# Contents

Notes taken wholly or mostly from the 5th edition have black section headings. The notes from section 5 (Network layer) onward are all taken wholly from the 5th edition.

For sections 1-4, I mostly copied notes I had taken from the 4th edition (mostly, I did that when section headings were the same). 4th edition notes are marked with light gray section headings. Where a section is copied wholesale from the 4th edition, expect that references to Figures are in sync with the 5th edition rather than the 4th.

Where a section contained nothing noteworthy (to me) in either edition, the heading is in black.

Some content got reorganized between the two editions. For example, in the 4th edition, physical layer multiplexing strategies were covered in sections that described their use in a specific context (e.g., CDMA was described in the mobile telephony section, others were described along with POTS), but in the 5th edition they form their own subsection. Because of this, in some instances a single topic may be covered in-depth twice.

Here's a list of sections in the 4th edition not in the 5th edition:

- 1.2.4 Network Hardware: Wireless Networks
- 1.2.5 Network Hardware: Home Networks
- 1.5.2 Example Networks: Connection-Oriented: X.25, Frame Relay, ATM
- 1.5.3 Example Networks: Ethernet
- 3.5 Protocol Verification (too bad, this one was really cool)
- 4.2.5 Multiple Access Protocols: WDMA (Wavelength Division Multiple Access Protocols)
- 4.3.1 Ethernet: Ethernet Cabling
- 4.3.2 Ethernet: Manchester Encoding
- 4.3.9 Ethernet: IEEE 802.2: Logical Link Control
- 4.3.10 Ethernet: Retrospective on Ethernet
- 4.6.5 Bluetooth: The Bluetooth Baseband Layer
- 4.6.6 Bluetooth: The Bluetooth L2CAP Layer
- 4.7.1 Data Link Layer Switching: Bridges from 802.x to 802.y
- 4.7.2 Data Link Layer Switching: Local Internetworking
- 4.7.4 Data Link Layer Switching: Remote Bridges

–Z

, sections 5 onward entirely from the 5th edition. Sections in gray are copied from the 4th edition notes nearly verbatim. Sections with black headings are notes from the 5th edition.

# 1 Introduction

## 1.1 Uses of Computer Networks

### 1.1.1 Business Applications

### 1.1.2 Home Applications

### 1.1.3 Mobile Users

### 1.1.4 Social Issues

## 1.2 Network Hardware

- Broadcast networks
  - ¤ broadcasting: message is received by every machine on the network
  - ¤ multicasting: transmission to some subset of machines on the network
  - ¤ point-to-point networks: many connections between individual pairs of machines
  - ¤ unicasting: P2P transmission with one sender and one receiver

### 1.2.1 Personal Area Networks

### 1.2.2 Local Area Networks

- topology
  - ¤ bus: a linear cable
    - ∗ only 1 machine (the master) transmits at once
    - ∗ requires an arbitration mechanism
    - ∗ Ethernet (IEEE 802.3) has decentralized control
      - > in the case of colliding packets, each machine waits a random time and then tries again
  - ¤ ring: circular cable
    - ∗ Each bit propagates on its own
    - ∗ also requires an arbitration mechanism (e.g., machines take turns)
    - ∗ e.g., IBM token ring (IEEE 802.5) and FDDI
- Broadcast networks can also be divided by allocating channels
  - ¤ static channel allocation
    - ∗ e.g., a round-robin algorithm
      - > Each machine has a time slice during which it may transmit
    - ∗ static forms tend to waste capacity when a machine has nothing to say
  - ¤ dynamic channel allocation: centralized or decentralized
    - ∗ centralized: a single entity (e.g., bus arbitration unit) determines who goes next
    - ∗ decentralized: each entity must itself decide whether to transmit

### 1.2.3 Metropolitan Area Networks

### 1.2.4 Wide Area Networks

- hosts connected by a subnet
- subnet is defined as a set of transmission lines and switching elements
  - ¤ (subnet now also used to describe a meaning related to network addressing)
- WANs in which members don't directly connect usually can still transmit to each other by passing data through intermediaries.
  - ¤ Store-and-forward/packet-switched subnet: an intermediary router accepts the entirety of a packet until it can forward it along.
  - ¤ Small packets of the same size are often called cells.
- Packet-switched LANs
  - ¤ message gets split into packets
  - ¤ Depending on network, packets might be routed separately, or may have to follow the same paht.
- Non-packet-switched networks
  - ¤ satellite system: routers mostly only hear the staellite's broadcast (and almost never each other)

### 1.2.5 Internetworks

## 1.3 Network Software

### 1.3.1 Protocol Hierarchies

- Layer n on one machine is a peer to layer n of another machine, they converse via the same protocol
- Between each pair of adjacent layers is an interface
- A set of layers and protocols is called a network architecture
- A list of protocols used by a system is its protocol stack.
- Fig 1-15: Whatever mangling lower layers do to aidin transmission, it must be well abstracted so that peer processes can imagine their communications are horizontal

### 1.3.2 Design Issues for the Layers

- Every layer needs to be able to designate senders and receivers, so all need a means of addressing
- Data transfer rules:
  - ¤ Directionality (one, both)
  - ¤ Number of logical channels (normal, urgent, ...)
- Error control
- Out-of-sequence messages / Order-preservation
- Flow control - ensuring a fast sender doesn't burden a slow receiver.
- Grouping small and splitting large messages to avoid inefficients
- Multiplexing and demultiplexing
- Routing

### 1.3.3 Connection-Oriented Versus Connectionless Service

- Layers can offer these kinds of service to the layer above them
  - ¤ Connection-oriented
    - ∗ Pattern: establish, use, then release the connection
    - ∗ May involve an initial negotiation about parameters
  - ¤ Connectionless

    - Each message is independent
  - Connection-oriented has higher quality of service, but tends to be slower
  - Connection-oriented Types
    - Messages sequences: Message boundaries are preserved
    - Byte stream: Consecutive bits are freely combinable/splittable
  - Unreliable (aka unacknowledged) connectionless services is often called datagram service
  - Acknowledged datagram service: Connectionless but gets a confirmation it was received.
  - Request-reply service: Sender transmits a single datagram with a request, reply contains the answer.
- prefer unreliable communication
  - Reliable isn't always available (Ethernet isn't, for example), leaving it up to higher level protocols to deal with the fallout.
  - Providing reliable service may result in unacceptable delays.
    - e.g., roundtrip ACK times required to make video and voice calls possible

#### 1.3.4 Service Primitives

- A service is formally specified by a set of *primitives* (operations) available to user processes to access the service. The set of primitives exposed will depend on the nature of the service being provided.

#### 1.3.5 The Relationship of Services to Protocols

- A *service* is a set of primitives that a layer provides as an interface to a layer above it; is usually implementation-agnostic
- A *protocol* is a set of rules governing the format & meaning of messages exchanged by peer entities within a layer.

### 1.4 Reference Models

- 2 example network architectures
- OSI reference model
  - model is still general and valid but protocols are no longer used
- TCP/IP reference model
  - model is not used, but protocols are widely used

#### 1.4.1 The OSI Reference Model

- ISO: Open Systems Interconnection
- Figure 1-20
- 7 layers came about so that each layer is a different abstraction, i.e.
  - Each performs a well-defined function
  - information flow across interfaces is minimal
- The layers
  - Physical
    - Makes sure that if 1 is sent, 1 is received
      - > e.g., voltages, duration per bit, connectors used, how to establish a connection
  - Data link
    - Transform raw transaction facility into a stream that pretends to be error-free
    - Breaks input into data frames (100s-1000s of bytes)
    - Sends sequentially
    - expects an acknowledgement frame
    - regulate flow (to protect against fast transmitters swamping slow receivers)
    - Handle errors

    - Broadcast networks also integrate channel sharing access control here via a 'medium access control sublayer'
  - Network layer
    - All operations relating to the subnet
    - Routing from source to destination
      - via static knowledge of network topology
      - or via per-session knowledge
      - or via some determination made per-packet
    - Congestion control
    - Quality of service (delay, transmit time, jitter, etc.)
    - Cross-network compatibility
      - Packet size
      - Addressing
      - Protocols
    - Tends to be very simple in broadcast networks
  - Transport layer
    - Accept data, split it up, ensure it arrrives correctly
    - Popular options in the transport layer
      - "Error-free": point-to-point delivery of messages/bytes in order sent
      - Transport isolated messages, with no guarantee of order of delivery
      - broadcast to multiple destinations
    - The lowest end-to-end layer
      - i.e., lowest layer that only source and destination pay attention to
      - Lower layers must take into account intermediate machines
  - Session layer
    - Establish sessions between users, allowing:
      - Dialog control: whose turn is it to transmit
      - Token management: Prevent multiple partners from performing the same critical action at the same time.
      - Synchronization: checkpointing long transmissions so they can be resumed if interrupted
    - Presentation layer
    - Syntax and semantics of data transferred
  - Application layer
    - e.g., HTTP, FTP, SMTP, NNTP, POP, IMAP

### 1.4.2 The TCP/IP Reference Model

- Came about when ARPANET needed to expand to include satellite and radio networks
- Since it's DoD, network needs to be robust
  - i.e., stay up if some hardware is lost to attack, with existing connections not breaking
- Internet Layer
  - Permit hosts to inject packets into any network
  - Packets travel independently to destination (perhaps via other networks)
  - Packets can arrive out-of-order
    - higher layers can choose how to fix
  - Defines the packet format and protocol as the *Internet Protocol*
  - Packet routing and congestion avoidance are its main gials
    - Thus, considered analogous to OSI model's network layer
- Transport Layer
  - Like the OSI transport layer
  - Is concerned with permitting source & destination hosts to carry on an uninterrupted conversation
  - 2 end-to-end transport protocols defined here
    - Transmission Control Protocol

        > Reliable, connection-oriented
        > Lets a byte-stream originating on one machine get delivered to another without errors
        > On source, fragments byte streams into discrete messages
        > On destination, reassembles discrete messages into byte streams
        > Also does flow control
      ∗ User Datagram Protocol
        > Unreliable, connectionless
        > used to get prompter delivery

- Application layer
  - ¤ e.g., Telnet, FTP, SMTP, DNS, NNTP, HTTP
- Host-to-network layer
  - ¤ TCP/IP doesn't care how IP packets get from one host's Internet Layer (lol, great trolling)

### 1.4.3 The Model Used in This Book

- Since the strength of OSI is the model itself, and the strength of TCP/IP is the protocols (which at this point are quite robust and well-tested), the focus will be on a blend of the two.
- Application, Transport, Network, Link, Physical

### 1.4.4 A Comparison of the OSI and TCP/IP Reference Models

- OSI very clearly distinguishes between *services*, *interfaces*, and *protocols*
  - ¤ A layer's service definition says what a layer does (its semantics)
  - ¤ A layer's interface tells processes above how to access it (parameters and results – a.k.a., syntax)
  - ¤ Protocols in a layer are its own business entirely
- OSI thus fits much better with modern (OO-esque) ideas of good abstractions
- TCP/IP was only later retrofitted to such designs
- OSI was designed before corresponding protocols, so in some ways its one-size-fits-all design doesn't fit anything all that well
  - ¤ e.g., data link layer not fitting with broadcast networks
  - ¤ e.g., grafting 'convergence sublayers' onto the model
- TCP/IP, meanwhile, has no generality across other protocol stacks
- Connections
  - ¤ OSI can be connectionless or connection-oriented in network layer
  - ¤ OSI can only be connection-oriented at transport layer
  - ¤ TCP/IP can only be connectionless in network layer
  - ¤ TCP/IP can be connectionless or connection-oriented in transport layer

### 1.4.5 A Critique of the OSI Model and Protocols

- Standard arrived too late–after TCP/IP was already widespread
- Bad tech
  - ¤ Two layers nearly empty (session and presentation)
  - ¤ 2 layers are overfull (data link and network)
  - ¤ Standard is too complex to implement
  - ¤ Too inefficient
  - ¤ Some functions reappear in every layer
    - ∗ addressing
    - ∗ flow control
    - ∗ error control
- Bad implementations

  - ¤ Overcomplicated standard led to huge, unwieldy, slow implementations
  - ¤ This was a huge hardship on even those eager to adopt it
- Bad politics
  - ¤ TCP/IP was thought to be part of UNIX
  - ¤ OSI was thought to be a collusion of European telecoms, European governments, and US government

#### 1.4.6 A Critique of the TCP/IP Reference Model

- No distinction between specification and implementation
- Not generalizable to other protocol stacks (e.g., bluetooth)
- Host-to-network layer is actually an interface
- No distinction between physical & data link layers
- Many app level protocols are rubbish (e.g., Telnet)

## 1.5 Example Networks

#### 1.5.1 The Internet

- History of ARPANET
  - ¤ TCP/IP to allow communication over internetworks
  - ¤ UCB created sockets in Berkeley UNIX to integrate TCP/IP
  - ¤ DNS invented to reduce cost of finding hosts
- Formation of ANS/Commercialization of networking
  - ¤ NSFNET backbone
    - ∗ was meant to allow non-DoD university research groups to connect with one another
    - ∗ was overwhelmed basically the moment it was turned on
  - ¤ Same with NSFNET v2
  - ¤ NSF noticed this was costly, convinced others to form ANS (Advanced Networks and Services) which created ANSNET
- ANS
  - ¤ 4 nationwide NAPs (Network Access Points)
    - ∗ All backbone operators needed to be able to connect to all four
    - ∗ Backbone operators also had to compete, then, to get traffic from the regional networks
- Architecture of the Internet
  - ¤ Point of Presence: the local office where your ISP snatches IP traffic off of POTS or cable line and sends it into the regional network
  - ¤ Carrier Hotels: rackspace rented out by backbones, in same room as their routers usually, that allow for faster connections
  - ¤ Private Peering: backbone routers that connnect directly to bypass NAPs

#### 1.5.2 Third-Generation Mobile Phone Networks

- 1G *AMPS* (Advanced Mobile Phone System) which transmitted continuously varying analog signals, was deployed in the US in 1982
- 2G switched to digital transmission to increase capacity, improve security, and allow for SMS
  - ¤ GSM, first deployed in 1991, is a 2G system
- 3G (really a lot of different standards) was first deployed in 2001.
  - ¤ *UMTS* (Universal Mobile Telecommunications System), aka WCDMA (Wideband Code Division Multiple Access), is the main system worldwide
- Radio spectrum scarcity/cost is one of the main limits in *cellular network* design

  - ¤ Coverage areas are divided into cells such that neighboring cells intefere with each other minimally. This allows *frequency reuse* without interference.
- Mobile phone network architecture
  - ¤ *Cellular base station* (aka *Node B*) is the only type of entity a mobile device actually communicates directly with
  - ¤ *Air interface* is the protocol used for mobile devices to talk to cellular base stations
  - ¤ *Radio Network Controller* (RNC) controls how spectrum is used
  - ¤ A cellular base station and its RNC are the *radio access network*
  - ¤ *Core network* is the rest of the mobile phone network, which carres traffic to/from a radio access network. The core network can contain both circuit-switched and packet-switched equipment
- User mobiligy creates unique challenges, such as *handover/handoff*, which is the process that needs to occur as a device changes service area (often the process is triggered when signals get weak). Older systems only supported *hard handovers*, where the device would leave one network before joining the next (and drop the call if it failed to join). Modern systems usually support *soft handovers*, where the mobile device is briefly on two networks at once.
- *Home Subscriber Server* (HSS)
  - ¤ Each mobile device is associated with a single HSS, which it registers with as it changes service areas. The HSS knows how to authenticate, authorize, and locate the mobile device's current service area.
- Security
  - ¤ *SIM* card (Subscriber Identity Module) is a removable chip with info a handset can use to authenticate the user with the network, and prevent the handset from joining an illegitimate network. SIM cards store keys used to encrypt the user's broadcast traffic.

### 1.5.3 Wireless LANs: 802.11

- With base station (access point) or without base station (ad-hoc networking)
- Challenges
  - ¤ finding a frequency band available worldwide
  - ¤ dealing with finite range
  - ¤ maintaining users' privacy
  - ¤ not draining batteries too much
  - ¤ ensuring a useful amount of bandwidth
  - ¤ Reconciling big differences in physical & data link layers
    - ∗ simultaneous transmissions
      - > reachable-by-wireless is not transitive
      - > unlike ethernet, can't just listen to make sure nobody else is already transmitting
      - > so two simultaneous transmitters could interfere with one another, or otherwise swamp a receiver
    - ∗ interference by multipath fading
      - > radio signals bound around and are received multiple times via multiple paths
    - ∗ software makes assumptions of non-mobility
    - ∗ computers need to have graceful handoff between base stations as they're moved
- A connection betweeen an 802.11 system and the outside world is a *portal*

### 1.5.4 RFID and Sensor Networks

- *RFID* (Radio Frequency IDentification): a small microchip with a unique identifier, and an antenna that can receive radio transmissions, and respond back in some identifying way.
- *passive RFID* contains no power source, and its response is powered only by the request it receives
- *active RFID* contains a power source
- *UHF RFID/backscatter*: readers send a signal, and the tags change the signal before reflecting it. Readable from a range of several meters. Often used for shipping pallets.
- *HF RFID* (High-Frequency RFID): has a short range (1m or less), and is used for contactless payment. Its range is short because it relies on induction rather than backscatter.
- *LF RFID* is used for animal tracking

- Collisions are problematic: if all tags respond to a read immediately, collisions will occur. Instead tags wait a short, random duration.
- RFID tags are too small to iplement cryptography, so it's hard to make it so they aren't trackable by malicious parties
- *Sensor network*
  - Put out a bunch of sensor nodes that can collect measurements, and have them deliver data to some collection point
  - Often require batteries or recharging mechanisms
  - Sometimes relay messages (along a *multi-hop network*) to the collection point

## 1.6 Network Standardization

### 1.6.1 Who's Who in the Telecommunications World

- In US, *common carriers* must describe offerings and prices in a document called a *tariff*, which must be approved by the FCC
- ITU: International Telecommunication Union (part of the UN)
  - ITU-R: Radiocommunications Sector
    - allocates frequencies worldwide
  - ITU-T: Telecommunications Standardization Sector
    - 4 kinds of members
      - national governments
      - Sector members: telecoms, equipment manufacturers/vendors, media companies, science nonprofits
      - associate members: smaller organizations usually interested in just one Study Group
      - regulatory agencies: e.g., the FCC
    - Recommends standards, but these recomendations are treated as law by anyone who has any desire for interoperability
    - 14 Study Groups, each covering a topic

### 1.6.2 Who's Who in the International Standards World

- ISO
  - Includes standards orgs of its 89 member countries, e.g. ANSI
  - 200 technical committes
    - each divides into subcommittees
      - each subcommittee divides into working groups
    - Lifecycle of a standard
  - Desire to internationalize a standard
  - working group forms to create a Committee Draft
  - CD criticized for 6 months by member bodies
  - if approved, a revised Draft International Standard is circulated for review
  - eventually, the final text, if approved, is an International Standard
- ANSI (American National Standards Institute)
  - a private NGO
- NIST (National Institute of Science and Technology)
  - determines what standards the US government may buy
- IEEE
  - Also organized into working groups
  - Networking WGs are in the 802 namespace

### 1.6.3 Who's Who in the Internet Standards World

- IAB (Internet Architecture Board)
  - ¤ They create RFCs
  - ¤ Now split into ???????
    - ∗ IRTF (Internet Research Task Force)
    - ∗ IETF (Internet Engineering Task Force)

## 1.7 Metric Units

## 1.8 Outline of the Rest of the Book

## 1.9 Summary

# 2 The Physical Layer

## 2.1 The Theoretical Basis for Data Communication

### 2.1.1 Fourier Analysis

- Any reasonably behaved periodic function g(t) with period T can be constructed as $g(t) = \frac{c}{2} + \sum_{n=1}^{\infty} a_n \, sin(2\pi nft) + \sum_{n=1}^{\infty} b_n \, cos(2\pi nft)$.
  - ¤ for f = 1/T, the fundamental
  - ¤ for $a_n$, $b_n$ = the sine and cosine amplitudes of the n'th harmonics (terms)
  - ¤ for c = some constant

### 2.1.2 Bandwidth-Limited Signals

- *Bandwidth*: The range of frequencies transmitted without being strongly attenuated.
- Limithing the bandwidth limits the data rate

### 2.1.3 The Maximum Data Rate of a Channel

- *Nyquist's Theorem*: maximum data rate (assuming no noise) = $2H \; log_2 V \frac{bits}{second}$
  - ¤ if the signal is made up of V discrete levels
  - ¤ if the signal has bandwidth H
    - ∗ due to low pass filter or other restrictions of the medium
- *Signal-to-noise*: For S=signal power, N=noise power,
  - ¤ $10 log_{10} \frac{S}{N}$ is the number of deciBels,
  - ¤ $\frac{S}{N}$ is the signal-to-noise ratio
- Shannon showed that maximum number of bits/second = $H \; log_2(1 + \frac{S}{N})$
  - ¤ for signal-to-noise ratio S/N
  - ¤ for a channel with a bandwidth of H Hz

## 2.2 Guided Transmission Media

### 2.2.1 Magnetic Media

- Store data on hard drives, ship them. Unbeatable for bandwidth and cost, but latency is high.

### 2.2.2 Twisted Pairs

- Twisting the wires prevents them from acting as an antenna
  - i.e., attenuating the signal by radiating it outward
- Cat 3, Cat 5, ... wires refer to how many twists/centimeter
  - more twists =¿ less crosstalk =¿ less noise =¿ higher bandwidth
- Referred to as Unshielded Twisted Pair to contrast with a standard put forth by IBM

### 2.2.3 Coaxial Cable

- Well-shielded copper wire
- Works well for longer distances and faster transmission speeds

### 2.2.4 Power Lines

- Data signal is superimposed on the power signal
- Difficulties
  - Electrical signals are normally sent at 50-60Hz. Wiring attenuates signals at frequencies high enough for fast data transfer. So there are not many frequencies available for data transfer over this medium.
  - Properties of the wire vary per-house, and as the set of plugged-in devices changes
  - Wiring acts as an antenna, radiating interference
- These constraints make it hard to apply outside of individual residences.
- It's practical to send at 100Mbps over typical household wiring, if using schemes that resist impaired frequencies and error bursts.

### 2.2.5 Fiber Optics

- Optical transmission system has 3 components
  - light source
  - transmission medium
  - detector
- The basics
  - Light is fired into a tunnel of some material
  - The material has some critical angle
  - Fired light strikes the tunnel at some angle of incidence
  - If the angle of incidence is less than the critical angle, the light refracts back into the tunnel; otherwise it 'breaks out' of the tunnel
- *multimode fiber*: a fiber with multiple rays bounding around inside
  - each ray of light is said to be a mode
- As fibers get thinner, light necessarily wastes less time travelling along the fiber's radius (and thus ends up traversing the fiber faster)
- Single-mode fibers are harder to fabricate but are less lossy and can transmit further without repeaters
- Attenuation (in deciBels) = $10log_{10}\,(\frac{transmitted\ power}{received\ power})$
- 3 wavelength bands are used, centered at 0.85, 1.30, and 1.55 microns

  - ¤ The last two have very low attenuation
  - ¤ The first uses gallium arsenide for lasers and electronics (not clear to me why this is good)
  - ¤ All 3 bands are 25,000 - 30,000 GHz wide
- *chromatic dispersion*: light pulses spread out in length as they propagate
  - ¤ spread-out pulses may overlap
  - ¤ Naively, you can deduce the risk of this by reducing the signaling rate :(
  - ¤ Alternately, if you fire pulses in a special shape related to the hyperbolic cosine, the dispersion effects nearly cancel out
- BTW, these distinct pulses are called *solitons*
- The cable (inside to out)
  - ¤ glass core with very high index of refraction
    - ∗ where the light stays
    - ∗ 50 microns diameter for multimode, 8-10 for single-mode
  - ¤ glass cladding
    - ∗ lower index of refraction to try harder to bounce light back into the core
  - ¤ plastic jacket
    - ∗ to protect the cladding
  - ¤ Multiple fibers of the above, bundled into a single sheath
- Fiber usually 1m below ground, or sitting flat on the ocean floor
- How fibers are connected/spliced
  - ¤ connectors plugged into fiber sockets
    - ∗ connectors lose 10-20
    - ∗ this lets you reconfigure/use/alter the signal
  - ¤ Mechanical splicing
    - ∗ a splice junction can be used to boost signal
    - ∗  10
    - ∗ can take  5 minutes to do
  - ¤ *Fusion splice*: Two pieces get fused/melted together
    - ∗ Almost as good as a single fiber, only slight attenuation at splice point
- Light sources: LEDs & semiconductor lasers (Figure 2-8 compares these)
- On the receiving end, a photodiode
  - ¤ Typical response time of 1 nsec, thus limiting data to 1 Gbps
- Fiber networks
  - ¤ Fiber optic ring
    - ∗ Uses fiber point-to-point along a ring
    - ∗ Each point is a computer which can dispatch messages to/from the ring
    - ∗ Passive interfaces
      - > Signal to computer is duplexed from photodiode (fiber receiver)
      - > signal from is duplexed into transmitter
      - > So computer is like a pit stop – if it goes down, the ring network can continue
    - ∗ Active interfaces
      - > photodiode -¿ computer -¿ transmitter
      - > Computer interrupting the line is used to regenerate weakened signals
      - > But if the computer goes down, the ring is broken
  - ¤ Passive Star Topology
    - ∗ Each interface sends its signals to a *passive star* (a silica cylinder) which all other interfaces are also fused into
    - ∗ All interfaces' signals are diffused inside the passive star
    - ∗ Passive star has outgoing fibers to all interfaces
    - ∗ So the passive star in effect acheives broadcasting
    - ∗ Since incoming energy gets divided among all outgoing lines, number of nodes is limited by sensitivity of photodiodes
- Fiber vs Copper Wire

  - Fiber requires repeaters every 50km (instead of 5km)
  - Fiber handles higher bandwidths
  - Fiber unaffected by power surges/failures and EM interference
  - Fiber is thin and lightweight
    - Telcos like fiber because they can increase capacity without having to widen cable ducts
  - Fibers don't leak/radiate, so are harder to tap
  - Fiber is fragile
  - Fiber interfaces are still expensive
  - Fiber requires different/greater knowledge to deal with, because it's still new

## 2.3 Wireless Transmission

### 2.3.1 The Electromagnetic Spectrum

- frequency (Hz): oscillations per second
- wavelength (): distance between consecutive maxima (or minima)
- speed of light (c): $3x10^8 \frac{meters}{second} = 30 \frac{centimeters}{nanosecond}$
  - EM waves are about $\frac{2}{3}c$ in copper wire
- f = c
  - for  in meters and f in MHz, f    300
  - so 100MHz waves are 3m long, etc.
- Fig 2-11: EM spectrum split into bands (based on wavelengths) with official ITU names
- An EM wave's information carrying capacity is related to its bandwidth
  - Part of why fiber optics are so attractive
- $\lambda f = c \mapsto f = \frac{c}{\lambda} \mapsto \frac{df}{d\lambda} = \frac{-c}{\lambda^2} \mapsto \Delta f = \frac{c\Delta\lambda}{\lambda^2}$
  - so given the width of a wavelength band, , we can compute the corresponding frequency band f, and from that, the data rate the band is capable of producing
  - Most transmissions stick to a narrow frequency band to get better reception (many watts/Hz)
- Uses for a wide band
  - Frequency-hopping spread spectrum
    - Transmitter changes spectrum 100s of times/second
    - Used by military – hard to detect, harder to jam
    - Resistant to multipath fading
      - direct signals arrive fastest
      - by the time reflected signals have arrived you're no longer listening to that frequency
    - Also used commercially, for bluetooth and 802.11
    - Invented by 'Austrian-born sex goddess' Hedy Lamarrr
  - Direct Sequence Spread Spectrum
    - Spread signal over a wide spectrum
    - Used with 2G and 3G wireless, plus some wireless LANs

### 2.3.2 Radio Transmission

- Radio waves
  - Easy to generate
  - can generate long distances
  - can penetrate buildings easily
  - are omnidirectional
  - interference from electrical equipment
- Low frequencies: pass well through obstacles, can't travel far

- High frequencies: better for straight-line communication
  - they can bounce
  - they can even be absorbed by rain
- VLF, LF, MF bands
  - ”travel along the ground”
  - up to 1000km
  - low bandwidth
  - e.g., AM radio
- HF and VHF bands
  - absorbed by the earth
  - can refract off of ionosphere (height 100-500km) to travel quite far
  - used by ham operators and military

### 2.3.3 Microwave Transmission

- Above 100MHz, waves travel in straight lines
  - By concentrating beams via parabolic antennae, we get higher energy beams that can get better signal-to-noise ratio
  - But transmitter and receiver need to be aligned
  - Directionality allows arrays of transmitters and receivers to be set up
    - Earth’s curvature is an obvious limit, as is signal attenuation
  - Susceptible to multipath fading
- Influenced by weather (i.e. absorbed by water)
- Some operators keep 10
- Bands up to 10GHz now in routine use, but microwaves get absorbed by water (e.g. rain) at 4GHz
- Signal attenuation
  - Repeater distance goes up with ssquare root of tower height
  - For towers 100m tall, repeaters can be 80km apart
- Microwaves require no right of way (unlike cabling)
- It’s relatively cheap to install towers above the Earth
- Politics of the EM Spectrum
  - ITU-R’s WARC tries to coordinate spectrum allication so that devices can keep working across more and more of the world
  - Once some spectrum gets assigned to some use, which carriers get which frequencies has gone by 3 algorithms
    - Beauty contest
      - Procedure: Each carrier describes how it would best serve the public interest by getting a favorable slice
      - Risk: Creates all sorts of nasty conflicts of interest, is way too subjective
    - Lottery
      - Risk: Companies with no interest in using the spectrum can gobble up valuable territory and monopolize a secondary market for its legitimate uses
    - Auction
      - Risk: Has ended up with desperate telecoms burying themselves in insurmountable debt to avoid getting shut out of their own industry
    - Or, don’t allocate frequencies at all, but regulate the power used
    - Most goverments have set aside ISM (Industrial, Scientific, Medical) bands for unlicensed usage, e.g., garage door openers, toys, bluetooth, 802.11

### 2.3.4 Infrared Transmission

- Short-range, unguided: remote controls
- Relatively directional, cheap, easy to build
- Don’t pass through solid objects

- ¤ This scope limitation makes it ideal for per-room networks
  - ∗ security is high because signal doesn't leak out
  - ∗ no interference with the next room's network
  - ∗ no license needed to operate one

### 2.3.5 Light Transmission

- Old examples of optical signalling: Lighthouses, Paul Revere
- In decent weather, lasers on one roof pointed at a photodiode on another roof can be used to network 2 buildings

## 2.4 Communication Satellites

- Basically a big microwave repeater in the sky
- Contains several transponders
  - ¤ Each transponder listens to a portion of the spectrum, ampliesfies, and then rebroadcasts on a different frequency to avoid interference
  - ¤ The rebroadcast can cover a very broad area or a very narrow area
  - ¤ The satellite is said to be a *bent pipe* if the rebroadcast area is narrow
- Orbital period
  - ¤ By Kepler's Law, orbital period varies in proportion to the orbit to the 3/2 power.
  - ¤ Near Earth's surface, the period would be 90 minutes
  - ¤ At 35,800km, it's 24 hours. At 384,000km, about 1 month (i.e., the moon)
- Van Allen Belts: layers of highly charged particles held in their orbit by Earth's magnetic field

### 2.4.1 Geostationary Satellites

- *Geostationary Earth Orbit* (GEO) satellites
  - ¤ At 38,500km in a circular equatorial orbit, a satellite rotates as fast as the earth, and is thus geostationary (stationary in relation to Earth)
- For interference reasons, you want GEO satellites to be ¿= 2 degrees apart
- Orbit slot allocation is done by ITU
- Usually survive for 10 years before running out of fuel and beginning to drift hopelessly
- ITU also (too late, though) tries to allocate certain bands for satellite users
  - ¤ see Fig 2-16
- Earliest satellites
  - ¤ division of transponders into channels was static
  - ¤ Bandwidth was split up into fixed frequency bands (*frequency division multiplexing*)
- Modern satellites
  - ¤ transponders are divided into timeslots, with various users taking turns (*time division multiplexing*)
- First GEO satellites had a single beam that illuminated 1/3 of the Earth (*footprint* was 1/3 of Earth's surface)
- Nowadays, *spot beams* are elliptically sloped, as small as hundreds of km in diameter
- New: Very Small Aperture Terminals (*VSAT*)
  - ¤ small (1m instead of 10m) antennas
  - ¤ used for end-user satellite TV broadcast
  - ¤ VSAT systems get uplink by communicating with a nearby ground station/hub
  - ¤ ground station sends data to satellite via high-gain antenna or other connection
  - ¤ Typical path for data from 1 end-user to another might be VSAT1 -¿ hub -¿ satellite -¿ VSAT2
- GEO has high latency – usually 270msec
  - ¤ Coax, fiber optic, microwave can travel 50,000km in that time

- Satellites are distinctive in that broadcasting is not any more expensive for them
- Cost is independent of distance
- Rapid to deploy
- Low error rate

#### 2.4.2 Medium-Earth Orbit Satellites

- Drift slowly in longitude, circling the planet in  6 hours
- Not used for telecommunications
- e.g., the 24 satellites used for GPS

#### 2.4.3 Low-Earth Orbit Satellites

- Cover much less area because of lower orbit
- But also, require less transmission power, and incur lower latency
- Examples
  - ¤ Motorola's Iridium Project
    - ∗ Purpose: Provides telecom to places without infrastructure (e.g. on sea)
    - ∗ Satellite Distribution:
      - > Altitude 750km in circular orbits at both poles
      - > "Necklaces" at each pole, 1 satellite every  32 degrees
      - > Total of 66 satellites
    - ∗ Each satellite has a max of 48 cells/spot beams (total 1628 for all of Earth)
    - ∗ Each has 3840 channels
    - ∗ Communications between distant customers are relayed solely through space
  - ¤ Globalstar
    - ∗ 48 LEO satellites
    - ∗ Calls routed along the ground, using bent-pipe satellite connections only at the endpoints
  - ¤ Teledesic
    - ∗ Goal: Millions of concurrent internet users getting ¡= 100Mbps up/720Mbps down using small, fixed, VSAT-style antennas
    - ∗ "scheduled to go live in 2005"

#### 2.4.4 Satellites Versus Fiber

- Fiber seems amazing, but satellite still has some niche markets fiber does not or cannot address
  - ¤ Fiber not oriented to provide individuals with high bandwidth, but a VSAT-style solution can
  - ¤ Fiber not suitable for mobile (cellular+fiber may reduce the role of satellite here)
  - ¤ Situations where broadcast is essential
  - ¤ Hostile terrain/poorly developed infrastructure
  - ¤ Areas where right of way is prohibitively expensive for laying fiber
  - ¤ Rapid deployment

## 2.5 Digital Modulation and Multiplexing

### 2.5.1 Baseband Transmission

- *NRZ* (Non-Return-to-Zero): positive voltage is a 1, negative voltage is a 0
- line codes: other schemes to convert between bits and signals
- Bandwidth efficiency
  - ¤ *Baud/Symbol rate*: the rate at which the symbol changes
  - ¤ *Bitrate*: $symbol\ rate \times \frac{bits}{symbol}$

- Clock recovery
  - Accurate clocks are too expensive for $n$ bps links, the drift must be $< \frac{1sec}{nl\ bits}$ to detect runs of the same bit of length $l$, which is way too hard for megabit links
  - *Manchester Encoding*: send a signal that xors with the data bitstream a 0, 1, 0, 1, ... clock at twice the bitrate of the data. Requires twice as much bandwidth.
  - *NRZI* (Non-Return-to-Zero Inverted): encode 1s as a voltage transition, 0s as no transition (USB uses this). Long runs of 1s are now fine, but long runs of 0s are still not. Possible solutions:
    - *4B/5B* maps each possible 4-bit sequence to a 5-bit sequence with at least two 1s
    - Use a *scrambler* that generates a pseudorandom sequence to xor with your normal bitstream. While this makes no guarantees of removing long runs, it can help to condition the signal to be more 'white-noisy', so that there is less radiation of EM interference
- Balanced signals
  - *Balanced signal*: as much positive voltage as negative voltage even over short timespans, giving a lack of DC electrical component. Some channels attenuate DC heavily.
  - *Capacitive coupling*: Method of attaching receiver to a channel that passes only the AC component of the signal
  - Balanced signals help synchronize senders and receivers on timing and decision levels (the level at which a voltage is judged a 0 or a 1)
  - *Bipolar encoding/AMI* (Alternate Mark Inversion): Use two alternating voltage levels to encode 1 (say, 1V and -1V), and 0V to encode 0, to keep the average near 0.
  - *8B/10B line code* converts (5 bit, 3 bit) sequences to (6 bit, 4 bit) sequences, where the latter sequence can be one of two possibilities, as needed to balance out the first sequence.

### 2.5.2 Passband Transmission

- Take a *baseband* signal occupying 0 to $B$ Hz and shift it up to occupy a passband of $S$ to $S + B$ Hz
- *ASK* (Amplitude Shift Keying): 0 and 1 are represented by two different amplitudes
- *FSK* (Frequency Shift Keying): two or more tones convey information
- *PSK* (Phase Shift Keying): the carrier wave is shifted 0 or $n$ degrees every symbol period
  - *QPSK* (Quadrature Phase Shift Keying): Phase is one of { 45°, 135°, 225°, 315°} every symbol period, yielding 2 bits of information
  - *Constellation diagram*: shows possible combinations of amplitude and phase of a transmission method
  - *QAM* (Quadrature Amplitude Modulation), denoted QAM-$n$, refers to a scheme that transmits $n$ bits per symbol by varying amplitude and phase.
  - Since a small bit of noise may cause a misreading, the constellation should be interpreted as a *Gray code* (a mapping between symbols and bit vectors such that a single bit of error produces a symbol off by only one value).

### 2.5.3 Frequency Division Multiplexing

- *FDM* takes advantage of passband transmission to share a channel. Divide the available spectrum into frequency bands, and assign the bands so that only one user is permitted to use each band.
- In telephony
  - Each voice-grade channel occupies about 3100Hz (and is filtered down to such) of bandwidth, but is allocated 4000Hz. The excess 900Hz is a *guard band*, which reduces the likelihood and impact of adjacent channels overlapping
- *OFDM* (Orthogonal FDM) works for digital data
  - Many different subcarriers send data independently, packed tightly together in the frequency domain, but orthogonal in other domains (e.g., phase/amplitude, if used with QAM) so that at the center of each subcarrier's frequency response band, neighboring carriers cannot interfere
  - This requires a guard time, but at much lower overall cost than guard frequencies required by plain FDM
  - Usually used to split one high-rate stream into many lower-rate streams sending in parallel, which allows compensation if interference is high at a few frequencies.

