Data Communication Principles

Mahalingam Ramkumar
Mississippi State University, MS

January 22, 2018
Data Communication Networks

- Data (bits)
- Communication (conveyance of data)
- Network (the facilitator)
- Bits (packets) exchanged between computers;
- The network is the Internet
Grading Policy

- 50% Take Home Activities (Multiple choice / Quantitative/ Short essays)
- 25% Final Exam
- 15% Socket Programming Assignments
- 10% Participation
Expectations

- For advanced Juniors, Seniors and Graduate students
- Self-learning required:
  - socket programming
  - Wireshark, tcpdump, etc., for capturing and analyzing network packets
A Systems Course

- System: a collection of entities working together towards a common purpose/goals.
- What are the goals (of the Internet)?
- How does each component work? How do they work together?
- Only course where you actually study a large, complex, distributed system
- Can we make things better? Where are we headed?
Course Outline

- Overview of the Internet;
- Data Communication Principles
- Internet Practices
Data Communication Principles

- Communication Theory
- Delays and Switching
- Reliable Data Transmission
- Error Detection/Correction
- ARQ Protocols
- Shared media
Internet Practices

- Protocols in different Internet layers
  - DL/MAC layer protocols
  - IP protocol
  - TCP and UDP
  - Socket Library and Application Protocols
Why Digital?

- More Efficient
  - eg., analog vs digital TV
- Great equalizer — for transmission/reception, storage and processing
  - Does not matter if data is video, audio, image or document, or anything else
- Better suited for long range transmission
  - Can use repeaters
  - Why is this not suitable for analog transmission?
Terminology

- Protocol (rigid rules/steps to be followed)
- Bit-rate (Kbps, Mbps, Gbps)
- Storage (KB, MB, GB, TB)
- Useful to be comfortable with powers of 2
  - $2^{10} = 1024 \approx 1000$, $2^{20} = 1,000,000$ (million), $2^{30} \approx 1,000,000,000$ (billion)
- IP (Internet Protocol) address, 32-bit unsigned number
- For convenience represented as 4 1-byte (8-bit) numbers (each in the range 0 to 255)
- Example, 129.36.42.207.
Network Components

- Hosts (users of the network) & Network Infrastructure (provider)
- Postal Network: sender receiver; USPS
- Telephone network: caller and callee (subscribers); Telephone companies (network)
- Mobile Telephone: subscribers; service provider.
Internet Components

- Network Hosts & Network Infrastructure
- Host computers: Clients and servers
- Infrastructure: Internet service providers (ISP)
- ISPs in different tiers. Tier III ISPs rely on Tier II ISPs, and so on
- Tier I ISPs are “Backbone operators.”
- Some large users (e.g., Google, Netflix, Amazon, Yahoo) may connect directly to Tier I ISPs.
- Infrastructure components: Routers and Links (fiber, twisted pair, coax, wireless, satellite)
Classical Telephone Network

- Telephones connected by a wire (twisted pair copper) to a *local switching office* (up to 10000 phones)
- Several local switching offices are connected to an *area switching office* (up to 1000 LSO per ASO)
- Area switching offices are interconnected (1000 ASOs)
- Each wire (from a phone to a switching office) is associated with a unique (phone) number — eg. (662) 323-1234
  - area code 662 identifies area switching office
  - 323 identifies local switching office;
  - every wire terminating at switching office 323 has a unique four digit number xxxx.
Classical Telephone Network

- When the telephone receiver is lifted off the base the wire is activated (user hears a dial tone)
- User dials a destination number
- this is an instruction to the LSO to establish a path between the two phones
- 662 323 5678 ↔ 301 506 7259
- 662 323 5678 ↔ LSO 323 ↔ ASO 662 ↔ ASO 301 ↔ LSO 506 ↔ 301 506 7259
- the path can be used for sending electrical signals between the two phones.
Classical Telephone Network

- Mouthpiece converts sound vibrations (pressure) to an electrical signal
- Electrical signals conveyed over the wire (at the speed of light in copper) to the telephone at the other end.
- At the receiver, earpiece converts electrical signals to pressure vibrations (sound)
- Replacing the telephone on the hook terminates the connection.
- Telephone company keeps a record of the destination and duration of your call (for billing purposes).
Postal Network

- Users address mail/packages and drop them into boxes.
- Picked up by postal carriers
- Depending on the final destination the mail/package may be routed over several hops
- Possibly different modes of transportation at each hop (foot, bicycle, van, train, airplane, ···)
- Mail finally delivered to the destination address.
- Postal network uses **packet switching**. Telephone network uses **circuit switching**
Postal Network vs Telephone Network

- Postal network uses **packet switching**.
- Telephone network uses **circuit switching**.
The Internet at a Glance

- Hosts (computers), inter-connected routers (subnet);
- Every host has a unique IP address (32-bit address, about 4 billion unique addresses).
- Any host can send a packet (an IP packet) to any host.
- IP packets have a header indicating the IP addresses of the sender and the destination.
- IP packet also have a payload (any data that the sending host wants to send to the destination host)
- IP Packets are typically delivered over multiple hops — a router at each hop.
Routers and Links

- Every host has a direct connection to one router.
- A router can determine the best path to any other router.
- Does packet switching.
- At each link (between any two hops) IP packets are carried inside a data-link frame.
- The type of the data-link frame depends on the physical nature of the link (fiber, twisted pair, coax, wireless, satellite, etc.)
Postal Network vs Internet

- The Internet is more like the postal network (and less like the telephone network)
  - Both employ packet switching
  - Street Address $\leftrightarrow$ IP address
  - Postal envelopes $\leftrightarrow$ IP packets;
  - Post office sorting facilities $\leftrightarrow$ routers
  - transportation modes (road+truck) $\leftrightarrow$ link + data-link frames
Postal Network vs Internet

- Postal envelopes carry a letter from sender to receiver; IP packets carry an *application message*;
- different transportation mechanisms over each hop
  - truck over road, train over rails, ships over oceans/waterways etc.
  - different types of data-link frames over different types of physical layers;
  - Ethernet over twisted pair, Wireless Ethernet over air, ALOHA over satellite links, FTTH over fiber, etc.
- The main difference is the speed — IP packets travel at the speed of light
- The Internet is postal network on steroids!
Virtual Link Between Hosts

- The speed of exchanges motivates sophisticated Internet based applications
- Applications running on hosts use the Internet to communicate with applications running on other hosts (possibly even at the other side of the globe)
- From the perspective of applications, the Internet enables a virtual link between hosts.
Functional Organization — Internet Layers

- Application Layer (AL)
- Transport Layer (TL)
- Network Layer (NL)
- Data-Link Layer (DL)
- Physical Layer (PL)

Each layer has a specific function.

Components in one layer can be modified without affecting other layers.

All Top Notch Donut Places (useful? mnemonic)
Lower and Upper Layers

- The two lower layers (PL and DL) are necessary for the functioning of the NL (to create a virtual link between hosts).
- The two upper layers (TL and AL) are required to use the virtual link.
- “Sockets” in upper layers are bound to a specific transport port number and a specific network (IP) address.
- “Network interfaces” in lower layers are bound to a specific IP address and a DL address.
Sockets and Network Interfaces

- **Transport layer** manages multiple sockets
- Each socket (bound to a port number and IP address) corresponds to an application.
- **Network layer** may manage several network interfaces
- Each interface (bound to an IP address and data-link address) has its own lower layers (datalink + physical)
- **Physical Layer** — convert data-link frames to electrical signals for delivery over a physical medium
- **Data-link Layer** — preparing data-link frames (which carry IP packets) for efficient utilization of the physical layer (physical layer could be unreliable)
- **Network layer** — determines how IP packets will need to be routed, depending on the destination IP address
Upper Layers

- **Transport Layer** — converts the unreliable “virtual link” between two hosts (thanks to lower layers) into a reliable *connection*, to carry *application messages*.

