

# CS536 DATA COMMUNICATION AND COMPUTER NETWORKS

© Xiyuan Chen

Spring Term, 2024

This work is licensed under a [Creative Commons “Attribution-NonCommercial-ShareAlike 4.0 International”](#) license.



## Information

- No slides, no textbook.
- No curving for the final grade.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Components of a computer network . . . . .	2
1.2	Communication . . . . .	2
1.3	Types of networks . . . . .	3
1.3.1	Point-to-point link . . . . .	3
1.3.2	Multi-access link . . . . .	4
1.3.3	Internetwork . . . . .	4
1.4	LAN vs WAN . . . . .	5
1.5	Port . . . . .	5
1.6	Network Design & Performance . . . . .	5
1.6.1	Emphasis of lightweight network core . . . . .	5
1.6.2	Performance yardsticks . . . . .	6
<b>2</b>	<b>Fundamentals of Data Transmission</b>	<b>7</b>
2.1	How are bits sent using physical signals? . . . . .	7
2.2	Frequency and Bandwidth . . . . .	8
2.3	Optical Fiber . . . . .	8
2.3.1	Compare to copper wire . . . . .	8
2.4	Sending more bits than the frequency bandwidth . . . . .	8
2.5	Multi-user communication . . . . .	8
2.5.1	Time-division multiplexing accessing (TDMA) . . . . .	9
2.5.2	Code-division multiplexing accessing (CDMA) . . . . .	10
2.5.3	Frequency-division multiplexing accessing (FDMA) . . . . .	11

# 1 Introduction

The lecture notes of this section is [lec1](#) and [lec2](#).

## 1.1 Components of a computer network

1. host devices (PC, server, laptop)
2. routers & switches (IP router, Ethernet switches, WiFi routers)
  - The global internet has 2 types of routing going on: 1. **Intranet**: routing within a domain. 2. **Internet**: routing between domains.
3. links (wires, fiber, quantum)
  - Confidentiality. We need to encrypt the data.
  - Authentication. We need to make sure the data is from the right source.
  - Integrity. We need to make sure the data is not modified.
  - Bounded by the speed of light. For wireless or wired today, we use eletromagnetic waves. FDM (Frequency Division Multiplexing) → OFDM (Orthogonal Frequency Division Multiplexing)
4. protocols (IP, TCP, UDP, ...). All protocols are part of an OS (in kernel mode). Protocol helps connect different parts of the network. Examples from low- to high-layer:
  - NIC: Network Interface Card. Such as Ethernet card, WALN card, etc. It is read only memory code. **Lower half of the OS.**
  - Device Drivers. **Lower half of the OS.**
  - ARP, RARP. **OS.**
  - IP. **OS.**
  - OSPF, RIP, BGP. OFPF, RIP: within organizations (intra-domain). BGP: global Internet (inter-domain). **OS.**
  - TCP, UDP. **OS.**
  - DNS, HTTP, SMTP, SNMP, SSL. **Application.**
  - SSH, web browser, PHP, P2P, YouTube etc. **Application.**
5. applications (DNS, HTTP, SMTP, SSL, ...)
6. humans and bots (spam, DoS, worm, ...)

1, 2 and 3 are **hardware**, 4 and 5 are **software**.

## 1.2 Communication

- Types of information transmtion: **analog** and **digital**.
- In today's networks, the content is digital (bits), but the transmission is analog (eletromagnetic waves) → use analog information to transmit digital information.

Capability of network and end systems:

1. information aabstraction:
  - digital content representatoin: encode/decode information.

- analog representation and transmission of digital content: analog signals over physical media.
2. information protection:
    - deal with information corruption (bits flip). Use BER (Bit Error Rate) to measure the quality of the link.
    - deal with information loss(packet drop at routers and hosts). e.g., culprit: buffer overflow.
    - security. e.g., confidentiality, authentication, integrity, protect from infrastructure attacks such as DoS.
  3. performance:
    - fast transmission: throughputs (bps), bottleneck can be software.  
 Why 1Gbps Ethernet is not 1Gbps throughputs? TCP is not one-time transmission. TCP is a **reliable** protocol, which means it will make sure the data is delivered. TCP will send a packet, wait for the ACK, and then send the next packet.
    - information latency: physical distance, buffering of messages at routers and hosts. Bad for real-time applications such as video streaming, online gaming, etc.

