Networking Fundamentals

Networking for Big Data and Laboratory

M.Sc. in Data Science

Paolo Di Lorenzo

Academic Year 2024/2025

SAPIENZA
UNIVERSITÀ DI ROMA

Table of Contents

Technological Evolution of Networks

1 Introduction

- **18th Century:** Era of great mechanical systems and the Industrial Revolution.
- **19th Century:** Age of the steam engine.
- **20th Century:** Dominated by information technology:
  - Installation of worldwide telephone networks.
  - Invention of radio and television.
  - Birth and growth of the computer industry.
  - Launching of communication satellites.
  - Emergence of the Internet.
- **21st Century:** Rapid convergence of technologies and the blurring of boundaries between data collection, transportation, storage, and processing.

Technological Evolution of Networks

1 Introduction

- Early computers were centralized, large, and room-sized.
- Modern computers are interconnected and distributed.
- Emergence of **computer networks** and **distributed systems**.

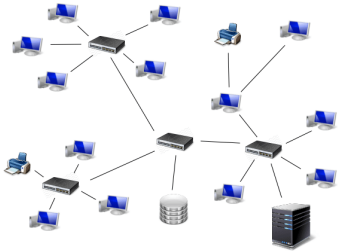
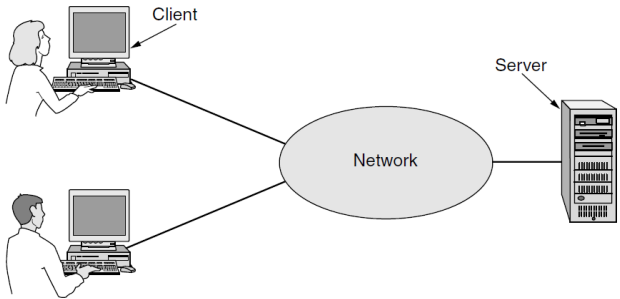


Figure: A computer network

Definitions

- **Computer Network:** A collection of autonomous computers interconnected by a technology.
- **Distributed System:** A software system built on top of a network, presenting a single coherent system to users.

Business Applications: Client-Server model



- A company's information system consists of one or more databases stored on powerful computers called *servers*.
- The employees have simpler machines, called *clients*, with which they access remote data.
- The client and server machines are connected by a *network*.

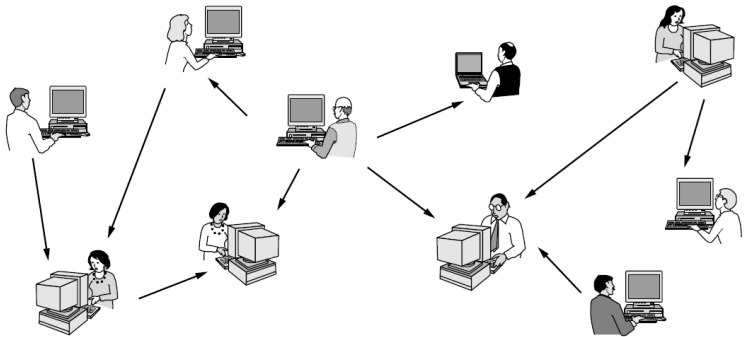
Business Applications: Enhanced Communications

1 Introduction

- *Email* is a common communication tool in organizations.
- *IP Telephony/VoIP* for cost-effective telephone communication.
- *Video conferencing* reduces travel costs and time.
- *Desktop sharing* for real-time collaboration on documents among distant employees.
- E-commerce (electronic commerce) has grown rapidly in recent years.
- Emerging technologies like *telemedicine* for remote coordination.

Figure: Networks for Communication between Employees

Home Applications: Peer-to-peer model



- Individuals who form a loose group can communicate with others in the group, without fixed division into clients and servers.
- Peer-to-peer communication is often used to share music, photos, software, and videos.

Home Applications: Person-to-Person Communications

1 Introduction

- **Email:** Widely used, growing rapidly, includes audio, video, text, and pictures.
- **Social Messaging:** Services like Whatsapp or Twitter for sending short messages to friends.
- **Audio and Video:** Internet radio and video services like YouTube.
- **Telelearning:** Attending classes remotely.
- **Social Networks:** Platforms like Facebook for updating profiles and sharing updates with friends.

Home Applications: Entertainment

- **TV shows** now reach many homes via IPTV (IP TeleVision) systems
- **Game playing:** Thousands of users can experience a shared reality with 3D graphics.

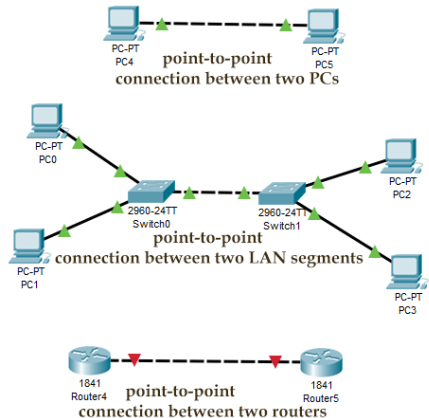
Home Applications: Smart Environments

- Ubiquitous sensors collect data for safety or smart monitoring
- As the cost of sensing and communication drops, more measurement and reporting will be done with networks.

Point-to-Point Links in Networks

1 Introduction

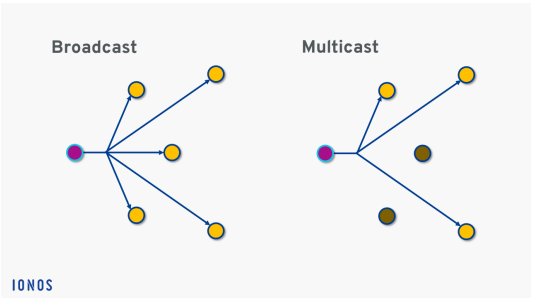
- Point-to-point links connect individual pairs of machines.
- Messages, called **packets**, may need to visit intermediate machines.
- Multiple routes of different lengths may exist between source and destination.
- Finding optimal routes is crucial in point-to-point networks.
- Point-to-point transmission with one sender and one receiver is called **unicasting**.



Broadcast Networks

1 Introduction

- Communication channel is shared by all machines on the network.
- Packets are received by all machines, but processed only by the intended recipient.
- Wireless networks are a common example of broadcast links.
- **Broadcasting:** Sending a packet to all machines using a special address code.
- **Multicasting:** Sending a packet to a subset of machines on the network.



Classification of Networks

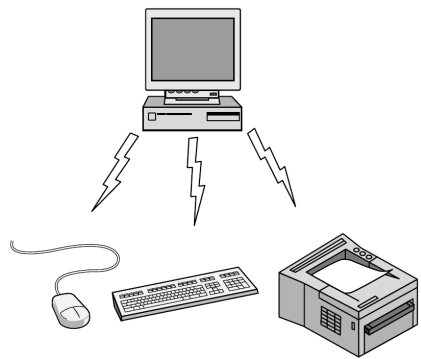
1 Introduction

Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	Local area network
100 m	Building	
1 km	Campus	
10 km	City	Metropolitan area network
100 km	Country	Wide area network
1000 km	Continent	
10,000 km	Planet	The Internet

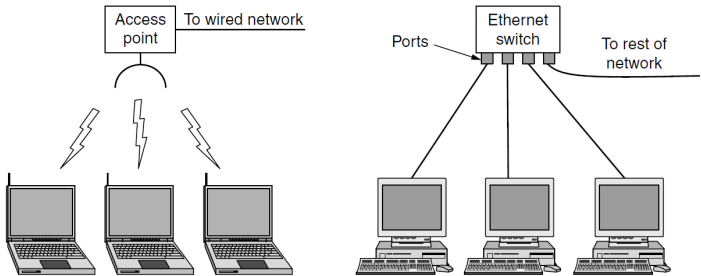
- A fundamental criterion for classifying networks is by scale.
- Distance is an important classification metric because different technologies are used at different scales.

Classification: Personal Area Networks

- PANs allows communicat over a short range.
- Common example: Wireless network connecting a computer with its peripherals.
- Bluetooth is a widely used PAN technology:
  - Eliminates the need for cables.
  - Simple setup: devices connect automatically.
  - Uses master-slave paradigm: PC (master) communicates with peripherals (slaves).
- Other PAN technologies:
  - Embedded medical devices communicating with remote controls.
  - RFID used in smartcards and library books.

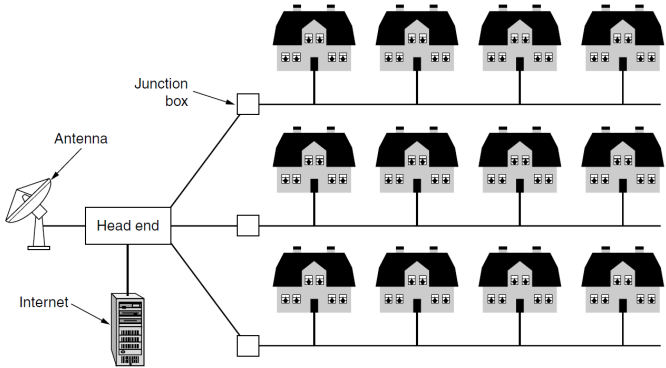


Classification: Local Area Networks



- A LAN is a privately owned network that operates within and nearby a single building like a home, office or factory.
- LANs are widely used to connect personal computers and consumer electronics via wireless (i.e., WI-FI) and/or wired (i.e., Ethernet) connections

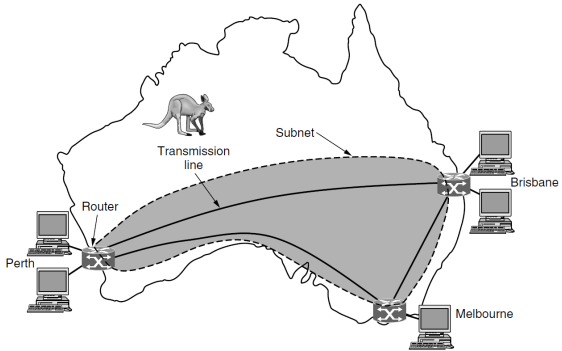
Classification: Metropolitan Area Networks



- A MAN (Metropolitan Area Network) covers a city.
- The best-known examples of MANs are the cable television networks available in many cities, mainly used in areas with poor over-the-air television reception.

Classification: Wide Area Networks

1 Introduction



- A WAN (Wide Area Network) spans a large geographical area, often a country or continent.
- The WAN is composed by *hosts*, a *subnet* that carries messages from host to host, and *routers* to send the data toward destination

Classification: Internet

- Many networks exist in the world, often with different hardware and software.
- People connected to one network often want to communicate with people attached to a different (and frequently incompatible) one.
- A collection of interconnected networks is called an internetwork or **internet**.

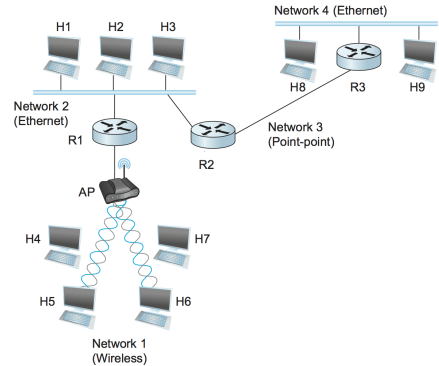


Figure: Internetworking Example

Layers and Protocols

- Networks are organized as a **stack of layers** to reduce design complexity.
- Each layer provides services to the higher layers, hiding implementation details.
- Communication rules between corresponding layers on different hosts are called **protocols**.
- Below layer 1 is the physical medium through which actual communication occurs.
- An **interface** links pairs of adjacent layers.
- A set of layers and protocols is called a **network architecture**. A list of the protocols used by a certain system is called a **protocol stack**.

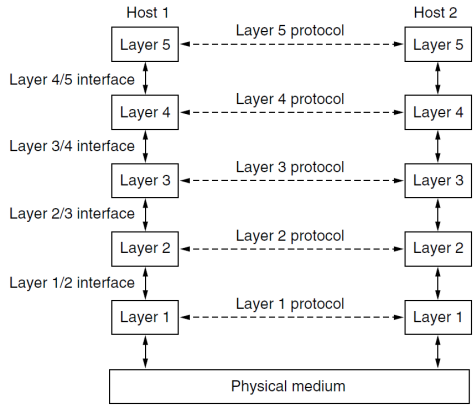


Figure: Example of Layered Network Model

Example of Information Flow

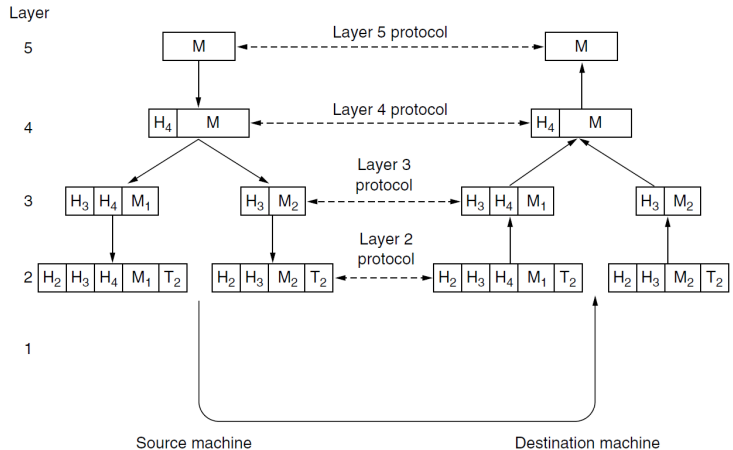


Figure: Example information flow supporting virtual communication in layer 5.

Example of Information Flow

- **Application Process (Layer 5):** Produces a message, M, for transmission.
- **Layer 4:**
  - Adds a header with control information (e.g., addresses).
  - Passes the message to Layer 3.
- **Layer 3:**
  - Breaks down large messages into smaller packets (e.g., M_1 and M_2).
  - Adds a Layer 3 header to each packet.
  - Decides the outgoing lines and passes packets to Layer 2.
- **Layer 2:**
  - Adds both a header and a trailer to each packet.
  - Passes the resulting units to Layer 1 for physical transmission.

Example of Information Flow

- **Layer 1:** Responsible for the physical transmission of data.
- **At the Receiving End:**
  - Message moves up from Layer 1 to Layer 5.
  - Headers are stripped off as the message progresses through layers.
- **Key Concepts:**
  - Headers for layers below n are not passed up to Layer n.
  - The virtual and actual communication differ; understanding protocols and interfaces is crucial.
  - The peer process abstraction simplifies network design by *breaking into several smaller, manageable design problems*, namely, the design of the individual layers.

Design Issues for Layers

- **Reliability:**
  - Ensuring network operation despite unreliable components.
  - **Error detection and correction** using redundant information.
- **Routing:** Finding (and adapting) a working path through a network with multiple paths.
- **Addressing/Naming:** Identifying senders and receivers involved in a message.
- **Internetworking:** Handling different limitations of various network technologies.
- **Resource Allocation:**
  - Dividing network resources to prevent interference among hosts.
  - **Flow control:** preventing fast senders from overwhelming slow receivers using feedback.
  - **Congestion control:** reducing demand during network overloading.

The OSI Reference Model

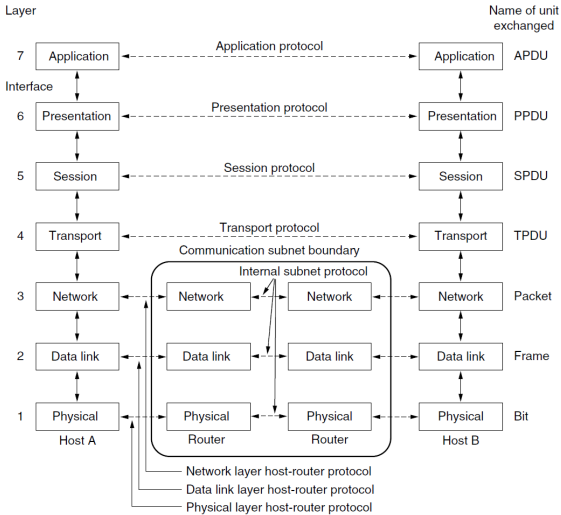
1 Introduction

The OSI (Open Systems Interconnection) model has seven layers. The principles that were applied to arrive at the seven layers can be briefly summarized as follows:

1. A layer should be created where a different abstraction is needed.
2. Each layer should perform a well-defined function.
3. The function of each layer should be chosen with an eye toward defining internationally standardized protocols.
4. The layer boundaries should be chosen to minimize the information flow across the interfaces.
5. The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity and small enough that the architecture does not become unwieldy.

The OSI Reference Model

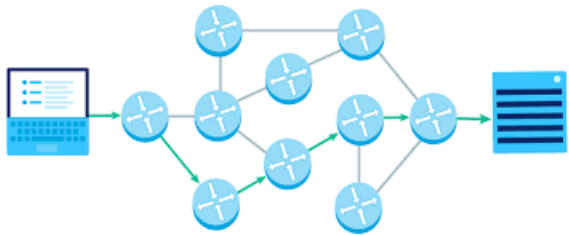
1 Introduction



The OSI Reference Model

- The **physical layer** is concerned with transmitting raw bits over a communication channel. The design issues have to do with making sure that when one side sends a 1 bit it is received by the other side as a 1 bit, not as a 0 bit.
- The main task of the **data link layer** is to transform a raw transmission facility into a line that appears free of undetected transmission errors.
  - The sender breaks up the input data into *data frames* (typically a few hundred or a few thousand bytes) and transmit the frames sequentially
  - If the service is reliable, the receiver confirms correct receipt of each frame by sending back an *acknowledgement frame*.
  - A special sublayer of the data link layer, the **medium access control** sublayer, deals with control access to a shared channel (e.g., broadcast).