- **Application Layer** — uses the reliable connection to send application messages (between an application running on one host and an application running on another host).
Apparent and Actual Data Paths

App. Message

AL  

AM  

AL
Apparent and Actual Data Paths

\[ AL \xrightarrow{TL} AM \xrightarrow{TH} AL \] App. Message

\[ TL \xrightarrow{TH} AM \xrightarrow{TL} \] Trans. Segment
Apparent and Actual Data Paths

App. Message

Net. Packet

Trans. Segment
Apparent and Actual Data Paths

- **AM** (App. Message)
- **TH||AM** (Trans. Segment)
- **NL** (Net. Packet)
- **NH||TH||AM**
- **TL** (Trans. Layer)
- **DL** (Data Layer)
- **DL Frame = DH||NH||TH||AM||DF**
Apparent and Actual Data Paths

**AL** \(\rightarrow\) **TL**

**NL** \(\rightarrow\) **NL**

**DL** \(\rightarrow\) **DL**

**PL** \(\rightarrow\) **PL**

**AM** \(\rightarrow\) App. Message

**TH||AM** Trans. Segment

**Net. Packet**

**DL Frame** = DH||NH||TH||AM||DF
Apparent and Actual Data Paths

- **AL** (App. Message)
- **TL** (Trans. Segment)
- **NL** (Net. Packet)
- **DL** (DL Frame)
- **PL** (Actual Path)

**Diagram Details**:
- **AM** (App. Message)
- **TH||AM** (Trans. Segment)
- **NH||TH||AM** (Net. Packet)
- **DL Frame = DH||NH||TH||AM||DF**
- **Virtual Link**
Apparent and Actual Data Paths

AM  App. Message

TH||AM  Trans. Segment

Net. Packet

NH||TH||AM

DL Frame

DL Frame = DH||NH||TH||AM||DF
Apparent and Actual Data Paths
Layers in Switches and Routers

Figure 1.24 Hosts, routers, and link-layer switches; each contains a different set of layers, reflecting their differences in functionality.
Physical Organization

- Several hosts connected to a router in the same local area network (LAN)
- LANs may be interconnected by routers to form small networks
- Small networks may be interconnected to form larger networks (for example, an organization)
- Organizations connect to an Internet Service Provider (ISP)
- ISPs may form several tiers
- Lower tier ISPs connect to higher tier ISPs
- ISPs at the highest tier interconnected by the Internet backbones.
Interconnection of Networks
Interconnection of Networks
Interconnection of Networks
Interconnection of Networks
Interconnection of ISPs
IP Addresses

- 32-bit unsigned number,
- for example, 10101110 01001000 11000010 11001001
- which is 2,924,004,041
- more conveniently represented as 174.72.194.201
- About 4 billion unique addresses
- Another example, 130.18.205.15 is actually

$$10000010 00010010 11001101 00001111 = 2,182,270,223 \ (1)$$
IP Address Chunks

- Typically, consecutive IP addresses assigned to hosts in the same LAN
- For example, a LAN with less than 128 (but greater than 64) computers can be assigned addresses 130.18.205.0 to 130.18.205.127, which can all be collectively represented as
  \[
  10000010 \ 00010010 \ 11001101 \ 0xxxxxxx 
  \]
  \[\text{(2)}\]
- First 32-7=25 bits are the same in all such addresses
- A short from for representing such collection is 130.18.205.0/25 (starting address, and the number of common bits for all addresses in the chunk).
- IP Prefix notation (to represent a chunk of \(2^r\) consecutive addresses, where \(r\) is an integer).
Assume an adjacent LAN has addresses in the range 130.18.205.128 to 130.18.205.255

Or prefix 130.18.205.128/25

$$\begin{align*}
10000010 & 00010010 & 11001101 & 1xxxxxxx 
\end{align*}$$ \hspace{1cm} (3)

130.18.205.0/25 and 130.18.205.128/25 can be consolidated as 130.18.205.0/24, or

$$\begin{align*}
10000010 & 00010010 & 11001101 & xxxxxxxx 
\end{align*}$$ \hspace{1cm} (4)
IP Aggregation

R1
130.18.205.0/25

R2
130.18.205.128/25

R3
130.18.205.0/24

R4
130.18.204.0/24
130.18.204.0/24

R5
130.18.204.0/23

130.18.204.0/24

Outline
Networks and Components
Layered Model
Data Communication Channels
Delays and Switching
Reliable Data Transfer

Functional Organization
Physical Organization
IP Aggregation

- 130.18.205.0/25
  - R1
  - 130.18.205.0/24
  - R3
  - 130.18.205.128/25
  - R2
- 130.18.204.0/23
  - R5
  - 130.18.204.0/24
  - R4
IP Aggregation

Outline
Networks and Components
Layered Model
Data Communication Channels
Delays and Switching
Reliable Data Transfer

Functional Organization
Physical Organization

130.18.205.0/25

130.18.205.128/25

R1

130.18.205.0/24

R3

130.18.205.0/24

R2

130.18.205.128/25

R4

130.18.204.0/24

130.18.204.0/24

R5

130.18.204.0/23

130.18.204.0/24
IP Aggregation

- R1: 130.18.205.0/25
- R2: 130.18.205.128/25
- R3: 130.18.205.0/24
- R4: 130.18.204.0/24
- R5: 130.18.204.0/23

Routing Table:

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Subnet Mask</th>
<th>Router</th>
</tr>
</thead>
<tbody>
<tr>
<td>130.18.205.0/25</td>
<td>25</td>
<td>R1</td>
</tr>
<tr>
<td>130.18.205.128/25</td>
<td>25</td>
<td>R2</td>
</tr>
<tr>
<td>130.18.205.0/24</td>
<td>24</td>
<td>R3</td>
</tr>
<tr>
<td>130.18.204.0/24</td>
<td>24</td>
<td>R4</td>
</tr>
<tr>
<td>130.18.204.0/23</td>
<td>23</td>
<td>R5</td>
</tr>
</tbody>
</table>
Autonomous Systems

- The Internet is an interconnection of *autonomous systems* (AS)
- ASes advertise IP prefixes to other ASes
- Every independent organization need not be an AS
- An AS may include any number of networks / organizations.
- What is the difference between an organizational network and an AS?
Interconnection of Autonomous Systems

Example Topology

Organization vs AS

- O1 · · · O3 are organizations under AS A1; (A1 advertises prefixes belonging to O1 · · · O3).
- ASes like A1 and A2 are ISPs;
- A3, A4, A5 are organizations that use ISPs like A1, A2.
- But they are also ASes because they advertise policies (reachability info) to other ASes.
- A4 and A5 have a “peering” arrangement
- Organization A3 uses two ISPs.
Network Layer Models

- TCP/IP Model (5 layers)
- OSI (Open Systems Interconnection) Model (7 layers)
  - Application
  - Presentation Layer
  - Session Layer
  - Transport (TCP / UDP)
  - Network (IP)
  - Datalink / MAC (Medium Access Control)
  - Physical layer
- Presentation and Session Layers missing in TCP/IP Model
- All People Seem To Need Data Processing (7-layer mnemonic)
Physical Layer

- Hardware for physically carrying data
- Over wires, or wireless links
- modems, Ethernet/Wifi card, etc.
- send a packet of bits from one computer to another \textit{when a direct connection exists between the computers}.
- Depends on the nature of the physical medium used.
- Two broad categories depending on if the physical medium is shared.
- For shared media DL protocols includes protocols for Medium Access Control (MAC).
  - Every computer that shares the physical medium should have a unique MAC layer address.
  - \textit{Contention} for channel access needs to be addressed.
Network Layer

- Provide a virtual link between two computers.
- To send a packet from one computer to another
- A unique *network address* for every computer
- Routers relay packets over *multiple hops*
- Internet protocol (IP) — every computer has a unique IP address
Application and Transport Layer

- Protocols for applications/processes running on different computers to communicate with each other
- Rely on the “virtual link” facilitated by the lower layers.
Application Layer

- Client-server model
  - (Client establishes a connection with the server)
  - Client sends a request
  - Server responds
  - (Close connection)

- Some applications: E-mail, WWW, IM, FTP, File sharing, ⋯
Transport Layer

- Provides a “reliable connection” between processes running on different computers
- Takes care of many low-level details for creating, maintaining and closing connections
- Different processes running on a computer differentiated by unique process addresses (port numbers)
Fundamental Factors

- Channel bandwidth $H$ Hz
- Signal power $S$ Watts
- Noise power $N$ Watts
Channel Capacity $R$

