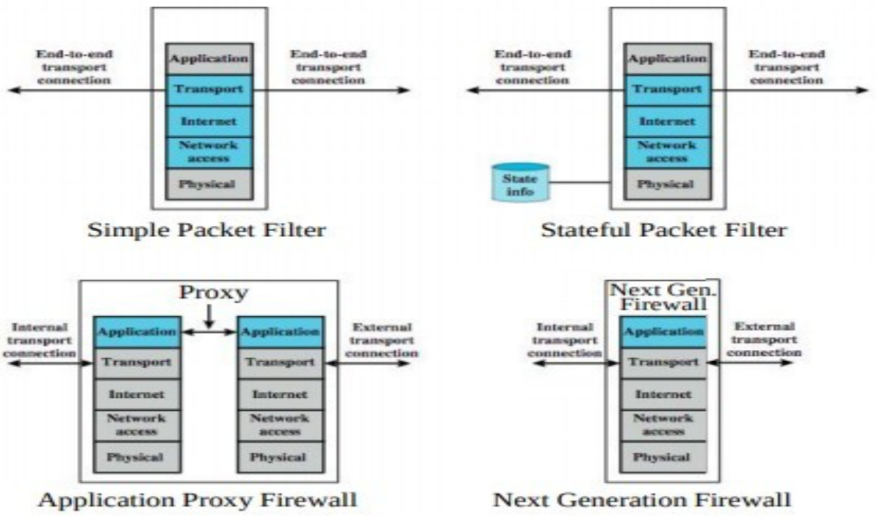
18.3 Operational Security

18.3.1 Firewalls - Overview

- All traffic entering or leaving must pass through firewall
- Must define criteria for what is (un)authorized
- Effectiveness of firewalls depends on specifying authorized traffic in terms of rules - what to let through, what to block, traffic patterns have to obey those rules

- Firewall itself must be effectively administered - updated with latest patches, correctly configured
- Firewalls can be implemented in both hardware and software or a combo of both
- Firewalls frequently become a cost/performance bottleneck

Types of Firewalls



18.3.2 Router-based Packet Filter

- A packet filter is a network router that can accept/reject packets based on headers
- Packet filters examine each packet's headers and make decisions based on attributes (source or destination IP Addresses/ Port numbers, Protocol)
- Unaware of session states at internal or external hosts
- High speed but primitive filter

18.3.3 Host-based Packet Filters

- Hosts can also perform packet filtering - stops some unwanted traffic reaching applications
- Kernel Firewalls

18.3.4 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

18.3.5 Application Layer Proxy

1. External client sends a request to the server which is intercepted by the outwards-facing firewall proxy
2. Inwards-facing proxy sends request to server on behalf of client
3. Server sends reply back to inwards-facing firewall proxy
4. Outwards facing proxy sends reply to the client

- Client and server both think they communicate directly with each other, not knowing that they actually talk with a proxy
- The proxy can inspect the application data at any level of detail and can even modify thee data

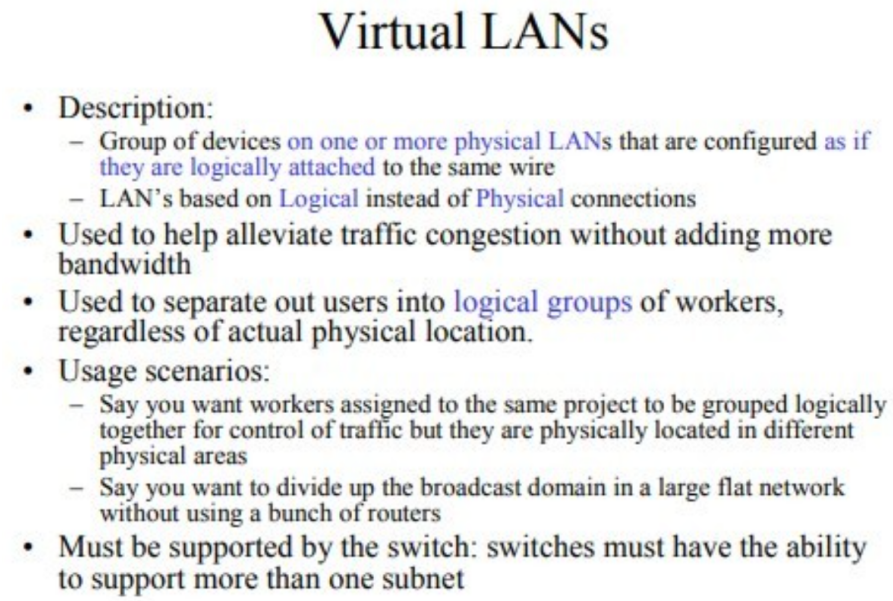
18.3.6 Next Generation Firewalls

- Inspects payload in end-to-end or proxy application connection
- Support specific application protocols - e.g. http, telnet, ftp, smtp etc., each protocol supported by a specific proxy HW/SW module
- Can be configured to filter specific user applications - e.g. Facebook, YouTube, LinkedIn, Can filter detailed elements in each specific user application
- Can support TLS/SSL encrypted traffic inspection
- Can provide intrusion detection and intrusion prevention
- Very high processing load in firewall – High volume needs high performance hardware, or else will be slow

18.3.7 Deep Packet Inspection (DPI)

- Deep Packet Inspection looks at application content instead of individual or multiple packets
- Deep inspection keeps track of application content across multiple packets
- Potentially unlimited level of detail in traffic filtering

18.3.8 Virtual LANs



See more in slide packet.

19 Lecture 19 - Intro to Cryptography

As computers get more powerful larger and larger keys are required.

19.1 Terminology

Out of Band (OOB) - transmitted using a different media to the main communication (Internet access to banks, but cellphone authorization to login).
Plaintext - Unencoded original message.
Nonce - Arbitrary or random number that is only used once, used to provide non-predictability.