The OSI Reference Model

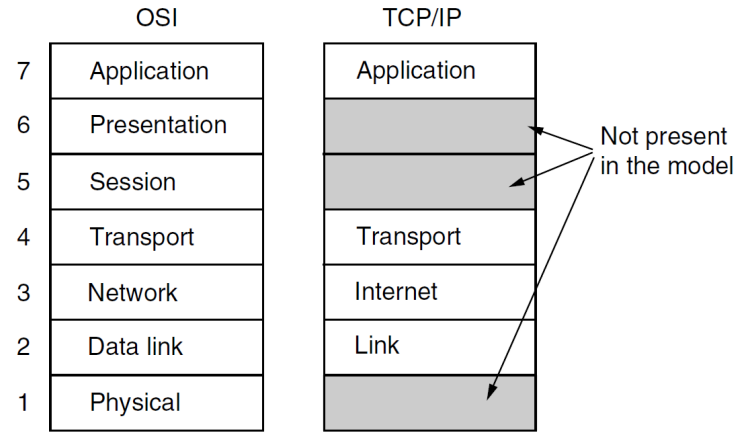


- The **network layer** controls the operation of the subnet. A key design issue is determining how packets are *routed* from source to destination.
  - Routes can be based on *static tables* that are “wired into” the network.
  - Routes they can be highly *dynamic*, being determined anew for each packet to reflect the current network load.
  - The provided *quality of service* (delay, transit time, etc.) is also a network layer issue.

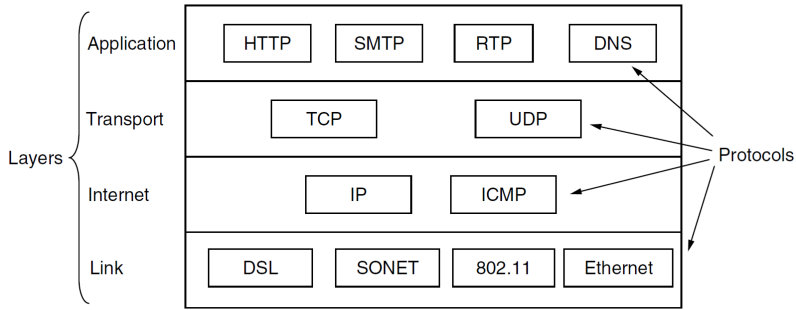
The OSI Reference Model

- The **transport layer** accepts data from above it, splits it up into smaller units if need be, pass these to the network layer, and ensures that the *pieces all arrive correctly at the other end*.
  - Example: An error-free point-to-point channel that delivers messages or bytes *in the order in which they were sent*.
  - The transport layer is a true *end-to-end layer* (from the source to the destination).
  - The transport layer helps *reducing network congestion* adapting the load of transmitted data to the state of the network.
- The **session layer** allows users on different machines to establish sessions between them, offering services such as dialog control, token management, and synchronization.
- The **presentation layer** is concerned with the syntax and semantics of the information
- The **application layer** contains a variety of protocols that are commonly needed by users. One widely used application protocol is HTTP (HyperText Transfer Protocol), which is the basis for the World Wide Web.

The TCP/IP Reference Model



The TCP/IP Reference Model



- The strength of the OSI reference model is the model itself, which has proven to be exceptionally useful for discussing computer networks.
- In contrast, the strength of the TCP/IP reference model is the protocols, which have been widely used for many years.

Our Reference Model

- The **physical layer** specifies how to transmit bits across different kinds of media as electrical (or other analog) signals.
- The **link layer** is concerned with how to send finite-length messages between directly connected computers with specified levels of reliability. Protocols: *Ethernet*, *802.11*, etc.
- The **network layer** deals with how to combine multiple links into networks so that we can send packets between distant computers. Protocol: *IP*.
- The **transport layer** provides delivery abstractions, such as a reliable byte stream, that match the needs of different applications. Protocols: *TCP*, *UDP*.
- The **application layer** includes programs that use the network.

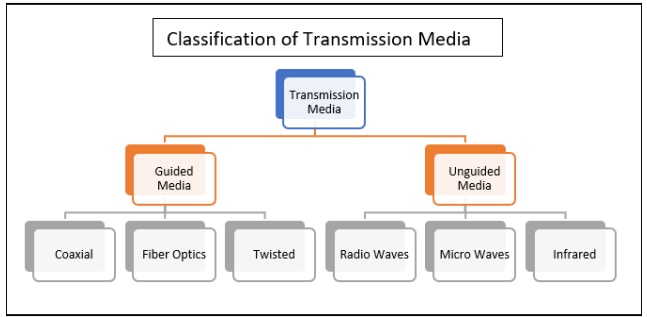
5	Application
4	Transport
3	Network
2	Link
1	Physical

Figure: Our reference model with 5 layers.

Table of Contents

2 The Physical Layer

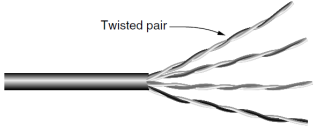
Transmission Media

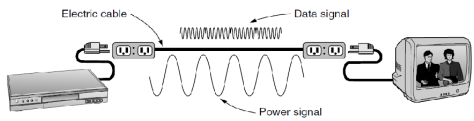


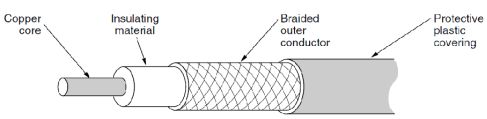
- The purpose of the physical layer is to transport bits from one machine to another.
- Various *physical media* can be used for the actual transmission.
- Media are roughly grouped into *guided media*, such as copper wire and fiber optic, and *unguided media*, such as terrestrial wireless, satellite, and lasers through the air.

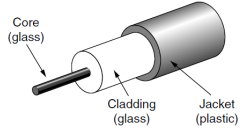
Guided Transmission Media

- Examples of guided physical media are:
  - Magnetic media
  - Twisted pairs
  - Coaxial cable
  - Power lines
  - Fiber Optics



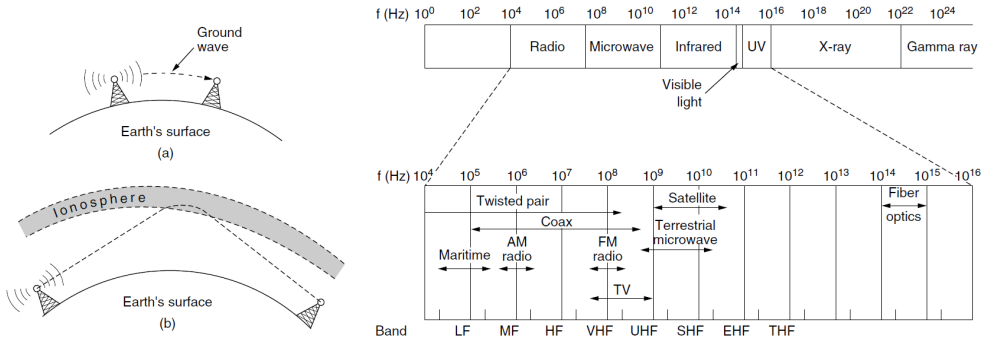






Wireless Communications

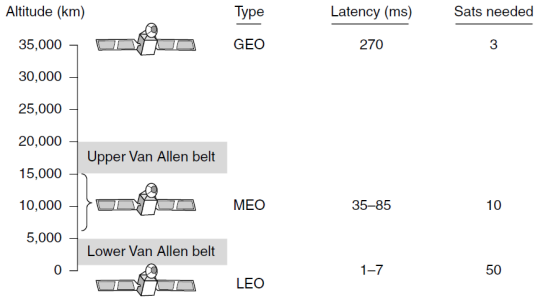
2 The Physical Layer

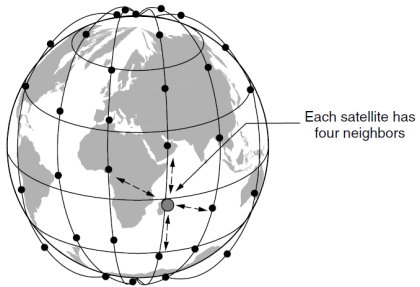


- Communication happens through electromagnetic waves that occupy a certain portion of the available spectrum of frequencies

Satellite Communications

2 The Physical Layer

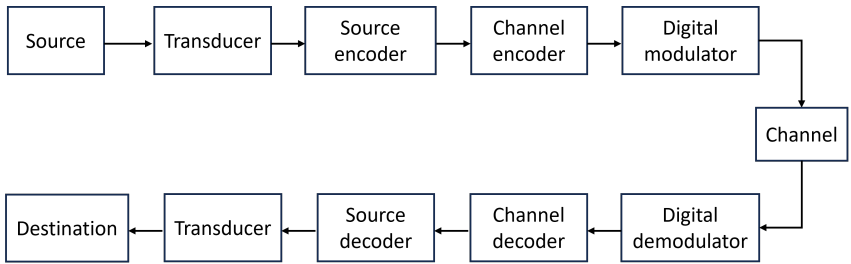




- A communication satellite can be thought of as a big microwave repeater in the sky.
- It contains several transponders, each of which listens to some portion of the spectrum, amplifies the incoming signal, and then rebroadcasts it at another frequency to avoid interference with the incoming signal.

Elements of a digital communication system

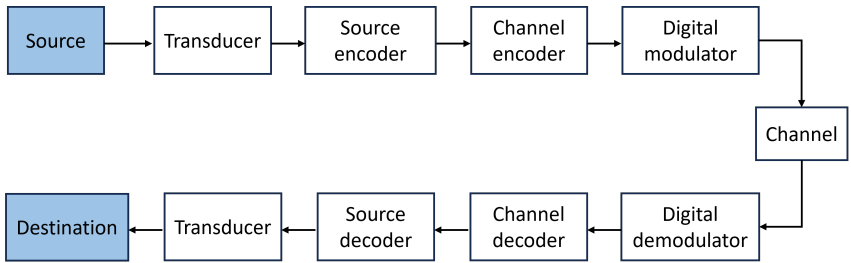
- The goal of a digital communication system is to transmit information from one source to one (or more) destination(s)



Block scheme of a digital communication system

Elements of a digital communications system

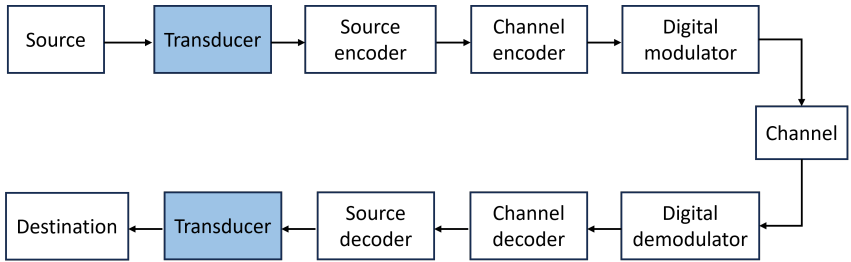
- The **source** can emit an analog signal (e.g., voice, video), or a digital signal with finite number of characters (e.g., a text), with the aim of reaching a **destination**



Block scheme of a digital communication system

Elements of a digital communications system

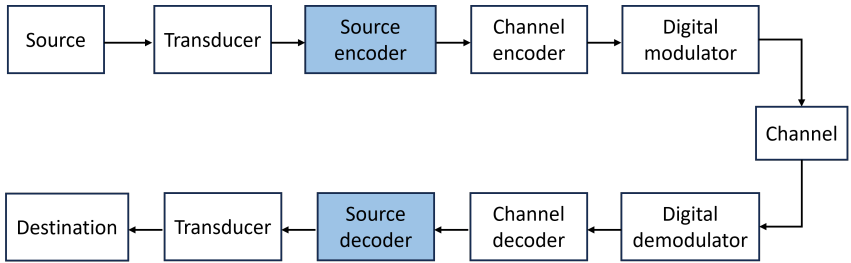
- A **transducer** changes the physical support over which information is carried. Example: A microphone (speaker) performs an audio/electrical (electrical/audio) conversion



Block scheme of a digital communication system

Elements of a digital communications system

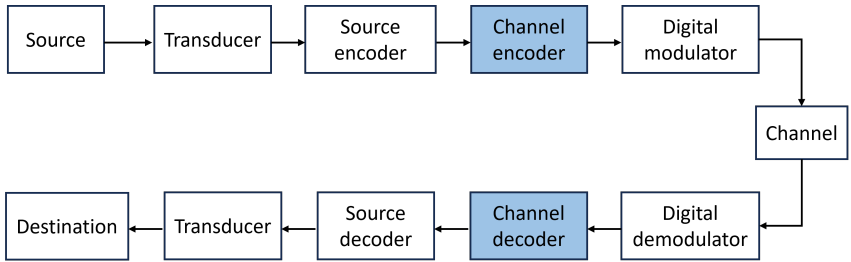
- The **source encoder** aims at representing the source message with the minimum possible number of bits, reducing as much as possible the redundancy in the data. It performs a data compression, which might be lossless (e.g., ZIP) or lossy (e.g., JPEG)



Block scheme of a digital communication system

Elements of a digital communications system

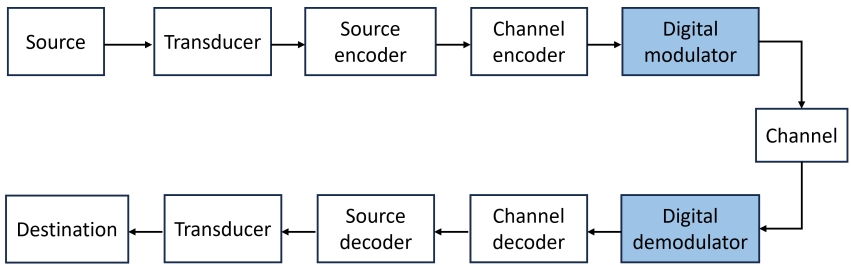
- The **channel encoder** makes the communication robust to impairments introduced by the channel (e.g., noise, attenuation, interference, etc.). It adds redundancy (i.e., additive bits) to enable error detection/correction capabilities



Block scheme of a digital communication system

Elements of a digital communications system

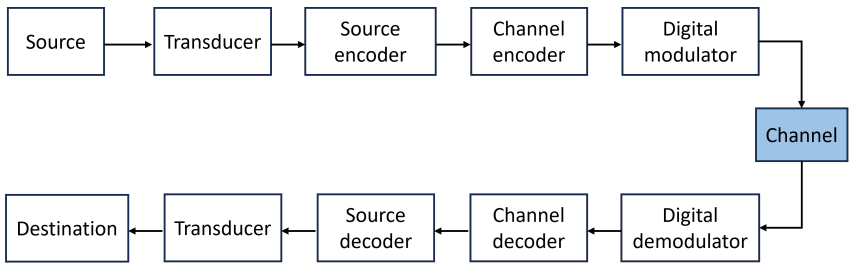
- The **digital modulator** maps the binary input data (i.e., a flow of bits) into analog signals (e.g., EM waves, light pulses) that can be transmitted over the physical channel



Block scheme of a digital communication system

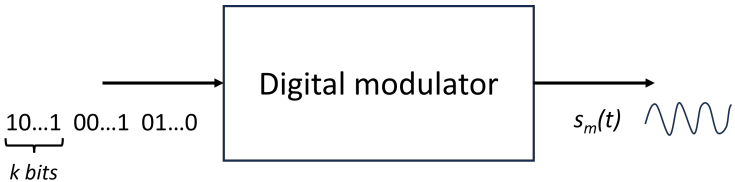
Elements of a digital communications system

- The **channel** is the physical medium (e.g., copper lines, optical fibers, air, etc.) used to transmit the signal from the source to the destination. The channel introduces several disturbances such as thermal and/or atmospheric noise, interference, etc.