### 1.3 Types of networks

1. connectivity:
  - point-to-point
  - multi-access (broadcast)
  - internetwork (network of networks such as Purdue's campus network)
2. medium:
  - wired
  - wireless
3. location:
  - stationary
  - mobile
    - Doppler effect: the frequency of the wave changes when the source or the receiver is moving.
    - Low frequency loses strength but passes through walls. (2.4GHz, 5GHz)

#### 1.3.1 Point-to-point link



Figure 1: Point-to-point link

- NIC at A, NIC at B
- A and B don't need names in principle.

### 1.3.2 Multi-access link

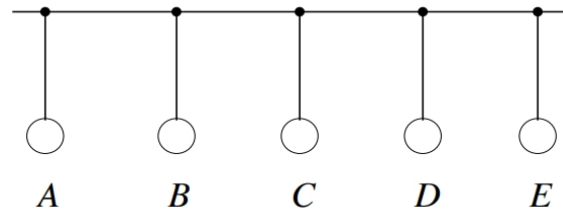


Figure 2: Multi-access link

- sometimes called bus
- names (i.e., addressing) necessary, called local area network (LAN) addresses.
- key issue of multi-access link communication: access control.
  - link is a shared resource.
  - myriad of LAN technologies and protocols.
- denial of service (DoS) attack: receivers need to process a minimum amount of data and analyze metadata to determine if a message is for them. When splashing a lot of messages, the receiver will be overwhelmed.
- Signals may add up to create **superposition**, which makes receivers hard to decode the message.

### 1.3.3 Internetwork

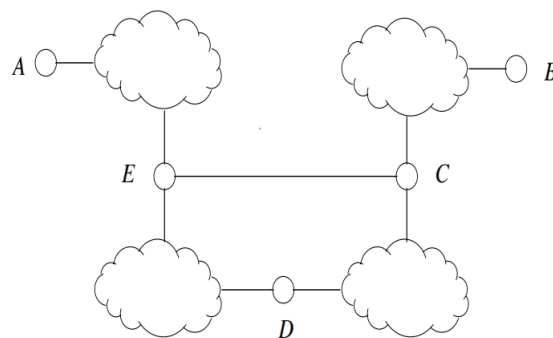


Figure 3: Internetwork

- recursive definition: network of networks.
- everything is the composition of point-to-point links and multi-access links.
- additional complications:
  - new names beyond LAN addresses: in principle, LAN addresses are unique and suffice. In practice, new names (i.e., network addresses) bring benefits despite overhead. IPv4: 32 bits, IPv6: 128 bits.
  - protocol translation: LANs speak different languages (e.g., Ethernet and WLAN). We call hosts that are able to do more than one network connection and IP address **multi-homed**.

- location management: mobility. e.g., handoff of mobile host among multiple networks.

## 1.4 LAN vs WAN

- LAN (Local Area Network): point-to-point links and multi-access links.
- WAN Wide Area Network: internetwork. **WAN is a network of LANs.**

Setellite, GPS, Starlink, printer to PC are all point-to-point links, thus they are all LANs. Communication between east coast and Europe is through cables under the ocean, thus it is still a LAN. (if we ignore repeaters during the transmission here.)

When we talk about conenctions within Purdue campus which involves hundreds of routers, switches and devices, it is WAN.

Takeaway: LAN and WAN are not about the size of the network, but the type of the network. If the connection is a large network of LANs, it is WAN. And when connecting 2 LANs, we need some intermediary who speaks both languages of the 2 lines being connected.

## 1.5 Port

When we communicate between devices, each device has an OS. When we try to send data, we are not sending it to OS (in most cases, except pinging), but to a **process** (or **thread**). How do we know which process to send to?

Attempt: use process ID. But process ID is **not predictable**. We cannot assign PID manually to a new process.

Solution: use (IP address, **port number**) pair. Port is a number that is predictable. We can assign a port number to a process. When we send data, we send it to a port number. The OS will know which process to send to.