### 2.5.4 Time Division Multiplexing

- Users take turns at occupying all the bandwidth (*time slots*), usually in a round-robin fashion
- May require a guard time to keep data streams in sync
- *STDM* (Statistical TDM) assigns time slots to match users' demand statistics, making it really just an alias for packet-switching.

### 2.5.5 Code Division Multiplexing

afdsasdfadsf

- *CDM* is often used for multiple access, hence it's often just called
- CDMA (*Code Division Multiple Access*)
  - standardized under IS-95
  - Is a particular form of *spread spectrum*, where a narrowband signal is spread out over a wider frequency
  - Each bit/time is divided into $m$ short intervals called *chips* (usually 64 or 128 chips)
  - *Chip sequences*
    - Each station gets a unique $m$-bit code called a chip sequence.
    - To transmit a 1-bit, station sends its chip sequence.
    - To transmit a 0-bit, station sends the one's complent of its chip sequence
  - To increase data rate from $b\frac{bits}{sec}$ to $mb\frac{chips}{sec}$, one needs to increase bandwidth by $m$. So CDMA is a kind of spread spectrum communication
  - All chip sequences are *Walsh codes*, meaning any pair of them is orthogonal (i.e., their dot product is 0)
  - When stations transmit simultaneously, their signals add linearly
  - To recover the bit stream of a distinct station, compute the normalized inner product of the received chip sequence and the chip sequence of the station you care about
    - This works because $(a_1 + a_2 + ... + a_n) \cdot C = 1$, for mutually orthogonal $a_1, a_2, ..., a_n$ iff $C \in a_1, a_2, ..., a_n$
  - This requires all chips to be synchronized in time.
    - Typically, CDMA uses a predefined chip sequence for synchronization between sender and receiver
  - Longer chip sequences improve tolerance to noise
  - Bit sequences sent with error-correction can also reduce the impact of such faults
  - The theory also assumes that base stations receiver signals of equal power (to make the signals correctly additive)
    - 1 base station $\leftrightarrow$ many mobile stations; power received at base station depends on distance of its mobile stations
    - Good heuristic: Each mobile station transmits to the base station at the inverse of the power level it last got from the base station
  - CDMA normally operates in a 1.25 MHz band

asdfadsfafdasfd

## 2.6 The Public Switched Telephone Network

### 2.6.1 Structure of the Telephone System

- 1876, telephone released. Lay your own P2P wire, resulted in lots of messy complete-graph wire topologies.
- By 1890, all local calls passed through a switching office
  - long distance calls required Bell to get the two switching offices connected
- Figure 2-21: 3 of hte 5 levels of hierarchy that were in use to connect a call
- Terms
  - *Local loop*: 2 cooper wires that connect a telephone to its end office
  - *End office/Local Central Office*: Directly connected to local telephones (even multiple lines in same house have independent connections to CO)

- ¤ *Tandem office/Toll office*
- ¤ *Toll connecting trunks*
- ¤ *Primary exchanges*, *Sectional exchanges*, *Regional exchanges*
- ¤ *Intertoll Trunks/Interoffice Trunks*

### 2.6.2 The Politics of Telephones

### 2.6.3 The Local Loop: Modems, ADSL, and Fiber

- End office: ¡= 10k local loops
- Modem: converts data from analog form to digital form, or vice versa
- 3 problems with transmission lines
  - ¤ Attenuation: loss of energy as the signal propages outward (in dB/km)
    - ∗ Varies by frequency; attentuation of a wave (if you think of it as a series of Fourier components) is quite uneven
  - ¤ Distortion: the result of the different Fourier components propagating at different speeds (*delay distortion*)
  - ¤ Noise: unwanted energy from sources other than the transmitter
    - ∗ Thermal noise: Unavoidable, caused by random motion of electrons in the wire
    - ∗ Crosstalk: Caused by inductive coupling between two wires close to one another
    - ∗ Impulse noise: e.g., spikes in the power line
- Modems
  - ¤ Since square waves require a wide spread of frequencies, which incurs both attenuation and delay distortion, baseband (DC) signaling is mostly not used
  - ¤ AC signaling with a *sine wave carrier* (continuous tone in 1000-2000Hz range)
    - ∗ used with *amplitude modulation*, different amplitudes will encode 0 vs 1
    - ∗ With *frequency modulation/frequency shift keying*, $\geq$ 2 tones are used
    - ∗ In *phase modulation*, the carrier wave is shifted 0 or 180 degrees at uniformly spaced intervals
      - > A better scheme is to use shifts of 45, 135, or 225 degrees every interval. It not only transmits 2 bits/interval, but also makes the boundaries of time intervals more recognizable
    - ∗ Modem (modulator-demodulator): Converts a serial stream of bits to a carrier wave modulated by ¿= 1 of these methods
  - ¤ An *n-baud* line transmits n symbols/second (n samples/second)
  - ¤ Each symbol can convey ¿= 1 bits, depending on modulation technique (amplitude, frequency, phase)
  - ¤ QPSK (Quadrature Phase Shift Keying): use 2 voltages and 4 phases to convey 2 bits/symbol
  - ¤ Alternatives to QPSK are Quadrature Amplitude Modulation (QAM), like QAM-16, QAM-64, ....
  - ¤ Constellation diagrams like Fig 2-25 show legal combinations of amplitude and phase for amodem, and modems can only talk if they both have the same such diagram
  - ¤ Higher-speed modems need more error-correction, use a *Trellis Coded Modulation*
    - ∗ V.32 modem standard transmits 4 data bits and 1 parity bit per symbol under QAM-32 at 2400 baud to get error-corrected 9600bps
    - ∗ V.32bis transmits 6 data bits and 1 parity bit per sample under QAM-128 at 2400baud, for 14,400bps. Faxes use this.
    - ∗ V.34 gets 28,800bps (12 data bits/symbol), V.34bis gets 33,600bps (14 data bits/symbol)
  - ¤ Modems may allow
    - ∗ *full duplex*: Simultaneous traffic in both directions
    - ∗ *half duplex*: Traffic in either direction, but only one at a time
    - ∗ *simplex*: Traffic only ever in one direction
  - ¤ 35kbps is the theoretical max for data from one local loop -¿ modem -¿ intermediate stuff -¿ local loop
    - ∗ Once source or destination is purely digital, the hard limit is up to 70kbps, on account of wire being slower than light
  - ¤ Telephone channel
    - ∗ 4000Hz wide; max samples/second is thus 8k
    - ∗ US had reserved 1 of the 8 bits per sample for control purposes
    - ∗ So the standard was created at 7 bits/sample x 800 samples/second = 56,000kbps

  - V.90: allows 33.6kbps up, 56kbps down
  - V.92
    - up to 48kbps up
    - faster negotiation of transmission speeds
    - capable of taking an internet interruption if the line has call-waiting

- Digital Subscriber Lines
- Digital Subscriber Lines
  - Voice-optimized local loops were attached at end office to a filter that attenuates ¡300Hz and ¿3400Hz
  - xDSL uses a different switch for local loops that lacks this filter
  - 4 major goals of xDSL
    - work over exisitng Cat 3 twisted pair local loops
    - must not affect existing telephones and fax machines
    - must be much faster than 56kbps
    - always one, metered per-month rather than per-minute
  - First xDSL offering
    - used FDM: divided the 1.1MHz into 3 bands: POTS, upstream, downstream
  - Discrete MultiTone (DMT)
    - divides the 1.1 MHz into 256 channels
    - each channel  4300Hz
    - Channel 0 for POTS
    - Channels 1-5 unused
    - 1 channel for upstream control, 1 for downstream control
    - 248 channels for data
  - Data usually allocated lopsided: 32 to upstream, 216 to downstream, hence *Asymmetric* in ADSL
  - ADSL (ANSI T1.413 and ITU G.992.1) in theory can offer 8Mbps down/1Mbps up, in practice usually less
  - Within each channel, similar to V.34, with a sampling rate of 4000 baud data sent with QAM, up to 15 bits/baud
  - Typical ADSL setup
    - Network Interface Device (NID) set up in customer's home
    - Near NID, a splitter, that splits POTS (0-4000Hz) off from data channels
      - A neat alternative
      - At End Office, POTS filtered out and sent to normal voice switch
    - Signal above 26kHz goes to a Digital Subscriber Line Access Multiplexer (DSLAM)
    - DSLAM recovers digital signal from modem into a bit stream
    - DSLAM sends bit stream off to the ISP
  - Instead of a splitter per home, could also use microfilters on each device
    - instead of 1 splitter at customer home, microfilters on each device
      - Phones would filter out data frequency ranges
      - NID would filter out POTS frequency range
    - Effects
      - Fewer technician visits
      - Slightly less reliable
    - NID is a digital signal processor that acts as 250 QAM modems operating in parallel at different frequencies
  - Usually ATM will run on top of xDSL

- Wireless Local Loops
  - Companies standing the best chance of competing with an *Incumbent Local Exchange Carrier* (ILEC) (former telephony monopolies) are long-distance telephone companies (IXCs: Interexchange Carriers)
  - To do so, they have to build an End Office, attract customers, and connect the customers. But stringing new wire is prohibitively expensive.
  - So these competitive LECs/CLECs use *Wireless Local Loop* (WLL)
    - Fixed Wireless: like a mobile phone, but 3 differences
      - High-speed connectivity is expected
      - Customer is okay with a CLEC technician pointing a directional antenna to the end office

      - > The user doesn't move, so connection handoff isn't a problem
    - ∗ Range of 50km, can penetrate rain & vegetation decently
    - ∗ Used for service called *Multichannel Multipoint Distribution Service* (MMDS)
      - > Tech for MMDS is readily available, but bandwidth available is small
    - ∗ *Local Multipoint Distribution Service* (LMDS)
      - > uses 1.3 GHz (@ 28-31 GHz in US, @ 40 GHz in Europe)
      - > relies on newer tech to operate so fast
      - > mm waves are highly directional, and range is only 2-5km, so many more towers are needed to cover an area
      - > Allocates 36 Gbps down/1 Gbps up total
      - > Requires clear line of sight, clear of foliage and rain, to work well
      - > LMDS is IEEE 802.16 standard (it is considered a wireless MAN)

### 2.6.4 Trunks and Multiplexing

- *Frequency Division Multiplexing*
  - ¤ Use FDM to mux 3 voice channels, each filtered to 3100 Hz
  - ¤ Allocate 4000 Hz to each channel
  - ¤ Raise 2nd and 3rd channels by 4000 Hz, 8000 Hz respectively
    - ∗ Between channels are guard bands, which are necessary because filters don't leave a sharp edge at the cutoff frequency
  - ¤ Typical FDM schemes
    - ∗ Twelve 4 kHz channels multiplexed into 60-108 kHz band (the whole unit called a *group*)
    - ∗ Many carriers offer a 48-56 kbps leased line service, based on this group
    - ∗ Five groups (60 voice channels) mux into a *supergroup.*
    - ∗ A *master group* is 5 supergroups (according to the CCITT standard) or 10 supergroups (the Bell system)
- *Wavelength Division Multiplexing*
  - ¤ For fiber optic channels
  - ¤ Multiple fibers can be joined for a length if their transmissions are on different wavelengths. Later on you can separate them out by wavelength.
  - ¤ This is used to multiplex many fibers together on a long-haul fiber
  - ¤ Uses an optical filter to select correct frequency after the joined signal is split. Optical filters are passive and reliable.
  - ¤ When the number of channels is very large and wavelengths are very close together (e.g., 0.1 nanometer), it's called *Dense WDM*
  - ¤ WDM works well because it's currenly impossible to convert between optical and electrical fast enough to deal with a few GHz
    - ∗ Running channels in parallel lets you use as many as 10 x 2500 GHz channels on a single fiber band
  - ¤ Optical amplification only needs to happen every 1000km, rather than incurring two opto-electrical conversions every 100km
- *Time Division Multiplexing*
  - ¤ Data from end offices gets combined to travel on outgoing trunks
  - ¤ *Pulse Code Modulation*
    - ∗ Analog signals digitized at end office by a *code* (coder-decoder)
    - ∗ produces a series of 8 bit numbers, in 8000 samples/sec (aka 125 µsec/sample)
    - ∗ this is enough to capture the 4 kHz telephone channel.
    - ∗ PCM is not consistent globally. North America and Japan use DS1 format / T1 carrier
      - > 24 voice channels multiplexed together
      - > Analog signals sampled round-robin by the codec which then produces a unified output stream
      - > Each channel inserts 8 bits into the stream, 7 data and 1 control, for 7 x 8000 = 56,000bps of data
    - ∗ A frame is 24 x 8 = 192 bits, + 1 used for framing, 1 frame sent every 125 µsec. The 193rd bit uses a predictable pattern like 010101... to let the receiver ensure it hasn't lost synchronization
    - ∗ When a T1 system is used entirely for data, the 24th bit gets used entirely for synchronization, to ensure faster recovery

- When CCITT finally decided to standardize PCM, this 1/8 control data approach was seen as too conservative. It provides two incompatible variations
  - *Common Channel Signaling*: The extra bit per frame takes on value 101010... in odd frames, and has signaling information for all channels in the even frames
  - *Channel-Associated Signaling*: Each channel has its own private signaling subchannel. For every 6 frames, on frame would devote one of its bits to signaling.
- CCITT also recommended a PCM carrier, *E1* at 2.048 Mbps
  - 23 8-bit samples into µsec frame
  - 30 channels for data, 2 for signaling
  - So, per pair of frames, 64 bits of signaling, control, and frame synchronization
- *Differential Pulse Code Modulation*: output amplitudes as deltas rather than as absolute values, since the actual values tend to change slowly. 5 bits suffice instead of 7
- In *delta modulation*, variant of DPCM, assume that amplitude changes by +1 or -1 from each preceding sample, which is only lossy if it changes quite quickly
- Alternatively, *predictive encoding*: Sender and receiver predict current amplitude from previous amplitudes using a shared algorithm, and encode only deltas from the predicted value
- *T2* is 4 T1 lines (4 x 1.544 Mbps = 6.176 Mbps), plus some additional for framing and recovery, for a total of 6.312 Mbps
- *T3* is 7 T2 lines, carrying a total of 44.736 Mbps
- *T4* is 6 T3 lines, carrying a total of 274.176 Mbps
- CCITT's standard calls for multiplexing 4 lines at every new level, rather than 4, 7, and 6, and calls for different framing and recovery as well

¤ SONET/SDH
- An attempt to standardize optical TDM systems in the early days of fiber
- *Synchronous Optical NETwork* (SONET). CCITT's related recommendations are called *Synchronous Digital Hierarchy* (SDH)
- Four major goals
  - Interoperability between different carriers (i.e., common signaling standards w.r.t. wavelength, timing, frame structure, ...)
  - Unify disparate US, European, & Japanese digital PCM systems
  - Multiplex digital channels to continue to speeds higher than T4
  - Provide support for *Operations, Administration, & Maintenance* (OAM) (huh?)
- Synchronous
  - Sender, receiver tied to a common metric clock, with accuracy of  1 part in $10^9$
  - Data sent at precise intervals of 125 µsec intervals, to match PCM, sent with crap if nothing needs sending
- Frames
  - A frame is a rectangle of bytes. 90 cols by 9 rows.
    - With 90 x 9 bytes x 8000 samples/sec, we have 51.84 Mbps
    - This is also called *Synchronous Transport Signal 1* (STS-1)
    - All SONET trunks are multiples of STS-1
  - First 3 columns of each frame are system management information
    - In the 1st 3 rows, these 3 columns are section overhead
    - In the other 6 rows, these 3 columsn are line overhead
  - First 2 bytes of each frame contain a fixed pattern
  - *Synchronous Payload Envelope* (SPE): the user data
    - Could start anywhere in the frame; the first row of line overhead will have a pointer to its start
    - First column of SPE is the header for the end-to-end path sublayer protocol (path overhead)
    - SPE can span two frames
- Optical carrier corresponding to STS-n is OC-n
  - When an OC-n is not multiplexed, but carries data from only a single source, we add a c to it (OC-nc)
  - So OC-3 designates a carrier consisting of 3 OC-1 carriers (summing to 155.52 Mbps)
  - But OC-3c designates a data stream from a single source at 155.52 Mbps
  - For OC-nc, columns from each stream are interleaved, resulting in frames that are 90c (huh? clarify) columns wide and 9 rows deep

### 2.6.5 Switching

- Circuit Switching
  - ¤ The switching equipment seeks out a physical path all the way from sender's phone to receiver's phone
    - ∗ Why fax is considered more secure: in theory, no misdelivery or onlookers on a dedicated circuit
  - ¤ In early days, the connection was made by the operator connecting input & output sockets with jumper cables
  - ¤ Invention of automated circuit switching by undertaker Almon Strowger
  - ¤ No data is sent while path is established (easily 10 seconds)
  - ¤ After call setup, propagation time (5 µsec/1000km) is the only delay
- Message Switching
  - ¤ *Store and forward* networks transmit messages only in their entirety (e.g., telegrams)
- Packet Switching
  - ¤ Meant to allow splitting of messages, because message switching required intermediate offices to sometimes store long blocks before forwarding
  - ¤ Also meant to prevent long messages from monopolizing lines for too long, so as to allow for interactive traffic
  - ¤ First packet can continue on before 2nd packet has arrived, improving throughput and reducing delay
  - ¤ Different packets may travel different paths, thus possibly out of order
  - ¤ Packet switching is more fault tolerant than circuit switching, since it assumes/requires the ability to route around obstacles like dead switches
  - ¤ Congestion may affect packet switching at time any packet is sent, whereas for circuit switching it may only impact at setup times
  - ¤ Packet switching wastes no bandwidth on low-traffic users
  - ¤ Basically, packet switching gives up a guarantee of service to reduce resource wastage
  - ¤ Store-and-forward of packet switching adds delay
  - ¤ Carrier decides basic parameters of packets (unlike over a circuit, where the signal is up to the sender and receiver only)
  - ¤ Billing/charging algorithm must change for packet switching
    - ∗ For packet switching, connect time is irrelevant
    - ∗ For packet switching, volume of traffic may be relevant
  - ¤ Fig 2-40 summarizes circuit vs packet switching along these kinds of parameters

## 2.7 The Mobile Telephone System

- Mobile phones have gone through 3 phases; analog voice → digital voice → digital voice and data
- For regulatory reasons, Europe stnadardized on GSM, but the US market has 2 major incompatible systems in place
- In US, mobile numbers are in the same area codes as landlines
  - ¤ The higher fees for a mobile call should only be borne by the caller if it is clear that the number they're calling is a mobile one.
  - ¤ This is why mobiles receiving a call are billed for it.
  - ¤ In Europe, mobile numbers have their own area codes, so calls to mobile phones are billed to the caller as you might expect
- Prepaid mobile phones also more prevalent in Europe (no monthly charge if rarely used, easy to replace, so quite a bit easier for kids to have)

### 2.7.1 First-Generation (coco1G) Mobile Phones: Analog Voice

- 1950s: radio-based push-to-talk systems as used by CB, taxis, police, use a high-powered transmitter on top of a tall building, using a single channel
- 1960s: *Improved Mobile Telephone System* (IMTS)
  - ¤ Similar, but with one sending frequency and one receiving frequency
  - ¤ So mobile users don't hear each others' outgoing messages
  - ¤ 23 channels from 150 MHz to 450 MHz

- ¤ Because transmitters are so powerful, systems need to be hundreds of km apart

- *Advanced Mobile Phone System* (AMPS)
  - ¤ Geographic regions divided into *cells*, usually 10-20km across
  - ¤ Cells use frequencies used by none of their neighbors
  - ¤ Capacity can be much higher if you just shrink the cell range
  - ¤ Smaller cells also allow smaller & cheaper handsets & transmitters
  - ¤ Cells are roughly circular, but get modeled as hexagons (!). E.g.,
    - ∗ 7 regular hexagons arranged tightly form a roughly hexagonal shape
    - ∗ At the center of each of these are 3 mutually adjacent hexagons that form a roughly triangular shape
    - ∗ This sort of clustering gives each cell a buffer 2-cells wide where its frequency is not reused
  - ¤ Center of each cell has a base station to which all telephones transmit
  - ¤ Base stations connect to one or more *Mobile Telephone Switching Office*s (MTSOs) or *Mobile Switching Center*s (MSCs), which are basically just end offices that communicate with each other and the PSTN via a packet-switched network
  - ¤ A base station has ownership of every phone in its cell and are in control of handoff
  - ¤ *Handoff* in the big picture
    - ∗ When one phone's signal fades, it asks neighboring base stations to take ownership of the phone
    - ∗ That new base station initiates handoff with the phone directly
    - ∗ The phone switches frequency to match its new cell
    - ∗ handoff takes about 300 µsec.
    - ∗ *Soft handoff*: phone sends/receives on both frequencies simultaneously during handoff, so there's no loss of continuity. 1G and 2G devices can't do this.
    - ∗ *Hard handoff*: Old base station drops phone before new station acquires it. If the new base station can't acquire it, the call is dropped.
  - ¤ MTSOs own how channels are assigned to base stations
  - ¤ AMPS channels
    - ∗ 832 simplex tx channels from 824-849 MHz, 832 simplex rx channels from 869-894 MHz
    - ∗ Because of 800 MHz signals' properties (radio waves that travel straight, can get absorbed by plant life, and bounce off of hard surfaces), echos and multipath fading are possible
    - ∗ The 832 channels are divided like so:
      - > Control (base to mobile) to manage the system
        - ○ 21 channels reserved for this, and they're implemented on phones in hardware (PROMs)
      - > Paging (based to mobile) for incoming call notifications
      - > Access (bidirectional) for call setup & channel assignment
      - > Data (bidirectional) for voice/fax/data payloads
  - ¤ Call management
    - ∗ Each phone has a 32-bit serial number and a 10-digit phone number in its PROM. Phone num is a 3 digit area code in 10 bits, 7 digit subscriber number in 24 bits.
    - ∗ When switched on, phone sscans a preprogrammed list of 21 control channels to find strongest signal, then broadcasts the serial number & phone number digitally, multiple times, with error-correction codes
    - ∗ Base station registers the phone with its own MTSO and with phone's home MTSO
    - ∗ Starting a call
      - > phone transmits callee's number and identity on access channel
      - > Base station informs its MTSO.
      - > If base station's MTSO is okay with servicing this call, it assigns an idle channel and sends it back on the control channel
    - ∗ Receiving a call
      - > Callee's home MTSO has registered the callee's current base station
      - > Caller requests this info from callee's home MTSO
      - > Caller sends a packet to the callee's current base station
      - > Callee's base station broadcasts an "Are you there?" within its cell
      - > Callee responds on the access channel
      - > Callee's base station allocates a channel and informs the callee

### 2.7.2 Second-Generation (2G) Mobile Phones: Digital Voice

- D-AMPS, GSM, CDMA, PDC
- PDC is basically jsut D-AMPS with some Japan-specific backwards-compatibility
- D-AMPS (*Digital Advanced Mobile Phone System*
  - IS-54 and IS-136 standards
  - Designed so 1G phones are compatible with it
  - Includes extra bandwidth (1850-1910 MHz for upstream and 1930-1990 MHz for downstream) to handle extra load. Waves for these frequencies are shorter, so handset antennas can be smaller
  - TDM on voice channels
    - On handset, voice gets digitized, then compressed by a vocoder, to decrease data use in both directions. This lets 3 voice users share a single frequency pair
    - Each frequency pair supporst 25 frames/sec of 40 msec each, with frames divided into six time slots of 6.67 msec each
    - Each slot is 324 bits long (260 are user payload)
      - 64 bits for guard times, synchronization, and control
      - 159 bits for compressed speech
      - 101 bits of error-correction
  - Superframes
    - 1 superframe made up of 16 frames
    - certain control information is repeated only a few times per superframe (rather than once per frame)
    - 6 main control channels: system configuration, real-time control, non-real-time control, paging, access response, short messages
  - *Mobile Assisted Hand Off* (MAHO)
    - 1/3 of the time, mobile is neither sending nor recieving. It uses this time to check line quality and complain about weak signals, so that handoff is handled proactively rather than last-minute.
- GSM (*Global System for Mobile Communications*)
  - Dominant everywhere but US and Japan (the US also uses it, but it's not dominant)
  - Quite similar to D-AMPS, but
    - frequency difference between upstream and downstream is lower (55 MHz higher instead of 80 MHz)
    - GSM channels are wider (200 kHz instead of 30 kHz) and hold fewer additional users (giving GSM a higher data rate)
  - 124 pairs of simplex channels, each 200 kHz wide and capable of supporting 8 connections via TDM
  - Each active station gets 1 timeslot on one channel pair
  - Theoretical capacity of 992 channels per cell, but many are not available (to prevent frequency conflicts with neighboring cells)
  - Data frame / slot
    - Three 0-bits (frame delineator) at start, 3 more at end
    - Two fields of 57 bits apiece holding information payload (each associated also with 1 bit indicating whether it's voice or data)
    - 26 bits used by receiver to maintain synchronization with sender
    - $Total = 3 + 3 + 57 + 1 + 57 + 1 + 26 = 148\ bits$
    - Occupies the channel for 577 µsec (which includes 30 µsec of guard time
  - Each TDM frame can hold 8 slots, hence total duration of a TDM frame is 4.615 msec
  - Gross rate of each channel is 270,833 bps for each of 8 users, or 33.854 kbps
  - Overhead leaves only 24.7 kbps per user before error-correction. After error-correction, there's only 13 kbps for speech
  - 26 TDM frames combine to form a 120 msec multi-frame. In a multiframe, slot 12 is used for control, slot 25 is reserved for future use
  - 51 slot multi-frames are also sometimes used, in which case several of the slots are used for specific control channels
    - Broadcast control channel: continuous stream of output from base station with its identity and channel status
    - Dedicated control channel: location updates, registration, and call setup. Used in general to help a base station maintain its DB of handsets in its jurisdiction
    - Common control channel: contains 3 logical subchannels

    > Paging channel: base station uses this to announce incoming calls. Each handset monitors it continuously
    > Random access channel: Allows users to request a slot on the dedicated control channels (which would let them initiate a call)
    > Access grant channel: When base station hsa assigned a slot on dedicated control channel for a handset, the assignment is announced here

- CDMA (*Code Division Multiple Access*)
  - ¤ standardized under IS-95
  - ¤ Each bit/time is divided into m short intervals called *chips* (usually 64 or 128 chips)
  - ¤ *Chip sequences*
    - ∗ Each station gets a unique m-bit code called a chip sequence.
    - ∗ To transmit a 1-bit, station sends its chip sequence.
    - ∗ To transmit a 0-bit, station sends the one's complent of its chip sequence
  - ¤ To increase data rate from b bits/sec to mb chips/sec, one needs to incrase bandwidth by m. So CDMA is a kind of spread spectrum communication
  - ¤ All chip sequences are *Walsh codes*, meaning any pair of them is orthogonal (i.e., their dot product is 0)
  - ¤ When stations transmit simultaneously, their signals add linearly
  - ¤ To recover the bit stream of a distinct station, compute the normalized inner product of the received chip sequence and the chip sequence of the station you care about
    - ∗ This works because $(a_1 + a_2 + ... + a_n) \cdot C = 1$, for mutually orthogonal $a_1, a_2, ..., a_n$ iff $C \in a_1, a_2, ..., a_n$
  - ¤ This requires all chips to be synchronized in time.
    - ∗ Typically, CDMA uses a predefined chip sequenc for synchronization between sender and receiver
  - ¤ Longer chip sequences improve tolerance to noise
  - ¤ Bit sequences sent with error-correction can also reduce the impact of such faults
  - ¤ The theory also assumes that base stations receiver signals of equal power (to make the signals correctly additive)
    - ∗ 1 base station ↔ many mobile stations; power received at base station depends on distance of its mobile stations
    - ∗ Good heuristic: Each mobile station transmits to the base station at the inverse of the power level it last got from the base station
  - ¤ CMD normally operates in a 1.25 MHz band

#### 2.7.3 Third-Generation (3G) Mobile Phones: Digital Voice and Data

- There's a notion of IMT-2000 (International Mobile Telecommunications)
- Some main proposals
  - ¤ W-CDMA (Wideband CDMA) / UMTS (Universal Mobile Telecommunications System)
    - ∗ Direct Sequence Spread Spectrum with a 5 MHz band
    - ∗ Not fully GSM backward-compatible, but GSM base stations would be able to continue a call started on a W-CDMA if mobile station moves into a GSM cell
  - ¤ CDMA 2000
    - ∗ No compatibility with GSM
- In the meantime...
  - ¤ EDGE (Enhanced Datarates for GSM Evolution): GSM with more bits per baud
  - ¤ GPRS (General Packet Radio Service): An overlay packet network on top of D-AMPS or GSM
    - ∗ timeslots divided into several logical channels, used for various purposes
    - ∗ base station tries to find idle timeslots in voice channel to give them over to data

### 2.8 Cable Television

#### 2.8.1 Community Antenna Television

- Cable TV was conceived to improve reception for remote areas
- The basics

  - Antenna on a hill receives signals
  - *head end* amplifies signal
  - Signal gets sent through a single coaxial cable
  - Coaxial cables splits off into individual subscribers' homes
- Anyone could set up one of these services

### 2.8.2 Internet over Cable

- *Hybrid Fiber Coax* (HFC) systems
  - Cabling between cities got replaced with fiber
  - *Fiber nodes* translated optical signal from fiber to electrical for coax, and vice versa
- When cable providers wanted to get into telephony and internet, the serial/broadcast network topology became a problem
  - One user could saturate the network
  - Users can read each others' traffic on a strictly broadcast downstream network
  - Communication must be re-engineered to allow for upstream communication

### 2.8.3 Spectrum Allocation

- Asymmetric allocation is typical
  - Upstream data usually 5-42 MHz, downstream 550-750 MHz
  - Not a big deal, as telephone companies' service is also up/down asymmetric "even though they have no technical reason for doing so"
- Usually use QAM-64 or QAM-256 on each 6-8 MHz channel, giving a net payload of 27-39 Mbps
- Upstream tends to use QPSK to reduce the effect of various types of noise, yielding only 2 bits per baud instead of the 6-8 that QAM gets
- Since the head end now has to talk to an ISP, it needs to be a more powerful computer, and now has a fancy name: *Cable Modem Termination System*

### 2.8.4 Cable Modems

- Motivations for standardizing cable modems
  - Reduces service provider/consumer lockdown on specific hardware vendors
    - which encourages price competition among vendors
    - which ultimately results in a lower price tag for the service
    - which provides a lower barrier to entry for consumers to adopt the service
  - Reduces the number of truck rolls needed to set up unusual hardware (especially if standardization encourages consumers to own their own modem)
- DOCSIS (Data Over Cable Service Interface Specification) and EuroDOCSIS
- When the modem is turned on
  - modem listens for a special kind of packet the head end sends periodically that has system parameters
  - modem announces its presence on one of the upstream channels
  - head end assigns the modem its upstream and downstream channels
  - Modem goes *ranging* to determine its distance from head end
    - It does this by sending a special packet and listening for a response
    - Knowing the distance lets the system coordinate timing on the upstream
  - Upstream is divided in time into *minislots*. Packets upstream must fit into one or more consecutive minislots
  - Head end periodically starts a new round of minislots, but these starts aren't heard simultaneously because of propagation delay
  - On modem initialization, head end also assigns it a minislot ot use for requesting upstream bandwidth
    - Often several modems have the same minislot assigned for this purpose

    - In the case multiple modems make such a request at the same time, their requests collide and the head end doesn't respond. They keep trying, using binary exponential backoff to reduce the chance of collisions
  - Downstream is different
    - Time Division Statistical Multiplexing
    - Traffic is much larger: Fixed packet size of 204 bytes, containing 20 bytes of overhead that includes a Reed-Solomon error-correcting code. 204 bytes was chosen to be compatible with MPEG-2
  - Initialization also
    - gets an IP address from the ISP using DHCP
    - gets accurate time of day from the head end
    - uses Diffie-Hellman to exchange private keys with the head end (to avoid snooping from others on the same coax cable)
  - Then modem sends its unique identifier over the secure channel to log in

### 2.8.5 ADSL Versus Cable

- Cable bandwidth is hard to predict because cable segments are shared
- Availability
  - Some spots are far away from an end office
  - Businesses tend not to have cable
- ADSL is P2P cabling, which is inherently more secure

## 2.9 Summary

# 3 Data Link Layer

- The main assumption here is a 'wirelike' channel–basically where bits arrive in the order sent
- Problems at this layer
  - error detection/handling
  - propagation delay

## 3.1 Data Link Layer Design Issues

- Major goals
  - Provide well-defined service interface to network layer
  - Deal with transmission errors
  - Regulate flow of data
- Data put in to frames having header, payload, and trailer

### 3.1.1 Services Provided to the Network Layer

- Connectionless Service
  - Unacknowledged, Connectionless Service
    - Frames do not receive any acknowledgement
    - Appropriate when actual error rate is low enough for higher layers to deal with
    - Also useful for realtime applications, where late data is worse than bad data
  - Acknowledged, Connectionless Service
    - Frames are acknowledged when received
    - A sender not getting an acknowledgement will re-send
  - Frame-level acknowledgements are only an optimization

    - Since a data payload can be split into many frames, it's relatively easy to request missing frames in the data link layer rather than requesting the entire payload in the network layer.
    - Frame-level acknowledgement is not useful for unreliable channels

- Connection-oriented Service
  - Frames sent are numbered
  - Data link layer guarantees each frame is received exactly once and in the right order
  - Data transfer is in 3 phases: establish connection, send frames, release connection

### 3.1.2 Framing

- Framing methods: howto split a byte stream into frames
  - Character count
    - In frame header, send a byte with count of characters belonging to this frame
    - Problem: garbled count will desynchronize frame boundaries between sender and receiver
  - Flag bytes with byte stuffing
    - *flag bytes*: special bytes that each frame starts and ends with
      - Makes it easy for the receiver to resynchronize
      - But binary data may contain the flag byte
        - *Byte stuffing*/character stuffing
          - The occurrence of the flag byte in the payload can be escaped with a special byte known as an escape character
          - But now occurrences of the escape character must also be escaped where it occurs in the payload
          - Byte stuffing can cause problems in the case of wide characters (where byte and character are not synonymous)
        - *Bit stuffing*
          - Every instance of five consecutive 1's in the payload gets a 0 inserted after it
          - Use flag byte 01111110
          - Receiver synchronizes frames by looking for the flag byte
          - Once receiver has clearly-delineated frames, strips out all 0's occurring after a sequence of five 1's

### 3.1.3 Error Control

- Typical for a receiver to send control frames with positive acknowledgements about received frames that are well formed, or negative acknowledgements about frames that have been received with errors
- When receiver receives no part of a frame, it can send no acknowledgement
  - So the data link layer keeeps a timer. After the timer for a frame expires, it re-sends it
  - But if the first copy of a frame was sent and received (and the only thing lost was the positive acknowledgement), you wouldn't want duplicate copies of the frame
  - So we also number the frames sequentially to permit collision detection

### 3.1.4 Flow Control

- Feedback-based flow control
  - Receiver occasionally gives sender permission to send a certain amount of data, sender obeys
- Rate-based flow control
  - Fixed maximum data speed
  - Not really used in the data link layer

## 3.2 Error Detection and Correction

- *Error-Correcting Codes* (ECCs): include enough redundant information that the receiver can deduce what the intended payload actually was. Also known as forward error correction.
- *Error-Detecting Codes* (EDCs): include just enough extra information to deduce that an error happened. Most useful on channels with few errors.

