Computer Networks

Personal notes based on lecture material and assigned reading from Princeton's COS 461: Computer Networks, taught by Nick Feamster.

Table of Contents

Network design principles 2

Link Layer **2**

Switched networks 2

Medium Access Control 3

Error detection and correction 4

Link layer communication 4

Network (IP) Layer **5**

Internetworking 5

Network Address Translation (NAT) 5

Firewalls 6

Routing **7**

Routing protocols 7

BGP route selection 7

Transport layer **8**

Transport layer overview 8

TCP details 9

TCP congestion control 10

Traffic shaping 12

Network security 14

Network measurement 17

Video streaming 18

Application Layer **18**

HTTP 18

Miscellaneous Topics **19**

Content Delivery Networks (CDNs) 19

Software Defined Networking (SDN) 20
Internet censorship 22

Network design principles

- Narrow waist
    - Requirement that every device must "speak" IP (network layer)
        - Only protocol at network layer
        - Advantages: any device that runs IP can get on internet
        - Disadvantages: difficult to make changes at network layer
            - Some progress recently: SDN, etc.
    - Guarantees
        - Link to network layer: point-to-point connectivity (i.e. on a LAN)
        - Network to transport layer: best-effort, end-to-end connectivity
        - Transport to application layer: reliable transport, congestion control
- End-to-end argument
    - Intelligence required to implement particular application on a communication system should be placed on endpoints, rather than middle of network
        - Dumb network, intelligent endpoints
    - Examples
        - Error handling in file transfer
        - End-to-end encryption
        - TCP/IP split in error handling, flow control, and congestion control
    - Trade off based on performance, not correctness
        - Error correction at lower levels can be a performance booster
    - Identifying the "ends"
        - Internet routing: ends may be routers, or may be ISPs
        - Transport protocol: ends may be end hosts

Link Layer

Switched networks

- Allow for nodes to communication with each other without direct connection between every 2 that wish to communicate
- Two types - circuit switched and packet switched networks
    - Circuit switching - employed by telephone system
        - Dedicated circuit across sequence of links first established
        - Source node sends stream of bits across circuit to destination node
        - Advantages

            - Resource control, better accounting, reservation of resources
            - Ability to pin paths between sender and receiver
    - Packet switching - nodes send discrete blocks of data to each other, called packets, or messages
        - Use *store-and-forward* strategy
        - Each node
            - First, receives a complete packet over some link
                - Stores the packet in internal memory
            - Then, forwards the complete packet to the next node
        - Nodes that store and forward packets are called *switches*
        - Properties
            - No end-to-end state established ahead of time
            - “Best effort” service
            - Shared resources (statistical multiplexing)
        - Implications
            - Sender never gets a busy signal
            - Variable delay, potential for loss (dropped packets)
        - Major advantages
            - Efficiency (ability to share resources)
            - Potentially better resilience
- Internetworks
    - Set of interconnected, independent networks
    - Node connected to 2 or more networks is a *router* or *gateway*

Medium Access Control

- Three broad categories - channel partitioning, random access, “taking turns”
- Channel partitioning
    - Time division multiple access (TDMA)
        - Access to channel in rounds, fixed time slots, unused slots go idle
    - Frequency division multiple access (FDMA)
        - Each station assigned frequency band
        - Unused transmission time in frequency bands go idle
- Random access protocols
    - Node sends packet at full channel data rate R, when desired
    - Requires collision detection and recovery
    - ALOHA
        - Transmit when data has to be sent (without listening to channel)
        - Detect collisions by timing out receival of ACK from data recipient
        - Recovery: retry after random delay
    - Carrier Sense Multiple Access (CSMA) - listen before transmit
        - If channel sensed idle, transmit entire frame

        - If channel sensed busy, defer transmission for a random backoff interval
        - Does not eliminate possibility of collisions, e.g. due to propagation delay
    - CSMA/CD (used by Ethernet)
        - In wired LANs: detect collisions by measuring signal strength, comparing transmitted/received signals
        - In wireless LANs: more difficult - received signal strength overwhelmed by local transmission strength
        - Better performance than ALOHA; simple, cheap, decentralized
- Taking turns
    - Channel partitioning - inefficient at low loads
    - Random access - collision overhead at high loads
    - Polling
        - A master node “invites” slave nodes to transmit in turn
        - Concerns: polling overhead, latency, single point of failure (master node)
    - Token passing
        - Control token passed from node to another sequentially
        - Concerns: token overhead, latency, single point of failure (token)