## 1.6 Network Design & Performance

### 1.6.1 Emphasis of lightweight network core

The priciple nowadays is to **keep the hardware simple** and **put the complexity in software**. We push heavyweight stuff toward the edge.

ATM: Asynchronous Transfer Mode. It is a circuit-switched network. It is a smart router designed in the 80s. It handles some of the jobs for packets. But IP ended up winning the competition.

The network is kept very small, we call it an **end-to-end paradigm**.

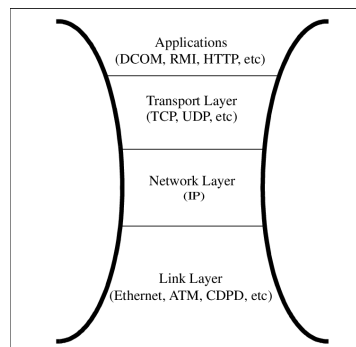


Figure 4: Hour-glass Design of Network

### 1.6.2 Performace yardsticks

- **Bandwidth** in bps (bits per second). We **ignore** protocol overhead in this metric.
- **Throughputs** in bps. Protocol overhead is included. In practice, app and user space OS overhead causes further slow-down.
- **Latency** in msec (milliseconds).
  - signal speed limit: speed of light.
  - processing and buffering delay: time to process a packet, time to wait in a queue.
- **Jitter** in msec: variation in latency. **Average delay small but max delay large**. Jitter is bad for real-time applications.

A single bit cannot go faster (bound by reality world and physics). The only way we can increase the speed is to **increase the number of bits** we send at a time (increase the **bandwidth**. i.e., bits packed into 1 second).

Internet traffic is **bursty**. e.g., real-time video streaming.

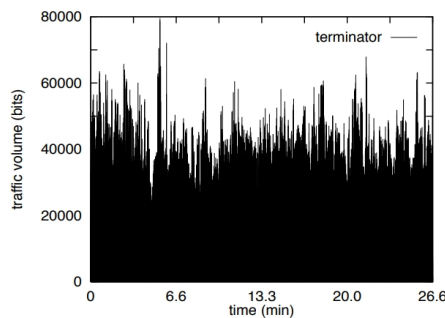


Figure 5: Bursty Internet Traffic Diagram

## 2 Fundamentals of Data Transmission

### 2.1 How are bits sent using physical signals?

Lecture notes of this section is [here](#).

E.g. A and B are connected by point-to-point link. A wants to send bits 011001 to B.

We use electromagnetic(EM) waves because it is fast.

Sine curves with different strengths are used to represent bits. The strength of the EM wave is called **amplitude**. The number of cycles per second is called **frequency**. **Wavelength** is the distance between 2 peaks of the wave. ( $\frac{1}{f} \times c = L$ , where  $c$  is the speed of light.)

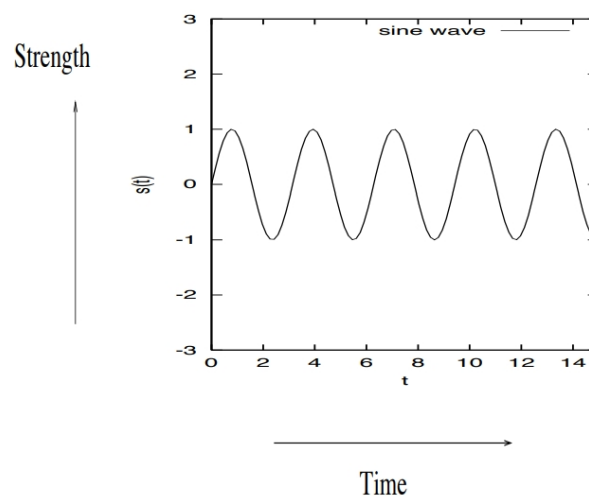
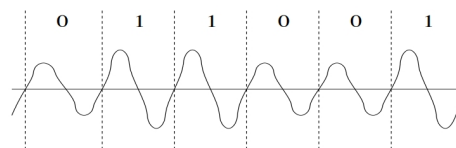


Figure 6: EM Signal

**Amplitude Modulation:** a process by which the wave signal is transmitted by modulating the amplitude of the signal. We use **high amplitude to represent 1** and **low amplitude to represent 0**. (low amplitude is **not** 0 amplitude, it is just lower than high amplitude.)