Block scheme of a digital communication system

Digital modulation



- Digital data are typically represented as a flow of bits
- To transmit over a given channel, the digital modulator converts the flow of bits into an analog signal that carries the information
- Digital modulation schemes typically divide the flow of bits into blocks of k bits, and associate a waveform $s_m(t)$, with $1 \leq m \leq M = 2^k$

Parameters of digital modulation

- **Symbol rate:** A waveform $s_m(t)$ is transmitted every T_s seconds. Thus, the system transmits with an overall rate equal to

$$R_s = \frac{1}{T_s} \quad \text{symbols/sec}$$

- **Bit rate:** Since each waveform (i.e., symbol) carries k bits, the bit rate is equal to

$$R = kR_s = \frac{\log_2 M}{T_s} \quad \text{bits/sec}$$

- **Average energy:** If E_m is the energy of $s_m(t)$, the average energy is

$$E_{avg} = \sum_{m=1}^{M} p_m E_m \quad \text{Joules}$$

where p_m is the probability to transmit the m-th signal

Synthesis of digital signals

- Given a set of N orthonormal functions $\{\phi_n(t)\}_{n=1}^{N}$, each signal $s_m(t)$ can be written as

$$s_m(t) = \sum_{n=1}^{N} s_{mn}\phi_n(t) \qquad m = 1, \ldots, M$$

- Equivalently, every signal $s_m(t)$ can be represented by the vector $\mathbf{s}_m = [s_{m1}, \ldots, s_{mN}]^T$
- The set of of vectors $\{\mathbf{s}_m\}_{m=1}^{M}$ is the discrete representation of the signal space, also known as **constellation diagram**
- In practical systems, the waveforms $s_m(t)$, $m = 1, \ldots, M$, are band-pass signals that differs in amplitude, frequency, phase, or combinations of these

Pulse amplitude modulation

- Pulse amplitude modulation (PAM) is characterized by waveforms $s_m(t)$ given by

$$s_m(t) = A_m\, g(t) \cos(2\pi f_0 t)$$

where f_0 is the carrier frequency, $g(t)$ is a finite-energy signal called *pulse shaper*, and

$$A_m = 2m - 1 - M \qquad m = 1, 2, \ldots, M$$

are discrete amplitude values, i.e., $\mp 1, \mp 3, \ldots, \mp M - 1$

- The energy of $s_m(t)$ is given by $E_m = A_m^2 E_g/2$, and the average energy of all signals is

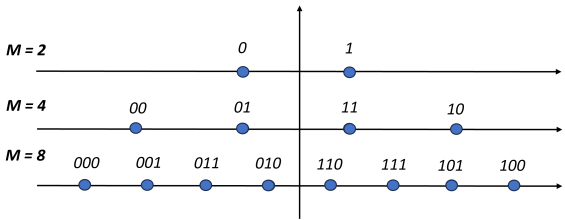
$$E_{avg} = \frac{E_g}{2M} \sum_{m=1}^{M} A_m^2 = \frac{(M^2 - 1)E_g}{6}$$

Pulse amplitude modulation

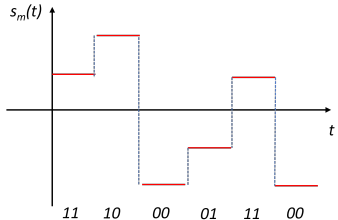
- PAM signals have dimension $N = 1$, i.e.,

$$s_m(t) = A_m\, g(t) \cos(2\pi f_0 t) = s_m\, \phi(t)$$

where $\phi(t) = \sqrt{\frac{2}{E_g}} \cos(2\pi f_0 t)$ and $s_m = A_m \sqrt{\frac{E_g}{2}}$ with $A_m = \mp 1, \ldots, \mp M - 1$



PAM constellation diagram



Example of baseband 4-PAM signal

Phase shift keying

- Phase shift keying (PSK) is characterized by waveforms $s_m(t)$ given by

$$s_m(t) = g(t)\cos(2\pi f_0 t + \theta_m)$$

where f_0 is the carrier frequency, $g(t)$ is a finite-energy signal called *pulse shaper*, and

$$\theta_m = \frac{2\pi}{M}(m-1) \qquad m = 1, 2, \ldots, M$$

are discrete phase values, i.e., $0, \frac{2\pi}{M}, \frac{2\pi}{M}2, \ldots, \frac{2\pi}{M}(M-1)$

- The energy of $s_m(t)$ is given by $E_m = E_g/2$ for all m, i.e., all the signals have the same energy, which clearly coincides with the average E_{avg}

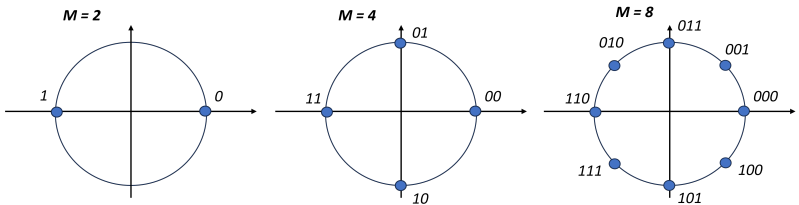
Phase shift keying

- PSK signals have dimension $N = 2$, i.e.,

$$s_m(t) = g(t)\cos\left(2\pi f_0 t + \frac{2\pi}{M}(m-1)\right) = s_{m1}\,\phi_1(t) + s_{m2}\,\phi_2(t)$$

where $\phi_1(t) = \sqrt{\frac{2}{E_g}}\cos(2\pi f_0 t)$ $\phi_2(t) = \sqrt{\frac{2}{E_g}}\sin(2\pi f_0 t)$

$$\mathbf{s}_m = [s_{m1}, s_{m2}]^T = \left[\sqrt{\frac{E_g}{2}}\cos\left(\tfrac{2\pi}{M}(m-1)\right), \sqrt{\frac{E_g}{2}}\sin\left(\tfrac{2\pi}{M}(m-1)\right)\right]^T$$



PSK constellation diagram

Quadrature amplitude modulation

- Quadrature amplitude modulation (QAM) is characterized by waveforms $s_m(t)$ given by

$$s_m(t) = A_{mc}\, g(t) \cos(2\pi f_0 t) + A_{ms}\, g(t) \sin(2\pi f_0 t)$$

  where f_0 is the carrier frequency, $g(t)$ is a finite-energy signal called *pulse shaper*, and

$$A_{mc}, A_{ms} = 2m - 1 - \sqrt{M} \qquad m = 1, 2, \ldots, \sqrt{M}$$

  are discrete amplitude values, i.e., $\mp 1, \mp 3, \ldots, \mp\sqrt{M} - 1$

- The energy of $s_m(t)$ is given by $E_m = (A_{mc}^2 + A_{ms}^2)E_g/2$, and the average energy of all signals is

$$E_{avg} = \frac{E_g}{2M} \sum_{m=1}^{\sqrt{M}} \sum_{n=1}^{\sqrt{M}} (A_m^2 + A_n^2) = \frac{(M-1)E_g}{3}$$

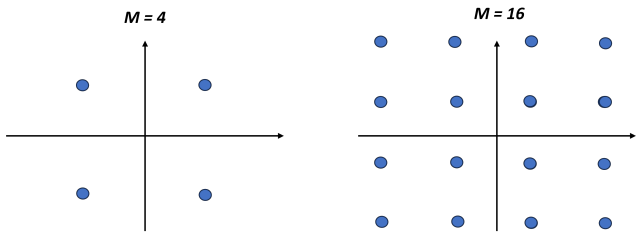
Quadrature amplitude modulation

- QAM signals have dimension $N = 2$, i.e.,

$$s_m(t) = A_{mc}\, g(t) \cos(2\pi f_0 t) + A_{ms}\, g(t) \sin(2\pi f_0 t) = s_{m1}\, \phi_1(t) + s_{m2}\, \phi_2(t)$$

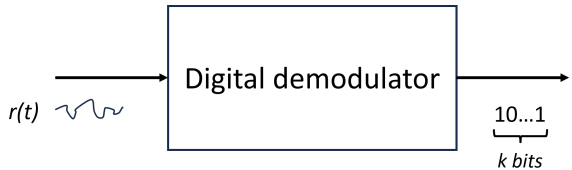
where $\phi_1(t) = \sqrt{\frac{2}{E_g}} \cos(2\pi f_0 t)$ $\phi_2(t) = \sqrt{\frac{2}{E_g}} \sin(2\pi f_0 t)$

$$\mathbf{s}_m = [s_{m1}, s_{m2}]^T = \left[\sqrt{\frac{E_g}{2}} A_{mc}, \sqrt{\frac{E_g}{2}} A_{ms}\right]^T$$



QAM constellation diagram

Digital demodulation



- The digital demodulator observes the (noisy) received signal $r(t)$ and makes a decision about which symbol m (and the associated k bits) was transmitted
- Typically, the optimal decision rule minimizes the error probability, i.e.,

$$P_e = P[\hat{m} \neq m]$$

where $\hat{m}$ is the estimated symbol, and m is the true transmitted symbol

Channel model

- **Additive white Gaussian noise (AWGN):** The channel is mathematically described as

$$r(t) = s_m(t) + n(t)$$

where $s_m(t)$ is the transmitted signal, and $n(t)$ is a stationary, Gaussian noise process with zero mean and variance $N_0/2$

- **Equivalent vector representation:** Using the discrete signal representation with the N orthonormal functions $\{\phi_j(t)\}_{j=1}^N$, we have the following equivalent channel model

$$\mathbf{r} = \mathbf{s}_m + \mathbf{n} \qquad m = 1, \ldots, M$$

$$s_m(t) = \sum_{j=1}^{N} s_{mj}\phi_j(t) \;\Rightarrow\; \mathbf{s}_m = [s_{m1}, \ldots, s_{mN}]^T \text{ with } s_{mj} = (\mathbf{s}_m, \boldsymbol{\phi}_j)$$

$$n(t) = \sum_{j=1}^{N} n_j\phi_j(t) \;\Rightarrow\; \mathbf{n} = [n_1, \ldots, n_N]^T \text{ with } n_j = (\mathbf{n}, \boldsymbol{\phi}_j)$$

Maximum a posteriori probability (MAP) decision

- **Decision rule:**

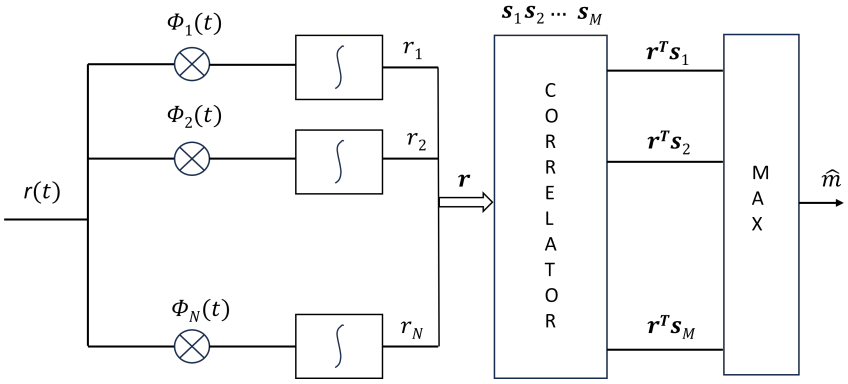
$$\hat{m} = \arg \min_{1 \le m \le M} P[\hat{m} \neq m] = \arg \max_{1 \le m \le M} P_m \, P[\mathbf{r} \mid \mathbf{s}_m]$$

where P_m is the symbol transmission probability, and $P[\mathbf{r} \mid \mathbf{s}_m]$ is the likelihood of message $\mathbf{s}_m$

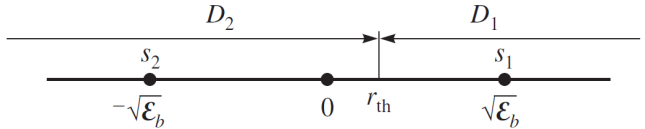
- Since $\mathbf{r} = \mathbf{s}_m + \mathbf{n}$, and $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \frac{N_0}{2}\mathbf{I})$, we obtain

$$\begin{aligned}
\hat{m} &= \arg \max_{1 \le m \le M} P_m P[\mathbf{r} \mid \mathbf{s}_m] = \arg \max_{1 \le m \le M} P_m \left(\frac{1}{\sqrt{\pi N_0}} \right)^N e^{-\frac{\|\mathbf{r}-\mathbf{s}_m\|^2}{N_0}} \\
&= \arg \max_{1 \le m \le M} \frac{N_0}{2} \log P_m - \frac{1}{2}\|\mathbf{r} - \mathbf{s}_m\|^2 = \arg \max_{1 \le m \le M} \underbrace{\frac{N_0}{2} \log P_m - \frac{1}{2} E_m}_{\eta_m} + (\mathbf{r}, \mathbf{s}_m) \\
&= \arg \max_{1 \le m \le M} \eta_m + (\mathbf{r}, \mathbf{s}_m) = \arg \max_{1 \le m \le M} \eta_m + \int_{-T_s/2}^{T_s/2} r(t) s_m(t) dt
\end{aligned}$$

MAP receiver



Block scheme of a MAP receiver



Antipodal signals:

$$s_1 = \sqrt{E_b}, \qquad s_2 = -\sqrt{E_b},$$

where E_b is the energy in each signal, i.e., per each transmitted bit.

- **Decision region D_1:**

$$D_1 = \left\{ r : r\sqrt{E_b} + \frac{N_0}{2} \ln p - \frac{1}{2}E_b > -r\sqrt{E_b} + \frac{N_0}{2} \ln(1-p) - \frac{1}{2}E_b \right\}$$
$$= \left\{ r : r > \frac{N_0}{4\sqrt{E_b}} \ln \frac{1-p}{p} \right\} = \{ r : r > r_{th} \}$$

- **Error probability:**

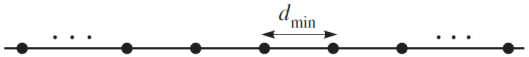
$$\begin{aligned} P_e &= p \int_{r_{\text{th}}}^{\infty} p(r|s=\sqrt{E_b})\,dr + (1-p)\int_{-\infty}^{r_{\text{th}}} p(r|s=-\sqrt{E_b})\,dr \\ &= p\,\mathrm{P}\left(\mathcal{N}\left(\sqrt{E_b},\frac{N_0}{2}\right) < r_{\text{th}}\right) + (1-p)\,\mathrm{P}\left(\mathcal{N}\left(-\sqrt{E_b},\frac{N_0}{2}\right) > r_{\text{th}}\right) \\ &= p\,Q\left(\frac{\sqrt{E_b}-r_{\text{th}}}{\sqrt{N_0/2}}\right) + (1-p)\,Q\left(\frac{r_{\text{th}}+\sqrt{E_b}}{\sqrt{N_0/2}}\right) \end{aligned}$$

- Special case: $p = \frac{1}{2}$ and $r_{\text{th}} = 0$:

$$P_e = Q\left(\sqrt{\frac{2E_b}{N_0}}\right).$$

- The error probability P_e is also the bit error probability P_b.
- The term $\gamma_b = \dfrac{E_b}{N_0}$ is the signal-to-noise ratio per bit.

Error Probability for PAM



- Types of points in the PAM constellation:
  - $M-2$ inner points, and 2 outer points.
  - Error in detection if $|n| > \dfrac{d_{min}}{2}$
- Error probabilities:

$$P_{ei} = 2Q\left(\frac{d_{\min}}{\sqrt{2N_0}}\right), \quad \text{for inner points.}$$

$$P_{eo} = \frac{1}{2}P_{ei} = Q\left(\frac{d_{\min}}{\sqrt{2N_0}}\right), \quad \text{for outer points.}$$

- Symbol error probability:

$$P_e = \frac{1}{M}\left[2(M-2)Q\left(\frac{d_{\min}}{\sqrt{2N_0}}\right) + 2Q\left(\frac{d_{\min}}{\sqrt{2N_0}}\right)\right] = \frac{2(M-1)}{M}Q\left(\frac{d_{\min}}{\sqrt{2N_0}}\right)$$

Error Probability for PAM

- The minimum distance for PAM constellation is:

$$d_{\text{min}} = \sqrt{2E_g} = \sqrt{\frac{12 \log_2 M}{M^2 - 1} E_{b\text{avg}}}.$$

- Final expression for P_e:

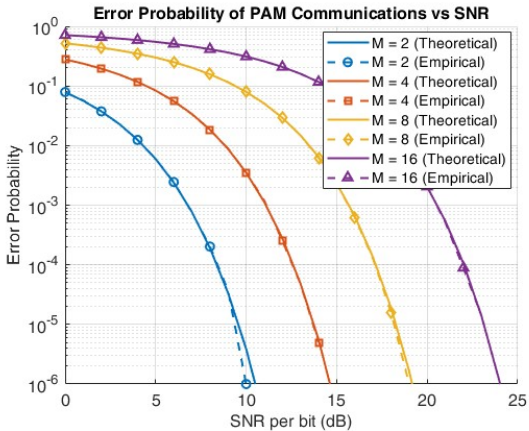
$$P_e = 2\left(1 - \frac{1}{M}\right) Q\left(\sqrt{\frac{6 \log_2 M}{M^2 - 1} \frac{E_{b\text{avg}}}{N_0}}\right).$$

- Approximation for large M:

$$P_e \approx 2Q\left(\sqrt{\frac{6 \log_2 M}{M^2 - 1} \frac{E_{b\text{avg}}}{N_0}}\right).$$

Performance of a digital communication system

- The error probability depends on the signal to noise ratio (i.e., the ratio between signal and noise powers) and the size of the constellation



Fundamental Limit on Communications

Channel coding Theorem: For any given degree of reliability, it is possible to communicate data at a rate R (bits per second) less than the **channel capacity** C with an arbitrarily small probability of error by using appropriate encoding and decoding schemes.

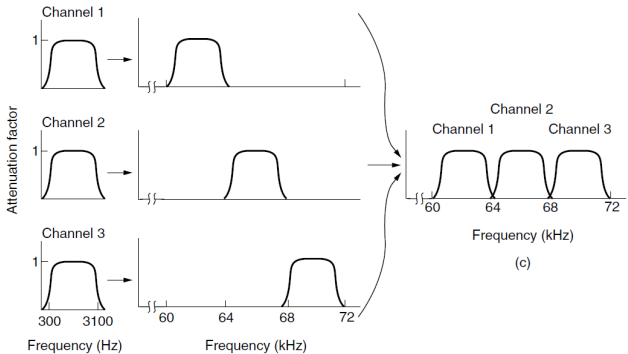
Shannon Capacity Formula: $C = B \log_2 (1 + SNR)$

- C is the channel capacity in bits per second (bps).
- B is the bandwidth of the channel in hertz (Hz).
- SNR is the signal-to-noise ratio, a dimensionless quantity.

Key Points:

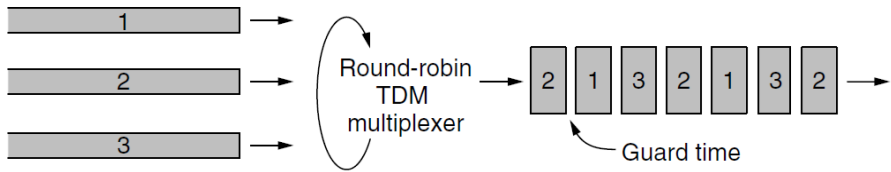
- The channel capacity C is the maximum rate at which information can be reliably transmitted over a communication channel.
- For rates $R < C$, there exist coding schemes that allow error-free transmission. If $R > C$, **reliable communication is not possible**, regardless of the coding scheme used.