Error detection and correction

- Parity checking
    - Single bit parity - allows detection of single bit errors
    - 2D bit parity - allows detection and correction of single bit errors
- Internet checksum
    - Detects errors (flipped bits) in transmitted segments (transport layer only)
    - Sender places checksum (1’s complement sum) of segment contents into UDP checksum field
    - Receiver computes checksum and compares against stored valued
- Cyclic redundancy check
    - Can detect all burst errors less than r+1 bits
    - Widely used in practice

Link layer communication

- ARP queries
- Learning switches
    - Queries broadcast if entry not contained in switch table
        - Otherwise, packet routed to appropriate port
    - Spanning trees
        - Computed on network topology to avoid broadcast loops
        - Root is identified: switch with smallest identifier

        - Each switch broadcasts tuple:
          (supposed root, distance to supposed root, origin of message)
    - LAN switches (bridges)
        - Switches used to forward packets between LANs
- Properties
    - Broadcast as means of packet forwarding

Network (IP) Layer

Internetworking

- Forwarding approaches
    - Datagram (connectionless) approach
        - Every packet contains complete destination address
        - Switch uses forwarding table to determine how to route received packets
        - Any packet can be immediately forwarded (no connection state required)
        - A host sending a packet has no way of knowing whether network is capable of delivering it or if destination is up and running
        - Packets are forwarded independently of previous packets, so two successive packets may follow completely different paths
        - Switch or link failure may not have serious effect on communication, if possible to find alternate route
    - Virtual circuit (connection-oriented) approach
    - Source routing (less commonly used)
- IP service model
    - Each network type has a maximum transmission unit (MTU), which is the largest IP datagram that it can carry in a frame
        - NOT: largest packet size on the network (IP datagram must fit in payload of link-layer frame)

Network Address Translation (NAT)

- Used for security, to save IPv4 addresses
- Much more widely deployed than IPv6
- All datagrams leaving local network share same single NAT IP address
- NAT maintains translation table
    - Maps LAN side address (IP and port) to WAN side address (IP and port)
    - Source address changed by NAT router for outgoing packets
    - Destination address changed by NAT router for incoming packets
- 16-bit port-number field allows 60,000 simultaneous connections

- Addressing a host behind a NAT
    - Solution 1: statically configure NAT to forward incoming connection requests to a certain port to a certain host (e.g. pre-populate translation table with entry)
    - Solution 2: Universal Plug and Play (UPnP), Internet Gateway Device Protocol (IGD)
        - Allows hosts inside a NAT to add/remove port mappings
    - Solution 3: use of a relay to which both external and NATed client connect
- Controversial
    - Port numbers meant to address processes
        - Use of ports to identify hosts makes it hard to run a server behind a NAT
    - Routers should only process up to layer 3
        - Network layer should not be looking at TCP ports at all
    - Violates end-to-end argument
        - Network nodes should not modify packets
    - Difficult to support P2P applications
        - P2P apps need a host to act a server
    - IPv6 is a cleaner solution

Firewalls

- Block-by-default security model
- Make decisions based on IP, TCP, and UDP information
    - Filter based on source/destination IP addresses/ports
- Modern firewalls can filter based on application-specific protocols (HTTP, Telnet, FTP)
- Advantages
    - Firewalls can be deployed by vendor, without requiring client support (as cryptography-based security schemes do)
    - Security encapsulated in a centralized place (e.g. outside a VPN), allowing for better management by a system administrator
- Disadvantages
    - Easy to bypass, by running code internally
    - Any parties granted access through firewall become security vulnerability
    - Vulnerable to exploitation of bugs found in machine inside firewall
        - Malware - viruses, worms, spyware
- Related tools - intrusion detection systems (IDS), intrusion prevent systems (IPS)
    - Detect and report anomalous activity (unusually large amounts of traffic directed at a host or port number)
    - May take direct action to mitigate attack