Kerckhoff's Principle - A cryptosystem should be secure even if everything about the system, except the secret key, is public knowledge.

Cryptographic key - piece of information (a parameter) that determines the functional output of a cryptographic algorithm

- Typically a string of bits of pre-determined length
- But anything can be a key

- **Symmetric key:** same key is used to encrypt and decrypt
- **Asymmetric key:** two different keys, one to encrypt, another to decrypt

Substitution cipher - substituting one thing for another (like caeser shift)

19.1.1 Codes vs. Cipher

- Code is a slightly broader term than Cipher
- Code is very specifically used as a term for "Block substitution"
- Ciphers are based on some form of permutation algorithm and a private key
- Attacks:
  - Analyze letter and word frequency
  - Compare known plaintext with ciphered text
  - Randomness in ciphered text is highly desirable
- Codes map digits (or letter groups) using a dictionary like codebook - need codebook or a large sample of ciphertext/context to break but much more overhead in setting up

19.2 One Time Pad

- Only unbreakable protocol
- Randomly generated "infinite" length key (pad)
- Combined with plaintext
- Both sides must have a copy of the pad (Key exchange problem)
- No other relationship between plain and cipher text
- Must only be used once, source pad must remain secret
- Because the key is perfectly random so is the ciphertext - no way to extract information, "Entropy" of message
- Suppose the key isn't perfectly random - this can be broken although it is non-trivial (difficult)
- Key length must be equal to or greater than the text
- There is no protection of the message - no way to know if the message has been changed

Challenges of widespread Encryption Use

- Encryption and Decryption users must be fast - considerable overhead on more elaborate schemes, infeasible for many users
- Decryption by attackers must be as impossible as practical
- Issue of key distribution and how to agree on a key in the first place

19.3 Public Key Cryptography

Public Key Cryptography - each user has a pair of keys, one key that is public, another that is private kept strictly secret, public key used to encrypt, private key used to decrypt.

Public Key Encryption

- Asymmetric - different key used for encryption/decryption
- Public key is available to encrypt message
- Private key used to decrypt
- Relies on mathematical functions with no efficient solution
- Typically incorporates random numbers - hence the source of random numbers can be a problem
- Typically used for short messages

19.4 Diffie-Hellman

There is a public base (g) and public modulus, person A has a private key (a) and person B has a private key (b), then person A has a public key (A) and person B has a public key (B)

19.5 RSA

Security relies on the difficulty of factoring large composite numbers

20 Lecture 20: Bluetooth

20.1 Introduction

Bluetooth solving problem of dumb devices - simple connectd "smart" objects with very limited capabilities.

Low Energy Devices Proliferating

- Mobile Phones

- Headphones, Microphones
- Fitness Trackeres, Health Monitoring, Watches
- Locks/Padlocks
- Keyboard, Mice, other peripherals
- Home Entertainment, Speakers
- Fridges, Lightbulbe

20.2 Protocol Overview

- Low power, near field communication protocol
- Frequency hopping
- Range 1-100m / 1-3Mbps
- Each device has a unique 48-bit identifier (manufacturer assigned)
- Discoverable Bluetooth devices broadcast: name, class, list of services, technical information

20.3 Architecture

- Bluetooth best described as the illegitimate child of a bunch of old telephony protocols, very top down/heavily defined - hard coded approach
- Hardware: Radio, Link Manager, Baseband, Access through Host Controller Interface (HCI)
- Host protocol stack: L2CAP, RFCOMM, BNEP, AVCTP, TCP...
- Profile implementations: Serial Port (SPP), Dial-up, HID...
- HID: Human Interface Device Profile
- SPP: Serial Port Profile
- L2CAP: Logical link control and adaption protocol
- RFCOMM: Radio frequency communication (RS-232 emulation)
- BNEP: Bluetooth network encapsulation protocol (personal area networking PAN), bound to L2CAP
- TCP: Telephony Control Protocol
- AVCTP: Audio/video data transport protocol - audio/video

20.4 Security

20.5 Bluetooth Tools

SEE SLIDES FOR REST, BLUETOOTH NOT ON TEST

21 TEST 2018

1. Computer Networks and the Internet
(a) (4 points) Describe at least two aspects of fiber-optic cables that require caution when they are being physically connected, disconnected or moved, and explain why they are issues.
There is glass in the fiber optic cables which can break so don't bend them too much. You also need to clean them every time you move or disconnect or reconnect them because dust particles can get stuck on them and block the lazer in the center since it is so small and therefore disrupt the communication. Also lazer

(b) (1 point) What is the IPv4 255.255.255.255 address used for?
It represents a broadcast address or place to route messages to be sent to every device within a network.

(c) (2 points) What guarantees does TCP/IP provide for connections using it?
It guarantees that all packages will be delivered or else the connection will be dropped. The delivery will also be delivered in the order it was sent. Also reliability.

(d) (2 points) What is the advantage of Classless Interdomain Routing (CIDR) over the previous Class A, B, C, D method of subnetting.
It slowed down the growth of internet routing tables and slowed down the exhaustion of IPv4 addresses. It therefore improves efficiency of IP address distribution.

(e) (2 points) Which two layers of the four layer network model does the Address Resolution Protocol(ARP) effectively bridge?
ARP works between network layers 2 and 3 of the Open Systems Interconnection model (OSI model). The MAC address exists on layer 2 of the OSI model, the data link layer, while the IP address exists on layer 3, the network layer. -> https://searchnetworking.techtarget.com/definition/Address-Resolution-Protocol-ARP.
It links between data link and network layer, LAN (network layer) and WAN (transport layer).
Use internet model stack.

(f) (1 point) What routing protocol is used between Internet Backbone Routers?
External Border Gateway protocol is used between Internet Backbone routers.

(g) (2 points) Where are the routing tables for the Internet backbone routers calculated?
On each router individually, locally to protect consensus (best way based on RTT).

