COMS3200-exam-notes - Summary Computer Networks I

Computer Networks I (University of Queensland)

Table of Contents

Chapter 1 – The Basics

- The network edge: hosts (clients and servers), access networks
- Packet transmission delay over a link: time to transmit L-bit packets = $\frac{L \text{ bits}}{R \text{ bits/sec}}$
- Guided media (wired) vs unguided (radio signals)
- Ethernet (Twisted Pair - Cat5 = 100Mbps, Cat6 = 10Gbps), Coax, fibre-optic
- Radio: terrestrial microwave (~45Mbps channels), Wi-Fi, wide-area (3G/4G), satellite
- The network core: mesh of interconnected routers
- Packet switching
    - Store-and-forward
        - L/R seconds to transmit (push out) L-bit packet into link at R bps
        - *Entire* packet must arrive at router before it can be transmitted
        - End-to-end delay = 2 * L/R (assuming no propagation delay)
- Key network core functions: **routing** (determining source-destination route of packets), **forwarding** (moving packets from input to appropriate output)
- Alternative: circuit switching – e.g. telephone networks
    - End-to-end resources reserved for 'call' between source and dest
    - Dedicated resources: no sharing (guarantees performance), means circuit segments are idle if not currently being used
    - Frequency division (FDM) versus time division (TDM)

Packet vs circuit switching

- Circuit switching has fixed number of potential users, packet switching can have more users so long as they aren't all communicating at once
- Packet switching:
    - Better for 'bursty data'
    - Can share resources
    - Simpler (no call setup)
    - Congestion is a problem (packet delay and loss)
    - Protocols needed for reliable data transfer and congestion control

Internet infrastructure: network of networks

- End systems connect to access ISPs
- Access ISPs are interconnected – overall network is complex
- Access ISPs not all interconnected – connected via 'global' or 'provider' ISPs
- Internet exchange points connect provider ISPs
- Sometimes regional networks connect access ISPs together, connecting to a provider ISP
- Content provider networks (e.g. Google, MS, Akamai) run their own networks

Delay and loss

- Delay/latency sources (total is sum of all):
    - Nodal processing: error checking, routing (typically tiny, < 1ms)
    - Queuing delay: (L*a)/R (a = avg arrival rate), congestion (arrive > receive)
    - Transmission delay: L/R
    - Propagation delay: d/s
        - d = length of physical length
        - s = propagation speed in medium e.g. ~2×10^8 m/s
- Delay can be seen using tracert

- Packet loss
    - Queue (buffer) is full, packet gets dropped and may be retransmitted

Throughput: rate of transfer (instantaneous or average)

- 'Bottlenecked' by slowest link (e.g. server link to backbone, or client link to backbone, rarely the backbone links)

Layered architecture

- Networks are complex – need organisation
- Layers allows identification of the relationships between system's pieces
- Modularisation allows maintenance, upgrades (change one layer without changing above/below layers)

IP Stack

- Application (e.g. FTP, SMTP, HTTP)
- Transport (TCP, UDP)
- Network (IP, routing protocols)
- Link (e.g. Ethernet, 802.111 (wifi), PPP)
- Physical (bits 'on the wire')
- Alternate: OSI model adds Presentation and Session layers between Application and Transport
- Data is encapsulated (message, segment, datagram, frame)

Chapter 2 – Application Layer

- Network applications: run on *end systems*, do not need to know about network core

Client-server architecture

- Server is an always-on host with a permanent IP address; data centres for scaling
- Clients communicate with server, may be intermittently connected, have dynamic IP addresses; they do not communicate with each other directly

P2P architecture

- No always-on server
- Clients act as *both* client and server
- End-systems directly communicate
- Peers request and provide service from and to other peers – self-scalable
- Intermittently connected

Process communication

- Sockets: process sends/receives to/from its socket
- IP address identifies the host
- Process / socket is identified by a port number

Role of the Application layer protocol – defines:

- Types of message exchanged, message syntax, message semantics, rules about when to send and respond to messages, open protocols (e.g. HTTP, SMTP, defined in RFCs), proprietary protocols (e.g. Skype)

Transport service requirements for applications

- Data integrity – some apps require 100% reliable transfer (e.g. file transfer), others can tolerate some loss (e.g. audio)
- Throughput – some apps require minimum throughput to be effective (e.g. multimedia streaming)
- Latency – some apps (e.g. VoIP, online gaming) require low delay to be effective
- Security

Applications can use TCP vs UDP

- TCP is reliable, has flow control (won't overwhelm receiver), congestion control
    - Does not provide latency, minimum throughput guarantees or security
    - 'Connection-oriented' – requires setup between client and server processes
- UDP is unreliable
    - Does not provide reliability, flow control, congestion control, throughput guarantee, security, connection setup
- Security can be done at app layer using SSL/TLS

The HTTP Protocol & the Web

- HTTP uses TCP (on port 80 by default)
- HTTP is stateless (no information about client is maintained by server)
- Can be non-persistent (connection closes after each request) or persistent

Non-persistent HTTP

- OS overhead to setup TCP connection for each request/response
- Can use parallel TCP connections to fetch referenced objects
- Response time: One RTT (round-trip time) to initiate TCP connection, one for HTTP request, file transmission time
    - Total response time = 2RTT + file transmission time

Persistent HTTP

- Connection left open between requests
- As little as 1 RTT for all referenced objects (depending)

HTTP request format

```
GET /index.html HTTP/1.1\r\n
Host: www.myserver.com\r\n
User-Agent: Firefox/3.6.10\r\n
Accept: text/html,application/xhtml+xml\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7\r\m
Keep-Alive: 115\r\n
Connection: keep-alive\r\n
\r\n
```

- Methods: GET, POST, PUT, DELETE, HEAD
- Uploading form input
    - Using POST: input is uploaded in entity body
    - Using GET (URL method): input is uploaded in URL field of request line

HTTP response format

```
HTTP/1.1 200 OK\r\n
Date: Sun, 26 Sep 2010 20:09:20 GMT\r\n
Server: Apache/2.0.52 (CentOS)\r\n
Last-Modified: Tue, 30 Oct 2007 17:00:02 GMT\r\n
ETag: "17dc6-a5c-bf716880"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 2652\r\n
Keep-Alive: timeout=10, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=ISO-8859- 1\r\n
\r\n
data data data data data ...
```

Web caching (proxy server)

- Goal: satisfy client request without involving the origin server
- Web browser sends HTTP requests to cache
    - If object in cache, cache returns object, otherwise fetches from server and then returns object
    - `Cache-Control: max-age=120\r\n`
- Cache is both client and server (server to the original client, client to the origin server)
- Cache may be installed by ISP
- Why? Reduce response time, reduce traffic/congestion, helps 'poor' content providers
- Can calculate benefit by supposing cache hit rate, access link utilisation, and estimating total delay

Conditional GET

- Don't send object *if* cache has up-to-date version
- Cache specifies date of cached copy in HTTP request (`If-modified-since: <date>`)
- Server returns 304 Not Modified if cached copy is up to date, returns object otherwise

File Transfer Protocol (FTP)

- Client/server model – RFC 959, usually port 21 for control, port 20 for data
- Uses separate control and data connections
- Server maintains 'state' (current directory, authentication)
- PASV mode versus active: in active mode, the server opens the data connection to the client
- Commands: `USER username`, `PASS password`, `LIST`, `RETR filename`, `STOR filename`
- Return codes: `331 Username OK, password required`, `125 data connection already open, transfer starting`, `425 Can't open data connection`, `452 Error writing file`

Email: principles

- User agents, mail servers, mail transfer protocol (SMTP)
- User agent: 'mail reader' for composing, editing, reading messages; outgoing/incoming messages are stored on a server
- Mail servers: 'mailbox' contains incoming messages, 'message queue' of outgoing messages, SMTP between mail servers to send messages

SMTP (RFC 2821)

- Uses TCP on port 25

- Persistent connections
- Direct transfer: sending server goes directly to receiving server
- Three phases: handshake, transfer of messages, closure
- Command/response interaction (as with HTTP, FTP)
    - Commands are 7-bit ASCII text, responses are status code and phrase
- Sample SMTP:

```
S: 220 hamburger.edu\r\n
C: HELO crepes.fr\r\n
S: 250 Hello crepes.fr, pleased to meet you\r\n
C: MAIL FROM: <alice@crepes.fr>\r\n
S: 250 alice@crepes.fr... Sender ok\r\n
C: RCPT TO: <bob@hamburger.edu>\r\n
S: 250 bob@hamburger.edu ... Recipient ok\r\n
C: DATA\r\n
S: 354 Enter mail, end with "." on a line by itself\r\n
C: Do you like ketchup?\r\n
C: How about pickles?\r\n
C: .\r\n
S: 250 Message accepted for delivery\r\n
C: QUIT\r\n
S: 221 hamburger.edu closing connection\r\n
```