Frequency Division Multiplexing



- **Multiplexing schemes** have been developed to share the same medium among many signals.
- **FDM (Frequency Division Multiplexing)** divides the spectrum into frequency bands, with each user having exclusive possession of some band in which to send their signal.

Time Division Multiplexing

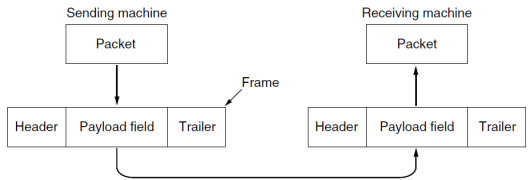


- In **TDM (Time Division Multiplexing)**, the users take turns (in a round-robin fashion), each one periodically getting the entire bandwidth for a little burst of time.
- Bits from each input stream are taken in a fixed time slot and output to the aggregate stream.
- Small intervals of guard time analogous to a frequency guard band may be added to accommodate small timing variations.

Table of Contents

Data Link layer design

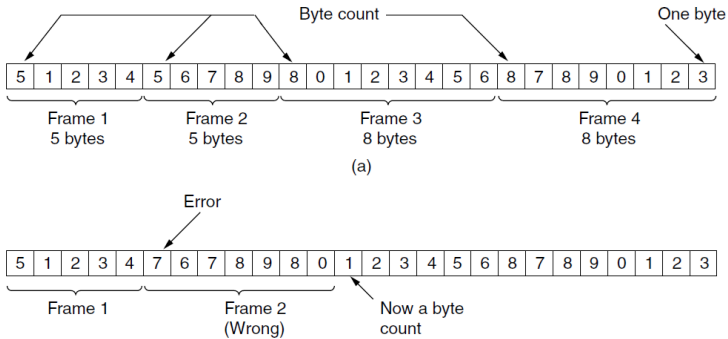
- The data link layer uses the services of the physical layer to send and receive bits over communication channels.
- It has a number of functions, including:
  - Providing a well-defined service interface to the network layer.
  - Dealing with transmission errors.
  - Handling access control to a shared medium.
- The data link layer takes the packets it gets from the network layer and encapsulates them into *frames*, containing a frame header, a payload field, and a frame trailer.



Framing

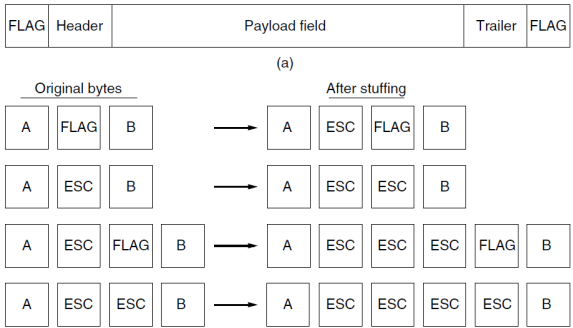
- The data link layer breaks up the bit stream into discrete frames, and computes a short token called a **checksum** for each frame.
- When a frame arrives at the destination, the checksum is recomputed. If the newly computed checksum is different from the one contained in the frame, the data link layer knows that an error has occurred and takes steps to deal with it (e.g., discarding the bad frame and possibly also sending back an error report).
- A good design of framing must **make it easy for a receiver to find the start of new frames** while using little of the channel bandwidth. We will look at three methods:
  - Byte count;
  - Flag bytes with byte stuffing;
  - Flag bits with bit stuffing.

Byte count



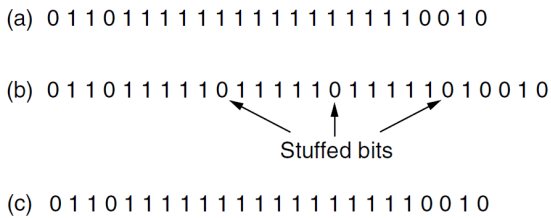
- This framing method uses a field in the header to specify the number of bytes in the frame.
- The trouble with this algorithm is that the count can be garbled by a transmission error.

Flag Bytes with Byte Stuffing



- Each frame starts and ends with special bytes, called a *flag byte* (FLAG).
- When a FLAG occurs in the data, a special escape byte (ESC) is inserted just before each FLAG.
- The data link layer on the receiving end removes the ESC before giving the data to the network layer. This technique is called **byte stuffing**.

Flag Bits with Bit Stuffing



- Framing can be also be done at the bit level, so frames can contain an arbitrary number of bits made up of units of any size.
- **HDLC (High-level Data Link Control) protocol:** Each frame begins and ends with a special bit pattern, 01111110 or $0x7E$ in hexadecimal.
- Whenever the sender's data link layer encounters five consecutive 1's in the data, it automatically stuffs a 0 bit into the outgoing bit stream.

Error Detection and Correction

- Errors typically happen during communications, and we have to deal with them
- Two basic approaches:
  - Include enough redundant information to enable the receiver to deduce what the transmitted data must have been. This strategy uses **error-correcting codes** and is often referred to as **Forward Error Correction (FEC).**
  - The other is to include only enough redundancy to allow the receiver to deduce that an error has occurred (but not which error) and have it request a retransmission. This strategy uses **error-detection codes** and is often referred to as **Automatic Repeat Request (ARQ).**
  - Depending on the service requirements, one technique can be preferred to the other, e.g., real-time services like video streaming use FEC

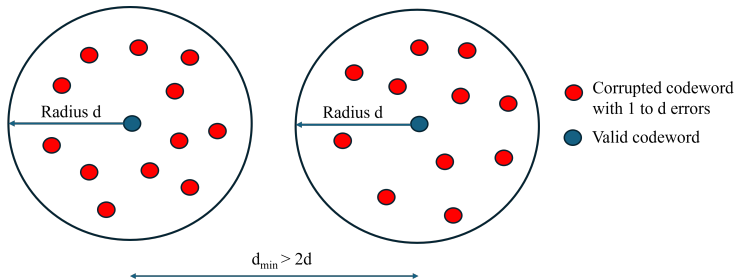
Error-Correcting Codes

- There exist several error-correcting codes: (1) Hamming codes, (2) Binary convolutional codes, (3) Reed-Solomon codes, (4) Low-Density Parity Check codes.
- All of these codes add redundancy to the information that is sent. A frame consists of m data (i.e., message) bits and r redundant (i.e. check) bits.
  - In a **block code**, the r check bits are computed solely as a function of the m data bits with which they are associated.
  - In a **systematic code**, the m data bits are sent directly, along with the check bits, rather than being encoded themselves before they are sent.
  - In a **linear code**, the r check bits are computed as a linear function of the m data bits.
- Let the total length of a block be n (i.e., $n = m + r$). We will describe this as an (n, m) code. An n-bit unit containing data and check bits is referred to as an n bit **codeword**.
- The **code rate** m/n is the fraction of the codeword that carries non redundant information.

Error-Correcting Codes

- The number of bit positions in which two codewords differ is called the **Hamming distance.** Its significance is that if two codewords are a Hamming distance d apart, it will require d single-bit errors to convert one into the other.
- **Example:** The codewords 10001001 and 10110001 have Hamming distance equal to 3.
- Given the complete list of the legal codewords, we can compute the smallest Hamming distance of the code.
- In most data transmission applications, all 2^m possible data messages are legal, but due to the way the check bits are computed, not all of the 2^n possible codewords are used.
- Only the small fraction of $2^m/2^n$ or $1/2^r$ of the possible messages will be legal codewords.
- It is the **sparseness** with which the message is embedded in the space of codewords that allows the receiver to detect and correct errors.

Error-Correcting Codes



- The error-detecting and error-correcting properties depend on the Hamming distance.
- To **reliably detect** d **errors**, you need a distance $d + 1$ code because with such a code there is no way that d single-bit errors can change a valid codeword into another valid codeword.
- T**o correct** d **errors**, you need a distance $2d + 1$ code because that way the legal codewords are so far apart that even with d changes the original codeword is still closer than any other.

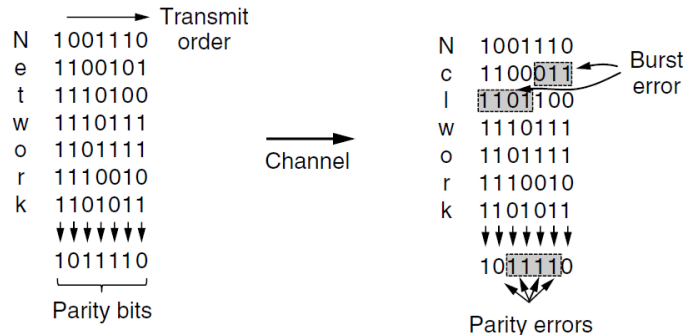
Error-Correcting Codes

- **Example:** Consider a code with only four valid codewords:
  - 0000000000
  - 0000011111
  - 1111100000
  - 1111111111
- This code has a distance of 5, i.e., it can correct double errors or detect quadruple errors.
- If the codeword 0000000111 arrives and we expect only single- or double-bit errors, the receiver will know that the original must have been 0000011111.
- If, however, a triple error changes 0000000000 into 0000000111, the error will not be corrected properly.
- Alternatively, if we expect all of these errors, we can detect them. None of the received codewords are legal codewords so an error must have occurred.

Error-Detecting Codes

- Error-detecting codes: (1) Parity, (2) Checksums, (3) Cyclic Redundancy Checks (CRCs).
- **Simplest approach:** Append a single **parity bit** to the data, chosen so that the number of 1 bits in the codeword is even (or odd).
- A code with a single parity bit has a distance of 2, since any single-bit error produces a codeword with the wrong parity. This means that it can **detect single-bit errors**.
- **Issue:** Long burst errors makes error detection complicated.
- **Possible solution:** Each block to be sent is regarded as a rectangular matrix n bits wide and k bits high. The parity bits over the data in a different order than the order in which the data bits are transmitted (**Interleaving**).

Error-Detecting Codes



- **Interleaving:** Compute a parity bit for each of the n columns and send all the data bits as k rows. At the last row, we send the n parity bits.
- It uses n parity bits on blocks of kn data bits to detect a single burst error of length n or less.

Error-Detecting Codes: Checksums

- **Checksum:** A group of check bits associated with a message used to detect errors in the transmitted data.

Functionality:

- Based on a running sum of data bits.
- Usually placed at the end of the message.
- Sum entire received codeword (data bits + checksum). Result = 0 indicates no error.

Example: 16-bit Internet Checksum (IP Protocol)

- Sum of message bits divided into 16-bit words.
- Because this method operates on words rather than on bits, as in parity, errors that leave the parity unchanged can still alter the sum and be detected.

The Medium Access Control Sub-layer

- Network links can be divided into two categories: those using point-to-point connections and those using broadcast channels (a.k.a. multiaccess channels or random access channels).
- In any broadcast network, the key issue is how to determine who gets to use the channel when there is competition for it.
- The protocols used to determine who goes next on a multiaccess channel belong to a sublayer of the data link layer called the **MAC (Medium Access Control) sublayer.**
- The MAC sublayer is especially important in LANs, particularly wireless ones because wireless is naturally a broadcast channel.

Static Channel Allocation

- Fixed multiplexing scheme, e.g., FDM. If there are N users, the bandwidth is divided into N equal-sized portions, with each user being assigned one portion.
- Simple and efficient in the presence of a small and constant number of users.
- When the number of senders is large and varying or the traffic is bursty, FDM has problems:
  - If fewer than N users are currently interested in communicating, a large piece of valuable spectrum will be wasted.
  - If more than N users want to communicate, some of them will be denied permission for lack of bandwidth.
- The same arguments that apply to time division multiplexing (TDM): If a user does not use the allocated time slot, it would just lie fallow.

Static FDM Performance Analysis

- The mean time delay T to send a frame on a channel with capacity C bps (M/M/1 queue):

$$T = \frac{1}{\mu C - \lambda}$$

  - λ: frame arrival rate (frames/sec)
  - $\frac{1}{\mu}$: mean frame length (bits)
  - μC: service rate (frames/sec)

- **Example:** $C = 100$ Mbps, $\frac{1}{\mu} = 10,000$ bits, $\lambda = 5000$ frames/sec $\quad\Rightarrow\quad T = 200\mu$s

- Dividing into N subchannels, each with capacity $\frac{C}{N}$ bps and input rate $\frac{\lambda}{N}$:

$$T_N = \frac{1}{\frac{\mu C}{N} - \frac{\lambda}{N}} = N \cdot T$$

  The mean delay increases proportionally to the number N of users

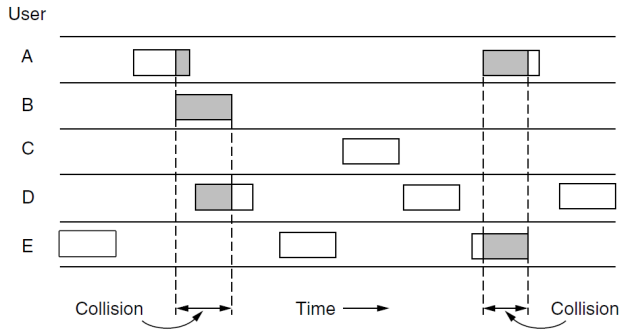
Dynamic Channel Allocation

Key assumptions:

1. **Independent Traffic:** N independent stations generate frames at rate λ. A station blocks after generating a frame until it's transmitted.
2. **Single Channel:** One channel for all communication, with equal-capacity stations. Protocols may assign different roles.
3. **Observable Collisions:** Simultaneous transmissions cause collisions, detectable by all stations. Collided frames must be retransmitted.
4. **Continuous or Slotted Time:** Time can be continuous (transmissions start anytime) or slotted (transmissions start at slot beginnings).
5. **Carrier Sense or No Carrier Sense:** With carrier sense, stations detect channel use before transmitting. Without it, they transmit without sensing and detect success later.

MAC Protocols: ALOHA

- **Simple idea:** let users transmit whenever they have data to be sent.
- Whenever two frames try to occupy the channel at the same time, there will be a **collision** and both will be garbled.
- After a collision, the sender just waits a random amount of time and sends it again.



MAC Protocols: ALOHA

- **Question**: What is the efficiency of ALOHA (i.e., what fraction of all frames escape collisions)?
- Let the **frame time** denote the amount of time needed to transmit a (fixed-length) frame
- Frames (newly) generated (and retransmitted) by the stations are well modeled by a Poisson distribution with a mean of G frames per frame time.
- **Throughput:** $S = GP_0$, where P_0 is the probability that a frame does not suffer a collision.
- The probability that k frames are generated during a given frame time, in which G frames are expected, is given by the Poisson distribution:

$$\Pr[k] = \frac{G^k e^{-G}}{k!}$$

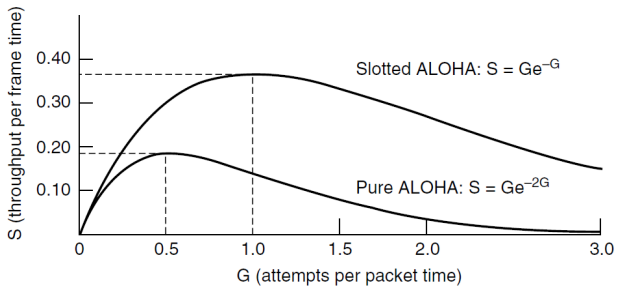
- In an interval two frame times long, we have $P_0 = e^{-2G}$ and, finally,

$$S = Ge^{-2G}$$

MAC Protocols: Slotted ALOHA

- The maximum throughput of Aloha occurs at $G = 0.5$, with $S = 1/2e$, which is about 0.184.
- **Slotted Aloha:** Divide time into discrete slots, each interval corresponding to one frame.
- Slotted time halves the vulnerable period. The probability of no other traffic during the same slot as our test frame is then e^{-G}, which leads to

$$S = Ge^{-G}$$



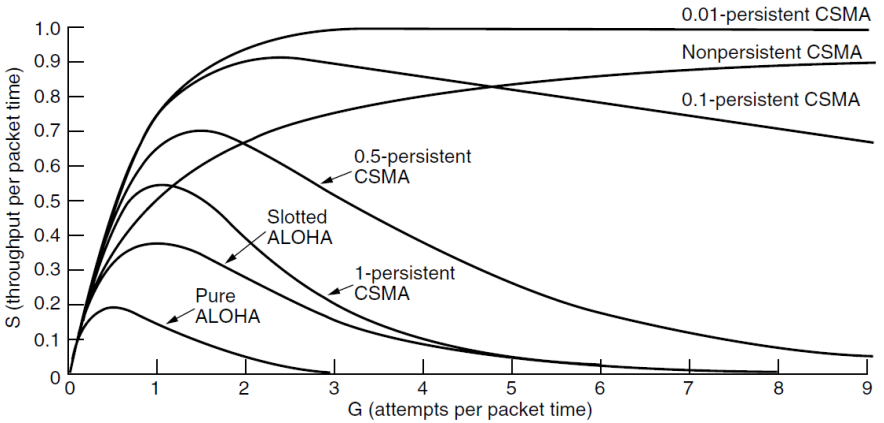
MAC Protocols: Carrier Sense Multiple Access (CSMA)

- Protocols in which stations listen for a carrier (i.e., a transmission) and act accordingly are called **carrier sense** protocols.
- **1-persistent CSMA:**
  - Every station listens to the channel: If idle, the stations sends its data; if busy, the station waits until it becomes idle. Then it transmits a frame.
  - If a collision occurs, the station waits a random amount of time and starts all over again.
  - 1-persistent: the station transmits with a probability of 1 when the channel is idle.
- Issues of 1-persistent CSMA:
  - Stations wait politely until a transmission ends, and then will begin transmitting exactly simultaneously, resulting in a collision.
  - The propagation delay has an important effect on collisions.

