

Historical Introduction

Early History

- Runners
 - Egypt, 2000 BC
 - Marathon-Athens, 490 BC
- Calling posts
 - Persia, 400 BC
 - Germany, 1796
- Mirrors
 - Greece, 400 BC
 - Arizona, US Army, 1886

Long Distance Communication Methods

Method	First Recorded Use	Last Recorded Use
Pigeons	Egypt 2900 BC	California 1981 AD
Runners/Carriers	Egypt 1928 BC	Pony Express 1860 AD
Beacons/Torches	Troy 1184 BC	England 1588 AD
Calling Posts	Persia 400 BC	Germany 1796 AD
Heliographs	Greece 400 BC	Arizona 1886 AD
Flags	Greece 400 BC	Maritime Use Today

Aeschylus' Line of Beacons

Location	Modern Name	Altitude (m)	Distance (km)
Troy	Troy	100	0
Mt Ida	Kaz Dagi	1774	55
Lemnos	Skopia at Lemnos	420	154
Mt Athos	Athos	2033	70
Macistus	Kastillion (Euboea)	1209	177
Messapius	Ktipas	1020	30
Cithaeron	Elatias	1410	25
Mt Aegiplanetus	Mt Jeraneia	1370	30
Aragnean Hgt	Arna	1199	50
Mycenae	Mycenae	150	20

More Early Examples

- Horses
 - Romans, 100 AD, 80 km/day
 - Mongols, 1100 AD, 160 km/day
 - Pony Express, 1860, 320 km/day
- Pigeons
 - Egypt, 3000 BC
 - England, WWI, 380 pigeoneers, 20,000 pigeons
 - California, 1981, sending microfilm 40 km between plant and test site

Still More Examples and Methods

- Fire Beacons
 - Greece, the fall of Troy (1180 BC)
 - North American Indians, 19th century
- Flags/Semaphores
 - Greece, 400 BC
 - Naval and general maritime use today
- Light Flashing
 - Babbage, 1851
 - Naval and general maritime use today

Other Methods: Pigeons

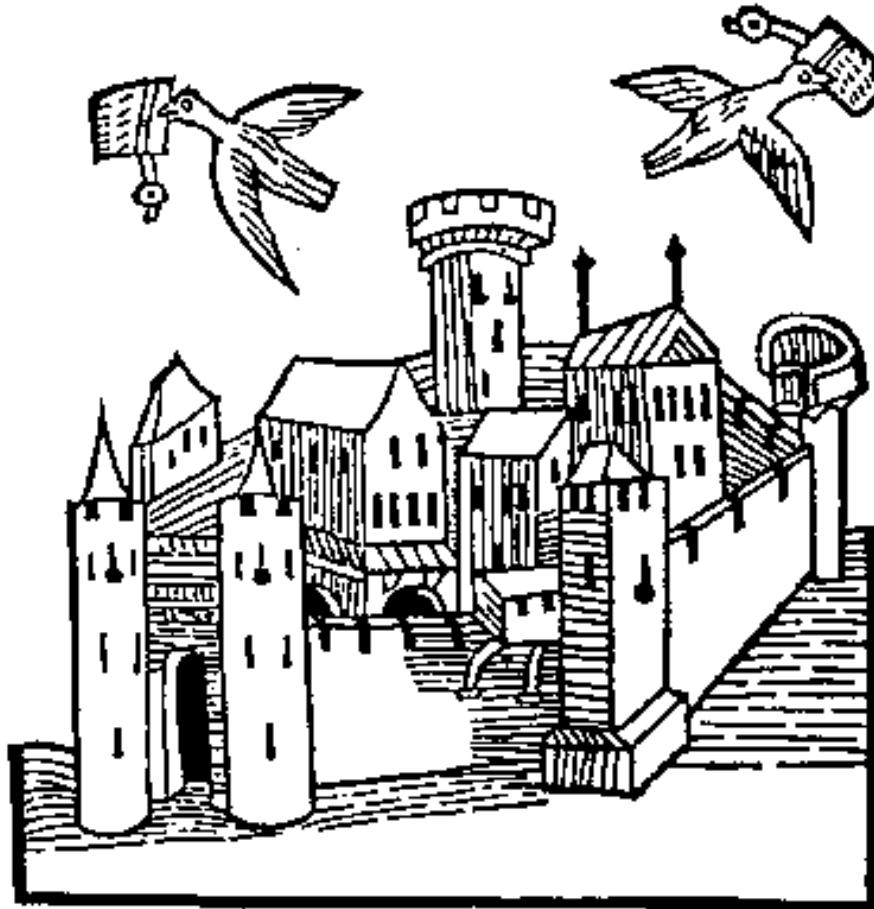


Figure 1.1 Pigeon Post, Woodcut from A.D. 1481.
(Coll. Bibl. de Genève, [Fabre 1963], p. 44)

Other Methods: Fire Beacons



Other Methods: Sound

- Sound (e.g., church bells)



Other Methods: Light

- Heliographs



Other Methods: Smoke

- Used by American Natives and elsewhere

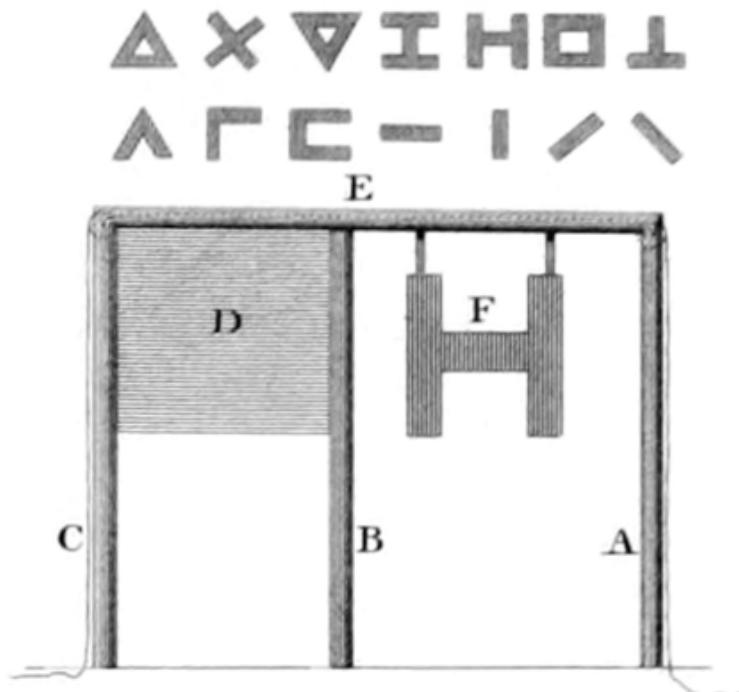


Other Methods: Magic

- There was a kind of mystique about message transmission.
 - Magic needles (people believed that if two needles were touched by same lodestone would always move parallel to each other!)
 - The ruthless French prime minister Richelieu was believed to have a set of needles because he was always so well-informed!

Hooke's Semaphore

- Various symbols that might be used are indicated at the top;

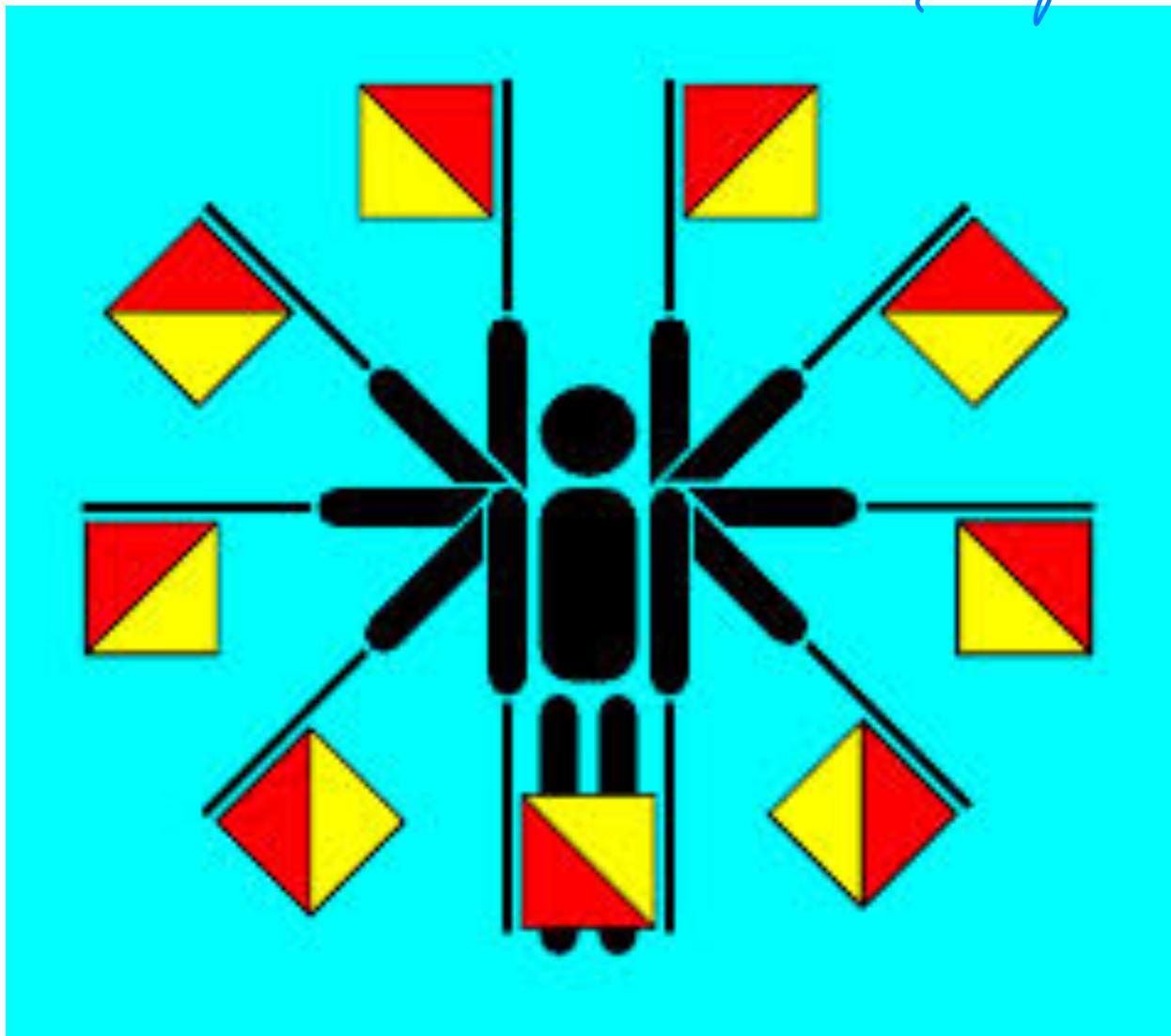


- ABCE indicates the frame.
- D indicates the screen behind which each of the symbols are hidden when not in use.

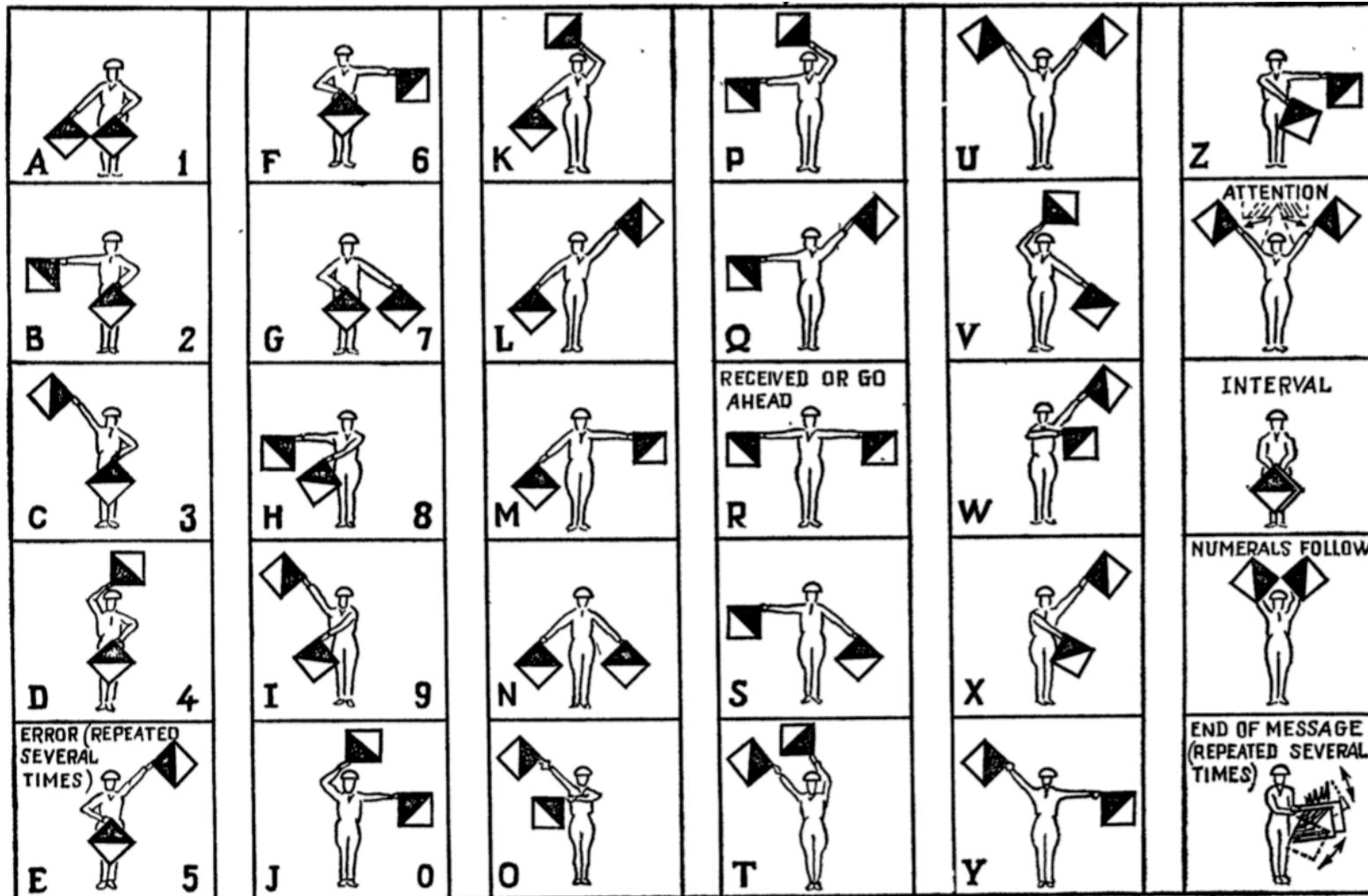
Greek

Semaphores

Flag Carriers

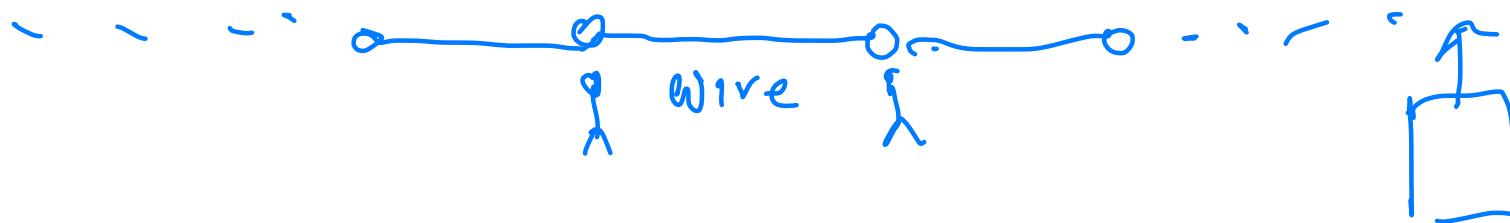


Semaphores: Alphabet



Electrical Networks

- They wanted to know if electricity can be used for transmission.
 - In April 1746 about 200 monks of the Carthusian Convent in Paris were arranged in a snake-like line
 - Each monk held one end of a 25 feet long iron wire.
 - The abbe (and noted scientist) Jean-Antoine Noillet connected a primitive electrical battery at one end and gave them all a powerful electrical shock!
- The experiment showed that electricity could be transported.



Electrical Networks



Electrical Networks

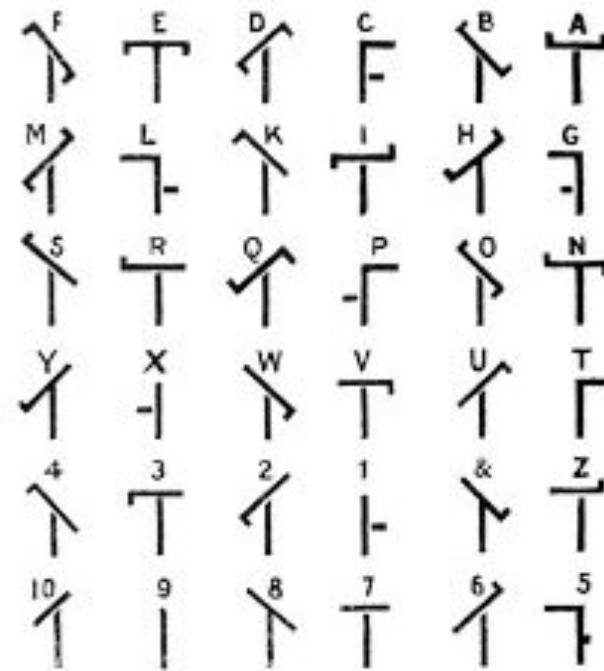
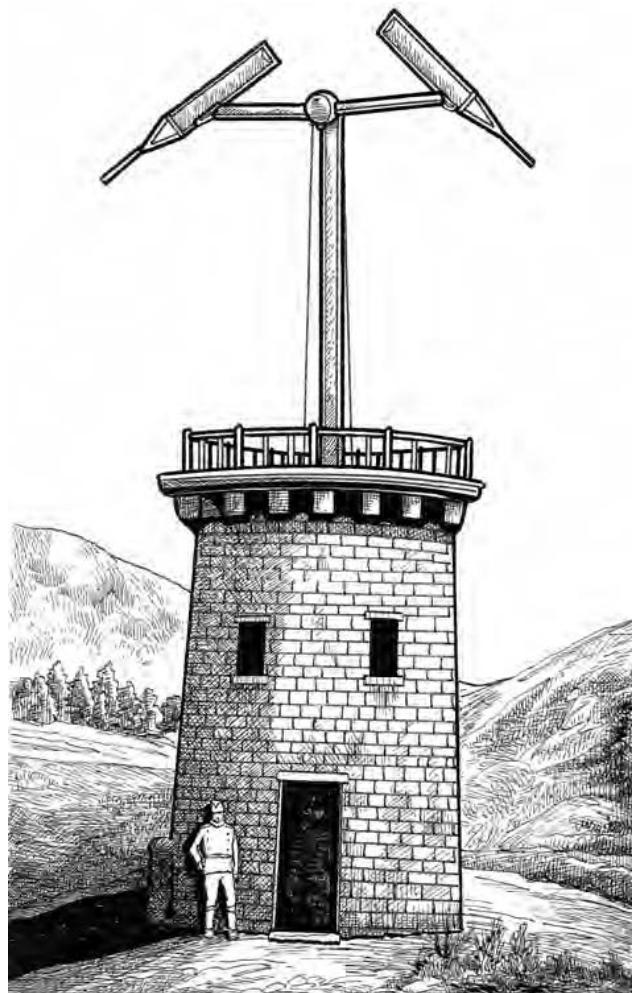


Important Pre-Telegraph Examples: France

- France - Claude Chappe, 1791
 - Semaphore system
 - Largest reach: Paris to Milan (720 Km)
 - 2 ~~symbols/min~~; 92 symbols; about .5 bits/sec



Chappe Network



Important Pre-Telegraph Examples: Sweden

- Sweden - Abraham Edelcrantz, 1794
 - 10 shutters, open or closed, 1024 symbols
 - In 1801, connected to Danish network to form first internet
 - 1805, 50 stations, 720 people

Edelcrantz Network



Telegraph and Telephone

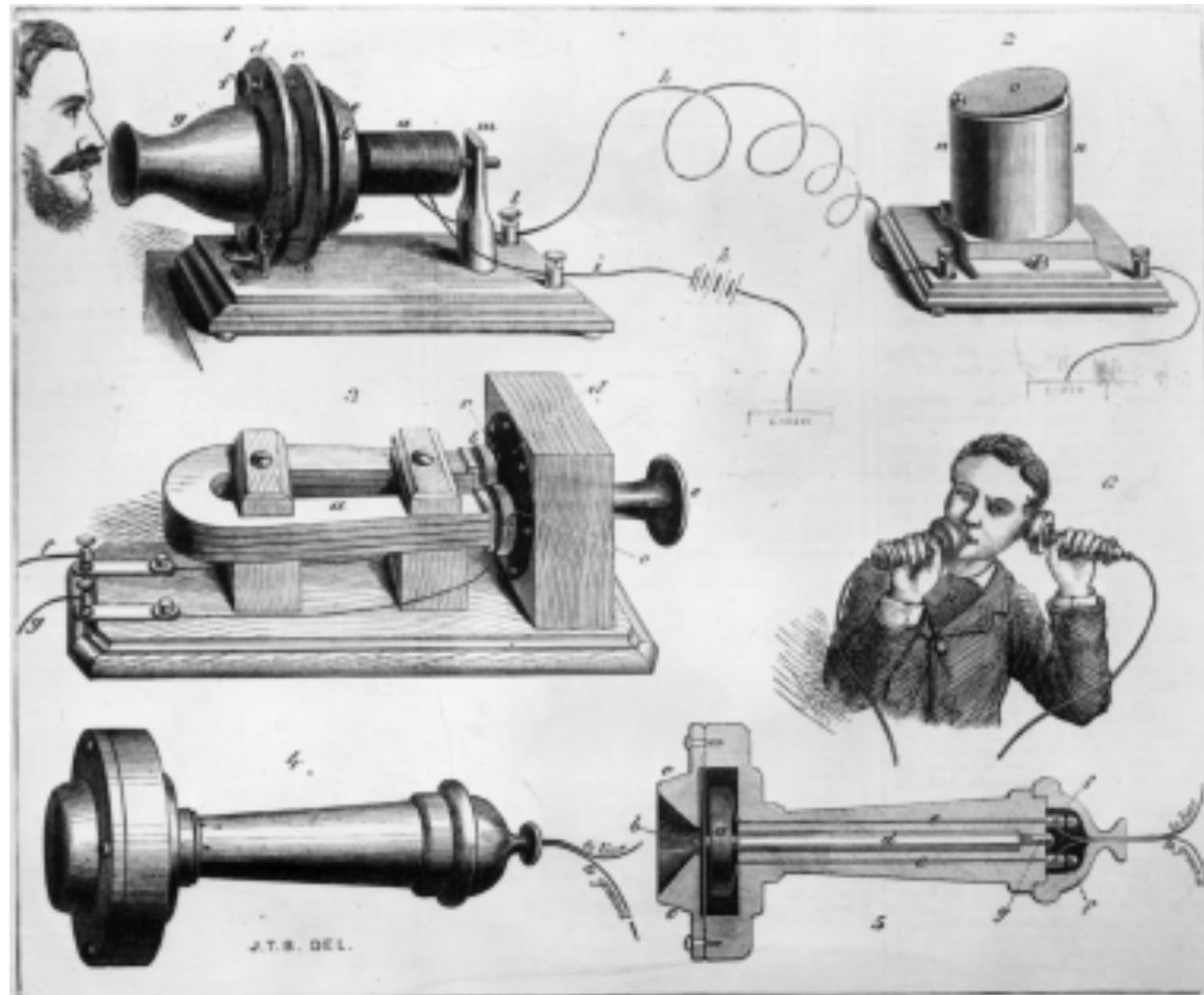
- Telegraph (“Far Writer”) developed in early 1800’s
- Most successful version due to Samuel Morse
- Led to developments in codes, eventually information theory
- Reached speeds of 1-2 characters/sec or about 10 bits/sec
- Telephone developed in 1876 by Alexander Graham Bell

Morse Telegraph

**Morse Telegraph Key
(circa 1844)**

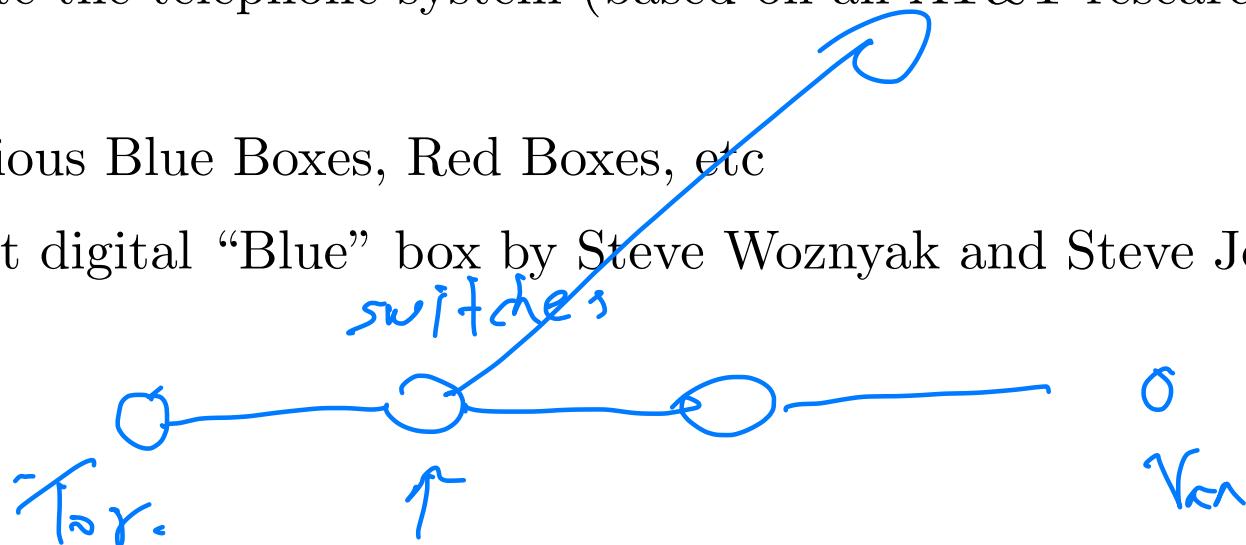


Alexander Graham Bell Telephone



Telephone Phreaks and Hackers

- Phone phreaks in the 60s scanning for interesting toll-free numbers.
 - Calling the White House (Direct line to President Nixon)
 - Calling CIA and NORAD security lines
 - Doing free teleconferences and phone calls
- Development of boxes taking advantage of 2,600 Hz lines to infiltrate the telephone system (based on an AT&T research article)
 - Various Blue Boxes, Red Boxes, etc
 - First digital “Blue” box by Steve Woznyak and Steve Jobs.