### 3.2.1 Error-Correcting Codes

- A frame normally has $m$ data bits and $r$ *check bits* (redundant bits), with $n = m + r$ bits total, or an n-bit *codeword*
- *Hamming Distance*: Given any two codewords A and B, count the 1 bits in A xor B
- All $2^m$ messages are used, but because of special ways check bits get computed, rarely are all $2^n$ codewords used
- *Hamming Code*: Once you know the list of legal codewords, you can figure out the pair of these with the minimum Hamming Distance to get the Hamming Distance of the code.
- To detect $d$ errors, you need a code with a Hamming Distance $d + 1$
  - ¤ Receivers getting an invalid codeword then know that the transmission error has received between 1 and $d$ errors, and will request a re-send
- To correct $d$ errors, you need a code with Hamming Distance $2d + 1$
  - ¤ This way the legal codewords are so far apart that even with $d$ changes, the received codeword is closer to the sent, legal codeword than to any other legal codeword.
- For $m$ message bits and $r$ check bits, attempting to correct any single error, we have
  - ¤ Each of the $2^m$ legal messages has $n$ illegal codewords at a Hamming Distance of 1 from it
  - ¤ So, in some sense, each of the $2^m$ legal messages occupies $n + 1$ bit patterns
  - ¤ So we have $(n + 1)2^m \leq 2^n$.
  - ¤ Because $n = m + r$, we have $n + 1 \leq 2^r$
  - ¤ So $m + 1 \leq 2^r$
  - ¤ so, given $m$, this gives a lower limit on the number of check bits needed
- *Hamming Codes*
  - ¤ 1st, 2nd, 4th, 8th, ... its are all used as check bits, the rest are data
  - ¤ The $k$'th bit is parity checked by the bits whose summed positions totals to $k$
  - ¤ Receivers can reconstruct a single incorrect bit by summing the positions of the parity bits which are out of parity with the bits they govern
  - ¤ Hamming codes can only correct 1 error per codeword
  - ¤ If your channel is prone to burst errors, use a different approach
    - ∗ Calculate $k$ Hamming Code encoded codeworrds.
    - ∗ Send the 1st bit of all 1..k codewords, then the 2nd bit of each.
    - ∗ In this way, a burst of up to length $k$ per every $kn$ bits can be tolerated, as each codeword will have only suffered 1 bit of error

### 3.2.2 Error-Detecting Codes

- Why use EDC on reliable channels when ECC exists and is so awesome?
  - ¤ Suppose error rate = $10^{-6}$ per bit, block size of 1000 bits
  - ¤ Requires 10 check bits/block for ECC, vs as little as 1 parity bit per block for EDC
  - ¤ Once every 1000 blocks, an extra block will need to be retransmitted with EDC (vs recovered without retransmit with ECC)
  - ¤ Total overhead would be 2001 bits for EDC, vs 10,000 bits for a Hamming code ECC
- *Burst error*: a sequence of bits such that the first and last bits are in error and there exists no contiguous subsequence of m correctly received symbols within the error burst.
- Method 1: Single parity bit per block
  - ¤ Chance of detection for burst errors is only 0.5
- Method 2: Multiple parity bits

- ¤ Regard block as an n column x k row matrix
- ¤ Compute a parity bit for each column, appending these parity bits as a new row
- ¤ Transmit matrix row-by-row
- ¤ Can detect a single burst of length n, but a burst of length n+1 where only the first and last bits are changed would go undetected, as both errored bits would be in the same column.
- ¤ In the case of very long burst, or many small bursts, the probability that any of the n columns will have the right parity by accident is 0.5
- ¤ So the probability of an error being undetected is $2^{-n}$

- Method 3: *Polynomial code / Cyclic Redundancy Check* (CRC)
  - ¤ How
    - ∗ An m-bit frame is regarded as the coefficient list for a polynomial $M(x)$ with terms $\{x^{n-1}, x^{n-2}, ..., x^0\}$
    - ∗ Sender and receiver agree on a *generator polynomial* $G(x)$ in advance that has highest and lowest bits = 1, and has $r < k$ terms.
    - ∗ Append r 0 bits to end of frame so it now has $m + r$ bits and corresponds to $x^r M(x)$
    - ∗ Calculate $x^r\ M(x)\ mod\ G(x)$ using modulo 2 division (basically, long division where the subtraction is done with xor)
    - ∗ Subtract the $\leq r$ bit remainder from $x^r M(x)$. This value, $T(x) = x^r\ M(x) - (x^r\ M(x)\ mod\ G(x))$, is the checksummed frame to be transmitted
    - ∗ Clearly $T(x)\ mod\ G(x) = 0$
  - ¤ Analysis
    - ∗ The transmission received is characterized as $T(x) + E(x)$, where $E(x)$ is the error (if 0, then there was no transmission error)
    - ∗ The error will be noticed if $(T(x) + E(x))\ mod\ G(x) \neq 0$ (i.e., if $E(x)\ mod\ G(x) \neq 0$).
    - ∗ One single-bit error
      - > i.e., where $E(x) = x^i$, for i determining which bit is in error
      - > $G(x)$ with two terms won't divide it, so these are always detected
    - ∗ 2 isolated single-bit errors
      - > i.e., for $i < j$, $E(x) = x^i + x^j = x^j(x^{i-j} + 1)$.
      - > As long as $G(x)\ mod\ x \neq 0$ and $G(x)\ mod\ x^k + 1 \neq 0$ for all $k \leq i - j$, these are detected
      - > Low-degree polynomials that detect errors well for long frames are known, e.g., $x^{15} + X^{14} + X^0$ won't divide any $x + 1$ for any $k < 32,768$
    - ∗ An odd number of bits in error
      - > Would have $E(x)$ with an odd number of terms
      - > No odd-termed polynomial has $x + 1$ as a factor in modulo 2
      - > So if $x + 1$ is a factor of $G(x)$, we can catch all errors of this type (so I guess we can't combine the above resistance against 2 single-bit errors ???)
    - ∗ Burst errors (assuming r check bits)
      - > error length up to $r$
        - ◦ Will be detected for sure
        - ◦ A burst error of length k is $E(x) = x^i(x^{k-1} + x^{k-2} + ... + 1)$, for i being the position where the error started (counted from the last bit in the frame)
        - ◦ $G(x)$ with an $x^0$ term doesn't have $x^i$ as a factor
        - ◦ So $E(x)\ mod\ G(x) \neq 0$ whenever the degree of $G(x)$ is greater than the degree of $(x^{k-1} + x^{k-2} + ... + 1)$
      - > error length of $r + 1$
        - ◦ $E(x)\ mod\ G(x) = 0$ only if $E(x) = G(x)$
        - ◦ By definition of a burst, first and last bits must be 1, so this depends on the $r - 1$ intermediate bits
        - ◦ If all bit combinations are equally likely, then the probability of this type of error going undetected is thus $(\frac{1}{2})^{r-1}$
      - > error length greater than $r + 1$
        - ◦ Probability of one of these burst errors going undetected is similar, $\frac{1}{2}^r$.
  - ¤ As the above indicates, $G(x)$ having or not having certain factors provides varying protection against different sorts of errors. Thus, a few specific $G(x)$ have become standards.
  - ¤ In practice, CRCs are easily dealt with by specialized hardware

- Analyses of frame contents given above assume that every bit has a 50% chance of being 1 or 0, which turns out to not be true. So the actual effectiveness at preventing error

## 3.3 Elementary Data Link Protocols

- Assumptions/simplifications to start with
  - Physical, Data Link, and Network layers are separate processes
    - In reality, Physical and Data Link layers are both in hardware
  - Unidirectional data transfer
  - Infinite supply of data needs to be transferred
  - Machines never crash
- Abstract interface for these protocols that assumes some common elements
  - to_physical_layer procedure that sends a frame over the wire
  - from_physical_layer procedure that pulls a frame off the wire
  - wait_for_event(event_type) procedure that only returns when there's a frame on the physical layer waiting to be picked up
    - Usually event handling is done by interrupts rather than an event listener sitting in a tight loop
  - Acknowledgement timers that can also trigger an interrupt to re-send unacknowledged frames
  - Procedures to signal network layer to stop/restart sending packets (to avoid being overwhelmed)
  - Sequence numbers that increment circularly

### 3.3.1 A Utopian Simplex Protocol

- A utopian sample protocol in which
  - Data travels in one direction
  - Networks layers are always ready on sender and receiver
    - Processing time of receiving network layer can be ignored
    - or, infinite buffer space is available
  - Frame loss and frame damage don't happen

### 3.3.2 A Simplex Stop-and-Wait Protocol for an Error-Free Channel

- Drops the assumption that receiving network layer has negligible processing time
- This means if the receiver requires time $Dt$ to execute from_physical_layer and to_network_layer, then we can send at a rate no faster than $\frac{1\,frame}{Dt}$
- Especially so if the receiver has no buffering, because then we run the risk of overwriting a frame being processed
- *Stop-and-wait* protocols are those in which the sender sends one frame, then sends no others until receiving an acknowledgement
- Requires a half-duplex or full duplex physical channel

### 3.3.3 A Simplex Stop-and-Wait Protocol for a Noisy Channel

- When frames can be lost/damaged, you can basically just add a resend-if-unacknowledged timer to the sender
  - This can break down if an acknowledgement is lost, because you don't want the receiving network layer to get duplicate packets from the receiving data link layer
  - So we use sequence numbers to let the receiving data link layer identify and discard duplicate data
  - Because it's still a stop-and-wait protocol, the receiver is only expectin ga resend of a frame it never received, or a resend of a frame it's received and acknowledged, or the next frame after the last one it acknowledged
  - So a 1-bit sequence number suffices
- Protocols in which the sender waits for a positive acknowledgement before continuing can be called *PAR* (Positive Acknowledgement with Retransmission) or *ARQ* (Automatic Repeat reQuest)

## 3.4 Sliding Window Protocols

- Objective: permit bidirectional transmission
- It's possible, of course, to allocate one forward channel for data, and a reverse channel for acknowledgements, with two such circuits for each machine pair, creating a severely underused reverse channel
- Or, to intermix traffic, *piggybacking*: When a frame A is received by host X from host Y, it waits a short duration to see if it can include its acknowledgement of A on a data frame it would already be sending to host Y
- *Sliding Window* Protocol
  - Sender maintains a list of sequence numbers it's permitted to send, called its *sending window*
    - Sending window represents unacknowledged sequence numbers (including sequence numbers not yet or not recently sent)
    - Acknowledgements increment the lower limit, packets from the network layer increment its upper limit
    - Sender may have to re-send any within its window, so it must also store the frames in a buffer
  - Receiver maintains a list of sequence numbers it's permitted to accept, called its *receiving window*
    - Discards all frames not within its window
    - Incoming frames equal to its lower limit get acknowledged and passed up to the network layer
    - Reciving window is always fixed size; if 1, this means it only accepts frames in order and only supports stop-and-wait
  - The sending and receiving windows needn't have same size, same upper limits, or same lower limits
  - In some sliding window protocols, window sizes are dynamic

### 3.4.1 A One-Bit Sliding Window Protocol

- Premature timeouts or simultaneous starts can cause strange situations where data received correctly would be re-sent multiple times

### 3.4.2 A Protocol Using Go-Back-N

- Blocking on acknowledgements foreces a lot of time to be idle
- Let the sender transmit up to w frames before blocking
- *Pipelining*
  - If channel capacity is b bits/second, frame size is l bits, and roundtrip propagation delay is R seconds
  - Time to transmit 1 frame is $\frac{l}{b}$ seconds
  - After that frame is sent, it is $\frac{R}{2}$ until that frame arrives at receiver, and it will be $\geq \frac{R}{2}$ more before an acknowledgement arrives
  - In stop-and-wait, the line is busy for $\frac{l}{b}$ and idle for $R$, giving line utilization $= l(l + bR)$
- Pipelining over an unreliable channel creates problems
- *Go Back N*
  - Receiver discards frames with errors, and all frames with higher sequence numbers.
  - This corresponds with a window size of 1
  - This can waste a lot of bandwidth
- *Selective Repeat*
  - Receiver buffers frames that arrive and acknowledges them, throwing away only errored frames
  - Sometimes also uses negative acknowledgements so that errored frames, or frames out of sequence, trigger a re-transmit before the sender's timers for those frames does
  - This requires buffer space to allow for the larger receiving window
  - This wastes less bandwidth
- When pipelining, the number of sequence numbers should be one larger than the window size
  - This is to avoid contiguous batches of sending-window frames to end on the same sequence number, the sender would send acknowledgements with that sequence number whether the entire batch had been received correctly, or whether none of it had been received at all.
- A sender receiving an acknowledgement for frame $n$ can be sure that all frames $n-1$, $n-2$, ... have also been received (a nice feature that helps advance sending window in case acknowledgements get lost)

- Mulitple outstanding frames demand multiple timers, easily managed by a linked list queue, each node having
  - clock ticks until expiry (as counted from the expiry time of the previous node)
  - the frame being timed
  - a pointer to the next node

### 3.4.3 A Protocol Using Selective Repeat

- Nonsequential receive poses problems
  - If after a successful receipt, the receiver sends a batch of acknowledgements that never make it to the sender, sender will resend original frames, which receiver would pass up to the network layer as duplicate data
  - The fix is to ensure that the maximum window size is $\leq \frac{1}{2}$ the range of sequence numbers
- On the reciever, the number of buffers and timers needed is the window size
- This sample protocol also removes the assumption of heavy symmetric traffic
  - Heavy symmetric traffic guarantees that piggybacking doesn't block the channel unnecessarily
  - How: When an in-sequence frame arrives, an ack timer is started. If this timer goes off, an acknowledgement is sent on its own. The timer is cancelled if an acknowledgement is sent piggybacked, and is set to a full interval if a new, in-sequence frame is received. Note: The ack timer must be set to a shorter interval than the sender's retransmit interval
- Negative acknowledgements (NAKs)
  - NAKs sent for damaged frames or frames other than the expected one
  - Receiver remembers whether it's sent a NAK for the frame it's expecting
  - When a NAK is sent, a timer is started. If the timer goes off, the receiver sends out an acknowledgement to synchronize the sender to its current status
- Sporadic roundtrip times require NAKS and also requires timers to be long, both of which cause wasted bandwidth
- When the roundtrip times have a low standard deviation compared to the time itself, timers can be short, NAKs rarely come into play, and bandwidth utilization is high

## 3.5 Example Data Link Protocols

### 3.5.1 Packet over SONET

- SONET is a physical layer protocol providing a bitstream that, whether there's data to send or not, delivers fixed-size byte payloads
- PPP runs on IP routers to provide framing
- PPP improves on the older *SLIP* (Serial Line Internet Protocol), and provides:
  - unambiguous delineation of frame beginnings and ends, as well as error detection
  - *LCP* (Link Control Protocol) to bring lines up, test them, negotiate options, and shut them down gracefully
  - A way to negotiate network layer options that's independent of which network layer is used, by implementing an *NCP* (Network Control Protocol) for each network layer supported.
- PPP frame format resembles that of *HDLC* (High-Level Data Link Control), with these differences
  - PPP is byte-oriented, so it uses byte-stuffing to reach an integral number of bytes
  - HDLC provides reliability via a sliding window, acknowledgements, and retransmission. Though PPP does support reliable transmission, it's rarely used.
- PPP frame format
  - Flag Byte (1 byte): 0x7E, marking the beginning of the frame. If a payload byte $b_n = 0x7E$, then it's replaced with 0x7D (an escape byte) and $b_{n+1}$ is replaced with $b_{n+1} \oplus 0x20$
  - Address (1 byte): Always set to 0xFF to indicate that all stations should accept the frame.
  - Control (1 byte): Indicates the type of frame, usually 0x03 for unreliable service (unnumbered, unacknowledged frames)
  - Protocol (2 bytes): Codes starting with 0 are network layer protocols, codes starting with 1 indicate a PPP configuration protocol

    - ¤ Payload (variable length, up to a negotiated maximum, default 1500 bytes)
    - ¤ Checksum (2 bytes): negotiable up to a 4-byte CRC
    - ¤ Flag Byte (1 byte): marks the end of the frame
- It's possible to negotiate Address and Control fields away, and also to negotiate the Protocol field down to 1 byte
- PPP payloads are XOR'd with long pseudorandom sequences before transmission, to avoid long runs (so that SONET doesn't lose synchronization).

### 3.5.2 ADSL (Asymmetric Digital Subscriber Loop)

- DSL modem at the home sends bits over the local loop to the *DSLAM* (DSL Access Multiplexer) at the local office
- IP packets are transferred on the line using a variety of protocols, most comonly the following sequence: PPP, AAL5, ATM, ADSL
- ADSL physical layer uses OFDM/discrete multitone
- ATM (Asynchronous Transfer Mode)
    - ¤ A link layer based on sending fixed-length *cells* of data, not necessarily continuously
    - ¤ Connection-oriented, with virtual circuit identifiers in every header
    - ¤ Cells have a 48-byte payload and 5-byte header (a political compromise between 32 and 64 bytes!?!)
- AAL5 (ATM Adaptation Layer 5)
    - ¤ Responsible for segmentation/reassembly of data into a sequence of cells
    - ¤ uses padding and an 8-byte trailer (with a data-length field and a CRC field)
- PPP using PPPoA (PPP over ATM)
    - ¤ PPP framing is not needed since it's redundant with what ATM provides, only the protocol and payload are used
- ADSL is a noisy channel, so on the physical layer a Reed-Solomon code is used for error correction, and a 1 byte CRC is used as well.

## 3.6 Summary

# 4 The Medium Access Control Sublayer

- Broadcast networks' key issue is figuring out who gets the channel when there is competition for it
- Broadcast channels are called *multiaccess channels* or *random access channels*
- MAC is especially important in LANs. WANs tend to use P2P links (i.e. the graph of physical-layer connections is sparser), and thus contention is lower.

## 4.1 The Channel Allocation Problem

### 4.1.1 Static Channel Allocation

- Static FDM
    - ¤ Best for a small, constant number of users with heavy, continuous traffic demands
    - ¤ Computer traffic, though, is typically bursty (peak traffic to mean traffic ratios of 1000:1 are common)
    - ¤ FDM channel utilization
        - ∗ Assumptions
            - > mean time delay T seconds
            - > channel of C bps
            - > arrival rate of l frames per second
            - > each frame of length drawn from an exponential probability density function with mean $\frac{1}{\mu}$ (in other words, mean frame length is µbits)
        - ∗ So arrival rate is l frames/second and service rate is µC frames/bit.
        - ∗ From queuing theory, it can be shown that for Poisson arrrival and service times, $T = \frac{l}{\mu C - \lambda}$

    - Meaning, for C = 100 Mbps, $\frac{1}{\mu} = 10,000$ bits, l = 5,000 frames/sec, then T = 200 µsec
    - The naive calculation that it should only take 100 µsec to send a 10,000 bit frame on a 100 Mbps network doesn't take channel contention into account.
    - If we're dividing the channel N-ways via static FDM, we have $T_{FDM} = \frac{1}{\mu(\frac{C}{N}) - \frac{\lambda}{N}} = N(\mu C - \lambda) = NT$. So time delay is N times worse.
- Static channel allocation via other mechanisms (static TDM, splitting network capacity physically into statically-many separate channels) are just as bad as static FDM

### 4.1.2 Assumptions for Dynamic Channel Allocation

- 5 key assumptions
  ¤ *Station Model*
    - The model contains N independent *stations/terminals*, each of which generates frames for transmission
    - Once a frame has been generated, the station is blocked until the frame has been successfully transmitted

  ¤ *Single Channel Assumption*
    - A single, full-duplex channel is available for all communication.
    - Stations have equivalent hardware

  ¤ *Collision Assumption*
    - If two frames are transmitted simultaneously, they collide (they overlap in time and both are garbled).
    - All stations can detect collisions
    - Collided frames must be retransmitted
    - There are no other errors

  ¤ Time: either of the following
    - Continuous time: Frame transmission can begin at any instant
    - Slotted time: Time is divided into discrete slots/intervals, with frame transmissions only happening at a slot's beginning
      > This may require a master clock to signal the slots' beginnings
      > A slot may contain 0 frames (idle slot), 1 frame (successful transmission by one station), or many frames (collision; all stations will have to re-transmit).

  ¤ *Carrier Sense*
    - If a protocol assumes carrier sense, it means it has a limited ability to tell if the channel is busy before trying to transmit on it. Such protocols may not be practical for use with all physical configurations.
    - In protocols with carrier sense, if the channel is busy, no additional station will try to use it until it's idle (e.g., LANs, where latency is low enough to make waiting and watching work)
    - If an allocation method doesn't assume carrier sense, stations must transmit when they can, and determine success afterwards (e.g., wireless networks: because connectivity isn't transitive, one host has no way of knowing if another host is already receiving)

## 4.2 Multiple Access Protocols

### 4.2.1 ALOHA

- Intended for ground-based radio networking
- Pure ALOHA
  ¤ Sends immediately, but as a broadcast medium gets feedback, can easily determine if there was a collision by listening to the overall signal (for LANs, feedback is immediate, for satellite, it's 270 msec)
  ¤ If listening while transmitting is impossible, ACKs would be needed
  ¤ If a frame gets destroyed, resend after a random interval
  ¤ ALOHA's efficiency
    - Assuming users are always in a typing state or waiting-for-response state
    - Let t (the frame time) denote the time to transmit a standard, fixed-length frame

    * Assume users generate frames with a Poisson distribution with mean of N frames per frame time (assume infinitely many users so that N doesn't vary with the number of blocked users)
    * Then for $N \geq 1$, the channel will always be blocked with colliding transmissions.
    * Also assume the probability of k transmission attempts per frame time (novel transmissions and retransmissions combined) is also in a Poisson distribution, with a mean probability of G frames per frame time
    * Clearly, $G \geq N$. For low load ($N \approx 0$), $G \approx N$
    * For $P_0$, the probability of a frame not colliding, the throughput $S = GP_0$
    * A frame sent at time $t_0 + t$ gets into a collision if any other frame is sent between $t_0$ and $t_0 + 2t$
    * $Pr[k] = \frac{G^k e^{-G}}{k!}$, so the probability of 0 frames is $e^{-G}$, with the mean being $2G$, and the probability of only one frame occurring with no others initiating while it's vulnerable is thus $P_0 = e^{-2G}$.
    * So we have $S = Ge^{-2G}$
    * The maximum throughput is at $G = \frac{1}{2}$, with $S \approx 0.184$

- Slotted ALOHA
  - Stations agree on slot boundaries, usually by having one special station emit a heartbeat to start each interval
  - This halves frames' vulnerable period, giving $S = Ge^{-G}$, which peaks at $G = 1$, giving at best 37% idle slots, 37% slots with a successful transmission, and the remaining 26% of slots containing collisions

### 4.2.2 Carrier Sense Multiple Access Protocols

- Persistent and Nonpersistent CSMA
  - 1-persistent CSMA
    * Listens, and transmits as soon as the channel is free
    * Called 1-persistent becaues chance of transmitting when channel is idle is 1.
    * Propagation delay can cause even a listening station to introduce a collision with a transmission that's just begun
    * Even with no propagation delay, 2 stations queuing a message during a transmission are guaranteed to collide when the channel becomes free.
  - Non-persistent CSMA
    * Similar to 1-persistent CSMA
    * If a station wants to send while the channel is in use, waits a random period of time before listening again.
    * This reduces the odds of contention when the in-progress transmission completes (i.e., it increases channel utilization but pays for it in longer delays)
  - p-persistent CSMA
    * Usable with slotted channels
    * When a station is ready to send, it senses the channel. If the channel is idle, transmits with probability p, else waits until next slot before repeating the same procedure with the same probability.
    * Stations with a message queued repeat until a frame is transmitted, or until some other boorish station starts transmitting. In the latter case, the waiting station acts as if there was a collision (it waits a random time then retries)
- CSMA with Collision Detection (*CSMA/CD*)
  - Widely used on LANs on MAC, such as Ethernet
  - As soon as frames are seen colliding, stations stop sending (to save time and bandwidth), then wait a random duration
  - Stations detect a collision by noticing unexpectedly high power or pulse width on the line
  - Intuitively, if a station waits the longest propagation delay observable between stations on the broadcast network without hearing a collision, it could be sure its frame did not collide. This is not correct, though.
    * Suppose the greatest signal propagation delay between two stations is t.
    * At $t_0$, station A starts transmitting. At t - e, just before the signal arrives at the furthest station (station B), station B also starts transmitting. Station B will will stop transmitting almost instantly, but a collision has occurred, and station A won't know about it until 2t - e.
  - So we would set contention intervals of 2t (about 5 µsec on a 1 km coaxial cable)
  - Collision detection is an analog process, where a station's hardware needs to detect if its input from the line differs from its output to the line. Usually, this requires special encodings (otherwise, two 0-volt signals might look the same)
  - CSMA/CD is inherently half-duplex. It's impossible for a station to receive frames while it sends frames because receiving logic is already in use to detect collisions

### 4.2.3 Collision-Free Protocols

- Resolve channel contention without incurring collisions
- For these, we assume exactly N stations, each with an identity from 0 to N - 1 hardwired into it.
- a Bit-Map Protocol
  - Each contention period has N slots. Each station j wishing to transmit sends a 1 bit in the j'th slot reserved for it. When the contention period is complete, all stations have knowledge of each others' intent, and they transmit in numeric order of station id. Once all have transmitted, a new contention period begins.
  - This is called a *reservation protocol*
  - Time is measured in terms of the contention bit slot, with data frames of length d.
  - Stations with a low number typically become ready to send during the contention period, so they have to wait $\frac{N}{2}$ slots for the scan to finish, then another N for the next contention period to finish, before sending data.
  - Meanwhile, those with a high number typically are ready to send just before their turn in the contention period, so their average wait time is only $\frac{N}{2}$.
  - Thus the average time from ready-to-send to sending is N.
  - The overhead per frame is N bits, data is d bits, for a total efficiency of $d(N + d)$
  - At high load, the price paid for the N-bit contention period is small compared to payloads (only 1 bit per frame), yielding $\frac{d}{d+1}$
  - 1 bit per station overhead becomes costly for networks containing thousands of stations
- Binary Countdown
  - Stations broadcast their address, one bit at a time, from high-order to low-order: $a_1, a_2, ..., a_n$
  - Per-bit, all stations OR the result together to produce a 0 or a 1.
  - If the value is 1, then all stations that transmitted a 1 stay in the running to transmit. If it is 0, all stations stay in the running to transmit.
  - In the end, only the station with the highest address will still be a candidate.
  - That station will transmit, then a new round will begin.
  - In essence, this means the stations bid to transmit, and the station with the highest valued address will win.
  - Channel efficiency is $d(d + \lg N)$
  - If the frame format is chosen so that the sender's address is the first field in the frame, even the $\lg N$ bits aren't wasted
  - A variant exists in which binary countdown works in parallel, rather than serially
  - In another variant, the transmitting station gets assigned to the lowest address after winning an auction, and stations formerly below it are each promoted by one value. In this way no one station hogs the channel.

### 4.2.4 Limited-Contention Protocols

- Approaches to channel acquisition are contention vs collision-free, best considered in terms of delay at low load vs channel efficiency at high load
- So far all protocols give stations symmetric probabilities of channel acquisition
  - $k$ stations, each with uniform chance $p$ of having a transmission to do
  - Probability of channel being acquired during a slot is $kp(1-p)^{k-1} = Pr[channel\ acquisition]$
  - To find optimal $p$, take $\frac{\Delta}{\Delta p} Pr[channel\ acquisition] = 0$, solve for $p$, yielding $\frac{1}{k}$
  - $Pr[success\ with\ optimal\ p] = (1 - \frac{1}{k})^{k-1} = (\frac{k-1}{k})^{k-1}$. With even 5 stations, this slams into its asymptotic performance, $\frac{1}{e}$.
- Limited-contention protocols are meant to decrease competition by introducting indirection
- Divide stations into groups (not necessarily disjoint) $0..i..n$
- Each slot $i$ can only be claimed by members of group $i$
- Assigning stations to slots
  - Each group has 1 member–i.e., each station has its own slot (e.g., binary countdown)
  - Each group has $n \geq 2$ members. The chance that two or more of these want to transmit at the same time grows factorially.
  - 1 group with all stations (e.g., slotted ALOHA)
  - So ideally, many stations/slot when load is low, few when high–thus ideally we'd use dynamic slot allocation varying with load

- *Adaptive Tree Walk Protocol*
  - ¤ Devised by US Army in WWII to test soldiers for syphilis
  - ¤ Initially, slots are open to all $k$ stations. If a slot, slot 0, has a collision, then a special contention-resolution series of slots will occur. Slot 1 will be open for stations $0..\lfloor\frac{k}{2}\rfloor$. If a conflict arises there, then slot 2 is for $0..\lfloor\frac{k}{4}\rfloor$.
  - ¤ It continues like so until a non-collision transmission at slot $i$, at which point slot $i+1$ will be for $(\lfloor\frac{k}{2}\rfloor+1)..(k-1)$, which will resolve similarly
  - ¤ Essentially, if stations are the leaves of a binary tree, this uses DFS to find out which stations want to talk
  - ¤ Optimizations
    - ∗ Avoid polling the top of the tree (level 0) if traffic has recently been high, and always start the search at some level $i > 0$. Ideally, if we know there are $q$ ready stations, uniformly distributed, the number below a specific node is just $2^{-i}q$, and we would start the search from each station's parent, so start polling at a level such that $2^{-i}q = 1$ (i.e., at level $i = \lg q$).
    - ∗ Or suppose the last 2 stations of 4 want to transmit. The slot that polls all stations would get a collision. The slot that polls the first half of the stations would have no transmissions. It is clear from this that the 3rd slot would have the same stations causing a collision as caused a collision in the first slot, and it can be skipped over.

### 4.2.5 Wireless LAN Protocols

- Can be modeled as a LAN, except that carrier sense is no longer transitive (due to distance, and objects blocking the signals).
- Receiver in range of $\geq 2$ transmitters gets signals of varying power from each, so signal additive approaches like CDMA are unsuitable
- Carrier sense is not transitive (so a CSMA approach also doesn't work):
  - ¤ Interference between sender and reciever is not measurable by sender
  - ¤ Each receiver potentially has a disjoint set of stations able to send to it
- For collinear stations A, B, C, D
  - ¤ *Hidden station problem*: If C listens while A is transmitting to B, it might not hear and thus could also transmit, making B unable to receive either signal.
  - ¤ *Exposed station problem*: If B is sending to A, and C senses the medium, it would hear transmission, and think that station D may already be listening to B, thus it wouldn't send (i.e., it waits unnecessarily, even if D is out of range of B's transmission).
- *MACA (Multiple Access with Collision Avoidance)*
  - ¤ Station A wants to send to station B, sends it a special *RTS* frame (Request To Send) of 30 bytes, stating the length of the frame to follow
  - ¤ B replies with a *CTS* (Clear To Send) also containing the data length
  - ¤ A starts sending
  - ¤ Stations hearing an RTS stay quiet long enough for A to hear the CTS it's expecting
  - ¤ Stations hearing a CTS stay quiet long enough for the CTS to arrive at A and for B to receive all data from A.
  - ¤ Collisions still occur in MACA, when two stations send RTSs to the same receiver at the same time. Both transmitters will, having not gotten a CTS in a reasonable time, wait a random duration using binary exponential backoff, before trying again.
- *MACAW (MACA for Wireless)*
  - ¤ uses ACK frames after each data frame
  - ¤ Limited attempts at carrier sense so tations don't tlak over each others' RTSs
  - ¤ Run backoff per-(sender, receiver) rather than per-sender, to increase fairness (to chatty senders???)
  - ¤ Has a mechanism for stations to communicate about congestion, and ways of tuning backoff to be less disruptive in such a case.

## 4.3 Ethernet

### 4.3.1 Classic Ethernet Physical Layer

- 10Base5

  - thick coax cable, 10Mbps, max 500m per segment, max 30 nodes per segment
  - marked every 2.5m where you can insert *vampire taps*–pins that penetrate to the core to split the signal
  - Optimization
    * Transceiver is clamped around the cable at the tap site to ensure a good connection
    * Transceiver does carrier detection and collision detection
    * Transceiver cable/drop cable runs 5 lines of twisted pair into a controller interface of a node (data in, data out, control in, control out, transceiver power)
- 10Base2
  - thin coax cable, 10Mbps, max 185m per segment, max 30 nodes per segment
  - Uses BNC connectors forming a T-junction to split the signal
- Detecting problems (cable breaks, excessive length, bad taps, loose connectors) in a coax is done by *time domain reflectometry*–sending a pulse of known shape and waiting for it to echo back up the line
- 10Base-T
  - twisted pair cable, 10Mbps, max 100m per segment, max 1024 nodes per segment
  - Each machine wired independently to a hub, so that a new host is less likely to break the existing network
- 10Base-F
  - fiber cable, 10Mbps, max 2000m per segment, max 1024 nodes per segment
- Wiring a building / network topologies
  - linear: a single cable runs from room to room, is tapped
  - spine: a thick cable runs from roof to basement, with horizontal cables split off from it via amplifier
  - tree
  - segmented
- Repeaters receive, amplify, and retransmit–in both directions
- No two transceivers may be more than 2.5km apart, and there can't be any path between any two transceivers that traverses more than 4 repeaters
- Information sent using the Manchester Encoding

### 4.3.2 Classic Ethernet MAC Sublayer Protocol

- Ethernet uses the DIX (DEC, Intel, Xerox) frame format
- Preamble: 8 bytes, each containing 10101010, for synchronization (which in Manchester Encoding is a 10MHz square wave lasting 6.4μsec)
- Destination address: 6 bytes (2 bytes also okay if not baseband)
  - High order bit is 0 normally, or 1 for group addresses (used for multicast. If all 1's, then the destination is full broadcast
  - Bit adjacent to high-order bit distinguishes local from global addresses. Global addresses (all $2^{46} \approx 10^{13}$ of them) are assigned by IEEE
- Source address: 6 bytes or 2, under same conditions as with dest addr
- Type: 2 bits, so receiver knows what to do with the frame (i.e., which network layer process does the data go to
- Data: 0 to 1500 bytes (the maximum is due to historical high prices for RAM)
- Pad: 0 to 46 bytes, used to fill out frames that don't meet the minimum frame length
- Checksum: 32-bit CRC of payload
- Minimum frame length
  - Helpful to distinguish valid frames from garbage (garbage frames can be created when a transceiver detects a collision during a frame's arrival)
  - Hence, minimum of 64 bytes from destination address to checksum, inclusive (padding allows minimal frames to be sent with no payload)
  - Minimum frame length is helpful for collisions. If a sender starts sending a frame at time 0, finishes sending at time $a$, and it takes time $t$ time for the frame to travel to the farthest station, it would be $a + 2t$ before the sender has heard of a collision. hence all frames must take $a \geq 2t$ to send.
  - With length 2500m, and 4 repeaters, roundtrip times are about 50μsec worst case. At 10Mbps, that duration is occupied by 500 bits, which is rounded up to 512 bits = 64 bytes for safety

  - Increased network speeds require shorter segments or larger frames
- IEEE standard vs DIX standard
  - In IEEE standard, Preamble is only 7 bits, the last bit replaced with a Start of Frame delimiter
  - In IEEE standard, Type replaced with Length, indicating number of payload bytes in Data
  - This means that a header has to be added to the beginning of the Data field to help receivers dispatch an incoming frame
  - The IEEE standard completed too late, and everybody basically just uses DIX
- The Binary Exponential Backoff Algorithm
  - After the $i-1$'th collision, choose a random number between 0 and $2^j - 1$ (for $j = \min(i, 10)$) and wait that many time slots before trying again
  - When $i = 16$, sender gives up and returns failure

### 4.3.3 Ethernet Performance

- Suppose $k$ stations always ready to transmit, a constant retransmit probability in each timeslot, each station having uniform probability $p$ of transmitting during a contention slot,
- As before, $A = Pr[channel\ acquisition] = kp(1-p)^{k-1}$
- $A$ is max for $p = \frac{1}{k}$, with $\lim_{k\to\infty} A = \frac{1}{e}$
- Probability of contention interval having exactly $j$ slots is $A(1-A)^{j-1}$, so the mean number of slots per contention is $\displaystyle\sum_{j=0}^{\infty} jA(1-A)^{j-1} = \frac{1}{A}$
- Each slot has duration 2t, so the mean contention interval $w = \frac{2t}{A}$. So $w$ is at most $2te \approx 5.4t$
- If mean frame takes $P$ seconds to transmit, the channel efficiency is $\frac{P}{P+\frac{2t}{A}}$.
- For frame length $F$, network bandwidth $B$, cable length $L$, speed of signal propagation $c$, and the optimal case of $e$ contention slots per frame, we have $P = \frac{F}{B}$ and $t = BL/cF$
- So in fact, channel efficiency can also be expressed as $\frac{1}{1+\frac{2BLe}{cF}}$
- This shows a relation between cable length and channel efficiency, which is why the standard specifies a maximum segment length, and also why a linear topology is uncommon.
- It also shows that increasing bandwidth or segment distance, but not frame length, reduces efficiency.

### 4.3.4 Switched Ethernet

- Switch: contains 4-32 cards, joined by a backplane connector
- Frames from a node on the switch are checked by the node's card and sent directly to another node on the same card, or are sent through the backplane to the relevant card
- If 2 machines on the same card transmit at the same time, it depends on the card's construction
  - All ports are wired together to form a local LAN, and collisions are dealt with as CSMA/CD. This establishes each card as a separate collision domain
  - Alternately, a card stores frames in on-board RAM, allowing it to receive on all ports and schedule forwarding as needed. In this case, each port is its own collision domain
- Switches act to concentrate traffic–transmission must make many more hops to get from source to destination, but contention is much lower at every segment, so overhead is reduced (increasing efficiency)

### 4.3.5 Fast Ethernet

- Ring-based optical LANs
  - FDDI (Fiber Distributed Data Interface) and Fibre Channel
  - Station management was too complicated, making implementation slow, difficult, and expensive
- 802.34: Fast Ethernet
- Same as before, but bit time down from 100 nsec to 10 nsec; using only 10Base-T wiring or faster
  - 100Base-T4

    - 4 cat 3 cables (1 to hub, 1 from hub, 2 that switch to the current transmission direction)
    - Uses ternary encoding, doesn't use Manchester Encoding, at 25MHz to give 100Mbps in one direction, 33.3Mbps in the other direction
    - This scheme is called 8B/6T (8 bits map to 6 trits)
  - 100Base-TX
    - cat 4 cable, full duplex
    - Two twisted pairs per station
    - Uses a 4B/5B scheme taken from FDDI
    - Every group of 5 clock periods uses 4 bits for payload and 1 for control, the combinations specially chosen to aid in synchronization
  - 100Base-FX: Two strands of multimode fiber
  - 100Base-T2
    - 2 pairs of cat 3 wiring
    - Uses a special encoding schemd that requires sophisticated DSP hardware, so it's expensive
- Interconnection usable with 100Base-T
  - Hubs
    - All incoming lines are logically connected forming a single collision domain
    - Since only one station can transmit at once, communication is half-duplex
  - Switch
    - A must for 100Base-FX, which is very collision-prone due to high cable lengths possible

### 4.3.6 Gigabit Ethernet

- Goals
  - 10 times faster and backwards-compatible
  - so, same 48-bit addressing, same frame format (including min/max length)
  - unacknowledged unicast and multicast datagram service
- full-duplex mode
  - switch with buffer has separate to/from lines for each node, so a collision is never possible
  - so length of cable is determined only by signal strength, rather than by time for collision to propagate back to sender
- half-duplex mode
  - when computers are attached to a hub, which simulates multidrop
  - uses CSMA/CD
  - Because of 100x higher speed, max segment distance would have to be only 25m–unacceptably low. These two techniques can be used to extend to 200m.
    - *Carrier Extension*: the hardware adds its own padding to extend frames to 512 bytes. i.e., now transmitting 46 bytes of payload per 512 bytes data, for 90% line efficiency
    - *Frame Bursting*: Sender buffers frames and sends multiple frames per burst. If a burst doesn't have 512 bytes, pad it again
- Signaling at $\geq 1$ Gbps over fiber requires a light source that can turn on and off in under 1 nsec.
  - So LEDs don't work.
  - But LASERs do! Two wavelengths are permitted: short (0.85 microns) and long (1.3 microns)
- 3 fiber diameters are permitted: 10, 50, 62.5 microns (latter two for multimode)
- Uses 8B/10B
  - Each 8-bit byte is encoded as 10 bits s.t.
    - No codeword has ¿ 4 identical bits in a row
    - No codeword has ¿ 6 0's or ¿ 6 1's
  - This keeps enough transitions in the stream for synchronization by keeping the counts and distribution of 0's and 1's close to even
  - When the encoder has a choice of codewords, it always chooses the one that moves toward equalizing the number of 0's and 1's. this balance lets the DC component of hte signal stay low enough to pass through transformers.