MAC Protocols: Carrier Sense Multiple Access (CSMA)

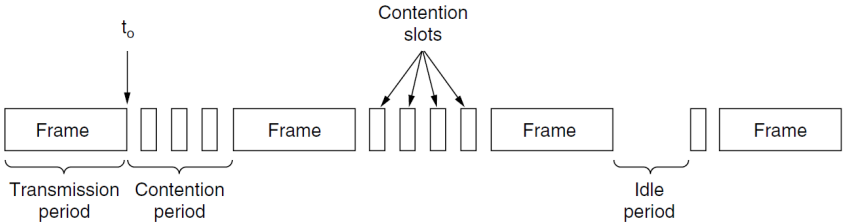
- **nonpersistent CSMA:** If the channel is already in use, the station does not continually sense it, but it waits a random period of time and then senses again
- p**-persistent CSMA:**
  - It applies to slotted channels
  - When a station becomes ready to send, it senses the channel. If it is idle, it transmits with a probability p. With a probability $q = 1 - p$, it defers until the next slot.
  - If that slot is also idle, it either transmits or defers again, with probabilities p and q.
  - This process is repeated until either the frame has been transmitted or another station has begun transmitting.
  - In the latter case, the station acts as if there had been a collision (i.e., it waits a random time and starts again).
  - If the station initially senses that the channel is busy, it waits until the next slot and applies the above algorithm.

MAC Protocols: Carrier Sense Multiple Access (CSMA)



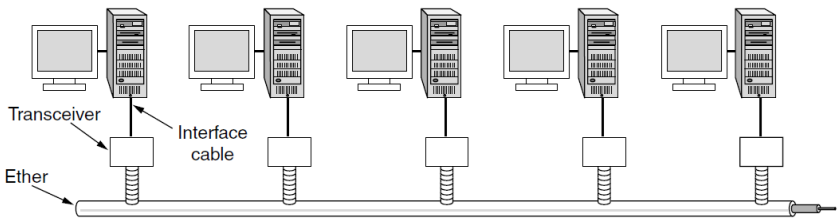
MAC Protocols: Carrier Sense Multiple Access (CSMA)

- **CSMA/CD (CSMA with Collision Detection):** Stations can quickly detect the collision and abruptly stop transmitting, (rather than finishing them) since they are garbled anyway
  - At the point t_0, a station has finished transmitting its frame. Any other station having a frame to send may now attempt to do so.
  - If two or more stations decide to transmit simultaneously, there will be a collision.
  - If a station detects a collision, it aborts its transmission, waits a random period of time, and then tries again.



Ethernet

- **Ethernet:** IEEE 802.3 standard. Probably the most ubiquitous kind of network in the world.
- Two kinds of Ethernet exist:
  - **Classic Ethernet**, which uses the MAC protocols we have studied;
  - **Switched Ethernet**, in which devices called *switches* connect different computers.
- Classic Ethernet looks as a single long cable to which all the computers were attached. There is a maximum cable length per segment, which can be extended through repreaters.



Classic Ethernet: Frame format

- Preamble of 8 bytes, each containing the bit pattern 10101010 (in the last byte, the last 2 bits are set to 11). This last byte is called the Start of Frame delimiter (SoF).
- Two *link-layer addresses* (6 bytes long), one for the destination and one for the source. Addresses are globally unique for every computer.
- The *Length* field contains the length of the frame.
- Next come the *data*, from 64 to 1500 bytes. If the data portion of a frame is less than 46 bytes, the *Pad* field is used to fill out the frame to the minimum size.
- The *checksum* contains an error-detecting code used to determine if the bits of the frame have been received correctly. If an error is detected, the frame is dropped.

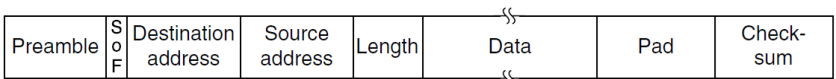


Figure: IEEE 802.3 frame format.

Classic Ethernet: MAC protocol

- Classic Ethernet uses the 1-persistent CSMA/CD algorithm.
- Collisions are detected and aborted with a jam signal.
- Random interval retransmission after collisions.

Random interval determination: Binary exponential backoff

- Time divided into discrete slots equal to worst-case roundtrip propagation time.
- Slot time set to 512 bit times, or 51.2 sec.
- After i collisions, wait a random number of slots between 0 and $2^i - 1$.
- Maximum randomization interval frozen at 1023 slots after 10 collisions.
- Failure reported after 16 collisions.
- The method adapts to number of stations sending, and ensures low delay with few collisions.

Classic Ethernet: Performance

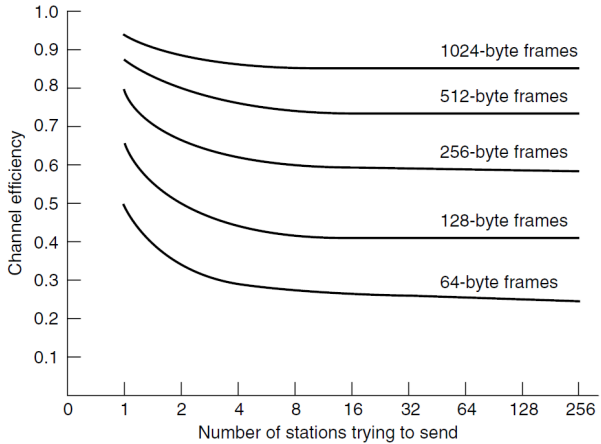
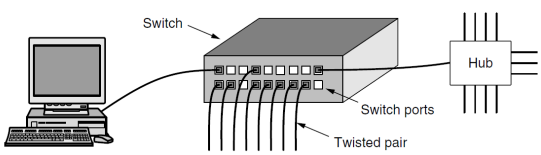
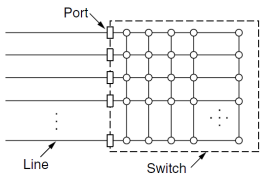


Figure: Efficiency of Ethernet at 10 Mbps with 512-bit slot times.

Switched Ethernet

- A switch contains a high-speed backplane that connects all of the ports (typically, 4 to 48)
- Each cable connects the switch to a single computer
- Switches only output frames to the ports for which those frames are destined. An association is made between Ethernet addresses and ports.
- In a switch, each port is its own independent collision domain. If cable is full duplex, both the station and the port can send a frame on the cable at the same time.





Switched Ethernet

3 The Link Layer

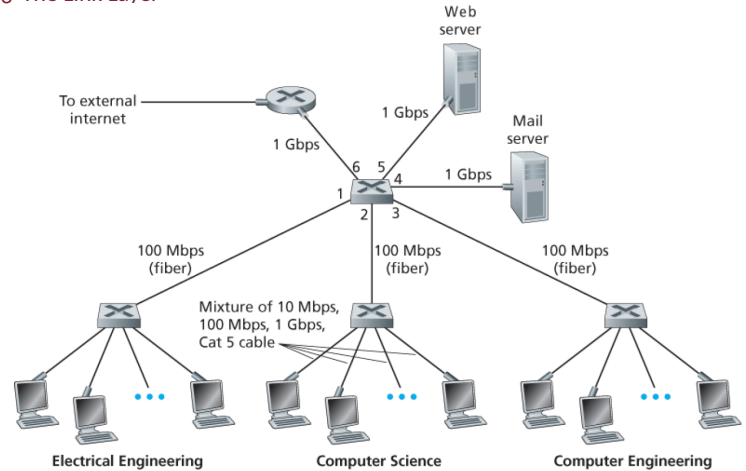


Figure: An institutional network connected together by four switches.

MAC addresses

- Hosts and routers have link-layer addresses associated with every adapter (i.e., interface).
- A link-layer address is called a LAN address, a physical address, or a MAC address.
- MAC addresses are 6 bytes long (i.e., 2^{48} possible addresses), and are typically expressed in hexadecimal notation.
- MAC addresses are uinique: First 24 bits are associated with the manufacturer; the other 24 bits are used to identify the single adapter.

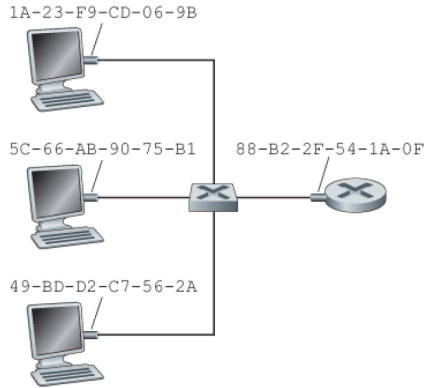
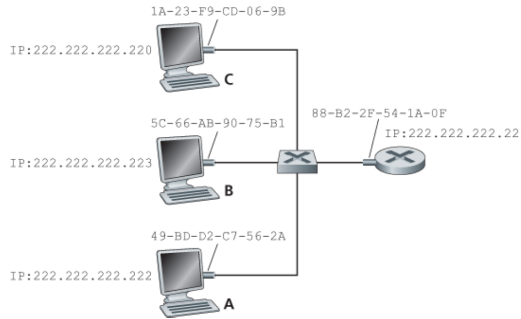


Figure: Interfaces and MAC addresses.

Address Resolution Protocol

- **Address Resolution Protocol (ARP):** Translate network-layer addresses (IP addresses) into MAC addresses
- ARP resolves IP addresses only for hosts and router interfaces on the same subnet.
- Each host and router has an ARP table in its memory, which contains mappings of IP addresses to MAC addresses.
- The ARP table also contains a timeto- live (TTL) value, which indicates when each mapping will be deleted from the table.



IP Address	MAC Address	TTL
222.222.222.221	88-B2-2F-54-1A-0F	13:45:00
222.222.222.223	5C-66-AB-90-75-B1	13:52:00

Figure: Interfaces, IP and MAC addresses.

Switch Filtering and Forwarding

- **Filtering** is the switch function that determines whether a frame should be forwarded to some interface or should just be dropped.
- **Forwarding** is the switch function that determines the interfaces to which a frame should be directed, and then moves the frame to those interfaces.
- Switch filtering and forwarding are done with a **switch table**.
- An entry in the switch table contains: (1) a MAC address, (2) the switch interface that leads toward that MAC address, and (3) the time at which the entry was placed in the table.

Address	Interface	Time
62-FE-F7-11-89-A3	1	9:32
7C-BA-B2-B4-91-10	3	9:36
....	....	....

Figure: Switch Table.

Switch Filtering and Forwarding

Example: Suppose a frame with destination address `DD-DD-DD-DD-DD-DD` arrives at the switch on interface `x`. The switch indexes its table with the MAC address. There are three possible cases:

1. **No entry in the table:**
   - The switch forwards copies of the frame to all interfaces except `x` (broadcasts the frame).
2. **Entry exists, associated with interface `x`:**
   - The frame is coming from a LAN segment containing the destination adapter.
   - The switch discards the frame (filtering function).
3. **Entry exists, associated with another interface `y`:**
   - The frame needs to be forwarded to the LAN segment attached to interface `y`.
   - The switch forwards the frame to interface `y` (forwarding function).

Self-Learning Switches

Switches build their tables automatically, dynamically, and autonomously, without administrator intervention. This self-learning capability is achieved as follows:

1. The switch table is initially empty.
2. For each incoming frame, the switch records in its table:
   - The source MAC address;
   - The arrival interface;
   - The current time.

   In this manner the switch records in its table the LAN segment on which the sender resides.
3. If no frames are received from a MAC address within a certain time (aging time), it is removed from the table. In this manner, if a PC is replaced by another PC (with a different adapter), the MAC address of the original PC will eventually be purged from the switch table.

Table of Contents

4 The Network Layer

- The network layer is concerned with getting packets from the source to the destination.
- To achieve its goals, the network layer must know about the topology of the network (i.e., the set of all routers and links) and choose appropriate paths through it.
- It must also take care when choosing routes to avoid overloading some of the communication lines and routers while leaving others idle.

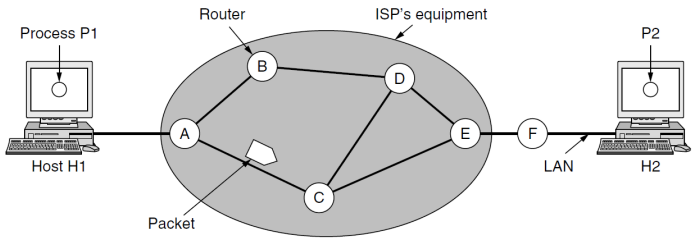
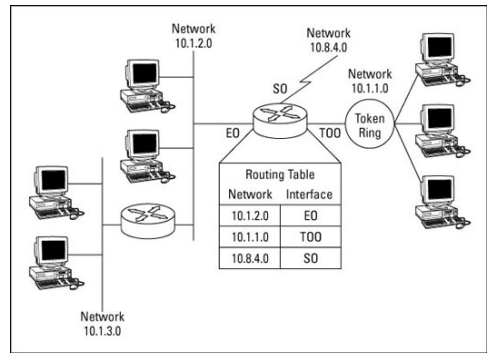


Figure: The environment of the network layer protocols.

Routing Algorithms

- The **routing algorithm** is that part of the network layer software responsible for deciding which output line an incoming packet should be transmitted on.
- A router has two processes inside it:
  - **Forwarding:** It handles each packet as it arrives, looking up the outgoing line to use for it in the *routing tables*.
  - **Routing:** Filling in and updating the routing tables.
- Routing algorithms can be static or adaptive.



The Optimality Principle

- **Bellman optimality principle:** It states that if router J is on the optimal path from router I to router K, then the optimal path from J to K also falls along the same route.
- Optimal routes from all sources to a destination form a **sink tree** rooted at the destination.
- A sink tree is not necessarily unique; other trees with the same path lengths may exist.

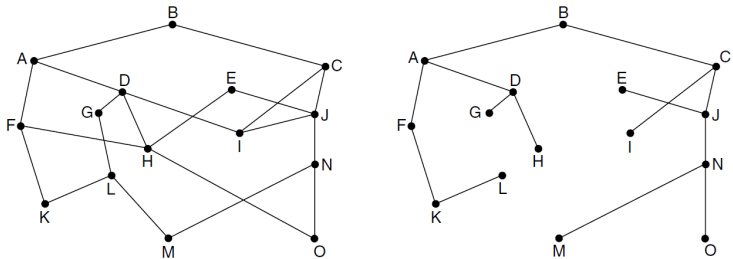


Figure: A network (Left). A sink tree for router B (Right).

The Shortest Path Algorithm

- **Idea:** To build a graph of the network, with each node of the graph representing a router and each edge of the graph representing a communication line, or link.
- To choose a route, the algorithm finds the **shortest path** between two nodes on the graph.
- Measures of path length: Number of hops, distance, mean delay, bandwidth, etc.

Dijkstra's Algorithm (1959)

- Nodes are labeled with the (non-negative) distance from the source along the best path.
- Initially, all nodes are labeled with infinity. Labels change as better paths are found.
  - Initially, all labels are tentative.
  - Once a label represents the shortest path, it becomes permanent and does not change.
  - Complexity $O(N^2)$, where N is the number of nodes.

The Shortest Path Algorithm

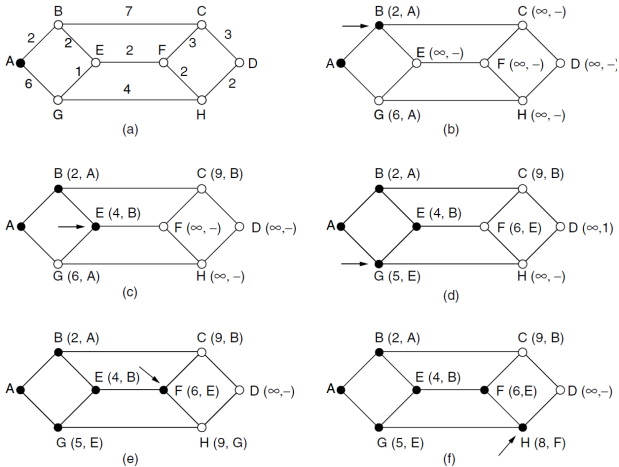


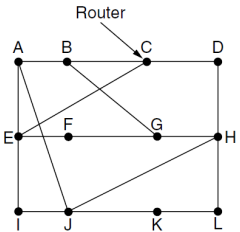
Figure: The first six steps used in computing the shortest path from A to D.

Distance Vector Routing

- A **distance vector routing** algorithm operates by having each router maintain a table giving the best known distance to each destination and which link to use to get there (next hop).
- The router is assumed to know the ”distance” to each of its neighbors.
- If the metric is hops, the distance is just one hop. If the metric is propagation delay, the router can measure it directly with special ECHO packets that the receiver just timestamps and sends back as fast as it can.
- **Example:** Once every T msec, each router sends to each neighbor a list of its estimated delays to each destination. It also receives a similar list from each neighbor.
  - From neighbor X: X_i denotes X’s estimate of how long it takes to get to router i.
  - If the router knows that the delay to X is m msec, it also knows that it can reach router i via X in X_i + m msec.