Article on Technical Specs



- A. Weaver, N. A. Newell, In-Band Single-Frequency Signaling, Nov. 1954, Bell System Technical Journal.

Blue Boxes on Sale



Invention	Year
Runners	-2000
Calling Posts	-400
Mirrors	-400
Horses	100
Pigeons	-3000
Fire Beacons	-1200
Semaphores	-400
Electrical Networks	1746
Semaphore Networks	1791
Morse Code	1832
Telephone	1876

Internet

- ARPANET (US Army, Advanced Research Project Agency)
- ETHERNET (Xerox PARC, Palo Alto. Developed by Metcalfe)
- NSFNET (Network for the US National Science Foundation)

ARPANET	1969
ETHERNET	1982
NSFNET	1987
MOSAIC	1991

Exercises^a

1. Would the internet revolution have happened without the need to satisfy military applications?
2. What are the key factors that influenced the development of the internet?
3. Give a heuristic discussion to estimate the speed of transmission through
 - (a) fire beacons.
 - (b) heliographs
4. Where are semaphores being used today?

^aDo not submit!

Bibliography

- J. Abbate, “Inventing the Internet”, MIT Press, 1999.
- G. J. Holzmann and B. Pehrson, “The Early History of Data Networks”, IEEE Computer Society, 1995.
- S. Segaller, “Nerds: A Brief History of the Internet”, TV Books, New York, 1998.
- T. Standage, “The Victorian Internet”, Walker and Company, New York, 1998.
- P. Lapsley, Exploding the Phone: The Untold Story of the Teenagers and Outlaws who Hacked Ma Bell. Grove Press, 2013.

lecture before class lec -
" after " cla - notes

Speed/Performance

of connection

Many aspects

- algos
- protocols

Frustrating!

- Very disappointing!
 - Why am I getting only a few % of the advertised 3G (or 4G, LTE, 5G) speed?
- It is advertised that the 3G downlink speed for stationary users should be 7.2 Mbps.
- When you try to download an email attachment of 3 MB, it often takes as long as one and half minutes.
 - 3 MB equals 24 Mbits; therefore downloading the attachment shouldn't take longer than 4 sec!
- You get around 267 kbps, 3.7% of what you might expect.
- Who took away the 96%?

Things Are Confusing! (1/2)

- Specs are decisions within Standardization bodies, which may not be universally accepted.
- The terms 3G, 4G, and 5G can be confusing.
 - One track following the standardization body 3GPP (3rd Generation Partnership Project) called UMTS (Universal Mobile Telecommunications System) or WCDMA
 - Another track in 3GPP2 called CDMA2000 (Code Division Multiple Access).
- Each also has several versions in between 2G and 3G, often called 2.5G, such as EVDO (Evolution-Data Optimized), EDGE, etc.

Things Are Confusing! (2/2)

- For 4G, the main track is called Long Term Evolution (LTE), with variants such as LTE light and LTE advanced.
- Another competing track is called WiMAX.
- Some refer to evolved versions of 3G (e.g., HSPA+) (HSPA = High Speed Packet Access) as 4G too.
- 5G is the term for new mobile technologies. Definitions differ and confusion is common. The ITU IMT-2020 (ITU = International Telecommunication Union, IMT = International Mobile Telecommunications) standard provides for speeds up to 20 gigabits per second and has only been demonstrated with millimeter waves of 15 gigahertz and higher frequency.
- All these have created quite a bit of confusion in a consumer's mind as to what really is a 3G (or even 4G and 5G) technology.

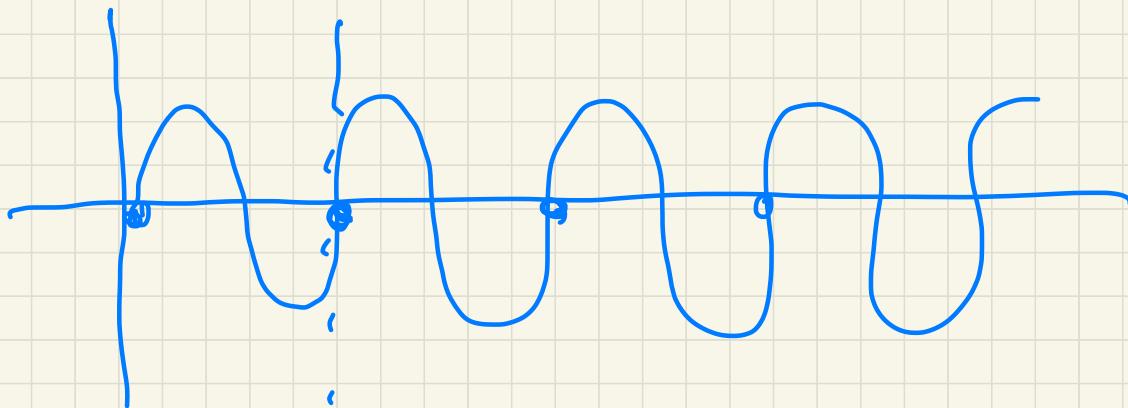
Trends

- Many countries have moved toward LTE.
- They use a range of techniques to increase the spectral efficiency, defined as the number of bits per second that each Hz of bandwidth can support.
- These include methods like OFDM (Orthogonal Frequency Division Multiplexing) and MIMO (Multiple-Input and Multiple-Output), and Cell Splitting (splitting a large cell into smaller ones).
- But the “user observed throughput” in 4G, while much higher than that for 3G, still falls short of the advertised numbers we often hear in the neighborhood of 300 Mbps.
- Why is that? Will 5G make a difference?

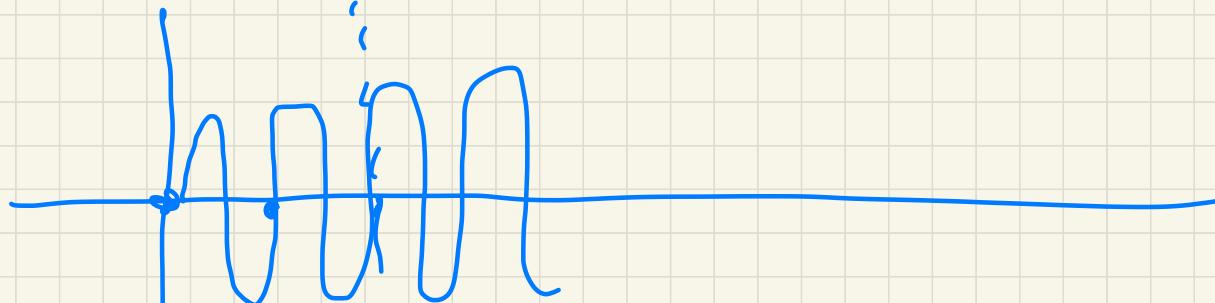
Hz = a measure
of frequency

$x \leftarrow$
what is $x = ?$

faster



even
faster



slow

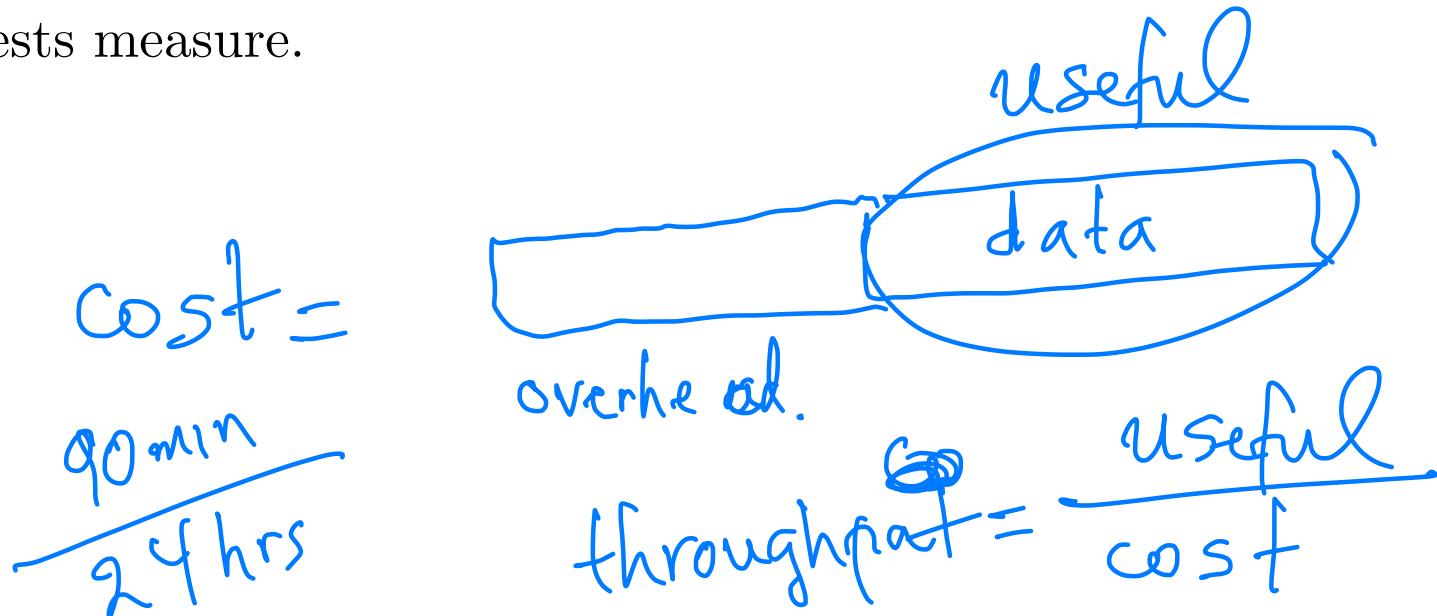


Two Main Reasons

- There are two main reasons:
 - non-ideal network conditions
 - overheads.
- Many parts of the wireless network exhibit non-ideal conditions, including both the air-interface and the backhaul network.
- Furthermore, networks, just like our lives, are dominated by overheads, such as the overhead of network management in the form of control bits in packets or control sequences in protocols.

Speed Reduction

- Useful throughput is defined as the number of bits of actual application data received, divided by the time it takes to get the data through.
 - It is some form of average.
- This is what you “feel” you are getting in your service, but might not be what advertisements talk about or what speed tests measure.

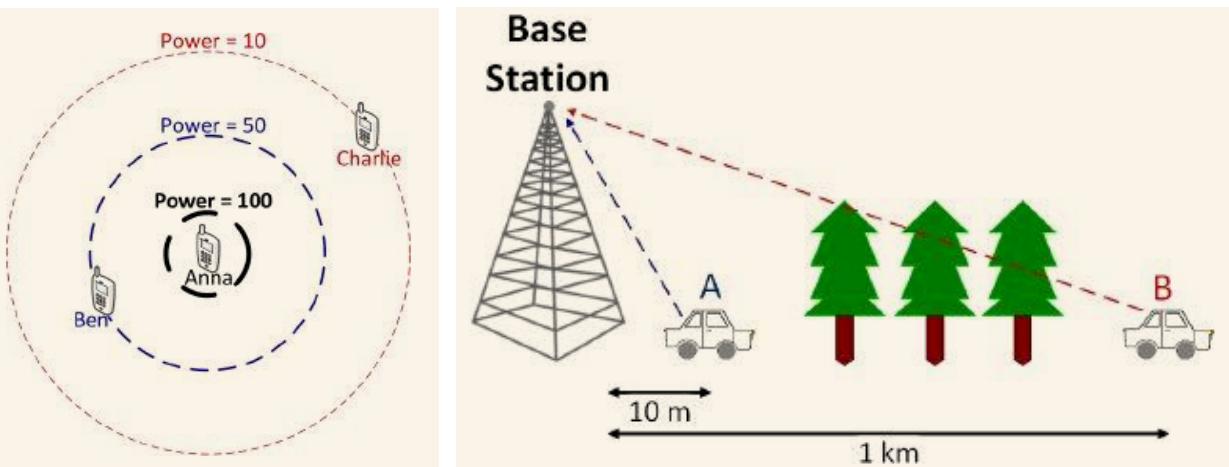


The Culprits

- Air-interface
- Backbone (or Backhaul) network
- Network (Communication) Protocols
- Simple Network Management Protocol
- Mathematical Modeling, Simulations, and Testing

Air-interface: Physical Characteristics

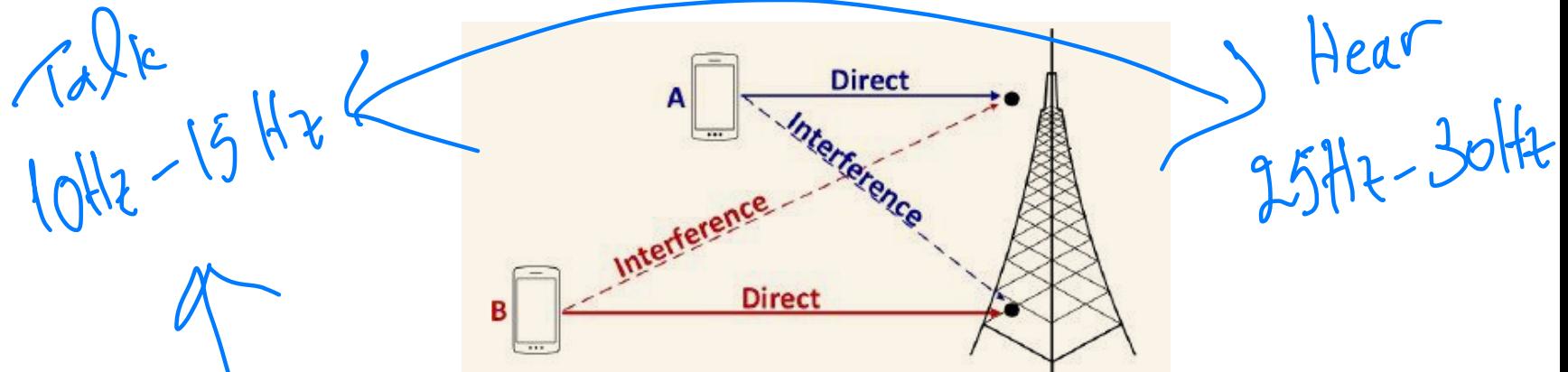
- Propagation channel degradation:
 - path loss (the signal strength drops as the distance of propagation increases),



- shadowing (obstruction by objects), and
- multipath fading (each signal bounces off of many objects and is collected at the receiver from multiple paths)

Air-interface: Interference

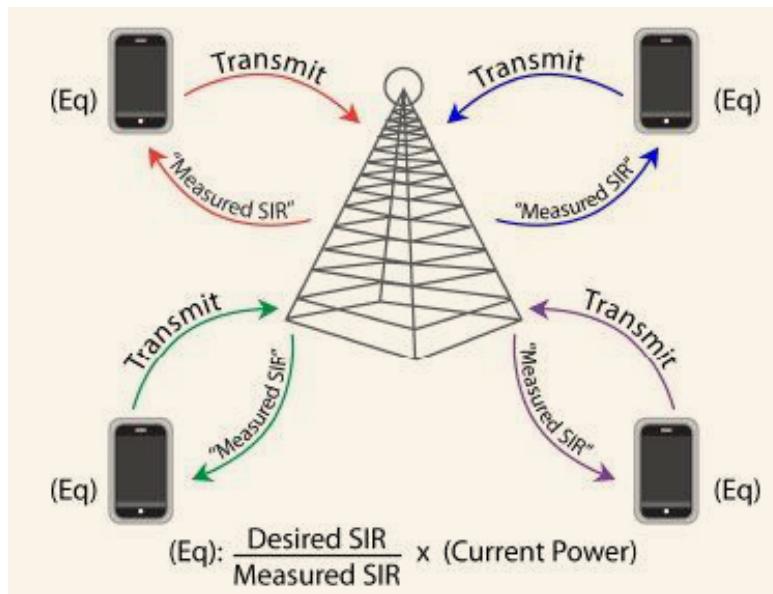
- There are many users, and they interfere with each other.



- Received SIR (Signal to Interference Ratio) might be so low that modulation needs to be toned down and transmission rate reduced so that the receiver can accurately decode.
- Typical instance of the problem in CDMA networks is the near far problem: Even power control cannot completely resolve this problem.

Backbone: Multiple Links

- There can be more than ten links traversed from the base station to the actual destination on the other side of a wireless session of, say, YouTube streaming.

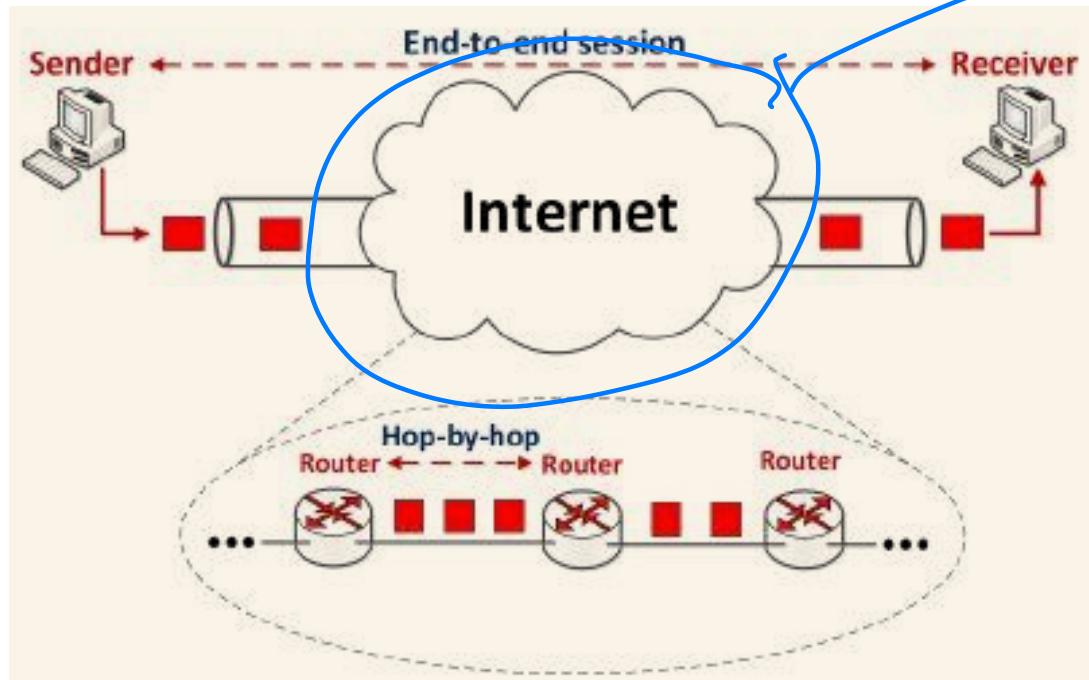


Queueing
Methodology

- Links: Users' traffic competes with the traffic of other users on the links behind the air-interface in the cellular network.

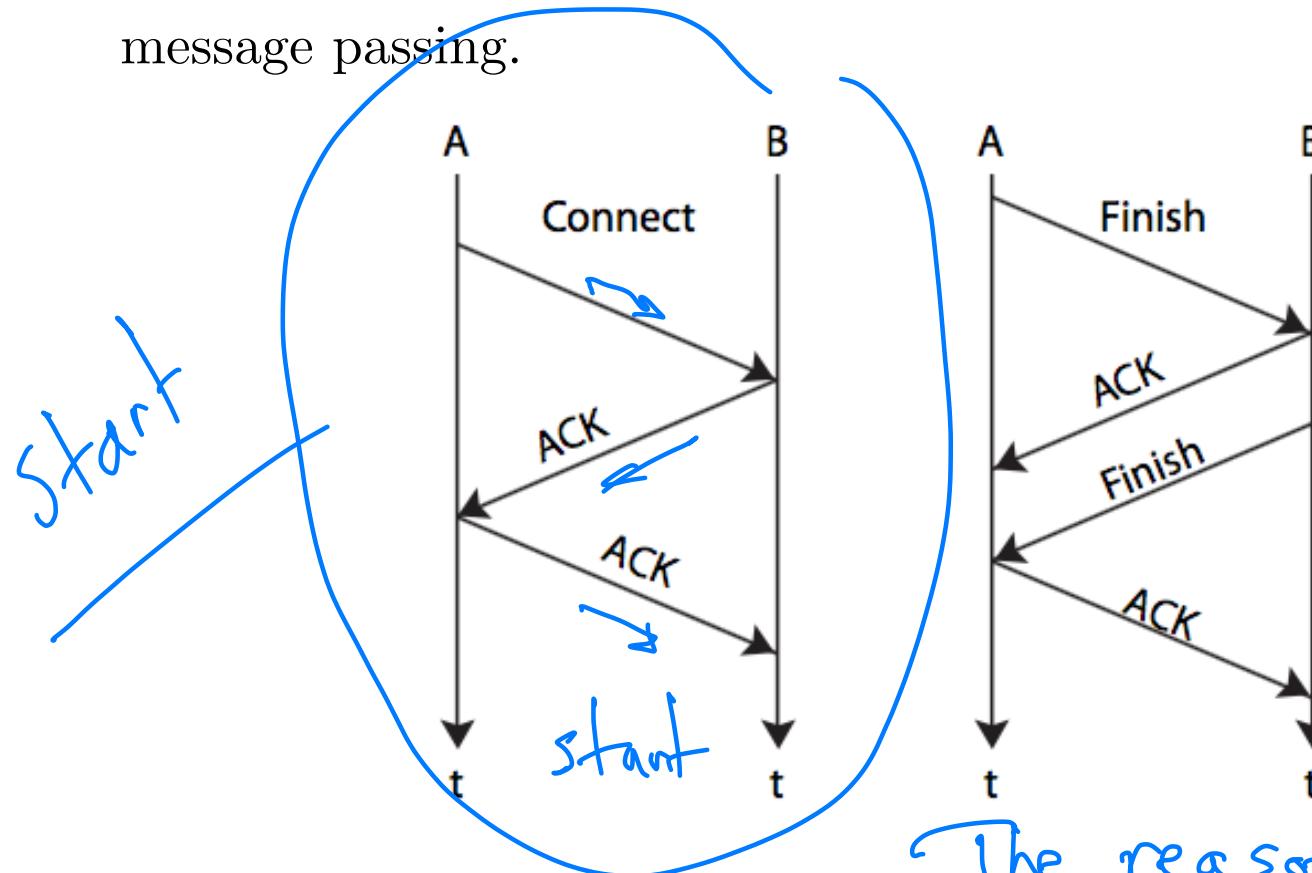
Backbone: Multiple Links

- Nodes: These links are connected through nodes of various kinds: gateways, switches, routers, servers, etc.



Protocols: Semantics

- Protocol semantics: Many functionalities require sequences of message passing.