- GigE on 1000Base-T uses a different encoding scheme
  - ¤ 5 twisted pairs allow 4 symbols sent at once
  - ¤ Each symbol is encoded with one of 4 voltage levels
- GigE needs to support flow control
  - ¤ So it doesn't swamp an Ethernet receiver and cause buffer overruns
  - ¤ One end sends a special control frame (with Type 0x8808) asking for a short pause.
  - ¤ First two bytes of data give command, further bytes are for parameters.

### 4.3.7 10-Gigabit Ethernet

- 10GBase-T
  - ¤ Uses 4 pairs of twisted pair cable, sending 2500Mbps in both directions over each pair
  - ¤ Uses a signaling rate of 800Msymbols/sec with symbols using 16 voltage levels. The symbols are the result of scrambling the data, use of an LDPC (Low Density Parity Check), and a little bit of error correction
- Fiber variants
  - ¤ 10GBase-SR uses multimode fiber (at 0.85µ) for ranges up to 300m
  - ¤ 10Gbase-LR and 10GBase-ER use single-mode fiber at 1.3µand 1.5µto reach longer distances, up to 40km.
  - ¤ They send a serial, scrambled stream of information encoded with a 64B/66B code.

### 4.3.8 Retrospective on Ethernet

- Simple (reliable, cheap, easy to maintain)
- Meshes well with TCP/IP, which is connectionless and dominant
- Has been able to evolve (speed increased without any real changes to software)

## 4.4 Wireless LANs

### 4.4.1 The 802.11 Architecture and Protocol Stack

- With 802.11, the transmission technology is tightly coupled with the MAC sublayer, so we'll consider them together
- Two modes
  - ¤ *Infrastructure mode*: each client associates with an *AP* (Access Point), through which it sends and receives packets. one or more APs can connect together into a *distribution system*
  - ¤ *Ad hoc network*: a collection of computers associating to send frames to one another (with no intermediate access point)
- 802.11 supports an infrared option and short-range radio options (FHSS and DSSS), both on the 2.4 GHz ISM band which is also used with household electronics, but at low enough power to avoid conflicts
- 802.11 is Infrared, 802.11b is HR-DSSS (11 Mbps), 802.11a is OFDM (54 Mbps), 802.11g is also OFDM but on a different frequency band (also 54 Mbps), 802.11n is MIMO-OFDM (up to 600 Mbps).

### 4.4.2 The 802.11 Physical Layer

- Uses *rate adaptation*: all versions of 802.11 adjust traffic rate to match conditions (e.g., signal strength), most of them allowing rate variations up to a factor of 10.
- 802.11 – Infrared
  - ¤ Uses diffused (not line of sight) transmission at 0.85 or 0.95 microns, at 1 or 2 Mbps
    - ∗ at 1 Mbps, groups of 4 bits are encoded as a gray code, having 15 0's and one 1. (???!)
    - ∗ at 2 Mbps, encoding takes 2 bits and produces a 4 bit codeword
  - ¤ Due to low-bandwidth and interference from sunlight, infrared is unpopular
- 802.11 – FHSS (Frequency Hopping Spread Spectrum)

  - 79 channels, each 1 MHz wide
  - PRNG chooses sequence of channels to hop to
  - Stations with same seed and with times in sync will hop to same frequency at same time
  - *Dwell time*: time spent at each frequency, is adjustable but must be $\geq$ 400 msec
  - Obscurity of seed and dwell time provides a little security, and hopping protects against multipath fading
  - Isn't sensitive to radio interference but bandwidth is low

- 802.11b at lower speeds – DSSS (Direct Sequence Spread Spectrum)
  - somewhat similar to CDMA, but with one spreading code shared among all users.
  - bits transmitted as 11 chips, using a *Barker sequence*, a sequence which has low autocorrelation (i.e., a receiver can easily align with the spectrum hopping)
  - uses BPSK or QPSK
- 802.11b – HR-DSSS (High Rate Direct Sequence Spread Spectrum)
  - Uses 11 million chips/sec for $\geq$ 11 Mbps in 2.4 GHz band
  - Supports 1, 2 Mbps at 1 baud, with 1 or 2 bits/baud (resp), using phase shift modulation (for DSSS compatbility)
  - Also supports 5.5 Mbps, 11 Mbps at 1.375 Mbaud, with 4 or 8 bits/baud, using Walsh/Hadamard codes
  - Though slower than 802.11a, range is 7x greater
- 802.11a/g – OFDM (Orthogonal Frequency Division Multiplexing)
  - 802.11a can do 54 Mbps in the 5 GHz ISM band; 802.11g does the same in the 2.4GHz band.
  - 52 frequencies used (48 for data, 4 for synchronization)
  - Splitting signal into many narrow bands gives resistance to narrow band interference, and an opportunity to use noncontiguous bands
  - Uses a complex phase shift modulation encoding. At 54 Mbps, 216 data bits are encoded into 288-bit symbols
  - Good spectrum efficiency (w.r.t. bits/Hz) and resistance to multipath fading
- 802.11n – MIMO-OFDM
  - The channel size doubled from 20MHz to 40MHz
  - Uses up to 4 antennas. The streams may interfere but they can be split with MIMO (Multiple Input Multiple Output) techniques
  - The extra antennas can be used for greater speed, reliability, or range

### 4.4.3 The 802.11 MAC Sublayer Protocol

- DCF (Distributed Coordination Function)
  - CSMA/CA (CSMA with Collision Avoidance)
    - Physical Channel Sensing: Stations wishing to send will sense, then transmit a full frame when the channel idles. In the case of a collision, both use binary exponential backoff before resending
    - Virtual Channel Sensing: based on MACAW: RTSs and CTSs trigger listening stations to stay quiet until the sender is done sending or the receiver sends and ACK
- Wireless in the ISM ranges are strongly affected by noise from other consumer electronics
  - 802.11 allows frames to be broken up and sent sequentilly in a single RTS-CTS-ACK channel acquisition in a stop-and-wait fashion (each fragment in such a fragment burst gets its own ACK)
- PCF (Point Coordination Function)
  - Base station polls other statoins to see if they need to transmit and then schedules them to avert collision
  - Base station polls with a *beacon frame* periodically (usually 10-100 per second) containing parameters (hopping sequences, dwell times, clock synchronization) and also inviting other stations to sign up for polling service (which in effect guarantees the station a fraction of available bandwidth).
  - Base stations can also instruct a mobile station to go to sleep until explicitly awakened, though in such a case a base station MUST buffer frames intended for the mobile station
- The hidden station and exposed station problems mean that CSMA/CD are unsuitable for 802.11 (also since most radios are half-duplex and can't listen for a collision while transmitting)
- Each frame carries a NAV (Network Allocation Vector) field inidcating how long the sequence (the one to which this frame belongs)will take to complete. Stations overhearing a NAV know the channel is busy for the period indicated, even if they can't hear a physical signal.

- RTS/CTS uses NAV to prevent terminals from sending at the same time as a hidden terminal.
- RTS/CTS not as useful as hoped
  - ¤ Doesn't help for short frames (they would be sent in the place of an RTS)
  - ¤ Not really different from MACA: fixes hidden terminal problems but not exposed terminal problems.
- Reliability
  - ¤ Main tactic to boost successful transmissions is to reduce the transmission rate
  - ¤ An alternative is to send shorter frames: For $Pr(a\ single\ bit\ in\ error) = p$, an $n$-bit frame has a $(1-p)^n$ chance of arriving without error.
  - ¤ 802.11 allows fragementing with maximum fragment sizes adjustable by the AP
  - ¤ Fragments are numbered and acknowledged with a stop-and-wait protocol.
- Power conservation
  - ¤ AP sends *beacon frames* periodically announcing its presence and parameters
  - ¤ Clients can announce to the AP that they're entering power-save mode. The AP will buffer the client's traffic. Beacon frames indicate which clients have data queued and how much, so a power-saving client only needs to poll the AP if it sees its own name
  - ¤ *APSD* (Automatic Power Save Delivery): the AP buffers frames per client and sends them only once the client sends it a frame.
- Quality of Service
  - ¤ After each frame is an interframe time interval of one of several types
    - ∗ SIFS (Short InterFrame Spacing): Lets stations with the talking stick coordinate (receiver can send a CTS to respond to an RTS, sender can send a new fragment after an ACK, receiver can send an ACK)
    - ∗ AIFSi 1 (Arbitration InterFrame Spacing): Stations with high-priority traffic can start talking
    - ∗ PIFS (PCF InterFrame Spacing): Base station can send beacon frame or poll frame
    - ∗ DIFS (DCF InterFrame Spacing): Any station can try to acquire the channel
    - ∗ AIFS 4: Stations with low-priority traffic can start talking
    - ∗ EIFS (Extended InterFrame Spacing): Used for stations to report that they've received a bad frame
- *TXOP* (Transmisssion Opportunity)
  - ¤ CSMA/CA only works for same-speed contenders. A fast sender will tend to be pulled down to the speed of the slowest contender.
  - ¤ TXOP assigns fixed equalish amounts of airtime per station, rather than fixed equalish numbers of frames

#### 4.4.4 The 802.11 Frame Structure

- 3 different frame types (data, control, management)
  - ¤ Data frames
    - ∗ Frame control (has 11 subfields) (2 bytes)
      - > Protocol version (2 bits)
      - > Type(2 bits): d, c, m
      - > Subtype (4 bits): RTS, CTS, ACK
      - > To DS (1 bit): true if going to intercell distribution system
      - > From DS (1 bit)
      - > MF (1 bit): true if more fragments are coming
      - > Retry (1 bit): marks a retransmission
      - > Power Management (1 bit): toggles base station's sleep state
      - > More (1 bit): true if sender has more frames for receiver
      - > W (1 bit): true if frame body is encrypted with *WEP* (Wired Equivalent Privacy)
      - > O (1 bit): true if frames must be processed strictly in order
    - ∗ Duration (2 bytes): estimates how long frame+ACK will occupy the channel
    - ∗ 4 different address fields (each 6 bytes)
      - > 2 addresses for source and destination
      - > 2 others are used for base stations at which traffic arrived or should leave the cell
    - ∗ Sequence (2 bytes): 12 bytes for frame identification, 4 bits for fragment identification

    - Data (up to 2312 bytes)
    - Checksum (4 bytes)
  - Management frames: similar, but with only one of the base station addresses
  - Control frames
    - Only one or two addresses, no Data field, no Sequence field
    - Chiefly used for Subtypes: RTS, CTS, ACK

#### 4.4.5 Services

- Must provide 5 distribution services (relate to managing cell membership and interacting with stations outside the cell) and 4 station services (relate to in-cell activity)
- Distribution Services
  - Association: allows mobile stations to connect to base stations
    - MS announces identity and capabilityes (data rate supported, need for PCF services, power management needs)
    - base station can accept or reject
    - MS must then authenticate itself
  - Disassociation: Can be initiated by base station or mobile station
  - Reassociation: used for a mobile station ot change its preferred bas station as it moves from one cell to another
  - Distribution: Determines how to route frames to a destination via the base station
  - Integration: For frames sent out of the cell, this service translates the addresses into something the destination network will understand
- Station services
  - Authentication
    - After association, base station sends a challenge frame
    - Mobile station encrypts the challenge frame and sends that back
  - Deauthentication: used by a mobile station wishing to leave the network
  - Privacy: Managing encryption and decrypting using RC4 (or AES, with WPA2)
  - Data Delivery: Not guaranteed to be reliable; higher layers must do work to detect and correct errors
- *QoS Traffic Scheduling* can give realtime traffic preferential treatment compared to normal/best-efforts service and background service
- *Transmit Power Control* service gives stations information they need to stay within local regulatory limits on power transmission
- *dynamic frequency selection* tells stations how to avoid transmitting on parts of the 5 GHz spectrum already used for radar

## 4.5 Broadband Wireless

- *WiMAX* (Worldwide Interoperability for Microwave Access)
- Meant for the "last mile": telecoms wanted a multi-megabit service for rural/remote customers
- 802.16: Air Interface for Fixed Broadband Wireless Access Systems / wireless MAN / wireless local loop

#### 4.5.1 Comparison of 802.16 with 802.11 and 3G

- Combines aspects of 802.11 and 3G, but is more like 3G
- Like 802.11, high speed is a design goal, based on MIMO and OFDM
- Like 3G
  - It needs to be able to support many subscribers in little spectrum
  - Coverage areas are large: up to 10x as large as an 802.11 network
  - Hence, transmissions are carefully scheduled by the base station: CSMA/CA is too collision-prone
- Closely resembles the 4G network *LTE* (Long Term Evolution)

### 4.5.2 The 802.16 Architecture and Protocol Stack

- Mobile stations and subscriber stations (fixed location stations) associate with a base station, which is connected to the Internet
- Like other 802 networks, but with more sublayers
  - Physical Transmission Sublayers
    - "Fixed WiMAX" sublayer (802.16a) uses OFDM
    - "Mobile WiMAX" sublayer (802.16e) uses scalable OFDMA
  - Transmission Convergence Sublayer: to hide the above from the data link layer
  - Security Sublayer: encryption, decryption, key management
  - MAC Sublayer common part
    - Base station controls the system, can schedule the downstream channels, and plays a part in managing upstream channels
    - Is connection-oriented, to provide quality of service guarantees
  - Service-specific Convergence Sublayer: interface to network layer protocols (both connectionless and connection-oriented)

### 4.5.3 The 802.16 Physical Layer

- The spec allows operation between 2GHz and 11GHz (usually WiMAX uses licensed spectrum around 2.5 or 3.5 GHz), with varying channel sizes
- Compared to 802.11, channels are split into many more subcarriers, and symbol lengths are longer (e.g., for Mobile WiMAX, 512 subcarriers on a 5MHz channel, and symbol duration of 100µsec
- Symbols are sent with a modulation scheme chosen to overcome the subcarrier's signal-to-noise ratio
- OFDMA (Orthogonal Frequency Division Multiple Access) is quite flexible, letting the base station assign subcarriers to stations that are best able to receive/transmit on those frequencies.
- TDD (Time Division Duplexing) or FDD (Frequency Division Duplexing) are usable to split upstream vs downstream traffic by time or frequency (respectively). TDD is preferred because it's easier to implement.
- Assuming TDD, time is sliced into downstream frames and upstream frames
  - Downstream frame starts with a Preamble on all subcarriers to synchronize all stations
  - Downlink map is broadcast on all subcarriers to announce station $\mapsto$ subcarrier assignments for downstream traffic
  - Uplink map is broadcast on a subset of subcarriers to announce station $\mapsto$ subcarrier assignments for upstream traffic. During this time, on the other subcarriers, data is already being sent.
  - After the downstream frame finishes, there's a small guard time before the Uplink frame
  - In the Uplink frame, a set of bursts is reserved for *ranging* (for new stations to request to connect to the base station)

### 4.5.4 The 802.16 MAC Sublayer Protocol

- Security sublayer
  - When a subscriber connects, they mutually authenticate with RSA using X.509 certificates
  - Payloads are encrypted with symmetric keys, using either DES with CBC, or triple DES with two keys
  - Integrity checking is done with SHA-1
- MAC sublayer common part
  - MAC frames occupy physical layer time slots, each composed of subframes. The first two subframes are downstream and upstream maps, which declare what is in which timeslot and which timeslots are free
  - Allocation of upstream
    - Is complicated because subscribers are competing and uncoordinated
    - Four classes of service
      - Constant bitrate service
        - Intended for uncompressed voice
        - Needs to send a fixed amount of data at predetermined time intervals
        - Time slots are allocated for the duration of the connection
      - Realtime variable bitrate service

- For compressed multimedia where bandwidth needed varies with time
- The base station repeatedly polls the subscriber at a fixed interval to determine how much will be needed

> Non-realtime variable bitrate service

- For heavy non-realtime transmissions, like file transfer
- Base station polls subscriber at non-fixed, frequent intervals
- A constant bitrate subscriber can set a bit in one of its outgoing frames to request such a poll in addition to its normal traffic load
- If a station doesn't respond within k contiguous pollings, the base station puts it into a multicast polling group, where any of the group's members may contend to respond to their shared polling. This minimizes the bandwidth wasted by quiet stations.

> Best efforts service

- Subscribers in this class of service are not polled–they must contend for bandwidth by seizing time slots marked as free in the upstream map timeslot
- Successful bandwidth requests are acknowledged by assignments in the downstream map, with binary exponential backoff used for collisions

¤ Bandwidth allocation

- Can be done per-station (such that each building represents its users' collectively and then doles out bandwidth at is sees fit)
- Can be done per-connection

### 4.5.5 The 802.16 Frame Structure

- MAC frames
  - ¤ 0 (1 bit)
  - ¤ EC (1 bit): whether payload is encrypted
  - ¤ Type (6 bits): frame type (e.g., whether packing or fragementation are in use)
  - ¤ 1 ignored bit
  - ¤ CI (1 bit): presence or absence of final checksum
  - ¤ EK (2 bits): which encryption key is in use
  - ¤ 1 ignored bit
  - ¤ Length (11 bits): length of complete frame (incl. header)
  - ¤ Connection ID (16 bits)
  - ¤ Header CRC (8 bits): checksum over header only
  - ¤ Data (optional): Not needed for control frames (e.g., requesting time slots)
  - ¤ CRC (4 bits, optional): CRC for full frame. If the frame has no data, this is excluded. It is also excluded for realtime connections, for which the physical layer does error-correction.
- Frames that request bandwidth
  - ¤ 1 bit, 0 bit (2 bits)
  - ¤ Type (6 bits)
  - ¤ Bytes needed (16 bits)
  - ¤ Connection ID (16 bits)
  - ¤ Header CRC (8 bits)
- Notice that these frames are defined to fill an integral number of bytes

## 4.6 Bluetooth

### 4.6.1 Bluetooth Architecture

- Basic concept is a *piconet*: 1 master node and $\geq 7$ active slave nodes within 10m
- *Scatternet*: multiple piconets interconnected via a bridge node
- $\geq 255$ parked nodes in a piconet (devices set to low power, can only respond to activation or beacon signals from the master)
- Master/slave configuration allows most bluetooth devices to be really dumb

### 4.6.2 Bluetooth Applications

- V1.1 spec gave 13 applications/*profiles* to be supported, each with a different protocol stack to support its particular needs. Only the first two are mandatory
- *Generic Access*: Provide a way to establish and maintain master/slave links
- *Service Discovery*: Permit devices to discover what other devices have to offer
- *Serial Port*: Emulates a serial line (and forms the basis of most of the other profiles)
- *Generic Object Exchange*: Defines a client-server relationships for moving data around
- *LAN Access*: Connect to a fixed network
- *Dial-up Networking*: Permit a laptop to connect to a mobile phone having a modem
- *Fax*: Allow receipt of faxes via mobile telephones
- *Cordless Telephony*: Connect a cordless telephone handset to its base station
- *Intercom*: Permit two telephones to act as walkie-talkies
- *Headset*: Hands-free voice communication between headset and base station
- *Object Push*: Exchange simple data objects
- *File Transfer*: General file transfer facility
- *Synchronization*: Allow mobile devices and computers to exchange information accumulated while apart

### 4.6.3 The Bluetooth Protocol Stack

- Figure 4-37
- Physical radio layer (radio transmission and modulation that are optimized for inexpensiveness)
- Baseband layer: how master controls timeslots and fits frames into them
- Link manager: Establishes logical channels between devices (power management, authentication, quality of service)
- Logical Link Control Adaptation Protocol (L2CAP) provices a slicker abstraction to layers above
- Middleware Layer: many protocols
  - RFcomm: emulates a serial port
  - Telephony: realtime protocol used for speech-oriented profiles (also does call setup and termination)
  - Service Discovery Protocol: locate services within the network
- Additionally, Audio and Control layers that don't pass through Link Manager, L2CAP, or middleware layers

### 4.6.4 The Bluetooth Radio Layer

- Divides the 2.4 GHz ISM band into 79 channels of 1 MHz apiece
- Modulated with frequency shift keying (1 bit/Hz $\rightarrow$ 1 Mbps gross data rate)
- FHSS with 1600 hops/sec and dwell time of 625 µsec
- In the same band as 802.11, so the two interfered, but Bluetooth now implements *adaptive frequency hopping* to exclude channels with other RF signals.

### 4.6.5 The Bluetooth Link Layers

- Link control / baseband layer: akin to a MAC sublayer
  - Turns raw bitstreams into frames
  - A piconet's master defines 625 µsec timeslots, master's transmissions starting in even slots, slaves' transmissions starting in odd slots, frames able to be {1, 3, 5} slots long
  - After a frequency hop, it takes 250-260 µsec for radio circuits to settle, leaving as little as 366 µsec to send up to 366 bits of a single frame. 126 of these are access code and header, leaving 240 bits for data
  - When 5 slots are strung together into a frame, 2781 of the 3125 bits are available for payload, making long frames much more efficient
- Link manager protocol
  - Frames are sent over a *link* (logical channel)
    - ACL (Asynchronous Connection-Less) link: packet switched data at irregular intervals. Dealt with on the L2CAP layer. Delivered on a best-efforts basis. Only 1 ACL link per (master, slave) pair

    - SCO (Synchronous Connection-Oriented) link: realtime data. Allocated a fixed slot in each direction. No retransmission (because realtime), but forward error correction is a supported option. A slave can have $\geq 3$ SCO links at a time with its master, each able to support a 64 kbps PCM audio channel

- L2CAP layer
  - Accepts packets $\leq 64KB$ from upper layers, breaks them into frames for transmission
  - Multiplexes and demultiplexes multiple packet sources
  - Handles quality of service requirements, and parameter compliance (e.g., max packet sizes)

### 4.6.6 The Bluetooth Frame Structure

- Many frame formats, most important format has these fields
  - Access code (72 bits): identifies the master so that two masters heaving a slave can figure out which traffic is for them
  - Header (54 bits): similar to other MAC frame headers. It's repeated 3 times per frame. For each header bit, receivers count the 1s and 0s and accept the value that has the highest count. This redundancy is meant to protect against problems relating to high noise/weak signals
    - Address (3 bits): identifies which of the other 8 devices the frame is for
    - Type (4 bits): frame type (ACL, SCO, poll, null), type of error correction, slot length
    - Flow (1 bit): used by a slave to slow traffic when its buffer is full
    - Acknowledgement (1 bit): for piggybacking an ACK onto a data frame
    - Sequence (1 bit): Implements stop-and-wait
    - Checksum (8 bits)
  - Data
    - ACL uses multiple formats
    - SCO data is {80, 160, 240} bits of payload, the rest being used for error correction
      - in the case of 80-bit payloads, the payload is just repeated 3 times
      - Slaves use the 800 odd timeslots/sec, yielding 64kbps for the highly-reliable 80-bit payloads–enough for a single full-duplex PCM voice channel to saturate the piconet
      - At 240 bit payloads, a single slave can support 3 slightly less reliable full-duplex PCM channels

## 4.7 RFID

- This section focuses on EPCs (Electronic Product Codes), specifically EPC Gen 2

### 4.7.1 EPC Gen 2 Architecture

- Tags
  - Store a 96 bit EPC identifier and a small amount of memory that can be read from or written to by a reader
  - Calss 1 tags have no batteries and gather power only from nearby RFID readers' transmissions
- Readers
  - In charge of when tags send and receive messages
  - Must resolve multiple access problem in case many tags are nearby
  - Usually have their own power source, sometimes have multiple antennas

### 4.7.2 EPC Gen 2 Physical Layer

- In the US, uses UHF (902-928 MHz)
- Hops frequencies every 400msec
- Uses ASK, and reader and tags take turns sending bits (half-duplex)
- Backscatter method
  - Reader transmits a signal at all times (a fixed carrier signal even when a tag is transmitting) to power the tags.

  - ¤ Tags alternate between absorbing and reflecting signals
  - ¤ Tag signal is usually weak, so transmission rate is low.
  - ¤ For readers: to reduce price and power costs on tags, only simple modulation is used: a short duration of high power between low power signals is a 0, a long duration of high power is a 1
  - ¤ Tags use between one and eight pulse periods to encode each 0 or 1 bit, depending on reliability desired.

### 4.7.3 EPC Gen 2 Tag Identification Layer

- uses a variant of slotted ALOHA to solve the multiple access problem
- Reader sends a Query request to begin the first slot, then $n-1$ QRepeat requests, for $n$ slots total
- Tags pick a random slot and send a short, random 16-bit reply in an RN16 response.
- Readers receiving an RN16 (and not observing a collision) acknowledge the RN16 with an ACK response. That tag now owns that slot.
- The goal of having RN16 replies (rather than sending EPC IDs) is to reduce the time lost due to collisions during slot assignment. Since the bitrate is so low, 16-bit RN16s are much shorter to send than 96 bit EPC IDs
- Readers seeing many collisions or too few slot acquisitions can send a QAdjust message to increase or decrease the range of slots over which tags may respond

### 4.7.4 Tag Identification Message Formats

- Query message
  - ¤ Command (4 bits): identifies the message as a query
  - ¤ DR (1 bit), M (2 bits), TR (1 bit): physical parameters for e.g. transmission rate
  - ¤ Sel (2 bits), Session (2 bits), Target (1 bit): These enable a reader to select classes of tags that should respond. Tags can maintain 4 concurrent sessions.
  - ¤ Q (4 bits): indicates the number of slots (so readers can resolve contention between at most $2^Q - 1$ in-range EPC tags)
  - ¤ CRC (5 bits)
- Tag-to-reader messages carry only the data payload requested

## 4.8 Data Link Layer Switching

- Bridge: device examines data link layer addresses to do routing, is thus agnostic to higher layers' protcols
- Router: device examines addresses in packets to do routing, is sensitive to protocols used above the data link layer
- Use cases for bridges
  - ¤ Join distinct/autonomous LANs (e.g., university departments)
  - ¤ Distance may make it costly or impossible to cable an entire organization on the same LAN
  - ¤ Allows multiple LANs to implement a single logical LAN in cases when the load exceeds 1 LAN's capacity
  - ¤ Reliability: prevent rogue nodes from bringing down the system with erratic transmissions by partitioning it
  - ¤ Security: isolates subparts of the network so that simple snooping can't see all traffic
- Ideally, moving a node from one bridge to another should require no reconfiguration, and machines on one should be able to communicate with nodes on another transparently

### 4.8.1 Uses of Bridges

- Use cases for bridges
  - ¤ Join distinct/autonomous LANs (e.g., university departments)
    - ∗ Frame formats may differ: Copying between LANs requires reformatting, checksum recalculation, and may introduce new errors
    - ∗ Interconnected LANs may have different data rates: Fast-to-slow may cause buffers to fill
    - ∗ Different maximum frame lengths: Data link layer protocols mostly don't support frame splitting/joining, so overly large frames would just get dropped

  - ¤ Distance may make it costly or impossible to cable an entire organization on the same LAN
  - ¤ Allows multiple LANs to implement a single logical LAN in cases when the load exceeds 1 LAN's capacity
  - ¤ Reliability: prevent rogue nodes from bringing down the system with erratic transmissions by partitioning it
  - ¤ Security: isolates subparts of the network so that simple snooping can't see all traffic
    - ∗ 802.11 and 802.16 both support DL layer encryption, 802.3 doesn't
    - ∗ A frame encrypted by an 802.11 sender would then arrive at the receiver undecryptable
    - ∗ Unencrypted frames from an 802.3 sender might, after passing through a bridge, be sent out in plaintext on an 802.16 network, which is also bad
    - ∗ Delegating encryption to a higher layer could work, but the process of delegating destroys transparency

### 4.8.2 Learning Bridges

- ”Ethernet switch” is a fancy name for a bridge joining multiple Ethernet LANs
- Transparent Bridging
  - ¤ Bridge operates in promiscuous mode, accepting all frames on all LANs it's attached to
  - ¤ For all frames, the bridge discards it (if the frame doesn't need to leave its LAN of origin) or forwards it (if the frame needs to be delivered outside its LAN of origin)
  - ¤ *backward learning* uses hashtables to discover destination $\mapsto$ LAN mappings and forward frames on to the relevant LANs
    - ∗ When frames come in to the bridge, the frame's source address and the LAN it came from are associated in the bridge's hash
    - ∗ Since topologies can change, these associations are also kept up-to-date with a time last seen
    - ∗ Every few minutes, the bridge's hashtable has old values swept out
    - ∗ Frames arriving with identical source LAN and destination LAN are discarded (they're already on the right network)
    - ∗ Frames arriving with destination LAN unknown are flooded onto all ports except the source port (once a mapping has been discovered, frames for that destination will no longer be flooded, but sent on directly)
    - ∗ This *cut-through switching/wormhole routing* is very fast. Because destination addresses are usually sent in headers, forwarding can begin with minimal queuing.

### 4.8.3 Spanning Tree Bridges

- For increased reliability (i.e., in case a single bridge or cable fails), some sites join a pair of LANs with multiple bridges, creating a topology that is problematic because of its loops
- e.g., suppose two bridges that don't have routing info for a certain destination both receive a packet for it. They'll flood the frame forward at each other, echoing copies on potentially forever.
- Basic solution: bridges communicate about the topology, form a spanning tree to pretend the loops don't exist
  - ¤ Bridges choose a root: all bridges have a guaranteed unique serial number and elect the one with the lowest serial number
  - ¤ From this node, a spanning tree is constructed
  - ¤ This procedure is rerun continually to detect topology changes

### 4.8.4 Repeaters, Hubs, Bridges, Switches, Routers, and Gateways

- *Repeaters* (physical layer)
  - ¤ Analog devices attached to 2 cable segments, and amplify voltages without understanding frames
- *Hubs* (physical layer)
  - ¤ Multiple input lines are joined electronically, and transmissions are blindly multiplexed onto every line (forming a single collision domain)
  - ¤ All lines must run at the same speed
- *Bridges* (data link layer)
  - ¤ Connects $\geq 2$ LANs by forwarding frames between them

- Connected LANs must all be of the same physical type (Ethernet, token ring, ...) so that the frames can be mutually read by all line cards without being reassembled
- Each line is its own collision domain
- LANs can have different network types and speeds

- *Switches* (data link layer)
  - Connects $\geq 2$ hosts, usually has many more line cards than a bridge because it is host-oriented
  - Each line is its own collision domain, thanks to frame buffering
  - Frame buffering does cause backups, so some implement *cut-through switching*: where the switch starts emitting the frame as soon as the destination address has fully arrived (assuming the outgoing line is free)
  - Since switches mostly work via hardware nowadays, the distinction between bridges and switches has become mostly marketing
- *Routers* (network layer)
  - Strips off frame header and trailer, then looks in the payload for routing instructions
- *Transport Gateways* (transport layer)
  - Translates from one connection-oriented protocol to another
- *Application Gateways* (application layer)
  - Able to parse data at the application level and translate from one format to another (e.g., gateway to turn SMSes into emails)

### 4.8.5 Virtual LANs

- It's desirable to split LANs by factors other than geography and cabling topology alone, which has these benefits (assuming splits by department/group)
  - Security: cross-group listening can't happen if LANs are split by group
  - Load: Isolates heavy-use groups from other groups (who may need realtime capacity always available)
  - Broadcasting: Isolation reduces the congestion caused by broadcast messaging
    - Sometimes a broken network interface will start creating an endless stream of broadcast frames, creating a *broadcast storm*. Since bridges and switches try to operate independently, these would usually propagate broadcast storms through the entire logical LAN.
- Virtual LAN
  - Older switches/bridges might not be VLAN-aware
  - Store a VLAN configuration table linking each port to a LAN and vice versa
  - How a bridge/switch knows what VLAN an incoming frame belongs to
    - Every port is assigned a VLAN: only works if all machines connected via a port belong to that one VLAN
    - Every MAC address is assigned a VLAN: Lets a bridge/switch mix VLANs on a physical LAN, requires the bridge/switch to kep a table of VLAN $\leftrightarrow$ MAC address mappings.
    - Every layer 3 protocol/IP address is assigned a VLAN: Bridge/switch examines payload and discriminates on traffic type and address. Of course, mixing of layers breaks layer abstractions (i.e., many switches have hardware that does this, and works for IPv4 but not for IPv6).
- The IEEE 802.1Q Standard
  - What really matters is not VLAN of the sender, but VLAN of the frame. Newer protocols (802.11, 802.16) use Connection Identifers somewhat in this spirti.
  - But we also want the same for good ol' Ethernet.
  - 802.1Q is an update ot the Ethernet standard that acheives this by adding a VLAN tag
  - Tags are added to frame not by the originator, but by first VLAN-aware bridge/switch (and stripped off by the last VLAN-aware bridge/switch) on the router, using the messy methods described above.
  - Frame length maximum is increased to 1522 bytes to make this work
  - In order for the last VLAN-aware switch to know whether to strip the tag off or not, it now needs to know for each port, what VLAN it belongs to, and whether that port is VLAN-aware.
  - The VLAN tag, inserted into the frame just before Length, contains
    - VLAN protocol ID (2 bytes): is always $\geq$ 0x8100 (that's greater than 1500, so hardware that's not VLAN-aware would never interpret it as a valid length)

    - Priority (3 bits): can be used to distinguish traffic by quality-of-service required (hard realtime, soft realtime, time-insensitive)
    - Canonical Format Indicator (CFI) (1 bit): indicates the payload is an untouched 802.5 frame looking for an 802.5 LAN at the destination. This is only there for political reasons.
    - VLAN identifier (12 bits)

## 4.9 Summary

- Figure 4-52: table of systems for allocating a common channel

# 5 The Network Layer

- Gets packets from source to destination, potentially making many hops
- Lowest layer involved in end-to-end transmission
- Must know the topology of the subnet and choose paths appropriately

## 5.1 Network Layer Design Issues

### 5.1.1 Store-And-Forward Packet Switching

- Host sends packet to nearest router (over a LAN or point-to-point line)
- The packet is stored until it's arrived fully and its checksum is verified
- The packet is forwarded to the next router along the path

### 5.1.2 Services Provided to the Transport Layer

- Goals of the service
  - ¤ Services should be independent of router technology
  - ¤ Transport layer should be shielded from number, type, and topology of routers present
  - ¤ Network addresses available to transport layer should be uniform, even across LANs and WANs
- Connectionless vs Connection-oriented service debate
  - ¤ Connectionless faction
    - The Internet is the champion of this faction
    - Network is inherently unreliable, regardless of its design
    - Error-control, flow control, and packet ordering are problems for higher layers
    - The only primitives needed are send_packet() and receive_packet()
  - ¤ Connection-oriented faction
    - Championed by telephone companies
    - Quality-of-service dominates other considerations, and is nigh-impossible without connections
  - ¤ IP/connectionless mas mostly won the war, though connection-ish technologies (such as VLANs and MPLS) as the internet develops needs for quality-of-service.

### 5.1.3 Implementation of Connectionless Service

- Packets are called *datagrams*, the network is called a *datagram network*
- Connection-oriented service: a source to destination path is allocated (called a *Virtual Circuit* (VC), in analogy with physical circuits allocated in telephone networks)
- Routers store a table mapping final destination to router, where the router is one of the ones connected to it directly. It uses this table to decide where to send each packet. Each packet must carry its full destination address for this lookup to be possible.
- This routing table (and accordingly, all routing decisions) are maintained by a *routing algorithm*.

### 5.1.4 Implementation of Connection-Oriented Service

- Once a VC is established, packets use short labels to convey which connection they belong to.
- If a VC routes using machine sequence (A, B, C, D), packets sent from A to B will contain A's identity, and A's internal ID for the connection. B looks up this pair to yield that the next machine to route to is C, and its own internal ID for the connection (which C will use in turn to make its own routing decision). For bidirectional routing on the VC, B would store such pairs for A and C both.
- This approach is sometimes called *label switching*, and is used in MPLS (MultiProtocol Label Switching)

### 5.1.5 Comparison of Virtual-Circuit and Datagram Networks

- Fig 5-4 shows the major issues
- VCs require connection setup time, but datagrams use time by parsing more address data at each router.

## 5.2 Routing Algorithms

- *Session routing*: packets follow an already-established route (as in a VC) for the duration of some session
- *Forwarding*: The actual act of receiving a packet, looking up its outgoing line, and transmitting it
- The hard part of routing is in updating the routing tables
- Desirable properties for a routing algorithm: correctness, simplicity, robustness, fairness, efficiency
  - ¤ Robustness: Tolerant to hardware failures, software failures, and topology changes
  - ¤ Stability: should be quick to reach equilibrium / converge on a fixed set of paths, to reduce disruption and overhead
  - ¤ Fairness (mean packet delay) and efficiency (total network throughput) can be (perversely) in conflict. As a compromise, we often try to optimized for minimum total travel distance for packets, or minimal packet hops
- *Nonadaptive routing algorithms* vs *dynamic routing*

### 5.2.1 The Optimality Principle

- Optimality principle: regardless of topology, the optimal J $\rightsquigarrow$ K path is entirely in the optimal I $\rightsquigarrow$ K path if the latter passes through J (stated by Bellman).
- From this, we know the optimal route to any destination from all sources forms a tree rooted at the destination (called a *sink tree*).
- A graph and a destination may have multiple distinct sink trees; all of these combined would form a DAG
- Assumptions for the sake of simplicity
  - ¤ Paths don't interfere (congestion on one path doesn't disrupt another even if they pass through common routers)
  - ¤ Links and routers (by going down at different times) won't have different models of the topology
  - ¤ How routers get topology information on the network as a whole

### 5.2.2 Shortest Path Algorithm

- Weighting function for edges usually some combination of distance, bandwidth, average traffic, communication cost, measured mean delay.
- Dijkstra's Single-Source-Shortest-Path algorithm

### 5.2.3 Flooding

- When a router has poor knowledge of the topology, it may flood packets outward
- To prevent infinite packet multiplications, flooded packets set a maximum hop count, set either to the hop count to the destination (if known), or the worst case hop distance (the diameter of the network)
- To additionally prevent exponential growth of the same flooded packet, a source router can assign a sequence number to packets coming from their hosts. Then intermediate routers need only remember a set of sequence numbers per source router to avoid reflooding the same packet again.

- ¤ Can also store a counter $k$ to indicate that all packets with sequence number $\geq k$ have been seen, which provides a mechanism for limiting the list's growth and the number of sequence numbers to inspect per incoming packet.

- Flooding does have its uses
  - ¤ Delivery to all nodes supplies a means for broadcasting
  - ¤ Robustness
    - ∗ Routers can be blown to bits in wars, and flooding will find a route if there is one
    - ∗ Routers can flood with little-to-no knowledge of the network, hence it being the building block of most routing algorithms' setup phase.