Distance Vector Routing

4 The Network Layer



To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K

JA delay is 8 | JI delay is 10 | JH delay is 12 | JK delay is 6

Vectors received from J's four neighbors

New routing table for J

Figure: (Left) A network. (Right) Input from A, I, H, K, and the new routing table for J.

Link State Routing

- Each router must do the following things:
  - Discover its neighbors and learn their network addresses.
  - Set the distance or cost metric to each of its neighbors.
  - Construct a packet telling all it has just learned.
  - Send this packet to and receive packets from all other routers.
  - Compute the shortest path to every other router.
- The complete topology is distributed to every router. Then Dijkstra's algorithm can be run at each router to find the shortest path to every other router.
- Variants of link state routing called IS-IS and OSPF are the routing algorithms that are most widely used inside the Internet today.

Link State Routing: Learning about Neighbors

- Routers send a special HELLO packet on each point-to-point line.
- The router on the other end is expected to send back a reply giving its (unique) name.
- When two or more routers are connected by a broadcast link (e.g., a switch, ring, or classic Ethernet), the LAN is modeled as an additive artificial node (”N” in the figure)

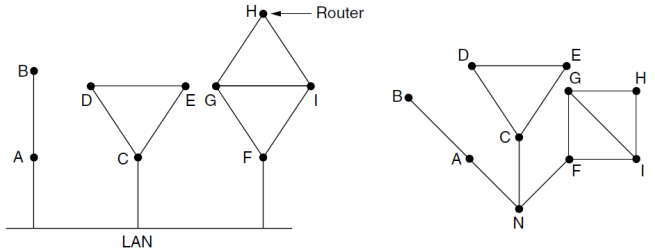


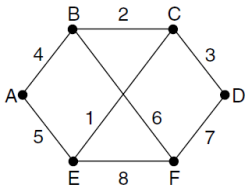
Figure: (Left) Nine routers and a broadcast LAN. (Right) A graph model.

Link State Routing: Setting Link Costs

- Each link requires a distance or cost metric for finding shortest paths.
- Cost to reach neighbors can be set automatically or configured by the network operator.
- A common choice: cost inversely proportional to the bandwidth of the link (e.g., 1-Gbps Ethernet: cost 1, 100-Mbps Ethernet: cost 10).
- In geographically spread out networks, link delay may be factored into the cost to favor shorter links.
- Delay can be determined by sending an ECHO packet and measuring the round-trip time, then dividing by two for an estimate.

Link State Routing: Building Link State Packets

- Once the information needed for the exchange has been collected, each router builds a packet containing all the data.
- The packet starts with the identity of the sender, followed by a sequence number, an age, and a list of neighbors. The cost to each neighbor is also given.



Link State Packets

A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8

Figure: (Left) A network. (Right) The link state packets for this network.

Link State Routing: Distributing Link State Packets

- Use flooding to distribute the link state packets to all routers. To keep the flood in check, each packet contains a sequence number that is incremented for each new packet sent.
- Routers keep track of all the (source router, sequence) pairs they see.
- When a new link state packet comes in, it is checked against the list of packets already seen.
  - If it is new, it is forwarded on all lines except the one it arrived on.
  - If it is a duplicate, it is discarded.
  - Packets with a sequence number lower than the highest one seen so far are rejected as being obsolete
- **Issues:** if the sequence numbers wrap around, confusion will reign. Also, if a router ever crashes, it will lose track of its sequence number.
- **Solution:** To include the age of each packet after the sequence number and decrement it once per second. When the age hits zero, the information from that router is discarded.

Link State Routing: Compute the New Routes

- Once a router has accumulated a full set of link state packets, it can construct the entire network graph because every link is represented.
- Every link is represented twice, once for each direction. The different directions may even have different costs.
- The Dijkstra's algorithm can be run locally to find the shortest paths to all destinations.
- The results of this algorithm tell the router which link to use to reach each destination. This information is installed in the routing tables.
- **Complexity:** For a network with n routers, each of which has k neighbors, the memory required is proportional to kn, and the computation time grows faster than kn
- In many practical situations, link state routing works well because it does not suffer from slow convergence problems.

Network Layer in the Internet

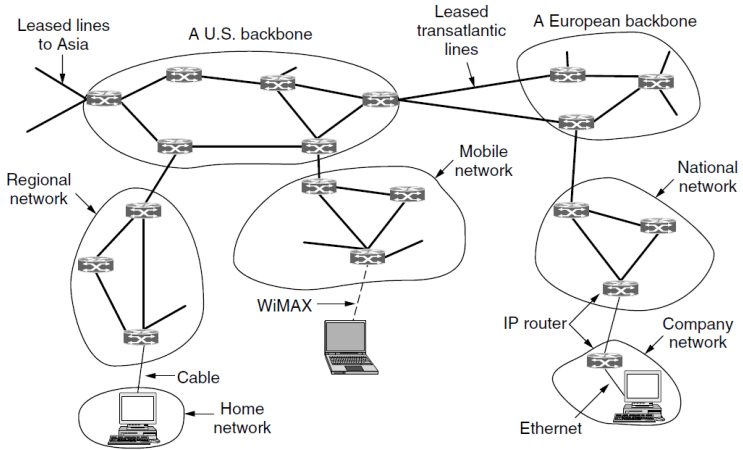


Figure: The Internet is an interconnected collection of many networks.

The Internet Protocol

- The **Internet protocol (IP)** is essential for the Internet's functionality.
- IP was designed with internetworking in mind from the beginning. It provides a best-effort way to transport packets from source to destination.
- **Communication over the Internet:**
  - The transport layer breaks data streams into IP packets (not more than 1500 bytes).
  - IP routers forward each packet through the Internet until the destination is reached.
  - A packet typically traverses several networks and a large number of IP routers before getting to destination.
  - At the destination, the network layer reassembles the data into the original datagram and hands it to the transport layer for the receiving process.

IP Version 4 Protocol

- An IPv4 datagram consists of a header part and a body or payload part.
- The header has a 20-byte fixed part and a variable-length optional part.

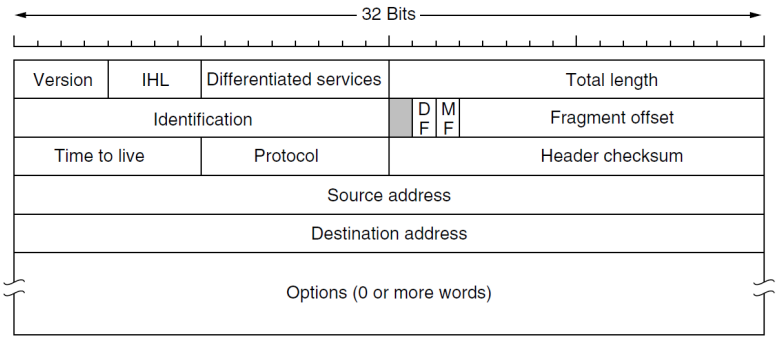


Figure: The IPv4 (Internet Protocol) header.

IP Version 4 Protocol

- **Version** field indicates which version of the protocol the datagram belongs to (e.g. 4 or 6).
- The **IHL** field tells how long the header is, in 32-bit words (from 5 to 15).
- The **Differentiated services** field distinguishes between different classes of service.
- The **Total length** includes everything in the datagram—both header and data.
- The **Identification** allows to determine which packet a newly arrived fragment belongs to.
- **DF** stands for Don't Fragment. **MF** stands for More Fragments. The **Fragment offset** tells where in the current packet this fragment belongs.
- The **TtL** (Time to live) field is a counter used to limit packet lifetimes.
- The **Protocol** field tells it which transport process to give the packet to (e.g., TCP or UDP).
- The **Header checksum** helps detect errors while the packet travels through the network.
- The **Source address** and **Destination address** contain the associated IP addresses.
- The **Options** field allow to include information not present in the original design.

IP addresses

- Each 32-bit address is comprised of a variable-length network portion in the top bits and a host portion in the bottom bits.
- The network portion (a.k.a., prefix) has the same value for all hosts on a single network.
- IP addresses are written in dotted decimal notation. Prefixes are described by their length, which corresponds to a binary **subnet mask** of 1s.
- **Typical notation:** 128.208.0.0/24 uses 24 bits for the network and 8 bits for the hosts.

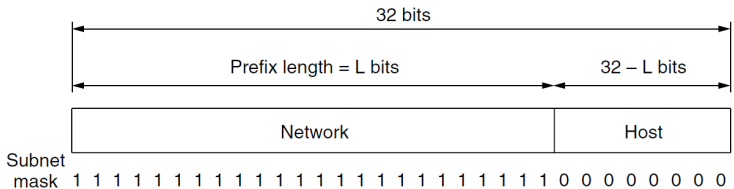
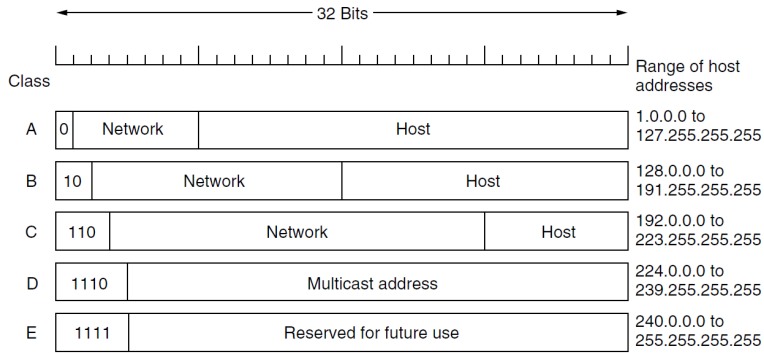


Figure: An IP prefix and a subnet mask.

Classful addressing

- Before 1993, IP addresses were divided into the five categories.
- Unfortunately, organizing the address space by classes wastes millions of them.



Subnets

- **Subnetting:** An available block of addresses is split into several parts for internal use as multiple networks, while still acting like a single network to the outside world.

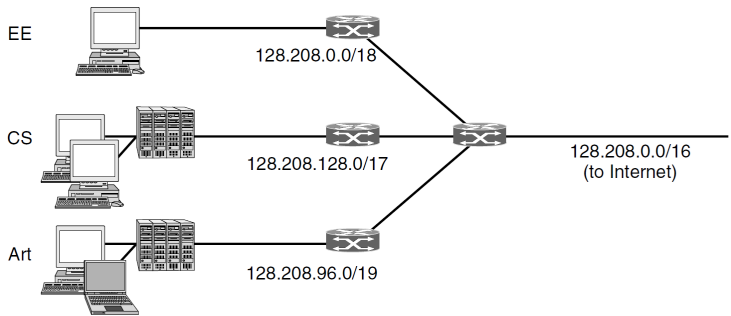


Figure: Splitting an IP prefix into separate networks with subnetting.

Subnetting

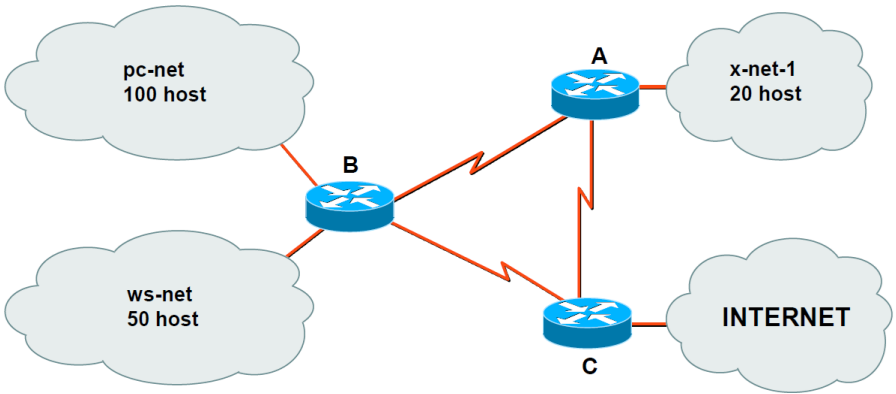
- The single /16 has been split into pieces. This split does not need to be even, but each piece must be aligned so that any bits can be used in the lower host portion.
  - Half of the block (a /17) is allocated to Computer Science
  - A quarter is allocated to Electrical Engineering (a /18).
  - One eighth (a /19) to Art.
  - The remaining eighth is unallocated.
- Resulting prefixes written in binary notation:

Computer Science:	`10000000 11010000 1|xxxxxxx xxxxxxxx`
Electrical Engineering:	`10000000 11010000 00|xxxxxx xxxxxxxx`
Art:	`10000000 11010000 011|xxxxx xxxxxxxx`

- When a packet arrives, the router looks at the destination address of the packet and checks which subnet it belongs to (ANDing the destination address with the mask for each subnet).

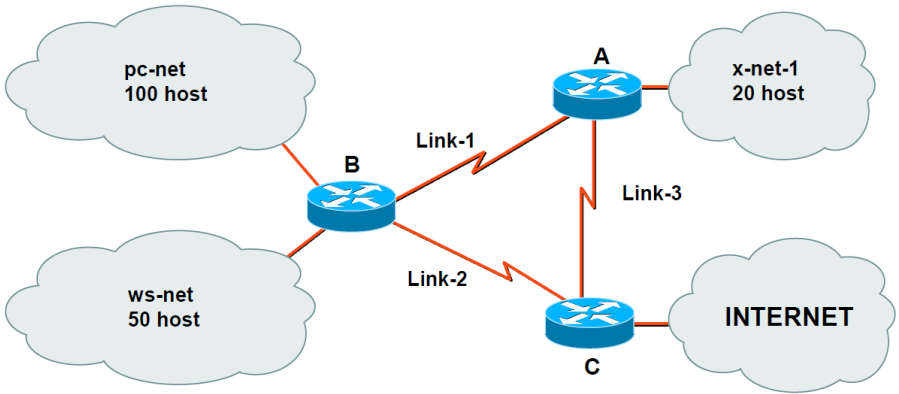
Exercise 1: IP Subnetting

- Subnetting from 192.168.0.0/24



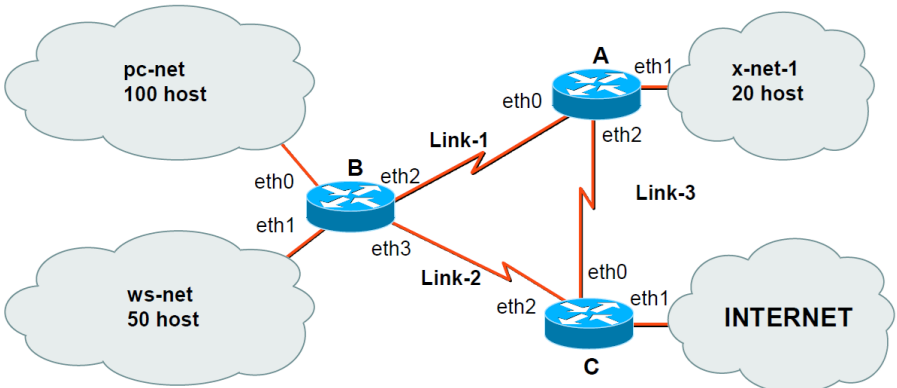
Exercise 1: IP Subnetting

- How many networks? **6**



Exercise 1: IP Subnetting

- IP addressess are associated to every interface



Exercise 1: IP Subnetting

n_h = n° of bits Host_Id $\rightarrow 2^{n_h} - 2 \geq$ n° of IP addresses of subnets

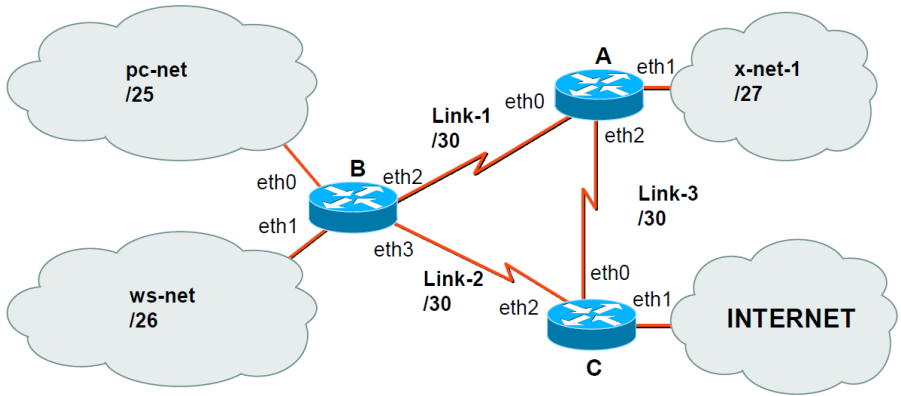
- **pc-net**
  - 101 IP addresses: 100 hosts + 1 (eth0 of router)
  - $n_h = 7 \rightarrow /25$
- **ws-net**
  - 51 IP addresses: 50 hosts + 1 (eth1 of router)
  - $n_h = 6 \rightarrow /26$
- **x-net-1**
  - 21 IP addresses: 20 hosts + 1 (eth1 of router)
  - $n_h = 5 \rightarrow /27$

Exercise 1: IP Subnetting

- **Link-1**
  - 2 IP addresses : eth2 of router B + eth0 of router A
  - $n_h = 2 \rightarrow /30$
- **Link-2**
  - 2 IP addresses : eth3 of router B + eth2 of router C
  - $n_h = 2 \rightarrow /30$
- **Link-3**
  - 2 IP addresses: eth2 of router A + eth0 of router C
  - $n_h = 2 \rightarrow /30$