Routing

Routing protocols

- Distance-vector routing
    - Routing Information Protocol (RIP)
        - Cost of edges - 1
        - Infinity - 16
        - Table refreshes - every 30 seconds
        - Updates - sent to all neighbors except one sending update (“split horizon”)
    - Problems
        - Count to infinity problem on link cost update
- Link-state routing
    - Variants
        - Open Shortest Paths First (OSPF)
        - Intermediate System-Intermediate System (ISIS)

BGP route selection

- List of criteria, in order
    - Highest “local preference”
        - Operator can set local pref values on routers (default: 100)
        - Set on incoming routes, to control outbound traffic
        - Can be used to differentiate primary route and backup route
        - Can control inbound traffic to some degree through use of BGP “community”, an announcement that causes a neighboring AS to adjust a local preference
    - Shortest AS path length
    - Multiple exit discriminator (MED)
    - Prefer eBGP over iBGP
    - Shortest IGP path to next hop (“hot potato”)
    - Tiebreak (arbitrary) - most “stable”, lowest router ID

Transport layer

Transport layer overview

- Services
    - Demultiplexing packets (via port numbers)
    - Detecting corrupted data (via checksums)
    - Optional: reliable delivery, flow control
- Error detection
    - Flipped bits detected in transmitted segments
    - Sender places 1's complement sum of segment contents into checksum field
    - Receiver computes checksum and compares against stored valued
- UDP
    - Goal: lightweight communication between processes
        - Avoids overhead of ordered, reliable delivery - no connection setup required, no in-kernel connection state
    - 8-byte header: SRC port, DST port, checksum, length
    - Used by popular apps
        - Query/response for DNS
        - Real-time data in VoIP
    - Advantages
        - Fine-grain control - sends message as soon as application writes
        - No connection setup delay - no connections used
        - No connection state - no buffers, parameters, sequence numbers, etc.
        - Small header overhead - only 8 bytes, versus TCP's 20 bytes
- TCP
    - Stream-of-bytes service
    - Connection oriented
        - Explicit set-up and tear-down of TCP connection required
    - Reliable, in-order delivery
        - Bit errors (corruption) detected via checksums
        - Missing/misordered data detected via sequence numbers
        - Recovery from lost data guaranteed via ACKs, retransmissions
    - Flow control
        - Prevent overflow of receiver's buffer space
        - Keep a fast sender from overwhelming a slow receiver
        - Uses receiver window (see TCP details)
    - Congestion control
        - Adapt to network congestion for greater good
        - Keep a set of senders from overloading the network
        - Uses congestion window (see TCP details)

TCP details

- Segment fields - srcPort, dstPort, sequenceNum, Acknowledgement, Flags, AdvertisedWindow, Checksum, etc.
- Establishing connection (3-way handshake)
  - Client -> server: SYN, sequenceNum = x
  - Server -> client: SYN + ACK, Acknowledgement = x + 1, sequenceNum = y
  - Client -> server: ACK, Acknowledgement = y + 1
- Terminating connection
  - Client -> server: FIN
  - Server -> client: ACK
  - Server -> client: FIN
  - Client -> server: ACK
- Sliding window
  - Objectives
    - Guarantee reliable delivery of data
    - Ensures data is delivered in order
    - Enforces flow control between receiver and sender
  - Sender maintains three pointers into send buffer
    - LastByteAcked $\leq$ LastByteSent $\leq$ LastByteWritten
  - Receiver maintains three pointers in receive buffer
    - LastByteRead $<$ NextByteExpected $\leq$ LastByteRcvd + 1
  - Buffer constraints
    - Send and receive buffer data must take up less space than MaxSendBuffer, MaxRecvBuffer respectively
      - LastByteWritten - LastByteAcked $\leq$ MaxSendBuffer
      - LastByteRcvd - LastByteRead $\leq$ MaxRecvBuffer
    - Sender must adhere to advertised window from receiver
      - Advertised Window = MaxRecvBuffer - ((NextByteExpected - 1) - LastByteRead)
      - LastByteSent - LastByteAcked $\leq$ Advertised Window
      - Define EffectiveWindow = Advertised Window - (LastByteSent - LastByteAcked), which from preceding bullet, must be greater than 0 before source can send more data
  - Other constraints
    - 32-bit sequence number must be large enough to make wraparound unlikely, given 120-second MSLs (maximum segment lifetimes)
    - 16-bit advertised window must be large enough to support delay x bandwidth worth of incoming data
  - Triggering transmission