→ large amplitude: 1

→ small amplitude: 0

Figure 7: Amplitude Modulation

In AM Radio, the signal is not in bits, but an analog signal. The signal is modulated by the amplitude of the EM wave. The receiver will demodulate the signal to get the original signal.

## 2.2 Frequency and Bandwidth

We measure the frequency in **Hertz**. 1Hz = 1 cycle per second. 1KHz = 1000Hz. 1MHz = 1000KHz. 1GHz = 1000MHz. This can be easily converted to the bandwidth (or bps).

If we have a signal with frequency 1Hz then the bandwidth is 1bps. If we have a signal with frequency 1KHz then the bandwidth is 1000bps. If we have a signal with frequency 1MHz then the bandwidth is 1Mbps. If we have a signal with frequency 1GHz then the bandwidth is 1Gbps.

So can we just increase the frequency to increase the bandwidth? No!

1. increasing frequency requires increase clock rate and processing speed, which is costly.
2. wireless propagation. To reach above 10 GHz requires **line of sight** (LOS). The signal cannot go through walls.
3. multi-user communication. The real world is not all point-to-point links. This is the key reason why we cannot just increase the frequency to increase the bandwidth.

## 2.3 Optical Fiber

When the light signal hits the surface of glass **within the critical angle**, it will be **reflected completely** back into the glass.

If the angle is not right, some light escapes and we lose some strength of the signal. The receiver will see a distorted signal. This is called **modal dispersion**.

### 2.3.1 Compare to copper wire

1. copper wires are easily influenced by the environment. e.g., the signal will be distorted by the magnetic field while optical fiber is not.
2. When two signals meet in copper wire, they will interfere with each other. But in optical fiber, they only add up (superposition). Of course the receivers need to be able to decode that merged signal.

## 2.4 Sending more bits than the frequency bandwidth

Question: How can we send more bits than the bandwidth given a fixed frequency? e.g., we have a 1Hz signal, but we want to send 2 bits per second.

Solution: Make the amplitude to be 4 levels instead of 2 levels thus we can send 2 bits at once per second.

Similarly, we can use 8 levels to represent 3 bits, 16 levels to represent 4 bits, etc.

Notice: when we increase the number of levels, the amplitude levels will have **less noise tolerance**.

## 2.5 Multi-user communication

### Interference



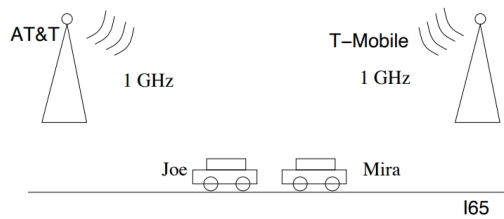


Figure 8: Multi-user Interference

Both cell towers are sending signals. Each receiver will receive the signal from both towers, particularly, the sum of two signals (superposition).

Problems:

- distorted signal
- interference
- hard to figure out what bits were sent

## Multiplexing

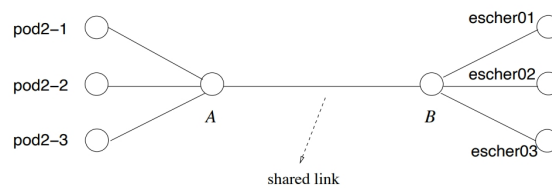


Figure 9: Multi-user Multiplexing

Point-to-point links between routers/switches (A and B) which forward multiple traffic streams.

### 2.5.1 Time-division multiplexing accessing (TDMA)

Bit streams sharing a same link. Subdivide time into equal sized slots and assign each slot to some users.

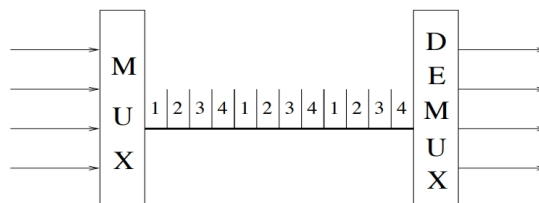


Figure 10: Time-division Multiplexing

Each user can get multiple slots and each slot can be assigned to multiple users. (Round robin)

In the multiplexing example, router A acts as a multiplexer (MUX) and router B acts as a demultiplexer (DEMUX).