Exercise 1: IP Subnetting

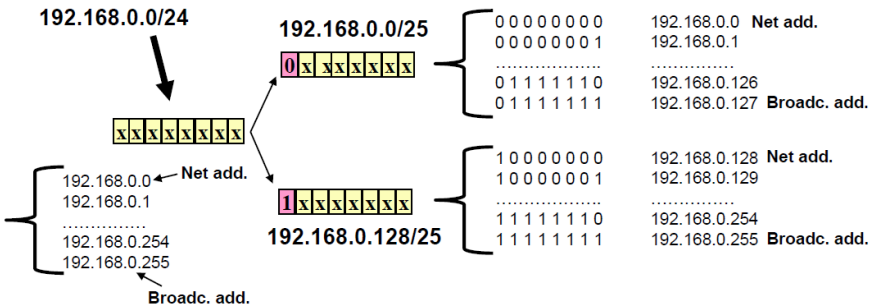
4 The Network Layer



Exercise 1: IP Subnetting

4 The Network Layer

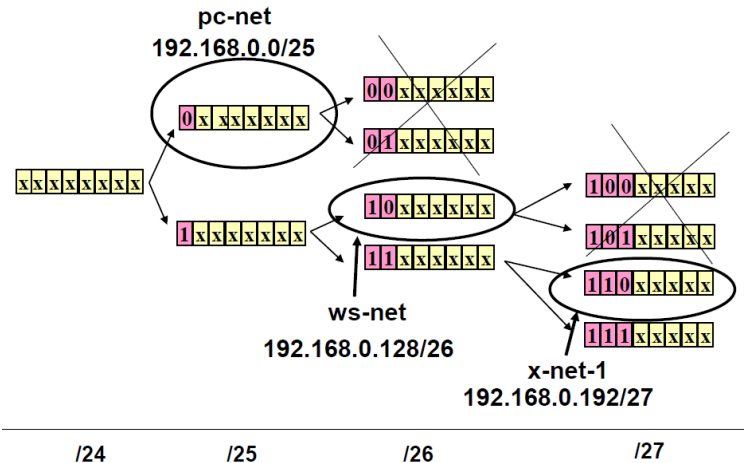
- Last byte of the IP address



Netmask	/24	/25

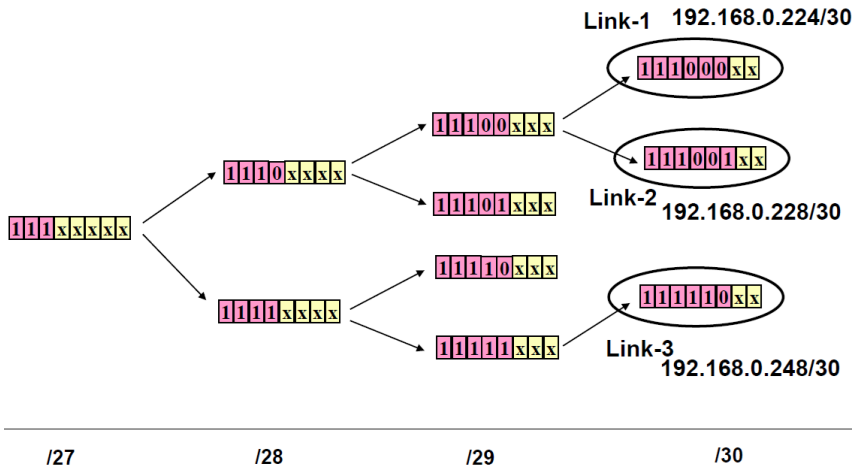
Exercise 1: IP Subnetting

4 The Network Layer

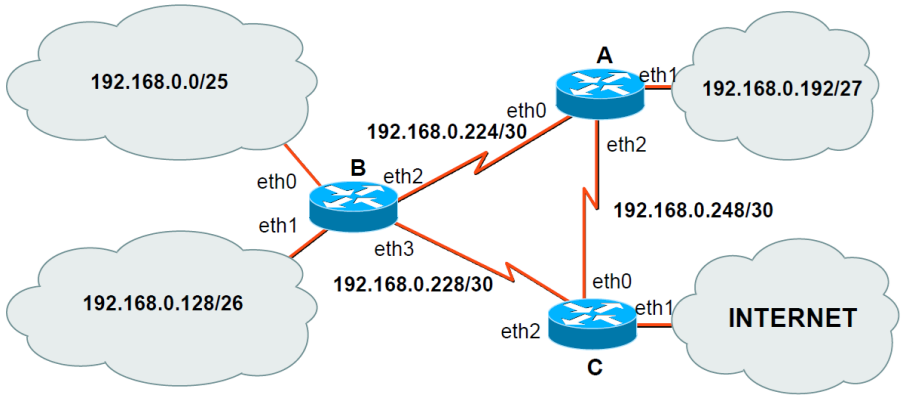


Exercise 1: IP Subnetting

4 The Network Layer



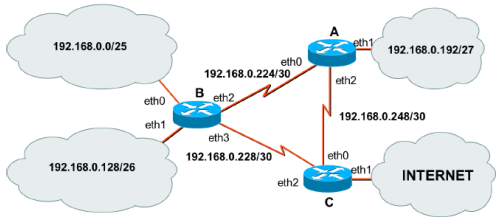
Exercise 1: IP Subnetting



Exercise 1: IP Assignment

- **pc-net**: 192.168.0.0 255.255.255.128
  - eth0 of router B: 192.168.0.1
  - Hosts (100): 192.168.0.2 → 192.168.0.101
- **ws-net:** 192.168.0.128 255.255.255.192
  - eth1 of router B: 192.168.0.129
  - Hosts (50): 192.168.0.130 → 192.168.0.179
- **x-net-1:** 192.168.0.192 255.255.255.224
  - eth1 of router A: 192.168.0.193
  - Hosts (20): 192.168.0.194 → 192.168.0.213
- **Link-1:** 192.168.0.224 255.255.255.252
  - eth2 of router B: 192.168.0.225; eth0 of router A: 192.168.0.226
- **Link-2:** 192.168.0.228 255.255.255.252
  - eth3 of router B: 192.168.0.229; eth2 of router C: 192.168.0.230
- **Link-3:** 192.168.0.248 255.255.255.252
  - eth2 of router A: 192.168.0.249; eth0 of router C: 192.168.0.250

Exercise 1: Routing Table of Router B



Destination N	Netmask M	Next hop NH	Interface I
192.168.0.0	255.255.255.128	d.c.	eth0
192.168.0.128	255.255.255.192	d.c.	eth1
192.168.0.224	255.255.255.252	d.c	eth2
192.168.0.228	255.255.255.252	d.c.	eth3
192.168.0.192	255.255.255.224	192.168.0.226 *	eth2
192.168.0.248	255.255.255.252	192.168.0.230 *	eth3
0.0.0.0	0.0.0.0	192.168.0.230 *	eth3

Classless Inter-Domain Routing

- **Route aggregation:** Combine multiple small prefixes into a single larger prefix (a.k.a. **supernet**). With aggregation, IP addresses are contained in prefixes of varying sizes.
- It avoids routing table explosion, and is called **Classless Inter-Domain Routing (CIDR)**.

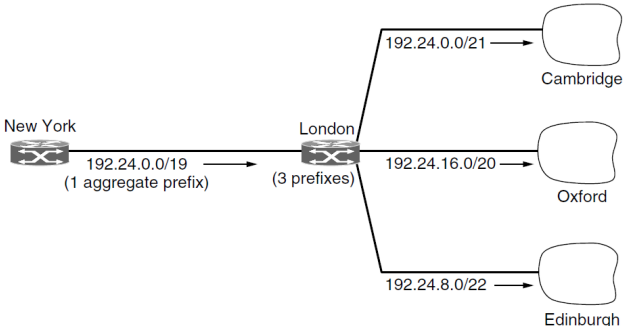


Figure: Aggregation of IP prefixes.

Classless Inter-Domain Routing

- Packets are sent in the direction of the most specific route: **longest matching prefix**.
- **Example:** New York router sending traffic to London and San Francisco.
- The router uses a single aggregate prefix to send traffic to three universities in London, but now allocates part of this prefix to a network in San Francisco.
- With longest matching prefix, one prefix is used to direct traffic to London; one more specific prefix is also used to direct a portion of the larger prefix to San Francisco.

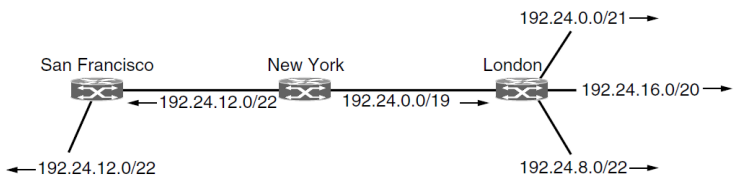


Figure: Longest matching prefix routing at the New York router.

IP Version 6 Protocol

- **Issue:** IPv4 is running out of addresses due to the exponential growth of the Internet
- IPv6 uses a larger addressing space with 128 bits: It provides an effectively unlimited supply of Internet addresses.
- IPv6 simplifies the header, allowing routers to process packets faster and thus improving throughput and delay.

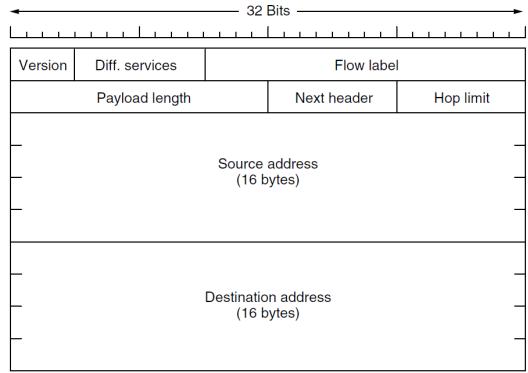


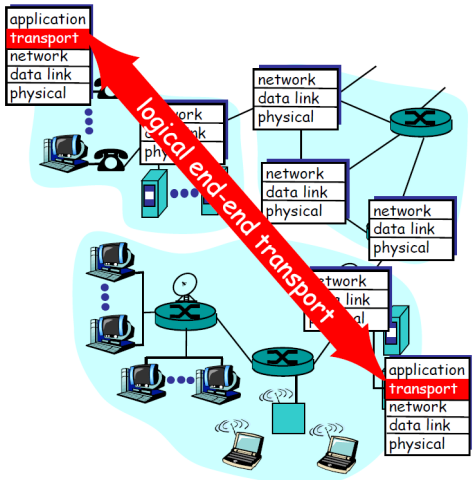
Figure: The IPv6 header.

Table of Contents

5 The Transport Layer

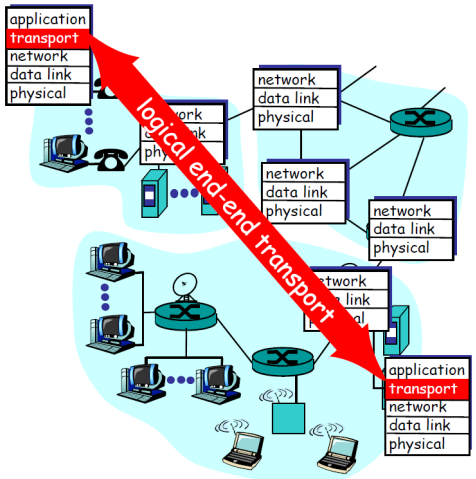
Transport Services and Protocols

- Provide **logical communication** between processes running on different hosts
- Transport protocols run in end systems
- Transport vs network layer services:
  - **Network layer:** data transfer between end systems
  - **Transport layer:** data transfer between processes
    - Relies on, enhances, network layer services



Transport Services and Protocols

- Transport Protocols:
  - **User Datagram Protocol (UDP):** unreliable (”best effort”), unordered data delivery
  - **Transmission Control Protocol (TCP):** reliable, in order data delivery
    - Congestion
    - Flow control
    - Connection setup



Segments

- We use the term **segment** for messages sent from transport entity to transport entity.

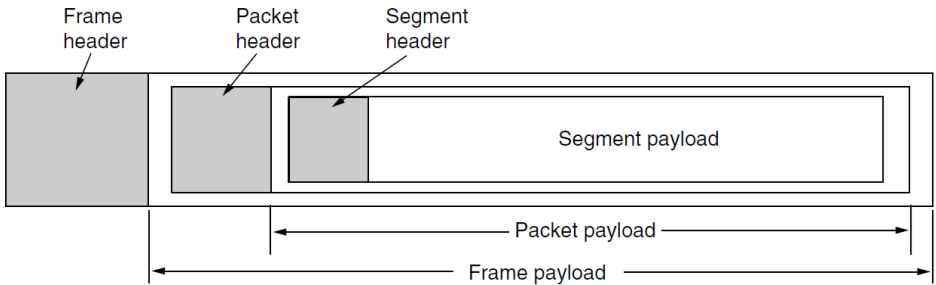


Figure: Nesting of segments, packets, and frames.

UDP: User Datagram Protocol

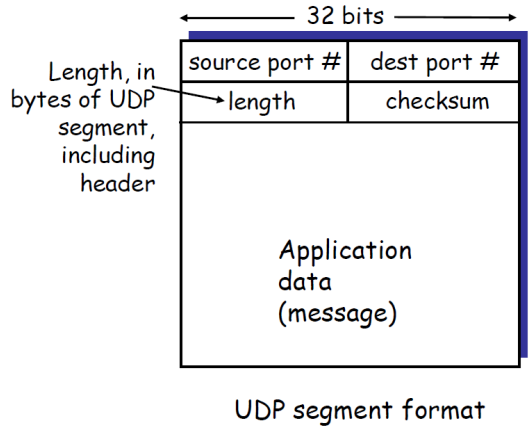
- ”Best effort”, UDP segments may be:
  - lost
  - delivered out of order to app
- **Connectionless**
  - no handshaking between UDP sender, receiver
  - each UDP segment handled independently of others

Why is there a UDP?

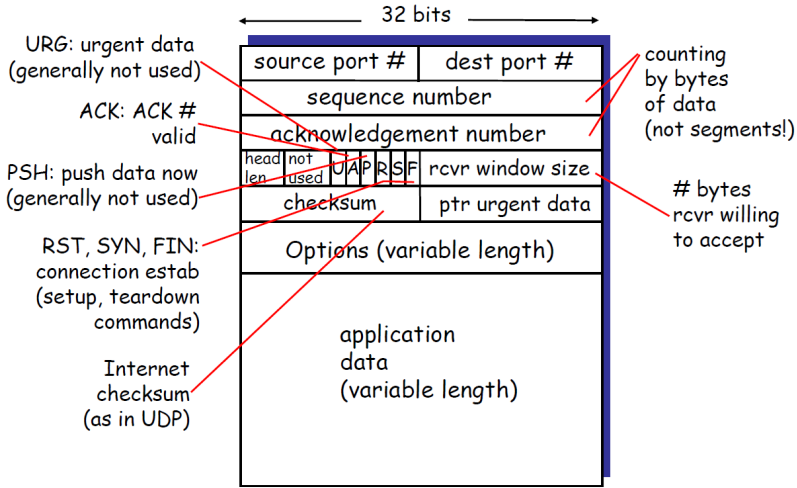
- No connection establishment (which can add delay)
- Simple: no connection state at sender, receiver
- Small segment header
- No congestion control: UDP can blast away as fast as desired

UDP: User Datagram Protocol

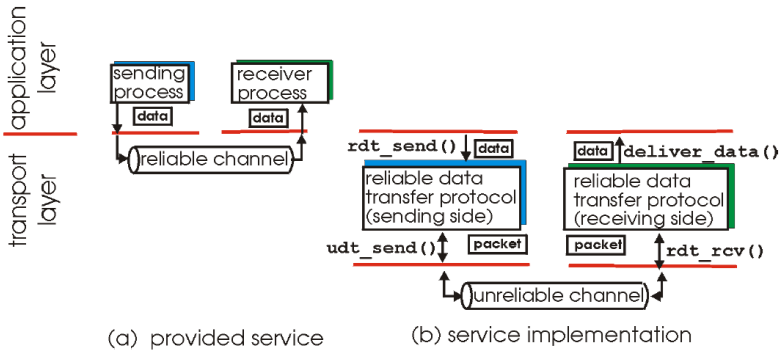
- Often used for streaming multimedia apps
  - loss tolerant
  - rate sensitive
- Reliable transfer over UDP: add reliability at application layer
  - application specific error recover!



TCP: Transport Control Protocol



Reliable Data Transfer

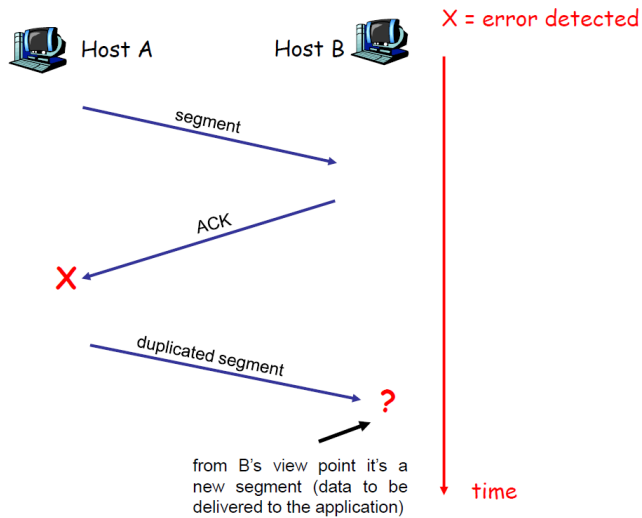


- Characteristics of unreliable channel will determine complexity of reliable data transfer protocol (rdt)
  - Errors, delay, packet loss

Bit Errors

- Underlying channel may flip bits in packet
  - recall: UDP checksum to detect bit errors
- The question: how does the sender know that an error occurs?
  - acknowledgements (ACKs): receiver explicitly tells sender that packet was received OK
  - negative acknowledgements (NAKs): receiver explicitly tells sender that packet had errors
  - sender retransmits packet on receipt of NAK
- **What happens if ACK/NAK corrupted:**
  - sender doesn't know what happened at receiver!
  - just retransmit: this might cause retransmission of correctly received packet!