            - Maximum segment size (MSS) reached
                - MTU - maximum size of IP packet
                - MSS - maximum size of TCP segment
                - MSS = MTU - size(TCP, IP headers)
            - Sending process invokes *push* operation
            - Timer fires (used to mitigate silly window syndrome)
    - Silly window syndrome
        - Always taking advantage of available window (even if less than MSS bytes) leads to introduction of small containers in stream, i.e fragmentation
        - Solution: Nagle's algorithm
- Adaptive retransmission
    - Original algorithm
        - Weighted average of RTTs
        - EstimatedRTT = $\boldsymbol{\alpha}$ * EstimatedRTT + (1 - $\boldsymbol{\alpha}$) * SampleRTT
    - Problem
        - Not clear whether ACK for original transmission or retransmission
    - Karn/Partridge algorithm
        - SampleRTT only measured for segments sent only once
        - After TCP retransmits, next timeout set to twice previous one (exponential backoff), to relieve potential congestion
    - Jacobson/Karels algorithm
        - Timeout set to weighted sum of EstimatedRTT and Deviation
- Defining record boundaries
    - Use of URG flag, UrgPtr field in TCP header
    - Use of push operation
    - Application specific record boundaries
- TCP extensions
    - RTT measurement accuracy
        - Add 32-bit timestamp to segment header, to be echoed back by receiver
        - Compute RTT on receipt as current time - timestamp
    - SequenceNum 32-bit field length limitation
        - Use timestamp to disambiguate equal sequence numbers
    - Advertised window 16-bit field length limitation
        - Reserve space in field to specify a scaling factor
    - Acknowledge receipt of non-contiguous segments
        - Invoke selective acknowledgement (SACK) option, by setting optional fields in header that refer to additional blocks of received data

TCP congestion control

- Introduced by Van Jacobson in late 1980s, eight years after TCP/IP became operational

- Each source determines available capacity on network by using received ACKs to pace transmission of packets (self-clocking)
- Problems
  - Difficult to gauge available capacity
  - Available bandwidth changes over time
- Additive Increase/Multiplicative Decrease (AIMD)
  - Source maintains CongestionWindow for each connection, analog to flow control's advertised window field
  - TCP source must send at speed no faster than slowest component (network or recipient host)
    - Define MaxWindow = MIN(CongestionWindow, AdvertisedWindow)
    - Redef EffectiveWindow = MaxWindow - (LastByteSent - LastByteAcked)
  - Observation: main reason packets are dropped is congestion, not transmission errors
  - Multiplicative decrease: every time timeout occurs, TCP halves window size (i.e. CongestionWindow)
  - Additive increase: every time window of packets successfully sent (each packet in last RTT ACKed), 1 packet added to CongestionWindow
    - For each ACK that arrives
      - Increment = MSS * (MSS / CongestionWindow)
      - CongestionWindow += Increment
  - Timeout
    - Function of average RTT, standard deviation in the average
    - Round-trip time only sampled once per RTT (not once per packet) using coarse-grained 500ms clock
- Slow Start
  - Practice of increasing congestion window rapidly when starting out, by using exponential increase factor rather than additive factor
  - Used to prevent burst of packets caused by immediately sending as many packets as in advertised window on start
  - Use cases
    - Beginning of connection - CongestionWindow doubled until packet drop occurs, at which point timeout causes multiplicative decrease
    - Dropped packet - if timeout occurs after all other packets have left transit, no ACK is received to clock retransmission; as a solution to this, the old CongestionWindow is saved in a field called CongestionThreshold, and the sending rate is increased exponentially to this target value
  - Problems
    - Many packets dropped in initial slow start period, if capacity cutoff is just above a slow start milestone (e.g. at 16 packets)
    - If delay * bandwidth product is large (e.g. 500 KB), up to that much data can be dropped at the beginning of each connection
  - Alternative: quick-start