Mail access protocols

- SMTP delivery/storage is on receiver's server
- Mail access protocol retrieves from server
- POP – Post Office Protocol – authorisation, download (stateless)
- IMAP – Internet Mail Access Protocol – more features, including manipulation of server-stored messages (stateful)
- HTTP – Webmail

Domain Name System (DNS)

- Hosts identified by 32-bit IP address and 'name'
- DNS maps between IP address and name
- Distributed database (in hierarchy of *name servers*)
- Application-layer protocol
- Provides hostname-to-IP-address translation, host aliasing, mail server aliasing, load distribution
- Decentralised – no single point of failure, deals with high volume, lower latency for requests, lower maintenance
- Heirarchical database – e.g. `www.amazon.com` -> queries root server to find the `com` DNS server -> queries the `com` DNS server to find the `amazon.com` server, queries the `amazon.com` DNS server to get IP for `www.amazon.com`
- 13 root name servers worldwide

Top-level Domains (TLDs) – authoritative servers

- E.g. com, org, ned, edu, and top-level country domains e.g. uk, fr, ca, jp, au
- Authoritative DNS servers
    - Organisation's own DNS server/s, providing authoritative hostname to IP mappings for that organisation's hosts
- Local DNS name server

    - Not strictly in hierarchy – ISPs have one ('default name server')
    - Queries are sent here first, acts as proxy to forward queries into hierarchy
- Queries
    - Iterated queries (contact server, server says 'I don't know this name, but ask this server')
    - Recursive queries (burden of name resolution on contacted server, possible heavy load on upper levels of hierarchy)

Caching

- Once any name server learns a mapping, it caches the mapping
- Mappings disappear / timeout after some time (TTL)
- TLD servers cached in local name servers (no need to visit root servers often)
- Cached entries may be out of date
    - If a named host changes IP address, this may not be known internet-wide until all TTLs expire
    - There exist proposed update/notify mechanisms (not widely used)

DNS records – Resource Record (RR)

- Format: `(name, value, type, ttl)`
- Types:
    - `A` – `name` is hostname, `value` is IP address
    - `NS` – `name` is domain, `value` is hostname of authoritative DNS server for this domain
    - `CNAME` – `name` is alias name for a 'canonical' name
    - `MX` – `value` is the name of mailserver associated with `name`
- Insertion
    - Register name at DNS registrar, providing names, IP addresses of authoritative server
    - Registrar inserts two RRs into TLD server – NS and A records

DNS Protocol / messages

- Query and reply messages have same format
    - 2-byte identification
    - 2-byte flags (query or reply, recursion desired, reply authoritative etc.)
    - 2-byte number of questions
    - 2-byte number of answer RRs
    - 2-byte number of authoritative RRs
    - 2-byte number of additional RRs
    - Questions, answers, authority, additional info in that order (variable size)

Attacks on DNS

- DDoS attacks
- Redirect attacks (DNS poisoning, man-in-the-middle)

P2P applications – e.g. BitTorrent, Skype

File distribution time for file with size F and N peers

- d_{min} = minimum client download rate
- u_s = server upload capacity
- u_i = client i's upload capacity

- For client-server: $D_{cs} \geq \max\{\frac{N\,F}{u_s}, \frac{F}{d_{min}}\}$
    - The former 'bottleneck' increases linearly with N
- For P2P: $D_{P2P} \geq \max\{\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{N\,F}{u_s+\sum u_i}\}$
    - The last bottleneck's top term increases linearly with n, but so does the bottom term (due to the summation of peer upload rates)

BitTorrent: file divided into 256Kb chunks

- Peer joining torrent accumulates chunks over time (requesting chunks, sending tit-for-tat)
- Registers with tracker to get list of peers, and connects to some of them
- Once peer has the entire file, it may seed (altruistically remain in torrent) or leach (selfishly leave)

Chapter 3 – Transport Layer

Point of transport layer services

- Logical communication between app processes on different hosts
- Send side: Break app messages into 'segments' passed to the network layer
- Receive side: reassemble segments into messages, passed to the app layer
- Transport layer vs network layer
    - Transport is logical communication between *processes*, network is logical communication between *hosts*

Multiplexing/demultiplexing

- Data is multiplexed at sender – handles data from multiple sockets, adds transport header
- Data is demultiplexed at receiver – uses transport header to deliver segments to correct socket
- Demux process
    - Datagram has source IP, destination IP, containing one transport layer segment
    - Each segment has a source and destination port number
    - Host uses IP address + port number to direct segment to appropriate socket
- Connectionless demux (UDP)
    - When sending datagram into UDP socket, need destination IP, destination port
    - When receiving UDP segment, checks destination port number, directs segment to socket with that port
    - Source IP / port does not matter
- Connection-oriented demux (TCP)
    - TCP socket identified by 4-tuple – source IP, source port, dest IP, dest port
    - Receiver uses all four values to direct segment to appropriate socket
    - Server host may support many simultaneous TCP sockets
    - Web servers use different sockets for each connecting client

User Datagram Protocol (UDP)

- 'bare bones' internet transport protocol
- 'best effort' service (segments may be lost or delivered out-of-order to app)
- Connectionless – no handshaking between sender, receiver, each segment is independent
- Used for streaming multimedia (loss tolerant, rate sensitive), DNS, SNMP
- Reliable transfer over UDP – reliability added at application layer (e.g. QUIC)

- UDP segment header: 16-bit source port #, 16-bit dest port #, 16-bit length (in bytes, including header), 16-bit UDP checksum
- UDP Checksum
    - Goal: detect 'errors' in transmitted segment
    - Sender treats segment contents (inc. headers) as sequence of 16-bit numbers
    - Sender calculates checksum by addition (one's complement sum) of segment contents
    - Sender puts the value into the checksum field
    - Receiver computes checksum of segment
    - Checks if computed checksum equals the checksum field value
    - If not, definitely errors
    - If yes, no detected error (though errors could still have occurred)

Principles of reliable data transfer

- Deals with bit errors using checksums, and ACKs and possibly NAKs to ask for retransmission of erroneous packets
- Adds sequence number to deal with duplicates
- Deals with loss by waiting for a 'reasonable' time for the ACK, retransmitting if no ACK

Pipelined protocols

- Stop-and-wait transmission causes very high latency
- Pipelining means sender allows multiple 'in-flight' unacked packets
- Go-back-N:
    - Sender can have up to N unacked packets in pipeline
    - Receiver sends cumulative ack (doesn't ack packet if there's a gap)
    - Sender has timer for oldest unacked packet, retransmitting *all* unacked packets when it expires
    - Has a 'window' of up to N consecutive unacked packets
    - There is a send_base (the first unacked packet), and a nextseqnum (the first not-yet-sent sequence number)
- Selective repeat
    - Sender can have up to N unacked packets in pipeline
    - Receiver sends individual acks for each packet
    - Sender maintains timer for each unacked packet, retransmitting only that packet on expiry
    - Packets are buffered for eventual in-order delivery to app layer
    - Sender window has N consecutive sequence numbers, sliding the window along as individual packets are acked

Transmission Control Protocol (TCP)

- Point-to-point, reliable in-order byte stream, pipelined, full duplex data (bi-directional flow), connection oriented, flow controlled
- MSS – Maximum Segment Size

TCP Segment structure

- 16-bit source port #
- 16-bit dest port #
- 32-bit sequence #

    - 32-bit ack #
    - 16-bit flagset: header length, unused padding, URG, ACK valid, PSH, RST, SYN, FIN
    - 16-bit receive window
    - 16-bit internet checksum
    - 16-bit URG data pointer
    - Variable length options
    - Variable length data
- Sequence numbers are 'byte stream' numbers of the first byte in segment's data
- Acknowledgements are sequence numbers of next byte expected from the other side
- TCP timeout – should be longer than RTT
    - Too short – premature timeout, unneeded retransmission
    - Too long – slow reaction to segment loss
- Round-trip delay time (RTT) varies – need estimation
    - Exponential weighted moving average, with typical $\alpha = 0.125$
    - $\text{EstimatedRTT} = (1 - \alpha) * \text{EstimatedRTT} + \alpha * \text{SampleRTT}$
- Timeout interval should include 'safety margin' based on variance ($\beta = 0.25$)
    - $\text{DevRTT} = (1 - \beta) * \text{DevRTT} + \beta * |\text{SampleRTT} - \text{EstimatedRTT}|$
    - $\text{TimeoutInterval} = \text{EstimatedRTT} + 4 * \text{DevRTT}$
- TCP creates reliable transfer on top of IP's unreliable service
    - Pipelined segments, cumulative acks and single retransmission timer, triggered by timeouts and duplicate acks