ACK/NAK corrupted



ACK/NAK corrupted

- Handling duplicates:
  - sender adds sequence number to each packet
  - sender retransmits current packet if ACK/NAK garbled
  - receiver discards duplicate packet

Stop and Wait

Sender sends one packet, then waits for receiver response.

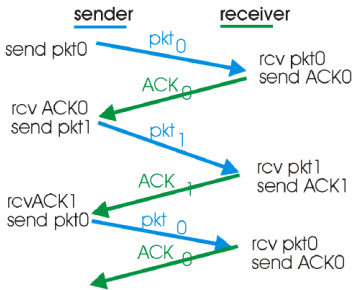
- **NAK-free:**
  - Instead of NAK, receiver sends ACK for last packet received OK
  - Receiver must explicitly include sequence number of the packet being ACKed

Channels with Errors and Loss

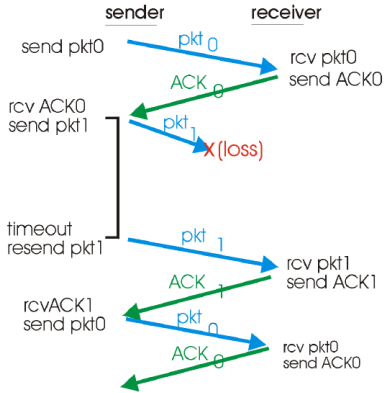
- New assumption:
  - Underlying channel can also lose packets (data or ACKs)
  - Checksum, sequence number, ACKs, retransmissions will be of help, but not enough
- **Q**: How to deal with loss?

- Approach:
  - Sender waits a “reasonable” amount of time for ACK
  - Retransmits if no ACK received in this time
  - If packet (or ACK) is just delayed (not lost):
    - Retransmission will be duplicate, but use of sequence numbers already handles this
    - Receiver must specify sequence number of packet being ACKed
  - Requires **countdown timer**

Reliable Data Transfer Protocol

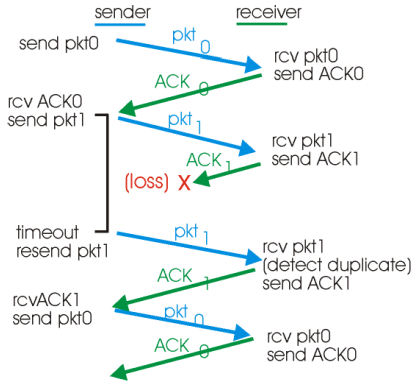


(a) operation with no loss

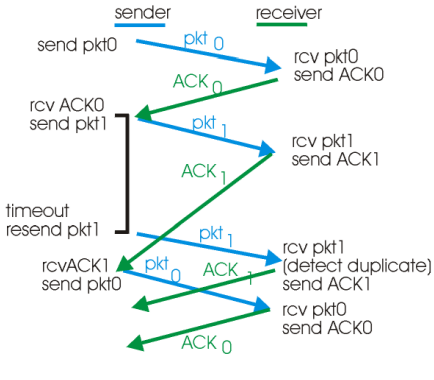


(b) lost packet

Reliable Data Transfer Protocol



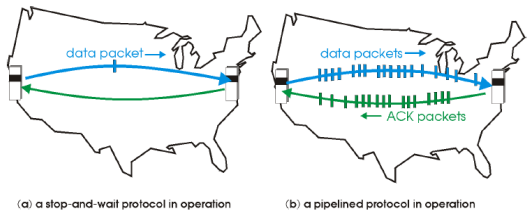
(c) lost ACK



(d) premature timeout

Pipelined Protocols

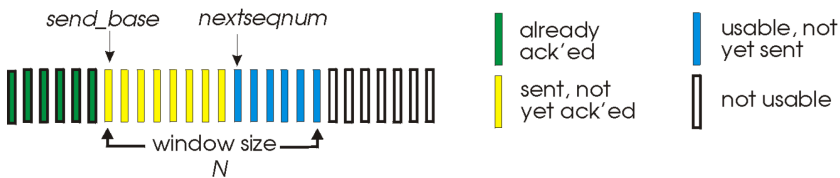
- **Pipelining:**
  - Sender allows multiple, ”in-flight”, yet to be acknowledged packets
  - Range of sequence numbers must be increased
  - Buffering at sender (Transmission Window) and/or receiver



- Two generic forms of pipelined protocols:
  - Go-Back-N: cumulative ACK, discard out-of-order packets
  - Selective Repeat: selective ACK, accept out-of-order packets → Rcv Buffer

GO-BACK-N

- **Sender:**
  - k bit seq # in pkt header
  - "window" of up to N consecutive unack'ed pkts allowed

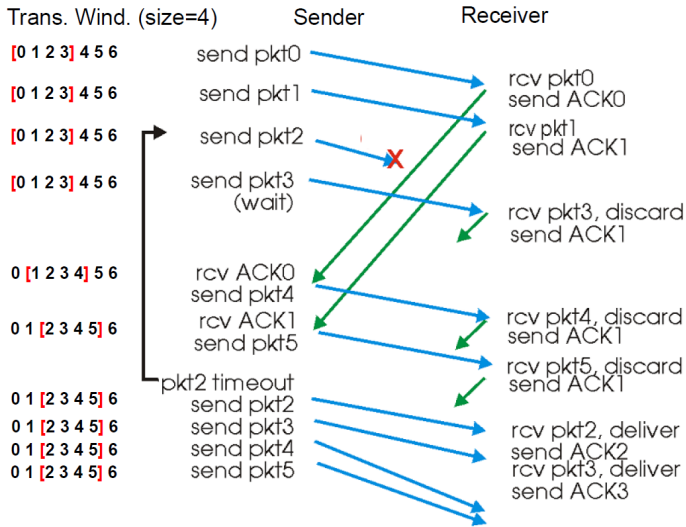


- ACK(n): ACKs all pkts up to, including seq # n. Cumulative ACK
- Timer for each in-flight pkt
  - Timeout(n): retransmit pkt n and all higher seq # pkts in window

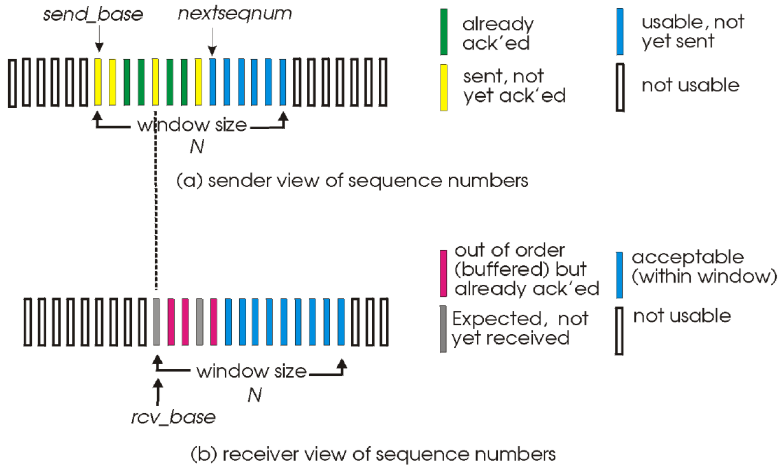
GO-BACK-N

- **The Receiver is simple:**
  - ACK only: always send ACK for correctly received pkt with highest in-order seq
    - may generate duplicate ACKs
    - need only remember `expectedseqnum`
  - Out-of-order pkt:
    - discard (don't buffer) $\rightarrow$ no receiver buffering
    - ACK pkt with highest in-order seq

GO-BACK-N


Trans. Wind. (size=4)
Sender
Receiver
[0 1 2 3] 4 5 6
send pkt0
rcv pkt0
send ACK0
[0 1 2 3] 4 5 6
send pkt1
rcv pkt1
send ACK1
[0 1 2 3] 4 5 6
send pkt2
[0 1 2 3] 4 5 6
send pkt3
(wait)
rcv pkt3, discard
send ACK1
0 [1 2 3 4] 5 6
rcv ACK0
send pkt4
0 1 [2 3 4 5] 6
rcv ACK1
send pkt5
rcv pkt4, discard
send ACK1
rcv pkt5, discard
send ACK1
pkt2 timeout
0 1 [2 3 4 5] 6
send pkt2
0 1 [2 3 4 5] 6
send pkt3
0 1 [2 3 4 5] 6
send pkt4
0 1 [2 3 4 5] 6
send pkt5
rcv pkt2, deliver
send ACK2
rcv pkt3, deliver
send ACK3

Selective Repeat



Selective Repeat

- Receiver individually acknowledges all correctly received pkts
  - Buffers pkts, as needed, for eventual in-order delivery to upper layer
- Sender only resends pkts for which ACK not received
  - Sender timer for each unACKed pkt
- Sender window
  - N consecutive seq #'s
  - Limits seq #'s of sent, unACKed pkts

Selective Repeat

Sender

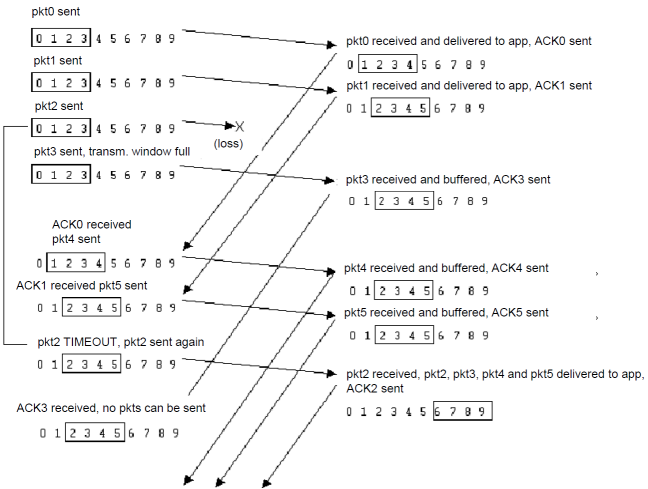
- Data from above:
  - If next available seq # in window, send pkt
- Timeout(n):
  - Resend pkt n, restart timer
- ACK(n) in `sendbase`, `sendbase+N`:
  - Mark pkt n as received
  - If n is the smallest unACKed pkt, advance window base to next unACKed seq #

Receiver

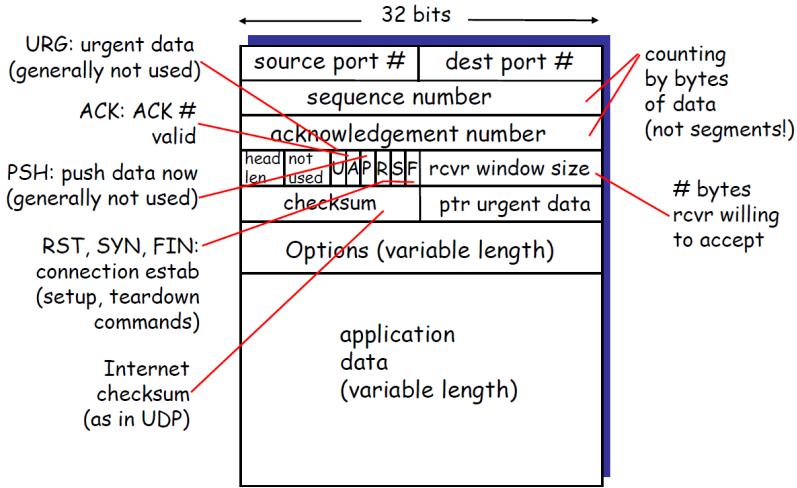
- pkt n in `[rcvbase, rcvbase+N-1]`:
  - Send ACK(n)
  - Out of order: buffer
  - In order: deliver (also deliver buffered, in-order pkts), advance window to next not yet received pkt
- pkt n in `[rcvbase-N, rcvbase-1]`:
  - ACK(n)
- Otherwise:
  - Ignore

Selective Repeat

5 The Transport Layer



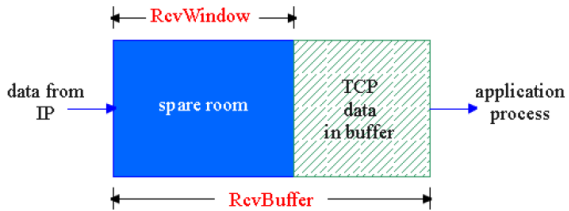
TCP: Transport Control Protocol



TCP Flow Control

`RcvBuffer` = size or TCP Receive Buffer

`RcvWindow` = amount of spare room in Buffer



- **Flow control:** sender won't overrun receiver's buffers by transmitting too much
- **Receiver:** Explicitly informs sender of dynamically changing amount of free buffer space. RcvWindow field in TCP segment
- **Sender:** Keeps the amount of transmitted, unACKed data less than most recent RcvWindow

Principles of Congestion Control

- **Congestion:**
  - Informally: “too many sources sending too much data too fast for network to handle”
  - Different from flow control!
  - Leads to:
    - Lost packets (buffer overflow at routers)
    - Long delays (queueing in router buffers)
  - Congestion Window (Congwin) to “limit” the transmission rate

TCP Congestion Control

- **Probing** for Usable Bandwidth
  - Ideally: transmit as fast as possible (Congwin as large as possible)
  - Increase Congwin until loss (congestion)
  - Loss: decrease Congwin, then begin probing (increasing) again

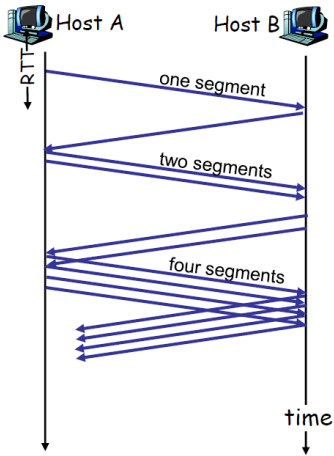
- Two “phases”
  - Slow start
  - Congestion avoidance
- Important variables:
  - Congwin
  - Threshold: defines threshold between the slow start phase and the congestion control phase

TCP Slowstart Algorithm

Slowstart Algorithm

```
 initialize:  Congwin = 1
for (each segment ACKed)
  Congwin = 2 * Congwin
until (loss event OR
  CongWin > threshold)
```

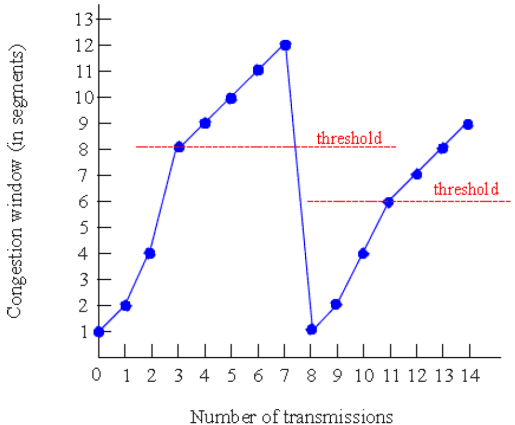
- exponential increase (per RTT) in window size (not so slow!)
- loss event: timeout (Tahoe TCP) and/or or three duplicate ACKs (Reno TCP)



TCP Congestion Avoidance

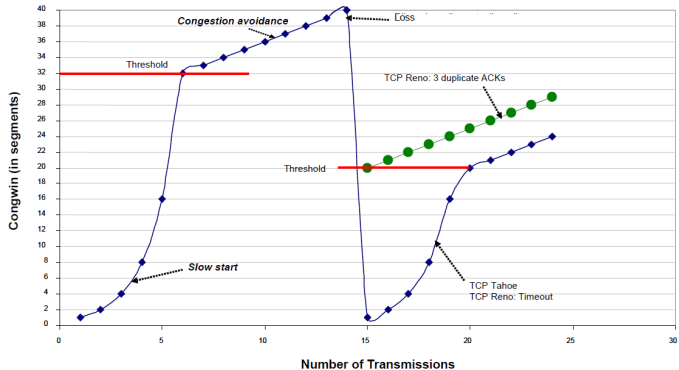
Congestion Avoidance

```
 /* slowstart is over */
/* Congwin > threshold /*
Until (loss event)
  {Congwin++}
threshold = Congwin / 2
Congwin = 1
perform slowstart
```



TCP Tahoe and TCP Reno

- **TCP Tahoe:**
  - Loss event → back to slow start
- **TCP Reno:**
  - Loss event revealed by 3 Duplicate ACKs → window to threshold value and increase by 1



Flow Control and Congestion Control

- Flow control procedure computes RcvWindow
- Congestion control procedure computes Congwin

Transmission Window

Transmission window = $\min(\text{RcvWindow}, \text{Congwin})$

TCP Connection Management

- TCP sender and receiver establish a ”connection” before exchanging data segments.
- Initialize TCP variables:
  - Sequence numbers
  - Buffers, flow control info (e.g., RcvWindow)

Three-Way Handshake:

1. **Step 1:** Client end system sends TCP SYN control segment to server.
   - Specifies initial sequence number.
2. **Step 2:** Server end system receives SYN and replies with SYN-ACK control segment.
   - Acknowledges received SYN.
   - Allocates buffers.
   - Specifies server to receiver initial sequence number.
3. **Step 3:** Client replies with ACK.

TCP: Transport Control Protocol

5 The Transport Layer