(h) (2 points) An Internet Certificate Authority(CA) is an entity that issues digital certificates. What does a digital certificate certify, and what is it used for?
It certifies that the public key contained in the certificate is the one that belongs to the one that was issued the certificate aka verify the person that is sending the message is who they say they are. Also gives the receiver the tools to encode a reply back to the sender. Used to obtain a public key to encrypt a message to a specific user.

(i) (2 points) An application uses a symmetric cipher to exchange information. How should two Internet connected endpoints electronically agree on the shared key used by the cipher for encrypting traffic?
They should use RSA (Rivest–Shamir–Adleman) to make a secure shared key electronically. Have to send to another person using asymmetric key (public sharing key method).

(j) (4 points) Explain the TCP starvation problem, and describe its implications for UDP traffic sent over the public Internet.
TCP starvation happens when the traffic is dominated by non-TCP based applications (UDP) which causes congestion. TCP is now priority, because udp is favored because nothing stops it.

(k) (4 points) Explain, including a simple diagram showing how data is transmitted from the end hosts, how statistical multiplexing works between an ISP provider when a 10 Gbps connection is being divided between 3 households, each sold connections of 10 Gbps each (3x oversubscription).
When this connection is divided between 3 households for the max amount of Gbps that the connection has then those three households are sharing that connection. This works with the assumption that they aren't all demanding 10Gbps at the same time and aren't using it for gaming (otherwise it will lag due to the shared connection and the fact that everything is sent as it arrives so when there are a lot of packages being sent there is a lot of lag. NEED DIAGRAM

(l) (2 points) If circuit switching is used for a Voice Over IP(VOIP) link, and each call requires 20kbps, how many calls can be supported on a 200kbps

link?
Circuit switching: dedicated circuit per call. It can support realistically, without oversubscribing, 100/20 = 10 calls at once.

2. Protocols
(a) (4 points) Place the following protocols in the correct layer of the Internet Protocol stack

- UDP
- Ethernet
- DNS
- IP
- ICMP
- RTP(VOIP)
- ARP

Application: DNS, RTP
Transport: UDP, ARP
Internet: IP, ICMP, ARP
Datalink/Link: Ethernet

(b) (4 points) For both of the following application types, explain whether you would use TCP/IP or UDP as the network layer protocol, and why.
i. (2 points) File transfer
TCP/IP would be best here because the connection works like a single stream of data. Also the information will be transferred in the right order and no packages will be dropped which is more important than getting it there fast when it comes to file transfers (the contents are more important than the speed it gets to a person).

ii. (2 points) Voice over IP(VOIP)
UDP because late audio packets are useless. It isn't as important to get each and every packet, just important to get it their in a speedy fashion.

(c) Network traffic can be described by four characteristics:

1. Latency
2. Jitter
3. Loss

4. Bandwidth

For each characteristic, provide a clear explanation of what it is, and describe for a networked application of your choice, how the application's behavior could be adversely affected by this characteristic. You may use the same application for more than one characteristic if you wish.
i. (2 points) Latency - The time it takes for a packet to transfer from sender to receiver. This has an affect in deciding whether to use UDP or TCP because TCP has a longer latency than UDP to ensure more reliability which takes time. When sending audio latency is a very important factor to consider. Also the delay in an application. This can affect an application in the sense that say the sender is expecting an ACK (acknowledgment) from the receiver but the delay in sending is longer than usual. Then the sender won't get an ACK in time and will think that the message got dropped. So it will send the message again, resulting in duplication of packages sent. Shooter game or voice chats.

ii. (2 points) Jitter - variation in arrival time causes a similar problem as with latency (the second problem there) that can cause quality issues for real time packages such as audio. This can affect an application in the sense that say the sender is expecting an ACK (acknowledgment) from the receiver but the delay in sending is longer than usual. Then the sender won't get an ACK in time and will think that the message got dropped. So it will send the message again, resulting in duplication of packages sent. Speaking and echo presentation skips forward, that is jitter.

iii. (2 points) Loss - dropped or discarded packets which cause congestion delays. This can corrupt data. Time to live lag.

iv. (2 points) Bandwidth - describes the maximum data transfer rate of a network or Internet connection. If too much information is sent then it could be fragmented and that increases the chance of loss. Shooter game: low bandwidth = higher latency. Downloading a game slowed down because one bandwidth is weak.

(d) (2 points) The Time to Live (TTL) field in the IPv4 header is decremented by 1 by each router the packet passes though. Why did this cause the IPv4 header checksum to be removed in IPv6?
All of this decreasing the header by 1 with each router the packet passes through causes the checksum to be recalculated with each router it passes through. This takes time and processing power. By dropping it in IPv6 we save processing time and speed up the packet forwarding because nothing needs to be recalculated at every router. Also redundant.

(e) (4 points) The physical layer was part of the OSI reference model, but not explicitly part of the Internet model. Discuss the different ways that the physical differences between Internet access over copper wires, fiber-optic cables and WiFi access can affect application layer handling of network traffic.
Copper - drops - low capacity, lots of error, bottleneck, Cross talk(electrical interference between two copper cables close together without casing), limited length (100 m max), requires repeaters, high error rate, cheap.
Fiber-optic - breaks - Lower error rate, longer range, fast, thinner, fewer repeats, breakable due to the glass, requires specific cleaning and when moving them you need special training to handle them correctly without damaging them.
WiFi - bursts of errors - Nothing stops a broadcast being performed on a point-to- point network, send same information to all nodes, lower info capacity than point-to-point network, but high loss due to interference, frequency use has to be carefully regulated, higher delays + jitter, lower security (more encrypted now).