TCP fast retransmit

- Time-out is often quite long (leading to long delays)
- Detect lost segments via duplicate acks
    - Many segments often sent back to back, so if segment is lost, likely duplicate acks
- TCP fast retransmit – if sender receives 3 acks for the same data ('triple duplicates'), resend unacked segment with the smallest seq # without waiting for timeout

TCP flow control

- Receiver controls sender, so sender won't overflow receiver's buffer
- Receiver 'advertises' free buffer space using `rwnd` (window) value in TCP header
- `RcvBuffer` size set via socket options (typically 4096 bytes)
- Sender limits number of 'in-flight' unacked packets

Connection management

- Before data exchange, sender/receiver perform 'handshake'
- 2-way handshake has failure scenarios
- TCP uses 3-way handshake
    - Client chooses initial seq # x, and sends TCP SYN message to server
    - Server chooses initial seq # y, and sends TCP SYNACK message
    - Client then knows server is live, sends ACK for the SYNACK (this segment may contain data)
    - Server then knows client is live and transmission begins proper
- The client and server both close their side of the connection, sending TCP segments with FIN bit set
- They respond to a received FIN with ACK, which may be combined with own FIN

Congestion control

- Different from flow control!
- Congestion: too many sources sending too much data too fast for the network
- Manifests in lost packets and long delays
- Causes: queuing, finite buffers, duplicates
- Costs: more work to transmit data, unnecessary retransmission, upstream capacity wasted
- Approaches:
    - End-to-end congestion control
        - No explicit feedback from network
        - Congestion inferred from end system observed loss and delay
        - Approach used by TCP
    - Network-assisted congestion control
        - Routers provide feedback to end systems, with a single bit indicating congestion

TCP congestion control

- Additive increase (increase cwnd by 1 MSS every RTT until loss detected)
- Multiplicative decrease (cut cwnd in half after loss)
- This leads to 'saw tooth' behaviour probing for bandwidth
- $\text{rate} \approx \frac{\text{cwnd}}{\text{RTT}} \text{bytes/sec}$
- TCP Slow Start
    - When connection begins, increase rate exponentially until first loss event, starting at cwnd = 1 MSS, doubling every RTT
    - If loss is indicated by timeout, cwnd is set back to 1 MSS, growing exponentially to threshold, before growing linearly
    - If loss is indicated by 3 duplicate acks (TCP RENO), dup ACKs mean the network is capable of delivering *some* segments, so cwnd is cut in half and then grows linearly
    - Alternate: TCP Tahoe always sets cwnd to 1 MSS regardless of loss reason
    - Threshold is set at ½ of its value before the last loss event
    - Using Slow Start, avg TCP throughput $= \frac{3}{4} \frac{\text{W}}{\text{RTT}}$ bytes/sec

TCP Fairness

- If K TCP sessions share a bottleneck link with bandwidth R, each should average rate of R/K
- TCP Slow Start guarantees fairness for single connections, but parallel connections may not be fair
- UDP does not guarantee fairness (e.g. for multimedia which does not want to be throttled)

Chapter 4 – Network Layer

Role

- Transport segment from sending to receiving host, encapsulated as datagrams
- Protocols in every host and router
- Core functions: forwarding (delivering packet from input to appropriate output and routing (determining the route taken by packets from source to dest)
- Routing algorithm determines path through network, forwarding table determines local forwarding at this router
- Some network architectures also do virtual connection setup

Network service models

- Determines what the service provides, what guarantees it gives
- E.g. guaranteed delivery, guaranteed delivery with <40ms delay, in-order delivery, guaranteed minimum bandwidth
- The Internet is best-effort, with no guarantees about bandwidth, loss, order or timing, with no congestion feedback

Virtual circuit networks (VCN) (not widely used today)

- Provides connection-oriented service analogous to TCP, but it is a service, implemented in the network core
- Source to dest path behaves much like a telephone circuit
- Call setup, teardown for each call before data can flow
- Each packet carries VC identifier (not dest address)
- Every router maintains connection 'state'
- Router and link resources may be allocated to VC
- Consists of a path, VC numbers, entries in forwarding tables
- Complexity inside network, with 'dumb' end systems (ala telephones)

Datagram networks

- Connectionless service (analogous to UDP)
- No call setup at network layer
- Routers maintain no state about end-to-end connections
- Packets forwarded using destination host address
- Destination forwarding tables use longest prefix matching
    - When looking for a forwarding table entry for an address, use the longest address prefix that matches
- Complexity at network 'edge', with 'dumb' routers and 'smart' end hosts

Router functionality

- Running routing algorithms/protocols
- Forwarding datagrams from incoming to outgoing links

Input ports

- Given datagram dest, find correct output port using forwarding table ('match plus action')
- Goal: complete input processing at line speed
- If datagrams arrive faster than forwarding rate, then queuing occurs at the input
- Head-of-the-Line (HOL) blocking – queued datagram at the front prevents others moving forward

Switching fabrics

- transfer packet from input buffer into appropriate output buffer
- Switching rate: rate at which packets can be transferred from input to output (for N inputs, want switching rate N times line rate)
- Fabrics: memory, bus, crossbar
    - Memory: traditional computers, packets copied to system memory and speed limited by memory bandwidth (2 bus crossings per datagram)
    - Bus: datagram goes from input to output via shared bus, limited by bus bandwidth
    - Crossbar: overcomes bus bandwidth limitations, many possible networks

Output ports

- Send datagrams out on link
- Buffering/queuing required from faster fabric rate (datagrams may be lost due to congestion / lack of buffers)
- Can schedule packets (e.g. priority scheduling)
- Rule of thumb for buffering: use 'typical' RTT (~250ms) times link capacity C, considering N flows: $\frac{\mathrm{RTT}*\mathrm{C}}{\sqrt{N}}$

IP Network Layer contains:

- Internet Protocol,
- ICMP,
- routing protocols (RIP, OSPF, BGP)

IP Datagram format

- 4-bit IP version
- 4-bit header length (in bytes)
- 8-bit service type (the 'type' of data)
- 16-bit total length (in bytes)
- 16-bit identifier
- 3-bit flags
- 5-bit fragment offset
- 4-bit TTL (Time to live) (remaining number of hops)
- 4-bit upper layer protocol
- 8-bit header checksum
- 32-bit source IP address
- 32-bit dest IP address
- Options (variable length)
- Data (variable length)

IP fragmentation & reassembly

- Network links have an MTU (max transfer size), which is the largest possible link-layer frame (varies)
- Large IP datagram fragmented within the link layer, split into several
- Only 'reassembled' at final destination
- IP header bits (length, identifier, flags, fragment offset) used to identify and order related fragments

IPv4 addressing

- 32-bit identifier for a host or router interface
- Interface: connection between host/router and physical link (routers may have multiple interfaces)
    - One IP address for each interface
    - Host typically has one or two interfaces (e.g. Ethernet and Wi-Fi)
- Subnets: isolated networks which can physically reach each other without intervening router
    - Subnet part of address is high-order bits, host part is low-order bits
- Classless InterDomain Routing (CIDR)

    - Subnet portion of address is arbitrarily long
    - Format of a.b.c.d/x where x is the number of bits in the subnet portion of the address
- Hosts are assigned IPs either hard-coded (static) or via DHCP

Dynamic Host Configuration Protocol (DHCP)

- Allows 'plug and play' – host gets address dynamically from a server
- Process:
    - Host broadcasts 'DHCP discover' message (optional)
    - DHCP server responds with 'DHCP offer' message (optional)
    - Host requests IP address via 'DHCP request' message
    - DHCP server sends address with 'DHCP Ack' message
- DHCP can also return address of first-hop router for client, name and IP of DNS server, and network mask (indicating network vs host portion of address)
- Network is allocated a portion of the provider ISP's address space
- Addressing is hierarchical, allowing efficient advertising of routing information
- ISP gets its block of addresses from Internet Corporation for Assigned Names and Numbers (ICANN)

Network Address Translation (NAT)