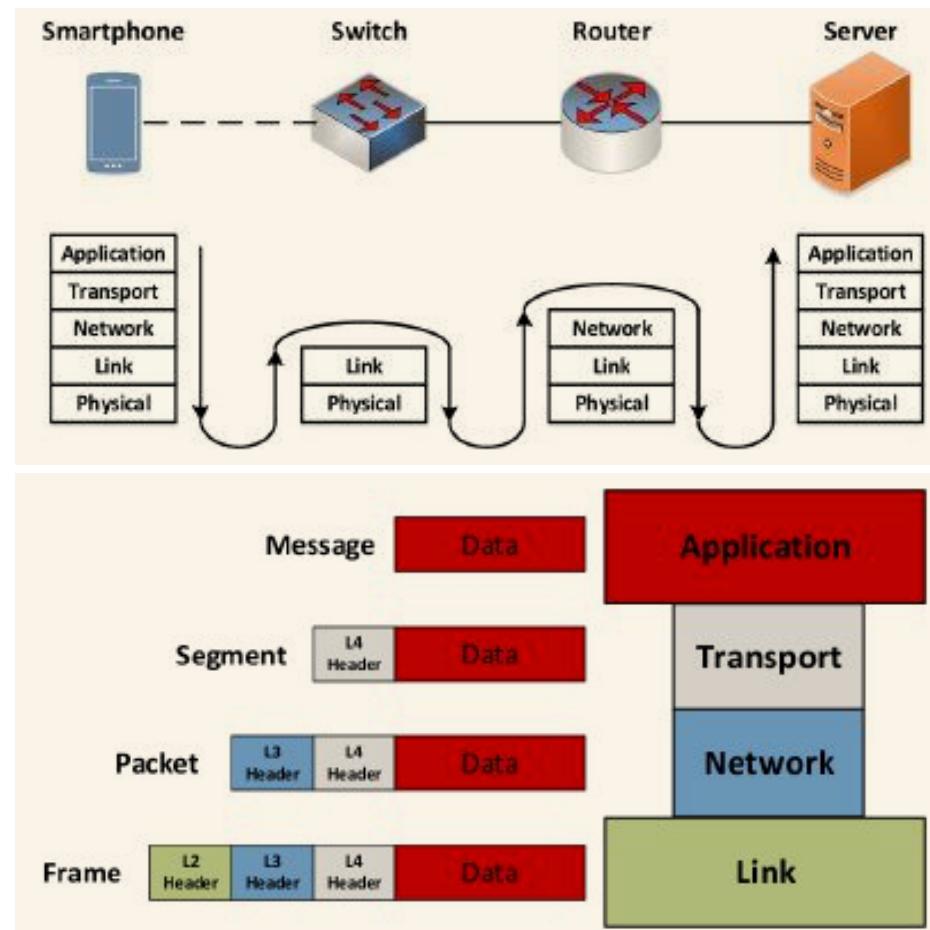


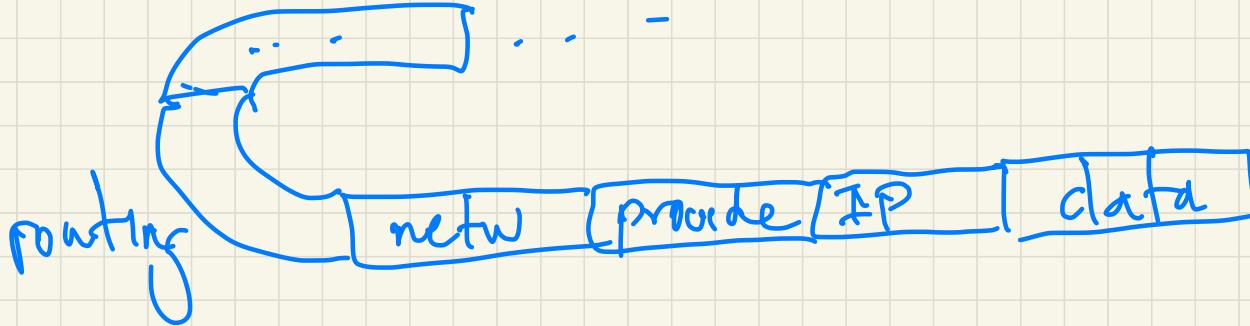
- Are the protocols adequate?

The reason things work is protocols

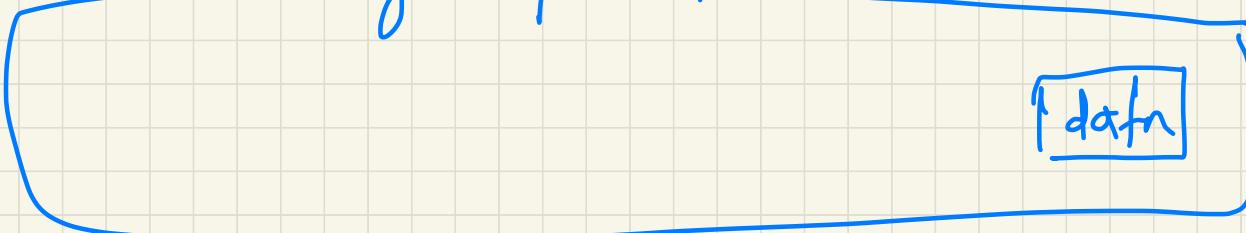
Protocols: Packetization

- Each layer adds a header to carry control information: address, protocol version number, quality of service, error check, etc.



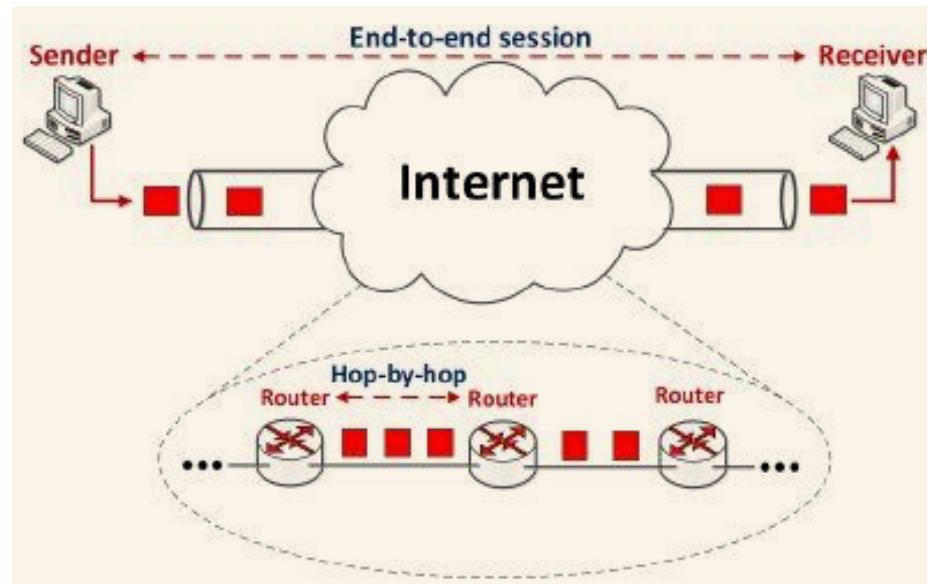


larger packet



Protocols: Control Plane Signaling

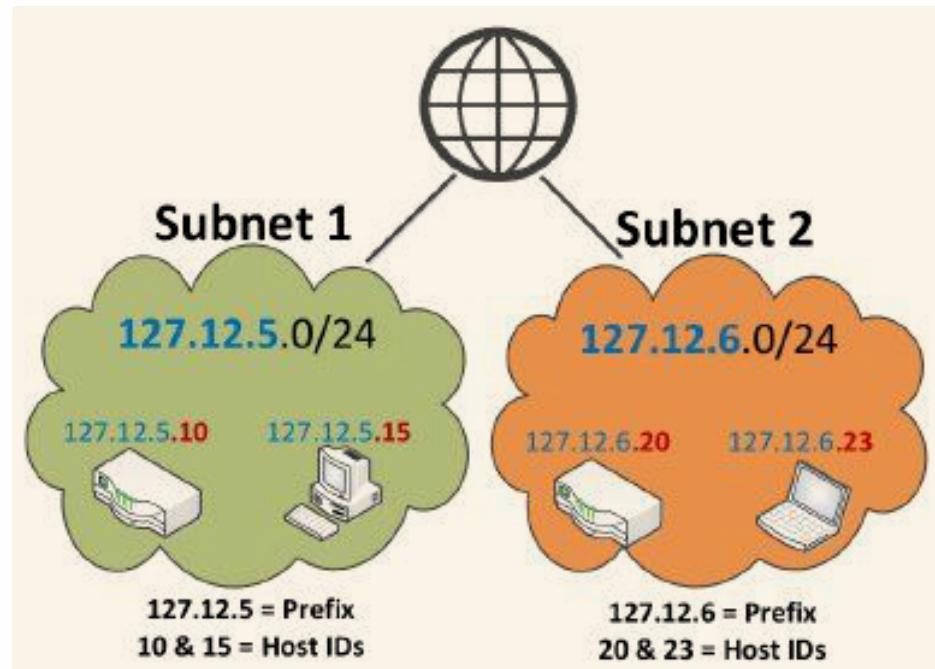
- Actual data traffic flows on data channels (a logical concept, rather than physical channels), while control signals travel on control channels. Control signals may have to travel half of the world even when the source and destination nodes are right next to each other.



- E.g., the actual traffic of people and cargo is carried by airplanes flying between airports following particular routes.

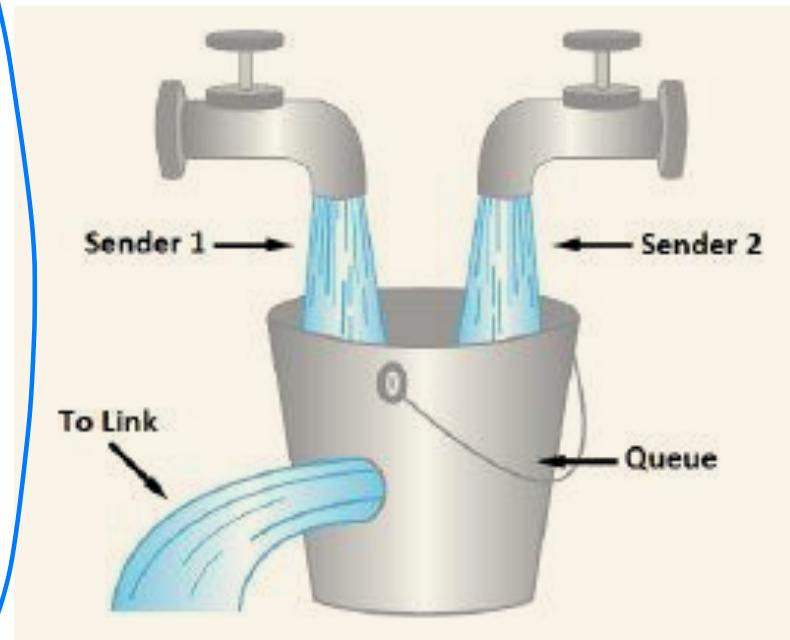
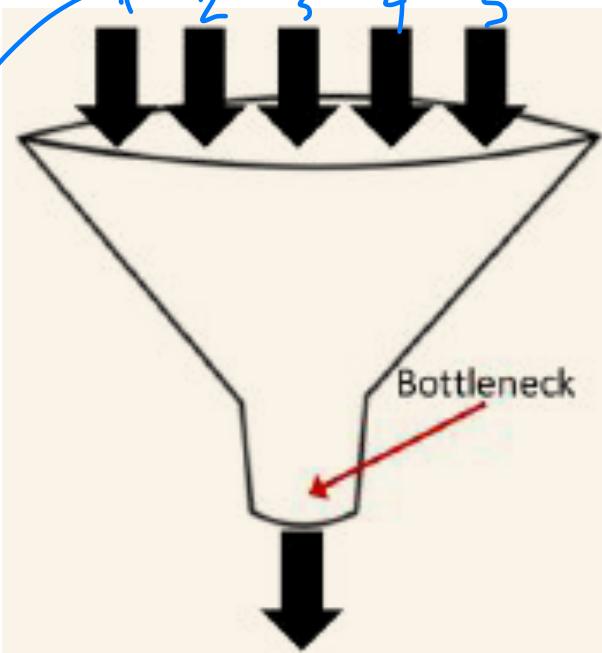
Protocols: Addressing

- Data cannot be routed and will be lost unless proper addressing schemes are used.



Bottlenecks

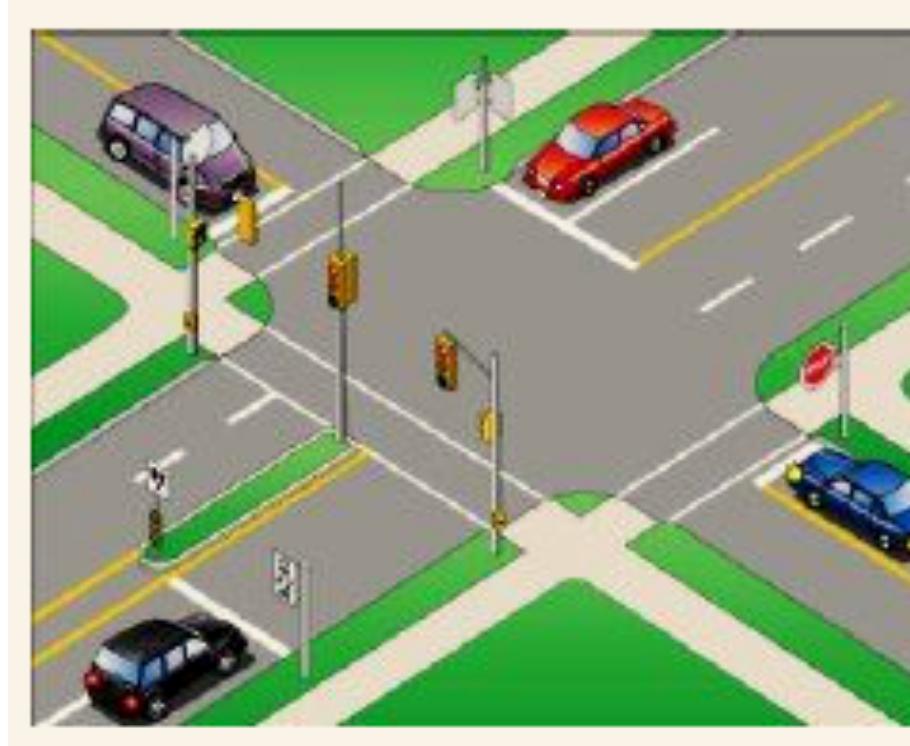
- All these issues create bottlenecks.



- Unless flow is managed bottlenecks can form.

Paradigms Help!

- Traffic scenarios tend to offer great analogies to communication networks.



Simple Network Management Protocol (SNMP)

- Performance:
Monitor, collect, and analyze performance metrics.
- Configuration:
of the control knobs in different protocols.
- Charging:
Charge each network usage in time-dependent pricing.
- Fault-management:
Monitor to see whether any link or node is down, and then contain, repair, and root-cause diagnose the fault.
- Security:
authentication, integrity, and confidentiality

Speed? What Speed?

- Improving Technologies in 802.11

Standard	Year	Frequency (GHz)	Maximum Speed (Mbps)
-	1997	2.4	2
b	1999	2.4	11
a	1999	5	54
g	2003	2.4	54
n	2009	2.4 & 5	100

What is the Speed of my Network?

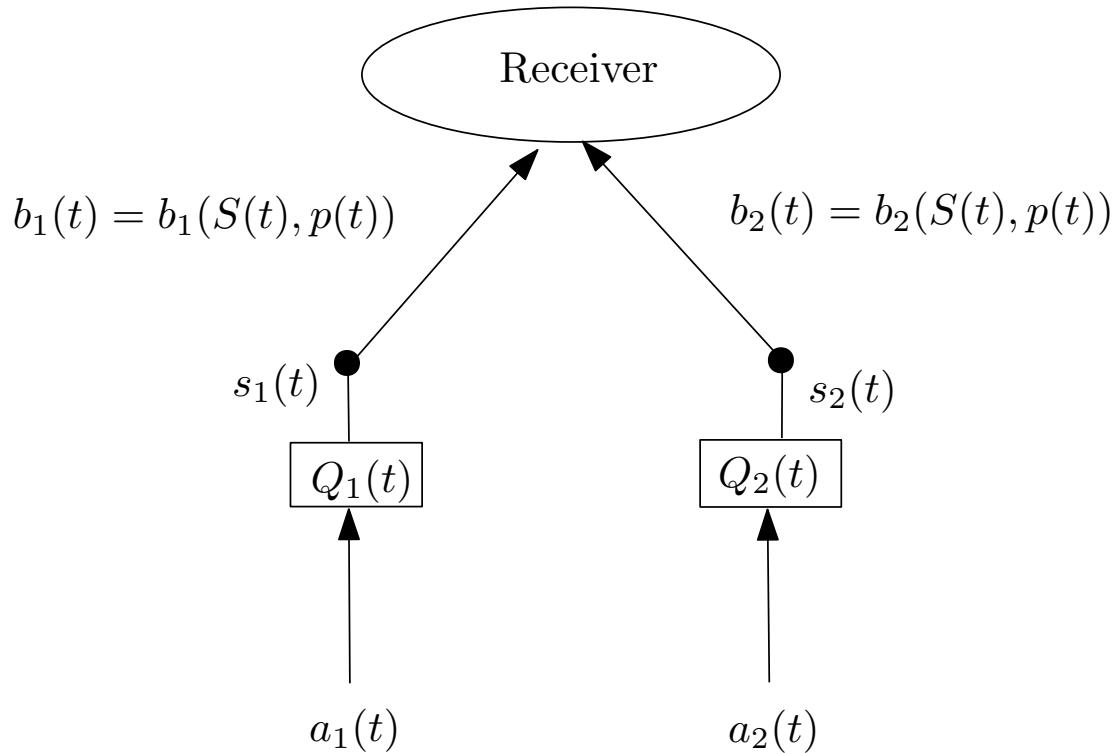
- The speed of your wireless (or wireline) Internet connection is not one number, but many numbers depending on the answers to the following four questions,
 - Which layer? MAC layer: Medium access control, RLC layer: Radio link control, and PDCP layer: Packet data convergence protocol.
 - Where is it measured? Depending on the locations of the two end points.
 - When is it measured? Traffic intensity during different hours of the day.
 - What application? Different traffic runs different sets of protocols and user utility and expectation.

Mathematics, Simulations, and Testing

- Mathematics plays an important role.
 - There is a need for adequate mathematical models that address the problems.
 - There is a plethora of such models, but not all models are the same.
- Simulations are needed to confirm the mathematical analysis.
- Testing of the models required in real world environments.

Mathematical modelling and analysis (1/2)

- Consider a 2-user wireless uplink that operates in slotted time $t \in \{0, 1, 2, \dots\}$.



- Every time slot new data randomly arrives to each user for transmission to a common receiver.

Mathematical modelling and analysis (2/2)

- $(a_1(t), a_2(t))$ is the vector of new arrivals on slot t , in bits.
- Data is stored in queues $Q_1(t)$ and $Q_2(t)$ to await transmission.
- Receiver coordinates network decisions every slot.
- Channel conditions $S(t) = (S_1(t), S_2(t))$ on slot t may change from slot to slot between users and the receiver.
- Every slot t , the network controller observes the current $S(t)$ and chooses a power allocation vector $p(t) = (p_1(t), p_2(t))$ within some set P of possible power allocations.
- This decision, together with the current $S(t)$, determines the transmission rate vector $(b_1(t), b_2(t))$ for slot t , where $b_k(t)$ represents the transmission rate (in bits/slot) from user $k \in \{1, 2\}$ to the receiver on slot t .

Exercises^a

1. The cellular backbone consists of some links (e.g., microwave links, free-space optical links, satellite links) that connect the air-interface with the rest of the end-to-end path, and then the cellular core network, and, finally, the public IP network.
Discuss how can all these factors reduce the useful throughput.
2. The Internet is complex (in number of tasks it has to manage) and big (in number of users). Discuss how hierarchical assignment plays an important role in end- to-end sessions: a YouTube streaming session from Google servers to your iPhone may traverse a wireless air-interface, a few links in the cellular core network, and then a sequence of even more links across possibly multiple ISPs in the public Internet.
3. Consider the two-queue system in the slides on mathematical

^aNot to hand in!

modelling and analysis.

- (a) Assume the arrival rate at $a_i(t)$ is $10i$ bits per sec, for $i = 1, 2$. What is the arrival rate per sec at the receiver?
 - (b) Assume the arrival rate at $a_i(t)$ is $10i$ bits per sec, for $i = 1, 2$. However, the second server $s_2(t)$ is faulty and loses 10% of its bits in processing. What is the arrival rate per sec at the receiver?
 - (c) Assume the arrival rate at $a_i(t)$ is $10i$ bits per sec, for $i = 1, 2$. Server 1 pays a cost of 2 units per bit transmitted while server 2 pays 3 units per bit transmitted. What is the payment per bit at the receiver?
4. Consider the two-queue system in the slides on mathematical modelling and analysis. For each $i = 1, 2$ at each time slot a packet arrives at node i with probability p_i . What is the total expected number of arrivals at the receiver after n time slots?

Reference

- M Chiang. Networked Life: 20 Questions and Answers, Princeton University Press, 2012.

lec -pdf
cla -

LAYERING

Networks are complex,
dynamic, adaptive,
communication systems

Layering and Network Architectures

- Design of network is very complex
- Modularity simplifies the design process
- Leads to hierarchy of *modules* where higher level modules are built from lower level in “black box” fashion

“data packetization”

“error control”

“transmission”

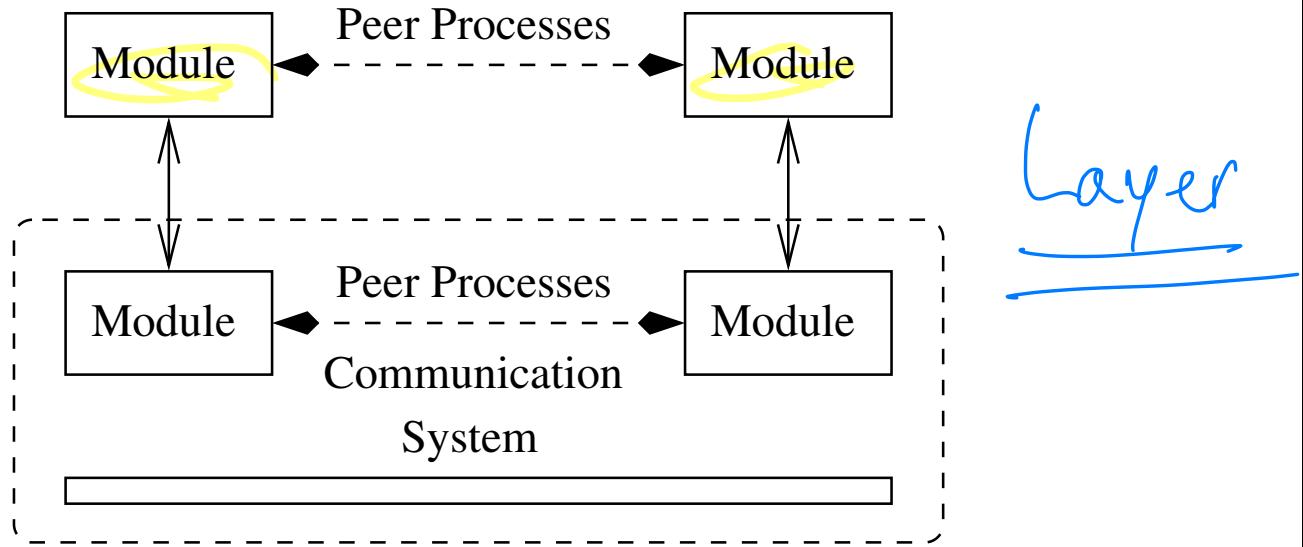
“routing”

Advantages of Layering

- Good design principle in general
- Simple and easy to understand
- Easy to modify and/or adapt to new situations/technologies
- Allows for different solutions for different situations
- Vendor competition

Layering in Networks

- Network provides for communication between two hosts
- Each host has the same hierarchy of modules
- Modules at the same level are called *peers*



- Layer $i - 1$ provides *services* to module i
- Services are realized by *protocols*

Two Important Layering Standards

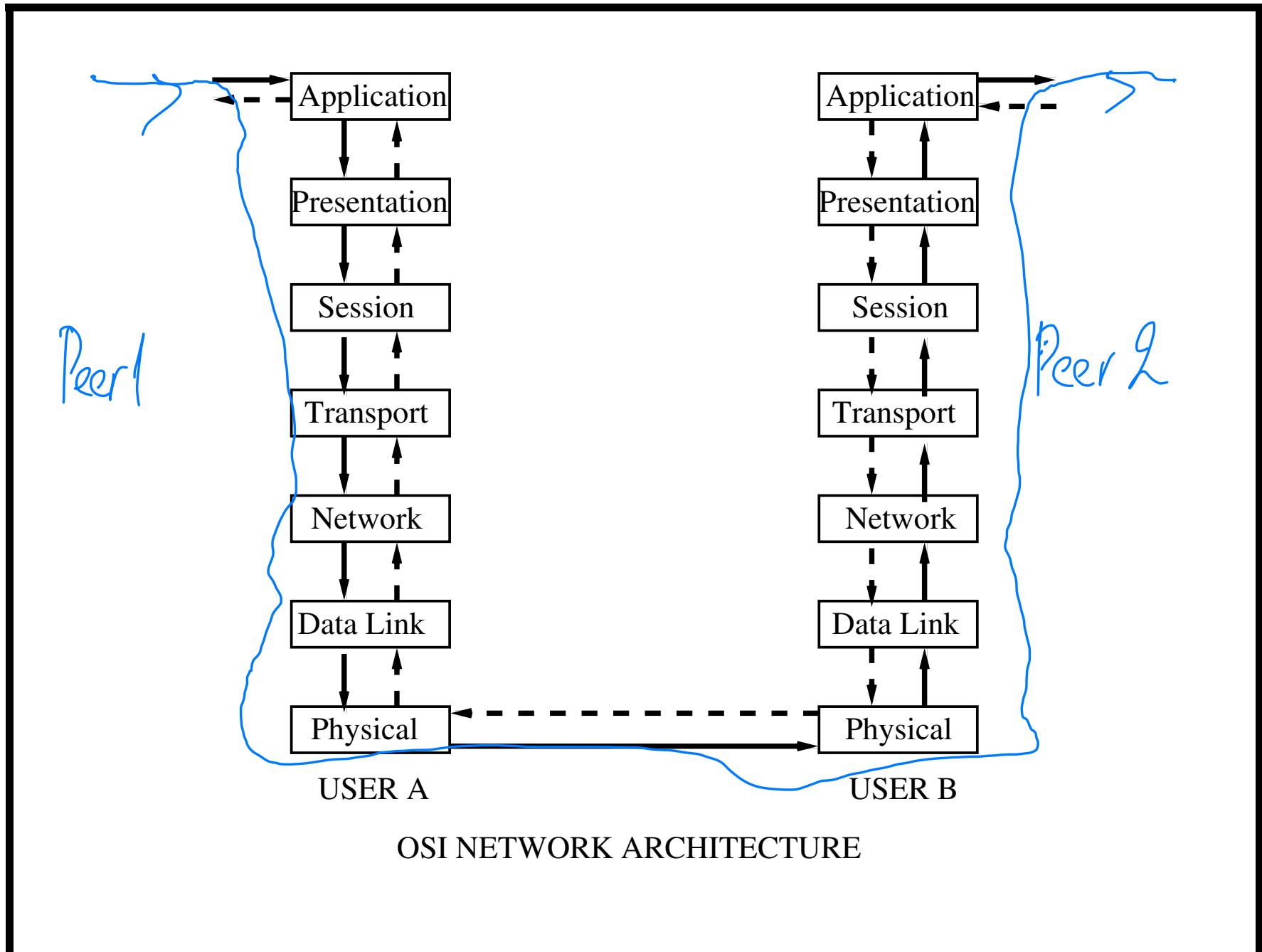
- OSI - Open Systems Interconnection
 - International Standards Organization (ISO), 1983
 - Academically important but largely ignored
- TCP/IP - Transmission Control Protocol/Internet Protocol
 - Invented by Cerf and Kahn, 1974, refined by others
 - TCP/IP is the dominant architecture
 - Practically important but difficult to describe.