3. Network Engineering (Remember to show your working and state any assumptions)
You need to urgently transfer a 20GB medical database from Reykjavik University to Landspitali University Hospital, a distance of 2km.
(a) (4 points) What is the fastest possible time it would take to transfer this dataset assuming an error-free 100 Mbps connection, a 10% protocol data overhead and ignoring any additional delays caused by TCP/IP rate adaption?
20*1.1/

$$t = \frac{s}{v} = \frac{20\ GB * 1000 \frac{MB}{GB} * 8 \frac{b}{B} * 1.10}{100\ Mbps} = 1760s = 29.33\ minutes$$

(b) (2 points) Owing to the General Data Protection Regulation(GDPR) all medical data being transferred over a network would have to be encrypted first. Assuming that encrypting the data doubles the transfer time, and that alternatively it takes 20 minutes to write the database to a Blueray disk, would it be better to perform the transfer using the Internet, or the fastest student cyclist (it's 4pm), you have standing by?
time to read + time to write = 40+x ¡ 80 min 40 s

4. Network Layer

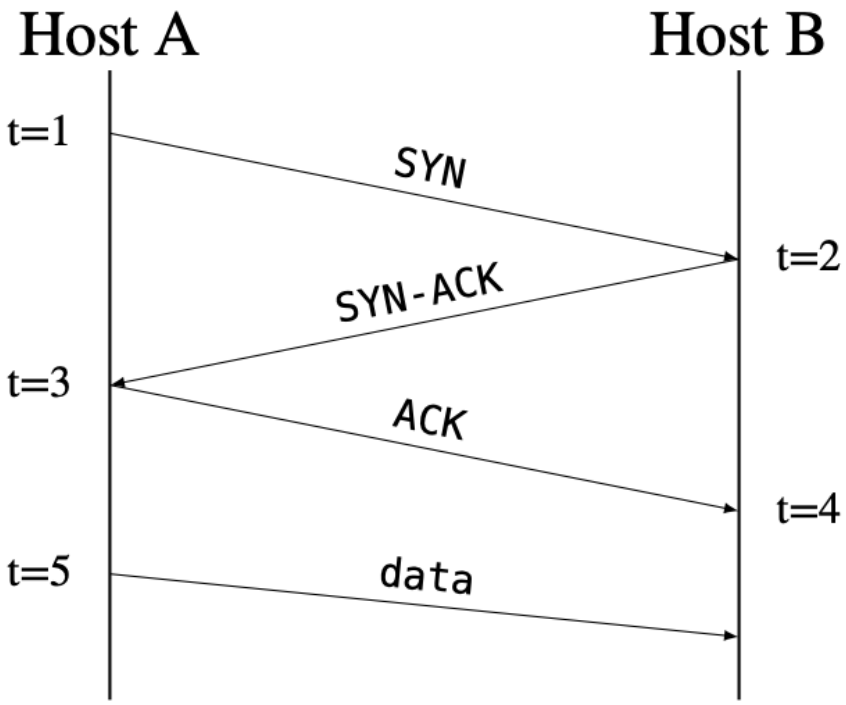


Figure 1: TCP/IP 3-Way Handshake

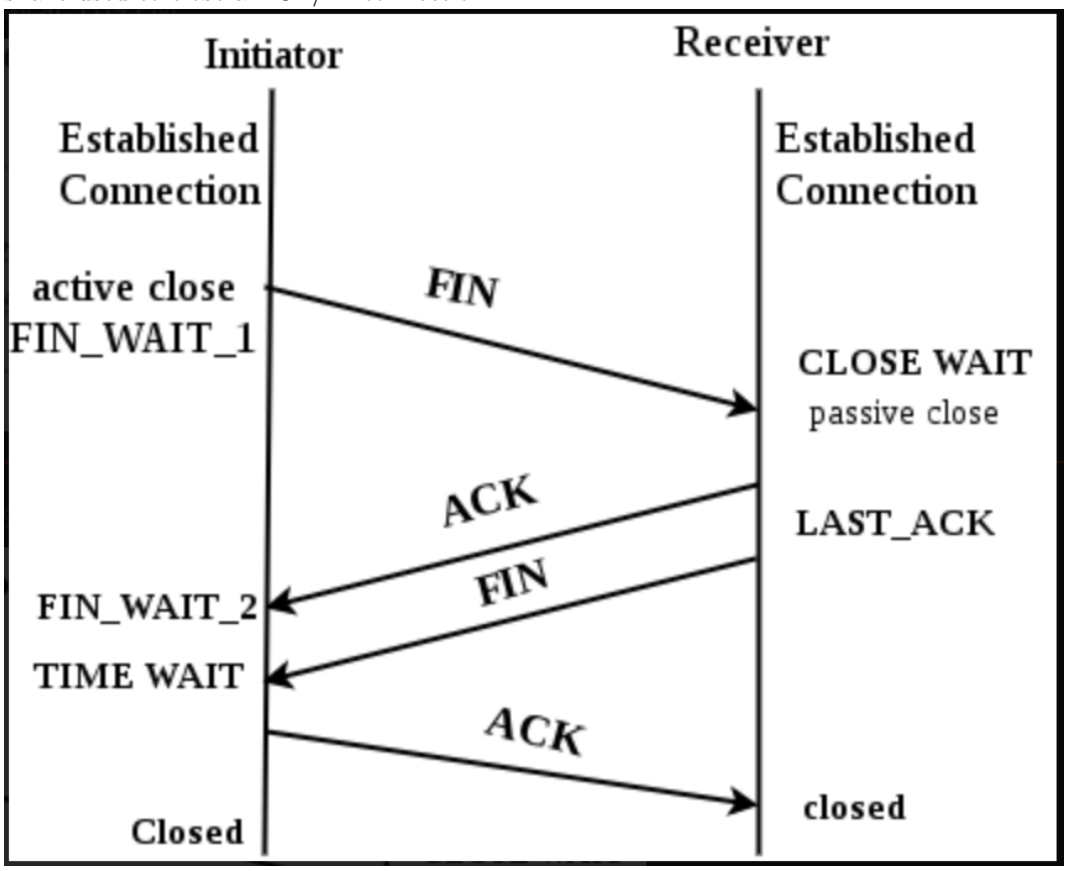
(a) In TCP/IP both sides of the connection maintain separate buffers in order to re-order delayed packets, and resend lost packets using the sliding window protocol.
i. (1 point) What does the sequence number on each segment in the TCP header identify?
Each sequence number identifies the byte in the stream of data from the sending TCP to the receiving TCP that the first byte of data in this segment represents. Amount of data already sent.