- Achievable bit-rate $R$
- Shannon’s Capacity Limit

$$ R = H \log_2(1 + S/N) \text{ bps.} $$  \hspace{1cm} (5)
Signal to Noise Ratio (SNR)
- Usually expressed in dB
- \((S/N)_{dB} = 10\log_{10}(S/N)\)
- 20 dB \(\rightarrow S/N = 100;\)
- 30 dB \(\rightarrow S/N = 1000;\)
Example

- What is the limit on achievable rate $R$ in a channel with bandwidth 1 MHz and SNR 30 dB?
  - $30 \text{dB} \rightarrow S/N = 1000$

  $$R = 1 \times 10^6 \times \log_2(1 + 1000) \approx 1 \times 10^6 \times 10 = 10 \text{Mbps} \quad (6)$$

- To double the rate to 20 Mbps, we can
  - Double the bandwidth, or
  - increase SNR to 60 dB (or increase $S/N \approx 1 \text{ million}$)
  - $\log_2(1,000,000) \approx 20$
Shannon-Nyquist Sampling Theorem

- Any signal “emerging” from a medium is bandwidth limited.
- Bandwidth - $H$ hertz (cycles/sec).
- A band-limited signal (limited to $H$ Hz) can be uniquely reconstructed from discrete samples taken at a rate $2H$ per second.
Assume each sample can take $V$ possible values
- $\log_2(V)$ bits per sample
- Nyquist capacity $C = 2H \log_2(V)$
- No of distinct recognizable values $V$ is determined by noise.
- If noise is zero, $V \to \infty$, so $C \to \infty$
- Every medium has inherent noise
- Even at absolute zero (0 Kelvin).
Transmission Media

- Two basic types
- Guided
  - Twisted pair
  - Coaxial cables
  - Fiber optics
  - Magnetic media
- Unguided
Twisted Pair

- Telephone cables
- Any wire is an antenna...
- Twisting substantially reduces interference
- Waves from different twists cancel out
- Number of twists per cm
- Categories 3 (16MHz), 5 (100 MHz), 6 (250), and 7(600)
- Ethernet cables

(a) (b)
Coaxial Cables

- Up to 1 GHz
- Twisted pair cables are catching up!
- Better noise immunity than twisted pairs
- Used for TV signals, cable TV...
Fiber Optics

- Extremely high bandwidth
- Single mode and multimode
- Single mode fibers can handle up to 50GHz over 100km!
Fiber vs Copper Wire

- Weight vs bandwidth
- Cost
- Security
Allocation of Spectrum

ITU-R, FCC
ISM (Industrial, Scientific, Medical) Bands

<table>
<thead>
<tr>
<th>Bandwidth</th>
<th>Freq.</th>
<th>902 MHz</th>
<th>928 MHz</th>
<th>2.4 GHz</th>
<th>2.4835 GHz</th>
<th>5.735 GHz</th>
<th>5.860 GHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>26 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>83.5 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>125 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Satellites

- Geo-stationary (about 36,000 km)
- Medium Earth Orbit (18,000 km, 6 hr orbits)
- Low Earth Orbit (less than 1000 km)
- Satellite vs Fiber
  - Remote / hostile areas
  - Point-to-point vs Broadcast
  - Mobile communications
  - “Right of way” for laying cables
Modulation

- Modulation: information signal + carrier signal $\rightarrow$ modulated signal
  - Speech signals have frequencies between 300 and 3000 Hz
  - Signals cannot propagate directly over all mediums
  - Speech signals $s(t)$ can be sent directly over a copper cable: but not over air or free-space
  - “Carrier” signals are used to carry the “information” signals

- Modulated signal transmitted over a medium
- Demodulation: extracting information signal from the modulated signal
- Types of modulation: Amplitude (AM), Phase (PM), Frequency (FM)
Bandwidth of Modulated Signals

- The same as the bandwidth of the “information” signal
- The bandwidth is now around a “center” frequency — the frequency of the “carrier”
- Many signals can coexist simultaneously in the medium.
- By using different carrier frequencies
Modulation for FAX/Dial-up Internet

- Send bits over telephone cable using Modem (Modulator - Demodulator)
- Cable meant for carrying voice signals between 300-3000 Hz
- Amplitude + phase modulation - series of sinusoidal pulses
- Modulator: Bits are chunked, represented using sinusoidal pulses
- Demodulator: Convert received sinusoidal pulse to bits
- Clipped sinusoid is the basic signal
- Variations achieved by modifying amplitude and phase
- Represented on a constellation diagram
QPSK, QAM-16, QAM-64

Quadrature Phase Shift keying (2 bits per sample), Quadrature Amplitude Modulation

(a) Quadrature Phase Shift keying (2 bits per sample)
(b) Quadrature Amplitude Modulation QAM-16
(c) Quadrature Amplitude Modulation QAM-64
Telephone/FAX Modem

- Each pulse represents multiple bits
- If we use 16 different types of pulses (eg., QAM 16) we can send 4 bits per pulse ($16 = 2^4$)
- Bit rate and Baud rate
  - Baud rate - number of sinusoidal pulses per second
  - Fixed at 2400 for telephone lines
  - Bit rate = Baud rate \times \text{number of bits per pulse}
  - How do we get 56 K with baud rate of 2400?
Trellis Coded Modulation

Add more bits per sample
Add extra bits to each sample for error correction
V.32 (4+1, 32) - 9600 bps, V.32 bis (6 + 1, 128) 14,400 bps
V.34 - 28,800 bps, V.34 bis - 36,600 bps
Modern Modems

- **Full duplex** - transmissions possible in both directions at the same time
- **Half-duplex** - Both directions, but not at the same time
- **Simplex** - Only one direction
- **V.90, V.92** - full duplex
- Dedicated uplink and downlink channel
- 56 kbps downlink, 33.6 kbps uplink (V.90)
- 48kbps uplink (V.92) + facility to detect incoming calls while online
Actual Capacity of a Telephone Line

- 50 Mbps
- 40 Mbps
- 30 Mbps
- 20 Mbps
- 10 Mbps
- 0 Mbps

Distance in Meters: 0, 1000, 2000, 3000, 4000, 5000, 6000
We have been sitting on a gold-mine!

- Telephones filter out higher frequencies to suppress noise
- 1.1 MHz spectrum divided into 256 channels - 4312 Hz each (DMT - Discrete Multitone)
- First channel used for POTS (plain old telephone service)
ADSL Equipment

Voice switch → Codec → Splitter → DSLAM → To ISP

Telephone company end office

Telephone line

NID

Customer premises

ADSL modem → Ethernet

Computer

To ISP

ADSL modem

Ethernet

Computer

Telephone

Splitter

Voice switch → Codec → Splitter → DSLAM → To ISP

Telephone company end office

Telephone line

NID

Customer premises

ADSL modem → Ethernet

Computer

To ISP

ADSL modem

Ethernet

Computer
Multiplexing and Demultiplexing

- **Multiplexing** - many to one
  - Many signal mixed together to produce one signal. The multiplexed signal is transmitted over a channel

- **Demultiplexing** - one to many
  - Multiplexed signal is split into individual components

- **Types**: FDM, TDM, CDM
Frequency division multiplexing (FDM)

Telephone - 4000Hz per channel (450 + 3100 + 450)

(a) Frequency (Hz)

(b) Frequency (kHz)

(c) Channel 1 Channel 2 Channel 3

Frequency (kHz)
Wavelength Division Multiplexing (WDM)

Fiber 1 spectrum

\[ \lambda \]

Fiber 2 spectrum

\[ \lambda \]

Fiber 3 spectrum

\[ \lambda \]

Fiber 4 spectrum

\[ \lambda \]

Spectrum on the shared fiber

\[ \lambda \]

Power in Fiber 1, Fiber 2, Fiber 3, and Fiber 4

\[ \lambda_1, \lambda_2, \lambda_3, \lambda_4 \]

Combiner

\[ \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 \]

Long-haul shared fiber

Splitter

\[ \lambda_1, \lambda_2, \lambda_3, \lambda_4 \]

Filter

\[ \lambda_2, \lambda_4, \lambda_1, \lambda_3 \]
Time Division Multiplexing (TDM)

- $N$ signals, each having $M$ samples per sec: $NM$ interleaved samples per sec
- POTS signal (speech or clipped sinusoids) sampled at 8000 1-byte samples per sec (64000 bps)
- T1 lines - 1.544 Mbps, 24 channels
TDM in Telephone Network

Telephone → End office → Toll office → Intermediate switching office(s) → Toll office → End office → Telephone