OSI (Open Systems Interconnection)

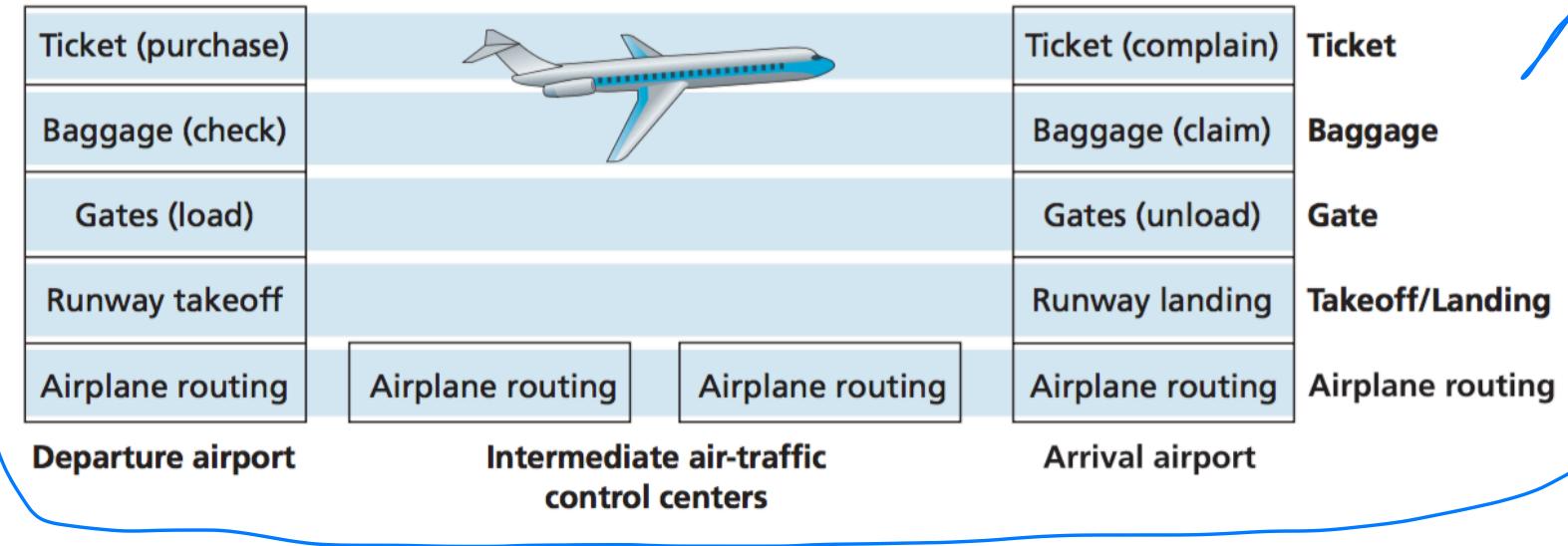
OSI

ISO (International Standards Organization) had the following guidelines in mind:

- One layer for each level of abstraction
- Layers perform well-defined functions
- Minimum of information flow between layers (i.e., simple *interfaces*)
- Layers not overloaded with functions
- Layers can be easily standardized



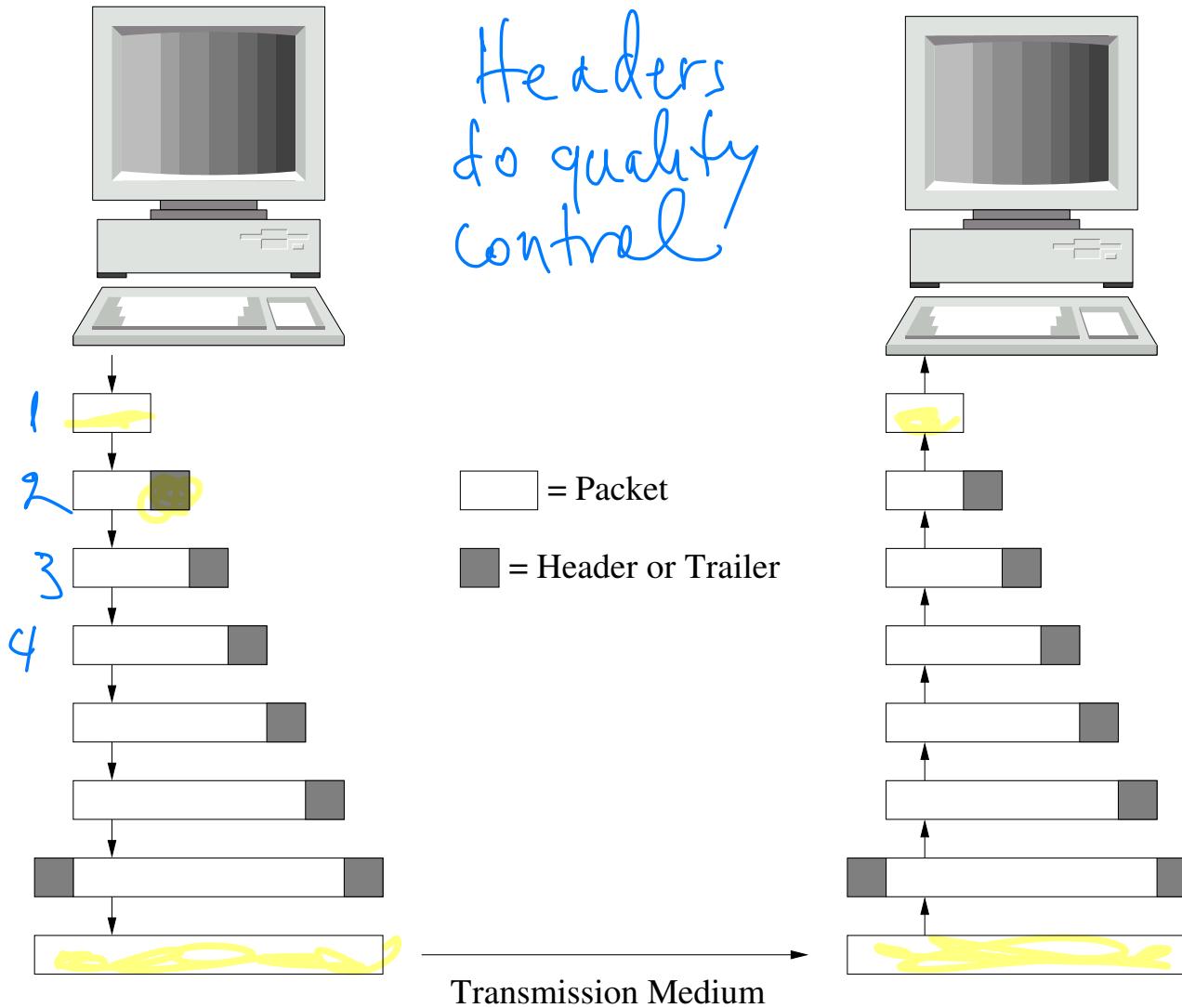
Layering in Airline Industry

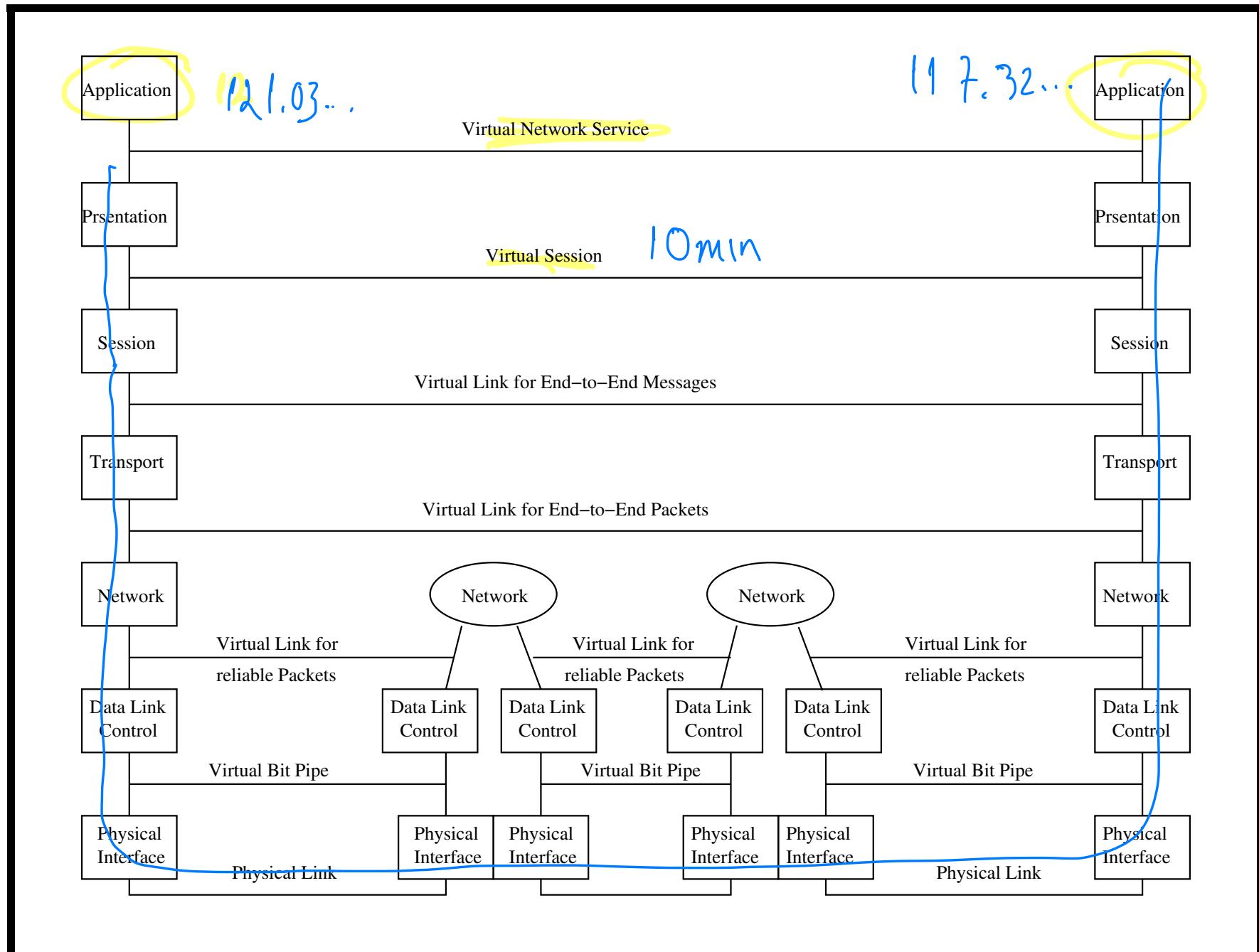


OSI's Seven Layers

- Physical
- Data Link
- Network
- Transport
- Session
- Presentation
- Application

OSI: Exchange





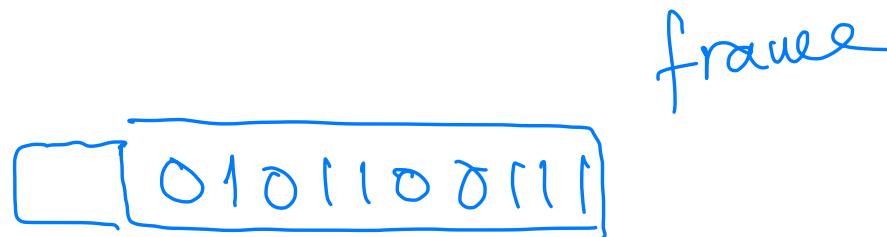
Physical Layer

- The Physical Layer is concerned with transmission of unstructured bit stream over a physical medium
 - Implements a *virtual bit pipe*
 - Has no idea what the bits mean
- Module that performs these functions is generally called a *modem*

modulates / demodulates

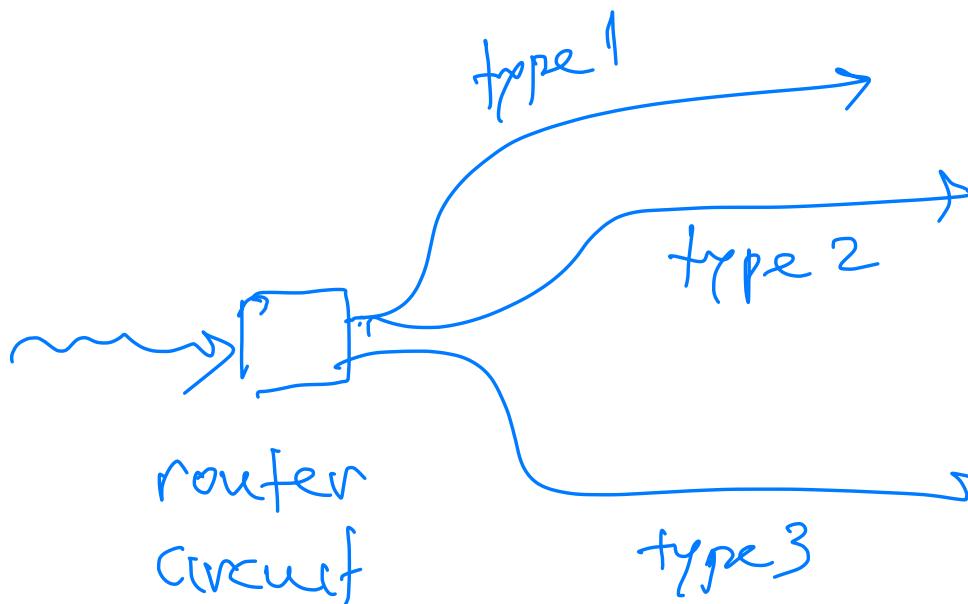
Data Link

- Converts unreliable bit pipe to virtual communication link
 - Sends packets asynchronously but error-free
 - Includes error detection and correction, framing (used to recognize the beginning and end of packet), retransmission strategies
- In broadcast networks there is a special Medium Access Control (MAC) sublayer



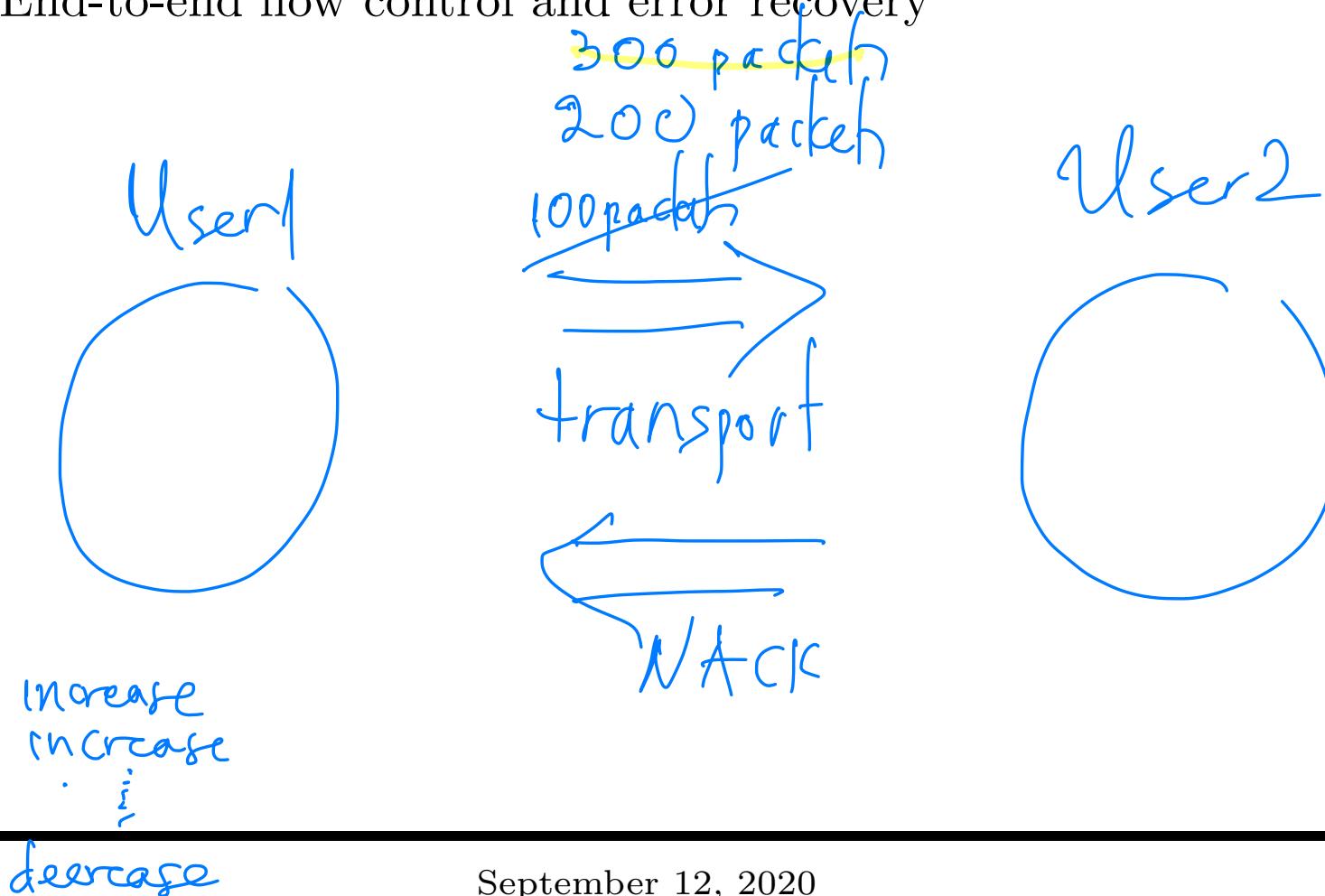
Network

- Responsible for establishing, maintaining and terminating connections
 - Routing and congestion control decisions
 - Internetworking
- Most distributed and therefore most complex layer



Transport

- Reliable, transparent data transfer between end points
- Packetization, multiplexing of sessions
- End-to-end flow control and error recovery



Session

- Control of communication between applications
- Load sharing
- Access rights
- Checkpointing for application recovery
- Directory assistance, ie, where services are available

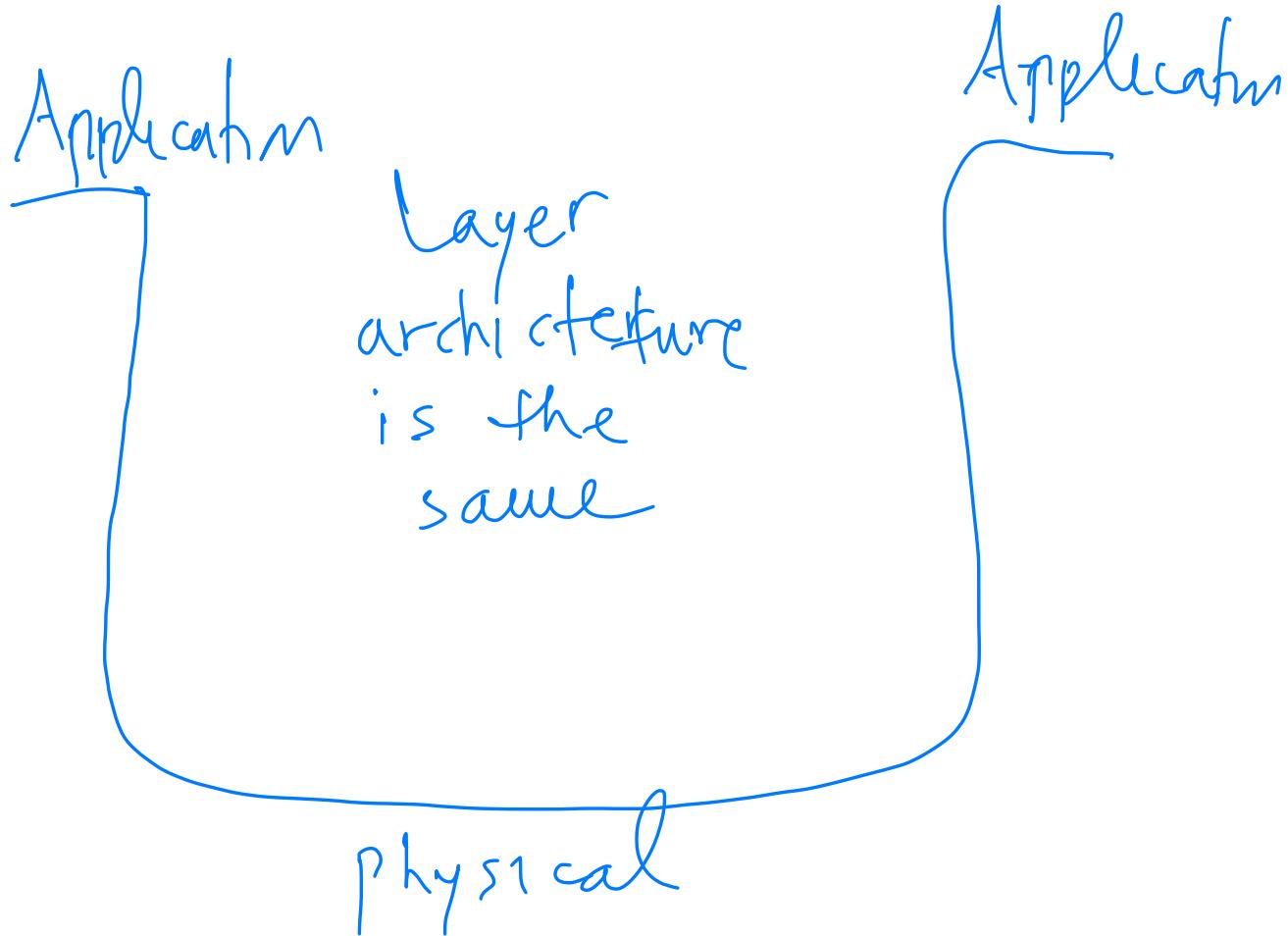
How long is the
communication to last.

Presentation

- Data encryption
- Data compression
- Code conversion

Application

- What ever is left over and things specific to an application



TCP/IP

TCP/IP is defined by protocols not by layer, although it is convenient to think of it as four layers:

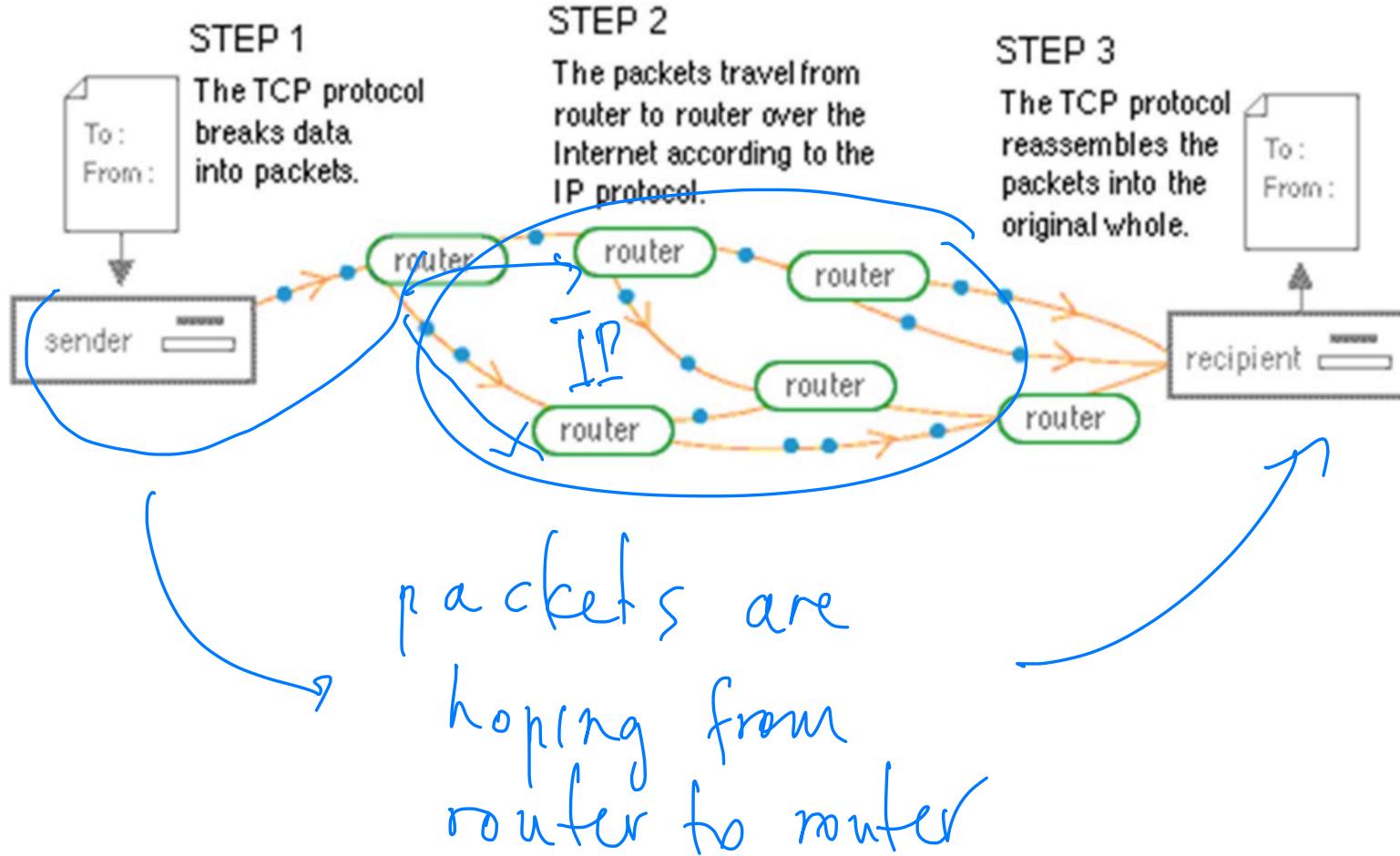
1. Application
2. Host-to-host transport (deals with variable length messages)
3. Internet (transmits datagrams), and
4. Physical (device drivers to perform bit-by-bit transmission).