ii. (1 point) What value does the acknowledgment field in the TCP header hold?
A 32 bit number that contains a receipt of any prior bytes plus what it expects to get next. Sequence number (previous bytes) + 1 (next byte).

iii. (3 points) Assume that A starts with a sequence number of 0, supply

the sequence and ACK values for the 3-way handshake shown in Figure 1
t=1: SYN=0, ACK=0
t=2: SYN=1, ACK=0
t=3: SYN=2, ACK=1
t=4: SYN=2, ACK=2
t=5: SYN=2, ACK=2
NOT ON TEST.

(b) (2 points) Draw a similar diagram to Figure 1 showing the 4-way handshake used to close a TCP/IP connection.



(c) (2 points) After setting up the connection, A sends B a 10 byte data segment at t=5. Assume that this segment is never delivered. What will A do, and why?
A will send it again if it doesn't get an ACK in time by the time the TTL expires.

(d) The minimum length of an IP header without options is 20 bytes, as is the minimum length of a TCP header (also without options).
i. (1 point) Assuming no options are being used when sending the 10 byte data segment to B, how many bytes does A actually send to B including any packet headers?
It sends 10 bytes of data plus 20+20=40 bytes of header so 50 bytes of data in total. Assuming he sends it twice then 100 bytes, most likely needs to send it twice.

ii. (2 points) If this data segment is the only data sent between A and B, and an orderly tear down of the connection occurs, what is the total number of bytes used between A and B to successfully deliver the 10 byte data segment, assuming no errors occur in transmission.
100 + 40 byte ACK + 3 way handshake (120) (open connection) + 4 way handshake (160) (close down connection) = 420 bytes. In total that is 420B to send one data packet of 10B.

(e) Assume that this is a data recording device, providing a daily update for a sensor network, over a satellite data network, which charges 100Kr per byte transmitted. The length of a UDP header is 8 bytes.
i. (2 points) How many bytes would be required in order to send a UDP packet with the same amount of data, assuming no errors in transmission?
10 bytes (data) + 8 bytes (UDP header) + 20 bytes(IP header) = 38 bytes.

ii. (4 points) Assume that there is a 0.1% chance that a datagram sent by the data recording device will be lost. Design an alternate solution using UDP that will save at least half the transmission cost (in Kr) of TCP/IP for this application.
Send UDP and wait to get back or send two times. Can send it up to 5 times more before TCP is better.

5. Network Addressing and Security

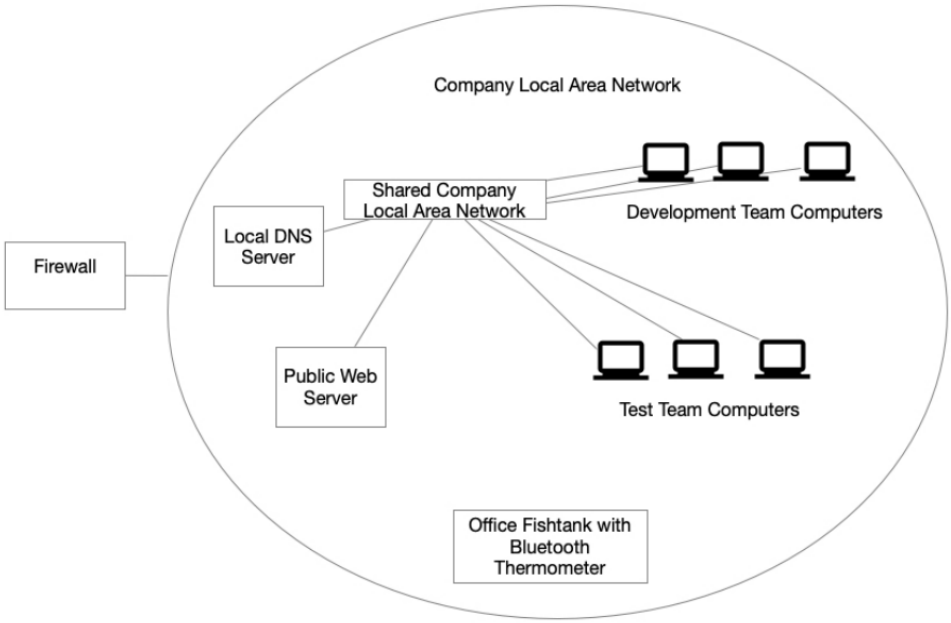


Figure 2: Network Security

You have just taken over responsibility for the small corporate network shown in Figure 2. The company has its public website, provided by a dedicated server, and uses the same machine to run its own DNS (domain name services) server. The Firewall's Access Control List(ACL) allows external users to access port 80 on the public web server, inside the company's LAN. You are asked to perform a security review, and discover the following "features" of the company network.
(a) Examining the DNS record for your company, you notice that it appears to have some strange content that was not included in the original update request.
i. (1 point) You update the DNS record to have the correct information for your company. How long will it typically take for this to take effect globally?
24-48 hours.

ii. (1 point) Name a DNS security protocol that could be enabled in order to prevent this occurring again?
DNSSEC ELEPHANT.

iii. (2 points) You notice that the DNS server is operating as an open resolver. Explain what potential security problem this poses, and who this might be a problem for.
An open DNS resolver means that it is vulnerable to DNS amplification attacks since open resolvers respond to all those who send to it. This makes it easy