In real world, TDMA is in **frames** to support multiple (specifically, 24 in the example) simultaneous users (called "channels").

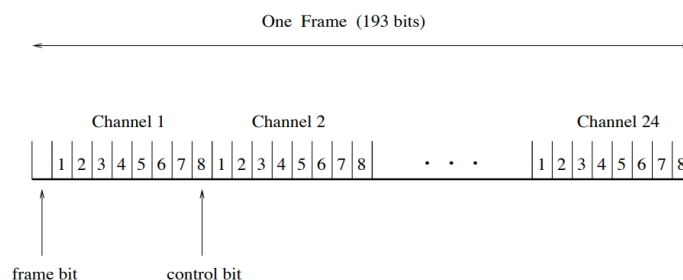


Figure 11: T1 carrier (1.544 Mbps)

- 8-bit block: Each user sends 8 consecutive bits.
- In total  $24 \times 8 = 192$  bits of payload. Plus 1 control bit, we have 193 bits per frame.
- 8000 frames are squeezed into 1 sec time interval. So we have  $193 \times 8000 = 1.544$  Mbps.

Problem: OK if frequency is managed by a central authority. Bad when frequency is shared by multiple providers  $\rightarrow$  interference.

### 2.5.2 Code-division multiplexing accessing (CDMA)

Sending multiple bit streams using coding method. Each user is assigned a unique vector.

#### 1. Cell tower:

- Create  $n$  **orthogonal** vectors (codes). Each user is assigned a unique vector.
- Map bits to vectors. (See below)
- Broadcast the synthesized vector.

#### 2. Receiver:

- Receive the synthesized vector.
- Multiply the signal with the assigned vector. (Notice that orthogonal vectors get cancels out.)
- If the result scaler is positive, then the user is sending 1. If the result scaler is negative, then the user is sending 0.

In step 1(b), how do we craft a vector such that after multiplication, positive means 1 and negative means 0? We do this by **applying linear combinations of user vectors using 1s and -1s**.  $\mathbf{z} = a_1\mathbf{x}_1 + a_2\mathbf{x}_2 + \dots + a_n\mathbf{x}_n$ .

e.g.,

$$\begin{bmatrix} 1 & -1 & 1 & 1 & -1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix}$$

where  $x_i$  is the vector assigned to user  $i$ . We can see that if user 1 sends 1, then the result after multiplied by the user vector again will be positive, vice versa. (For user  $i$ , the result will be  $\mathbf{z} \odot x_i = a_i(x_i \odot x_i)$ .)

### 2.5.3 Frequency-division multiplexing accessing (FDMA)

Given the frequency  $f$ , the sine wave can be described as  $\sin ft$ .

When we change amplitude over time to carry bits (AM), it is no longer a pure sine wave, but a **signal**  $s(t)$ .

For **single-user** scenario, that user gets all frequency channel.

For **multi-user** scenario, each user gets one frequency channel and apply FDMA.

In optical fiber, people use term "Wave-length Division Multiplexing" (WDM) instead of FDM. But they basically mean the same thing.

Other applications:

- **Frequency Hopping:** change frequency over time to carry bits. (e.g. avoid enemy monitoring).
- **Anti-jamming:** spreading bits over multiple frequencies makes jamming harder.
- **Frequency-selective fading:** naturally, signals get distorted in the real world. But that distortion is greater or smaller depending upon the frequency that we're choosing. By spreading the signal over multiple frequencies, we can reduce the distortion, also known as **spread spectrum** (↓ see below).

**Spread spectrum** bits are spread over multiple frequencies.

We use frequencies with width from  $f_1$  to  $f_n$ , this range is called **bandwidth**. e.g.,  $n = 10$ ,  $f_1 = 1$  GHz,  $f_n = 1.9$  GHz, then the bandwidth is  $1.9 - 1 = 0.9$  GHz.