- Local loop
- Toll connecting trunk
- Very high bandwidth intertoll trunks
- Toll connecting trunk
- Local loop
T1, T2, T3, T4

- **T1** (1.544 Mbps): 4 T1 streams in, 7 T2 streams in, 6 T3 streams in, 1 T2 stream out.
- **T2** (6.312 Mbps): 4:1 multiplexing, 7:1 multiplexing.
- **T3** (44.736 Mbps): 6 T3 streams in, 7 T2 streams in.
- **T4** (274.176 Mbps): 6:1 multiplexing.
Code Division Multiple Access (CDMA)

- TDM - different channels can overlap in frequencies, but not in time
- FDM - no overlap in frequencies, overlap in time
- CDMA - overlap in both time and frequency
- Separable by orthogonality of codes
- Two vectors are orthogonal if their inner-product is zero.
CDMA

- Each bit is converted to a vector of chips
- Different users assigned different orthogonal chip vectors
- A user assigned vector $\vec{x}$ transmits $\vec{x}$ to send a one or $-\vec{x}$ to send a zero
- $\vec{x}.\vec{x} = 1, \vec{x}.(-\vec{x}) = -1$
- $\vec{x}.\vec{y} = 0$ if $\vec{y}$ is orthogonal to $\vec{x}$.
- In practice difficult to obtain strict orthogonality
- $\vec{x}.\vec{y}$ can be made very small compared to 1
- “Tolerable” interference with quasi-orthogonal sequences
- CDMA is also more efficient (closer to Shannon’s limit)
Why CDMA?

- If $n$ chips per bit, then chip frequency $n$ times higher than bit-frequency
- Energy to sent a bit spread over a larger frequency
- Useful during war-time as transmission virtually indistinguishable for noise in any narrow band.
- CDMA is also more efficient
- Higher frequency and lower $S/N$ takes us closer to Shannon’s limit
- Capacity increases linearly with bandwidth (only logarithmically with $S/N$)
Characterizing Data Transmission

- Bit-rate $R$, transmission delay $\tau_p$
- Propagation speed $c$, propagation delay $\tau_c$
- Round trip time (RTT)
Packet Duration

- Bit duration is the inverse of the bit-rate $R$
- If bit rate is 1 Mbps, the bit duration is $1 \mu s$
- $1 \text{ Kbps} \leftrightarrow 1 \text{ ms}$, $1 \text{ Gbps} \leftrightarrow 1 \text{ ns}$.
- Transmission delay is bit-duration times packet size (number of bits in the packet)
- $10 \text{ Mbps}$ bit-rate, 100 byte packets:
  \[ \tau_p = 100 \times 8 \times 0.1 = 80 \mu s. \]
- $1 \text{ Gbps}$ bit-rate, 100 byte packet:
  \[ \tau_p = 100 \times 8 = 800 ns = 0.8 \mu s. \]
Total Delay

- propagation at the speed $c$ (light speed in the medium)
- $c = 3 \times 10^8$ m/s in free-space and air, $c = 2.5 \times 10^8$ m/s in copper.
- Distance traveled in 1 $\mu$s is 250 m (25cm in 1 ns)
- Over a 10 Mbps line of length 2500 m, how long does it take to receive a 100 byte packet?
  - propagation delay is $\tau_c = 10\mu s$
  - bit duration is 0.1$\mu s$; transmission delay
    $\tau_p = 100 \times 8 \times 0.1 = 80\mu s$ at time $t = 0$
  - leading edge of the packet sent at $t = 0$; trailing edge sent at time 80$\mu s$
  - leading edge arrives at $t = 10\mu s$; trailing edge arrives at $t = 90\mu s$
Total Delay

Pictorial Representation

Total Delay

- Total delay = Propagation delay $\tau_c$ + transmission delay $\tau_p$
- $y$ axis is time
- $x$ axis represents distance
- the slope represents speed of propagation
Repeaters

- Used in long transmission lines
- Converts electrical signals to bits, and back to electrical signals, for retransmission
- Repeaters placed strategically to ensure that signal strength received at repeaters is strong enough to eliminate errors
- The conversion process eliminates the effects of noise between repeaters or between repeaters and end-points.
- Repeaters introduce additional propagation delay
- For a line of length 2500 m, with four repeaters (assume each repeater introduces a delay of $4 \mu s$) the propagation delay is $10 + 4 \times 4 = 26 \mu s$. 
Repeaters do not affect transmission delay.

Repeaters do not have to receive the entire packet before starting the conversion.

Why not repeaters for analog communications?
Processing Delay

- It takes a finite amount of time for the receiver to “process” the received packet (say, $\tau_r$).
- Data-link frames typically have a cyclic redundancy check (CRC) that will be verified (if inconsistent drop the corrupted packet).
- Routers will need time to look-up routing tables to determine the best next hop.
- There may also be *queuing* delay at routers.
- Routers may have several incoming and outgoing interfaces.
- A queue in an incoming interface may be stuck if the outgoing interface for the packet at the head of the queue is busy.
Typically, every packet is acknowledged.

Round trip time (RTT) = trans. delay $\tau_p$ + propagation delay $\tau_c$ + processing delay $\tau_r$ + trans. delay for ack $\tau_p'$ + ack propagation delay $\tau_c$ + ack processing delay $\tau_r'$.

A decent approximation is $\text{RTT} = 2 \times (\tau_p + \tau_c + \tau_r)$

What is the RTT over a 10 Mbps line of length 2500 m, if the packet size (both sent and ACK) is 100 bytes, and the processing delay is 5 $\mu$s?

For the same length, packet size, and processing delay, what is the RTT if the bit-rate is 1 Gbps?
Multi-hop Propagation

One way propagation delay for a channel with \( n \) hops is

\[ n(\tau_c + \tau_p + \tau_r) \]
A Simpler Representation of Delays

**Representation**

- Forward path AB, reverse path BC
- \( \tau = \tau_c + \tau_p + \tau_r \) total forward path delay (\( \tau' = \tau'_c + \tau'_p + \tau'_r \) total reverse path delay)
- In the simpler representation only the lines AB and BC will be depicted
- Also often rotated by 90 degrees (time in X-axis)
Switching

- Circuit Switching
- Cut-through Switching
- Store and Forward Switching

Circuit switching was used in Telephone networks (not any longer)
Pkt vs Ckt Switching

\[ \tau_{CS} = 3\tau_c + \tau_p + \tau_r \]
\[ \tau_{PS} = 3\tau_c + 3\tau_p + 3\tau_r \]

For \( n \) hops: \( \tau_{PS} = n(\tau_c + \tau_p + \tau_r) \), \( \tau_{PS} = n\tau_c + \tau_p + \tau_r \)
Circuit Switching

- Physical path established over multiple links
- Circuit established at the physical layer level
- Propagation delay is the sum of propagation delays over each link
- Transmission delay is *not* affected
- Was used in Telephone networks (not any longer)
Cut-through Switching

- Begin forwarding a packet to next hop even before the entire packet has been received
- Leading bytes in the packet will indicate destination
- Useful when look-up for destination can be accomplished very fast
- Delay for multi-hop propagation similar to the situation when repeaters are used.
Store and Forward or Packet Switching

- Entire packet received before the packet can be sent to next hop
- Delay at each hop is transmission delay + propagation delay + processing delay
- Total delay is the delays at each hop
- Packet duration has a significant impact on total delay.
- Two types: Virtual Circuits and Datagram Switching
Datagram Switching

- Routers maintain a routing table indicating best next hop for each destination.
- Routing tables are dynamic.
- Every packet routed individually.
- Paths may change dynamically.
- Packets not guaranteed to be received in the same order.
Virtual Circuit Switching

Virtual Circuit
- A path is established before even the first packet can be sent
- Path accepted by all routers in the path
- Packets marked with a path identifier
- Path identifier helps the router determine how to forward the next packet
- Size of routing tables in a router depends on number of active paths through the router.