- Local network uses just one IP address as far as the outside world is concerned
    - Only requires one IP address from ISP, can change addresses within local network without notifying the outside world, and local devices aren't explicitly visible to outside world (security plus)
- A NAT router must:
    - Replace source IP and port of every outgoing datagram with the NAT IP and a new port (remote hosts will respond to the NAT IP and the new port)
    - Remember (using NAT translation table) every mapping between (source IP, source port) and (NAT IP, new port)
    - Replace NAT IP and new port on incoming datagrams with the correct source IP and source port
- 16-bit port number field (~60000 simultaneous connections for a single NAT address)
- NAT is controversial
    - Violates layering (routers should only process up to Network layer, but ports are Transport layer)
    - Violates end-to-end argument (NAT must be considered by designers e.g. P2P apps)
    - Address shortage 'should' instead be solved by IPv6

Internet Control Message Protocol (ICMP)

- Used by hosts and routers to communicate network level information: error reporting, echo request/reply (e.g. ping)
- ICMP message: type, code, plus first 8 bytes of IP datagram causing error
- Traceroute
    - Source sends series of UDP segments to dest
    - First set has TTL=1, second has TTL=2 etc.
    - Uses unlikely port number

    - When nth set of datagrams arrives at the nth router, the router discards the datagrams and sends source ICMP messages (type 11, code 0) including name of router and IP address
    - When ICMP messages arrive back at source, RTT is recorded
    - Destination may return 'ICMP port unreachable'

IPv6

- Motivation: 32-bit address space soon to be completely allocated
- Also, header format helps speed processing/forwarding, allows QoS

IPv6 datagram format

- Adds priority (identify priority among other datagrams in the flow)
- Adds flow label (identify datagrams in the same 'flow' – not well defined)
- Adds next header – identify the upper layer protocol for data
- Removes checksum entirely
- Options are allowed but are outside of header (indicated by next header)
- Uses ICMPv6, featuring additional messages (e.g. Packet Too Big), multicast functions

Problems

- Not all routers can be upgraded from IPv4 to IPv6 simultaneously
- Mixed network of IPv4 and IPv6 is a problem
- Can be solved by tunnelling – Ipv6 datagram carried as payload in IPv4 datagram among IPv4 routers

Adoption

- ~3% of US industry routers
- ~11% of US government routers
- Long time for deployment – 20 years and counting

Routing algorithms (graph abstraction, routers being vertices and links being edges)

- Routing algorithm finds the least cost path between two nodes

Classification

- Global versus decentralised
    - Global: all routers have complete topology, link cost info (link state)
    - Decentralised: router knows physically connected neighbours and route costs to each, with iterative process of computation and info exchange (distance vector)
- Static (routes change slowly over time) vs dynamic (routes change more quickly)

Link-State routing algorithms – Dijsktra's algorithm

- Global (all costs known to all nodes), accomplished via 'link state broadcast'
- Iterative – after k iterations, know least cost path to k dests
- Notation:
    - c(x, y) – link cost from x to y, ∞ if not direct neighbours
    - D(v) – current value of cost of path from source to dest v
    - p(v) – predecessor node along path from source to v
    - N' – set of nodes with definitively known least cost paths
    - u – source node
- Algorithm:

```
N' = {u}
for all nodes v
    if v adjacent to u
        D(v) = c(u, v)
    else
        D(v) = ∞

while N' does not contain all nodes
    find w not in N' such that D(w) is a minimum
    add w to N'
    for all v adjacent to w and not in N'
        D(v) = min(D(v), D(w) + c(w, v))
```

- Shortest path tree constructed by tracing predecessor nodes
- Ties broken arbitrarily
- Algorithmic complexity: basic version is $O(n^2)$, can be done in O(n log(n))

Distance vector algorithm – Bellman-Ford equation (dynamic programming)

- Let $d_x(y)$:= cost of least-cost path from x to y
- Then: $d_x(y) = \min_v\{c(x,v) + d_v(y)\}$
  - Taken over all neighbours v of x
- $D_x(y)$ is estimate of least cost from x to y
  - X maintains distance vector $\mathbf{D_x} = [D_x(y): y \in N]$
- Each node x:
  - Knows cost to each neighbour v
  - Maintains its neighbours distance vectors $\mathbf{D_v} = [D_v(y): y \in N]$
- Key idea:
  - From time-to-time, each node sends its own distance vector estimate to neighbours
  - When x receives a new DV estimate, it updates its own DV using the BF equation
  - Under natural conditions the estimate converges to the actual least cost
- DV algorithm is iterative, asynchronous and distributed
- Link cost changes
  - Node detects local link cost change, updates routing info, recalculates distance vector, and notifies neighbours if need be
  - 'Good news travels fast'
  - 'Bad news travels slow' – count to infinity problem
  - Algorithm takes many iterations to stabilise
  - 'Poisoned reverse' to help solve count to infinity problem

Comparison of LS and DV algorithms

- Complexity: LS with n nodes and E links sends O(nE) messages, DV exchanges between neighbours only, with variable convergence
- Convergence speed: LS is $O(n^2)$, DV is variable as there may be routing loops and count-to-infinity problems
- Robustness: LS nodes can advertise incorrect link cost, and each node computes only its own table, DV node can advertise incorrect path cost, and each node's table used by others (errors propagate through network)

Hierarchical routing

- Algorithms so far assume 'flat' network, not true in practice

    - Flat network impossible with scale and administrative autonomy
- Routers aggregated into regions – 'autonomous systems' (AS)
- Routers in same AS run same routing protocol (intra-AS routing)
- Different ASes can run different intra-AS protocols
- Gateway router on the 'edge' of its own AS, links router to other AS
- Gateway routers configured with both intra-AS and inter-AS routing protocols
- 'Hot potato routing' – send packet towards closest of two possible routers

Routing in the internet

- Routing Information Protocol (RIP)
    - Came from BSD UNIX in 1982
    - Uses Distance Vector algorithm with distance metric of number of hops
    - DVs exchanged every 30 seconds
    - Each advertisement is a list of up to 25 destination subnets
    - If no advertisement heard after 180 second, link is declared dead
    - Routes invalidated, new advertisements sent to neighbours
    - Failure info quickly propagates to entire net
    - Poisoned reverse used to prevent ping-pong loops (infinite distance = 16 hops)
    - RIP routing tables managed by app-level process called route-d
    - Advertisements in UDP packets

Open Shortest Path First (OSPF)

- 'open' protocol – publicly available
- Uses Link State algorithm
    - Topology map at each node
    - Computation via Dijkstra's algorithm
- OSPF advertisement carries one entry per neighbour
- Advertisements flooded to entire AS
- IS-IS routing protocol is near identical to OSPF
- Provides 'advanced features' not in RIP
    - Security (authentication), multiple same cost paths (multipath), multicast support, hierarchical OSPF for large domains
        - Hierarchical: two level hierarchy (local area, backbone)

Internet inter-AS routing: Border Gateway Protocol (BGP)

- Provides each AS a means to:
    - eBGP – obtain subnet reachability info from neighbouring ASes
    - iBGP – propagate reachability information to all AS-internal routers
    - Determine 'good' routes to other networks based on reachability and policy
    - Allows subnet to advertise its existence to the internet – 'I am here'

Chapter 5 – Link Layer

- Terminology
    - hosts and routers are 'nodes'
    - Communication channels that link adjacent nodes are 'links'
    - The layer-2 pack is a 'frame'
- The link layer has the responsibility of transferring a datagram from one node to a physically adjacent node, over a link

- Datagram may use different link protocols over different links on its journey
- Datagram encapsulated in frame, adding header and trailer, with MAC addresses to identify source, dest
- Services: flow control, error detection, error correction, half-duplex or full-duplex
    - Half-duplex nodes at ends of a link can both transmit, but not at the same time
- Link layer implemented in every host via 'adaptor' or Network Interface Card (NIC) attached to host buses
- Sender encapsulates datagram in frame, adds error checking bits, reliable data transport, flow control etc.
- Receiver looks for errors, reliable data transport, flow control, extracts datagram, passing it to upper layer

Error detection

- EDC – error detection and correction bits (redundancy), D – protected data, may include headers
- Not 100% reliable (errors may be missed, rarely), larger EDC field yields better detection/correction
- Parity checking
    - Single bit parity – detects single bit errors
    - Two-dimensional bit parity – can detect and correct single bit errors
- Internet checksum
    - Detects 'errors' in transmitted packet (only used at transport layer)
- Cyclic redundancy check (CRC)
    - More powerful error-detection coding
    - Views data bits D as a binary number
    - Chooses an r+1 bit pattern (generator), G
    - Goal: choose r CRC bits, R, such that:
        - <D, R> exactly divisible by G (modulo 2)
        - Receiver knows G, divides <D, R> by G. If non-zero remainder, error detected
    - Can detect all burst errors less than r+1 bits
    - Widely used (Ethernet, Wi-Fi etc.)
    - Formula want: $\mathrm{D} * 2^{\mathrm{r}} \mathrm{XOR} \mathrm{R} = \mathrm{n} * \mathrm{G}$