        - Undergoing standardization at IETF
        - TCP sender can request initial sending rate by putting a requested rate in its SYN packet as an IP option
        - Routers determine if network can support that rate, given current level of congestion
            - If so, source begins sending at higher rate
            - If not, source falls back to standard slow start
            - Requires greater cooperation of routers than standard TCP
- Fast Retransmit
    - Heuristic that triggers retransmission of a dropped packet sooner than timeout
    - Receiver sends duplicate ACK when a packet is received out of order
    - Packet is resent when three duplicate ACKs are received
    - Effective for
        - Long data transfers (e.g. many packets)
        - Large window size
- Fast Recovery
    - When fast retransmit signal congestion, ACKs used to reclock instead of invoking slow start/reducing congestion window to 1 again
    - Congestion window cut to half, and additive increase resumed

Traffic shaping

- Traffic types
    - Data - bursty, weakly periodic, strongly regular
    - Audio - continuous, strongly periodic, strongly regular
    - Video - continuous, bursty (compression), strongly periodic, weakly regular
- Policing criteria
    - Average rate
        - Long-term average rate (packets per time interval) at which flow's packets are sent into a network
        - 100 packets/second more constraining than 6000 packets/minute
    - Peak rate
        - Limits maximum number of packets that can be sent over a shorter period of time, e.g. 1500 packets/second for an average rate of 6000 packets/minute
    - Burst size
        - Limits maximum number of packets that can be sent over an extremely short interval of time
- Leaky Bucket
    - One bucket per flow
    - Data arrives in bucket of capacity *b* and drains at average rate *r*
    - If bucket is full, packets are dropped

    - Extension: use two leaky buckets to police a flow's peak rate in addition to its long-term average rate
    - Applications: audio streaming
- (r,T) traffic shaping
    - Traffic divided into T-bit frames
    - Flow can inject $\leq$ r bits into any T-bit frame
    - Suited for fixed-rate flows
- Token bucket
    - Bucket can hold up to *b* tokens
    - New tokens generated at rate of *r* tokens per second, and added to bucket if it contains less than *b* tokens at any given time
    - Transmitting a packet requires consuming a token from the bucket
    - Setup limits burst rate to *b* and long-term average rate to *r*
    - Bucket often combined with a data buffer, which holds (i.e. buffers) incoming data until it is ready to send (i.e. enough tokens in bucket to send a packet)

<u>Leaky Bucket</u>	<u>Token Bucket</u>
Forces bursty traffic to smooth out	Permits bursty traffic, but bounds it
Never sends more than r packets per second	Bounds burstiness • Flow never sends more than b + rt tokens worth of data in interval t • Long-term transmission rate does not exceed r
Priority policy	No discard or priority policy
Rigid	Flexible

- Extensions
    - To police flow's peak rate (in addition to long-term average rate)
        - Use two leaky buckets in series
    - To throttle a flow's rate after a delay
        - Use composite shaper (token bucket followed by leaky bucket)
- Difficulties of policing
    - Over any period, flow can exceed rate by b tokens
    - Flow can "cheat" by sending b + rt tokens of data every consecutive interval, if a network only measures traffic by interval

Network security

- Types of attacks
  - Routing (BGP)
  - Naming (DNS)
    - Reflection (DDOS)
    - Phishing
  - Resource exhaustion
  - Connections (TCP)
- Internet design
  - Simplicity
  - On by default
  - Hosts are insecure
  - Attacks can look like “normal” traffic
- Components
  - Availability - ability to use a resource
  - Confidentiality - concealing information
  - Authenticity - assures origin of information
  - Integrity - prevents unauthorized changes
- SYN flood attack
  - Occurs in TCP three-way handshake
  - Client floods server with SYNs from many spoofed IPs
    - If server allocates space on receipt of SYN (~280 bytes), can quickly exhaust resources, preventing it from serving legitimate requests
  - Solution: SYN cookies
    - Server sends 32-bit sequence number, which is hash of source IP, source port, destination IP, destination port, random nonce, timestamp
    - Client must respond with sequence number, which server then validates
- DNS cache poisoning
  - Causes
    - DNS resolvers trust responses
      - No authentication of responses
    - Resolver query generates race condition
      - DNS is connectionless (UDP)
  - Schuba attack (1993)
    - NS record for www.evil.org points to ns.yahoo.com
    - A record for ns.yahoo.com points to 1.2.3.4 (wrong IP)
    - 1.2.3.4 is mistakenly cached by resolver
    - Solution: bailiwick system
  - Canonical cache poisoning attack
    - Client makes a DNS query (e.g. A record for google.com)
    - Attacker sends recursive resolver many crafted replies for query