### 5.2.4 Distance Vector Routing

- Distance Vector Routing: each router maintains a table of (destination, best known distance to that destination, which link to use to get there) vectors. These get exchanged with neighbors continually until they all agree on the best paths
- Also called *distributed Bellman-Ford routing*
- The *count-to-infinity* problem
  - ¤ When a router $A$ goes down, suppose its neighbors (in order of increasing cost) are $a_0$, $a_1$, ..., $a_n$. $a_0$ starts reporting its cost to $A$ as $cost(a_1, A) + cost(a_0, a_1)$, a route which incorrectly relies on $A$ still being reachable through $a_1$. In this way, the cost to reach $A$ increases across the network, but very slowly.

### 5.2.5 Link State Routing

- Replaced DVR by dealing with the above problem
- Link State Routing variants IS-IS and OSPF are in most common use now
- Each router must do all of the following
  - ¤ Discover its neighbors and learn their addresses
  - ¤ Set distance (or some other cost metric) for each neighbor
  - ¤ Construct a packet with its findings
  - ¤ Send this packet to (and receive it from) all other routers
  - ¤ Compute the shortest path to all other routers
- Neighbor Discovery
  - ¤ Newly booted-up routers greet their neighbors and expect a response from each with their globally unique identifiers
  - ¤ Broadcast links are modeled as individual nodes $N$, in which case one router on the LAN becomes the designated router to represent $N$.
- Determining Link Costs
  - ¤ Cost can be inverse of link bandwidth (such that higher capacity paths are better choices)
  - ¤ Delay can be a useful cost metric for geographically spread-out networks
- Building Link State Packets
  - ¤ Each packet has a router identifier, sequence number, age, and a list of (neighbor, cost) pairs
  - ¤ These can be built either periodically, or when a significant change to topologies or costs occurs
- Distributing Link State Packets
  - ¤ LSPs are flooded, with sequence numbers used to decide whether to discard, or to use and forward
  - ¤ Wraparound of sequence numbers can cause a problem here, but we sidestep it by using 32-bit numbers, which will fill 137 years at a rate of 1 $\frac{LSP}{sec}$
  - ¤ Routers that crash and reissue a sequence number 0 could get ignored for a very long time
  - ¤ Corrupted sequence numbers may cause whol swaths of LSPs to be ignored (e.g., 65,540 and 4 are only 1 bit different)
  - ¤ The problems above are solved with a TTL on each LSP. The TTL of the LSP's information gets decremented every second and discarded when 0. In this way, outdated sequence numbers, outdated costs, and phantom packets disappear within a fixed duration.
  - ¤ LSPs coming in for flooding are put into a holding area briefly (in case a link is going down / coming up and is thus sending out naive topology information)
  - ¤ All LSPs get acknowledged

  - Each router stores a tuple of (source router ID, sequence number, TTL) followed by 2 bits per adjacent router, indicating whether it's already sent or received this information to/from this router
- Computing new routes
  - Each router independently runs Dijkstra's SSSP algorithm
  - LSR requires more memory than DVR: for $n$ routers, each with $k$ neighbors, the memory table is $nk$ entries large, and the duration spent running Dijkstra's SSSP algorithm grows faster than that product alone.
- IS-IS (Intermediate System-Intermediate System) is used by many ISPs
- OSPF (Open Shortest Path First) designed later by IETF, uses many features from IS-IS
  - self-stabilizing method for flooding link status updates
  - concept of designated routers on a LAN
  - allows computing and supporting path splitting and multiple metrics
- IS-IS's main benefit is the ability to carry multiple network layer protocols at the same time
- Generally, routing algorithms assume that all routers are working cooperatively and in good working condition. A router that routinely corrupts or does not forward packets, or misreports link status, can disrupt the network severely. The key is to arrange the network so that the damage will be limited.

### 5.2.6 Hierarchical Routing

- At scale it's desirable to segment the network hierarchically so routers needn't know about every other router.
- Routers are divided into regions, with routers knowing all other routers within their region, but knowing nothing about other regions' interiors
- The value here is decreased CPU and memory costs per-router, but the price paid is increased path length: routing decisions will be based not on the path length that'd be shortest for that packet, but may be based on other factors (e.g., shortest median path between regions' members)
- The optimal number of levels for an $N$ router network is $\lg N$, requiring $e \ln N$ entries per router.

### 5.2.7 Broadcast Routing

- Simplest way to achieve all-destinations routing (broadcasting): send a distinct packet with the same payload to each destination
  - slow, wastes bandwidth, requires source to have a complete list of destinations
- Multidestination Routing
  - Each packet has either a list of destinations or a bitmap indicating the desired destinations
  - Routers receiving such packets may clone them to mux them out on multiple lines
  - "Multidestination routing is like using separately addressed packets, except that when several packets must follow the same route, one of them pays full fare and the rest ride free"
  - This not only requires the source to know all the destinations, but it also makes routers do significantly more work to route such a packet.
- Flooding
  - With per-source sequence numbers, links are used efficiently without abusing routers
- Reverse Path Forwarding
  - When router $B$ sees a broadcast packet from $A$, it checks whether it came on the line it usually uses to send to $A$. If so, chances are it's the fastest it could've seen that packet, so it floods it onward. Otherwise, it discards it as a likely duplicate.
  - In this way, RPF basically goes in reverse along the shortest paths the destinations' packets would take to the source.
  - This makes it generally efficient and easy to implement, while also not overusing lines, and not requiring the broadcaster to know a full list of the destinations
  - question from markz: what if path costs are higher from broadcaster to receiver on the best receiver to broadcaster path?
- Spanning Tree Flooding
  - A subset of the network containing no loops and all routers (sink trees are a type of spanning tree)

  - If each router knows of a common spanning tree in the network, it just forwards the packet on every line belonging to the spanning tree except its incoming line. This ensures that we send the absolute minimum number of packets to do the job.
  - Routers don't always have knowledge of a spanning tree (they would with Link State Routing, but not with Distance Vector Routing)

### 5.2.8 Multicast Routing

- MMO games and realtime video streaming set a need for sending messages to well-defined groups that are large in size, but small compared to the network as a whole
- Current multicasting schemes all require some means to create/destroy groups, and to identify which routers belong to a group
- Multicast is based around broadcasting to a spanning tree in the network, with the optimal spanning tree determined by the group and the network
- For a dense group, routing is usually done to a subset of the network's spanning tree, pruned to disregard links not connecting any members
  - When LSR is used, each router can construct and prune a spanning tree itself by constructing a sink tree and pruning links that don't connect the sink to any members of the group. MOSPF (Multicast OSPF) does this.
  - When DVR is used, a variant of reverse path forwarding is used.
    - The first packet is multicast to all routers.
    - A router $X$ whose hosts are not group members will respond to the router $Y$ that forwarded to $X$ with a PRUNE message.
    - If all $W$, $X$, $Z$ that $Y$ forwarded to respond with a PRUNE message, then $Y$ also replies upstream with a PRUNE message.
    - Thus the tree is pruned recursively.
    - DVMRP (Distance Vector Multicast Routing Protocol) works this way
- An alternative is the core-based tree approach, where all routers use the same spanning tree per group
  - All routers agree on a core / rendezvous point and build a tree by each member sending a packet to the root (the tree being the union of these paths)
  - Multicast packets are routed to the core, and outward from there (following forks in the tree as they occur)
  - In the case of multicast senders not on the tree, bandwidth may be wasted. The sender-to-core route may miss fast links to members simply because the packet isn't yet traveling on the line.
  - To ease this, cores are usually chosen close to the multicast's main sender(s).
- Shared tree approaches save on storage and computation, so such approaches are used (e.g., for PIM (Protocol Independent Multicast)).

### 5.2.9 AnyCast Routing

- AnyCast: a packet is delivered to the nearest group member
- To enable this, members of an anycast group are assigned a common address, in effect reducing the group to some virtual node
- This works with nothing special in DVR. In LSR, it requires only that link state protocols don't try to route through the virtual node at a cost of 0 (this is easy if the protocol can distinction between routers and hosts)

### 5.2.10 Routing for Mobile Hosts

- Mobile hosts: hosts that move (in bursts or constantly)
- Rejected solutions
  - Fully dynamic model
    - Recompute routes on every topology change. But this way the network is never settled if many hosts are mobile
  - Implement mobility above network layer
    - Happens currently with laptops: old addresses and new addresses are no way related

    * However, this makes it hard to send data to the mobile host – the only fix is to reestablish connections as addresses change

- The considered solution is for mobile hosts to have a *home location* that identifies it.
- Every time a host moves, it registers its new address with its *home agent* (a host at its home location), which can then forward packets to it.
- The local address for a mobile host is its *care-of address*. The mobile host registers this with its home agent via a control agent.
- Packets intended for a mobile host are sent to its permanent address. Its home network encapsulates the packet with a new header and sends it along to the care-of address (a process called *tunneling*)
- Mobile hosts can reply directly to the sender, at which point the sender may learn of the care-of address.
- This is called *triangle routing* because some packets might have to travel quite far even if source host and destination host are nearby.
- This is basically how the internet (IPv6) and IP-based cellular networks (UMTS) work.

#### 5.2.11 Routing in Ad-Hoc Networks

- When even routers are mobile, e.g. emergency work, or laptops where 802.11 isn't running
- *ad-hoc networks* or MANETs (Mobile Ad hoc NETworks)
- Topology may be changing continuously
- MANETs are hardly used, so it's unclear which proposed protocols are most useful in practice
- AODV (Ad hoc On-demand Distance Vector)
  - ¤ Similar to fixed-router Distance Vector Routing, adapted for nodes being mobile, with bandwidth and battery life limited.
  - ¤ Route Discovery
    - ∗ Routes are discovered on-demand, only when a packet needs to be sent.
    - ∗ Source floods/broadcasts a ROUTE REQUEST packet, each node rebroadcasting until it reaches the destination
    - ∗ Destination constructs a ROUTE REPLY packet and sends it along the reverse of the path followed by the ROUTE REQUEST to which it is responding
    - ∗ Intermediate nodes increment a hop count as the two packet types are sent, giving a rough distance metric
    - ∗ As the reply returns to the source node, intermediate nodes add the route to their routing tables.
    - ∗ In large networks, this flooding is very costly. One strategy used is for ROUTE REQUESTs to first be sent out with a TTL of 1 hop, then if it gets no reply, another with 2 hops, etc..
  - ¤ Route Maintenance
    - ∗ In order to detect topology changes, each node periodically sends out a greeting. If no reply from $B$ back to $A$ comes, $A$ prunes all routes relying on that former neighbor $B$ and broadcasts a notice that $A \to B$ routes are no longer usable, which its neighbors will rebroadcast
    - ∗ To avoid slow convergence/count-to-infinity problems, routers also keep a sequence number, which is incremented with every ROUTE REPLY it generates. To speed up convergence, intermediate nodes use this to keep only the latest route to each destination.
    - ∗ Routes also time out, so nodes needn't store outdated data.
- Other schemes
  - ¤ DSR (Dynamic Source Routing)
  - ¤ GPSR (Greedy Perimeter Stateless Routing) uses information about nodes' geographical positions to guess a route by just aiming in the right direction, and circling back to escape dead ends.

### 5.3 Congestion Control Algorithms

- Network layer shares this responsibility with transport layer
- Congestion collapse: when performance plummets because offered load exceeds capacity.
- Network delay can cause packets to exceed their TTL, get dropped, and be retransmitted by the senders, wasting all bandwidth spent on the delayed packets. Hence, we prefer to measure *goodput*, the rate at which usable packets are delivered by the network, rather than throughput.
- Buffering can help, but it is not helpful if delays cause messages to be queued until they time out instead of being dropped outright.

- Congestion control refers to the capacity of the overall network; flow control, meanwhile, refers to the ability of a single receiver to keep up with its sender.

### 5.3.1 Approaches to Congestion Control

- Network Provisioning: build/upgrade the network in anticipation of heavy load
- Traffic-aware Routing: Make routing aware of load patterns so that traffic is spread evenly.
- Admission Control: In VC networks, connections can be denied if load is too high to support more traffic.
- Traffic Throttling: Detect congestion and request sources to slow down
  - ¤ Detecting congestion can be seen from average load, queuing delays, or packet loss
  - ¤ The rate at which congestion control mechanisms throttle matters: overly twitchy, and the system will show volatile/bursty patterns; too slow and the network will suffer for a long time before congestion is alleviated.
- Load shedding: Discard packets that the network can't deliver

### 5.3.2 Traffic-Aware Routing

- Simplest way: include average queuing delay in link weights, so that lightly loaded paths will have lower weights.
- This may cause load to oscillate between several links rather than settle evenly between them.
- Multipath routing is a solution that can avoid this, by allowing a single source and destination to have multiple good paths available at once
- Alternately, if weights change only very slowly due to load, the oscillations will be dampened and converge to an even split
- The Internet's routing protocols don't pay attention to load directly, only via changing parameters, via *traffic engineering*

### 5.3.3 Admission Control

- Even VC networks don't have homogeneous per-connection bandwidth needs (unlike the telephone network), so VC networks need some characterization of their traffic to meaningfully apply this approach.
- e.g., web traffic is very bursty – heavy or nothing at all – and is often described as a leaky bucket / token bucket.
- Some approaches
  - ¤ During setup, source states its expected bandwidth usage and any router along the connection path can refuse to provide that guarantee.
  - ¤ Combine with traffic-aware routing to prefer low-load paths for new connections

### 5.3.4 Traffic Throttling

- Load feedback is business-as-usual rather than exception, with the overall goal of maintaining high usage just shy of congestion.
- Routers need to monitor for signs of congestion
  - ¤ Utilization of output lines: burstiness of traffic makes this metric not meaningful at any particular point in time, or with any particular running average
  - ¤ Count of packets lost due to insufficient buffering: This is a valuable metric, but it only gives a positive too late
  - ¤ Occupancy of packet queue inside router: usually low until the network is backed up. Usually uses an EWMA (Exponentially Weighted Moving Average), which smooths out fluctuations:
    - ∗ $d_{new} = \alpha d_{old} + (1 + \alpha)s$, for
    - ∗ $d$ = delay estimate,
    - ∗ $s$ = sample of queue's length, and
    - ∗ $\alpha$, a constant determining how fast the router forgets recent history
- Load feedback needs to be sent in a way that won't further clog the network
  - ¤ Choke packets
    - ∗ Select a congested packet and generate a *choke packet* to transmit to its sender, noting the original packet's destination, and send at a low rate

    * Source host would then be expected to reduce traffic by a fixed amount (50%ish?). Source hosts would ignore subsequent choke packets for a short period until its throttling could be felt by the network
  - Explicit Congestion Notification (ECN)
    * Routers tag packets going through with a bit marking it as congested. On the destination's next reply, it will notify the sender.
  - Hop-by-hop Backpressure
    * Congested routers select a congested packet and send a choke packet back to its sender.
    * Every intermediate router, and the sender, will throttle traffic along that route.
    * This immediately addresses congestion where it's happening, at the cost of sometimes overthrottling all routers on that path unnecessarily.

### 5.3.5 Load Shedding

- When dropping packets, deciding which to keep may be important.
- Two basic policies:
  - *Milk* – new is better than old – relevant for realtime media, where old packets are mostly useless
  - *Wine* – old is better than new – relevant for file transfer, where receivers can't use newer packets until old ones have arrived
- More intelligent policies are possible whereby senders can mark packets' priorities
  - e.g., routing packets should never be dropped
  - e.g., when sending MPEG traffic, packets containing key frames shouldn't be dropped, packets containing a frame's diffs from the key frame can
  - Such a policy can be abused, but externalizing the cost of prioritizing packets (as additional fees) can reduce that.
- Random Early Detection
  - TCP primarily detects congestion by noticing packet loss, which is already too late (also, historically, this is why wireless needs to recover transmission errors in the link layer, to play nice with TCP).
  - RED routers maintain a running average of their queue lengths
  - When that length exceeds some threshold for a link, it starts dropping packets at random. Random dropping ensures that the faster the sender, the sooner they'll lose a packet.
  - Senders will notice packet loss and start throttling automatically (the network usage of choke packets is not incurred)
  - RED is graceful when hosts can't receive explicit ECNs, but ECNs will trigger throttling sooner.

## 5.4 Quality of Service

- *Overprovisioning* works (e.g., when have you ever picked up a telephone and not gotten a ringtone) but is expensive
- Four main issues
  - What do applications need from the network
  - How do you regulate traffic entering the network
  - How do you reserve resources at the routers to guarantee performance
  - How do you determine when the network can safely accept more traffic

### 5.4.1 Application Requirements

- A *flow*: A stream of packets from a source to a destination
- A flow's requirements are determined primarily by four parameters (see Fig 5-27)
  - bandwidth
  - delay
  - jitter
  - loss
- ATM networks provided QoS categories that still shape types of service typically offered.
  - constant bitrate (e.g., telephony)

  - realtime variable bitrate (e.g., compressed video conferencing)
  - non-realtime variable bitrate (e.g., watching a movie)
  - available bitrate (e.g., file transfer)

### 5.4.2 Traffic Shaping

- Data networks are bursty: traffic rate varies (compressed data), users interact with applications, computers switch between tasks
- With *traffic shaping*, user and network can agree on a traffic pattern so that the network can deliver on a QoS standard. Such agreements are sometimes called Service Level Agreements (SLAs)
- *Traffic Policing*: when networks monitor that a customer's traffic pattern is within agreed-upon parameters. May also involve deprioritizing or dropping that customer's packets
- Leaky & Token Buckets
  - Commonalities in the metaphors: there is a bucket of capacity B and traffic (either traffic in or traffic out) with rate R
  - Leaky Bucket Algorithm
    - To send a packet, the bucket must not be full
    - If the bucket is full when a packet arrives, the packet must be queued until the bucket can hold it (forcing the sending/forwarding host to do its own shaping), or the packet goes into the bucket and immediately spills out (via a provider's network interface policing traffic)
  - Token Bucket Algorithm
    - To send a packet, we must be able to remove a token from the bucket
    - A bucket's capacity limits its long-term data transfer rate, while R reflects how well it can cope with (and pass along) bursts
    - Token bucket implementation
      - Keep a counter for the level of the bucket that advances at $\frac{R}{\Delta T}$ for every clock tick of duration $\Delta T$.
      - As traffic arrives, the counter is decremented, either by the count of packets arriving, or by the number of bytes arriving
      - Packets are only sent if the residual byte count is high enough
      - To determine the length of the maximum burst, we need to consider tokens arriving during the burst. For $S =$ the number of seconds in the burst, $M =$ the maximum output rate, we have $B + RS = MS$ or $S = \frac{B}{M-R}$
    - Token bucket algorithms reduce large bursts down to the long-term rate $R$; but it may be better only to ensure that hosts stick to the long-term rate while merely moderating the peak rate
    - One way to do this is with a second bucket: the first one limits the peak traffic that will enter the network, the second has a higher rate, so it will moderate the rate at which permissible traffic enters the network.

### 5.4.3 Packet Scheduling

- Packet scheduling algorithms allocate router resources to packets, so that QoS guarantees can be made when there are competing flows
- Flows can be allocated any of these 3: bandwidth, buffer space, CPU
- The most straightforward scheduler: a FIFO queue
  - Normally, packets arriving at a full queue are the ones to get dropped; these dropped packets are called *tail drops*
  - The RED algorithm and other alternatives exist
  - FIFO is poor on fairness: a single bursty flow can lodge itself in the queue and cause delays and drops for other flows
- Nagle's *fair queuing*
  - For each (flow, output line), there is a queue. These are scanned round-robin style
  - This gives unfair advantage to flows with larger packets
  - There's an improvement which calculates for each packet an arrival time, an expected sending duration, and sends those with the lowest sum first
  - Another problem is that this gives all hosts the same priority
- *Weighted Fair Queuing* (WFQ)

  - $F_i = max(A_i, F_{i-1}) + \frac{L_i}{W}$, for $W$ = the weight of a flow, $A_i$ = the arrival time of packet $i$, $F_i$ = predicted finish time of packet $i$, $L_i$ = the length of packet $i$
  - Implementation
    - WFQ requires insertion into a sorted queue, at best $O(\lg N)$ with $N$ flows
    - *Deficit round-robin* is an approximation of WFQ that does insertion in $O(1)$
- Priority scheduling
  - Packets always send higher priority packets first
  - Disadvantage: High priority flows with entirely starve low priority ones.
  - WFQ with large weight differences between $N$ 'virtual flows' can simulate a fairer priority scheduler with $N$ priority levels
- Timestamp-order scheduler
  - Packets sent with a timestamp are forwarded in timestamp order
  - Packets which have been held up longer will thus get sent sooner
  - This results in a more consistent amount of delay

### 5.4.4 Admission Control

- Routers along the path from source to destination(s) will reserve resources if possible, or else reject the flow
- If the flow is rejected along the best path, *QoS routing* (if available) will try to find alternate paths with excess capacity, or split capacity over multiple paths
- The accept/reject decision isn't actually simple
  - Even applications that know their bandwidth requirements can't be expected to model the costs of the routers' operation (w.r.t. their buffers and CPU cycles)
  - Some applications are more tolerant of missed deadlines than others
  - Some applications may be able to negotiate interactively about flow parameters
- Senders start with a *flow specification* - a specific proposed guarantee
- Example flow specification (based on RFCs 2210-2211)
  - Token bucket size, token bucket rate
  - Peak data rate
  - Minimum packet size, maximum packet size
- Maximum packet rate helps routers determine which links can not be reserved
- Choosing resources to reserve based on a flow specification is not straightforward
  - Queues can fill up (and thus delays can occur) for a network, even if operating below its theoretical capacity
  - For packets arriving at random, with mean arrival rate of $\lambda \frac{packets}{sec}$, with random length, able to be sent on the link with mean service rate of $\mu \frac{packets}{sec}$, if both arrival and service distributions have a Poisson distribution (an M/M/1 queuing system), queuing theory proves that the mean time delay for any packet is $T = (\frac{1}{\mu}) \times (\frac{1}{(1-\lambda)\mu}) = (\frac{1}{\mu}) \times (\frac{1}{1-\rho})$, where $\rho = \frac{\lambda}{\mu}$ is the CPU utilization
  - $\frac{1}{\mu}$ represents the service time in the absence of competition
  - The second factor is the slowdown due to competition
- One good way of translating flow specifications
  - Traffic sources are shaped by (R, B) token buckets and WFQ at routers
  - WFQ uses a weight $W$ large enough such that $R < \frac{W \times C}{\Sigma weights}$, for outgoing line capacity $C$
  - This guarantees new flows a minimum bandwidth
  - The largest queuing delay the flow can see is then $D = \frac{B}{R}$
  - These are hard guarantees: token buckets bound sources' burstiness and WFQ bounds bandwidth per flow, so that no competing flow can make the router break the guarantee.

### 5.4.5 Integrated Services

- Intended for multicast (and unicast as a special subcase thereof) of streaming media (TV) or large (video)conferences.
- IN such a case, new destinations may join at any time, so reservation of resources in advance won't work

- RSVP (Resource reSerVation Protocol)
  - RFCs 2205-2210
  - Capabilities
    - Multiple senders can send to multiple groups of receivers
    - individual receivers can switch channels freely
    - bandwidth is used optimally and congestion is reduced
  - Normal multicast routing with spanning trees, except with an extra packet periodically telling routers on the tree to maintain certain data structures
  - If a receiver is getting poor reception and wants more bandwidth sent to it, it sends a reservation request to the source through the spanning tree (using reverse path forwarding)
  - Each router along the path reserves more resources or rejects the request
  - If the request makes it to the source, then all resources have already been reserved for the source-to-receiver path in the tree
  - A receiver can request multiple sources at once, and can specify whether this set of sources is to be fixed for the duration of the reservation, or whether the choices might change

#### 5.4.6 Differentiated Services

- Integrated services aren't widely deployed
- Flow-based algorithms scale poorly when there are many flows
- IETF has designed a simpler approach to QoS which doesn't require routers to know the whole path−*class-based* QoS
- e.g., Differentiated Services, RFCs 2474, 2475, and others
- Routers in an administrative domain (ISP, telco) define a set of service classes and associated forwarding rules
- Customers subscribing to a differentiated service entering the domain are marked with the class, each class defining a *per-hop behavior* that the class entitles at every router
- e.g., internet telephony traffic as a whole may be given certain behaviors that amount to a QoS guarantee, even though no one call has a guarantee of its own
- Service classes may vary between providers, but for interoperability, IETF has specified two taxonomies of service classes
  - Expedited forwarding (RFC 3246)
    - There are two kinds of packets, expedited and regular
    - Expedited packets are not delayed by non-expedited packets
    - Intended for low-loss, low-delay, low-jitter service
    - sending host, or ingress (first) router labels packets so that they'll get the preferential treatment while on participating links / in the participating domain
  - Assured Forwarding (RFC 2597)
    - Four priority classes (these determine preferred delay characteristics) and 3 discard classes (for packets experiencing congestion), for a total of 12 classes
    - Each priority class is given its own traffic policer, which identifies how the priority class's packets fit into bursts (small burst, normal burst, oversized burst), encoding that with the packets
    - A priority scheduler (perhaps using WFQ) will allocate more bandwidth to higher priority classes, and an algorithm such as RED will discard packets from each priority class as necessary to keep all four priority class's queues usable

### 5.5 Internetworking

- It's naive to think a homogeneous network is possible: even if one wired network was used everywhere, wireless needs will necessarily have a lot of variation in terms of receiver power/distance/contention.
- Network heterogeneity is here to stay, but it's haaaaaaaaard.

#### 5.5.1 How Networks Differ

- Service offered (e.g., connectionless, connection-oriented)
  - connectionless traffic passing across a connection-oriented link will need to spend overhead setting up a (dummy) connection

- Addressing (different address sizes/formats; flat vs hierarchical)
- Broadcasting/multicasting capability
- Packet size
- Ordering (is ordered delivery guaranteed?)
- QoS
- Security
- Parameters (timeouts, flow specifications)
- Accounting (cost per bit/byte/packet/connection duration)

### 5.5.2 How Networks Can Be Connected

- TCP/IP protocols early on provided a common layer to hide these sorts of differences
- IP provides a universal packet format
- Packets must carry a network layer address that can identify any host (on any network in the internetwork)
- Thus, routers (unlike switches/bridges at the DL layer) must be able to understand the network layer protocol(s) being used
- *Multiprotocol routers* can understand multiple protocols
  - Some delegate connections to a higher layer
    - This is bad because new applications must then share a protocol
  - Some translate from one network layer protocol to another at the network boundaries
    - This is also bad, because there may be fundamental differences (e.g., IPv6 addresses are way bigger than IPv4 addresses, making this impossible)

### 5.5.3 Tunneling

- *Tunneling* is used for the special case when source and destination networks are of the same type and intermediate networks are of others
- The packet can simply be boxed up and put inside a packet of another type, to be unboxed when it enters the destination network (or another network of that type)
- When modeling isolated networks of one type joined by other networks using tunneling, we speak of the derived network as an *overlay*, since it abstracts the other network types away

### 5.5.4 Internetwork Routing

- Complications
  - Different routing algorithms: hop-based routing networks which don't precalculate paths will disrupt networks whose routing expects to calculate the entire path ahead of time
  - Different operators may calculate paths weights differently, resulting in wasteful inconsistencies in routing
  - Different operators may wish not to reveal their networks' interior
  - A large enough internetwork may require a structure that's not flat, even though each network may not be able to model a hierarchical structure
- Two-level routing
  - Within each network is a single *intradomain gateway protocol*/*interior gateway protocol* for routing
  - Across all networks in an internetwork is a single *interdomain gateway protocol*/exterior gateway protocol
  - In the Internet, *BGP* (Border Gateway Protocol) is used as the interdomain gateway protocol
- Since each network operates independently, it can be called an *AS* (Autonomous System)
- Such an abstraction may still allow some non-optimal routing, but the operational freedom (and thus scalability) is hard to beat

### 5.5.5 Packet Fragmentation

- Each network/link may impose its own maximum packet size

- Ideally, hosts could know the path a packet will take to the destination, and thus the smallest packet size that satisfies all networks it must travel on. That smallest packet size is the *MTU* (Maximum Transmission Unit) of the path. This foreknowledge isn't always available
- The other solution is to break packets into fragments and recombine them later
- Transparent fragmentation
  - ¤ When a packet is fragmented, the fragments are all sent to the same exit router and recombined when they've all arrived there
  - ¤ Subsequent networks see no effects of the fragmentation
  - ¤ Problem: the exit router must know when it has all the fragments
  - ¤ Problem: Fragments' routes must travel to the same exit router, an extra constraint on routing
  - ¤ Problem: Routers must do a lot of buffering and calculation to keep track of and reassemble fragments. This may happen at multiple network boundaries per packet, which is even more costly.
- Nontransparent fragmentation
  - ¤ Once a packet is split, each fragment is treated as its own packet, to be reconstructed only at the destination
  - ¤ This requires fragments to be fortified with ordering information
  - ¤ IP
    - ∗ Every fragment packet shares the same packet number as the source, and additionally has a byte offset into the original packet's payload, and a flag indicating whether it is the last fragment or not
    - ∗ This permits successive fragmentations down to any fragment size, and correct reassembly if fragments arrive out of order
    - ∗ This also creates extra costs on the networks as many more packets spring into existence, and creates header overhead with each new fragmentation
- The modern Internet uses *Path MTU Discovery* to avoid fragmentation
  - ¤ Intermediate routers return oversized packets to the source with the maximum size they can handle. The source does its own fragmenting and tries again until it stops getting these messages.
  - ¤ This is costly too, as it creates heavy delays when a source and destination start using a new (or updated) path
  - ¤ A better solution may just be to truncate packets to maximum size at every intermediate router, so the destination learns the path MTU as fast as possible, while also getting some of the payload. The path MTU can be fed back to the source in acknowledgments.

## 5.6 The Network Layer in the Internet

- Top 10 design principles
  - ¤ **Make sure it works**: try it out before finalizing or standardizing
  - ¤ **Keep it simple**: fight features and leave out the non-essential
  - ¤ **Make clear choices**: only specify a single way of doing a thing, even if (political) pressures demand more
  - ¤ **Exploit modularity**: split functionality out into multiple modules (or better, a stack of layers) where possible
  - ¤ **Expect heterogeneity**: simple, general, and flexible designs are the only ones able to sit on many different underlying platforms
  - ¤ **Avoid static options and parameters**: where parameters are needed, make senders and receivers negotiate them. Never use fixed values.
  - ¤ **Look for a good design, don't try to make it perfect**: A good design will handle most needs; leave it to consumers with unusual needs to adapt their usage to suit the design
  - ¤ **Be strict when sending and tolerant when receiving**: Outgoing messages should be strictly compliant, try hard to understand incoming messages even if they're somewhat broken
  - ¤ **Think about scalability**: Resources are spread out, so load should be too. Avoid stipulating a monolothic or central authority.
  - ¤ **Consider performance and cost**: Disprefer elements that will be barriers to adoption
- The Internet is made of a collection of AS's
- *Tier 1 Networks* (major backbones) connect to ISPs, data centers and co-location facilities, and regional/mid-level networks

### 5.6.1 The IP Version 4 Protocol

- IPv4 datagram headers: 20 byte fixed-length part, optional variable-length part, in big-endian network byte order
  - Version (4 bits)
  - IHL (4 bits)
    - number of 32-bit words in the header (min 5, max 15). This limits the options to 40 bytes. Some options note the route the packet's taken, for which 40 bytes is way too small
  - Differentiated Services (8 bits)
    - This used to be used for type of service: 3 bits to signal priority, 3 to indicate which of {delay, throughput, reliability} was important
    - Nobody knew how to use those values, so IETF repurposed those 6 bits to mark service classes
    - Bottom 2 bits indicate congestion experienced by the packet
  - Total length (16 bits): total length of the dtaagram
  - Identification (16 bits): packet's ID (fragments of a packet will have identical values here)
  - unused (1 bit): the Evil bit
  - DF (1 bit): Don't Fragment
    - Formerly signaled the packet shouldn't be fragmented.
    - Nowadays, if true, then the sender is using Path MTU Discovery and wants the packet back if it **would** be fragmented
  - MF (1 bit): More Fragments: true for all but the last fragment of a packet
  - Fragment offset (13 bits)
    - Where in the packet this fragment belongs: all fragments but the last must be a multiple of 8 bytes
  - TTL (8 bits)
    - Intended to count seconds, but only reliably counts hops. When 0, the packet is discarded and the source is warned
  - Protocol (8 bits)
    - Which transport process to give the packet to. Numbering is the same globally and is managed by IANA.
  - Header checksum (16 bits)
    - All 16 bit halfwords are added together with one's complement math, then the one's complement is taken of that. This should be 0 on final arrival
  - Source address (32 bits)
  - Destination address (32 bits)
  - Options (variable length, not required): each option starts with a 1-byte identifier, and options are padded out to a multiple of 4 bytes. Original options
    - Security: Military could have used this to ensure secure packets aren't routed through untrusted territory.
    - Strict Source Routing: Prescribes a specific set of routers the packet must follow (no more, no less). Most useful for emergency packets when routing tables are corrupted, or for making timing measurements.
    - Loose Source Routing: Prescribes a specific set of routers the packet must pass through. Useful to prefer or avoid certain paths for political or economic reasons.
    - Record Route: Requests that every router append its address to the options field, for route debugging.
    - Timestamp: Requests that each router append its address and timestamp to the options field, for verbose route debugging.

### 5.6.2 IP Addresses

- IP addresses refer to network interfaces, not hosts
- Hierarchical structure
  - Top bits are variable-length network portion, lowest bits designate a host portion
  - Thus networks occupy contiguous blocks of IP address space with a common bit *prefix*.
  - Networks/prefixes are usually referred to by the lowest address in the block and the size of the block (number of bits in the network portion: $A.B.C.D/L$ means that the first L bits of the bit representation of $A.B.C.D$ identify the network.
  - A *subnet mask* is $L$ true bits followed by $32 - L$ false bits: an address ANDed with its subnet mask extracts only the network portion of the address

  - ¤ Advantage: routers outside of the destination network only need to care about the network portion, which shrinks their routing tables (from 4 billion addresses to about 300,000 networks)
  - ¤ Disadvantage: IP addresses must belong to a network, and routers must send packets through a network. This may result in poor routing for mobile devices which change locations
  - ¤ Disadvantage: Networks receiving and underutilizing allocated blocks result in wasted address space
- Network numbers are managed by ICANN
- Subnetting
  - ¤ Allowing a block of addresses allocated to a network to be split into multiple parts (for internal use as separate networks) while still behaving as a single network to the outside world
  - ¤ Within a network, subdividing the block among different logical networks
  - ¤ Router uses a subnet mask for each such internal network to determine which link to use to forward to that network
- CIDR - Classless InterDomain Routing
  - ¤ Exit routers at source networks needn't know what line reaches the destination address. Sending on to the ISP is a sane default.
  - ¤ Routers at ISPs and regional networks need a mechanism to contact all possible destinations, without having any such sane defaults.
  - ¤ Such core routers are said to be in a *default-free zone* of the Internet
  - ¤ Storing, propagating, receiving, and calculating routing information for every network on the Internet would be too heavyweight.
  - ¤ *Route aggregation*: a router can combine multiple networks' address blocks into a single larger address block so that a router needs only know general rules on how to contact a small number of *supernets*, and forget the smaller address blocks.
  - ¤ In the case that a supernet's larger address block (i.e., shorter prefix) contains address blocks for networks that should use one outgoing link, but some longer prefixes contained within that block should use other links, the *longest matching prefix* determines which of these links will be used to route a packet.
  - ¤ In this way, a very large supernet can save a lot of routing resources, without actually causing traffic to be diverted to the wrong link.
- Classful and Special Addressing
  - ¤ Before CIDR, classful addressing was in use
  - ¤ It implemented 4 classes of addresses, each with fixed properties and fixed length network portions and host portions
    - ∗ Class A: 0, 7-bit network portion, 24-bit host portion
      - > 16 billion addresses is unnecessarily big for most organizations
    - ∗ Class C: 110, 12-bit network portion, 8-bit host portion
      - > 256 addresses is too few for most organizations
    - ∗ Class B: 10, 14-bit network portion, 16-bit host portion
      - > Just right for most organizations, so everyone wanted one
      - > Exhaustion of class B networks was called the three bears problem
    - ∗ Class D: 1110, 28-bit multicast address
    - ∗ Class E: 1111, reserved for future use (so unspecified)
  - ¤ Routing in the system was very easy: determine class, mask off the host number to look up the network's outgoing line in a table (one table of a fixed size per class), and send
  - ¤ Special addresses
    - ∗ 0.0.0.0: used by hosts being booted
    - ∗ All 0's in the network number: used to refer to the current network (can be useful if the host doesn't know the network number, though the host must already know the network mask)
    - ∗ 255.255.255.255: All hosts on the network (used for local broadcast)
    - ∗ All 1's in the host number: used to broadcast to a remote network (many administrators block this as a security hazard)
    - ∗ 127.xx.yy.zz: Used for loopback: Packets to such addresses are processed locally and treated as incoming packets. This can be used to test a network interface when a host doesn't even know its own host number.
- NAT - Network Address Translation
  - ¤ An attempt to overcome IP address scarcity
    - ∗ When dialup was dominant, it was okay to just lend users an address for the duration of their connection

        - With always-on broadband connections, this model can't work
    - ISPs assign each customer a single IP address for IP traffic
    - Within customers' networks, each host gets its own IP address used for routing intramural traffic
    - Traffic exiting a customer's network is translated to the shared IP address
    - Inside customers' networks, only specific ranges of addresses are used: ranges that are marked as private in the IP spec, which is the only way we know they won't be used outside the customers' networks
        - 10.0.0.0 - 10.255.255.255/8 (16,777,216 hosts)
        - 172.16.0.0 - 172.31.255.255/12 (1,048,576 hosts)
        - 192.168.0.0 - 192.168.255.255/16 (65,536 hosts)
    - TCP/UDP packets leaving the NAT box get their Source Port replaced by a value which the NAT box can later use to map back to (NAT-internal IP address, original Source Port), and header checksums (TCP/UDP, then IP) are then recomputed
    - Problems with NAT
        - IP addresses no longer identify machines uniquely
        - end-to-end connectivity is broken: machines from behind a NAT box may be clients (initiate contact/connectivity), but they are otherwise not reachable (at least, not until the NAT box has mapped a port and a NAT-internal address to make it so)
        - NAT is connection-oriented: the NAT box's mapping table is a de facto connection; if it crashes, a file download will not be able to resume, which introduces fragility.
        - NAT only works by breaking layer independence
        - NAT fails for transport protocols other than TCP/UDP
        - Some applications use multiple TCP/IP connections or UDP ports, which NAT breaks by messing with the port numbers.