Multiple access protocols

- Links can be point-to-point (PPP for dialup, point-to-point link between Ethernet switch and host)
- Links can be broadcast (old-fashioned bus Ethernet, upstream HFC, Wi-Fi)
- In multiple access protocols, there is a single shared broadcast channel
- Two or more transmissions by nodes cause interference (collision)
- A multiple access protocol is a distributed algorithm determining which nodes may transmit
    - Communication about channel sharing must use channel itself! (no out-of-bound coordination channel)
- Ideal protocol
    - Given broadcast channel of rate R bps
    - When one node wants to transmit, it sends at rate R
    - When M nodes want to transmit each sends at average rate R/M
    - Fully decentralised (no special node to coordinate transmission, no clock/slot sync)

- Simple

MAC protocol taxonomy – three broad classes

- Channel partitioning: divide channel into smaller 'pieces' (via time slots, frequency, code) and allocate a piece for each node exclusively
- Random access: Channel not divided and allow collisions, with collision recovery
- Taking turns: nodes take turns, but nodes with more to send can take longer turns

Time Division Multiple Access (TDMA)

- Access to channel in 'rounds'
- Each station gets fixed length slot (length = packet transmission time) in each round
- Unused slots go idle

Frequency Division Multiple Access (FDMA)

- Channel spectrum divided into frequency bands
- Each station assigned fixed frequency band
- Unused transmission time in a band goes idle

Carrier Sense Multiple Access (CSMA)

- 'Listen before transmit' – if channel sensed idle, transmit entire frame, if channel sensed busy, defer transmission
- Analogy: don't interrupt others
- Collisions may occur – propagation delay means nodes may not hear other's transmission
- Collision causes entire packet transmission time to be wasted

CSMA/CD (Collision Detection)

- Carrier sensing and deferral as per CSMA
- Collisions detected within short time
- Colliding transmissions aborted, reducing channel wasted
- Collision detection
  - Easy for wired LANs: measure signal strengths, compare transmitted and received signals
  - Difficult for wireless LANs: received signal strength overwhelmed by local transmission strength
- Process:
  - NIC receives datagram from layer 3, creates frame
  - If NIC senses channel idle, it starts transmission, otherwise it waits until channel idle
  - If transmission of entire frame occurs without detecting transmission, done!
  - If NIC detects another transmission while transmitting, aborts and sends jam signal
  - After aborting, NIC enters 'binary (exponential) backoff'
    - After *m*th collision, NIC chooses K at random from {0, 1, 2, …, 2^m-1}
    - NIC waits K*512 bit times before trying transmission again
    - Longer backoff interval with more collisions
- Efficiency
  - t_{prop} = max prop delay between 2 nodes in LAN
  - t_{trans} = time to transmit max size frame
  - $\text{efficiency} = \frac{1}{1+5*\frac{t_{prop}}{t_{trans}}}$
  - Efficiency goes to 1 as t_{prop} goes to 0 or as t_{trans} goes to infinity

‘Taking turns’ MAC protocols

- Tries for ‘best of both worlds’ regarding random access vs channel partitioning methods
- Polling
    - Master node ‘invites’ slave nodes to transmit in turn (typically ‘dumb’ slaves)
    - Concerns: polling overhead, latency, single point of failure at master
- Token passing
    - Control token passed from one node to the next sequentially
    - Concerns: token overhead, latency, single point of failure at node currently with token

MAC addresses

- Used ‘locally’ to get frame from one interface to another physically-connected interface
- 48-bit MAC addresses burned into NIC ROM, occasionally software-settable
    - Represented in hexadecimal with dashes or colons separating each byte
    - E.g. 1A-2F-BB-76-09-AD
    - Allocation administered by IEEE
    - Address space bought by manufacturer
- Each adapter on a LAN has a unique LAN address

Address Resolution Protocol (ARP)

- Determine interface’s MAC address, knowing its IP address
- Uses ARP table containing IP/MAC mappings for LAN nodes, with TTLs (typically forgotten after ~20mins)
- ‘Plug and play’
- Protocol (same LAN)
    - A wants to send datagram to B, but B’s MAC is not in A’s ARP table
    - A broadcasts an ARP query packet containing B’s IP address (to destination MAC of FF-FF-FF-FF-FF-FF), received by all nodes on LAN
    - B receives ARP packet, replies to A with its MAC address (sent point-to-point)
    - A caches IP-to-MAC address pair in its ARP table until information times out
- Protocol (different LAN with shared router)
    - A wants to send datagram to B (on a different LAN), via router R
    - A creates IP datagram with source A, dest B
    - A creates link layer frame using R’s MAC address as dest, frame contains A-to-B IP datagram
    - A sends frame to R, which forwards IP datagram with source A, dest B
    - R creates frame with B’s MAC address as dest
    - R forwards frame to B

Ethernet

- Dominant wired LAN technology, first widely used technologies
- Topology
    - ‘Bus’ topology popular through to mid 90s
        - All nodes in same collision domain, along the same ‘bus’
    - ‘Star’ topology used today
        - Active ‘switch’ in centre, each ‘spoke’ running a separate Ethernet protocol
        - No collisions

Frame structure

- Preamble: 8 bytes
    - 7 bytes with pattern 10101010, followed by one byte with pattern 10101011
    - Used to sync clock rates between sender and receiver
    - Dest address: 6 byte MAC address
    - If adapter receives frame with matching dest address or broadcast address, it passes the data to the network layer, otherwise it discards it
- Source address: 6 byte MAC address
- Type: indivates higher layer protocol (usually IP)
- Data (payload)
- CRC: Cyclic Redundancy Check for receiver
- Connectionless (no handshakes), unreliable (no acks etc.), uses CSMA/CD with binary backoff
- Many different Ethernet standards, with common MAC protocol and frame format
    - Different speeds (2Mbps, 10, 100, 1Gbps, 10Gbps), physical media (fibre, copper)

Ethernet switches

- Link-layer device: takes an active role
    - Stores and forwards Ethernet frames
    - Examines incoming frame's MAC address and selectively forwards frame to one or more outgoing links, uses CSMA/CD to access segment
- Transparent (hosts are unaware of switches)
- Plug-and-play, self-learning
- 'Star' topology: hosts have dedicated, direct connection to switch
- Switch buffers packets
- Ethernet protocol used on each incoming length, but no collisions because full-duplex
- A-to-A' and B-to-B' can transmit simultaneously without collisions
- Uses switch table (like routing table) mapping MAC addresses to interfaces needed to reach host, and TTL
- Self-learning
    - Switch learns which hosts can be reached through each interface
    - When frame received, switch learns location of *sender*, recording it in the table
- Frame filtering/forwarding
    - When a switch receives a frame, it records the incoming link, MAC address of sender, accesses the switch table
    - If entry found for destination in switch table
        - If destination is on the segment from which frame arrived, drop frame, otherwise forward frame selectively
    - Otherwise, flood (forward on all interfaces except the arriving interface)
- Switches may be connected together (self-learning still works
- Switches vs routers
    - Routers are network-layer, switches are link-layer
    - Both use forwarding tables

Virtual Local Area Networks (VLANs)

- Used in institutional networks with many computers
- Can define multiple virtual LANs over single physical LAN infrastructure
- Port-based VLAN

    - Switch ports grouped by switch management software, so that a single physical switch operates as multiple virtual switches
    - Example of 16-port switch divided in two
        - Traffic isolation: Frames to/from ports 1-8 can only reach ports 1-8
        - Dynamic membership: ports can be dynamically assigned between VLANs
        - Forwarding between VLANS: done via routing
        - Could also be defined based on MAC addresses rather than switch ports
- VLANs spanning multiple switches
    - Trunk port: carries frames between VLANs defined over multiple physical switches
    - Frames forwarded within VLAN can't be vanilla 802.1 frames (need VLAN ID info)
    - 802.1q protocol changes header fields for frames forwarded between trunk ports
- 802.1q VLAN frame format
    - Preamble, dest address, source address (as per Ethernet 802.1 frame)
    - 2 byte Tag Protocol Identifier, then Tag Control Information (12-bit VLAN ID field, 3-bit priority field like IP TOS)
    - Then Type, Data, CRC as per Ethernet frame