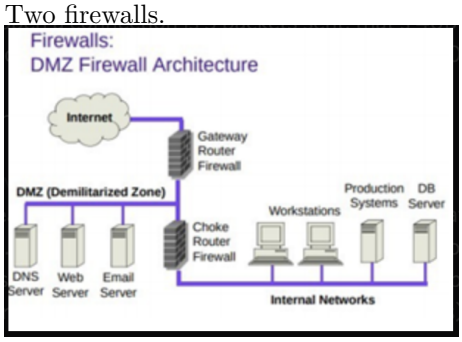
for a user to use a botnet of computers to send small requests with spoofed IP addresses to the resolver and the resolver will try to respond to each and send a large response back (since the request often asks for all the information it has) to the victim and that will possibly flood that server.

iv. (2 points) The Bluetooth enabled thermometer on the fish tank is running a pre-2016 version of Bluetooth and there is no way to update the firmware. How could this represent a security problem, and what would you recommend to resolve it?
NOT ON TEST.

(b) One of the internal problems that the company is facing is a problem between the development and test teams, where the development team is claiming that the test team is modifying files on their computers to introduce extra bugs (in order to claim extra overtime.)
i. (2 points) Given that the Company's networking equipment provides Virtual LAN (VLAN) support, draw a diagram showing how to reorganize the network to prevent the two teams from accessing each other's computers.
NEED DIAGRAM.

ii. (2 points) Identify one other problem with the company's network security, and explain why it is a problem.
They have a public web server connected directly to their shared LAN, which makes their LAN easy to access through the public web server.

iii. (4 points) Explain, using text and a diagram, how you would fix the problem you identified above. You may assume you have a reasonable budget for any additional equipment required.
Two firewalls.



6. Bluetooth
(a) (4 points) Explain, with reference to a diagram, how bluetooth devices form

piconets, and how piconets can join together to allow interworking between piconets. Include a description of any significant restrictions on connectivity between devices, and clearly label the roles of the devices in the piconets.

(b) (2 points) What problem might the inventor of the world's first fusion powered, bluetooth controlled anti-gravity hoverboard encounter if they tried to make it accessible by other bluetooth devices?

(c) (2 points) With possible reference to Question 7c describe how you would solve this problem if you were the inventor in question?

7. Bonus Questions
(a) (2 points) Should MD5 be used as a hashing solution for passwords?
No it shouldn't be, not secure.

(b) (4 points) Explain the steps required to resolve a query for www.ru.is from a host located at the BBC using a recursive DNS lookup, assuming that the requested information is not cached anywhere.
Talk to someone that is connected to Ru and another who is connected to BBC and down the tree it goes.

(c) (4 points) John Postel famously remarked that "The nice thing about standards is that there are so many of them."
Why are there so many standards? (Your experiences in Projects 2 and 3 may be helpful here.)
No one agrees and usually there are standards to fix other standards.

22 TEST 2019

1. Computer Networks and the Internet
(a) (2 points) What is the Media Access Control(MAC) address used for in Local Area Networks (LANs)?
A media access control address (MAC address) is a unique identifier assigned to a network interface controller (NIC) for use as a network address in communications within a network segment (from Wikipedia about MAC). To give a computer a unique ID.

(b) (2 points) Give two things that you should never do with a fiber optic cable.
Bend it too much and look into the other end of the cable, touch the end of the cable.

(c) (2 points) What is a firewall?
It blocks incoming connections, can be bypassed by using port 80 in some events. Filters packets.

(d) (4 points) Provide an example of byte stuffing, and explain when it should be used.
For instance used to put certain bytes at the beginning and the end to show where the messages start.

(e) (2 points) What guarantees does the TCP/IP protocol provide for its users?
Reliable in order delivery.

(f) i. (2 points) What is the Internet Control Message Protocol (ICMP) used for?
Used to send error messages and used by traceroute and ping.

ii. (2 points) Provide one example of a program that uses ICMP, and describe how it is used.
Traceroute and ping.

(g) (2 points) What is Kerckhoff's principle?
"A cryptosystem should be secure even if every- thing about the system, except the secret key, is public knowledge", Nothing matters but the key in encryption (take good care of it)

(h) (2 points) What is a nonce?
A random number that is put in an encryption to prevent someone from sending you an encryption with the same nonci, then he wont run the later one.

(i) (2 points) Why is it important for internet congestion management that

UDP packets are preferentially dropped.
Because TCP protocol reacts to congestion by slowing down but UDP doesn't have anything like that which is the reason why UDP is favored and the cause of TCP starvation.

(j) (2 points) What potential problem in routing is the Time to Live (TTL) flag in IP designed to prevent?
To ensure packets don't go on a never ending loop for instance if someone broadcasts to everyone about a broadcast to everyone.

(k) (2 points) What is the Fisher consensus problem?
Asynchronous connected nodes - cannot be sure that they agree on a single bit value.

(l) (4 points) Explain from the network packet perspective, including a diagram, how statistical multiplexing works when an ISP is providing a 5 Gbps connection split between 5 households, each of which has been sold connections of 5 Gbps each (5x oversubscription).
See image on discord and go from there.