            - If attacker wins the race, incorrect DNS reply is cached in the resolver
                - Can’t be removed until entry expires
            - Complications
                - Resolver adds 16-bit ID numbers to outgoing queries
                - Attacker can try to guess ID number...
                - Attacker can have hundreds of clients send same DNS request
                    - For each request, sends a DNS reply with different guess
                    - With high probability, one will match (birthday paradox)
        - Kaminsky attack (2008)
            - Previously, on losing a race, attacker had to wait for TTL expiration
            - Attacker generates queries for subdomains (e.g. 1.google.com, 2.google.com), each of which triggers a new race
            - Eventually attacker is able to insert an NS record for x.google.com, with an accompanying false A record for google.com
        - Defenses
            - Query ID (can be guessed)
            - Randomize ID (tougher to guess, but still only 16 bits)
            - Randomize source port (another 16 bits of entropy)
            - 0x20 encoding (DNS is case insensitive)
                - Randomly capitalize characters in URL
                - Since query included in response, becomes harder to guess
            - DNSSec (use crypto)
                - Responses include signature on (IP address, public key of referred party) tuple (e.g. in an NS record)
- DNS amplification
    - Exploits asymmetry in size between DNS query and response
        - E.g. query could be 60 bytes, reply could be 3000 bytes
        - Only small amount of request traffic needed to overwhelm victim
    - Defenses
        - Prevent IP address spoofing
        - Disable open resolvers
- Denial of service
    - Attempt to exhaust resources
        - Network: bandwidth
        - Transport: TCP connections
        - Application: server resources
    - Defenses
        - Ingress filtering
            - Router that connects to a stub AS can drop all packets from it that don’t originate in its IP range
            - Doesn’t work well near “core” of network
        - uRPF (reverse path filtering) checks
            - Used in the core of a network

            - Routing tables used to determine if packet could actually originate from an interface
            - Requires symmetric routing
        - SYN cookies (TCP)
- Routing security (BGP)
    - Control plane authentication
        - Session: protects point-to-point communication between routers
        - Path: protects AS path
        - Origin: protects origin AS in AS path
    - Attacks on routing
        - Configuration error
        - Compromised router
        - Unscrupulous ISPs
    - Types of attacks
        - Reconfigure router
        - Tamper with software
        - Tamper with routing data
    - Kapela attack
        - AS path poisoning
        - MITM hijacks traffic to origin by advertising a route from itself to origin
        - Attacker can evade traceroute, by not decrementing TTL in its AS
    - Path shortening attack
        - Attacker advertises a route that excludes an AS actually in the route
    - Secure BGP (BGPSec)
        - Proposal to add signatures to route advertisements
        - Origin (address) attestation
            - Certificate binding an IP prefix to its owner
            - Certificate signed by trusted third party (e.g. routing registry)
        - Path attestation
            - Signatures along AS path
            - AS k forwards two signed path attestations to AS k+1
                - Path from origin to AS k
                - Path from origin to AS k, plus AS k+1
            - Prevents hijacking (Kapela), path shortening, modification
            - Does not prevent against route suppression and some replay attacks (e.g. premature advertisement of withdrawn route)
        - Session authentication
            - Goal: authenticate TCP session
            - MD5 authentication used
                - Key negotiated out-of-band
            - TTL hack
                - Sender transmits packets with TTL of 255
                - Receiver drops packets with TTL < 254