TCP/IP

TCP/IP

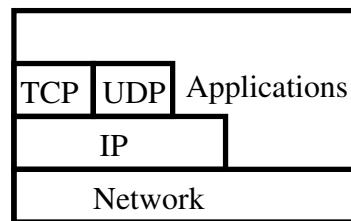
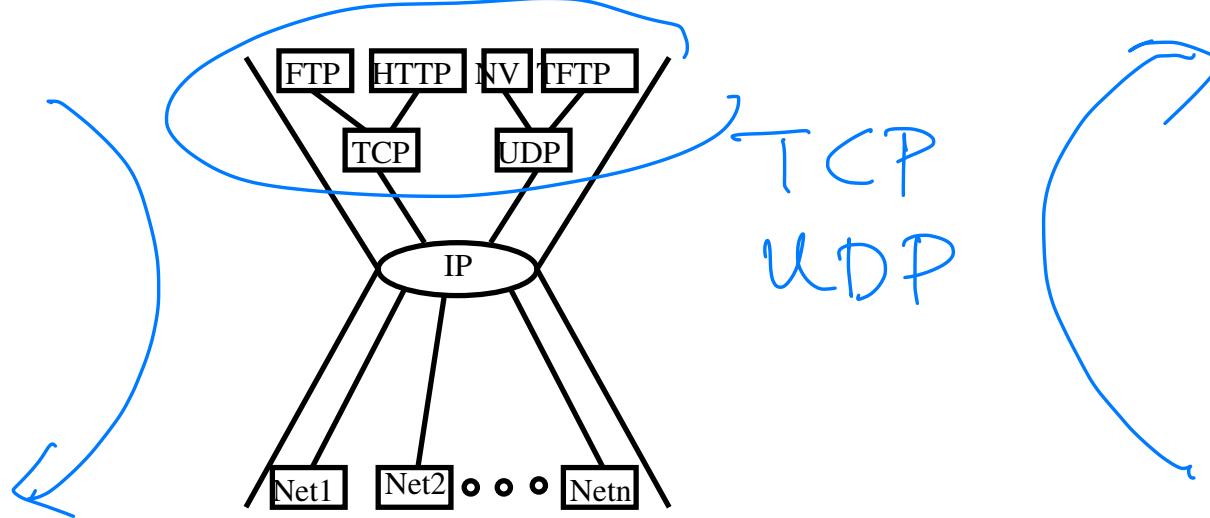
- Physical layer same as OSI
- Network and Data Link combined to form Network Access layer
- Remaining layers are
 - Internet layer
 - Transport layer
 - Application layer
- TCP/IP is not cleanly delineated!

TCP/IP: How it Works



TCP/IP Design

- Looks like a “funnel” with a design that is not clean.



TCP/IP ARCHITECTURE

- But is is very important!

Network Access Layer

- Concerned with data exchange within subnets
- Nothing specified other than the ability to send and receive IP packets
- No distinction made between the Data Link, MAC, internal subnet Network layers

Internet Layer

- Provides connectionless service between hosts
- Receives IP packets from host on any subnet and delivers them to host on any other subnet (possibly out of order)
- The internetworking portion of the OSI network layer
- Routing and congestion control between subnets are the major issues

Transport Layer

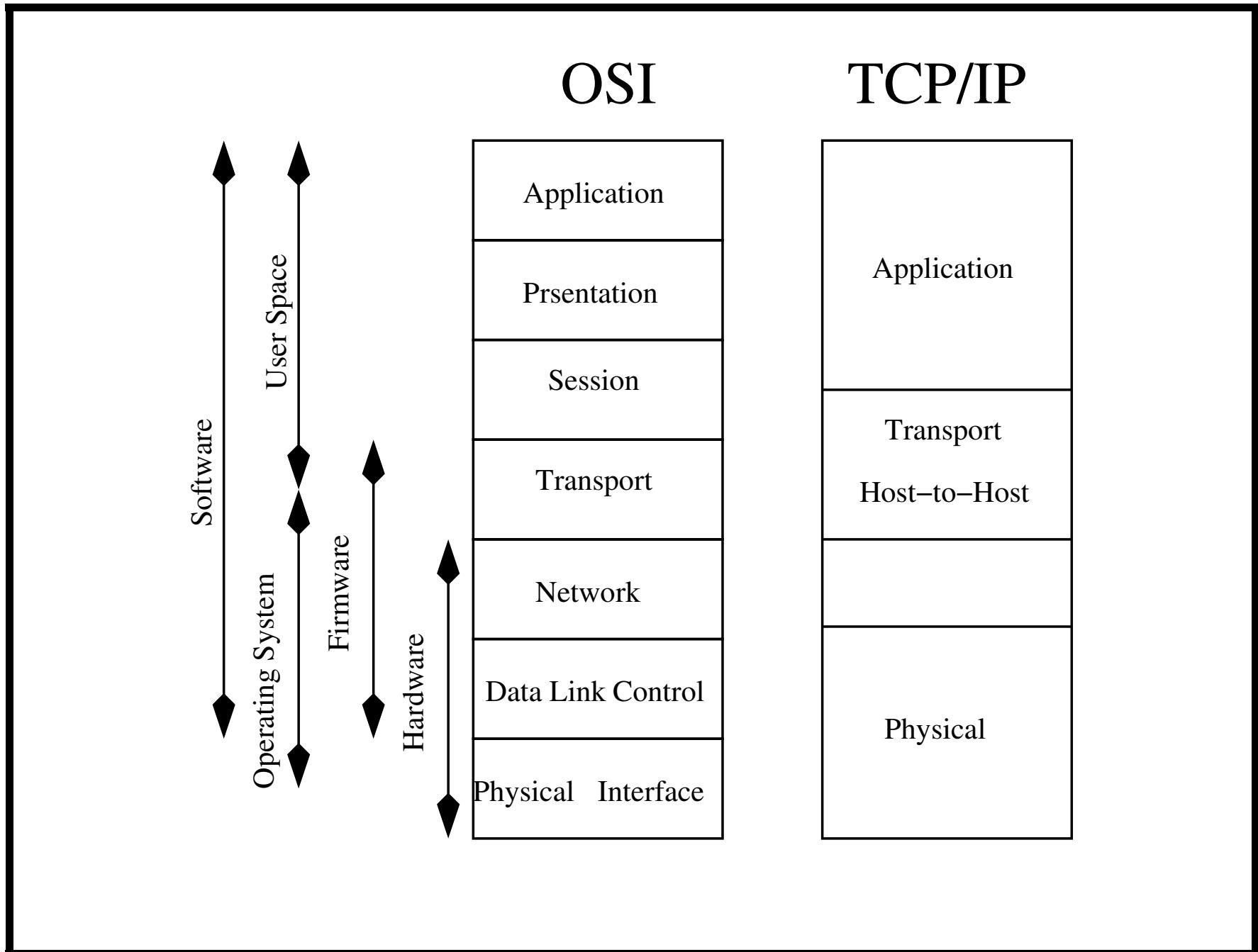
- Similar to OSI transport layer (packetization, error recovery)
- Provides two end-to-end protocols
 - TCP - Transmission Control Protocol - reliable connection-oriented service with flow control
 - UDP - User Datagram Protocol - unreliable connectionless service

Application Layer

- No presentation or session layers
- Examples of application layer protocols are
 - Telnet,
 - FTP,
 - SSH,
 - E-mail,
 - HTTP.

Comparison of OSI and TCP/IP

- OSI
 - Clean, thought out, explicit OO design
 - Not biased towards any protocol
 - Good for discussion but bad for implementation (too many layers)
- TCP/IP
 - Dirty afterthought to already developed protocol
 - Lower layers unspecified
 - Sloppy but practical



Load Sharing :-

Task
is to
be performed

20 MB

One processor

Dissamble the file

Part 1 $\Rightarrow g_1$

Part 2 $\Rightarrow g_2$

:

:

Part 20 $\Rightarrow g_{20}$

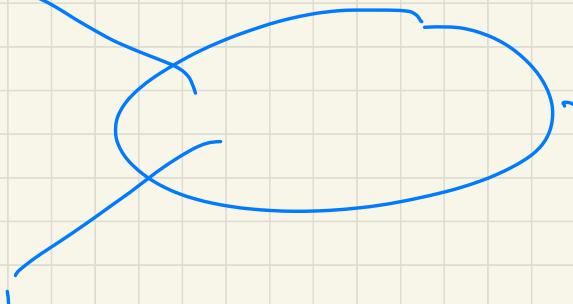
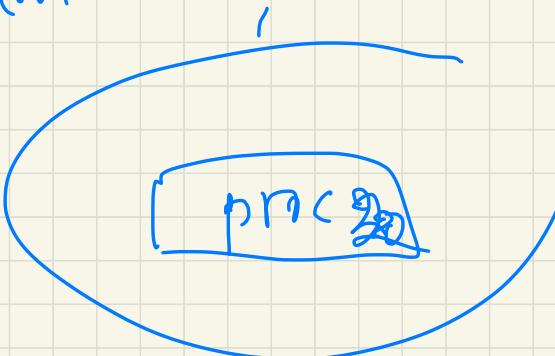
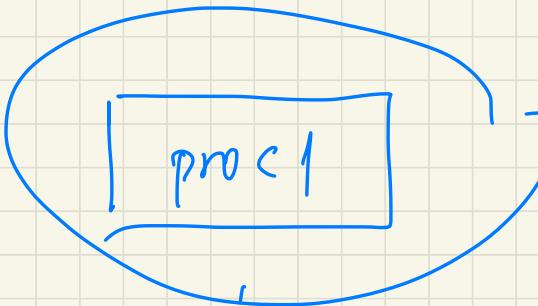
[Part₁]

.

.

[partition]

[Part₂₀]



In general, load sharing

partitions resources

so that work is balanced

among different processors

Load Sharing : processor
process .

Current and Future Trends

Robustness of Layering

- Keep in mind that layering is meant to provide guidance in computer network design.
- It is used in “copy, adapt, and use” manner:
 - *copy*: because you copy from older designs
 - *adapt*: because you modify designs in a new computer network, and
 - *use*: because you use it in the new network.
- Layering is “present” but “invisible” in all networks current and future.
- As such layering has proved to be robust.
- In some instances you can avoid layering when you design “overlay” networks. Nevertheless your network design “exists” between some of the OSI layers.

Current and Future Trends

- High Speed Networking
- Mobility
- Virtual Machines
- Intelligent Agents
- Ubiquitous Computing
- Pervasive Computing
- Wireless/Ad-hoc/Sensor Networks
- Satellite Networks
- Quantum Networks

High Speed Networking

- Integration of voice, video, data
- Optical media
- ATM (Asynchronous Transfer Mode)
- Gigabit Ethernet
- Trend towards “smart” networks

Mobility

- Wireless - untethered, eg, cordless phone
- Nomadic - geographic flexibility, e.g., extension phone
- Mobile - user in motion, e.g., car phone
- From wireless LAN to Teledesic (Satellite)

Virtual Machine Layer

- A layer between transport and application built into operating systems (JAVA).
- Network computers
- Functionality in the network not at the desktop
- Dynamic deployment of applications solves “Who goes first” problem

Intelligent Agents

- Autonomous active software
- Making appointments
- Suggesting purchases
- Intelligent search engines

Ubiquitous Computing

- Anytime, anywhere computing
- Computers as common as TVs and telephones
- Wearable computers? (Cyberborg)
- Virtual Reality Apps (e.g., Augmented Reality)
- Convergence of computing and communications

Satellite Networks: SpaceX^a

- September, 3, 2020: 12th batch of satellites launched.
- Has launched 715 Starlink satellites so far.
- Plans to put nearly 12,000 satellites in orbit and possibly expand to as many as 42,000.
- Goal is to provide internet around the world, particularly in areas where access has so far been unreliable or nonexistent.
- How fast data travels from the satellites to customers, and then back to the rest of the internet?
- Tests indicate download speeds higher than 100 Mbs per second. SpaceX has promised that Starlink will eventually be able to provide Gbs speeds once more satellites are operating.

^aNew Scientist, Sep 11, 2020

Quantum Networks

- Quantum networks are more secure than regular networks, because they rely on the quantum properties of photons, rather than computer code that can be cracked.
- Can build quantum networks using multiplexing
 - “**entanglement**”: a quantum property that links a pair of photons, so that measuring one of them instantly influences the measured state of the other, regardless of distance.
- Rather than connecting users one-to-one, multiplexing entanglement splits photons from a single laser according to their wavelength.
- Each wavelength can hold a data stream, meaning the system could support between 50 and 100 users with existing hardware
- Despite this, building one is still expensive.

Introduction to Queues

A queue in "every day" life is a line of people waiting for service.

Outline

1. Motivation

2. Collisions

3. Queues

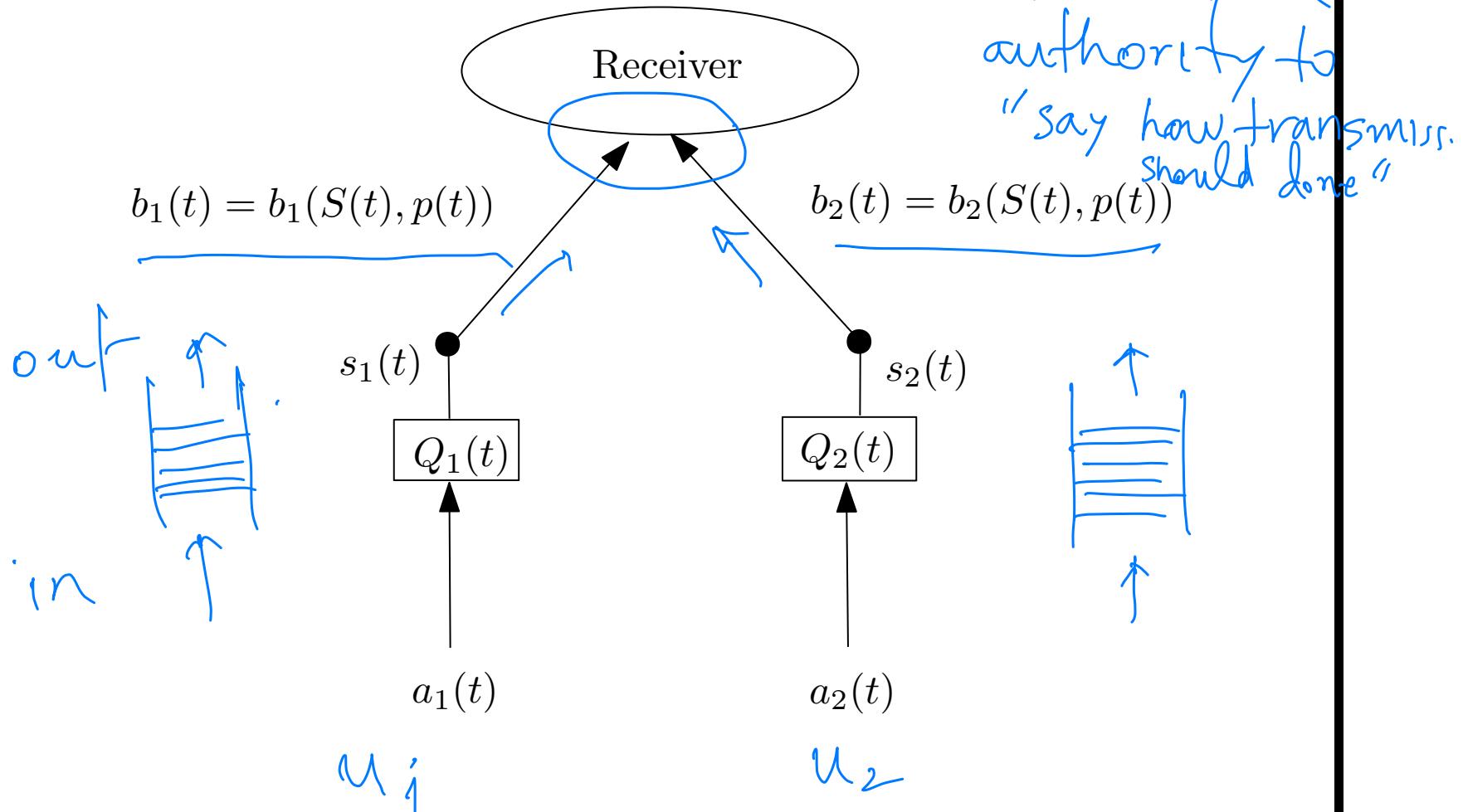


Important to understand
Frequency of collision
If we understand that,
then we can design
algorithms, protocols, etc

Motivation

Scheduling in a Simple Queueing System (1/2)

- Consider a 2-user wireless uplink that operates in slotted time $t \in \{0, 1, 2, \dots\}$.

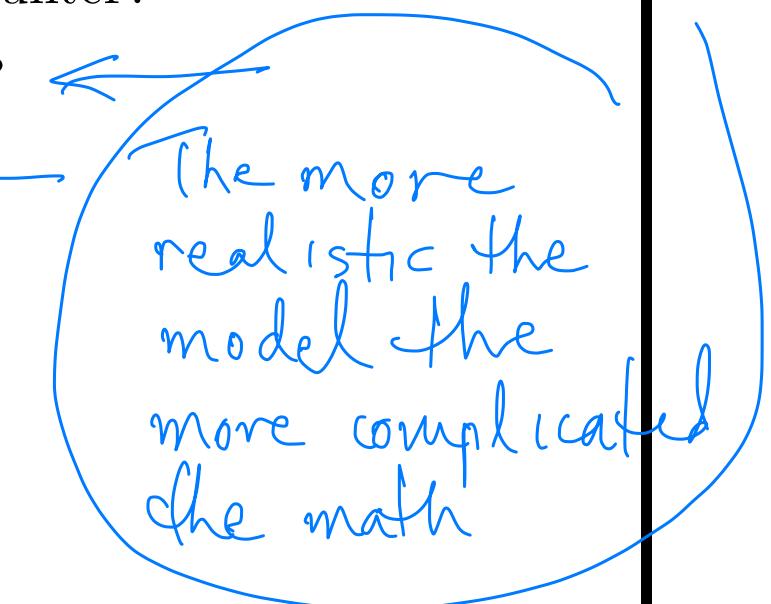


Scheduling in a Simple Queueing System (2/2)

- The unit of transmission of a network is the packet: basically an array of bits transmitted as a “unit”.
- Every time slot new data randomly arrives to each user for transmission to a common receiver.
time
- Let $(a_1(t), a_2(t))$ be the vector of new arrivals on slot t , in units of bits.
- The data is stored in queues $Q_1(t)$ and $Q_2(t)$ to await transmission.
- The receiver coordinates network decisions every slot.

Modelling the Behavior of the System

- Although we have not yet defined what a packet is we can still try to understand what difficulties arise in network transmission.
- In particular, we would like to answer:
 - How can we model a system of packets?
 - What kind of difficulties do we encounter?
 - What is adequate mathematization?
 - Does the model reflect reality?
- Can't do all of them at once!
- Lets just look at packet collisions.

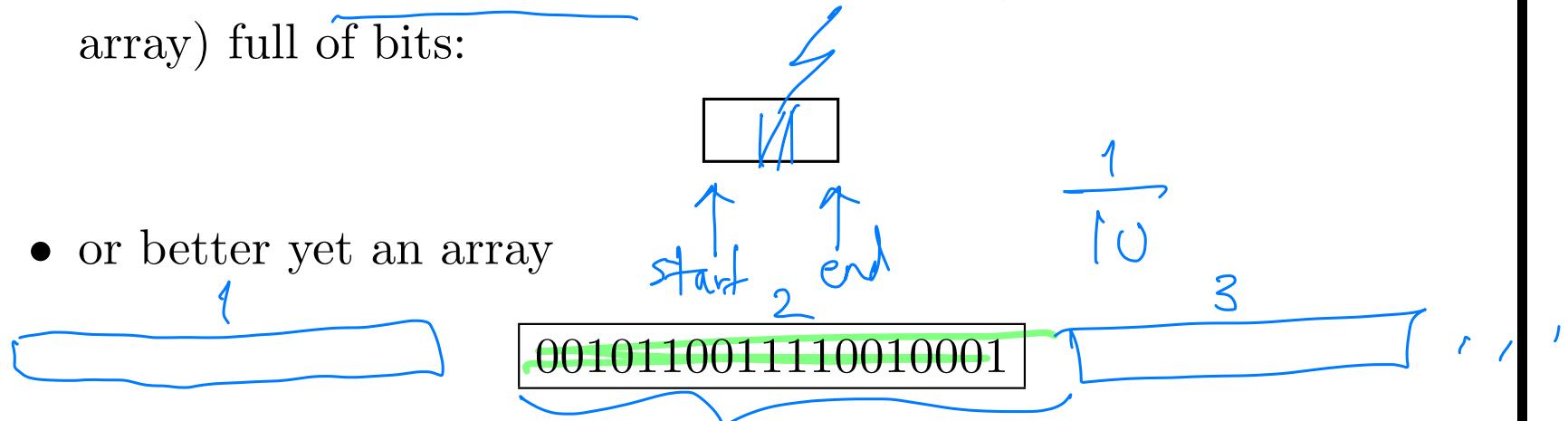


The more
realistic the
model the
more complicated
the math

Two blue arrows point from the text "The more realistic the model the more complicated the math" to the first four bullet points in the list above.

What is a Network Packet (1/2)

- Think of a network packet as a rectangle (in fact a rectangular array) full of bits:



- or better yet an array

whose length depends on the network technology being used.

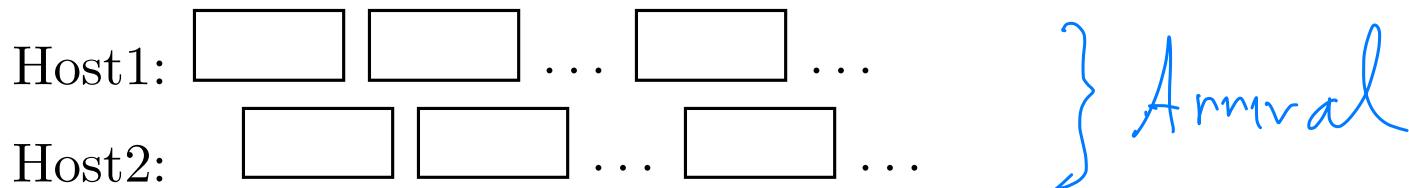
- How long” It can be thousands of bits long!
- The rectangle “occupies” the medium for its transmission from its start to its end.
- To arrive correctly it cannot be interrupted!

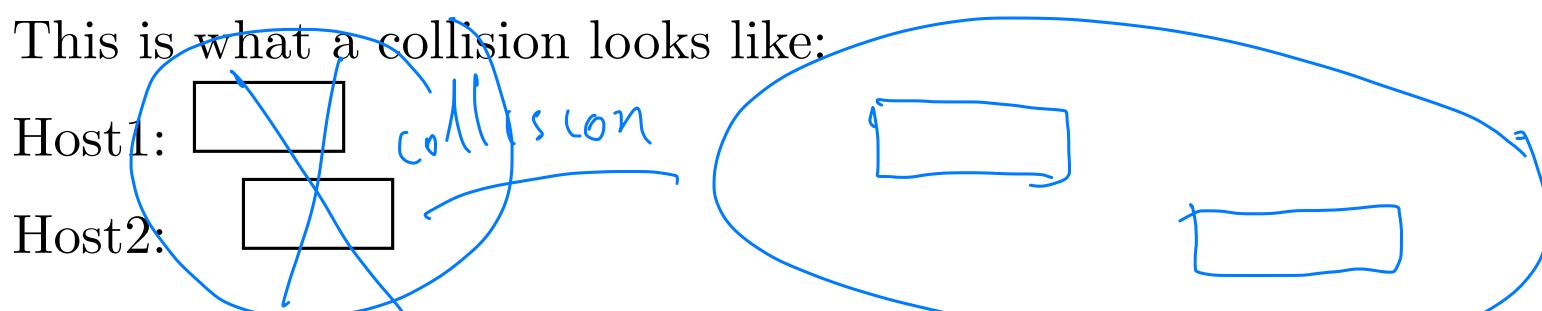
Length of packet depends on
technology used. Rule of thumb

- Wireline Networks,
packets are "longer"
- Wireless Medium
packets are "shorter"

What is a Network Packet (2/2)

- What are some causes for interruption?
- Multiple packets may be transmitted by 2 different hosts, e.g.,

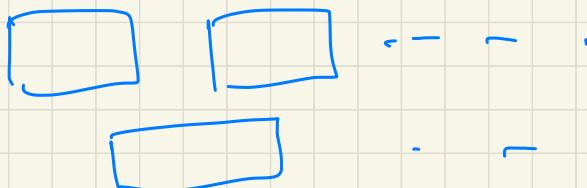


- Notice that the hosts are a bit off sync! Unless they can be differentiated somehow by the receiver the packets may collide!
- This is what a collision looks like:

Host1: 
Host2: 
- How likely are collisions? How can we analyze this problem in a transmission system with multiple users sharing the medium?