### 5.6.3 IP Version 6

- IPv4 had problems; to fix these IETF tried to find a new version with these properties
    - Support billions of hosts
    - Reduce the size of routing tables
    - Simplify the protocol (so packets can be sent faster)
    - Provide better security (authentication and privacy)
    - Better support for types of service (e.g., for realtime data)
    - Aid multicasting
    - Permit hosts to roam without changing address
    - Allow the protocol to evolve in the future
    - Permit the two protocols to co-exist during migration
- One proposal was to run TCP over CLNP
    - Since CLNP was based partly on IP, this wasn't so bad
    - But it was a no-go politically because CLNP came from the OSI camp
    - It was also poor on service types
- SIPP (Simple Internet Protocol Plus) was selected and renamed IPv6
    - 128 bit addresses (so $2^{128} \approx 10^{38}$ – plenty!)
    - Simpler header (7 fields versus 13)
    - Better support for options (so that we can make up for the missing header fields)
    - Security is built-in
    - Quality of Service
- The Main IPv6 Header
    - The fields
        - Version (4 bits)
            - Always 6 for IPv6, always 4 for IPv4
            - Checking wastes a few cycles, and some routers had been built that skipped the Type check (perhaps intending to rely instead on Ethernet, which indicated the same information in its Type field)

    - Differentiated services / Traffic Class (8 bits): the same as for IPv4: top 6 bits indicate delivery requirements, bottom 2 bits flag congestion experienced
    - Flow label (20 bits)
        - A flow (determined by the source address, the destination address, and this label) may be assigned special requirements (w.r.t. delay, bandwidth, etc.)
        - Intermediate routers try to reserve resources for flows, so the result is a pseudo-connection
    - Payload Length (16 bits): Number of bytes in the payload
    - Next header (8 bits): flags which of the extension headers follow the main header (if none, then this flags which transport protocol to hand the packet to)
    - Hop limit (8 bits)
    - Source address (128 bits)
    - Destination address (128 bits)
  - Addresses
    - Written as 8 groups, each of 4 hexadecimal digits
    - Since may addresses will stay unused for a long time, expect many 0's
    - Leading 0's can be omitted
    - Consecutive groups of 16 0-bits can be replaced by a pair of colons
    - IPv4 addresses can be written as a pair of colons and a dotted quad
    - $2^{128} \approx 10^{38}$ addresses is a lot: $7 \times 10^{23}$ per square meter on Earth's surface. Even so, the address space is not likely to be used efficiently, so better to overprovision than underprovision.
  - Fragmentation isn't done in IPv6
    - Hosts use MTU path discovery to determine the correct maximum packet size for the path, since this is more efficient, and leave packet splitting to the source host.
    - Meanwhile, the spec requires that routers be able to forward larger packets (the new minimum is up to 1280 bytes from 576 bytes)
  - Fields in IPv4 not in IPv6
    - IHL, because IPv6 uses a fixed-length main header
    - Protocol, because it's covered by Next Header
    - Checksum
        - Networks are more reliable nowadays
        - Data link and Transport layers tend to have their own checksums
        - Checksums have a high performance cost

- Extension Headers
  - Some headers have a fixed format; some have a variable number of options, each of variable length
  - All extensions begin with a 1 byte Next Header field, and a 1 byte Header Extension Length field (indicating the length of the header in bytes, not including the first 8)
  - Format of variable-length options: Each is a (Type, Length, Value) tuple
    - Type (1 byte)
        - Indicates which option it is
        - First two bits tell routers what to do with the packet if they don't know about the option: skip the option; discard the packet; discard the packet and send back an ICMP packet; discard the packet but send back no ICMP packets for multicast addresses
    - Length (1 byte)
    - Value (up to 255 bytes)
  - Defined extensions
    - Hop-by-hop header: defines info al routers on the path must examine
        - So far, only defined option is one to mark the packet as a jumbogram, with up to just under 4GB
    - Destination options header: not yet used, but reserved for options that only need to be used at the destination host
    - Routing header: A list of routers that must be visited in order along the path to the destination
    - Fragment header: Allows fragmenting at the source host by carrying a datagram identifier, fragment identifier, and a bit indicating whether more bits follow
    - Authentication header: Allows destinations to verify that the payload comes from the expected host

- Controversies
  - Hop limits were viewed by some as too small
  - Maximum packet size viewed as too small for some applications (supercomputers)
  - Checksums' removal reduces data integrity
  - Mobile hosts lack explicit support
  - Security
    - Security buffs won't deign to trust security provided by intermediate, potentially-untrusted hosts, and will avoid the options entirely
    - Export laws related to cryptography may limit adoption
- In theory, IPv4 networks would convert to IPv6 networks, and IPv6 islands would have gained traction as these IPv6 islands tunneled their traffic over IPv4
- In reality, IPv4 networks don't convert often, and the focus now is on dual-stack hosts and routers

### 5.6.4 Internet Control Protocols

- ICMP - the Internet Control Message Protocol
  - About a dozen different types of messages, passing operational information
  - Main message types
    - Destination unreachable
      - When a router can't locate the destination, or when a DF packet arrives at a small packet network
    - Time exceeded
      - When a packet is dropped because its TTL hits 0
      - traceroute uses this by sending out packets with TTLs of {1, 2, 3, ...} to discover which hosts send back time exceeded ICMP packets
    - Parameter problem
      - An illegal value was found in a header field
    - Source quench
      - Used to be used to signal to hosts that they were sending too many packets
      - No longer used: it's too little, too late; and adding packets to a congested network is counterproductive
      - Nowadays, congestion throttling is done in the transport layer by senders when detecting packet loss
    - Redirect
      - Sent back when a router notices that a packet seems to be routed incorrectly
    - Echo, Echo reply
      - Used to see if a destination is still alive (used by ping)
    - Timestamp request, Timestamp reply
      - Similar to echo/echo reply, but also returns arrival time of request and departure time of reply
    - Router advertisement, Router solicitation
      - Lets hosts find nearby routers
- ARP - Address Resolution Protocol (RFC 826)
  - Process by which senders determine what network interface on its network to send to to reach an IP address
  - Send out an ARP request (containing the destination IP and its own (IP, interface) mapping). The host having that IP will respond.
  - Values are cached, but time out every few minutes to detect configuration changes
  - *Gratuitous ARP*: Newly configured machines broadcast their own (IP, interface) mapping so it can be found by members of its own network
  - If a response unexpectedly arrives, they share the same IP address as some other host, and the issue must be resolved
  - Hosts will be configured to store the IP address of a *default gateway*–the network's exit router
  - Before a host can send out of the network, it needs to use the normal ARP mechanism to determine the router's interface from its IP address
  - Routers use the same procedure to discover the next hop to make
  - *Proxy ARP*: Routers can also be configured to respond to ARP requests for some hosts not on their network. This is usually done when a host wants to pretend to be on one network when it's actually on another

- DHCP - Dynamic Host Configuration Protocol
  - ¤ On a DHCP network, one host is responsible for disseminating IP addresses
  - ¤ When a network interface is started, it sends a DHCP DISCOVER packet (which the router may be configured to relay to the DHCP server)
  - ¤ The DHCP server will allocate an IP address and send it back to the host in a DHCP OFFER packet
  - ¤ IP addresses in DHCP are *leased*, with the assignments lasting a limited time only (to prevent exhausting all available addresses as hosts exit and re-enter the network). If hosts don't renew their leases before expiration, they lose their IP addresses
  - ¤ DHCP also carries lots of other network configuration parameters, like network mask, default gateway IP address, and the IP addresses of DNS and time servers

### 5.6.5 Label Switching and MPLS

- MPLS (MultiProtocol Label Switching) – used primarily by ISPs for routing within their network – attaches a label in front of the packet, using that (rather than destination address) to forward the packet
- MPLS is between network and data link layers
- When an MPLS packet arrives at an LSR (Label Switched Router), the label is used as an index into routing tables. The value here determines which outgoing line to use and what new label will replace the packet's current one.
- The main advantage of MPLS is that the header is very compact and easy to process; this speeds up routing for networks that are known to be homogeneous.
- MPLS labels only have local significance within a network; so they can only be added (and must be removed) when crossing a LER (Label Edge Router)
- Routers often group multiple flows into a single label (into a single FEC (Forwarding Equivalence Class)) if their QoS and destination are the same.
- This is different from Virtual Circuits because this only treats multiple flows as equivalent within the network: the flows' differing destinations addresses are retained as different within the IP packet.
- MPLS can also do something like route aggregation
  - ¤ If several flows are best off carpooling then separating to reach their different destinations, packets will get a common label followed by distinct labels. The common label will be used and replaced at routers along the common path, until it fails to designate instructions. Then it will be thrown away and next label (the one that distinguishes the flows) will come into play.
  - ¤ Generally speaking, packets get a stack with a variable number of labels.
- MPLS uses a series of control protocols and routing protocols to ensure that neighbors can use each others' labels and can route through one another

### 5.6.6 OSPF - An Interior Gateway Routing Protocol

- Intradomain Routing / Interior Gateway Protocol
- OSPF requirements / goals
  - ¤ Open (published, not proprietary)
  - ¤ Supports a variety of distance metrics
  - ¤ Dynamic (adapts quickly and automatically to topology changes)
  - ¤ Supports routing based on type of service
  - ¤ Load balancing
  - ¤ Supports hierarchical systems where routers don't know the entire topology
  - ¤ Secure, so it couldn't be thrown off by malicious routing information
  - ¤ Able to deal with routers connected via a tunnel
- Supports point-to-point links (e.g., SONET), broadcast networks, and even *multiaccess networks* (networks with multiple routers that can reach each other directly), which previous protocols dealt with poorly
- Abstracts the networks, routers, and links into a directed graph with weighted edges
  - ¤ Routers and networks are vertices
  - ¤ A point-to-point connection between two routers is an edge in each direction, potentially with different weights
  - ¤ A broadcast network (e.g. a LAN) is represented as 1 vertex with edges of weight 0 to and from each router/vertex it contains

  - Other networks have incoming edges but no outgoing edges (this represents the fact that these can't be routed through to reach destinations they don't contain: packets entering will the cost of travelling in the network without getting closer to the destination)
- Each router uses an ECMP (Equal Cost MultiPath) algorithm to calculate and store the shortest path(s) from itself to all destinations.
- ASes are divided into numbered *areas*–sets of mutually disjoint contiguous networks which aren't necessarily exhaustive.
  - Routers entirely in one area are *internal routers*
  - Routers in one area can know of destinations in other areas, but not the topologies
- Every AS has a *backbone area*, area 0, to which all areas are connected (even if by tunnels)
- *Area border router*: a router connected to $\geq 2$ areas
  - Summarizes the destinations in an area and sends the summary to all other areas it's connected to
  - The summary includes cost information so other areas can choose which area border router to use for each packet
  - *Stub area*: an area with only one border router. Stub areas it doesn't even need to receive these summaries, as inter-area traffic only has one border router to go through.
- *AS boundary router*: injects into the AS info on destinations in adjoining ASes
- Inter-area traffic always goes from one area to the backbone area then to the destination area, forcing traffic into a hub+spokes pattern
- LSR updates are issued to adjacent routers. When multiple routers are present on a broadcast network, one router is elected the *designated router*, and is said to be adjacent to other routers on the network. A backup designated router is always kept up-to-date as well.
- In essence, since the backbone area adjoins all areas, every backbone router is aware of all possible destinations
  - Its routing tables use the summaries of area boundary routers to construct shortest paths to all destinations
  - These routes are propagated back to area boundary routers, summarized there, and propagated into all other areas.
  - In this way, backbone routers decide the best routes to all destinations by using the summaries, and non-backbone routers discover the best routes by receiving the summaries of those decisions.

### 5.6.7 BGP - the Exterior Gateway Routing Protocol

- While intradomain protocols need only be concerned with routing, interdomain protocols need to enforce political routing policies
- Transit service
  - When a customer ISP pays a provider ISP to deliver packets to and from other destinations on the Internet.
  - Provider ISP advertises routes to all destinations on the Internet
  - Customer ISP advertises routes to the provider ISP only for destinations in its AS
- IXPs (Internet eXchange Points): facilities that many ISPs use to connect with one another.
- Peering
  - Two ISPs send traffic through one another for free to avoid the cost of using a provider ISP
  - Two peering ISPs will advertise to each other routes residing on their networks (but non-transitively)
- *Stub networks* keep connectivity through only one ISP. They don't participate meaningfully in interdomain routing
- *Multihoming*: when a network is connected to multiple ISPs – usually for increased reliability. A multihoming network will run BGP so other ASes can make informed decisions on the lowest-cost ISP (and thus path) to use to reach the network.
- BGP is based on Distance Vector Routing, with some differences
  - Distance is less important as a routing criterion than policy
  - BGP routers keep not just costs stored locally, but also the path, making it a *path vector protocol.*
  - BGP routers communicate with one another via TCP connections, which improves reliability
- Route advertisement
  - Routes from router $R_a$ would be advertised are triples (destination $D$, *AS Path* $A$, next hop router $R_a$) – indicating that for destination $D$, a path that crosses ASes in $A$ should be sent next to router $R_a$.
  - If a peer AS or provider AS propagates such an advertisement over a link from router $R_b$, it appends its own AS identifier to the AS Path, and replaced the next hop router with $R_b$.

  - The complete path of ASes is carried so the source router can detect routing loops: routers receiving a route with their own AS in the list of ASes ignore the route advertisement
  - Despite this, BGP does have a variant of the count-to-infinity problem, where routes are slow to converge and may briefly have loops
- Keeping a list of ASes as a path is very nongranular, but ASes may not wish to expose internal paths or costs
- In addition to external BGP, internal BGP (iBGP) is used to propagate BGP routes throughout an ISP's interior when multiple routes are advertised to it
- Common strategies for GBP to choose a route for a given destination (when multiple routes are advertised to it)
  - Money: Customer routes are given highest preference (because this earns money), and peered routes are given next preference (because this doesn't cost money)
  - Fewer ASes: A path with fewer ASes is chosen. This is a bit of a hack, since number of ASes is a traversed is a weak proxy for more meaningful metrics like delay
  - *Early exit routing / Hot potato routing*
    - Routes are chosen to take the lowest-cost route through the AS
    - This has a high chance of producing very different routes from host $A \rightsquigarrow B$ than from $B \rightsquigarrow A$

### 5.6.8 Internet Multicasting

- IP supports multicasting with class D addresses: 224.0.0.0/28
- Within that range, 224.0.0.0/24 is reserved for multicasting on the local network
  - These packets are broadcast to all hosts on the LAN, with some hosts ignoring them
  - Special addresses
    - 224.0.0.1: all systems
    - 224.0.0.2: all routers
    - 224.0.0.5: all OSPF routers
    - 224.0.0.251: all DNS servers
- *IGMP* - Internet Group Management Protocol (RFC 3376)
  - Manages membership in multicast groups when multicasting outside of a network
  - Permits hosts to request to join or request to leave a group
  - Each host must remember which groups it belongs to
  - Once a minute, each multicast router asks hosts on its LAN which groups they belong to (this request is sent to 224.0.0.1); each host responds back with the set of class D addresses they are subscribed to
  - Within an AS, the main protocol for building multicast spanning trees is PIM (Protocol Independent Multicast)
    - Dense Mode PIM
      - Creates a pruned reverse path forwarding tree
      - Good when members are ubiquitous within the AS
    - Sparse Mode PIM
      - Creates a core-based tree
      - Best when subscribers are relatively few and are spread out
    - Source-specific Multicast PIM
      - Optimal when the multicast group has only one sender
- For multicasting outside of an AS, one must tunnel or use (standardized) extensions to BGP

### 5.6.9 Mobile IP

- Overview
  - Every site permitting users to roam has to contain a *home agent*
  - A mobile host at a foreign site gets an IP address there (its care-of address), and notifies the home agent of it
  - Packets arriving for a mobile host that is elsewhere arrive at the home agent, which tunnels them to the mobile host
- This solution meets most goals of roaming, except incurs the high cost of sending packets on lengthy detours for mobile hosts

    - While CIDR's longest-prefix matching is theoretically capable of dealing with a small number of mobile hosts without detours, too many mobile hosts causes too much growth in routing tables (which defeats the purpose of CIDR's route aggregation in the first place)
    - Changing the IP addresses of the mobile hosts can also work to fix the detour problem, but it breaks other things. It means that IP addresses stop uniquely identifying the host, which breaks some things on the application layer until reconfigured.
    - 802.11, meanwhile, handles some mobility at the link layer–but only as long as the host stays on the same access point.
- IPv4's model of mobility (RFC 3344)
    - Mobile hosts listen for ICMP router advertisements (and/or send periodic solicitations) to discover which router is nearest, and use this to discover if they've changed networks
    - To acquire a care-of address, the mobile host either uses DHCP, or registers with a foreign agent operating on the foreign network
    - When the mobile host is registered with the home agent as being away, the home agent uses proxy ARP (packets using ARP to find the mobile host are replied to by the home agent with its own Ethernet address)
    - When the mobile host leaves or arrives, it or the home agent sends itself a gratuitous ARP request with the right mapping, so that the router will see it.
    - Forwarding is handled by tunneling the IP packet within IP
- Problems
    - Tunneling mobile IP over IP naively used to break NAT: NAT expected to peek into the TCP/UDP packet inside the IP packet, but only expected to have to go one IP header deep to do so
    - Some ISPs do *ingress filtering*, a check that the source address of an incoming packet "is coming from the right direction", to minimize malicious spoofed traffic
        - This is especially problematic for traffic from the mobile host which uses its IP address, since packets will appear to be originating from the wrong place. The best solution is more detours: the mobile host tunnels traffic from its care-of address to its home agent, which then forwards the packets on to the intended destination.
    - Home agents also need a mechanism to ensure that only the mobile host itself can register itself as way (to avoid other hosts installing themselves as a Man-in-the-Middle on the mobile host's traffic)
- IPv6 supports route optimization for mobile hosts so that tunneling is only incurred on the first packet/first few packets
- Mobile Networks (e.g., a wifi network on an airplane) are also supported in IPv6

### 5.7 Summary

# 6 The Transport Layer

- The transport layer receives data from a process on the source host, uses the network layer to get packets to the destination host, and then must get the packets' data to a process on the destination host.

## 6.1 The Transport Service

- *Transport Entity*: the software/hardware used at the transport layer
- Transport layer services are very similar to network layer services
- Why make transport layer and network layer distinct?
    - Transport is meant to run entirely on users' machines; network layer runs on routers in between
    - So transport is responsible for smoothing over situations where the network is somehow inadequate (e.g., losing or damaging packets), whereas the network layer represents real networks, warts and all.
    - If a network connection dies, a transport entity which is decoupled from the network entity can remain in service to establish a new network connection without having to forget all state related to the transmission.
    - Transport isolates applications from the heterogeneity and flaws of networks
- Layers 1-4 are sometimes called the *transport service provider*, upper layers the *transport service user*

### 6.1.1 Transport Service Primitives

- It's most worthwhile to focus on connection-oriented/reliable transport services, since unreliable (datagram) services offer little but a thin wrapper over the network layer
- Since applications/users use the transport layer's services, the service must be easy to use
- Messages from one transport entity to another are called *segments* or *TPDMs* (Transport Protocol Data Units)
- Example: server wth $n$ remote clients
  - Server executes LISTEN primitive that causes it to block until a client arrives
  - Clients execute a CONNECT primitive (which blocks the caller and sends a packet to the server, this packet's payload being a transport layer message to the server's transport entity)
  - The segment sent by the client to the server is a CONNECTION REQUEST primitive
  - If the server was blocked on a LISTEN, then it replies to the client with a CONNECTION ACCEPTED. The client unblocks and the connection is now established
  - Either party can block in a RECEIVE state in order to wait for the other party to SEND. An arriving segment unblocks it, and it has a chance to process the segment and reply.
  - This blocking model is fine as long as both transport entities can keep track of whose turn it is to send.
  - Asymmetric disconnection
    - Either transport user issues a DISCONNECT; once it arrives at the other side, the connection is released (like with telephones)
  - Symmetric disconnection
    - Either transport user issues a DISCONNECT, meaning it has no more data to send over the connection
    - But the connection isn't released until both sides have acknowledged they're ready to disconnect.

### 6.1.2 Berkeley Sockets

- First released in BSD 4.2 in 1983 (windows implements them with winsock)
- Primitives
  - SOCKET
    - Creates a new endpoint and allocates table space for it within the transport entity
    - Parameters specifying addressing format, type of service, and protocol
    - On success, returns a file descriptor
  - BIND
    - Assigns a network address to a socket
    - This is separate from SOCKET because some processes care a lot about their address and some don't care at all (clients never need one, for example, so they don't bind an address)
  - LISTEN
    - Allocates space to queue incoming calls, should different clients try to connect concurrently
  - ACCEPT
    - Blocks to wait for an incoming connection
    - Incoming segments asking for a connection are given a new socket cloned from the original one so that the connection can be handled by a child process/thread
  - CONNECT
    - Blocks the caller to establish a connection
  - SEND/RECEIVE
    - Both sides can use these – a socket is full-duplex
    - UNIX READ and WRITE can be used instead if the special options of SEND and RECEIVE aren't needed
  - CLOSE
    - Connection release is symmetric
- TCP and sockets form a de-facto standard for connection-oriented services, referred to as a *reliable byte stream*
- Other uses of sockets
  - In use with a connectionless transport service
  - With expanded sets of calls (e.g. SENDTO and RECEIVEFROM) that allow applications to work with multiple transport peers

    - In combination with other transport protocols that lack congestion control or send message streams rather than byte streams

- Sockets are imperfect. e.g., web browsers are an application that works with a group of related streams – they may request many resources concurrently from the same transport entity, but each request will get a different socket, meaning the overhead for all the requests is applied independently for each (rather than over the group)

### 6.1.3 An Example of Socket Programming: An Internet File Server

## 6.2 Elements of Transport Protocols

- Transport protocols, like data link protocols, need to deal with error control, sequencing, and flow control. But:
    - Transport protocols communicate over not a physical channel, but an entire network
    - This means addressing is required (so that routes can be chosen at the network layer)
    - Establishing connections is far more complicated, since the other end maybe reachable in many ways (rather than 0 or 1)
    - Networks (via delay, or bad routing) may store packets, making duplicate, late, and out-of-order packets possible
    - Quantities are different: At the transport layer, variability in number of connections, bandwidth, etc., make allocating separate resources per connection too costly (e.g., think of the number of hosts on the internet versus the number on a particular LAN).

### 6.2.1 Addressing

- *TSAP*s (Transport Service Access Points): transport addresses to which processes listen for connection requests.
- On the Internet, TSAPs are referred to as *port*s
- How does an application know which TSAP it should contact
    - Some TSAPs/ports may be reserved (whether de facto or de jure) for specific applications. e.g., /etc/services
    - Otherwise, a *portmapper* (running on a fixed/known TSAP) would convert a service name to a TSAP
- *Initial Connection Protocol*
    - Rarely used applications would register with a *process server* (such as inetd) rather than sit listening all day
    - The process server listens to a range of ports for whichever services have registered with it
    - When a connection request lands at the process server, it spawns the requested server (creating it on-demand in a way that it inherits the existing connection)

### 6.2.2 Connection Establishment

- A major problem at the transport layer is assuring that delayed duplicate packets are rejected as duplicates.
- Bad solution: use throwaway TSAPs
    - This requires that addresses are never reused once their connection has been released
    - This makes it too hard to connect in the first place
- Bad solution: each connection gets a unique identifier
    - Initiating party proposes a connection identifier, and both parties set a (peer transport entity, connection identifier) pair as unusable for the future
    - This requires every transport entity to be able to store an indefinite history of such pairs, such that a pair is never lost (even if the entity crashes). This is impossible.
- Instead, take packet lifetime into account
- Basic strategies
    - Restricted network design
        * Requires delay to be bounded when congestion is taken into account and guaranteeing that packets don't loop
    - Timestamp each packet
        * Requires routers to agree on how old is too old, and to have synchronized clocks
    - Put a hop counter in each packet

- Hop counts are a good enough proxy for time.
- We also need to ensure not only that packets are dead, but also all acknowledgements to it
- Use $T >$ the true maximum packet lifetime, multiplied by some value greater than 1 (to be conservative)
- $T$ seconds after a packet's been sent, we can be sure to hear nothing more of it or its acknowledgements
- The method: label segments with sequence numbers that won't be reused within $T$ seconds, with the set of sequence numbers bounded in size by $T$ and the rate of packets/second
- To deal with machines crashing, we use a clock at each host that needs to fulfill some criteria
  - The clocks needn't be synchronized
  - The clock must be a binary counter with more bits than the largest conceivable sequence number
  - A clock increments at uniform intervals
  - The clock continues to run even if the host goes down
- Connection requests use the clock's low-order $k$-bits as sequence number, guaranteeing the sequence numbers won't be reused.
- With a connection having been established, sliding window protocols can be used to discard duplicates of accepted packets
- Problems arise if segments are sent faster than the clock ticks, because then there will be duplicates. This favors short clock ticks. The same is true if transport entities resume sending after a crash before the clock has incremented.
- Meanwhile, if the clock ticks too fast relative to the sequence numbers (i.e., if the segment rate is too much less than the clock rate), the clock will wrap around too fast
- For clock rate $C$ and sequence number space of size $S$, $S/C > T$ prevents overly fast sending.
- *Three-way handshake* for establishing connections
  - overview
    - host 1 sends a CONNECTION REQUEST to host 2 with sequence number: CR(seq=x)
    - host 2 replies to host 1 with an ACK segment containing x and announcing its own sequence number y: ACK(ack=x,seq=y)
    - host 1 acknowledges y in the first data segment it sends: DATA(data,ack=y)
  - Delayed duplicates
    - A delayed CR with x arrives at host 2 without host 1 knowing
    - host 2 replies to host 1 with an acknowledgement containing x and y
    - host 1 has no memory of x, so it issues a REJECT segment to host 2
  - TCP uses the three-way handshake (more or less)
    - Within a connection, PAWS (Protection Against Wrapped Sequence numbers) adds a timestamp to a 32-bit sequence number, to ensure that they don't wrap during the maximum packet lifetime
    - TCP doesn't use the clock-based scheme. Timing attacks could be used by an attacker to determine the next initial sequence, and use this to forge a connection. In practice, TCP instead uses PRNs and tries to make sure that they don't repeat within an appropriate interval.

### 6.2.3 Connection Release

- In the case of asymmetric release, there is a problem when one host sends a disconnect request before the other is done sending data.
- Symmetric release suffers from the *two-army problem*, where neither host can be truly sure not only that the other is ready to disconnect, but also that the other host knows it is ready to disconnect; thus a disconnect is never fully immune to data loss
- A three-way handshake gets most of the way there
  - host 1 issues a DISCONNECT REQUEST to host 2 and starts a timer
  - host 2 replies to host 1's DR with its own DR, and starts its own timer
  - If host 1 receives host 2's DR before its timer expires, it replies with an ACK and releases the connection.
  - If host 2 does not receive host 2's DR before its timer expires, it assumes host 2 never received its own DR and tries re-sending
  - If host 1 times out $N$ times (for some $N$), it will drop the connection anyway
  - If host 2 times out without receiving host 1's ACK, it will drop the connection
- The greatest weakness with the three-way handshake disconnect is if the initial DR can't get through on any of the $N$ attempts: in this case, the other host will have a half-open connection.

- This isn't so bad if transport entities keep a timer per connection that's reset with every segment received, and drop connections whose timer expires
- In the worst case, this requires dummy segments to keep otherwise idle connections open.

### 6.2.4 Error Control and Flow Control

- Recap of data link layer solutions to these issues
  - CRCs or checksums to detect errors
  - ARQ: each frame has a sequence number and is retransmitted unless ACKed
  - Sliding window: Senders have a maximum number of outstanding frames, which prevents them from overwhelming receivers. Larger windows improve pipelining on fast links, short windows reduce the need for buffers at sender and receiver. Stop-and-wait uses a window of size 1 to guarantee in-order delivery.
- The *end-to-end argument*
  - Protections offered by the data link layer on the wire don't prevent corruption in routers
  - End-to-end checks therefore must happen at the transport layer to guarantee correctness.
- Small sliding windows are more problematic in the transport layer than at the data link layer because segments must travel the whole length of the network rather than over one link (higher bandwidth-delay product)
  - Low bandwidth-delay product on 802.11 means that the link can only store one frame at a time, because distances are short and physical medium is fast: having a window size greater than 1 frame would introduce complexity without improving performance
  - High bandwidth-delay product implies a long roundtrip time between sending a segment to receiving an acknowledgement, so a short window would cripple performance
- Careful buffering is needed: at sender for all the unacknowledged packets, and at the receiver for out-of-order segments (due to larger windows and network delivery aberrations)
- Tradeoffs between buffering solutions depend on type of traffic
  - Low-bandwidth bursty traffic like an interactive terminal works well with dynamic allocation of buffer space at both ends
  - For high-bandwidth traffic, sender and receiver are best off allocating fixed blocks of buffer space
- Buffer size
  - A pool of identically sized buffers (sized for the largest expected segment) wastes space on small segments
  - Variable-size buffers allocated as needed from a fixed sized block of space would give good utilization, but be complicated to implement (e.g., avoiding fragmentation)
  - Using a single large circular buffer per connection is elegant, but only ideal when connections are heavily loaded
- Buffer allocation should be dynamically negotiated between transport entities
  - Buffers can be allocated per connection, or for all connections the two hosts share.
  - Receivers (knowing their own buffer status, but not the offered traffic) can announce how much space is free so the sender can adjust its window appropriately.
  - Specifics
    - Sender requests a certain number of buffers during CR
    - Receiver responds with its allocation, sender remembers this
    - Every segment sender transmits, it decrements the number of free receiver buffers it knows about
    - Every ACK the sender receives, it sets the number of free receiver buffers to the value indicated.
    - The sender stops sending if the number of free receiver buffers is 0
    - Receiver's responses to data acknowledge the highest sequence number seen, as well as the current number of free buffers allocated to the sender. This allows receivers to increase/decrease allocated buffers dynamically (as number of connections fluctuates)
    - Control segments are periodically resent in case the sender is blocked waiting to send and hasn't heard a set of missed ACKs
- Buffer space is usually not a big deal anymore. The new bottleneck is the carrying capacity of the network.
  - If adjacent routers can exchange at most $x\frac{packets}{sec}$ and a pair of hosts has only $k$ disjoint paths between them, they can only exchange $kx\frac{segments}{sec}$
  - Senders sending too much traffic will congest the network

  - If the network can handle $c\frac{segments}{sec}$ and the roundtrip time (for everything, including acknowledgements, queuing, and delays) is $r$, the sender's window should be $cr$.

### 6.2.5 Multiplexing

- Multiplexing: when several transport connections must share a single network connection
- Inverse multiplexing: when a user wants more bandwidth and has $k$ network connections, traffic can be distributed round-robin among them.

### 6.2.6 Crash Recovery

- For some purposes (e.g., large file transfer) it's desirable to resume a connection if one host crashes and comes back relatively quickly.
- Figure 6-18 shows that no matter what sender and receiver strategies are for dealing with crashes, there are scenarios when data is lost or duplicated
  - Two receiver strategies: ACK then Write; Write then ACK
  - Four sender strategies: always retransmit the last segment; never retransmit the last segment; retransmit only if 1 segment is outstanding; retransmit only if no segments are outstanding
- In general, recovering from a layer $N$ crash can only be done in layer $N + 1$.

## 6.3 Congestion Control

- Though routers are the first to discover congestion, it's best reduced by transport entities

### 6.3.1 Desirable Bandwidth Allocation

- Goal: Good performance (available bandwidth is used without congestion) and fair for competing transport entities
- Efficiency and Power
  - Goodput rises mostly linearly with offered load until the onset of congestion (when burstiness starts to cause delays)
  - The load for a transport entity to put onto the network is the load which yields the highest *power* (for $Power = \frac{Load}{Delay}$)
- Max-Min Fairness
  - Even networks using Differentiated Services still leave individual flows within a class to compete with one another for bandwidth
  - Fair division is hard to discuss meaningfully if flows have different but overlapping network paths. Or even if different connections with the same endpoints use different routes that have different capacities.
  - An allocation is *max-min fair* iff the bandwidth given to one flow can't be increased without decreasing the bandwidth given to another flow that has an equal or lesser allocation.
  - Min-max allocations can be computed with global knowledge of the network; or, if all flows ramp up and back off at an equal rate they can also share spare capacity politely.
  - Fairness can be measured in many different was: at the level of connections; at the level of connections shared per host pair; or all connections a host knows about
- Convergence
  - The ideal operating point of the network varies over time
  - A good congestion control algorithm converges quickly without being so twitchy that it can't settle if the network is stable.

### 6.3.2 Regulating the Sending Rate

- Ways the sending rate is arrested (note: these look similar at the sender, but arise from different problems and require different solutions)
  - Flow control; for when receivers have insufficient buffer space for the offered load

- Congestion control; for when the network has insufficient capacity for the offered load

- Feedback from the network may be explicit or implicit, precise or imprecise
- Control laws: rules for determining how to enact an increase or a decrease of rates
- *AIMD* (Additive Increase, Multiplicative Decrease)
  - Chiu and Jain modeled this with 2 competing users on an xy-axis to demonstrate that it is good for efficiency and fairness, with $x$ being the allocation to one of the users and $y$ the allocation to the other user.
  - $y = x$ is the line of fairness, where both senders have equal allocations
  - $y = N - x$ is the line of efficiency – for $N$ being the highest total load just before the onset of congestion – along this line, the network is operating efficiently.
  - At any time, the allocation lies at some point on this graph.
  - If users don't get a congestion signal, there's an additive increase. At the next time, the allocation will move a small amount at a 45-degree angle.
  - If users do get a congestion signal, there's a multiplicative decrease. This moves the allocation toward the origin. This reduces the bandwidth allocated for each and reduces the difference between their allocations.
  - TCP uses AIMD because of this efficiency & fairness argument, and also because it aggressively mitigates congestion.
  - Its fairness is biased, though, since different hosts get congestion feedback at different rates (due to different roundtrip times)
  - TCP changes its window size $W$ with respect to the observed roundtrip time $RTT$ – $\frac{W}{RTT}$ bounds its sending rate
- Different transport protocols also need a way to cooperate on congestion control. Since TCP is dominant, new protocols have the burden of adapting to it (and some get labelled *TCP-friendly.*

### 6.3.3 Wireless Issues

- TCP with AIMD breaks on wireless because it uses packet loss as a positive signal for congestion.
- Throughput on AIMD goes up with the inverse square of packet loss; such that 10% loss (which is low or normal for wireless) makes the connection stop working
- To work well, only packet loss induced by insufficient bandwidth should be signal congestion.
- One solution is to retransmit at the data link layer to mask losses below the network layer: 802.11 uses a stop-and-wait protocol with retransmissions to reduce these kinds of transient transmission errors
- Lossy links with long roundtrip times (e.g., satellites) can't afford the time cost of retransmission. For these, typically Forward Error Correction is used, or transport protocols which use some other signal for congestion control.
- Capacity variability also changes much faster on wireless networks as nodes move and as noise levels change. Congestion control algorithms are already able to deal with capacity that changes with usage, so this only poses a problem if the network's overall capacity is very volatile.

## 6.4 The Internet Transport Protocols: UDP

### 6.4.1 Introduction to UDP

- User Datagram Protocol, RFC 768
- Header
  - Source port, Destination port (16 bits apiece)
  - UDP Length (16 bits): number of bytes in the segment, minimum 8 (length of header), maximum 65,515 (maximum IP packet size)
  - UDP Checksum (16 bits) (optional): The checksum is calculated to include an IP pseudoheader, which helps a little to detect misdelivered packets
- All UDP offers is
  - A lightweight interface to IP
  - Process multiplexing (a.k.a., ports)
  - Optional end-to-end error detection

### 6.4.2 Remote Procedure Call (RPC)

- Client (calling procedure) sends a request to server and blocks
- Client (calling procedure) sends a request to server (called procedure) and blocks waiting for a response
- *Client stub*, *Server stub*: procedures on each machine which represent the remote call as a local one
- Client stub packs the parameters into a message (*marshaling*), which the OS sends to the server stub. The server stub unmarshals the parameters and calls the server procedure with them.
- Problems with RPC
  - Pass-by-reference won't work: side-effects can only be modelled if the client and server stubs are willing to do a lot of replacements to passed structures and deep copies.
  - Underlying assumptions about marshalled parameters may differ (e.g., if one host uses null-pointers to terminate lists and the other doesn't)
  - Inferring parameter types isn't always easy or possible
  - Shared state is not shared
- UDP is alright for RPC, but there are problems with the architecture
  - Additional timers are needed so clients don't hang forever on a lost request/response (and can reissue the request)
  - In such a case, servers may need to be clever about non-idempotent procedures (receive one request, send a response which is lost, receive the same request again)
  - Large requests/responses might not fit into datagrams
  - Concurrency requires identifiers to match request/reply pairs.

### 6.4.3 Real-Time Transport Protocols

- RTP (RFC 3550)
  - A generic, application-independent protocol that bundles multimedia data streams, providing transport service from the application layer
  - Each packet has a sequence number, and this is used by the upstream application to decide how to cope with missing values (e.g., skip a video frame, approximate missing audio samples)
  - Retransmission and acknowledgement are not offered
  - RTP payloads can contain multiple samples, each belonging to a profile (each profile offering multiple formats)
  - Samples contain timestamps
    - This allows clients to buffer upcoming samples received early
    - Allows multiple streams to be synchronized
  - RTP header
    - Version (2 bits)
    - P (1 bit): indicates if the packet was padded (to a multiple of 4 bytes). If so, the last byte of padding indicates how many bytes of padding were added.
    - X (1 bit): indicates presence of an extension header
    - CC (4 bits): number of sources present
    - M (1 bit): an application-specific marker bit
    - Payload Type (7 bits): indicates the encoding algorithm used
    - Sequence Number (16 bits): To detect gaps that imply lost packets
    - Timestamp (32 bits): As above: reduces jitter at receiver by decoupling playback time from packet arrival time
    - Synchronization Source Identifier (32 bits): Identifies which stream the packet belongs to
    - Contributing Source Identifiers: Identifies other streams being mixed, in the case that the SSI is a mixer.
- RTCP - the Real-Time Transport Control Protocol
  - Handles feedback, synchronization, and user interface, but does not transport media samples, so that the source can use this feedback to change data rates or encodings
  - RTCP feedback reports are sent to all participants. This is potentially a large amount of traffic for multicast groups, so RTCP scales its report rate to consume no more than some small amount of bandwidth (which it does by listening to other RTCP reports from its group and the amount of bandwidth those other participants are using)
  - RTCP handles interstream synchronization (in case different streams' clocks have different granularities or drift rates)
- Playout with Buffering and Jitter Control

- ¤ Jitter: variation in delay
- ¤ Buffering will smoothe over jitter, but long buffers don't make sense for live applications (e.g., videoconferencing needs to be responsive.
- ¤ The *playback point* – how long to wait at the receiver for media before playing it out – depends on how much jitter is present
- ¤ Measuring jitter
  - ∗ Look for differences between the RTP timestamps and arrival times, to give a sample of the delay
  - ∗ This allows the playback point to be adjusted dynamically. If not done well, this produces user-visible glitches.
  - ∗ One way to avoid this (with audio) is to adjust the playback point during moments of silence in audio (between *talkspurts*). The M marker bit in RTP is used to mark a new talkspurt
- ¤ But, of course, sometimes absolute delay, or jitter, is just too high for a certain application.