2. TCP/IP
(a) TCP/IP Protocol

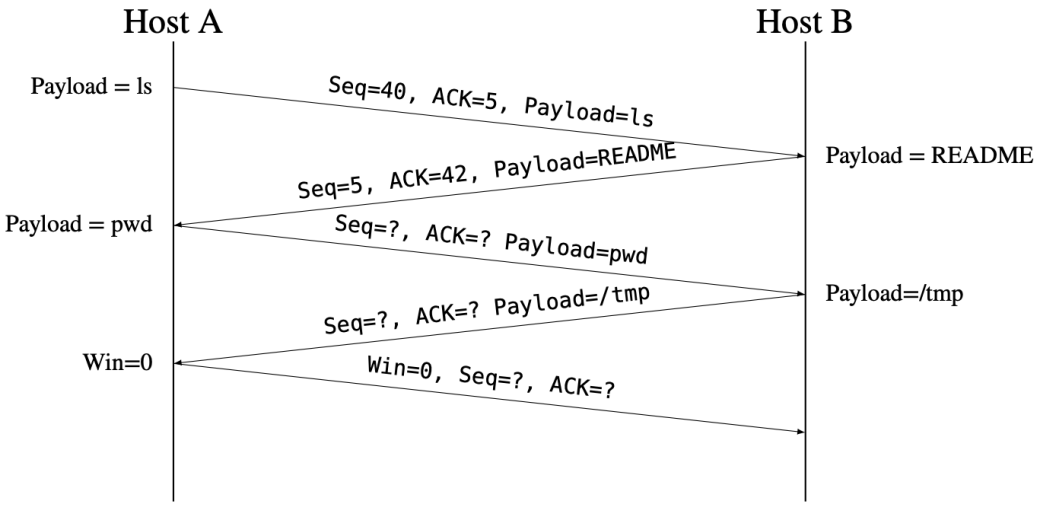


Figure 1: TCP/IP Data Exchange between 2 hosts

Figure 1 shows the middle of a data exchange for a remote terminal operation operating between two Internet hosts. The payload shown is the contents of data field being sent from the originating host using the single TCP segment to the other host shown in the diagram.
i. (3 points) Provide the correct values for the Sequence and Acknowledgment

fields for the segments, shown in the diagram as "?".
seq is an ack that he got, ack is seq + payload that he got.

ii. (2 points) What does the Win=0 segment sent by Host A signify?
It means that the computer can't accept any more data because buffer is full.

iii. (2 points) What will Host B do in response?
Wait for a bigger window size.

iv. (2 points) The application at Host B now reads a single byte from the network, and the underlying TCP/IP handling at Host B triggers a condition know as "Silly Window Syndrome". Explain what is meant by this term.
Window size is close to 0 and then he gets sent very little data (a big portion of them are headers)

v. (2 points) Describe a suitable method for avoiding Silly Window Syndrome.
Host A doesn't send Host B that they are ready to receive information again until the window size has become a certain size.

(b) i. (2 points) You are specifying a network for a workplace where several surveillance videos will be simultaneously streamed to separate groups of workstations for analysis. Given the choice between WiFi, copper cables, or fiber optic cables, each with identical host-to-host bandwidth, which would you specify and why?
WiFi works best because it naturally broadcasts, copper and fiber optic cables need to send manually to others.

ii. (2 points) What network layer protocol choice would you recommend for this application, and why?
Multicast to make sure it is secure, possible to see what IP addresses were sent to.

3. Network Engineering COMES ON THE TEST (Remember to show your working and state any assumptions)
(a) A group of enterprising RU students are proposing to setup a new "Data by Drone" service for large file transfers within Iceland. The idea is that a fleet of drones will be provided, which can be summoned to carry USB drives to transport data physically between sites. You have been asked by the Marketing department to perform the following engineering calculations for this service to determine when it will be time effective to use. You can assume that the average speed of the drones is 20 Kilometers per hour, including landing and takeoff and their maximum range is 50 km.
i. (4 points) Assuming you have a fiber optic 100 Mbps connection between two sites in Reykjavik, how long will it take to transfer a 250GB file from one site to the other, assuming a 10% protocol data overhead and ignoring any additional

delays caused by TCP/IP rate adaption?
$250GB * 1.1 * 8 = 2200Gb = 2,200,000Mb$
$2,200,000Mb/100Mbps = 22.000/60s = 366.66min/60min = 6.11hr$

ii. (4 points) If the two sites are 20 km apart, how long will it take to transfer the same file using Drones, assuming a USB writing speed of 250 MBps, a reading speed also of 250 MBps, and the Drone is ready at the sending site?
$250GB = 250,000MB/250MBps = 100/60s = 16.66min$
write + read = 33.33 min
1 hr and 33 min ¡ 6.1 hrs —> Drone wins

4. Networks and Addressing

(a) (3 points) For each of the following subnet addresses, provide an example of an IP Address that can be assigned to that subnet, and one that cannot.

Subnet	IP Address in the Subnet Host Range	IP address not in the Subnet
44.36.35.0/27	44.36.35.1 to 44.36.35.30	8.8.8.8
10.12.13.0/24	10.12.13.1 to 10.12.13.254	8.8.8.8
18.0.0.0/8	18.0.0.1 to 18.255.255.254	8.8.8.8

Bitmask/24 —> first 24 bites are set.

(b) (2 points) Because Network Address Translation (NAT) violates the end to end principle of the Internet, initiating connections to addresses behind NAT routers is problematic. Name one of the techniques that can be used used to overcome this problem, and briefly describe how it works.

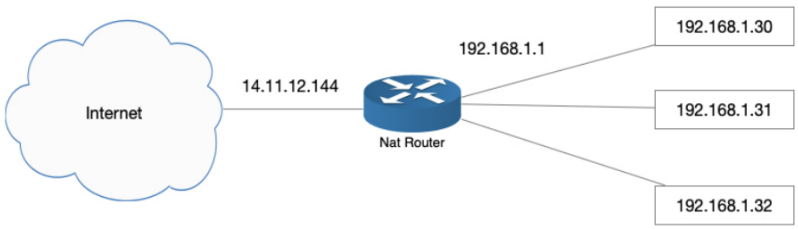


Figure 2: Domestic Network

Port forwarding in problem 3 assignment 2.

(c) (6 points) The residential router shown in Figure 2 has an external IP address of 14.11.12.144 facing the internet, and an internal IP address of 192.168.1.1. There are three internal devices with IP addresses as shown.
Hosts IP 192.168.1.30 (using outbound port 54011) and IP 192.168.1.31 (outbound port 54012) both set up individual, direct, ssh connections to host IP

130.208.240.8 at port 22. Give the values of the IP addresses and ports for packets at each step of the round trip between the originating hosts and the destination address, and explain what is happening. (You may choose port numbers where necessary.)