What is the receiver?

Physical { wire
medium

Protocol }
Session



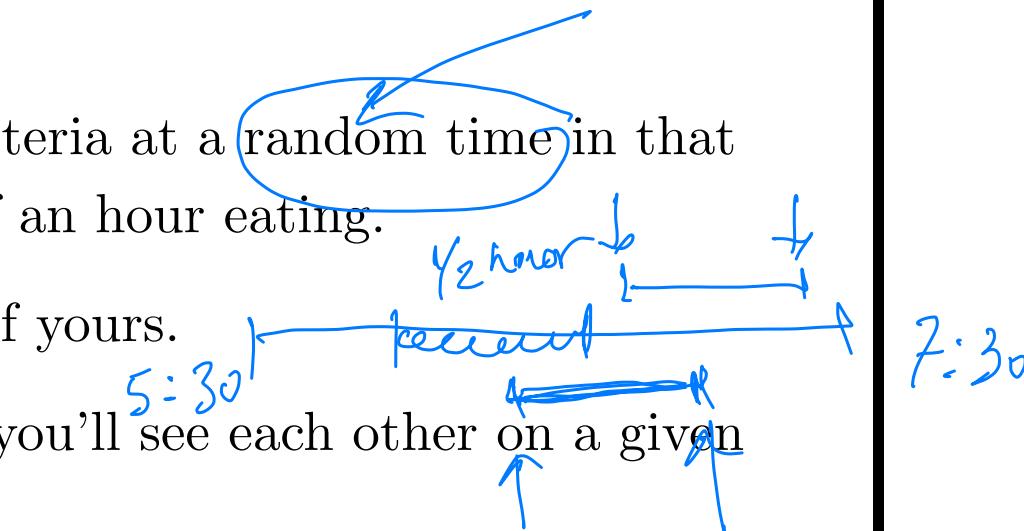
Collisions

Arrival Times in a Meeting

- There are many instances in every day life where collisions may occur.
 - Waiting for the bus or taxi to arrive
 - Waiting in a bank queue to make a transaction
 - Going to a doctor appointment
 - Waiting for an office hour
 - Shopping at a supermarket
- In the sequel, we use the cafeteria paradigm!

Meeting a Friend at the Cafeteria

- The cafeteria is open for dinner between the hours of 5:30 and 7:30 pm.
- You tend to arrive at the cafeteria at a random time in that interval, and you stay for half an hour eating.
- The same is true of a friend of yours.
- What is the probability that you'll see each other on a given day?
- Why is this the same as the packet collision problem?
- Note: the times you and your friend arrive at the cafeteria for dinner vary in a continuous space since they can be any moments between 5:30 and 7:30 pm.



Arrival Times

- What does it mean you arrive at the cafeteria at a random time between 5:30 and 7:30 pm?

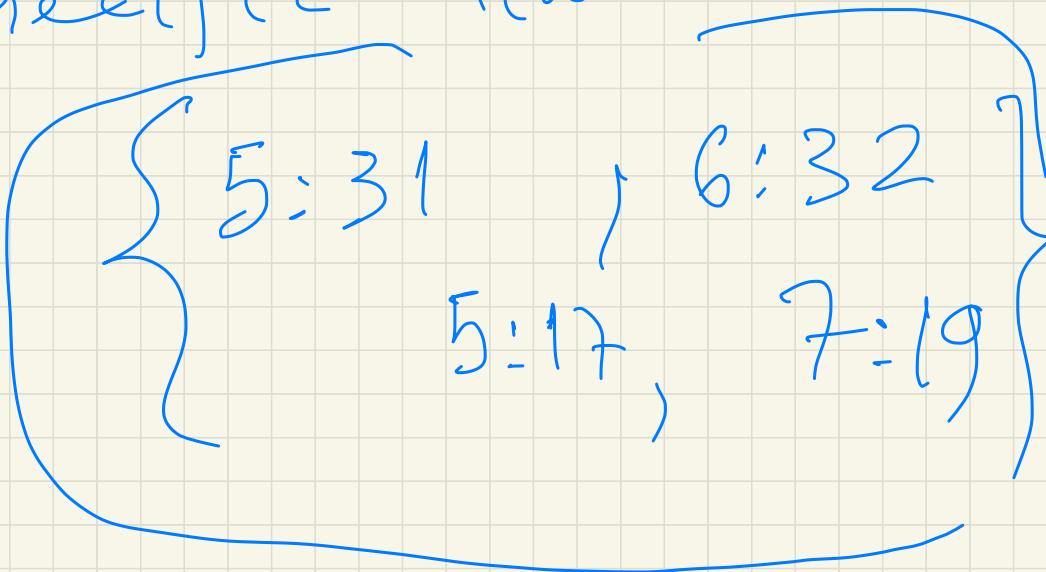
the probability that you'll arrive in a given interval is equal to the probability that you'll arrive in any other interval of the same duration.

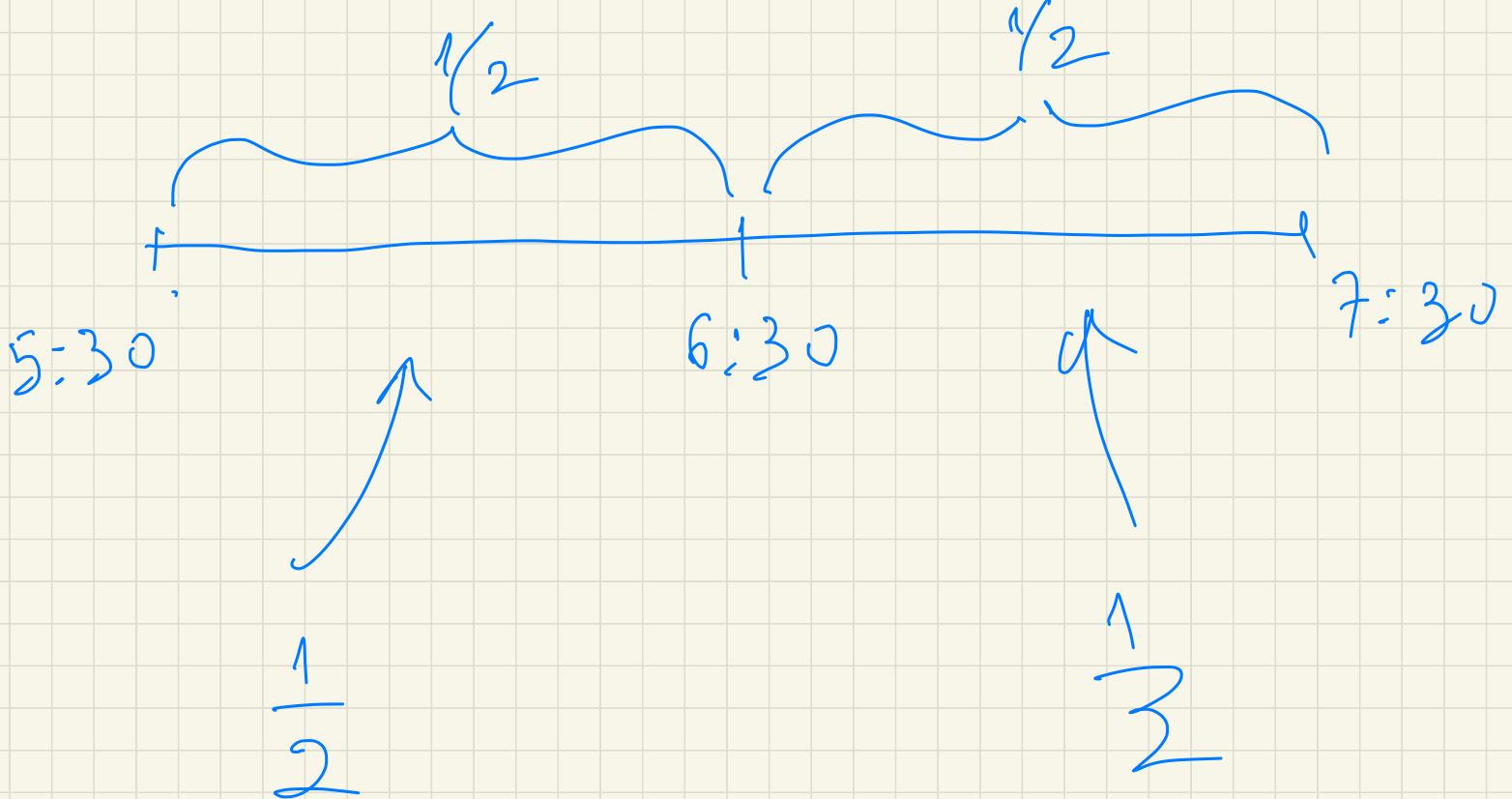
- Since the whole interval has length 2 hours, if the choice is arriving either between 5:30 and 6:30 or between 6:30 and 7:30 then the probability of each must be exactly $1/2$.
- Similarly, the probability of our arriving in any given hour-long interval, say, between 5:37 and 6:37 must be $1/2$ as well.
- Similarly, the probability of arriving in any half-hour interval must be $1/4$; and, in general, the probability of our arrival in any interval is one-half the length, in hours, of that interval.

Uniform or Poisson

Probability of arrival at
an given time is "independent"
of the specific time.

Equally
likely
to arrive



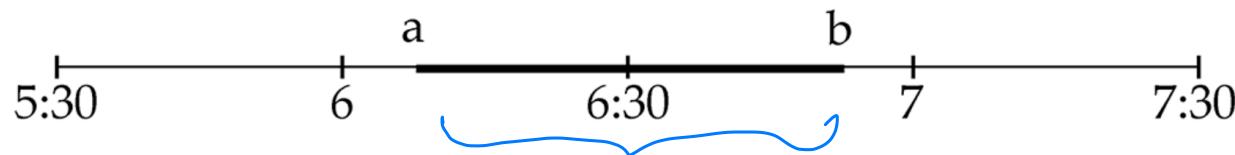


An Interval Representation

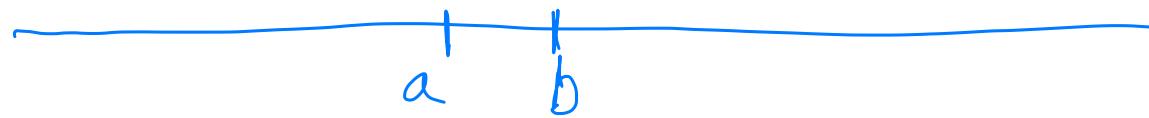
- If we “normalize” the parameters, i.e., think of the entire line (time segment between 5:30 and 7:30 pm) as having length 1 then the probability that your time of arrival t is between two given times a and b is given by the formula

$$\Pr[a \leq t \leq b] = b - a.$$

- If we represent your arrival time as a point on the line segment $[0, 1]$ (a number t between 0 and 1) then we can represent



your friend's time of arrival as a number s between 0 and 1.



$$b = 6:30$$

$$6:30 - 5:22$$

$$a = 5:22$$

$$= 68 \text{ min.}$$

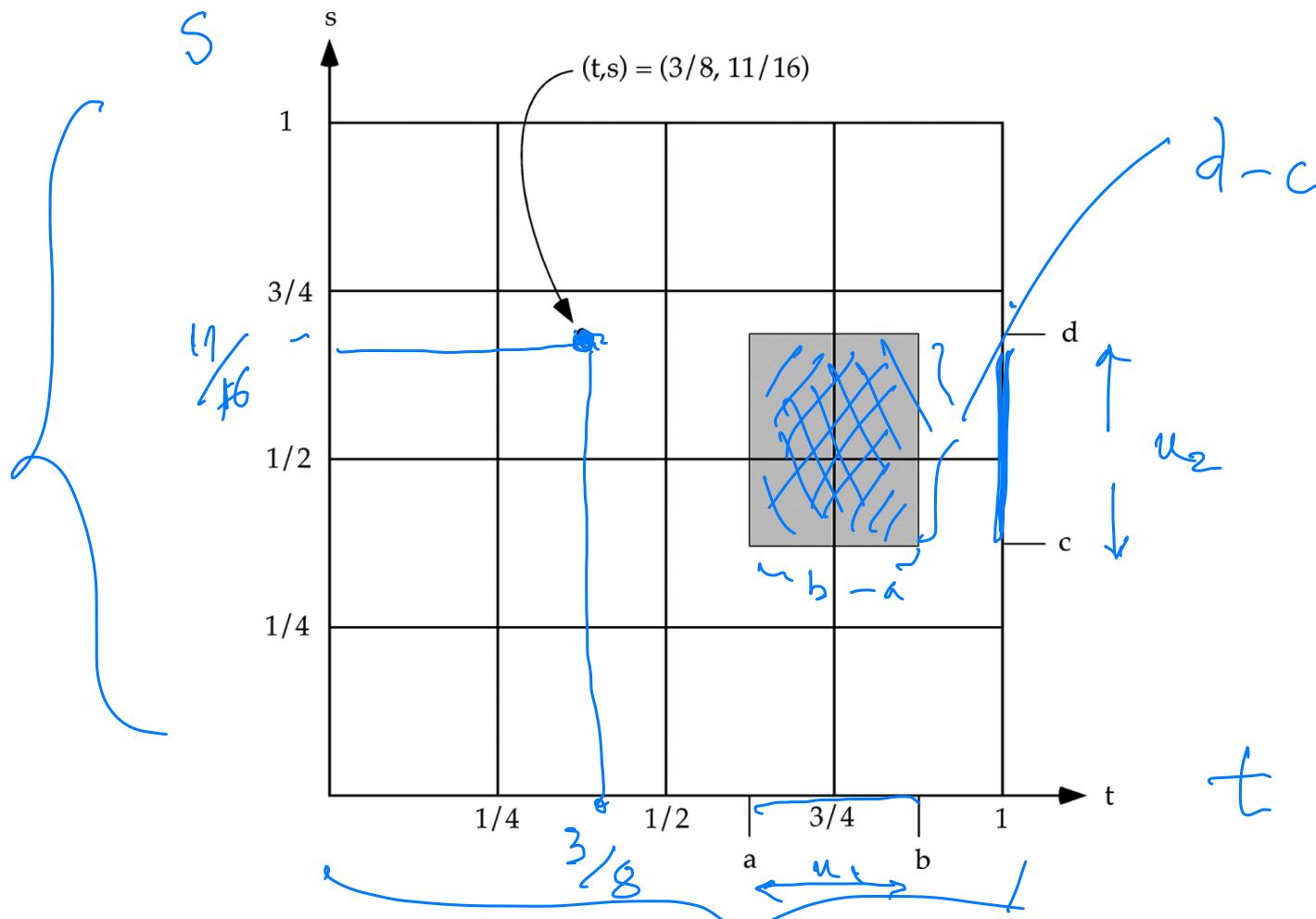
$$\Pr[a < t < b] = \frac{68}{120} = \frac{17}{30}$$

≈ 56

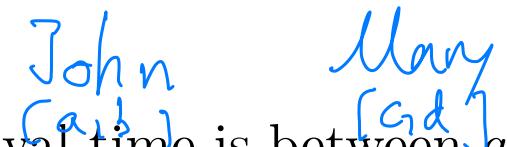
$$2 \text{ hrs} = 120 \text{ min}$$

A Unit Square Representation

- We can then view the pair of times s and t as a point in a square of unit side length 1.



Independence of Arrivals

- The next crucial assumption is that your time of arrival and your friend's are independent events
- In that case, the probability that your arrival time is between a and b and your friend's arrival time is between c and d is the product of the probabilities of the two individual events:


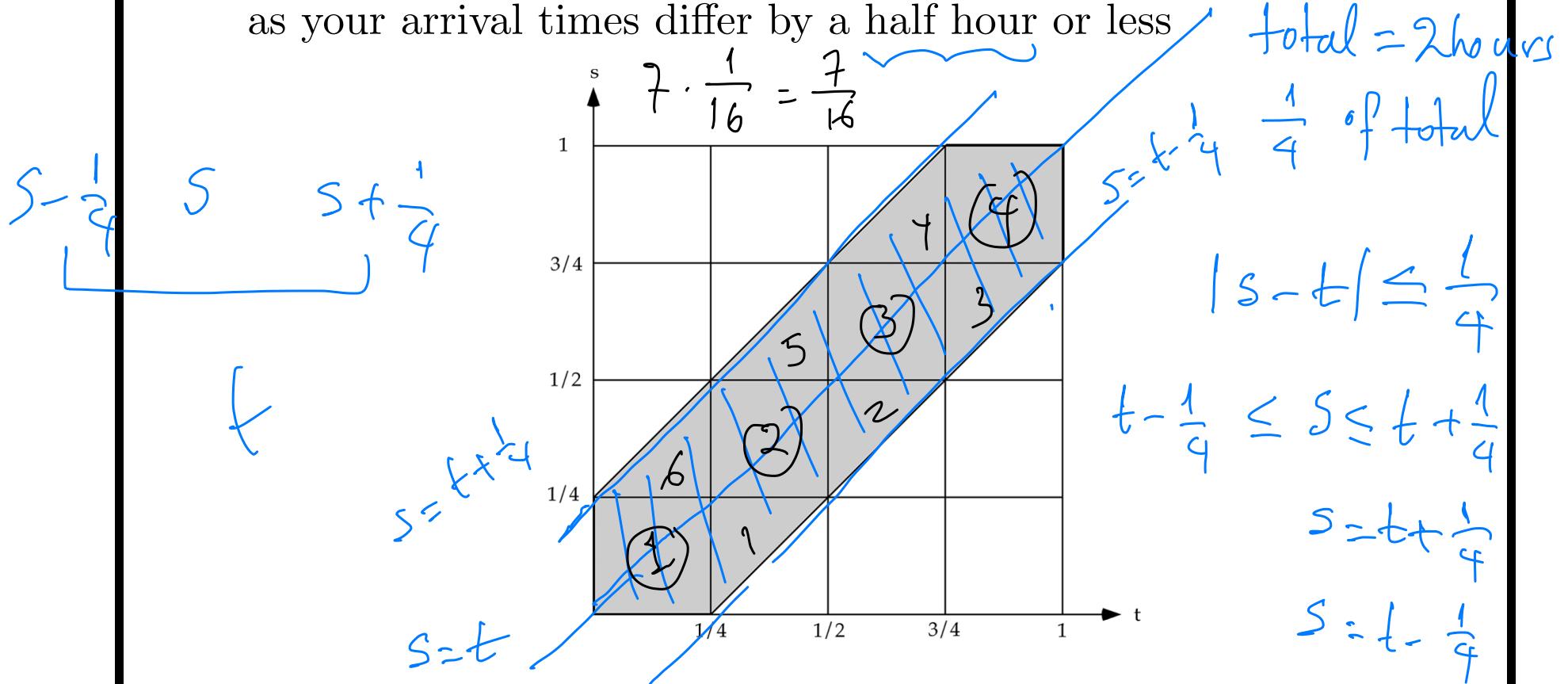
$$(b - a)(d - c)$$

which is the area of a rectangle with side lengths $b - a$ and $d - c$.

- More generally, if your time of arrival and your friends are independent, then the probability that (t, s) falls in any region in the square is equal to the area of that region.

Overlap (1/2)

- Observe that you and your friend's dinners overlap is the same as your arrival times differ by a half hour or less



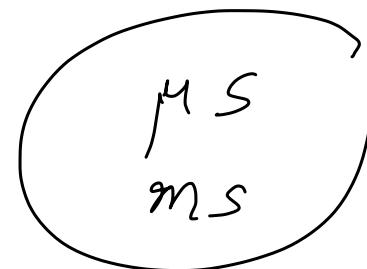
- Since a half hour is one-quarter of the time between 5:30 and 7:30, this is represented by the inequality $|s - t| \leq \frac{1}{4}$.

Overlap (2/2)

- What is the area of this shaded region?
- The grid drawn breaks the region up into 4 squares, each $1/4$ by $1/4$, plus 6 isosceles right triangles with side length $1/4$.
- The squares have area $1/16$, and the triangles half that, or $1/32$; the total area is

$$4 \cdot \frac{1}{16} + 6 \cdot \frac{1}{32} = \frac{7}{16}.$$

- So, the probability their dinners overlap is $\frac{7}{16}$!



In case of a network, You must know the following things.

1) Length of packet

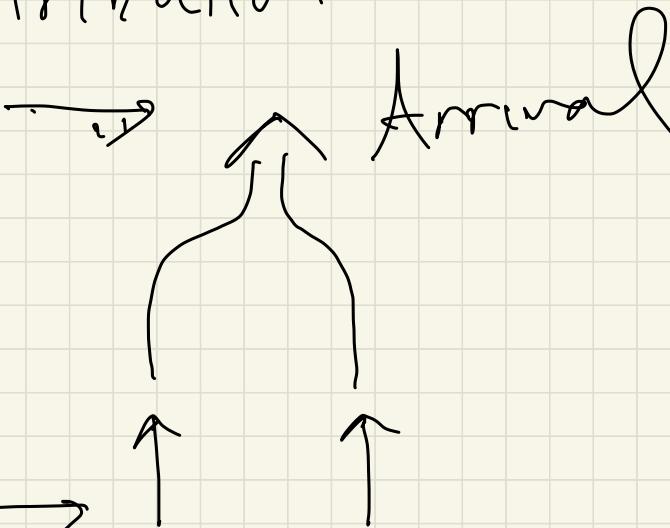
2) Sending distribution

3) Arrival

{ If prob. of collision is

$$10^{-3}$$

$$10^{-6}$$



Are you Being Stalked? (1/2)

- Assume you arrive at the cafeteria at random times between 5:30 and 7:30 pm.
- You notice at one point that you're seeing a lot of this one other person.
- You decide to keep track, and over the course of the next 10 days you see them 8 times at dinner.
- Can this be a random occurrence, or are you being stalked?
- We are asking is the following question:
what is the likelihood of seeing this person at dinner 8 or more times in 10 days, assuming that their arrival times are in fact random.

Are you Being Stalked? (2/2)

- We've shown, the probability their dinners overlap is $\frac{7}{16}$.
- Treat the 10 days as a series of Bernoulli trials, in which case as we have seen the chances of seeing them exactly 8 times out of 10 or exactly 9 times out of 10 or exactly 10 times would be

Bernoulli trials

$$\left\{ \begin{array}{l} \binom{10}{8} \cdot \left(\frac{7}{16}\right)^8 \cdot \left(\frac{9}{16}\right)^2 \approx 0.019 \\ \binom{10}{9} \cdot \left(\frac{7}{16}\right)^9 \cdot \left(\frac{9}{16}\right)^1 \approx 0.0033 \\ \binom{10}{10} \cdot \left(\frac{7}{16}\right)^{10} \cdot \left(\frac{9}{16}\right)^0 \approx 0.00025 \end{array} \right.$$

$$\left(\frac{10}{8} \right) p^8 (1-p)^2$$

$$\left(\frac{10}{9} \right) p^9 (1-p)^1$$

$$\left(\frac{10}{10} \right) p^{10}$$

≈ 0.02

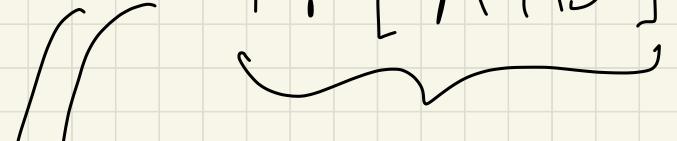
- Adding these up, we see that assuming that the other person's arrival times are random, there is only a 1-in-50 chance that you would see them at dinner 8 or more times in 10 days.

Interpretations

- What we have established so far is that the probability of at least 8 sightings in 10 days, assuming random arrival times, is about 0.02.
- To put it more succinctly, let A be the event that you see this person 8 or more times in 10 days, and let B be the event that their arrival times at the cafeteria are random.
 - What we know then is that $\Pr[A \text{ assuming } B] \approx 0.02$
 - So: it is unlikely that you'd see this person that often, if their arrival times were indeed random.
 - What we are actually asking, though, is something different: we are asking, what is the probability that their arrival times are random, given that you have seen them 8 or more times in 10 days? In other words, what is $\Pr[B \text{ assuming } A]$?

$$\Pr[A | B]$$

know $\Pr[A|B]$ = $\frac{\Pr[(A \& B)]}{\Pr[B]}$



$\Pr[B|A] = \frac{\Pr[(B \& A)]}{\Pr[A]}$



can compute

Bayes' Rule:

Bayes Rule

- Note that $\Pr[A \text{ assuming } B]$ and $\Pr[B \text{ assuming } A]$ are not the same thing!
- According to Bayes' theorem, in order to relate the two we would need to know $\Pr[B]$, or, equivalently, $P[\text{not } B]$, that is what were the odds that you were being stalked independently of this observation.
- In other words, are stalkers a commonplace part of your life?
- In sum: the fact that you see this person as often as you do may be significant: as a rule of thumb, statisticians view occurrences with less than a 5% probability as significant, but it certainly doesn't mean that the probability you're being stalked is 98%.