VC Subnet

A path is established before even the first packet can be sent.
Path accepted by all routers in the path.
Packets marked with a path identifier.
Path identifier helps the router determine how to forward the next packet.
Size of routing tables in a router depends on number of active paths through the router.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>Process P1</td>
</tr>
<tr>
<td>H3</td>
<td>Process P3</td>
</tr>
<tr>
<td>A</td>
<td>Carrier's equipment</td>
</tr>
<tr>
<td>B</td>
<td>Router</td>
</tr>
<tr>
<td>C</td>
<td>A's table</td>
</tr>
<tr>
<td>D</td>
<td>C's table</td>
</tr>
<tr>
<td>E</td>
<td>E's table</td>
</tr>
<tr>
<td>F</td>
<td>Process P2</td>
</tr>
</tbody>
</table>

Routing tables:

- A's table:
  - H1: 1  C1: 1  A: 1  E: 1  C: 1  F: 1
- C's table:
  - H1: 2  C2: 2  A: 2  E: 2  C: 2  F: 2
- E's table:
  - A: 3  C: 3  E: 3  F: 3

LAN connection.
## Datagram vs VC

<table>
<thead>
<tr>
<th>Issue</th>
<th>Datagram subnet</th>
<th>Virtual-circuit subnet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circuit setup</td>
<td>Not needed</td>
<td>Required</td>
</tr>
<tr>
<td>Addressing</td>
<td>Each packet contains the full source and destination address</td>
<td>Each packet contains a short VC number</td>
</tr>
<tr>
<td>State information</td>
<td>Routers do not hold state information about connections</td>
<td>Each VC requires router table space per connection</td>
</tr>
<tr>
<td>Routing</td>
<td>Each packet is routed independently</td>
<td>Route chosen when VC is set up; all packets follow it</td>
</tr>
<tr>
<td>Effect of router failures</td>
<td>None, except for packets lost during the crash</td>
<td>All VCs that passed through the failed router are terminated</td>
</tr>
<tr>
<td>Quality of service</td>
<td>Difficult</td>
<td>Easy if enough resources can be allocated in advance for each VC</td>
</tr>
<tr>
<td>Congestion control</td>
<td>Difficult</td>
<td>Easy if enough resources can be allocated in advance for each VC</td>
</tr>
</tbody>
</table>
Connectionless vs Connection-Oriented

- Connectionless: Snail mail, telegram, etc.
- Establishing connections makes the job of end points easier.
- Circuit switching established a physical layer connection
- Virtual circuits are established at the network layer level
- In datagram-routing connection can be established at a higher level (transport layer).
Latency Example (Assume 1µs processing delay)

- \( R = 100 \text{ kbps} \) Audio channel
- 10 hops, each 500 Km; propagation time each hop 2000µs
- packet size 1000 bytes, trans. delay \( \tau_p = 80,000 \mu s \)
- Packet switching: delay
  \[ 10 \times 2ms + 10 \times 0.001ms + 10 \times 80ms \approx 820ms \]
- Circuit switching: delay =
  \[ 10 \times 2ms + 1 \times 0.001ms + 1 \times 80ms \approx 100ms \]
- If packet size is 100 bytes?
  - packet switching: delay \( 20 + 0.01 + 10 \times 8 \approx 100ms \)
  - circuit switching: delay \( 20 + 0.001 + 8 \approx 28ms \)
- What if channel rate is 100 Mbps? (Homework)
To realize low latency channels

- For circuit switching and cut-through switching packet duration is not an issue.
- For packet switching use smaller packet sizes - especially in low-bit rate channels.
- In high bit-rate links packet duration (transmission delay) can be negligible (compared to propagation delay).
- With ever increasing bit-rates the advantages of Circuit Switching and Cut-through Switching become less relevant.
Reliable Data Transfer

- Reliably transmit a sequence of packets over a link.
- This is a requirement for data-link and transport layers
- Link can be real (data link layer) or virtual (transport layer)
- Packets can be corrupted (errors)
- Over a virtual link packets can also arrive out of order, be duplicated, dropped, · · ·
- How does the sender know if the receiver got a specific packet?
- How does the receiver know that the sender knows that the receiver knows? (ARQ Protocols)
Errors Are Unavoidable

- Need to detect errors
- correct errors / request retransmission
- The key: **redundancy** — extra bits need to be transmitted for detecting / correcting errors
- Example: parity bits for error detection
Hamming Distance

- A metric for measuring “distances” between two sequences of bits
- $A = 1000110$
- $B = 1010100$
- How many bits need to be flipped to get $B$ or $A$ (or vice-versa)?
- Two bits — the Hamming distance between $A$ and $B$ is 2.
Error Detection with Parity Bit

- \( A = 1000110 \)
- With even parity \( A = 10001101 \)
- With odd parity \( A = 10001100 \)
- For all subsequent discussions we will only use even parity
- If any of the eight (7+1) bits of \( A \) is flipped during transmission parity check will fail.
- What happens if two bits get flipped?
- Parity check fails to detect error...
- With one redundant bit we can only detect one-bit errors.
- Actually any odd number of bit-errors
Redundancy

- Need to transmit $m$ bit symbols ($2^m$ possible symbols)
- $r$ redundant bits (for error detection / correction)
- $n = m + r$ bits actually transmitted.
- Efficiency is $\frac{m}{m+r} = \frac{m}{n}$
- In general, more the redundancy, more the errors that can be detected / corrected
A Simple-Minded Approach

- Just repeat every bit twice (efficiency 0.5) - any one bit error can be detected
- Parity bit (just add one extra bit) - efficiency $\frac{m}{m+1}$ is high for large $m$.
- How can we correct single bit errors automatically?
- Repeat each bit two more times (efficiency 0.33)
- Are there better ways?
Error Detection vs Correction

- Error correction needs more redundancy
- Error detection is much more crucial than error correction — why?
- Practical error detection techniques address the problem of detecting multiple-bit errors.
- Error correction is usually used only for correction of single bit errors.
Let us assume that the probability that any bit may be erroneously received is $p = 10^{-9}$.

If $N = 10^3$ bits are transmitted, probability that there may be a bit error is roughly one in a million.

Probability that there may be two errors is roughly one in a trillion (million x million)

We need to detect even rare errors — but not a big deal if we can not correct it (request re-Tx).
Our Focus

- **How do we correct one-bit errors?**
  - If the receiver is able to correct errors, the sender does not need to retransmit

- **How do we detect multiple-bit errors?**
  - If error is detected, the receiver can request the sender to retransmit
  - Or simply *not* acknowledge reception
(4, 7) Hamming Code

- 4 information bits; 3 parity bits.
- $b_7 \ b_6 \ b_5 \ b_4 \ b_3 \ b_2 \ b_1$
- $b_1$ is the parity bits for bits $b_7, b_5, b_3$ (all odd positions 7,5,3,1)
- $b_2$ is the parity for bits $b_3, b_6, b_7$ (positions 2,3,6,7)
- $b_4$ is the parity for bits $b_5, b_6, b_7$ (positions 4,5,6,7)
- What is special about 1,2, and 4? Powers of 2 ($2^0, 2^1, 2^2$).
- Assume we want to send $m = 4$ bits 1011
- $b_7 = 1 \ b_6 = 0 \ b_5 = 1 \ b_4 = \ ? \ b_3 = 1 \ b_2 = \ ? \ b_1 = \ ?$
- $b_1 = 1, \ b_2 = 0, \ b_4 = 0$ are the correct parities
- The code for 1011 is 1010101
(4, 7) Hamming Code

- The receiver checks parity bits and writes down the results of the check as $x_4x_2x_1$
- $x_4 = 0$ if parity $b_4$ is correct; $x_4 = 1$ if parity $b_4$ is wrong (and similarly for $x_2$ and $x_1$)
- If no bit is flipped by the channel then all parity bits will be correct (result 000)
- If $b_7$ is flipped all three parities will be wrong (result 111)
- If $b_3$ is flipped result is 011
- If $b_5$ is flipped result is 101
- See a pattern? The result gives the position of the erroneous bit.
- Go ahead and flip it back.
Hamming Code Example

- Code $X = 1010101$
- If $Y = 1010101$ (no error) result 000
- If $Y = 1010100$ (pos 1 error) result 001
- If $Y = 1010111$ (pos 2 error) result 010
- If $Y = 1010001$ (pos 3 error) result 011
- If $Y = 1011101$ (pos 4 error) result 100
- If $Y = 1000101$ (pos 5 error) result 101
- If $Y = 1110101$ (pos 6 error) result 110
- If $Y = 0010101$ (pos 7 error) result 111
For efficient codes $n = 2^r - 1$: example
(4, 7), (11, 15), ... (1013, 1023), ... Why?