Link Virtualisation: Multiprotocol Label Switching (MPLS)

- Goal: high-speed IP forwarding using fixed length label instead of IP address
    - Gives faster lookup, borrows ideas from Virtual Circuit approach, still keeps IP address
- Inserts MPLS header between link layer and network layer headers
    - 20-bit Label, 3-bit Exp, 1-bit S, 5-bit TTL
- MPLS capable routers
    - 'Label-switched' routers
    - Forward packets to outgoing interface based only on label value without inspecting IP
    - Forwarding table distinct from IP forwarding tables
    - Flexibility: MPLS forwarding decisions can differ from those of IP
        - Use labels, dest *and* source addresses to route flows to same destination difficulty
        - Re-route flows quickly if link fails: pre-computed backup paths (fast re-route)
- MPLS signalling
    - Modifies OSPF link-state flooding protocols to carry info used by MPLS routing
        - E.g. link bandwidth, 'reserved' link bandwidth
        - Uses RSVP-TE signalling protocol to set up MPLS forwarding at downstream routers

'Day in the life' of a web request

- Scenario based on layers 5-2
- Scenario: student attaches laptop to campus networks, goes to www.google.com

Process

- Connecting laptop needs IP address, address of first-hop router, DNS server, using DHCP
    - DHCP request encapsulated in UDP, encapsulated in IP, encapsulated in Ethernet
    - Ethernet frame broadcast (dest FF-FF-FF-FF-FF-FF on LAN) received at router with DHCP server

    - Ethernet demuxed to IP demuxed to UDP demuxed to DHCP
    - DHCP server formulates DHCP ack containing client IP, first-hop router IP, name and IP address of DNS server
    - Encapsulation at DHCP server, frame forwarded through LAN (switch learning), demuxed at client
    - DHCP client receives DHCP Ack reply
    - *Client now has IP address, knows name and addr of DNS server, IP of first-hop router*
- Now needs to find IP address of www.google.com via DNS
    - DNS query created, encapsulated (UDP, IP, Eth.), but need MAC address of router interface via ARP
    - ARP query broadcast, received by router which replies with ARP reply giving MAC address
    - Client now knows MAC address of first hop router and can send DNS query frame
    - IP datagram containing DNS query forwarded via LAN switch from client to first hop router
    - Datagram forwarded from campus network onto Comcast network, routed (using tables created by RIP, OSPF and/or BGP) to DNS server
    - Demuxed at DNS server, which replies to client with IP address of google
- Now needs to send HTTP request over TCP
    - First, client opens TCP socket and sends TCP SYN segment to webserver via inter-domain routing
    - Web server responds with TCP SYNACK
    - TCP connection established
    - Client sends HTTP request into TCP socket, the IP datagram for which is routed to google
    - Web server responds with HTTP reply containing web page, routed back to client

Chapter 6 – Wireless and Mobile

Elements of a wireless network

- Wireless hosts (e.g. laptop, phone) – may be stationary or mobile
- Base station (e.g. cell tower, 802.11 access point)
    - Typically connected to wired network
    - Relay – sends packets between wired network and wireless hosts in its 'area'
- Wireless link

Wireless links

- Used to connect mobiles to base station, and also used as backbone link
- Multiple access protocol coordinates link access
- Various data rates, transmission distances
    - 802.11n transmits at up to 200Mbps, in 10-30m range
        - Older 802.11 specs: a,g at 54Mbps, b at 11Mbps, 802.15 at 1Mbps
    - Point-to-point 802.11a,g at ~50Mbps up to several km
    - 4G LTE/WiMAX broadcasts at 5-50Mbps at up to 20km
    - 3G UMTS/WCDMA-HSPDA broadcasts at ~4Mbps at up to 20km
    - 2.5G UMTS/WCDMA broadcasts at ~0.384Mbps at up to 20km
    - 2G CDMA or GSM broadcasts at ~0.056Mbps at up to 20km

Modes

- Infrastructure mode
    - Base station connects mobiles into wired network
    - Handoff: mobile can change base station providing connection into wired network
- Ad hoc mode
    - No base stations
    - Nodes can only transmit to other nodes within link coverage
    - Nodes form a network, route amongst themselves

Wireless link characteristics

- Different to wired links
- Decreased signal strength: radio signal attenuates as it propagates through matter (path loss)
- Interference: standard wireless frequencies (e.g. 2.4GHz) shared by other devices, devices like motors also interfere
- Multipath propagation: radio signals bounds off objects, arriving at destination at different times
- Signal to Noise Ratio (SNR)
    - Larger SNR means easier to extract signal from noise
    - SNR and BER (Bit Error Rate) relationship
        - Given physical layer, can increase power which increases SNR and decreases BER
        - Given SNR, choose physical layer meeting BER requirement
            - SNR may change with mobility: can dynamically adapt physical layer (e.g. modulation technique and rate)
- Hidden terminal problem – three devices (A, B, C)
    - B, A hear each other, B, C hear each other, but A, C cannot hear each other, and are therefore unaware of their interference at B
- Signal attenuation problem (similar to hidden terminal problem)

Code Division Multiple Access (CDMA)

- Unique 'code' assigned to each user for partitioning
- All users share the same frequency, but each user has own 'chipping' sequence to encode data
- Allows multiple users to coexist and transmit simultaneously with minimal interference (if codes are 'orthogonal')
- Encoded data = original data * chipping sequence
- Decoding: inner product of encoded signal and chipping sequence

IEEE 802.11 Wireless LAN (Wi-Fi)

- Uses CSMA/CA for multiple access
- All versions have base-station and ad-hoc network versions

Versions

- 802.11b – 2.4-5GHz spectrum, up to 11 Mbps, Direct Sequence Spread Spectrum (DSSS) in physical layer; all hosts use same chipping code
- 802.11a – 5-6 GHz range, up to 54Mbps
- 802.11g – 2.4-5GHz range, up to 54Mbps

- 802.11n – multiple antennae, 2.4-5GHz range, up to 200Mbps

Architecture

- Wireless host communicates with base station or 'access point' (AP)
- Basic Service Set (BSS), aka 'cell' in infrastructure mode contains wireless hosts, Aps, or hosts only in ad-hoc mode

Channels

- 802.11b: 2.4GHz – 2.485GHz spectrum divided into 11 channels at different frequencies
- AP admin chooses frequency to AP, and interference is possible (channel could be same as that chosen by neighbouring AP
- Host must associate with an AP
    - Host scans channels, listening for 'beacon frames' containing AP's name (SSID) and MAC address
    - Selects AP to associate with and may perform authentication
    - Will then typically run DHCP to get an IP address in the AP's subnet

Passive scanning

- Beacon frames sent from AP
- Association Request frame sent from host to selected AP
- Association Response frame sent from AP to host

Active scanning

- Probe Request frame broadcast from host
- Probe Response frames sent from Aps
- Association Request frame sent from host to selected AP
- Association Response frame sent from AP to host

802.11 MAC Protocol: CSMA/CA (Collision Avoidance)

- 802.11 does not use collision detections – difficult to sense collisions when transmitting due to weak received signals; also hidden terminal problem
- CSMA/CA process
    - Sender:
        - If senses channel idle for DIFS time, then transmit entire frame (no CD)
        - If senses channel busy then start random backoff time, timer counts while channel idle and transmit when timer expires; if no ACK is received, increase random backoff and repeat
    - Receiver:
        - If frame received OK, return ACK after SIFS time (ACK solves hidden terminal issue)
- Also, allow sender to 'reserve' channel rather than use random access of data frames, to avoid collisions of long data frames
    - Sender first transmits *small* request-to-send (RTS) packets to AP using CSMA
        - RTSs may still collide with each other, but they're short
    - AP broadcasts clear-to-send (CTS) in response
    - CTS heard by all nodes, sender transmits data frame and other stations defer transmission

Mobility

Spectrum of possibilities from no mobility to high mobility

- Mobile wireless user, always using same access point
- Mobile user disconnecting/connecting to network with DHCP
- Mobile user, passing through multiple access points while maintaining ongoing connections (e.g. a phone call)

Terminology

- Home network – permanent 'home' of mobile
- Home agent – entity that will perform mobility functions on behalf of mobile when remote
- Permanent address – address in home network that can *always* be used to reach mobile
- Visited network – network in which mobile currently resides
- Care-of-address – address in visited network
- Foreign agent – entity in visited network performing mobility functions on behalf of mobile
- Correspondent – wants to communicate with mobile

Problem: how to contact friend who frequently changes addresses?