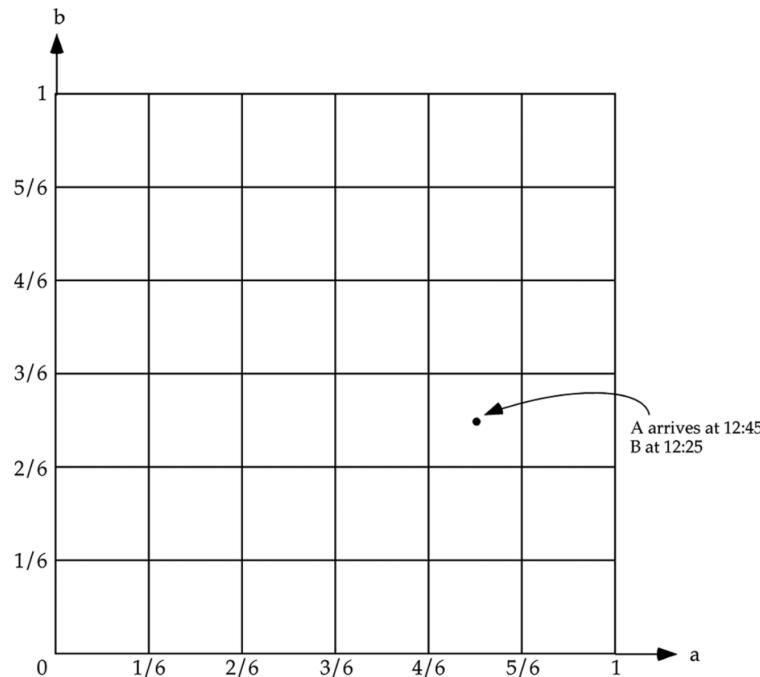
Queues

Customers

- You are the manager of a remote post office branch which employs just one clerk.
- You are concerned that with only one clerk, some of your customers may have to wait for service.
- Observe the traffic over a period of time, and you see
 1. 4 customers come in at midday:
 2. 1 (called Early Bird) comes in every day at noon:
 3. 1 (called Late Bird) comes in every day at 1, and
 4. 2 others (called A, B) come in (in any order) at independent random times between 12 and 1.
- Transactions with each of these customers take 10 minutes.
- Since no one at all shows up between 11 and 12, the teller is always available right at 12:00.

Representation

- Let a (resp. b) be the time in hours after 12 that A (resp. B) shows up.
- Draw a square representing possible arrival times (a, b) .



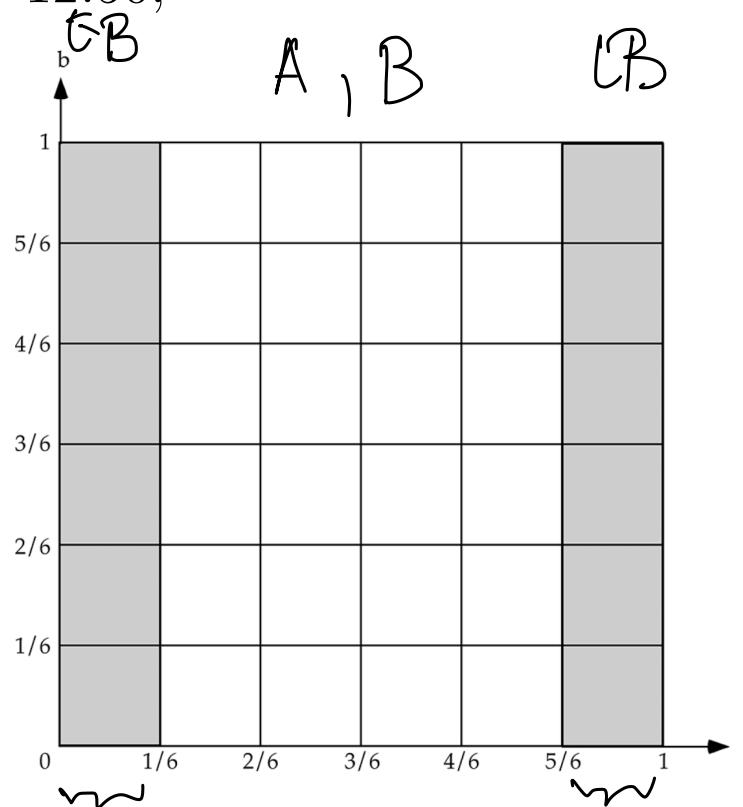
- What is the probability that one or more of the customers will have to wait?

Representation

- The sides of the square represent the hour between 12 and 1, marked off into six increments of $1/6$ of an hour, or 10 minutes.
- As before, we make the sides of the square of length 1, so that as before the probability that the pair (a, b) of arrival times lies in a given region in the square is equal to the area of that region.
- Mark off the possible arrival times (a, b) corresponding to scenarios where one or more of the customers will have to wait.

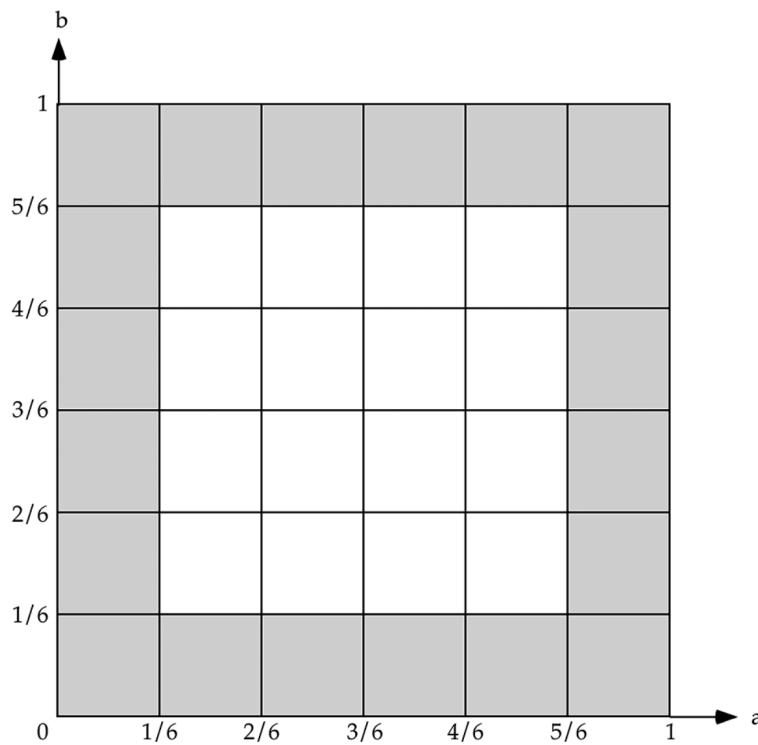
Representation

- Early Bird never has to wait, since the teller is always available when he shows up at 12:00.
- If A arrives before 12:10, she will have to wait; and likewise if she arrives after 12:50,



Representation

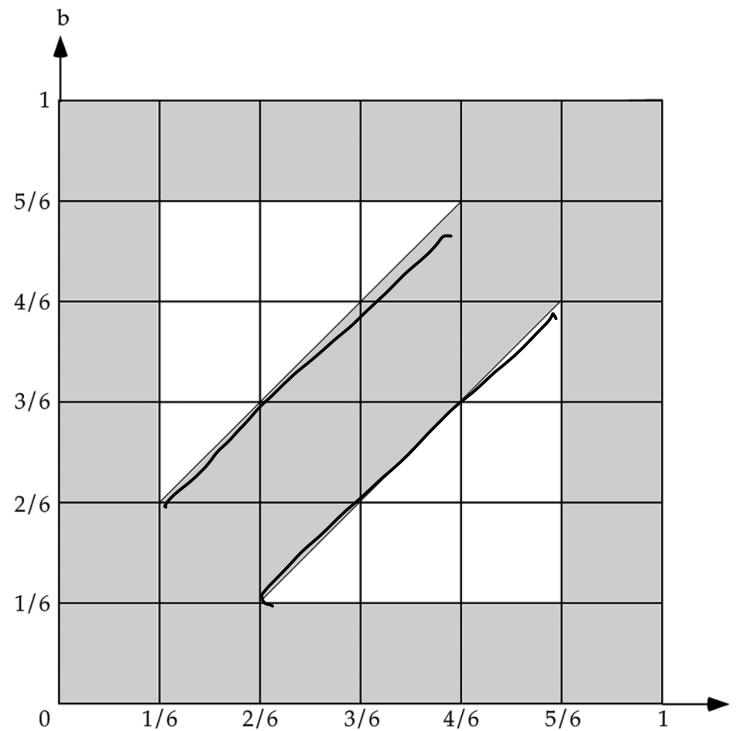
- The same is true for B as well *if she arrives before 12:10 or after 12:50, someone is going to have to wait) and we mark off those outcomes as well:



A : vertical
B : horizontal

Representation

- Finally, if A and B arrive within 10 minutes of each other, one of them will have to wait, so we mark off points (a, b) with a and b within $1/6$ of each other, that is, within $1/6$ of the diagonal either horizontally or vertically.



- And this is what we are left with!

Outcome

- Finally, we are ready to answer the question of how likely it is that someone will have to wait: this is just the area of the shaded portion of our square.
- To calculate it, it might be easier just to find the area of the rest (the locus of (a, b) corresponding to outcomes where no one has to wait).
- This region consists of two triangles, that fit together to form a $1/2 \times 1/2$ square; so the probability that no one will have to wait is $\frac{1}{2} \cdot \frac{1}{2} = 1/4$, and the likelihood that someone will have to wait is $1 - 1/4 = 3/4$, or three in four

Conclusion

- That was an extremely simple example of a more general problem, belonging to an area called queuing theory.
- It is concerned with what happens when a number of people show up at random times, but where the number of people may be large, and the total number of people is unknown.
- It's relevant not only to service businesses, as in the example we just did, but also to things like computer, data packet, and phone networks: if people make calls at random times, how large does the bandwidth of the network have to be to have, say, only a 5% chance of an overload occurring during a given call?

Erlang

MEASURING PERFORMANCE

How fast the network is
How much buffer do you need

How do we quantify these terms

Technology Independent Models

Time : n^2 , $n \log n$, 2^n time

Space : n^3 , n bits

Technology Dependent Models

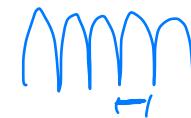
What is the unit?

Tech. Dep. Complements Tech. Indep.
Models

Frequency and Period

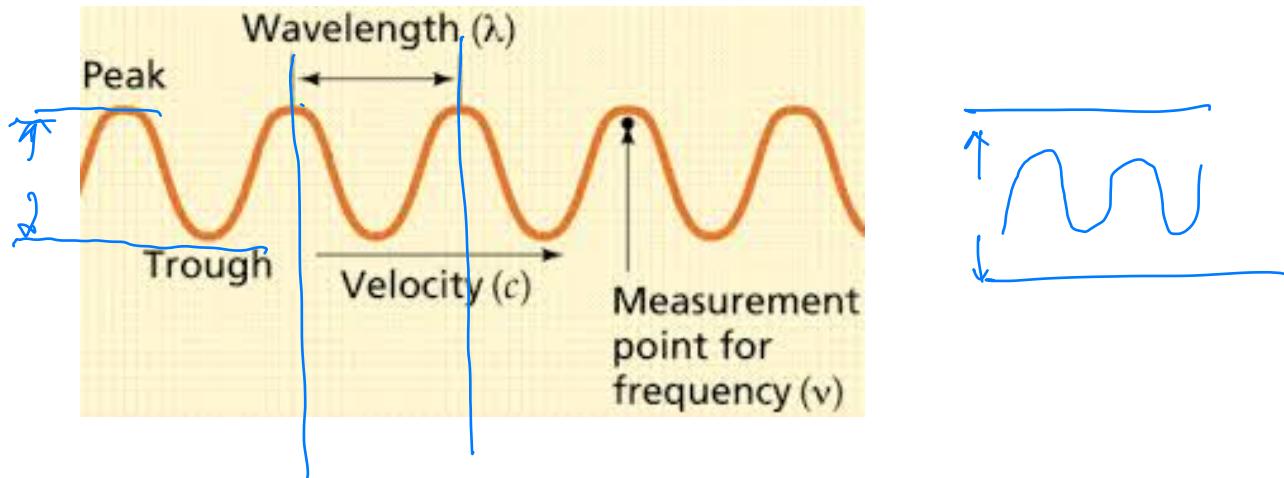
- Frequency measures how often a repeating event occurs.
- Measuring it is accomplished by counting the number of times that event occurs within a specific time period, then dividing the count by the length of the time period.
- *Frequency f* (measured in *Hz*) is the number of occurrences of a repeating event per unit time.
- The *period T* (measured in *sec*) is the duration of one cycle in a repeating event, so the period is the reciprocal of the frequency.
- Therefore we have the basic relation between period and frequency

$$T = \frac{1}{f}$$



Wavelength

- *Wavelength* of a sinusoidal wave is the spatial period of the wave, i.e., the distance over which the wave's shape repeats.



- From physics we know

$$\lambda = vT = v \frac{1}{f}$$

$$\lambda = vT$$

$$\text{Speed} = \frac{\text{distance}}{\text{Time}}$$

- Thus, for periodic waves, frequency has an inverse relationship to the concept of wavelength; simply, frequency is inversely proportional to *wavelength* λ (lambda).

Frequency for Periodic Waves

- The frequency f is equal to the phase velocity v of the wave divided by the wavelength λ of the wave:

$$f = \frac{v}{\lambda}.$$

$$\lambda = \frac{v}{f}$$

- In the special case of electromagnetic waves moving through a vacuum, then $v = c$, where c is the speed of light in a vacuum, and this expression becomes:

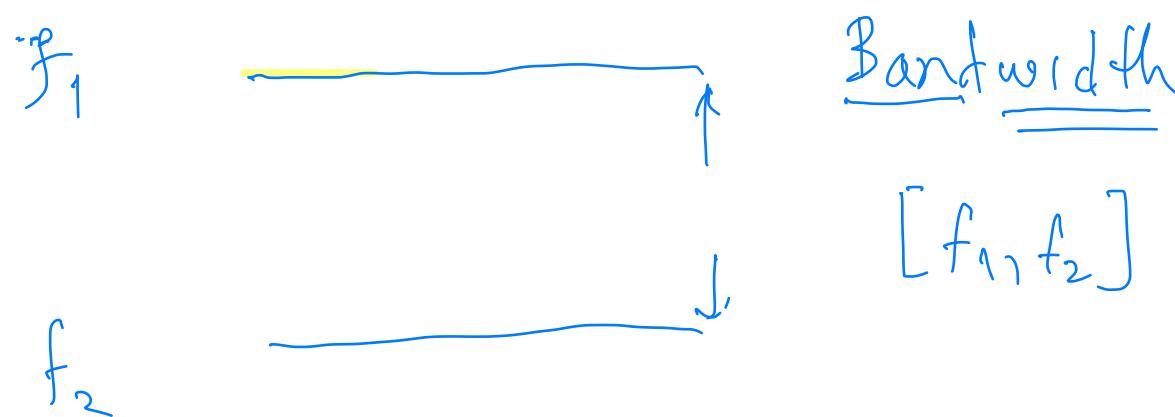
$$f = \frac{c}{\lambda}.$$

speed of light

- Note the interdependence between wavelength, frequency, and speed of medium.

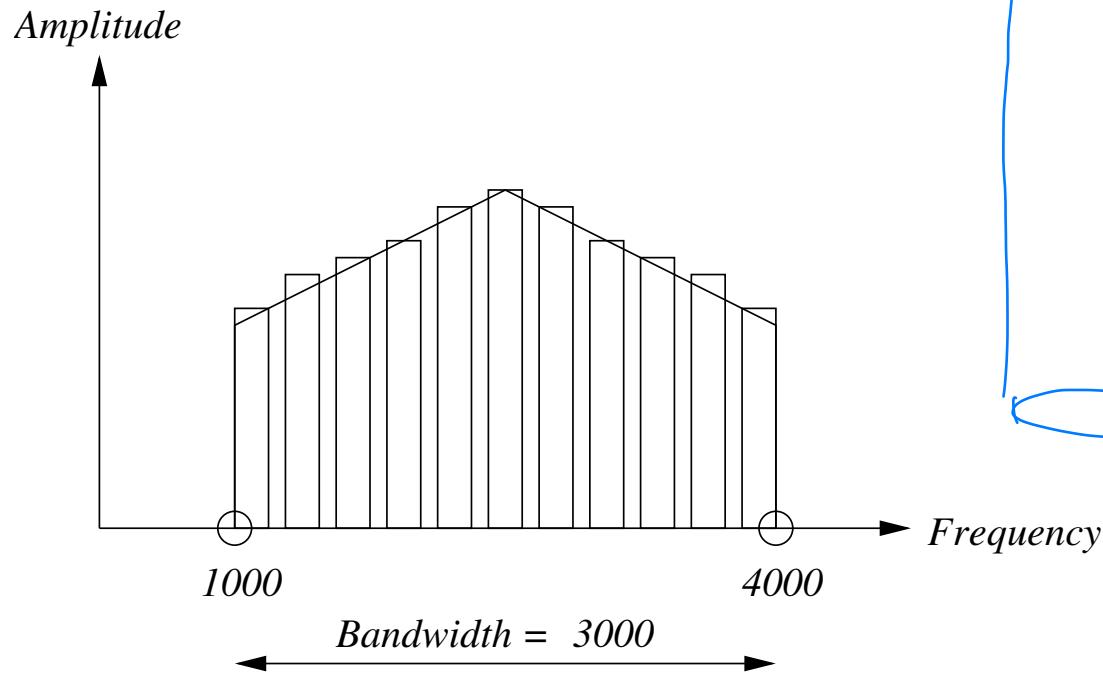
Network Performance

- Network performance is measured using two parameters
 - **Bandwidth**, and
 - **Latency** (also called **Delay**).
- Both delay and bandwidth depend on the medium.
- Intuitively, bandwidth measures “length” (but can be measured in Hz), while “delay” measures time, but there many types of delay.



Bandwidth

- Bandwidth is a property of the medium.



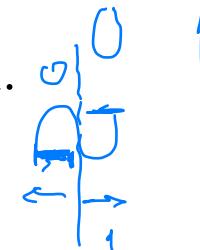
- **Bandwidth** is the difference between the highest and lowest frequencies that the medium can “pass” satisfactorily.

Bandwidth: Examples

- Different bands may be used per application.
 - But you must fit the application within the selected bands!
- Voice's spectrum is between 300 and 3,300 units of frequency and so the bandwidth is 3,000.
 - Within this band you are interested to know how many bits can be transmitted per second.
- Different applications have different bandwidth requirements.
 - Instant messaging conversation: less than 1,000 bps;
 - VoIP requires 56 Kbps to sound smooth and clear;
 - Standard definition video (480p) works at 1 Mbps.
 - HD video (720p) wants around 4 Mbps, and
 - HDX video more than 7 Mbps.

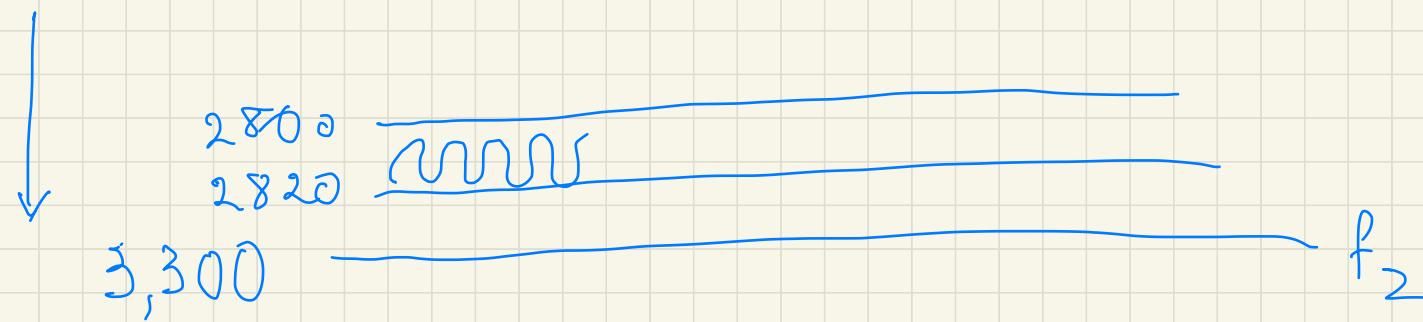
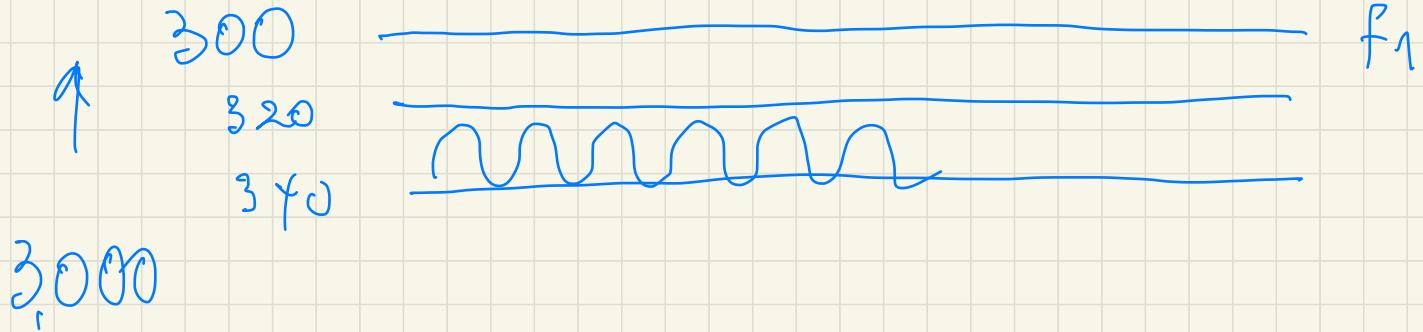
Bandwidth

- **Bandwidth** is also measured as “throughput”, i.e., in “# of bits per unit time”, when we don’t care about the actual bands being used;
 - it is a measure of the width of the frequency band.
- **Example:** the Bandwidth of a network is
 1. 100Mbps (able to deliver 100 million bits per second)
 2. 100 MBps (able to deliver 100 million bytes per second)
- Sometimes we think of bandwidth in terms of “how long it takes to transmit one bit”.
- **Example:** on 100 Mbps network it takes $0.01 \mu s$ (microseconds) to transmit a bit.



$$[f_1, f_2]$$

$[f_1, f_2]$



300



320

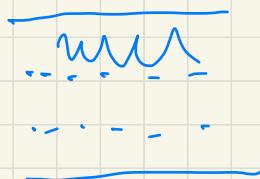


1G, 2G, 3G, 4G, 5G wireless

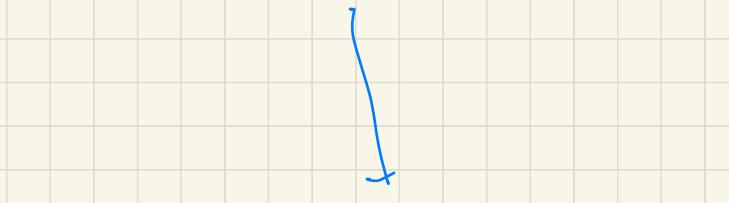
Band



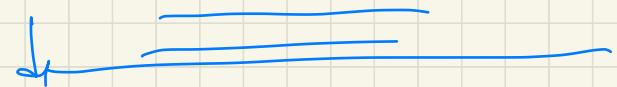
entire
band



$[f_1, f_2]$



subband



Measuring Conventions

- We use
 - b for bit, and B for Byte (8 bits).
 - K for kilo, M for Mega, G for Giga.
- In measuring quantities we assume $10^3 \approx 2^{10}$.
- We use powers of 2 for bits and powers of 10 for frequencies!
 - Depending on where it is being used K can mean either 10^3 or 2^{10} , and M can mean either 10^6 or 2^{20} , etc.
- Although measured in Mbps, bandwidth is governed by the clock speed which is *pacing the transmission*.
 - A clock is measured in Hz. So in 1 Hz we can transmit 1 bit. Therefore, 10 MHz bandwidth is the same as 10 Mbps.

bit powers of 2
 H_2 vs $\cdot 10^3$

$$2^{10} = 1,024 = 10^3$$

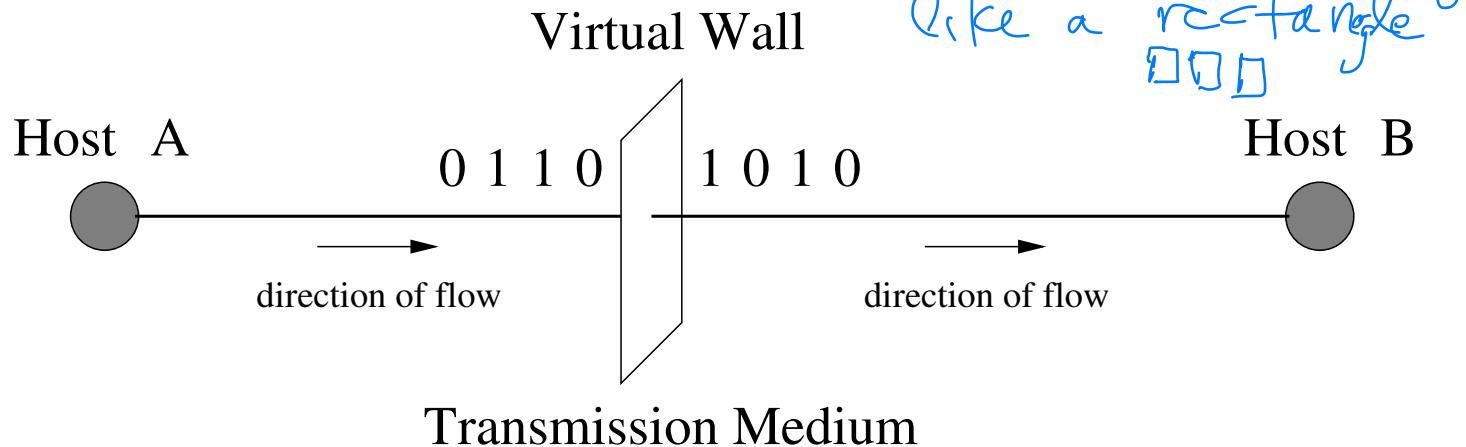
$\log_{10} = \text{Decibels}$, $\log_2 = CS$, $\log_e = \ln CS$

International System of Units (ISU)

Unit	Abbreviation	Value
pico	p	10^{-12}
nano	n	10^{-9}
micro	μ	10^{-6}
mini	m	10^{-3}
deca	da	10^1
kilo	k or K	10^3
mega	M	10^6
giga	G	10^9
tera	T	10^{12}
peta	P	10^{15}

Throughput

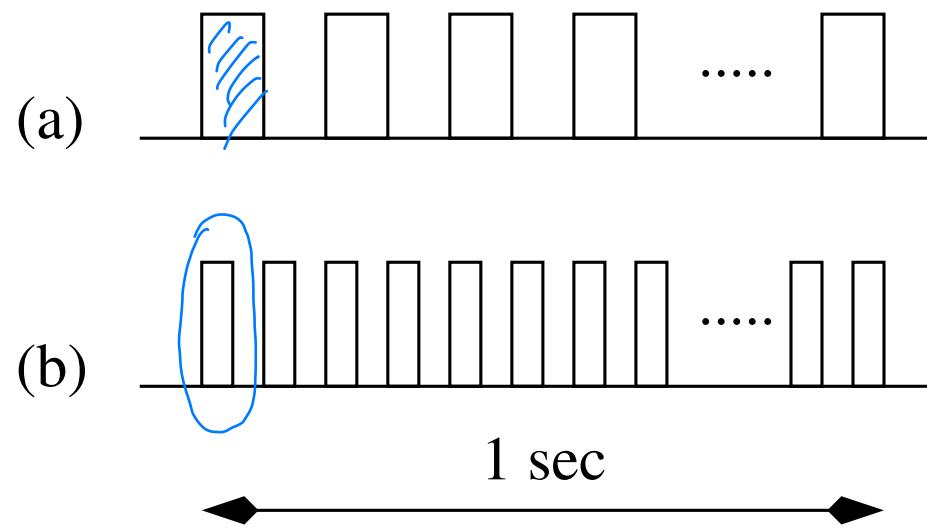
- Throughput is the measurement of how fast data can pass through a point.