If only one bit can be flipped, we have $n + 1$ possible outcomes when we transmit a $n$ bit value (?)

The $r$ parity bits help identify which of the $n + 1$ possible outcomes is true

So we need $2^r \geq n + 1$;

$2^r = n + 1$ for efficient codes.
**Hamming Code**

Code $b_{15} \cdots b_1$

$b_1, b_2, b_4, b_8$ are parity bits

Example $0\ 1\ 0\ 1\ 0\ 1\ 0\ ?\ 0\ 0\ 1\ ?\ 0\ ?\ ?\ ?$ (11-bit information $01010100010$)

<table>
<thead>
<tr>
<th>$b_{15}$</th>
<th>$b_{14}$</th>
<th>$b_{13}$</th>
<th>$b_{12}$</th>
<th>$b_{11}$</th>
<th>$b_{10}$</th>
<th>$b_9$</th>
<th>$b_8$</th>
<th>$b_7$</th>
<th>$b_6$</th>
<th>$b_5$</th>
<th>$b_4$</th>
<th>$b_3$</th>
<th>$b_2$</th>
<th>$b_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>?</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>?</td>
<td>0</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>?</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>?</td>
<td>0</td>
<td>?</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>?</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Result written as $x_8x_4x_2x_1$.

If bit 13 is flipped result is 1101
Syndrome Coding

- A syndrome is a “collection of symptoms.”
- Each result of parity check is a symptom: each bit of the result (for example 1011 if $r = 4$ is a symptom)
- The syndrome should *unambiguously* indicate the error.
- With $r$ redundant bits, we have $2^r$ unique syndromes
- We can detect $2^r - 1$ different “diseases” (one syndrome corresponds to “no illness”).
- $n \leq 2^r - 1$
Error Detection

- Single error detection with parity bit
- Burst error detection: Interleaving
Multi-bit Error Detection

- Double errors
- arrange $n = w \times h$ bits as a matrix
- Add parity bit for each row (we now have $(w + 1) \times h$ matrix)
- Parity bit for each column — we end up with a $(w + 1) \times (h + 1)$ matrix
- If 2 errors happen in the same row (column) they cannot be in the same column (row)
Cyclic Redundancy Checks (CRC)

- A polynomial \( y = P(x) = x^r + a_{r-1}x^{r-1} + \cdots + a_1x + a_0 \)
- If \( P(x) \) is in the field of real numbers, \( x, a_0 \ldots a_{r-1}, y \) can be any real number
- Any bit string can be seen coefficients of a polynomial in the finite field \( \{0, 1\} \)
- \( P(x), x, a_0 \ldots a_{r-1} \) can only be 0 or 1
- Example: 110101 is \( P(x) = x^5 + x^4 + x^2 + x^0 \).
- Addition: \( 1 + 1 = 0 = 0 + 0, 1 + 0 = 0 + 1 = 1 \)
- Subtraction is the same as addition!
- Multiplication: \( 1 \times 1 = 1, 0 \times x = 0 \).
- \( P(0) = 1. \ P(1) = 1 + 1 + 1 + 1 = 0. \)
Polynomial Arithmetic

- \( P(x) = x^5 + x^4 + x^2 + 1 \). \( Q(x) = x^2 + x^1 \)
- \( P(x) + Q(x) = x^5 + x^4 + x^1 + 1 \)
- \( P(x) \times Q(x) = (x^5 + x^4 + x^2 + 1)(x^2 + x^1) = x^7 + x^6 + x^4 + x^2 + x^6 + x^5 + x^3 + x^1 = x^7 + x^5 + x^4 + x^3 + x^2 + x^1. \)
- Polynomial division - what is \( P(x)/Q(x) \)

\[
\begin{array}{c|cccc|c}
110 & 110101 & 1001 & (quotient) \\
110 & & & & \\
\hline
110 & 000101 & & & \\
\hline
110 & & & & \\
\hline
011 & (remainder) & & & \\
\end{array}
\]
Choose a generator polynomial $G(x)$ of degree $r$ ($r + 1$ bit number).

Message to be sent $M(x)$

Append $r$ zeros to $M(x)$. The result is $x^rM(x)$.

Evaluate the remainder of $x^r M(x)/G(x)$ (remainder will be a polynomial of degree less than $r$) - say $R(x)$

Transmit $T(x) = x^r M(x) - R(x) = x^r M(x) + R(x)$

Receiver receives $T(x)$

In case of error receiver gets $T'(x) = T(x) + E(x)$

Receiver checks if remainder of $T'(x)/G(x)$ is zero

If remainder is not zero receiver decides that there is an error

As $T(x)/G(x) = 0$, $T'(x)/G(x) = E(x)/G(x)$

Errors where $E(x)/G(x)$ is zero goes undetected
Good choice for $G(x)$

- Both highest and lowest order bits should be 1 (1xx · · · xx1)
- If $G(x)$ has degree more than 1 all single bit errors will be detected - $E(x) = x^i$.
- Two isolated single bit errors $E(x) = x^i + x^j = x^j(x^{i-j} + 1)$: any $G(x)$ that does not have $x^k + 1$ as a factor (for all $k \leq m$) is sufficient
- If $x + 1$ is not a factor, then all odd number of errors can be detected.
- Any burst error of length $\leq r$ can be detected
- 32 bit polynomial specified in IEEE 802x
ARQ (Automatic Repeat Request) Protocols

- Packets need to be acknowledged
- Time-out for retransmission
- But acknowledgements can get lost too!
Can both sender and receiver agree that $P1'$ is a retransmission of $P1$ and not $P2$?

(A)

Timeout Interval

P1
ACK
ACK

P2

(B)

Timeout Interval

P1
X
ACK
ACK

P1'

ACK
Packets need to be numbered
Obviously numbers will be “recycled”
How many unique sequence numbers do we need?
Ambiguous Acknowledgements

The ACK was P1 was delayed. Sender retransmits P1. Soon after, ACK for P1 is received and sender sends P2 which gets lost. How does the sender know that the second acknowledgement is for P1 or P2?

Timeout Interval

(C)
Acknowledgements need to be numbered too!

- ABP protocol (alternating bit protocol)
- One bit packets numbers - 0 or 1
- One bit ACK numbers
- Also called Stop and Wait Protocol (SWP)
- Is ABP/SWP unambiguous?
Alternating Bit Protocol (ABP)

Diagram:
- Timeout Interval
- Packet No
- Seq No
- Ack No
- Packet No

(D)
ABP is Unambiguous!

(E) Timeout Interval

(Packet No) 0 1 0 1 0 1
(Seq No) 0 1 X 1 0
(Ack No) 0 1 2

(F) Timeout Interval

(Packet No) 0 1 0 1 0 2
(Seq No) 0 1 X 1
(Ack No) 0 1 2

ABP is Unambiguous! The diagram illustrates the correct operation of the Alternating Bit Protocol (ABP) with examples (E) and (F) showing how the protocol ensures unambiguous data transfer.
ABP Sender Rules

- If the last packet sent was numbered 0,
  - allowed to send the next packet with number 1 only after ACK for 0 is received
  - (else retransmit 0)
- If the last packet sent was numbered 1,
  - allowed to send the next packet with number 0 only after ACK for 1 is received
  - (else retransmit 1)
ABP Receiver Rules

- If last packet received was numbered 0
  - If 1 received ACK and keep (new packet)
  - If 0 received ACK and drop (retransmission of previous packet)
- If last packet received was numbered 1
  - If 0 received ACK and keep (new packet)
  - If 1 received ACK and drop (retransmission of previous packet)
Effective Data Rate

- 100 Mbps link $A \leftrightarrow B$ ($R = 100 \times 10^6$), $L = 2000m$, packet size $F = 100$ bits, $\tau_r = 1 \mu s$
- $P = \tau_p = \frac{100}{100000000} = 1 \mu s$, $\tau_c = \frac{2000}{2.5 \times 10^8} = 8 \mu s$
- What is the effective data rate between $A$ and $B$ when ABP is used?
- $\text{RTT} = 2\tau_p + 2\tau_c + 2\tau_r = 2 + 16 + 2 = 20$.
- In each $20 \mu s$ interval transmission occurs only for $1 \mu s$
- Effective data rate is $5$ Mbps ($P/\text{RTT} \times R$)
Improving Throughput