- Bad idea: 'let routing handle it'
- Better idea: 'let end systems handle it'

Direct / indirect routing

- Direct routing: correspondent gets foreign address of mobile, sends directly to mobile
- Indirect routing: communication from correspondent goes through home agent, forwarded to remote

Registration

- Mobile contacts foreign agent on entering visited network
- Foreign agent contacts home agent home: 'this mobile is resident in my network'
- Then: both home and foreign agents know location of mobile

Indirect routing process

- Correspondent addresses packets using home address
- Home agent intercepts packets and forwards to foreign agent
- Foreign agent receives packets, forwards to mobile
- Mobile replies directly to correspondent
- Each mobile has two addresses: permanent and care-of-address
- Foreign agent functions may be done by mobile itself
- Triangle routing inefficient when correspondent, mobile are in same network
- Mobility, changing foreign networks transparent, so ongoing connections can be maintained

Direct routing process

- Correspondent requests foreign address of mobile from home agent, and receives it
- Correspondent forwards packets to foreign agent, which forwards them to mobile
- Mobile replies directly to correspondent
- Overcomes triangle routing problem
- Non-transparent to correspondent: correspondent must get care-of-address from home agent
- Problem if mobile changes visited network!

- Workaround: anchor foreign agent: FA in first visited network
- Data always routed first to anchor FA
- When mobile moves, new FA arranges to have data forwarded from old FA (chaining)

Chapter 7 – Multimedia Networking

Multimedia: audio

- Analog audio signal sampled at constant rate
  - Telephone uses 8000Hz, CDs are 44100Hz
- Each sample is quantised (i.e. rounded)
  - E.g. 256 possible quantised values, i.e. 8 bits per sample
- $\text{Bit rate} = \text{samples per second} * \text{bits per sample}$
  - E.g. 8000Hz, 256 values = 64 kbps
  - CD: 1.411 Mbps
  - MP3: 96, 128, 160, 320 kbps
  - Internet telephony: 5.3 kbps and up

Multimedia: video

- Sequence of images displayed at constant rate
- Each image is an array of pixels represented by bits
- Coding: use redundancy within and between images to decrease number of bits needed to encode image
  - Spatial coding: instead of sending N values of the same colour together, send just the colour and the number of repeated values
  - Temporal coding: instead of sending complete frame at i+1, send only differences from frame i
- Constant Bit Rate (CBR): video encoding rate is fixed
- Variable Bit Rate (VBR): video encoding rate changes depending on number of spatial, temporal coding changes
- E.g.:
  - MPEG1 (CD-ROM): 1.5 Mbps
  - MPEG2 (DVD): 3-6 Mbps
  - MPEG4 (< 1Mbps)

Types of multimedia applications:

- streaming stored,
- conversational,
- live streaming

Streaming stored video

- Streaming: can begin playout before downloading entire file
- Stored (at server): can transmit faster than audio/video is rendered (video is stored or buffered at client)
- Three steps: video recorded, video sent, video played
- Continuous playout constraint: once client playout begins, playback must match original timing
  - but network delays are variable (jitter), client-side buffer is required

    - In general, if average fill rate is lower than playout rate, the buffer eventually empty (causing video playout freezing)
    - If average fill rate is higher than playout rate, the buffer will not empty *so long as* the initial playout delay is large enough to absorb variability in fill rate
        - Trade-off: longer initial playout delay makes buffer starvation less likely, but means user has to wait longer to start the video
- Other challenges:
    - Client interactivity: pausing, fast-forwarding, rewind, seeking through video
    - Video packets may be lost and need retransmission

UDP multimedia streaming

- Server sends at rate appropriate for client (often send rate = encoding rate = constant rate)
    - Transmission rate may be oblivious to congestion levels
- Short playout delay (2-5 seconds) to remove network jitter
- Error recovery is application level and must be timely
- UDP may not go through firewalls

HTTP multimedia streaming

- Multimedia file retrieved via HTTP GET
- Send at maximum possible rate under TCP
- Fill rate fluctuates due to TCP congestion control and retransmissions
- Requires larger playout delay to smooth TCP delivery rate
- HTTP/TCP passes more easily through firewalls

Dynamic, Adaptive Streaming over HTTP (DASH)

- Server-side:
    - Divides video file into chunks
    - Each chunk stored, encoded at multiple different rates
    - Manifest file provides URLs for each chunk at given coding rates
- Client-side:
    - Periodically measures server-to-client bandwidth
    - Consulting manifests, requests one chunk at a time
        - Chooses maximum sustainable coding rate given bandwidth
        - Can choose different coding rates at different points in time based on bandwidth variability
- 'Intelligence' is at client, as client decides:
    - When to request chunk (to prevent buffer starvation or overflow)
    - What encoding rate to request (higher quality with more bandwidth available)
    - Where to request chunk from (Can request from URL server 'close' to client or with high available bandwidth)

Voice-over-IP (VoIP)

- End-to-end delay requirement: needed to maintain 'conversational' aspect
    - Higher delays are noticeable, and impair interactivity
    - < 150ms is good, > 400ms is bad
    - Included application-level network delays
- Session initialisation: how does callee advertise IP address, port, encoding algorithms?
- Value-added services: call forwarding, screening, recording

- Emergency services

Characteristics

- Speaker's audio: alternating talk spurts, silent periods
- 64 kbps during talk spurt
- Packets only generated during talk spurts
- Split into 20 ms chunks at 8 KB/s (160 bytes)
- App-level header adder to each chunk and encapsulated into UDP or TCP, sent every 20ms during talkspurt

Delay

- Network loss: datagram loss due to network congestion (router buffer overflow)
- Delay loss: datagram arrives too late for playout at receiver due to delays through network
  - ~400ms is typical max tolerable delay
- Loss tolerance: depending on voice encoding and loss concealment, loss rates of 1%-10% can be tolerated

Fixed playout delay

- Receiver attempts to playout each chunk exactly q ms after chunk was generated
  - If chunk has time stamp t: play out chunk at t+q, if it arrives after t+q data 'lost'
  - Trade-off: large q means less packet loss, small q provides better interactivity

Adaptive playout delay

- Estimates network delay using exponentially-weighted moving average
  - d_i = delay estimate after *i*th packet
  - r_i = time received
  - t_i = time sent (timestamp)
  - α = small constant, typically ~0.1
  - $d_i = (1-\alpha)d_{i-1} + \alpha(r_i - t_i)$
- Adjusts playout delay at the beginning of each talk spurt
- Silent periods are compressed and elongated
- Chunks still played out every 20ms during talk spurt
- Receiver determines whether a packet is the first in a spurt:
  - If no loss, use successive timestamps (if > 20ms between, new talk spurt)
  - If loss possible, also consider sequence numbers

Recovering from packet loss

- Using ACKs/NAKs introduces delay
- Instead, use Forward Error Correction (FEC)
  - Send enough bits to allow recovery without retransmission (e.g. 2D parity)
  - Simple FEC process:
    - For every group of n chunks, create redundant chunk by XORing originals
    - Send n+1 chunks, increasing needed bandwidth by factor of 1/n
    - Can reconstruct original chunks if at most chunk is lost from the n+1 chunks

Real-Time Protocol (RTP)

- Runs on end systems
- Specifies packet structure for packets carrying audio, video data
- Provides payload type identification, sequence numbering, time stamping

- RTP packets encapsulated in UDP segments (essentially, transport-layer interface extending UDP)
- Interoperable: if two VoIP applications use RTP, they may be able to work together
- Does not provide mechanisms to ensure timely delivery or Quality of Service (QoS) guarantees
- RTP encapsulation only seen at end systems, not intermediate routers

RTP header structure

- 7-bit Payload type: the type of encoding currently being used (may change during 'call')
    - E.g. type 0 – PCM mu-law, 64kbps, type 33, MPEG2 video
- 16-bit sequence number: increment by one for each RTP packet sense (to detect packet loss and restore sequence)
- 32-bit timestamp: sampling instant of first byte in the RTP packet
    - For audio, timestamp increments by one for each sampling period
    - E.g. if app generates chunks of 160 encoded samples, timestamp increases by 160 for each RTP packet while source is active. Timestamp continues to increase when source is inactive
- 32-bit SSRC: identifies source of the RTP stream, each stream in session has distinct SSRC

Real-Time Control Protocol (RTCP)