Network measurement

- Two types
    - Passive measurement
        - Collection of packets, flow statistics already on network
    - Active measurement
        - Inject additional traffic to measure characteristics
        - Techniques used: ping, traceroute
- Reasons to measure
    - Billing
    - Security
- Passive measurement
    - Simple Network Management Protocol (SNMP)
        - Management Information Base (MIB) can be queried for information
        - Poll interface byte and packet counts periodically, take differences
        - Pros: ubiquitous
        - Cons: coarse
    - Packet monitoring
        - Monitor looks at flow packet contents or packet headers
        - E.g. tcpdump, ethereal, wireshark
        - Sometimes requires special hardware, e.g. monitoring card
            - Mounted on servers, alongside routers that forward traffic
        - Pros: lots of detail (timing, header information)
        - Cons: high overhead
    - Flow monitoring
        - Monitors record statistics per flow
        - Components of a flow
            - Packets that share common src and dst IP, port; protocol type; TOS byte; interface
                - Other header fields: next-hop IP, src/dst AS and prefix
            - Packets that appear close together in time
            - Flow record finalized if no packet with matching set of header fields appears in some time interval (generally 15 or 30 seconds)
        - Sampling - build flow statistics based on samples of packets
        - Pros: less overhead (than packet monitoring)
        - Cons: more coarse, no packets/payloads
- Active measurement
    - Traceroute
        - Overview
            - Diagnostic tool that displays path and transit delays of packets
            - Records RTTs of packets retrieved from successive hosts on path

                - Proceeds unless all three sent packets are lost more than twice
            - How it works
                - Sends packets with increasing TTL values
                - Nodes along IP layer path decrement TTL
                - When TTL = 0, nodes return ICMP “time exceeded” message
            - Problems
                - Can’t unambiguously identify one-way outages
                - ICMP messages may be filtered or rate-limited
                - IP address of “time exceeded” packet may be the outgoing interface of the return packet, not the desired, incoming interface
                - Can be skewed by load balancers

Video streaming

- TCP is not a good fit
    - TCP retransmits packets, but don’t always need (want) this
    - TCP slows down sending rate after packet loss, which could cause starvation
    - TCP has overhead, including a 20-byte header, acknowledgements
- UDP as potential solution
    - Does not retransmit packets
    - Does not adapt sending rate
    - Has smaller header
    - These problems must be solved by higher layers (e.g. application)
- Playout buffer
    - Smooths playout rate experienced by user
- YouTube
    - Uploaded videos converted to Flash or HTML5
    - Use of HTTP/TCP
    - Use of CDNs
- Skype/VoIP
    - Analog signal digitized through A/D conversion
    - Digitized signal sent over Internet

Application Layer

HTTP

- HTTP 1.1
    - Default behavior - persistent connections with pipelining
- Persistent connections

    - Multiple HTTP requests/responses multiplexed onto same TCP connection
    - Delimiters, content length header indicate end of requests
- Pipelining
    - Client sends requests as soon as it encounters referenced object
    - Client does not wait for a response to send a new request
- HTTP caching
    - Lots of objects don't change (e.g. static content)
    - Challenges
        - Significant fraction of web content uncachable
        - Want to limit staleness of cached objects
- DASH - Dynamic Adaptive Streaming over HTTP
    - Server 1) divides video file into chunks, 2) encodes, stores chunks at different rates, 3) provides URL for different chunks in a *manifest file*
    - Client 1) periodically measures server-to-client bandwidth, 2) requests one chunk at a time, consulting the manifest file, 3) chooses maximum coding rate sustainable at current bandwidth
    - Intelligent client; client determines
        - When to request chunk (to prevent buffer starvation/overflow)
        - What encoding rate to request
        - Where to request chunk from (e.g. a proximate server vs. one with high available bandwidth)

Miscellaneous Topics

Content Delivery Networks (CDNs)

- Overlay network of web caches design to deliver data to client from optimal location
- Goal: replicate content on many geographically disparate servers
- Owners
    - Content providers (e.g. Google)
    - Networks/ISPs
- Operational challenges
    - How to choose server replica? (server selection)
    - How to direct client to chosen replica? (content routing)
- Server selection
    - Any "alive" server (availability, fault tolerance)
    - Lowest load (load balancing)
    - Lowest latency
    - How to pick a "good" CDN node to stream to client
        - Pick CDN node geographically closest to client
            - How to determine client's location?