## 6.5 The Internet Transport Protocols: TCP

### 6.5.1 Introduction to TCP

- Senders and receivers use sockets (each representable as an (IP address, port) pair). Multiple connections may use the same sockets.
- Any port ¡ 1024 is a *well-known port* and is reserved for standard services
- TCP is a byte stream, not a message stream
- TCP by default is allowed to buffer data into batches before sending
  - ¤ Data sent through TCP with a PUSH flag (TCP_NODELAY) is requested to be sent immediately, rather than batched
  - ¤ Semi-deprecated URGENT flag is specified but is not in use

### 6.5.2 The TCP Protocol

- Every byte on the connection has its own 32-bit sequence number. This set of sequence numbers used to take a week to cycle through, but faster connections reuse them much more quickly
- TCP segments contain: a fixed 20-byte header, optional extra headers, and $\geq 0$ data bytes.
- Segment size is limited by the maximum IP payload (65,515 bytes) and the MTU of the link adjacent to the sender

### 6.5.3 The TCP Segment Header

- Fields
  - ¤ Source port, Destination port (16 bits apiece)
  - ¤ Sequence number (32 bits)
  - ¤ Acknowledgment Number (32 bits): Specifies the next in-order byte expected
  - ¤ TCP Header Length (4 bits): The cumulative length of the header, aka the offset at which the payload begins
  - ¤ unused space (4 bits)
  - ¤ CWR (Congestion Window Reduced) (1 bit) and ECE (Explicit Congestion Echo) (1 bit) are used for ECN
    - ∗ ECE indicates to a sender that congestion was experienced.
    - ∗ CWR acknowledges that the sender has reduced its window size and the receiver can stop notifying about congestion
  - ¤ URG (1 bit): indicates that urgent data can be found at the offset specified by the Urgent Pointer field
  - ¤ ACK (1 bit): if 0, the acknowledgment number is ignored
  - ¤ PSH (PUSH) (1 bit): Indicates the data should not be sent batched
  - ¤ RST (RESET) (1 bit): Indicates a problem. It's used to reset the connection, refuse to open a connection, or reject an invalid segment.
  - ¤ SYN (SYNCHRONIZE) (1 bit): Requests a connection
    - ∗ SYN=1 denotes both CONNECTION REQUEST and CONNECTION ACCEPTED, and ACK=1 distinguishes a segment as the latter.
  - ¤ FIN (FINALIZE) (1 bit): Indicates the sender has no more data to transmit
  - ¤ Window Size (16 bits): Tells how many bytes may be sent to this sender (as few as 0 if it has a backlog)
  - ¤ Checksum (16 bits): The same as for UDP (even with an IP pseudoheader)

- ¤ Urgent Pointer (16 bits)
- ¤ Options ($\geq 0$ 32-bit words, padded with 0's), each encoded as a (Type, Length, Value) triple.
  - ∗ Allow hosts to specify a *Maximum Segment Size* (MSS) during conection setup
    - > Used to amortize cost of fixed headers over larger payloads
    - > Default MSS is 536 bytes (all Internet hosts are required to accept 556 byte packets (mz: packets, or segments? both use 20 byte headers))
  - ∗ *Window scale* option allows sender and receiver to negotiate a larger window size
    - > 16-bit window size limits the number of bytes pending acknowledgment, which is costly for high-bandwidth and/or high-delay lines
    - > This option lets window size be set to a multiple of the value in the Window Size header field (valid multiples are $2^n$ for $2 \leq n \leq 14$)
  - ∗ *Timestamp* option carries a timestamp from the sender, and echoes it back from the receiver
    - > After this option is negotiated during connection setup, timestamps will be included in every segment sent
    - > These timestamps are used to compute roundtrip time samples
    - > These are also used as a logical extension of the 32-bit sequence numbers (permitting PAWS to discard wrapped sequence numbers)
  - ∗ *Selective ACKnowledgment* (SACK) permits receivers to acknowledge ranges of sequence numbers received (in addition to the lowest non-received sequence number) to reduce overall congestion in lossy networks.

### 6.5.4 TCP Connection Establishment

- Uses a three-way handshake
- In order to avoid delayed duplicates, initial sequence numbers need to cycle slowly
- But this creates a burden on hosts to remember sequence numbers they issue when accepting a connection request.
- A *SYN flood* is an attack which initiates so many handshakes at a listening host that it exhausts its memory trying to hold onto all the sequence numbers
- One fix is *SYN cookies*
  - ¤ A host responding to a SYN will construct a variable sequence number in a way that can be reconstructed if the third handshake happens
  - ¤ It generates $f$ by encrypting the other host's IP address and port number using a key only it knows (mz: and, presumably, changes periodically, to avoid wraparound problems?)
  - ¤ When the handshake completes, it receives an acknowledgment of $f + 1$, from which it can verify the correctness of the sequence number without storing it.

### 6.5.5 TCP Connection Release

- FIN segments start releasing the connection
- After a FIN is issued, a timer (with an interval to two maximum packet lifetimes) ensures the connection is released even if no ACK segment comes.

### 6.5.6 TCP Connection Management Modeling

- Fig 6-38: 11 states for TCP Connection Management
- Fig 6-39: Finite State Machine for TCP Connection Management

### 6.5.7 TCP Sliding Window

- TCP decouples acknowledging that segments were received from acknowledging the receiver's buffer allocation
- Some data can be sent when window size is 0
  - ¤ Urgent data may be sent (e.g., to issue a Ctrl+C to a remote terminal)
  - ¤ A *window probe* is a special segment requesting that the receiver announce its window size and the next sequence number expected

- Sending data and acknowledgments the moment they arrive from the application layer is extremely undesirable for some applications (e.g., telnet)
- *Delayed acknowledgments*: Delaying ACKs and window updates by up to 500 msec might give enough time for other data to show up (and reduce the number of control-only segments)
- *Nagle's Algorithm*, good for reducing the number of short segments
  - ¤ When data comes in small portions, send one portion and buffer the rest until the first portion is acknowledged (or until enough is buffered to send a segment of maximum size)
  - ¤ Nagle's algorithm causes burstiness that's not acceptable for some applications (highly responsive/interactive ones)
  - ¤ It can also result in inefficient temporary deadlocks if the receiver is using delayed acknowledgments: the receiver will be stuck waiting for data on which to piggyback an acknowledgment while the other will be waiting for an acknowledgment before sending more data
  - ¤ TCP_NODELAY disables Nagle's algorithm
- *Silly window syndrome*: When receiving application consumes data regularly but slowly, the receiving transport entity would end up announcing small window sizes frequently, causing many small segments to be exchanged relative to the total amount of data
  - ¤ Clark's solution: receiver won't announce window size until it can handle either the maximum segment size it advertised during connection setup, or until it's buffer is half empty, whichever is sooner
  - ¤ The sender can also help this by not sending tiny segments
- Clark and Nagle's solutions are attempts to boost efficiency by preventing senders from sending tiny segments, and by preventing receivers from asking for tiny segments.

### 6.5.8 TCP Timer Management

- *Retransmission TimeOut* (RTO): When a segment is sent, a timer is started. If the timer goes off before the segment is acknowledged, the segment is retransmitted
  - ¤ In the transport layer, delay variance is much higher (than in data link layer), so timer durations are harder to tune. Overly long intervals cause long delay for lost packets, overly short intervals waste bandwidth with unnecessary retransmissions.
  - ¤ The solution is a dynamic algorithm that adapts the interval to observed network performance.
  - ¤ Each connection keeps a variable, SRTT (Smoothed RoundTrip Time) which is an estimate of roundtrip time to the destination
    - ∗ Use an EWMA (Exponentially Weighted Moving Average), a low-pass filter that discards noise in the samples
    - ∗ $SRTT = \alpha SRTT + (1 - \alpha)R$
    - ∗ for $R =$ the time elapsed when an ACK comes in.
    - ∗ for $\alpha =$ a smoothing factor (usually $\alpha = \frac{7}{8}$)
  - ¤ Accurate measurements of RTT (which the SRTT is) are necessary but not sufficient
    - ∗ Early implementations of TCP used $2 \times RTT$ as the timer interval
    - ∗ But a constant multiplier fail as networks reach capacity and RTT variance becomes high
    - ∗ In such a case, retransmissions will trigger much more often and push the network over capacity.
  - ¤ Jacobson proposed also keeping track of RTT variance as $RTTVAR = \beta RTTVAR + (1 - \beta)\mathsf{SRTT - R}$ (also an EWMA, usually with $\beta = \frac{3}{4}$)
  - ¤ Then $RTO = SRTT + 4 \times RTTVAR$
  - ¤ This RTO, using SRTT and RTTVAR, very cleverly does all calculations with simple integer operations, hence some constant multipliers being chosen for efficient implementation rather than for statistical soundness
  - ¤ *Karn's Algorithm*: Samples of $R$ must not be taken for segments that have been retransmitted, because it's impossible to know whether the ACK is for the original transmission or for the retransmission.
- *Persistence Timer* prevents deadlock in the case that a receiver announces a window size of 0 and its next window size increase announcement is lost. When the timer goes off at the sender, the sender issues a window probe to the receiver.
- *Keepalive Timer* triggers a check that an idle connection is still there, and if not, the connection is ended.
- TCP also uses a timer of twice the maximum packet lifetime to time out a connection after sending a FIN.

### 6.5.9 TCP Congestion Control

- TCP implements AIMD using packet loss as the binary signal of congestion and its *congestion window* (a maximum number of bytes a sender can have in the network at once) as the mechanism for limiting bandwidth use.
- Congestion control was implemented on top of existing TCP deployments in 1988 due to an Internet-wide *congestion collapse*
- The congestion window is calculated as the difference between bytes acknowledged and bytes sent
- Complication: Large packets may clog slow links down the path
  - ¤ Acknowledgments return to the sender at about the rate that packets can be sent on the path's slowest link
  - ¤ Using these *ack clock* timings to bound the sending frequency, the sender more or less ensures their packets are offered at a smooth rate that doesn't cause queuing at routers.
- Complication: AIMD takes a long time to converge if congestion windows start small, but large windows may cause congestion
  - ¤ *Slow start algorithm* is used. It's a mixture of multiplicative and additive increase
  - ¤ The congestion window starts at 4 segments. For each segment acknowledged before its RTO, the congestion window is increased by one MSS (thus each segment acknowledged results in two more sent)
  - ¤ This results in exponential growth of the congestion window
  - ¤ To arrest this growth, a *slow start threshold* is kept. Initially, its value is arbitrarily high, but once the first RTO happens, the threshold is set to half the congestion window and the slow start algorithm is started again with the congestion window back to 4.
  - ¤ Once the congestion window exceed the threshold, the congestion window ($CW$) goes into an additive increase mode, where the congestion window is increased by usually $\frac{MSS \times MSS}{CW}$ any time one of the $\frac{CW}{MSS}$ packets is acknowledged.
  - ¤ This blended approach allows senders to get a quick idea of the optimal sending rate, gives (a little) time for the network to recover after it is initially inundated by the slow start, and keeps the sending rate from growing too fast beyond the known-safe threshold.
- Complication: Waiting for a timeout is wasteful
  - ¤ When packet has been lost and its successors' acknowledgments arrive at the sender, each will bear the same acknowledgment number.
  - ¤ These *duplicated acknowledgments* can be an early signal that a packet has been lost.
  - ¤ Because out-of-order delivery is possible, TCP only registers a lost packet after 3 duplicate acknowledgments.
  - ¤ *Fast retransmission*: Then the packet is retransmitted, even though its RTO hasn't actually happened yet
  - ¤ The threshold is set to half of the current congestion window, and some implementations will restart slow start with a congestion window of 1 segment (so that the lost packet is acknowledged before adding more load to receiver buffers)
- Complication: At the moment of fast retransmission, the congestion window is too large (the network has become congested), but having to ramp up traffic from a blank slow start is...slow.
  - ¤ *Fast recovery* uses the ack clock (counting incoming acknowledgments and their rate) to resume sending once the number of segments leaving the window is at the slow start threshold.
  - ¤ When a fast retransmission happens, the congestion window is halved.
  - ¤ After about $\frac{1}{2}RTT$, the number of unacknowledged packets matches the slow start threshold, and new segments can be sent out on a 1-for-1 basis as segments are acknowledged.
  - ¤ Another $\frac{1}{2}RTT$ later, duplicate acknowledgments should stop coming in and fast recovery is over
  - ¤ In general, fast recovery avoids slow start mode except at connection startup and when a timeout happens
- SACK (Selective ACKnowledgments) addresses better the issue of multiple unacknowledged segments
  - ¤ During connection setup, the machines agree on SACK using the 'SACK permitted' option
  - ¤ The Acknowledgment Number in the main header has the same meaning, but the SACK option gives up to 3 byte ranges that have been received which are higher-numbered than that
  - ¤ The sender can use this to retransmit the up-to-3 missing byte ranges, without having to retransmit the others, which reduces overall network usage
- ECN (eXplicit Congestion Notification) is implemented with the CWR and ECE header fields

### 6.5.10 The Future of TCP

- TCP's transport semantics aren't a perfect fit for all applications: related streams can't be grouped (a limitation from sockets); network paths can't be controlled

- Packet loss rate is a poor congestion signal as speeds increase: Lost packets are becoming rarer, making it a poorer indicator of congestion.

## 6.6 Performance Issues

### 6.6.1 Performance Problems in Computer Networks

- Structural resource imbalances (fast senders vs slow receivers)
- Synchronous overload (DHCP trampled by a thundering herd after a power outage)
- Improer tuning/settings (e.g., flow control window, timer intervals)
- Jitter/delay

### 6.6.2 Network Performance Measurement

- Seminal work by Mogul (1993)
- Caveats / Suggestions
  - ¤ Use adequate sample sizes
    - ∗ Reduces uncertainty from startup/initialization costs, and queuing varability
  - ¤ Take representative samples
    - ∗ Traffic patterns vary with time of day, day of week, etc.
    - ∗ For wireless, reachability isn't transitive, so ensure your measurements are from the one relevant client, or are taken from a correct distribution of vantage points
  - ¤ Caching obscures performance costs
  - ¤ Avoid unexpected interference
  - ¤ Take many samples if using a slow clock
  - ¤ Don't extrapolate your results

### 6.6.3 Host Design for Fast Networks

- Focus on host speed over network speed: Superfast lines are already the norm, and it's hosts that have to catch up.
- Fewer packets minimizes the impact of overhead
  - ¤ Per-packet overhead is high at low layers: arriving packets will cause interrupts, requiring a context switch at the CPU level
- Touch data minimally
  - ¤ Avoid making many copies of data as it is moved up and down the stack
- Minimize context switches
  - ¤ Buffering on sender (before sending data out) and on receiver (before processing data) lets the CPU pipeline effectively
- Avoiding congestion beats recovering from it
- Avoid timeouts

### 6.6.4 Fast Segment Processing

- Segment overhead is of two types: per-segment overhead, and per-byte overhead
- Make sure to optimize for the normal case(s)
  - ¤ Headers of consecutive data segments are usually quite similar
    - ∗ On sender, a prototype header is stored in the transport entity. Outgoing segments can clone this, and typically only need to change the sequence number and checksum (for IP, only the Identification and the header checksum need to change)
    - ∗ On receiver, finding the right connection for a segment to go to can be shortcircuited 90% of the time by just seeing if the last connection used is the right one.

    - REceivers also use *header prediction* – check that the header is as expected (next un-ACKed sequence number, ensure connection is in a good state, and segment uses no unusual/unexpected flags), then a special fast codepath is used
  - Well-tuned timers rarely fire, so optimize for that
    - A normal approach of timers is to keep a list of timers sorted by time until expiration (or as a list of deltas from the current time or from the previous timer)
    - Better is to keep a circular list, one entry for each clock tick, with each entry being a linked list of timers

### 6.6.5 Header Compression

- When bandwidth is scarce (e.g., wireless), header overhead is costly
- Header compression uses knowledge of the headers formats and defaults to exclude uninteresting data, can reduce a TCP/IP header from 40 bytes min to 3 bytes average
- ROHC (RObust Header Compression), RFC 5795, is designed to tolerate the lossy nature of wireless
  - Encodes a protocol profile (a combination of protocols) and a context (not unlike a connection ID), and can cut the 40 bytes of IP/UDP/RTP down to 1-3 bytes

### 6.6.6 Protocols for Long Fat Networks

- Long fat networks: Fast networks / fat pipes running over great distances
- Problems to overcome
  - Sequence number space: $2^{32}$ sequence numbers wrap around in 34 seconds on 1 Gbps line, far lesss than the standard 120 sec maximum packet lifetime
  - Flow control window too small
    - Bandwidth Delay Product = bandwidth $(\frac{bits}{sec}) \times$ roundtrip delay time $(sec)$
    - Receiver's window must be at least the bandwidth-delay product so that senders aren't always blocked waiting for an acknowledgment (5MB for a transcontinental gigabit line)
  - More complicated retransmission schemes are needed
    - Go-back-N fails when bandwidth-delay product is high because error notifications are slow to arrive and retransmitting a whole window is wasteful
  - Long gigabit lines are delay-limited, not bandwidth-limited: this is dictated by the speed of light
  - Communication speeds have grown far faster than computing speeds have
    - Data comes faster, so new protocols need to be written so they can be processed faster.
- In general: Design for speed, not for bandwidth optimization
- An obvious solution is to implement new protocols in auxiliary hardware for speed
  - This leaves the CPU idling for data, and introduces race conditions when the processors communicate
- Packet layout: make headers small, and word-aligned
- Maximum data size should be higher (e.g., jumbograms)
- Reducing feedback can cut down on some delay-related factors
  - Sliding windows are a good start, but communicating changes in window size is still a bottleneck
  - Rate-limited protocols are promising
  - slow start, with its initial probes of the network, suffers a lot from high delay; reserving resources upfront saves several iterations
- Connection-oriented operation can help some of these issues

## 6.7 Delay-Tolerant Networking

- Some networks (LEO satellites) may go a long time without an end-to-end path, may never have a single end-to-end path at any one moment, or may even go long periods of time without even a single peer
- Such occasionally connected networks tend to use store-and-forward, aka *message switching*, and is called a DTN (Delay-Tolerant Network / Disruption-Tolerant Network)
- DTN models are usable for, e.g., bulk transfer of data in off-peak hours

### 6.7.1 DTN Architecture

- *bundle*: a message
- DTN nodes have a lot of storage
- A working link is called a *contact*
- Nodes may move while storing data
- Example: images taken by Disaster Monitoring Constellation of LEO satellites
  - ¤ Satellites may lack connectivity when taking an image
  - ¤ No single contact may last long enough to send the images, but the 3 terrestrially-connected ground stations combined can pick up a lot of data
  - ¤ Terrestrial ground stations can also store bundles for quite a while, in case the link between them and the central collection point is busy when the bundles arrive on Earth. This means data downloaded from the satellite only needs to be downloaded once
- Routing via DTN nodes can be static (as with satellites) or dynamic (send bundles along different DTN routes and let bundles expire in case they get misrouted)

### 6.7.2 The Bundle Protocol

- Provides transport-like service in the application layer
- Usable on top of different kinds of internets, with convergence layers needed so that it can receive consistent service from them.
- Custodian: the party currently responsible for seeing the bundle delivered
  - ¤ ON the Internet, that's normally the source (it does retransmission, etc.)
  - ¤ DTNs do custody transfer, since the source may not remain connected for long
- Routing
  - ¤ Identifers for DTN (source, destination, custodian) need to be routable at the application level, since it is above so many other possible networking protocols
  - ¤ DTN uses a dictionary to store identifiers, since a single node may be reference by multiple fields (the custodian may also be the source)
- DTN bundles carry a time created and a lifetime (which forces DTN nodes to keep loosely synchronized clocks)

## 6.8 Summary

# 7 The Application Layer

## 7.1 DNS - The Domain Name System

- high-level, human-readable names decouple machines names from machine addresses
- Back in ARPANET days, hosts.txt contained this mapping and was fetched fresh every night from the site's administrator
  - ¤ Such a file would get to be too big over time
  - ¤ Hostname conflicts are inevitable unless the names are managed centrally
- Overall, DNS works by calling a *resolver* library procedure with a hostname. The resolver makes a UDP request to the local DNS server, which will return the IP address

### 7.1.1 The DNS Name Space

- ICANN (Internet Corporation for Assigned Names and Numbers) manages the top of the naming hierarchy.
- 250 *top-level domains* (TLDs): generic TLDs and country TLDs (one per country)
- TLDs are run by *registrars* appointed by ICANN

- Each domain is named by the path up from the domain name tree from it to the root (separated by dot), such that absolute domain names always end with a dot
- Domain names are case-insensitive, with each component $\leq 63$ characters and the entire path $\leq 255$ characters.

### 7.1.2 Domain Resource Records

- Every domain can have a set of *resource records* associated with it: these are what the resolver returns for the domain
- Resource records are a 5-tuple
  - Domain name: the search key used to satisfy queries
  - Time to live: How stable the record is (in seconds)
  - Class: For Internet information, this is always IN
  - Type
    - SOA (Start of Authority): Gives parameters for the zone (flags, timeouts, admin's contact information, serial number)
    - A (Address): A 32-bit IPv4 address for the domain
    - AAAA (quad A): A 128-bit IPv6 address
    - MX (Mail eXchange): Specifies the host that receives email for the domain
    - NS (Name Server): Specifies the name server for the domain
    - CNAME (Canonical NAME): Specifies that the domain is an alias for some other domain
    - PTR (Pointer): In practice, it is used to permit *reverse lookups*, to find a domain name given an IP address
    - SRV (Service): Generalizes the MX record type, by letting you determine the appropriate domain to use for a given (domain name, service) pair.
    - SPF (Sender Policy Framework): Lets a domain specify a full list of domains/hosts from which its valid messages are sent (so that it's easy to flag as spam mail pretending to come from that domain but in reality coming from other hosts)
    - TXT (Text): Usually encodes machine-readable information
  - Value

### 7.1.3 Name Servers

- TO reduce load on DNS servers, and to avoid having a single point of failure, the DNS namespace is divided into non-overlapping *zones*
- Each zone has $\geq 1$ name servers; usually a primary one and several secondary name servers that clone the primary's database
- *Name resolution*: the process of looking up a name to find an address
  - If the local name server has jurisdiction over the name searched, it returns the *authoritative record* (i.e., a record that is sure to not be out-of-date)
  - If the local name server does not have jurisdiction over the name searched, and has no cached information on it, it begins a sequence of remote queries
    - It sends a query to one of the *root name servers* (heavily replicated machines having authoritative knowledge about TLDs, distributed in many geographical locations to be easily reachable by anycast routing) to find the next name server to ask for information
    - A similar procedure continues until some name server gives an authoritative response for the full domain name queried
    - In this scenario, the local name server recursively resolves the name, allowing hosts in its domain (hosts it serves) to stay ignorant of other name servers
  - Cached answers are stored at the local name server, cached for up to 1 day (as specified by the records' Time To Live)
  - Since UDP is not reliable, DNS clients need a mechanism to retransmit if a response isn't received within some duration
- Changing a name-to-address or address-to-name mapping is potentially quite dangerous, so DNSSEC has been created as a security extension to this service
- Content Distribution Networks (CDNs) also pose an interesting edge case to DNS.

## 7.2 Electronic Mail

- Informal, and a low threshold of use: ”With email, a brilliant email from a summer student can have more impact than a dumb one from an executive vice president.”

### 7.2.1 Architecture and Services

- Two main subsystems: *user agents* and *message transfer agents / mail servers*

### 7.2.2 The User Agent

### 7.2.3 Message Formats

- RFC 5322 - The Internet Message Format
  - ¤ Consists of an envelope, some header fields, a blank line, and a body
  - ¤ Fields
    - ∗ To, Cc, Bcc, From
    - ∗ Sender: the address of the person who actually transmitted the message (i.e., an assistant)
    - ∗ Received: One Received header is added for every mail server along the way
    - ∗ Return-Path: usually, just the sender’s address
    - ∗ Reply-To: When the message’s replies should go elsewhere than to the sender.
    - ∗ Message-Id: A number used to link messages together (e.g., it’s used by the In-Reply-To header) and to detect duplicate delivery
- MIME - The Multipurpose Internet Mail Extensions
  - ¤ Intended for non-text (and non-ASCII) content
  - ¤ Defines 5 new headers
    - ∗ MIME-Version
    - ∗ Content-Description: a plaintext description of the MIME-encoded data
    - ∗ Content-Id: A unique id for the encoded data
    - ∗ Content-Transfer-Encoding: Specifies one of 5 transfer encoding schemes (necessary because SMTP expected line lengths $\leq$ 1000 ASCII characters, and ASCII uses only 7 bits per byte) and an escape to new schemes
      - > normal ASCII text
      - > 8-bit characters, still adhering to the max line length
      - > True binary encoding
      - > Binary data transmitted in ASCII via *base64 encoding*
        - ◦ Groups of 24 bits are broken up into four 6-bit units, represented as A-Za-z0-9+/, with special sequences == and = representing a group of only 8 or 16 bits, respectively
      - > *Quoted-printable encoding*: 7-bit ASCII, with characters having value less than 255 but greater than 127 replaced with = followed by its value in 2 hex digits This is useful when the text is mostly but not all ASCII.
    - ∗ Content-Type: the MIME type and subtype
      - > text (examples are text/plain, text/html, text/xml)
      - > image
      - > audio, video
      - > model: for 3D models. Not widely used
      - > application: A catch-all so UAs can install handlers and process data for arbitrary applications (note that this invites security problems)
      - > message: Lets one message be fully encapsulated within another
      - > multipart: Lets the message contain multiple types
        - ◦ multipart/mixed: The different parts may be of different types, and their beginnings, ends, and types will be clearly delineated
        - ◦ multipart/alternative: A single message can be included multiple times in different formats
        - ◦ multipart/parallel: When all parts must be viewed simultaneously (e.g., audio and video tracks)
        - ◦ multipart/digest: used for mailing lists, when multiple messages are packed together into a single composite message

### 7.2.4 Message Transfer

- SMTP has two different modes of operation
  - Mail submission: by which user agents send messages into the mail system
  - Message transfer: by which sending mail transfer agents delivers to a receiving mail transfer agent
- Separate from SMTP is final delivery: how the message gets from the receiving mail transfer agent to the user
- SMTP (Simple Mail Transfer Protocol) and Extensions
  - Server listens on port 25 over TCP, expects to receive commands in ASCII
  - SMTP Limitations
    - The FROM command lets a sender pretend to be any address
    - ASCII-only requires inefficient base64 encoding for binary data
    - Transmitting messages, sender address, and receiver address in plaintext allows snooping
  - Thus ESMTP (Extended SMTP), which starts client handshakes with a different greeting to elicit a list of server-supported extensions (such as client authentication, binary message compatibility, TLS, UTF8 addresses, and message size options)
- Mail submission
  - RFC 4409 recommends strictures to put in place for mail submission
  - Common tricks include requiring ESMTP with the AUTH extension
  - *open mail relays* permit spammers to launder the sender's information
- Message Transfer
  - Sending mail transfer agents use DNS to find the receiving mail transfer agent, and open a connection to port 25 on that machine
  - The receiving mail transfer agent then stores the message in the receiver's mailbox
  - This is mostly done in a single hop; though if the receiving mail transfer agent is tasked with forwarding messages for an individual to a different address (or forwarding messages for a mailing list to a set of addresses), it will be relayed again.
  - Spam detection at the receiving mail transfer agent
    - Lookup sender's address in DNS and ensure the sender mail transfer agent's IP address matches
    - Use DNS to look up the sending domain's mail sending policies (TXT and SPF records) and reject if the message doesn't match

### 7.2.5 Final Delivery

- IMAP - The Internet Message Access Protocol
  - Mail server runs an IMAP server listening on port 143
  - Allows an authenticated client to perform a number of manipulations (search, delete, create) messages and folders stored on the server
  - Compared with its predecessor, POP3, it's harder on the server but easier on the user
- Webmail

## 7.3 The World Wide Web

### 7.3.1 Architectural Overview

- The Client Side
- URL: a scheme/protocol, a DNS name, a path identifying the page
- Sometimes you'd want to point to a resoure without having to specify where it is (no DNS name???), hence generalizing URLs into URIs (uniform resource identifiers), a superset which includes both URLs and URNs (uniform resource names)
- MIME Types
  - Servers' responses include MIME Type for the resource
  - Types not handled natively by the browser can be handled by plug-ins or helper applications

- ¤ Plug-ins: implement some set of procedures expected by the browser, and browser may implement some services for the plug-in to consume
- ¤ Helper applications: A complete program, launched by the browser, running in a separate process

- The Server Side
  - ¤ To serve multiple requests concurrently, multi-threaded designs. e.g., 1 thread accepts incoming connections, and dispatches to one of $k$ other threads for processing, all in 1 process, so that all $k+1$ threads can share a single cache.
- Cookies (RFC 2109)
  - ¤ Fields
    - ∗ Domain: Browsers should discard cookies issued by the wrong domain; domains are expected to store less than 20 cookies on a client
    - ∗ Path: which part of the server's directory structure dmay use the cookie (often just /)
    - ∗ Content: name=value pairs, can be anything the server wants
    - ∗ Expires
      - > If absent, cookie is discarded on browser exit (a *non-persistent cookie*
      - > Persistent cookie: if a time in GMT is given, the browser removes that cookie when that time has passed.
      - > Servers can overwrite a cookie with one having an Expires in the past to guarantee deletion
    - ∗ Secure: Indicates the browser will only send the cookie using a secure transport (SSL/TLS)
  - ¤ Each cookie must be less than 4KB
- Web Tracking
  - ¤ Site $S$ puts a tracking image $I_1, I_2, I_3, ...$ on pages $P_1, P_2, P_3, ....$ Visitors on $P_i$ will load $I_i$, and trigger a request for $I_i$ sent with their cookies for $S$. Thus $S$ builds anonymized profiles of which pages a unique visitor has visited.
  - ¤ Many browers let you block *third-party cookies*–letting you prevent storage and sending of cookies with a domain different from that of the URL you're visiting.

### 7.3.2 Static Web Pages

- HTML
- Input and Forms
- CSS
  - ¤ Allows use of logical, shared styles
  - ¤ Format is: selector { property1:value1; ... }
  - ¤ Separate CSS files allow shared code, and allow HTML pages to remain compact (caching can prevent extra *.css file fetches from being too burdensome as future pages using the same css file are loaded)

### 7.3.3 Dynamic Web Pages and Web Applications

- Server-Side Dynamic Web Page Generation
  - ¤ Two common APIs to invoke programs
    - ∗ CGI-style: an interface mediates communication between web server and some backend script
    - ∗ PHP-style: script is embedded into an HTML document and is executed as the page is served (JSP and ASP.NET are both of this style)
- Client-Side Dynamice Web Page Generation
  - ¤ Javascript, VBScript
  - ¤ Applets: small Java programs compiled to be executed client-side by the JVM
  - ¤ ActiveX controls: programs compiled to x86 machine language for execution on bare hardware
- AJAX: Asynchronous Javascript and XML
  - ¤ Relies on the DOM (Document Object Model: a tree representation of an HTML page's structure). Such a representation lets parts be changed simply and in isolation without having to re-render the entire page.
  - ¤ XML: a language for specifying structured content
  - ¤ XSLT: like CSS but more powerful, to allow transformations of XML into HTML
  - ¤ XML is also stricter, making parsing easier and unambiguous

- XHTML blends these: it's HTML interpreted strictly
- SOAP (Simple Object Access Protocol): a generic way of implementing web services via RPC in a language-independent and system-independent way. Clients send XML messages over HTTP and receive XML responses
- The key for AJAX to be a responsive UX is asynchronous I/O as it processes data client-side, or fetches data from server-side

### 7.3.4 HTTP - The HyperText Transfer Protocol

- Connections
  - Originally, each resource spawned its own connection: high overhead for fetching many icons
  - HTTP/1.1 supports *persistent connections/connection reuse*, which amortizes the cost of connection setup/startup/release, and allows pipelining of requests
  - This invites the question (along with the use of AJAX) of when to shut down the connection: usually, a simple heuristic is in place (it's shut down when the client or server has too many open connections, or when the connection has been idle for 60 seconds)
  - *Parallel connections* are sometimes also used, where multiple streams are set up and requests are balanced over them. This causes the TCP streams to compete, and ties up more resources on both client and server, but may result in faster performance.
- Methods
  - GET, HEAD, POST, PUT, DELETE
  - TRACE: the server returns the request back (used only for debugging)
  - CONNECT: lets a user make a connection with a webserver through an intermediate device such as a web cache
  - OPTIONS: server will respond with the set of methods and headers usable for that page
- Message Headers
  - If-Modified-Since, If-None-Match: for caching
  - Accept, Accept-Charset, Accept-Encoding, Accept-Language
  - Host: The DNS name to handle the request (in case the webserver handles multiple vhosts)
  - Authorization: for client to prove it's permitted to perform the method on the resource
  - User-Agent
  - Referer
  - Cookie, Set-Cookie: see RFC 2109 (RFC 2965 is newer, but not widely implemented)
  - Last-Modified, Expires: for caching
  - Location: to inform the client it should try a different URL
  - Accept-Ranges: indicates the server supports incremental fetching of the resource
  - ETag, Cache-Control: for caching
  - Upgrade: indicates a protocol the sender (whether client or server) wants to switch to
- Caching
  - Pages are cached by the browser on disk
  - On a request, the cache is checked for a valid version of the page
  - A page is clearly invalid if its previous Expires time has passed; else heuristic must be used. Often a very old Last-Modified will indicate a page that changes rarely and the cached version can be used despite passage of its Expires.
  - If a cached copy exists but might be invalid, a *conditional GET* is sent that identifies the cached copy. Servers can respond with the full resource or with an indication that the cached copy is still valid.
    - If-Modified-Since would hold the cached copy's Last-Modified so that the server can determine on time alone whether the cached resource is still valid
    - Servers using ETags would return an ETag (a cryptographic hash identifying the resource's contents) with every response. Clients would then send ETags in the If-None-Match of a conditional GET. This is good for cases when resources might be aliased or renamed often (i.e., when a (name, date) pair doesn't identify contents well).
  - Cache-Control can override either of these methods
  - Pages requiring authorization are never cached
  - Caching is also done outside the browser by intermediate hosts (ISPs)

### 7.3.5 The Mobile Web

- Some difficulties
  - ¤ Small screens require different designs
  - ¤ Input mechanisms are more tedious
  - ¤ Network bandwidth is limited or slow
  - ¤ Connectivity may be intermittent
  - ¤ Computing power is limited
- WAP (Wireless Application Protocol) was designed by Nokia to replace parts of the web stack to better suit mobile, but devices and networks advanced too quickly for it to (need to) be adopted.
- W3C encourages sites to use User-Agent to detect mobile devices and to have separate versions of their site available for mobile use
- XHTML Basic is a stripped down subset of XHTML designed to suit these constraints
- Another option is a computer between the mobile device and the server that performs *content transformation*/transcoding to make the page mobile-friendly
- Header compression (e.g., ROHC) and other protocol changes have also been tried

### 7.3.6 Web Search

- *Web crawling*: finding all pages on the web by traversing links
  - ¤ Some content is hard to find because it's only generated dynamically (the *deep Web*)
  - ¤ indexing and storing the pages seen is messy and expensive
- Naming is now higher-level due to search: in the same way DNS made it possible to forget hard-to-remember IP addresses, web search lets us forget hard-to-remember URLs in lieu of page content

## 7.4 Streaming Audio and Video

- Delay and variation in delay (jitter) are major difficulties in any kind of realtime transmission (continuous media / streaming media)

### 7.4.1 Streaming Audio

- Sound variation of only a few milliseconds is noticeable (not so for visual variation)
- ADCs (Analog-to-Digital Converters) convert electrical voltages to binary numbers
- If a sound is a linear superposition of sine waves where the highest frequency component is $f$, the Nyquist theorem tells us there is no benefit to sampling at a frequency more than $2f$
- Each sample taken of a given size (a finite number of bits) introduces *quantization noise*
- Examples of sampling
  - ¤ PCM over telephones: 8000 $\frac{samples}{sec}$, 8-bit samples, encoded in either µ-law (in North American and Japan) or A-law (Europe and elsewhere) with nonlinear sampling scales. Frequencies over 4kHz are lost.
  - ¤ CD audio: 44,100 $\frac{samples}{sec}$, 16-bit samples, chi-distributed linearly over the range of amplitudes. Note that 65,536 values is far less than the dynamic range of the human ear (over 1 million).
- Audio Compression
  - ¤ Usually media is encoded once and decoded many times, so decoding needs to be faster. This is not so in the case of teleconferencing, where both encoding and decoding need to both be realtime.
  - ¤ Encoding/decoding may be non-invertible/lossy. If encoding is lossy, that's okay for many applications
  - ¤ Significant research has been done on *vocoders* (encoders good at compressing human speech) since the invention of the telephone
  - ¤ Two approaches to audio compression
    - ∗ *waveform coding*: the signal is decomposed mathematically into frequency components, and the amplitude of each is encoded minimally, which produces a faithful/representative output
    - ∗ *perceptual coding*: exploits flaws in human auditory processing (studied in *psychoacoustics*) to produce a sound that is different, but different only in ways humans can't easily detect.