- During the first micro second sender sends the first packet
- Sender does nothing for 19 $\mu s$. ACK for first packet received after 20$\mu s$
- What if sender is allowed to send 20 packets before receiving the ACK for the first packet?
- Only after the ACK for the first packet is received the sender can send packet 21
- Only after the ACK for the second packet is received can the sender send packet 22, and so on
- Window size of 20 packets
- More generally, Window size $\geq$ RTT/packet_duration
Pipelining Example, $W \geq \frac{RTT}{\tau_p} = 13$
Pipelining Example, \( W \geq \frac{RTT}{\tau_p} = 13 \)
Pipelining Example, $W \geq \frac{RTT}{\tau_p} = 13$
Pipelining Example, $W \geq \frac{RTT}{\tau_p} = 13$

\[
\begin{array}{cccccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\
\end{array}
\]
Pipelining Example, $W \geq \frac{RTT}{\tau_p} = 13$

13 packets (0 to 12) sent in one RTT

ACK for 0 processed; ready to send 13
Pipelining Example, \( W \geq \frac{RTT}{\tau_p} = 13 \)

13 packets (0 to 12) sent in one RTT

ACK for 0 processed; ready to send 13

ACK for 1 rcd; ready to send 14
Pipelining Example, $W \geq RTT/\tau_p = 13$

13 packets (0 to 12) sent in one RTT

ACK for 0 processed; ready to send 13

ACK for 1 rcd; ready to send 14
Pipelining

- The secret to higher throughput
- ABP needs only half-duplex channels, for pipelining we implicitly assume full duplex channels
- ACKs and packets cross each other
- Nodes may need to buffer packets when things do not go well
- What is the buffer size needed?
- How do we number packets and acknowledgements?
Selective Repeat Protocol

- Similar to ABP, with window size $W$ packets
- Or ABP is SRP with window size 1 packet
- Buffer size ($W$. Why?)
- Numbering packets? ($2W$ unique numbers. Why?)
SRP Protocol

Sender Window (size 5)

- Not yet sent
- ACK pending
- Sent and ACKed

SRP Sender Rules

- Legal to send any packet within the window
- Head of the queue blocked by an outstanding ACK.
- Sender window can not advance until ACK is received for the blocking packet.
- Normally next unsent packet (blue) is sent.
- On time-out for the blocking red packet resend the packet.
- Sender may have to buffer up to $W$ packets.
SRP Window

Receiver Rules

- Past Window and Future Window; each of size $W$.
- Packets left of the Past Window: Receipt of ACK confirmed by sender
- Packets in the Past Window: ACKs not confirmed by sender
- Future Window blocked by earliest pending packet.
- Next packet received can be from either window
- If packet from past window ACK and drop packet
- If packet from future window ACK and store packet
- Past Window is the lower bound on sender window position

Past Window

Future Window

Rx. Window

- received
- pending

p.w f.w

3 4 5 6 7 8 9 0 1 2 3 4
Receiver Rules

- Up to $W$ packets may need to be buffered.
- $2W$ possibilities for next packet. Using $2W$ distinct numbers eliminates ambiguities.
- If next received packet is numbered 4, 5, 6, 7 or 8 ACK and discard.
- If next received packet is numbered 9, 0, 1, 2 or 3 ACK and store.
Go Back $N$ - GBN Protocol

- Window size $W$
- Sender logic is the same
- Receiver does not buffer packets (no window)
- Out of sequence packets are simply discarded
- Periodically sends acknowledgement for the last received packet number
- Example, if Rx gets packets 1,2,3,4,5,7,8,9 the ACK for 5 is sent. 7,8,9 not ACKed.
- Sender retransmits 6,7,8,9...
Windowing Protocols in DL vs TL

- Window size need not always be expressed in packets
- If expressed in packets we assume equal duration packets (window size is “number of packet durations in one RTT”)
- It can be in bits too (“number of bits that can be sent in one RTT”)
- Or bytes (“number of bytes that can be sent in one RTT”)
- In transport layer window size is expressed in bytes
- In data link layer, the link parameters like $\tau_p$, $\tau_c$, $\tau_r$, RTT etc., are known at design time. (Easy to compute $W \geq RTT/\tau_p$).
- In TL how well do we know the “virtual link” parameters? How do we choose window size?
TDM in Autonomously Shared Channels

- Multiple *autonomous* senders (stations) using the same medium
- *Contention* for medium access
- *Collision* occurs when packets from any two stations overlap
- Strategies for medium access control (MAC) used by the data link layer
To Reserve Or Not?

- Scheduling can be done with/without making reservations
- No reservation
  - Stations themselves decide when to send
  - Collisions can occur
- Reservation based
  - Reservation slots
  - Collision-free
Three Scenarios in No-Reservation MAC

- Classification based on the nature of the medium.
  - Can a station *sense* on-going transmissions from *other* stations?
  - Can occurrence of collisions be *detected*? (as soon as collision occurs)
- Scenario 1: Stations contending for the medium can *not* sense transmissions from other stations (eg., Satellite communication)
- Scenario 2: Stations *can* sense transmissions by other stations, but can *not* detect collisions (can not listen and transmit at the same time).
- Scenario 3: Stations can sense the medium, and *can* detect collisions (eg. Ethernet).
ALOHA

- $n$ stations share a medium
- Stations transmit whenever a packet arrives (from the higher layer)
- ACK sent by the receiver (satellite).
- (No issues in transmissions from satellite to stations. Why?)
- If no ACK received, stations wait for a random amount of time and retransmit.
- Assume all packets are of the same duration (we will use that as the unit of time)
ALOHA - Vulnerable Period

- If a station is transmitting, what is the duration for which no other station should transmit?
- Vulnerable duration is twice the packet duration
ALOHA - Vulnerable Period
Slotted ALOHA

- $n$ stations have synchronized clocks
- All stations agree on beginning of each time slot
- Each slot the same as packet duration.
- Stations transmit in the next time-slot (after arrival of the packet)
- If two or more stations receive a packet during a specific time slot they all transmit at the next slot
- Vulnerable period is one time slot.
Assume each station transmits with probability $p$ in any slot.

A packet is successful if no other station transmits during the same slot.

What is the probability that a packet transmitted during a specific time-slot reaches the satellite?

$$ S = Np(1 - p)^{N-1} $$

How do we maximize $S$?

Only thing the stations can do is control $p$. 

$$ S = Np(1 - p)^{N-1} $$
Maximizing Throughput $S$

- Let $p = x/N$. Then $S = x(1 - x/N)^{N−1}$.
- What value of $x$ maximizes $S$? (Compute $\frac{dS}{dx}$ and set it to 0)
- $x = 1$ maximizes $S$: or best choice of $p$ is $p = 1/N$.
- Best throughput $S^* = N(1/N)(1 - 1/N)^{N−1} \approx 1/e \approx 0.368$ (for $N \to \infty$)
- For “plain” (not slotted) ALOHA, $p$ is the probability of sending a packet in two time slots
- Throughput $\approx 1/e$ in two packet durations
- Or $S^*_P \approx 1/2e \approx 0.184$ in each packet duration
- Slotted ALOHA twice as efficient as plain ALOHA
Slotted ALOHA Example

- 1 Mbps link to satellite.
- Packet size 1000 bits.
- Satellite can receive 1,000 packets per sec.
  Packet-interval/slot-size 1 ms.
- Best case scenario for slotted ALOHA
  - Occurs when all stations together send one packet per slot interval (on an average)
  - In each second 1000 packets transmitted (on an average)
  - only 368 of them successful
  - 632 packets suffer collision every second (and have to be retransmitted)
Plain ALOHA Example

- 1 Mbps link to satellite.
- Packet size 1000 bits.
- Satellite can receive 1,000 packets per sec.
  Packet-interval/slot-size 1 ms.
- Best case scenario for plain ALOHA
  - Occurs when all stations together send one packet every two packet durations (on an average)
  - In each second 500 packets are transmitted (by all stations together).
  - Only 184 reach the satellite
  - 316 suffer collision (and have to be retransmitted)
Stability of ALOHA

- What happens when $p$ increases?
- Simulate (Project idea 1)
- Through simulations study the characteristics of ALOHA and slotted ALOHA
Channel Utilization With ALOHA