        - Pick CDN node with shortest delay (or min # hops) to client
            - CDN nodes periodically ping access ISPs, reporting results to CDN DNS server
        - IP anycast
            - Same IP prefix advertised from multiple locations
- Caching
    - In browser (load)
    - In network (local ISP, CDNs)
    - How to direct a client to cache
        - Configure browser to use a cache
        - Server directs request
- Content routing
    - Routing (e.g. anycast)
        - Number all replicas with same IP address
        - Rely on routing to take client to closest replica
        - Simple, but coarse
    - Application-based (e.g. HTTP redirect)
        - Requires client to go to server first, increasing latency
        - Simple, but incurs delays
    - Naming-based (e.g. DNS)
        - Response to DNS query contains IP address of particular cache
        - Offers fine-grained control, fast
- CDNs and ISPs
    - Symbiotic relationship
    - CDNs peer with ISPs because
        - Better throughput (lower latency)
        - Offers redundancy
        - Eases burstiness
    - ISPs peer with CDNs
        - Good performance for customers
        - Lower transit costs

Software Defined Networking (SDN)

- Components
    - Control plane - network's "brain"; computes forwarding rules; can be run separately from devices
    - Data plane - programmable hardware; controlled by control plane
- Data plane responsibilities
    - Forward traffic according to control plane logic
    - Examples: layer 2 switching, IP forwarding
- Control plane responsibilities

- Logic for controlling forwarding behavior
    - Examples: routing protocols, network middlebox configuration
- Applications
    - Data centers - VM migration, layer 2 routing
    - Routing - more control over decision logic
    - Wide-area backbone networks
    - Enterprise networks - security applications (network access control)
    - Research - coexistence with production
    - Also: Internet Exchange Points (IXPs), home networks
- SDN overview
    - Control network behavior from single, high-level control *program*
- Advantages
    - Faster innovation - removes dependencies on vendors, IETF
    - Simpler management - no need to invert control-plane operations
    - Easier interoperability between vendors - compatibility only necessary in “wire” protocols
    - Simpler, cheaper routers - no software needs to be written on routers
- Routing Control Platform (RCP)
    - Goal: solve problems inherent to BGP, including poor interaction with other protocols (IGP)
        - Possibility of forwarding loops
        - Requires tagging routes with state (“don’t forward path from/to provider A to provider B”)
    - Represents an AS, computing routes for all routers inside of it
        - Can compute consistent router-level paths, pinning paths if necessary for traffic engineering or other purposes
        - Implements policies in terms of known AS relationships
    - Exchanges routing info with RCPs in other ASes
        - Can utilize knowledge of all externally learned routes
    - Implementation
        - Problems: backward compatibility, deployment incentives
    - Applications
        - Problem: failures or maintenance can change path weights, causing oscillations in path topology
            - Solution: RCP can pin paths as weights changes
        - Problem: routers don’t know which routers need more specific (prefix length) external routes
            - Solution: RCP performs efficient aggregation of route info
        - Egress selection - in case of a planned maintenance event, RCP can give customers control on egress selection
        - Interdomain routing security - can detect bogus routes and reshape network weights to avoid them
    - Scalability

        - Must store routes and compute routing decisions for every router in an AS
        - Solutions
            - Eliminate redundancy - store single copy of each route
            - Accelerate lookup - maintain indexes to identify affected routers
            - Only perform BGP routing
    - Reliability
        - Replicate RCP, running multiple identical servers ("hot spare")
            - Each replica receives the same inputs and runs same routing algorithm
            - Claim: no need for consistency protocol if both replicas always seem same information
        - Possible consistency issue
            - Routers could suggest conflict routes (causing loops), in cases where AS is partitioned
            - Silver lining: flooding-based IGP protocol (e.g. OSPF, IS-IS) means each RCP replica knows which partition it connects to
        - Solution
            - RCPs receive same state from each partition they can reach
            - Only act on partition if it has complete state

Internet censorship

- Technical enforcement
    - Blocking a website
    - Blocking transfer of specific content
        - Filtering - keyword-based, IP address, DNS
        - Taking down a server (web, DNS)
        - Internet "kill switch" (Egypt)
            - Stop advertising BGP routes
            - Drop all incoming routes
        - Unlist search results (Google)
    - Rate limiting performance to site/service
- Means of resistance
    - Community wireless networks (i.e. BYOI)
        - Commotion Wireless - open-source communication tool that uses wireless devices to create decentralized mesh networks
    - Anonymous routing schemes
        - Tor - system with the stated goal of enabling anonymous communication; uses packet header stripping, encryption, and series of relays to to anonymize traffic; sometimes used as a censorship circumvention tool
    - Distributed services

- FreedomBox - toolkit for building personal servers that run free software for distributed social networking, email, and audio/video communications