Dest	NAT	NAT	Origin	
inbound	130.208.240.130 22	14.11.12.144 X	192.168.1.1 X	192.168.1.30 54011
inbound	130.208.240.130 22	14.11.12.144 Y	192.168.1.1 Y	192.168.1.31 54012
outbound	130.208.240.130 60000	14.11.12.144 Y	192.168.1.1 Y	192.168.1.30 54011
outbound	130.208.240.130 70000	14.11.12.144 Y	192.168.1.1 Y	192.168.1.31 54012

Assigned ports are arbitrary, but must be consistent. deduct half a mark for each incorrect port or ip, cannot be less than 0.
Answer given.

(d) (2 points) Explain what will happen to the ssh connection between the two hosts if the router is power cycled, and why?
Power cycled: computer restart —> translation table gets lost and therefore the SSH connection.

5. Space, the Final Frontier!
(a) (2 points) You are responsible for setting up the Internet communication network with Iceland's new settlement at the edge of the Boreales Scopuli, near the Martian North Pole. At its closest approach to the Earth, communication time to Mars from Earth at light speed is approximately 4 minutes, and at its furthest it is 24 minutes. What is the maximum and minimum round trip time between Earth and Mars?
max RTT = 48 min
min RTT = 8 min

(b) (6 points) Assume all packets will be sent at light speed, with a segment size of 1400 bytes, and that there is a small, but not negligible, chance that any given packet will have one or more transmission errors due to solar radiation interference. If TCP/IP is used for this application, and the transmission speed is 100Mb/s, how big do the send and receive window sizes need to be for each connection when Mars is most distant from Earth, and when it is closest?
$8min * 60s * 100Mbs = 48,000Mb = 6,000MB = 6GB$
$48min * 60s * 100Mbs = 288,000Mb = 36,000MB = 36GB$
To be able to receive all that we can until we get a retransmission.

(c) (2 points) Examining the traffic you notice that it seems to consist of a continuous stream of high definition cat videos being sent to Mars, and a single text report sent back each day, at 23.00. Should Nagle's algorithm be enabled on this connection?
Nagle's theorem should not be enabled because it takes at least 8 minutes to

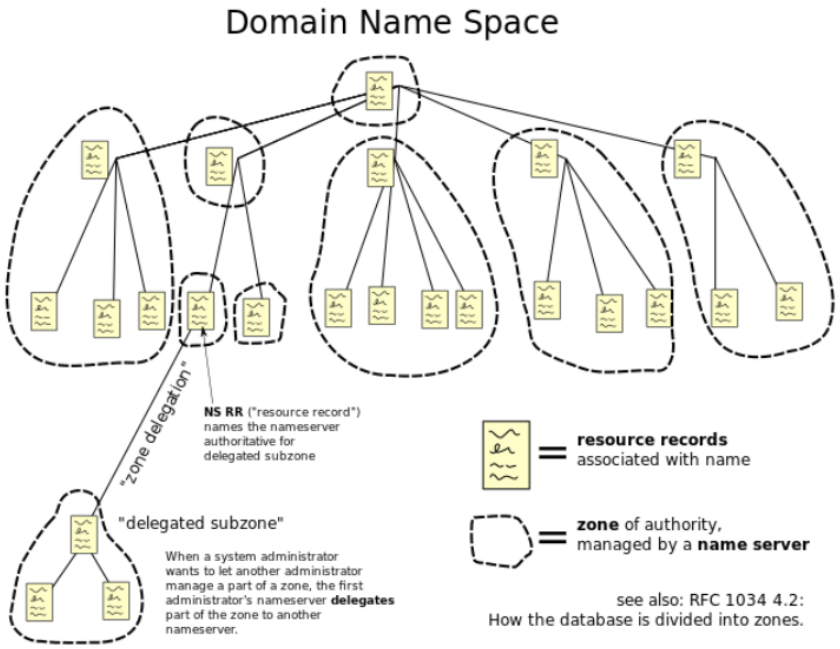
wait for ACK from Mars.

(d) (4 points) You decide to redesign this system using UDP, taking the asymmetric application traffic profile into account. While it is vital that the daily reports are received without error, nobody cares if the occasional cat video gets corrupted. Describe in detail how your new application protocol would send traffic in either direction.
When we are sending a report then there are 3 copies sent and then they are compared to one another (or use TCP only for report and UDP for rest).

6. Networks and Scaling
(a) (6 points) Networked applications can be classified according to their topology, into one of three broad types, hierarchical, full mesh, and peer-to-peer (p2p). List three general tradeoffs between the p2p and hierarchical topologies that should be considered when designing applications.
Answer is given, see the pdf.

(b) (2 points) Which topology is the Internet itself classified as.
Peer-to-peer mesh net
DNS net —> hierarchical

(c) (4 points) One of the key goals of any large scale, distributed system is to distribute load across many servers. Describe how this problem is solved in the Domain Name System(DNS), including a diagram of the relationships between DNS servers.



DNS sends messages up the tree until someone knows where the error is.
After it is gotten once it's stored in memory for a certain amount of time (also part of the URL for instance knows where .is is after visiting mbl.is).

(d) (4 points) DNS servers support two types of query, iterative and recursive. Explain how each of these works, including a diagram if necessary.
Recursive: I ask one who asks another who asks another ... until found
Iterative: Same one asks everyone until someone knows

(e) (2 points) What security protocol can be used to protect a DNS server against unauthorized updates?
DNSSEC - makes sure no one is doing DNS spoofing.

(f) (2 points) When auditing your companies network you discover that your companies external DNS server is operating as an open resolver. Explain what security issue this represents and exactly how this could be used to attack other Internet hosts.
Possible to perform an attack which makes all the bottom nodes in a hierarchical ping the ones on top which puts too much pressure on them.