Slotted ALOHA: $S = Ge^{-G}$

Pure ALOHA: $S = Ge^{-2G}$
Carrier Sense Multiple Access Protocols

- Listen first. If no one is currently using the channel, then transmit
- Collisions can still occur
  - Propagation delay (B has started transmission, but A does not know that yet)
  - A and B wait for C (which is currently transmitting) to finish; A and B start at the same time.
- Collisions inferred through the lack of ACKs
- Persistent and non-persistent CSMA
Non-persistent CSMA

1. If channel is free transmit.
2. If not stop sensing and wait (for a random time)
3. If the channel is free now, transmit
4. If not, back to 2 (stop sensing and wait)
Persistent CSMA

- 1-persistent - if busy keep sensing the line till it becomes available and start transmitting immediately
- $p$-persistent (for slotted channels)
  - if busy keep sensing the channel
  - If free, transmit at the next slot - but with probability $p$
  - Probability $q = 1 - p$ that a station will yield.
  - Process continues till the packet has been transmitted or another station takes over the channel.
Channel Utilization With CSMA / ALOHA

- Pure ALOHA
- Slotted ALOHA
- 1-persistent CSMA
- 0.5-persistent CSMA
- Nonpersistent CSMA
- 0.1-persistent CSMA
Project Idea 2

- Analyze throughput vs traffic (total attempts per packet duration) for CSMA
- obviously, will depend on the strategy (non-persistent, p-persistent)
CSMA With Collision Detection (CSMA-CD)

- In CSMA (and ALOHA) collisions are *inferred* through the lack of ACKs.
- What if stations can send and listen at the same time?
- Stations can *stop* transmitting as soon as they detect *an other* transmission.
- Collision uses up a *smaller* fraction of the time.
- CSMA-Protocol
  - Stations sense channel, and transmit only if the channel is free.
  - If collision is detected - stop sending the packet (only part of a packet may be sent).
  - Retry (*p*-persistent? non-persistent?)
  - Actually *p*-persistent with dynamic *p* (more on this later)
CSMA-CD States

- Transmission period
- Contention period
- Contention slots
- Idle period
Seizing the Channel with CSMA-CD

- After a station starts transmission of a packet, it should be able to detect within a period $\tau_d$ that no collision is going to occur for that packet.

- If $\tau$ is the maximum transmission delay between any two stations in the channel $\tau_d < 2\tau$. Why?

- After time $\tau_d$, the station is sure that it has “seized” the channel.

- $\tau_d$ is the contention period.

- CSMA-CD is employed in Ethernet.
2\(\tau\) for Collision Detection

(a) Packet starts at time 0

(b) Packet almost at B at \(\tau - \epsilon\)

(c) Collision at

(d) Noise burst gets back to A at \(2\tau\)
Restrictions On Packet Size

- With CSMA-CD there is also a restriction on *minimum* packet duration!
- What happens if packet duration is less than $2\tau$?
- It is possible that $A$ completes the entire packet before it *sensed* the collision.
- So $A$ does not even *know* that a collision occurred!
- Packet sizes have to be chosen such that the packet duration is greater than $2\tau$
Minimum Packet Size

- For 10 Mbps LAN, maximum distance of 2500 m between two stations
- In practice, upto 4 repeaters may be used
- Propagation delay $25\mu s = 10 + 15\mu s$ overall repeater delay.
- $\tau_d = 50\mu s$.
- Smallest allowed packet size is 500 bits (Ethernet prescribes 64 bytes (512 bits) as the minimum packet size).
- For 100 Mbps LAN Ethernet standard recommends max length 250 m
- $1\mu s + 1.5\mu s$ repeater delay. $\tau_d = 5\mu s$.
- Minimum packet size is again 500 bits (64 bytes)
CSMA-CD Efficiency

- Let average number of contention slots be $x$
- The channel is occupied by
  - useful packets of duration $P$ and
  - useless contentions (on an average $x$ contentions of duration $2\tau$ each).
- Efficiency
  \[ \nu = \frac{P}{P + 2\tau x} \]
- $x \approx e$ (the reason is a little complex for now. We will get to this later)
CSMA-CD Efficiency

- $F$ - frame size (in bits)
- $R$ - data rate (bits per sec)
- $P = \frac{F}{R}$
- $c$ - velocity of propagation
- $L$ - maximum separation between two stations)
- $\tau = \frac{L}{c}$

$$\nu = \frac{P}{P + 2\tau e} = \frac{1}{1 + 2RLe/cF}$$
Hidden / Exposed Station Problem

- CSMA assumes that all stations can hear all other stations vying for the same channel.
- What happens if each station has a small transmission range?
- Hidden and Exposed Station Problems

(a)

(b)
Hidden Station Problem

- $A \rightarrow B$.

- $C$ senses medium but does not hear $A$’s transmission as $C$ is out of range.

- If $C$ transmits we have collision at $B$.

- Hidden Station Problem. $A$ is hidden from $C$

![Diagram](a)

![Diagram](b)
Exposed Station Problem

- $B \rightarrow A$.
- $C$ needs to transmit to $D$.
- $C$ senses medium and assumes that it cannot transmit.
- Actually it can as $A$'s reception will not be affected by $C$’s transmission.
- Interference at the Rx is the problem: not the interference at the transmitter
- Exposed Station Problem (leads to loss of opportunity)

(a)
MACA Protocol

- Multiple Access with Collision Avoidance (MACA)
- Rx should send a short frame to indicate that it is ready to receive from a source.
- Other stations hearing the Rx will yield.
- RTS (Request to Send) and CTS (Clear to Send) packets.
**A → B With RTS and CTS**

1. **Range of A's transmitter**
   - **C**
   - **A** → **RTS** → **B**
   - **D**
   - **E**

2. **Range of B's transmitter**
   - **C**
   - **A** → **CTS** → **B**
   - **D**
   - **E**

---

**Outline**
- Networks and Components
- Layered Model
- Data Communication Channels
- Delays and Switching
- Reliable Data Transfer

**Error Control**
- ARQ Protocols
- Alternating Bit Protocol
- Reliable and Efficient Data Transfer
- ALOHA
- CSMA
- CSMA-CD
- CSMA-CA
- Reservation Based Protocols
- $A \to B$ RTS. $C$ hears RTS.
- $B \to A$ CTS response.
- $C$ waits a reasonable amount of time, but does not hear CTS.
- $C$ can now initiate a transmission on its own (by sending an RTS).
- $D$ hears CTS but not RTS.
- $D$ should desist until the data packet following CTS is finished.
MACA Rules

- RTS and CTS packets indicate sender, intended receiver, and size of data packet to be sent.
- If you hear only RTS ($A, B, n$)
  wait for some time to make sure receiver $B$ is out of range before initiating an RTS (to some other station)
- If you hear CTS (either CTS only or both RTS and CTS) the receiver is within range.
  do not transmit until the data transmission ($n$ bytes from $A \rightarrow B$) is completed.
MACAW - MACA for Wireless

- Some enhancements to MACA
- MAC layer acknowledgements. To receive ACK there should be no collision at the transmitter of the original packet. So exposed station problem is deliberately not addressed.
- CSMA for transmitting RTS
- Back-off algorithm for each source-destination pair instead of for each station.
- Additional information exchange about congestion.
Collision-Free Protocols

- Bit-map protocol
- Binary Countdown
- Especially suitable for small number of stations that are close to each other (very little delay)
Bit Map Protocol

8 Contention slots
0 1 2 3 4 5 6 7
1 1 1 1 1 1 5 11 3 7
2 3 4 5 6 7
0 1 2 3 4 5 6 7
18 Contention slots Frames 8 Contention slots
2
d

177 / 179
Binary Countdown

Bit time
0 1 2 3

0 0 1 0  
0 1 0 0
0 1 0 0
1 0 0 1
1 0 1 0

Result
1 0 1 0

Stations 0010 and 0100 see this
Station 1001 sees this 1
Width of each reservation slot should be of the order of the maximum delay between stations (so there is no ambiguity as to which station(s) is making a reservation)

Number of reservation slots proportional to number of stations (in bit-map protocol)

Number of reservation slots proportional to log\textsubscript{2} of number of stations for each packet (in binary countdown)

However, if number of packets is proportional to number of stations, even for binary countdown the total overhead (for reservation slots) will depend on the number of stations!