- Used with RTP
- Reports statistics useful to application – num packets sent, lost, jitter
- Feedback used to control performance
- Typically, each participant in RTP session periodically sends RTCP packets to others
- Types: receiver report packets, sender report packets, source description packets
- RTCP can synchronise media streams within an RTP session, e.g. video/audio streams
- Bandwidth scaling: RTCP attempts to limit its traffic to 5% of session bandwidth

Session Initiation Protocol (SIP)

- Designed to replace traditional telephone calls (using the internet)
- Uses 'Keep It Simple Stupid' principle
- People identified by names or email addresses, rather than phone numbers
- Can reach callee no matter where callee roams, no matter what IP device they use
- SIP provides mechanisms for call setup / teardown, determining current IP address of callee, and call management
- Can be sent over TCP or UDP, potentially over RTP+UDP
- Call setup provides codec negotiation, call rejection
- SIP registrar function: when a user starts SIP client, they send a SIP REGISTER message to their registrar server (which maps their current IP to their persistent address)
- SIP proxy: server can route SIP messages to callee, possibly through multiple proxies
- Similar to H.323
    - H.323 is complete vertically integrated suite of protocols, whereas SIP is a single component
    - H.323 is 'telephony flavoured', SIP is 'web flavoured'

Network support for multimedia

Best effort service

- Treats all traffic equally, with little to no guarantee, no network support. Low complexity and wide availability

Dimensioning best effort networks

- Try to deploy enough link capacity to prevent congestion
- Low complexity of network mechanisms, but high bandwidth costs
- Challenges: network dimensioning (what is 'enough' bandwidth?), estimating traffic demand

Providing multiple classes of service

- Partition traffic into classes, which network treats differently (e.g. VIP service vs regular)
- Provides granularity amongst multiple classes, not among individual connections
- Principles:
    - Packet marking needed for routers to distinguish between classes, with router policy to treat packets accordingly
    - Provide protection (isolation) for one class from others with policing
    - While providing isolation, try to use resources efficiently (allocating fixed bandwidth causes inefficiency if flow doesn't use its allocation)

Differentiated services

- Uses traffic 'classes', provides little to no guarantee, network uses packet market, scheduling, policing. Moderate complexity and some availability
- Want qualitative service classes, which are scalable (simple functions in network core, complex functions at edge routers or hosts)
- Don't define service classes, provide functional components to build classes
- Diffserv:
    - Edge router: per-flow traffic management, marks packets in-profile and out-profile
    - Core router: per-class traffic management, buffering/scheduling at edge, preference given to in-profile over out-profile packets
    - Packet marked using Type-of-Service (ToS) in IPv4, Traffic Class in IPv6
    - 6-bit Differentiated Service Code Point (DCSP), 2-bits currently unused

Per-connection QoS

- Provides per-connection flow granularity, with potential for strong guarantees, using packet market, scheduling, policing, call admission. High complexity and little to no deployment
- Principle: flow declares its needs; network may block call if it cannot meet needs
- Resource reservation: call setup, signalling (RSVP); traffic, QoS declaration; per-element admission control

Chapter 8 – Security

Basics

- Confidentiality: only sender, intended receiver should 'understand' message contents
- Authentication: sender, receiver want to confirm each other's identity
- Message integrity: sender, receiver want to ensure message has not been altered (in transit or afterwards) without detection
- Access and availability: services must be accessible by users

What can a 'bad guy' do?

- Eavesdrop: intercept messages
- Insert messages into the connection
- Impersonation: can spoof source address (or any field in packet)
- Hijacking: 'take over' ongoing connection by removing sender or receiver, inserting themselves
- Denial of service: prevent service being used by others (e.g. overloading resources)

Principles of Cryptography

- Terminology
    - m = plaintext message
    - $\mathrm{K_A(m)}$ = ciphertext, encrypted with key $\mathrm{K_A}$
    - $\mathrm{m = K_B(K_A(m))}$

Breaking encryption

- Cipher-text only attack (attacker has ciphertext to analyse)
- Brute force: search all keys
- Statistical analysis
- Known-plaintext attack
    - Attacker has plaintext corresponding to cyphertext
- Chosen-plaintext attack
    - Attacker can get ciphertext for chosen plaintext

Symmetric key cryptography

- Bob and Alice share same symmetric key $\mathrm{K_S}$ - i.e. $\mathrm{m = K_S(K_S(m))}$
- Simple example: substitution cipher
- More sophisticated: n substitution ciphers M1, M2, …, M2 with cycling pattern
    - For each plaintext symbol, use next substitution pattern cyclically
    - Encryption key – the pattern

Data Encryption Standard (DES)

- US encryption standard, with 56-bit symmetric key, 64-bit plaintext input
- Block cipher with block chaining
- Not very secure: 56-bit key phrase can be brute-force decrypted in under a day
- No known good analytic attack
- Can be made more secure with 3DES (encrypt 3 times with different keys)

Advanced Encryption Standard (AES)

- NIST symmetric-key standard, replaced DES
- Processes data in 128-bit blocks
- Uses 128, 192 or 256-bit keys
- Brute force decryption taking 1sec on DES takes 149 trillion years for AES

Public key cryptography

- Sender, receiver do not share secret key
- Each has a 'public' encryption key known to all, and a 'private' decryption key known only to receiver – i.e. $\mathrm{m = K_B^-(K_B^+(m))}$
- Rivest, Shamir, Adelson (RSA) algorithm
    - Keys work in either order – $\mathrm{K_B^-(K_B^+(m)) = m = K_B^+(K_B^-(m))}$

    - Very secure – knowing public key, need to find huge prime factors, which is hard
    - Slow performance: Exponentiation in RSA is computationally expensive
        - DES is at least 100x faster than RSA
    - Solution to performance: use public key crypto to establish secure connection, then establish and exchange symmetric session key for encrypting data

Message Integrity and Digital Signatures

Digital signatures

- Crypto technique analogous to hand-written signatures
- Sender digitally signs document, establishing they are document owner/creator
- Verifiable, non-forgeable: recipient can prove to someone that only the sender could have signed the document
- E.g. simple digital signature: signing by attaching the message with a version encrypted with your private key
    - Anyone can decrypt with public key, but only you could have encrypted it
    - Recipient decrypts signature, and checks it matches the message
- Message digests: computationally expensive to use public key crypto on long messages
    - Instead, apply hash function to m to get fixed-size message to encrypt
    - Internet checksum is like a hash function, but it is a bad crypto hash function, because given a hash value, it is easy to find another message with the same hash
    - MD5 hash – 128-bit message digest computed in 4-step process
    - SHA-1 – 160-bit message digest, US standard
- Problem: public keys need to be certified (i.e. registered so an attacker can't pretend a public key is someone else's)
    - Certificate Authority (CA) binds a public key to an entity E
    - When someone wants someone else's public key, they get their certificate, apply the CA's public key to the certificate to get the person's public key

Operational Security

Firewalls

- Isolates organisation's internal net from the internet, allowing only some packets through
- Used to prevent denial of service, illegal access of data
- Examines headers only (not payload)
- Stateless packet filtering
    - Internal network connected to internet via router firewall
    - Router filters packet-by-packet, based on source/dest IP, TCP/UDP source and dest ports, ICMP message type, TCP SYN and ACK bits
    - Often heavy handed, can't consider what 'makes sense'
- Stateful packet filtering
    - Track status of every TCP connection (including setup/teardown)
    - Timeout inactive connections
    - Checks that packets 'make sense'
- Application gateways
    - Filters packets on application data as well as IP/TCP/UDP fields
- Issues: IP spoofing, multiple applications need their own gateways, client software needs setup, filters are often 'all or nothing' for UDP
- Trade-off between degree of communication with the outside world and security

Intrusion Detection Systems (IDS)

- Deep packet inspection: look at packet contents (e.g. check character strings against database of known virus and attack strings)
- Examine correlations between multiple packets (e.g. port scanning, network mapping, DoS attacks)
- Often use multiple IDSs to do different kinds of checking at different locations

Chapter 9 – Network Management

Network management

- Autonomous systems ('network'): thousands of interacting hardware/software components
- Other complex systems which require monitoring and control: aeroplane, nuclear reactor…
- "Network management includes the deployment, integration and coordination of the hardware, software and human elements to monitor, test, poll, configure, analyse, evaluate and control the network and element resources to meet the real-time, operational performance, and QoS requirements at a reasonable cost"

Network management infrastructure

- Managed devices contain managed objects whose data is gathered into a Management Information Base (MIB)

Standards

- OSI Common Management Information Protocol (CMIP)
    - Designed in the 80's as a standard, but never took off
- Simple Network Management Protocol (SNMP)
    - De facto network management standard