> primarily takes into account *masking* (the ability of one sound to hide another)
> *frequency masking*: a loud sound in one frequency band may hide a softer sound in another frequency band
> *temporal masking*: a loud sound that occurred recently may hide softer sounds of similar frequencies occurring now.
> Samples are processed in small batches, each batch divided into frequency bands, the bands passed through the psychoacoustic model, and the budget of bits per sample is allocated preferentially to frequency bands having greatest unmasked spectral power.

### 7.4.2 Digital Video

- In general
  - ¤ pixels use 8 bits for each of R, G, B
  - ¤ frame rates: NTSC (29.97 $\frac{frames}{sec}$), PAL (25 $\frac{frames}{sec}$), 35mm film (24 $\frac{frames}{sec}$)
  - ¤ For smoother motion, images are split into 2 *fields*, one of odd numbered scan lines, the other of even. They're sent separately at 50 or 60 $\frac{frames}{sec}$ and *interlaced*.
  - ¤ Faster redraw speeds on modern screens (computer monitors) makes interlacing unnecessary and leads to *progressive scan*
  - ¤ Interlaced videos on a progressive screen will cause *combing*–short horizontal lines visible near sharp edges.
  - ¤ Aspect ratio vs resolution
- Video Compression is necessary: uncompressed 640x480 video with 24 color bits per pixel at 30 $\frac{frames}{sec}$ would need more than 200 Mbps.
- The JPEG Standard (International Standard 10918, lossy sequential mode) follows these steps (assuming a single frame of 640x480 video with 24 bits of color per pixel)
  - ¤ 1. Block preparation
    - ∗ Eyes are more sensitive to *luminance* (brightness) than to *chrominance* (color)
    - ∗ Calculate $Y$, $C_b$, $C_r$ for every pixel.
    - ∗ The $C_b$, $C_r$ matrixes are quartered in size (by averaging neighboring 2x2 squares), losing color information
    - ∗ Subtract 128 from all matrices to yield a mode of 0.
    - ∗ Divide each matrix into 8x8 blocks, yielding $Y$ with 4800 blocks, and $C_b$ and $C_r$ with 1200 blocks apiece
  - ¤ 2. DCT (Discrete Cosine Transform)
    - ∗ perform DCT on each block, for each block yielding an 8x8 matrix of DCT coefficients
    - ∗ These coefficients represent spectral power at each spatial frequency (recall that each block represents luminance or chrominance of 8x8 or 16x16 pixels, hence space $\times$ frequency)
    - ∗ DCT coefficients will normally have (0, 0) as the average of the block, with coefficients rapid decaying radiating outward
  - ¤ 3. Quantization
    - ∗ Divide DCT coefficients by weights taken from a fixed table (quantization table) to wipe out less important signals. The quantization table, provided by the application, controls the lossiness of the encoding.
  - ¤ 4. Differential Quantization
    - ∗ Reduce the (0, 0) value of each block by its delta from the same value in the previous block
  - ¤ 5. Run-length encoding
    - ∗ Each block (matrix of values) is diagonalized into a linear list of values. Because values shrink with distance from (0, 0), run-length encoding reduces the storage cost of many consecutive low values.
  - ¤ 6. Huffman-encode the list of numbers
- JPEG commonly results in a 20:1 compression, with roughly symmetric encoding/decoding speeds.
- The MPEG Standard
  - ¤ Redundancies present in video are of two types
    - ∗ Spatial redundancy: nearby pixels look similar (JPEG covers this fine)
    - ∗ Temporal redundancy: frames may look similar to one another
  - ¤ Naively, temporal redundancy can be dealt with by subtracting adjacent frames and JPEG encoding the difference. But this fails if the camera or background moves (in which case, strong similarities remain but are not compressed)
  - ¤ MPEG compensates for motion with 3 different frame types

- I-frames (Intracoded): still picture is compressed in isolation
  - I-frames should be sent every 2-3 seconds so there's a baseline frame that allows seeking (rewind, fast-forward) and that allows recovery in case some data is lost
- P-frames (Predictive): block-by-block difference with previous frames
  - Based on *macroblocks* (16x16 pixels of luminance space and 8x8 pixels of chrominance space)
  - A decoder looks for a certain macroblock or something like it in the previous frame so it can make changes to that macroblock
  - Searching for the macroblock to change is left up to implementation, meaning encoding can be made very hard if you want to.
  - Decoding P-frames requires extra buffering, since viewed frames must be kept around for comparison.
- B-frames (Bidirectional): block-by-block differences between nearby frames
  - Can reference macroblocks in previous or future frames, useful when objects move around behind one another.
  - Encoder and decoder need to buffer more, and decoder must wait for successive frames (imposing delay) to render a current one.

### 7.4.3 Streaming Stored Media

- *Metafile* architecture: browser can request an URL on rtsp scheme, which fetches some small file with a content type handled by another application. The application then reads the file and uses its information to contact a media server for streaming content
- Media Players have 4 major jobs
  - Manage the UI
  - Handle transmission error
  - Decompress content
  - Eliminate jitter
- Small losses of data are easy to recover from, but an entire packet could lose many samples or frames. Two techniques to reduce the problem
  - FEC (Forward Error Correction)
    - For every 4 packets, send a 5th *parity packet* that is an xor of the 4
    - This has a high bandwidth cost, adds latency to decoding, and is still not foolproof
    - Previously, we used parity to detect errors, not correct them. In this case, we know from checksums and sequence numbers that arriving packets are error-free or not, and which one is missing/outstanding. These two properties allow us to use parity to reconstruct the *erasure* or total loss of data(contrast with fixing corruption)
  - Interleaving
    - Loss of a packet (or a burst of them) will be spread out over time
    - Restricts your compression options
    - No additional bandwidth, but still adds to latency.
- Decoding needs to be tolerant to packet loss (hence MPEG's I-frames)
- Eliminating jitter
  - Don't start media playout until the buffer is full to the *low-water mark*
  - Jitter has 2 basic sources
    - Competing network traffic
    - Compressibility of the content
  - To avoid buffering too much data and unnecessary consumption of network resources, use a *high-water mark*. When a buffer is full to that point, pause the media server from sending more until the buffer is at the low-water mark.
  - Use bandwidth-delay product to calculate appropriate high- and low-water marks.
- RTSP (usually over TCP) provides a "remote control" for streaming of content.
  - Provides capabilities to start, stop, play specific intervals, play faster or slower, and to seek to specific positions
  - TCP provides reliability, buffering, and a complete copy of the data, at the cost of higher startup latency and a higher low-water mark.

### 7.4.4 Streaming Live Media

- Buffers need to be large enough to handle the maximum network jitter.
- Multicasting (RTP over UDP) is a natural choice
  - ¤ FEC has an added benefit here: different clients may lose different packets, and the parity packet provides equal protection for all of them at a fixed cost.
  - ¤ IP multicast is not broadly available on the Internet
- On the Internet, TCP is still usually used for 1:many live broadcasts
  - ¤ i.e., each of $N$ users establishes a separate connection to the media server, making up $N$ competing streams for the same data
  - ¤ Many firewalls will reject traffic that's not a connection request on port 80, so TCP disguised as HTTP traffic has that going for it too.

### 7.4.5 Real-Time Conferencing

- One-way latency of 150msec is acceptable in POTS, but more is annoying
- Long receiver-side buffering and retransmission aren't options because of latency
- Latency is brought on by distance, primarily
- Keeping latency low also requires sending packets sooner/buffering less on the sender (which reduces bandwidth efficiency)
- Software overhead (encoding and decoding) also eats into the 'delay budget'
- Quality of Service at the network layer
  - ¤ Differentiated Services
    - ∗ for VoIP, the DS codepoint is the Expedited Forwarding class with Low Delay type of service
  - ¤ Integrated Services
    - ∗ Reserving resources via IS is not widely deployed
    - ∗ Individual clients may make SLAs with their ISPs, or change settings (quality vs compression settings) on their application to minimize their own bandwidth needs
- H.323
  - ¤ An architecture for Internet telephony (not a protocol)
  - ¤ A gateway connects the Internet to the telephone network
  - ¤ A LAN may have a *gatekeeper* controlling the telephony traffic leaving its jurisdiction/*zone*
  - ¤ Communicating devices are *terminals*
  - ¤ H.323 must support standard G.711 for telephony encoding/decoding
  - ¤ H.245 is used by terminals to negotiate compression algorithms and connection details
  - ¤ RTCP is used to control RTP channels
  - ¤ Q.931 is used to interface with standard telephony (dialtones, ringing, connection setup and release)
  - ¤ H.225 lets terminals and the gatekeeper communicate, over the RAS (Registration/Admission/Status) channel
- SIP - the Session Initiation Protocol
  - ¤ RFC 3261, meant to be more flexible and lightweight than H.323
  - ¤ To establish, maintain, and terminate 1:1, many:many, and 1:many connections, leaving other protocols to deal with data transport
  - ¤ A text-based protocol, like HTTP, with URLs using the sip schema.

## 7.5 Content Delivery

- With communication, there is a specific end node you want to talk to. With content, what matters is finding any end node with the content you want.
- CDN (Content Delivery Network) has a provider setting up many machines all over the Internet to deliver content.
- P2P (Peer-to-Peer) has a collection of computers pooling resources to serve to one another.

### 7.5.1 Content and Internet Traffic

- Changes to the distributions and types of Internet traffic happen quickly and often.
- Traffic is skewed: some content is super-popular, some not
  - Traffic roughly follows a Zipf distribution (i.e., the $N$th most popular item is used about $\alpha$ times more often than the $\alpha N$th most popular)
  - Zipf distributions belong to a family known as *power laws* – patterns that describe a few big players and many small ones, and show up as roughly linear on a log-log scale
  - Unlike with exponential decay, in a Zipf distribution the long tail makes up a substantial part of a curve's area (or, in terms of the Internet, unpopular sites as a whole account for a substantial part of the traffic)

### 7.5.2 Server Farms and Web Proxies

- Server Farms
  - One possibility is to have DNS spread requests out over multiple distinct IPs
  - Another solution is to have a *front end* (a link layer switch or an IP router) that sprays requests over the pool of servers
    - It can in some cases send all traffic to all servers in the pool, and servers would choose based on some partitioning strategy to respond or not
  - Alternately, it can *load balance* the traffic: inspect the traffic, then forward it onto some specific server in the farm, based on some sort of traffic mapping policy. In this case, the box is a switch or a router. Since it interposes itself into the network path in a way that violates network layer expectations, it's called a *middle box*
- Web Proxies
  - Purpose is to share a cache among users to reduce response time and network load incurred for popular pages
  - Doesn't work well for encrypted traffic, pages that require authentication, pages that change constantly (current weather, current stock prices), and pages that are generated dynamically
  - Multiple proxies may be used: a browser (with its own cache) might send a request through an *upstream proxy* maintained by a company, upstream of which is another proxy run by the company's ISP
  - Other benefits of proxies are request filtering, and anonymizing traffic by bundling it together.

### 7.5.3 Content Delivery Networks

- Content is distributable from a CDN origin server to a number of CDN nodes. Individual requests are sent to CDN nodes and never to the origin.
  - A tree structure like this keeps distribution efficient
  - Each client gets good performance by fetching from nearby sources (i.e., TCP costs related to bandwidth-delay product are reduced) and the likelihood of experiencing congestion is reduced
  - Total load placed on the network is not only reduced by taking shorter paths, but is also isolated
- Getting clients to use the CDN tree
  - Proxy servers: Clients (whether organizations or individuals) would have to configure the CDN to be used, which is unlikely
  - *Mirroring*: Let the users select which of many mirrors (CDN nodes with a full copy of the content) to fetch data from
  - *DNS Redirection*: Name servers run by the domain will use the requesting IP address to decide which CDN node is best, and return DNS records relevant for that CDN node. The best CDN node is typically chosen based on some combination of network path, network capacity, and the current load of the CDN nodes
- CDNs work well combined with peering
- Links to CDN hosted content must be rewritten, usually to $vhost.$cdn.com or $cdn.com/$your_company so the CDN company can rejigger their DNS/network without needing to synchronize with their clients.

### 7.5.4 Peer-to-Peer Networks

- Basically, many computers pool their resources to form a content distribution network, but without dedicated infrastructure
- BitTorrent
  - Content provider creates a *torrent* containing
    - Name of a *tracker*, a server that leads peers to the content.
    - A list of *chunks*, equal-sized portions of the content, each specified as a 160-bit SHA1. Usually each chunk is 64-512KB, so torrents are typically 3 or more orders of magnitude smaller than the content.
  - Peer receiving a torrent contacts the tracker, which maintains information on the *swarm* (the set of peers actively downloading/uploading) via periodic contact with each peer
  - Peers preferentially exchange rarer chunks, to reduce reliance on a small number of *seeders* (peers with the whole content).
  - This is more robust than all exchanging chunks in the same order (which creates a bottleneck at, and a dependence on, the seeder(s))
  - Peers randomly sample other peers, to determine god upload behavior. Over time, peers with comparable upload and download rates are matched with one another
  - Peers who don't upload or do so slowly, get *choked* (cut off from downloading) to reduce the number of *free-riders/leechers.*
- DHTs – Distributed Hash Tables
  - BitTorrent trackers remain centralized, and thus a liability
  - Properties needed for P2P indexes that are fully distributed but still performant
    - Nodes keep only a small amount of information about each other so the index can kept up-to-date
    - Nodes can look up entries in the index quickly
    - Each node can use the index at the same time, so that an increase in nodes only increases (never decreases) performance
  - These work by imposing a regular structure on inter-node communication, thus DHTs are called *structured P2P networks*
  - Chord DHT system
    - A circular doubly linked list of max size $2^m$ is created.
    - Each node stores knowledge of its own node ID $(0..2^m - 1)$, a *finger table* of $m$ other nodes (the first being its successor) distributed through the list, forming a shallow skip list.
    - A client wishing to join hashes its IP address to find which node ID it should register with.

## 7.6 Summary

# 8 Network Security

- Roughly four different problems
  - Secrecy/confidentiality: Keep information out of unauthorized hands
  - Authentication: Determining you're talking to the right person
  - Non-repudiation/signatures: Proving a specific entity performed a specific action
  - Integrity Control: Being certain that a message received has not been modified or forged outright.

## 8.1 Cryptography

### 8.1.1 Introduction to Cryptography

- $C = E_K(P)$, $P = D_K(C)$
- *Kerckhoff's Principle*: Assume that all algorithms are public, and only keys are private
- Security by obscurity: trying to keep algorithms private
- 3 variations on cryptanalysis

  - ciphertext-only: attacker has access only to ciphertext
  - known plaintext: attacker has matched some plaintext with some ciphertext
  - chosen plaintext: attacker can encrypt pieces of plaintext of their choosing
- *work factor* of an algorithm is the function that represents how hard it is to break.

### 8.1.2 Substitution Ciphers

- Substitution cipher: replacing $geq1$ letters with $\geq 1$ other letters
- Caesar cipher is a shift cipher that shifts upward by 3 characters
- Monoalphabetic substitution cipher is the non-shifting version, but is still easy to crack with n-gram frequencies or word frequencies (if the plaintext is expected to use certain words)

### 8.1.3 Transposition Ciphers

- Letters are reordered in a reversible manner
- Character frequencies are preserved, but digrams/trigrams/words are split apart
- Still, knowing a certain stringly (likely) recurs in the plaintext would let you find if those letters are displaced from one another by a fixed distance.

### 8.1.4 One-Time Pads

- There is no information in the ciphertext, since any message of the same length as the plaintext can be obtained by choosing an appropriately misleading decryption key.
- In Practice
  - Key must be held, uncorrupted, by both sender and receiver
  - Information encryptable is bounded by the amount of key shared
  - Key cannot be reused
- Quantum Cryptography
  - Orientation of light waves can be horizontal, vertical, or either of 2 diagonals
  - Alice and Bob (and Trudy) randomly choose a sequence of bases (one basis per bit, each basis being two orthogonal lines)
  - Each bit is sent as a single photon (packet of light)–a qubit
  - At any moment, Alice can send horizontal, vertical, $y = x$, or $y = -x$ oriented light, and chooses these to represent 0 or 1 bits.
  - Bob's basis either sees the light Alice sends or doesn't, Trudy's basis sees a different subset of Alice's message
  - Alice and Bob first communicate via plaintext what orientations map to 0 and which to 1
  - Bob responds with which bits he received correctly.
  - alice nad Bob will use this sequence as their one-time pad, and use techniques to turn this knowledge they share (that Trudy lacks) to generate a longer key that's harder to brute force (*privacy amplification*)
  - Alice eventually sends Bob information using a heavy Forward Error Correcting Code. This means that if Trudy (who doesn't have the whole key) tries to tamper, it will be detected. Because it is impossible to inspect and clone a photon, Trudy can't snoop passively, much less inject a false message.

### 8.1.5 Two Fundamental Cryptographic Principles

- Redundancy
  - Is needed to prevent garbage sent by active intruders being acted on as if valid after decrypted
  - Usually done with CRC polynomials or cryptographic hashes
- Freshness
  - Must be some way to prevent replay attacks from being acted on

## 8.2 Symmetric Key Algorithms

- use the same key for encryption as for decryption
- Block ciphers: n-bit blocks of plaintext turn into n-bit blocks of ciphertext
  - ¤ P-box: permutes some bits into a different order
  - ¤ S-box: substitutes some characters into different ones
  - ¤ Product cipher: Alternates P-boxes with S-boxes in some number of rounds

### 8.2.1 DES: The Data Encryption Standard

- 64 bit blocks, 56 bit key, 19 stages (transposition, then 16 rounds (each of which using different parts of the key), then swapping the two halves of ciphertext, then transposing again)
- *whitening* adds two 64-bit keys to the mix, one before DES encryption and one just after DES encryption (but before transmission)
- Triple DES
  - ¤ Use 2 keys and 3 stages: $C = E(K_1, D(K_2, E(K_1, P)))$, $P = D(K_1, E(K_2, D(K_1, C)))$
  - ¤ Gives 112 bits of key
  - ¤ EDE (rather than EEE) grants backward compatibility with DES: in the case that $K_1 = K_2$, $E(K_1, D(K_2, E(K_1, x))) = E(K_1, x)$ for any x

### 8.2.2 AES - The Advanced Encryption Standard

- Two variants (practically speaking): 128 bit blocks with either a 128 bit key or with a 256 bit key

### 8.2.3 Cipher Modes

- ECB mode (Electronic Code Book): plaintext is broken up into blocks (maybe with padding added to the last block)
- CBC mode (Cipher Block Chaining)
  - ¤ Starting with some initialization vector (IV) as $C_0$, each block $C_n = E(K_1, C_{n-1} xor P_n)$
  - ¤ Main advantage: repeated plaintext blocks within a message result in non-same ciphertext blocks
- Cipher Feedback Mode
  - ¤ A shift register is used
- Stream Cipher Mode
  - ¤ Encrypt an IV with your key to get a *keystream* which you can use as a one-time pad
  - ¤ A reused (key, IV) pair opens you up to a keystream reuse attack. If $C_p = P xor K_1$ and $C_q = Q xor K_1$, then $C_p xor C_q = P xor Q$. If part of P or Q is known, the other can be discovered
- Counter Mode
  - ¤ Invented to allow non-sequential access to the plaintext of encrypted data.
  - ¤ $C = E(K, IV + offset) xor P$
  - ¤ Vulnerability: If K and IV are reused, then the keystream $E(K, IV + offset)$ can be recovered.

### 8.2.4 Other Ciphers

### 8.2.5 Cryptanalysis

- Differential cryptanalysis for block ciphers: starting with 2 very similar plaintext blocks, see what happens when they're encrypted (opens up probabilistic attacks)
- Linear cryptanalysis: By xoring certain parts of ciphertext and plaintext together, if there's a disparity in the number of 0s and 1s, it is a bias that can be used to reduce the work factor
- Power consumption analysis: Processing a 1 takes 3 voltes, a 0 takes 0 volts, so power consumption during encryption/decryption can be revealing
- Timing analysis: Different branches of an if/else may take different directions, in whcih case timing alones can tell you things about the key or the plaintext.

## 8.3 Public-Key Algorithms

### 8.3.1 RSA

### 8.3.2 Other Public-Key Algorithms

### 8.3.3 Digital Signatures

- Goals
  - ¤ Receiver can verify identity claimed by sender (authentication)
  - ¤ Sender can not deny the message (non-repudiation)
  - ¤ Reciver couldn't have forged the message

### 8.3.4 Symmetric-Key Signatures

- A central authority has established a key with each individual (the central authority and the individual) each have a copy, nobody else does.
- For Alice to send P to Bob, send A, $K_A(B, R_A, t, P)$ to central authority $BB$.
- $BB$ sends to Bob $K_B(A, R_A, t, P, K_{BB}(A, t, P))$
- $BB$ sending the message along implies that Alice's key was used to send the message to $BB$ (rather than a 3rd party, Trudy)
- $K_{BB}(A, t, P)$ proves that $BB$'s key was used to send the message to Bob, not some other 3rd party
- Timestamps prevent very old replays; random numbers prevent recent replays

### 8.3.5 Public-Key Signatures

- For sender Alice and receiver Bob, if Alice sends $E_B(D_A(P))$
  - ¤ Only Bob can decrypt the message to get $D_A(P)$
  - ¤ Anyone can apply Alice's public key to get P from that.
- This is not secure if htere is any doubt that Alice's key is not secret, or if Alice has changed her key.

### 8.3.6 Message Digests

- Four important properties of a message digest
  - ¤ Given $P$, $MD(P)$ is easy to compute
  - ¤ Given $MD(P)$, $P$ is not feasible to compute
  - ¤ Given $P$, one cannot feasibly find a $P'$ such that $MD(P) = MD(P')$
  - ¤ Changing 1 bit in $P$ changes $MD(P)$ drastically
- Symmetric-Key
  - ¤ Instead of signing a message with $K_{BB}(A, tP)$, $BB$ can instead produce $K_{BB}(A, t, MD(P))$, which should be faster since $MD(P)$ is usually shorter than $P$ and $MD(P)$ is fast to calculate
- Public-Key
  - ¤ For sending messages with don't need secrecy, Alice can send to Bob $P, D_A(MD(P))$. Trudy and Bob can both see $P$ and calculate $MD(P)$, but nobody has $D_A()$ but Alice, so nobody can forge such a request.
- SHA-1 and SHA-1
- MD5

### 8.3.7 The Birthday Attack

- If there is some mapping between inputs and outputs with $n$ inputs (people) and $k$ outputs (birthdays), if $\frac{n(n-1)}{2} > k$, a match is likely (or, if $n < \sqrt{2}k$)

## 8.4 Management of Public Keys

- It's not sufficient to put public keys on your website. Some MitM could just change the key so its their own public key, and thus they could inject themselves into Alice and Bob's private communication.

### 8.4.1 Certificates

- Certificate Authority (CA) binds a public key to the name of a principal (individual/organization) using the CA's private key
- All parties knowing the CA's public key can then authenticate the identify of the certificate, which allows all to identify tampering
- Certificates can also be used to bind a key to an attribute. The holder could send that certificate to a site, which verifies the certificate was issued by a trusted CA, then use the public key to issue a challenge to the certificate holder
- This attribute notion can be extended to authorization, and allows a party to be authorized or to delegate authorization.

### 8.4.2 X.509

- The ITU's standard for certificates (accepted by IETF in RFC5280)
- Certificates encoded with OSI ASN.1 (Abstract Syntax Notation 1)

### 8.4.3 Public Key Infrastructures

- Bad ways of doing it
  - ¤ A centralized, single CA is a failure-prone bottleneck
  - ¤ Multiple CAs using the same private key makes it much harder to keep the key from being stolen
  - ¤ No single organization is universally considered trustworthy and legitimate
- Hierarchical CAs
  - ¤ One root authority globally, it certifies a number of authorities which themselves may certify other authorities
  - ¤ Each authority holds a cert issued by any authority which has directly certified it
    - ∗ If Alice wants to talk to Bob, she asks for his cert and the identity of his cert's issuing authority, then asks that issuing authority for its cert and the identity of its cert's issuing authority, ... until she gets the cert issued by the root authority, which she must have a priori knowledge of. If this lineage of CAs checks out, she now has Bob's public key and a chain of trust
    - ∗ If Bob is a really nice guy, he caches the cert lineage and gives it to Alice all at once.
    - ∗ This lineage is called a *chain of trust* or a *certification path*
    - ∗ In practice, 100s of root CAs are used, each root called a *trust anchor*, resulting in a forest of certification
    - ∗ Users have to trust browsers to ship with only trustworthy trust anchors
- Directories: where certs are stored
  - ¤ Inconvenient for users to store them
  - ¤ Some suggest using DNS for this, others recommend dedicated directory servers
- Revocation via Certificate Revocation Lists (CRLs)
  - ¤ CAs periodically issue a CRL litsing unexpired certs it has revoked
  - ¤ Creates timing issues/race conditions; forces clients to always have to check with CA to get up-to-date
  - ¤ There's no provision for reinstatement of revoked certificates
  - ¤ CRLs grow with the length that certificates are alive

## 8.5 Communication Security

### 8.5.1 IPsec

- Designed at the transport layer so that the layperson can benefit a little, even though the greatest security comes with proficient use at the application layer

- Services offered: secrecy, data integrity, protection from replay attacks
- IPsec is designed with room to plug in multiple algorithms so it can be used as algorithms are discovered to be insecure
- Provides multiple granularities as well: a single connection, all traffic between a pair of hosts, all traffic between a pair of routers, etc..
- *Security Association* (SA) - a simplex connnection between two endpoints, having an identifier
- Two main parts
  - ¤ Two new headers that can be used to carry the extra information
  - ¤ ISAKMP (Internet Security Association and Key Management Protocol), a framework for establishing keys (it mainly uses IKE (Internet Key Exchange))
- Two modes
  - ¤ *Transport Mode*: IPsec header is inserted just after the IP header, with IP header's Protocol field set ot the value for IPsec
  - ¤ *Tunnel Mode*
    - ∗ The entire IP packet is encapsulated in the body of a new packet
    - ∗ Useful when the tunnel ends somewhere other than the final destination (e.g., a VPN)
    - ∗ Also useful to aggregate a bundle of TCP connections as one stream – hosts located along the length of the tunnel can't see which sender is transmitting to which receiver (i.e., defies *traffic analysis* along the tunnel)
    - ∗ Adds a lot of overhead (an entire IP header)
- Protocols
  - ¤ AH (Authentication Header)
    - ∗ Provides integrity checking, replay detection, but not secrecy
    - ∗ In transport mode
      - > Next header (8 bits): the original value of the IP packet's Protocol field
      - > Payload Length (8 bits): number of 32-bit words in the AH, minus 2
      - > 16 bits reserved
      - > Security parameters index (32 bits): connection identifier
      - > Sequence number (32 bits): Each packet (even a retransmission) gets a new sequence number, to detect replay attacks. If sequence numbers wrap around, the protocol requires establishing and switching to another SA
      - > Authentication data (variable length): Payload's signature, built with an HMAC (Hashed Message Authentication Code), a hash which is quick to compute and uses the shared key
  - ¤ ESP (Encapsulating Security Payload)
    - ∗ Contains this data (where

      $$x$$

      means that x is Authenticated, {y} means that y is Encrypted
      - > In transport mode: IP header,

        $$ESPheader, \{TCPheader, Payload + Padding\}$$

        , HMAC
      - > In tunnel mode: New IP header,

        $$ESPheader, \{OldIPheader, TCPheader, Payload + Padding\}$$

        , HMAC
    - ∗ Fields: Security Parameters Index, Sequence Number (both 32-bit), and an Initialization Vector may follow
    - ∗ Putting the HMAC at the end means it can be computed as bits stream out the network interface and appended at the end – as opposed to AH's HMAC-in-header, which requires buffering
  - ¤ ESP beats AH on all fronts, but AH lives on for historical reasons

### 8.5.2 Firewalls

- A single point through which traffic must pass
- Acts on the network layer as a *packet filter*, removing packets

- usually filters on (IP, port), but also often on application-level information (acting as an *application-level gateway*), such as blocking certain protocols, or preventing certain kinds of email attachments
  - Fragile: traffic masquerading on a port number usually used by a 'safe' application can still get through
  - Fragile: IPsec hides higher-layer info from the firewall's view
  - Fragile: Applications whose packets are dropped get no feedback from the firewall, only that ACKs never come.
- A normal architecture is to have a DMZ (DeMilitarized Zone) separating a firewalled network from the Internet. The DMZ then contains some servers in contact with the world at large, and the DMZ servers form the only entry/exit points with that network.
- Firewalls have also become stateful, to allow connection-oriented filtering
- DoS, DDoS attacks

### 8.5.3 Virtual Private Networks

- Private networks are expensive, but firewalls+IPsec with ESP in tunnel mode provide integrity control, secrecy, and some protection against traffic analysis
- It's also possible to set up a VPN via the ISP, using MPLS to keep the VPN traffic unseen by those outside of it
- Transparent to user, software

### 8.5.4 Wireless Security

- 802.11 security
  - WEP (Wired Equivalent Privacy) is weak, WPA2 (Wifi Protected Access) replaces it
  - Usable with multiple (username, password) pairs
    - 802.1x: access point lets client authenticate with authentication server and just listens in
    - EAP (Extended Authentication Protocol): a framework that dictates how client and server interact to authenticate
  - Also usable with a single password
  - After authentication, a session key is derived that will be used to encrypt traffic.
  - Session key
    - What part of the session key is derived from non-public information ???
  - AP and client negotiate a hidden session key, and AP distributes a group key (used to broadcast and multicast to the LAN)
  - But CCMP (Countermode with Cipherblock Chaining Message Authentication code Protocol) is not
  - TKIP (Temporary Key Integrity Protocol) is weak
    - Uses AES with a 128-bit key, 128-bit block size, in counter mode, to provide confidentiality
    - For integrity, the message is encrypted in CBC mode and stripped to the last 128 bits, to provide a MIC (Message Integrity Check) that also gets sent along
- Bluetooth Security
  - Physical layer uses frequency hopping for a little security
  - When a slave asks for a channel with the master
    - Deprecated: slave/master sold as a unit would ship with a shared secret key
    - One device has a hardwired *passkey* of 4 digits the user has to enter on the other
    - Since 2.1, via simple secure pairing, devices pick a 6 digit passkey
    - Slave and master ensure they both know the passkey, negotiate what security is required (encryption, integrity control, ...) and select a 128 bit session key
    - Encryption uses a stream cipher, $E_0$, ingerity control uses SAFER+; both are symmetic key block ciphers (???). SAFER+ was a candidate for AES eliminated for being slower than Rijndael
  - A security issue with Bluetooth is that it authenticates devices, not users, so a stolen phone may still be a breach.

## 8.6 Authentication Protocols

- Basic goal: Alice contacts Bob, or a trusted KDC (Key Distribution Center) and eventually Alice and Bob know they are talking with one another, in a way that their communication can't be replayed, modified, or forged, even if intercepted.

- Usually, these protocols also set up a *session key* that only Alice and Bob know, suitable for use in symmetric-key communication

### 8.6.1 Authentication Based on a Shared Secret Key

- For: $A$ the identity of Alice; $B$ the identity of Bob; $R_i$ a challenge issued by $i$; $K_S$ a session key (for S being the identities of the parties in the session)
- Avoiding common authentication protocol pitfalls
  - ¤ Initiator should prove their identity before responder
  - ¤ Initiator and responder should use different shared keys
  - ¤ Initiator and responder should draw challenges from different sets
  - ¤ Should resist attacks that use a second parallel session to gather info for the first session
- $$\begin{array}{lllll} A & \rightarrow & R_A & \rightarrow & B \\ A & \leftarrow & R_B,\, HMAC(R_A, R_B, A, B, K_{AB}) & \leftarrow & B \\ A & \rightarrow & HMAC(R_A, R_B, K_{AB}) & \rightarrow & B \end{array}$$

### 8.6.2 Establishing a Shared Key: The Diffie-Hellman Key Exchange

- Allows parties to establish a secret key, but doesn't authenticate. So a MitM could sit in the middle setting up shared keys with both Alice and Bob

### 8.6.3 Authentication Using a Key Distribution Center

- Mitigating replay attacks
  - ¤ Timestamps: require time to be in sync, and replays are still possible for a short time
  - ¤ Nonces: must be remembered forever and never be reused, or else an attacker with greater storage wins
- Needham-Schroeder authentication protocol (multiway challenge-response)
  - ¤ KDC gives initiator a ticket that is decrypteable only by Bob. Upon decryption, the ticket identifies Alice and her session key, then Alice and Bob use this to convince each other they're speaking directly.
- Otway-Rees is only 4 steps long and is slightly more secure

### 8.6.4 Authentication Using Kerberos

- Involves these services
  - ¤ Authentication Server (AS) to verify users at login
  - ¤ Ticket-Granting Server (TGS) to issue proof of identity
  - ¤ Synchronized clocks

### 8.6.5 Authentication Using Public-Key Cryptography

## 8.7 Email Security

### 8.7.1 PGP (Pretty Good Privacy)

- Uses a block cipher IDEA (International Data Encryption Algorithm) which uses 128-bit keys
- RSA for key management, MD5 for data integrity
- Procedure
  - ¤ *encrypted_hash* := RSA( $D_A$, MD5($P$) );
  - ¤ $P1$ := concat( $P$, *encrypted_hash* );
  - ¤ $P1.Z$ := zip($P1$)

- $K_M$ := some entropy from Alice (a message key)
- *output* := base64( concat( IDEA($K_M$, $P1.Z$), RSA($E_B$, $K_M$) ) )

- RSA is only used twice, both times on 128 bit quantities, so strong RSA key lengths are advised. Key lengths supported are "casual" (384 bits), "commercial" (512 bits), "military" (1024 bits), "alien" (2048 bits)
- Key Management
  - Each user maintains a private key ring and public key ring
  - Private key ring: $\geq 1$ private/public key pairs
    - Lets users change keys (and keep the old ones around long enough to finalize old communications) in case of compromise
    - Each pair has an identifier so receiver knows which public key to use for decryption
    - Private keys are encrypted on disk
  - Public key ring: public keys of user's correspondents
    - Each public key has the associated identifier, and some indifcation of how trusted the key is

### 8.7.2 S/MIME

- Provides authentication, data integrity, secrecy, and nonrepudiation
- Does not have a rigid certificate hierarchy, instead allowing users to designate multiple trust anchors

## 8.8 Web Security

### 8.8.1 Threats

### 8.8.2 Secure Naming

- DNS Spoofing
  - When Alice's ISP requests the IP for Bob's server, Trudy replies first with her own IP address, which then gets cached. This works because DNS responses aren't authenticated–DNS servers only check that the response claims to be from the right IP address
  - The only other difficulty for Trudy is getting her forged response to have the right sequence number to look like a response for Alice's request. This is doable if Trudy sets up her own name server: when trudy observes the request leaving Alice's ISP, she immediately issues a request to Alice's ISP for her own domain. When Alice's ISP contacts her DNS server, she knows its sequence number and can guess nearby values.
- SecureDNS/DNSSEC
  - Every DNS zone has a public/private key pair, and all information sent by a DNS server is signed
  - DNS records are grouped into RRSets (Resource Record Sets) such that all members of the set have the same name, class, and type. Each RRSet is hashed and signed with the zone's private key.
  - RRSets can be cached (with the hash as the key) even at untrusted locations since the whole thing is signed
  - New record types
    - KEY: holds the public key of some principal (zone, user, host), plus algorithm to use, protocol to use, and some other metadata
    - SIG: holds the signed hash according to the algorithm in the KEY record, as related to the records in the returned RRSet (including KEY records, but excluding itself)
  - RRSets are stored pre-signed, and in fact a zone's private key can be kept offline if you're willing to do RRSet signing in bulk
  - Introduces a dependency on getting trusted public keys: it is assumed that clients ship with public keys for all TLDs
  - Also introduces an optional antispoofing measure

### 8.8.3 SSL (The Secure Sockets Layer)

- Builds a secure connection between 2 sockets, including
  - Parameter negotiation between client and server

  - Authentication of the server by the client
  - Secret communication
  - Data integrity protection

- Two subprotocols: one for establishing a secure connection, the other for using it
- Connection Establishment

| | | | |
|---|---|---|---|
| A | $\rightarrow$ | B | SSL version, options/preferences, a nonce $R_A$ |
| A | $\leftarrow$ | B | SSL version, choices (based on options), a nonce $R_B$ |
| A | $\leftarrow$ | B | X.509 certificate chain to authenticate Bob to Alice |
| A | $\leftarrow$ | B | server done |
| A | $\rightarrow$ | B | $E_B(premaster\_key)$, where the premaster key is a randomly chosen 384 bit key |
| A | $\rightarrow$ | B | let's switch to the agreed-upon cipher |
| A | $\rightarrow$ | B | Done establishing connection |
| A | $\leftarrow$ | B | ok, switching cipher |
| A | $\leftarrow$ | B | I'm also done establishing the connection |

- Transport
  - Fragment into 16KB chunks
  - Compress
  - append(hash(compressed_data, secret_key)), for a secret key derived from the two nonces and the premaster key
  - Encrypt
  - Add a fragment header

- *TLS* (Transport Layer Security) is based on SSLv3, but they're not interoperable

### 8.8.4 Mobile Code Security

- *Mobile code*: Java applets, ActiveX controls, javascript
- Java applet security
  - aplets are code compiled to JVM (stack-oriented machine)
  - Applets are run by an interpreter which examines instructions before executing, meaning it can be stopped depending on default security policies and whether the particular applet is trusted or not.
  - Secure in principle, insecure in practice.

- ActiveX: x86 binary programs embeddable in webpages
  - You either run or you don't: not interpreted or sandboxed
  - Controls are signed: before executing, browsers check integrity of the code, then see if the creator is trusted

- Javascript
- Browser extensions
- Viruses

### 8.8.5 Social Issues

### 8.8.6 Privacy

- Anonymous remailers
  - An anonymous remailer that stores a mapping from a real email address to a pseudonym (a type 1 remailer) is bad because it's not subpoena-proof
  - *Cypherpunk remailers*: send a message with no From field, and with the message encrypted by the remailer's public key, to the remailer. The remailer agrees to keep no logs and will resend the content. These can be chained .
  - Extra safety precautions to thwart traffic analysis:
    - Remailer holds the message for a random duration, adds or removes junk to message, reorders messages in its queue

- *Onion routing*: similar to cypherpunk remailers

### 8.8.7 Freedom of Speech

- Steganography

### 8.8.8 Copyright

### 8.8.9 Summary