- If we consider such a “measuring point” as a virtual wall then throughput is the number of bits that pass through the wall per second.
- In a way, *throughput is bandwidth that refers to measured performance.*

Distance and Time

- If you think of time as distance and the bit as a pulse of a certain width then bandwidth is how many bits fit in a unit distance.



- **Example:**
 - in picture (a) we have bits transmitted in a 1 Mbps line,
 - in picture (b) we have bits transmitted in a 2 Mbps line.

Example: Delay

- An upper bound (or best case) on latency in a medium is determined by the speed of light in that medium.
- This speed varies per medium.

Medium	Speed of Light
Vacuum	$3.0 \cdot 10^8$ m/s
Cable	$2.8 \cdot 10^8$ m/s
Fiber	$2.0 \cdot 10^8$ m/s

Technology dependent

In addition to these you have
"errors" caused by imperfections
of the medium

Common Sense Examples

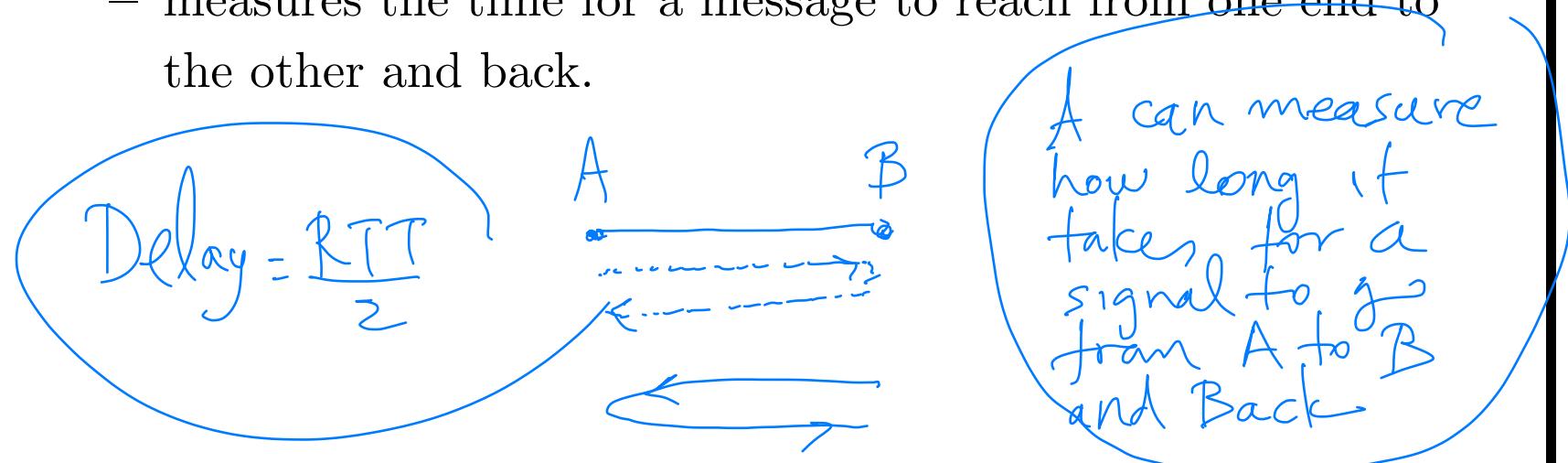
- A voice grade telephone supports a frequency band ranging from 300 Hz to $3,300 \text{ Hz}$. Its bandwidth will be

$$3,300 - 300 = 3,000 \text{ Hz}.$$

- When referring to communication links bandwidth refers to the number of bits per second that can be transmitted in that link.
 - The bandwidth of Ethernet is 10 Mbps . *wired technology*
 - This can vary per technology used. *wireless*
- Throughput is bandwidth that refers to measured performance.
- Bandwidth requirements of an application is the number of bits per second that it needs to transmit over the network in order to have acceptable performance.

Latency and RTT

- **Latency** (or **Delay**)
 - is “how long it takes a message to travel from one end of a network to another” and is measured in time units.
- E.g., the latency of a network might be 10 ms, i.e., it takes 10 ms to travel from one end to another.
- **Round Trip Time** (abbreviated RTT)
 - measures the time for a message to reach from one end to the other and back.



Bandwidth: Example

- *How many bits can a transcontinental channel hold if it has one-way latency of 60 ms and bandwidth of 50 Mbps?*
- You find the number of bits if you multiply the latency by the bandwidth.
 - Latency: $60 \text{ ms} = 60 \times 10^{-3} \text{ sec.}$
 - Bandwidth: $50 \text{ Mbps} = 50 \times 10^6 \text{ bits per sec.}$
 - Therefore we have

$$\begin{aligned}\#\text{ of bits} &= (60 \times 10^{-3} \text{ sec}) \times (50 \times 10^6 \text{ bits per sec}) \\ &= (60 \times 50) \times 10^{6-3} \text{ bits} \\ &= 3 \times 10^6 \text{ bits} \\ &= 3 \text{ Mb}\end{aligned}$$

Transmission Media: Parameters

- **Speed:** Max number of bits per sec that can be transmitted reliably.
- **Attenuation:** Tendency of a signal to become weak or distorted over distance.
 - Signal absorbed/dissipated during transmission.
- **Electromagnetic Interference (EMI):** Susceptibility of medium to external electromagnetic energy which is inadvertently introduced onto a link
 - causes “static audio” and “visual snow” in corresponding media.
- **Cost:** Materials plus installation.

Transmission Media: Comparison

Medium	Cost	Speed (in bps)	Attenuation	EMI	Security
UTP	Lo	1-100 M	Hi	Hi	Lo
STP	Mo	1-150 M	Hi	Mo	Lo
Coax	Mo	1 M-1 G	Mo	Mo	Lo
Optical Fiber	Hi	10 M-2 G	Lo	Lo	Hi
Radio	Mo	1-10 M	Lo-Hi	Hi	Lo
Microwave	Hi	1 M-10 G	Va	Hi	Mo
Satellite	Hi	1 M-10 G	Va	Hi	Mo
Cellular	Hi	9.6-19.2 K	Lo	Mo	Lo

Hi = High, Mo = Moderate, Va = Variable, Lo = Low

UTP = Unshielded Twisted Pair, STP = Shielded Twisted Pair

How Long Does it Take?

- The time it takes to transmit a unit of data in a network depends on the network bandwidth and data-unit (or packet) size.
- There are also Queuing delays in a network due to buffering, and switching.

$$\text{Delay} = \text{Propagation} + \text{Transmit} + \text{Queue}$$

$$\text{Propagation-Delay} = \frac{\text{Distance}}{\text{Speed of Light}}$$

$$\text{Transmit-Delay} = \frac{\text{Packet Size}}{\text{Bandwidth}}$$

Distance
Sp. of. Light in medium

- Bandwidth and Latency define and dominate the performance characteristics of a network.

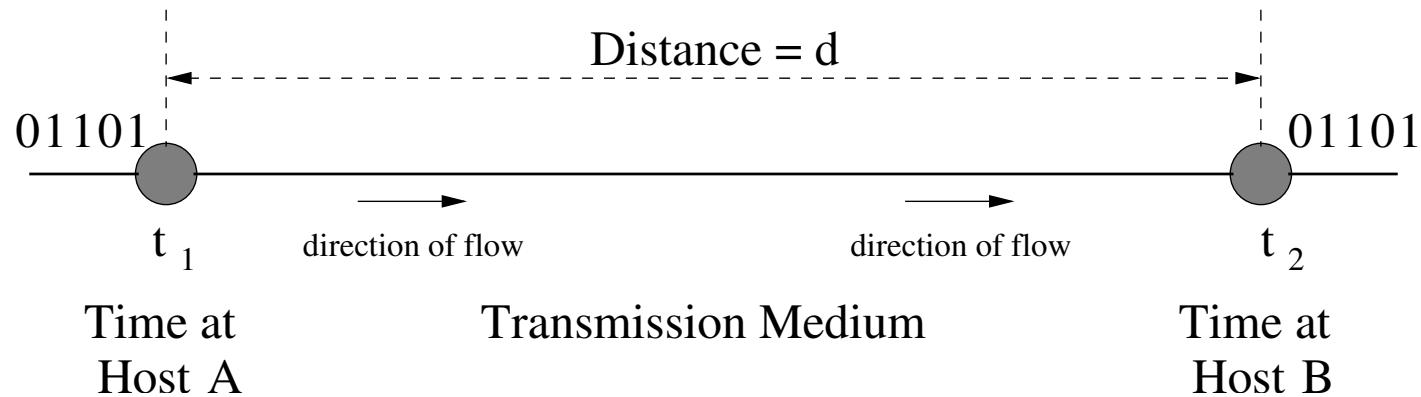
$$\frac{P}{B} \gg \frac{P}{kB}$$

$$TD = \frac{P}{B} = \frac{kP}{kB}$$

Propagation Speed/Time

- **Propagation speed**

- measures “the distance a bit can travel in one second”.



- **Propagation time**

- measures the time it takes for a bit to travel from one end to another.

- From physics we know:

$$\text{Distance} = \text{Propagation speed} \times \text{Propagation time}$$

Examples

- The propagation time (normalized in kilometers) for Twisted Pair is

$$\begin{aligned}\text{Propagation time} &= 1000 \text{ m}/(2 \times 10^8 \text{ m/s}) \\ &= 5 \times 10^{-6} \text{ s} \\ &= 5 \mu\text{s}\end{aligned}$$

- The propagation time (normalized in kilometers) for Coaxial or Fiber Optic Cable is

$$\begin{aligned}\text{Propagation time} &= 1000 \text{ m}/(3 \times 10^8 \text{ m/s}) \\ &= 3.33 \times 10^{-6} \text{ s} \\ &= 3.33 \mu\text{s}\end{aligned}$$

Latency and Bandwidth

- When combined they “determine” network performance.
- Consider a client and a server that exchange messages.
- Is there a (noticeable) difference between transmitting across
 - the room with 1 *ms* RTT, and
 - a transcontinental channel with 100 *ms* RTT?
- Which of the two is more important:
 - Latency?
 - Bandwidth?
- Their importance is relative and depends on the application.

$$\frac{P}{B}$$

Latency may Dominate Bandwidth: A Keystroke

- Consider the following application:
 - A client sends a keystroke (this is 1 Byte) to a server and receives back a 1 Byte message.
- The application performs differently across the room than across the transcontinental channel.
 - Whether the channel is 1 $Mbps$ or 100 $Mbps$ is not very relevant:
 - (a) in a 1 $Mbps$ channel this takes $8 \mu s$, while
 - (b) in a 100 $Mbps$ channel it takes $0.08 \mu s$.
- Here bandwidth is insignificant!

Bandwidth may Dominate Latency: An Archive

- Consider the following application:
 - A client is downloading a $25\ MB$ archive from a digital library.
- In this case the more the bandwidth available the faster it will return the archive to the client.
- It takes

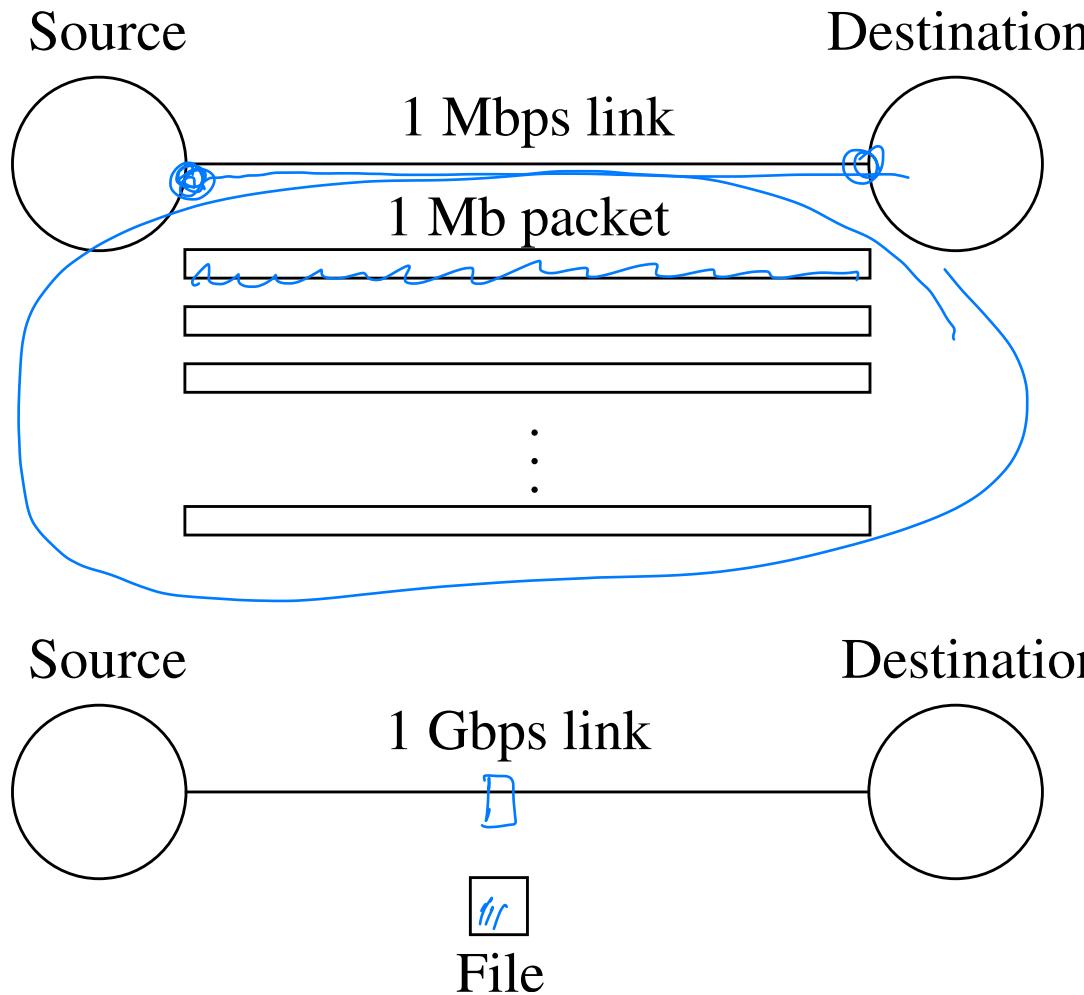
$$\frac{25\ MB}{10\ Mbs} = \frac{200\ Mb}{10\ Mbs} = 20\ sec$$

to transmit this in a channel with bandwidth $10\ Mbps$.

- It is not very important now on whether the archive is located across the room or across the continent.
 - In the former case (a) the response time is $20.001\ sec$ and in the latter case (b) $20.1\ sec$.

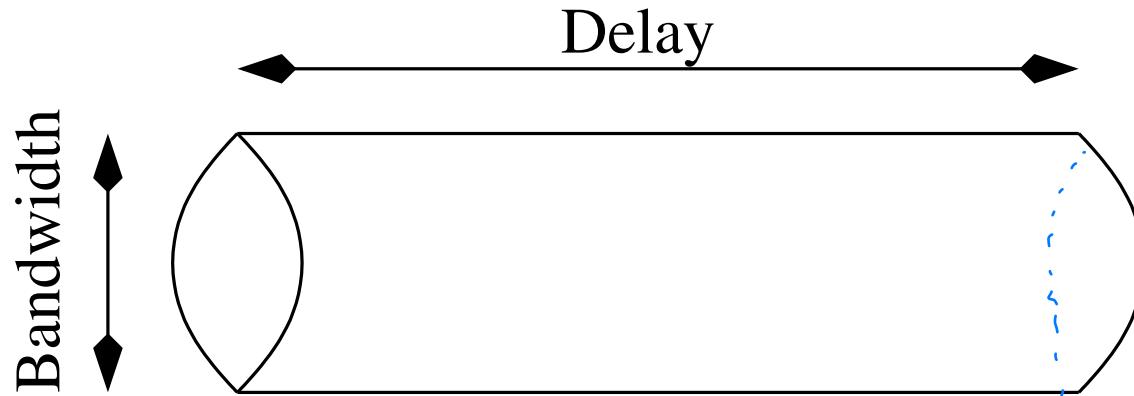
Impact of Speed: An Intuitive Comparison

A 1 MB file over a 1 Mbps- and 1 Gbps-network.



“Delay \times Bandwidth” Product

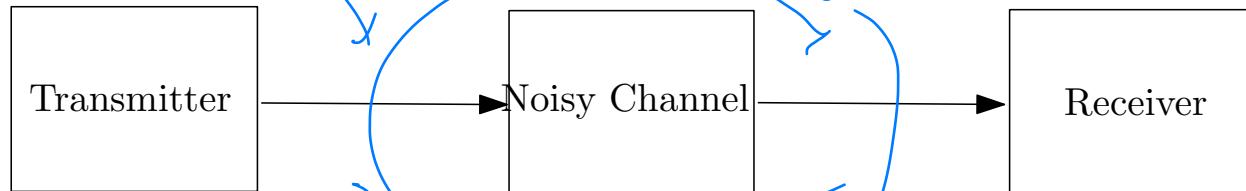
- Delay is measured in “time units”
- Bandwidth in “bits per time-unit”.
- The Delay-Bandwidth Product is measured in bits.



- Delay-Bandwidth Product corresponds to the number of bits the sender must send before the first bit arrives at the receiver.
- Also the sender will send $2 \cdot (\text{Delay} \times \text{Bandwidth})$ bits of data before it receives a bit from the receiver!

Channel Capacity

- Data are transmitted through channels



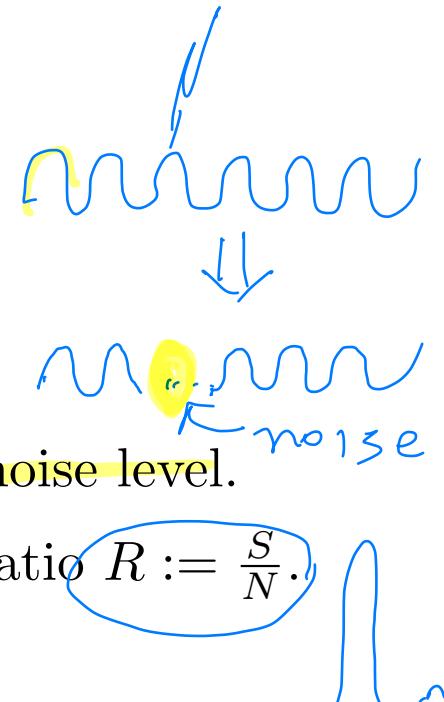
...which can be imperfect!

- Quality affected by
 - bandwidth,
 - signal strength,
 - noise level.

- Let S = be the signal strength and N = noise level.

– This gives rise to the Signal to Noise ratio $R := \frac{S}{N}$.

- Let the bandwidth be B .



Shannon Capacity

- What is the maximum capacity in bits per sec of the channel?
- **Shannon Capacity** C measures precisely this theoretically highest maximum:

If $\frac{S}{N}$ is measured in dB then the log is in base 10

$$C = B \log_2 \left(1 + \frac{S}{N} \right) \text{ in bits per sec.}$$

per unit of bandwidth.

$\log_2 \left(1 + \frac{S}{N} \right)$
per unit of frequency

- This is a mathematically sophisticated result!
 - You may wonder where does the log come from?
- An extremely noisy channel will have Signal to Noise ratio R close to “zero”:
 - this is because $\log(1 + R)$, and hence also the capacity C is close to “zero”.

$$R = \frac{S}{N} \text{ of talking in a room} \quad \parallel \quad \frac{kS}{kN} \text{ of talking in a highway}$$

$k=20$

Shannon Capacity: Example

- A typical telephone line has bandwidth $3,000 \text{ Hz}$ and signal to noise ratio of $SN = 3,162$ (usually measured in dB).

- Hence,

$$\begin{aligned} C &= B \log(1 + \frac{S}{N}) \\ &= 3000 \times \log(1 + 3162) \\ &= 34860 \text{ bps} \end{aligned}$$

- Hence, 34860 bps is the highest bit rate of a telephone line.

Where does
the \log
come from

\log_2

Information

- Justification for Shannon's capacity is based on a “very basic” but “subtle” question concerning “information”.
- Given a random experiment with n equiprobable outcomes E_1, E_2, \dots, E_n (each may happen with probability p) how much information is conveyed on the average by a message M telling us which of these events occurred?
- A reasonable measure of this would be “the length of the message M ” however written in an economical way!
 - Coin (H, T): one bit 0, 1
 - Dice (1, 2, 3, 4, 5, 6): three bits 000, 001, 010, 011, 100, 101.
- Each outcome E_i will be encoded as a bit sequence in binary

$$b_1^i b_2^i \cdots b_\ell^i$$


Example

- Consider the outcomes of rolling a die:
 - each side occurring with probability $p = 1/6$.
- Such a sequence of outcomes would be E_1, E_2, \dots, E_6

$$\text{“1”, “2”, “3”, “4”, “5”, “6”} \quad \frac{1}{6} = 6$$

- Observe that we have $n = \frac{1}{p} = 6$ possible outcomes!
- Notice that you need 3 bits to encode the outcome, and
 - 3 is the smallest exponent ℓ such that $6 \leq 2^\ell$.

Shannon Capacity and Information

*ceil*ing

- Observe that in the general setting you have $n = \lceil \frac{1}{p} \rceil$ events.
- If we use binary codes of length ℓ to encode this message we would need to choose ℓ so that $n \leq 2^\ell$.
- The \min such value of ℓ must satisfy

$$0 \leq \ell - \log_2 n < 1$$

Why is the above true?

$$2^{\ell-1} < n \leq 2^\ell$$

$$\ell - 1 < \log_2 n \leq \ell$$

- The quantity

$$I := \log_2 \left(\frac{1}{p} \right) (= \log_2 n)$$

is a reasonable measure of the average amount of information in the message M telling us which of these events occurred.

Explanations on Shannon Capacity: About $\log_2(1 + \frac{S}{N})$

Assume a channel of unit bandwidth: $B = 1$.

- For $N = \text{noise}$, $S = \text{signal}$ uniformly random, $p = \frac{N}{S+N}$ will be the probability of “noise”.
- Now look at the logarithm

$$\log_2 \left(1 + \frac{S}{N} \right) = \log_2 \frac{S + N}{N} = \log_2 \frac{1}{\frac{N}{S+N}} = \log_2 \left(\frac{1}{p} \right).$$

- Set $p := \frac{N}{S+N}$: assume $N = 1$
 - if $S = 1$ then $p = 1/2$
 - if $S = 3$ then $p = 1/4$
 - if $S = 2^k - 1$ then $p = 1/2^k$
- So, $\log_2 (1 + \frac{S}{N})$ represents the average amount of “bits” (or information) per unit bandwidth that can be transmitted in such a channel.

UNIX ping command

- **ping**
utilizes the ICMP protocol's ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from the specified host or network gateway.
- **ping -s hostname**
When the -s flag is specified, ping sends one datagram per second

ICMP

- **ICMP** (Internet Control Message Protocol) is the error and control message protocol used by the Internet protocol family. It is used by the kernel to handle and report errors in protocol processing.
- ICMP is a datagram protocol layered above IP. It is used internally by the protocol code for various purposes including routing, fault isolation, and congestion control.
- Receipt of an ICMP “redirec” message will add a new entry in the routing table, or modify an existing one.

TRANSMISSION PRINCIPLES

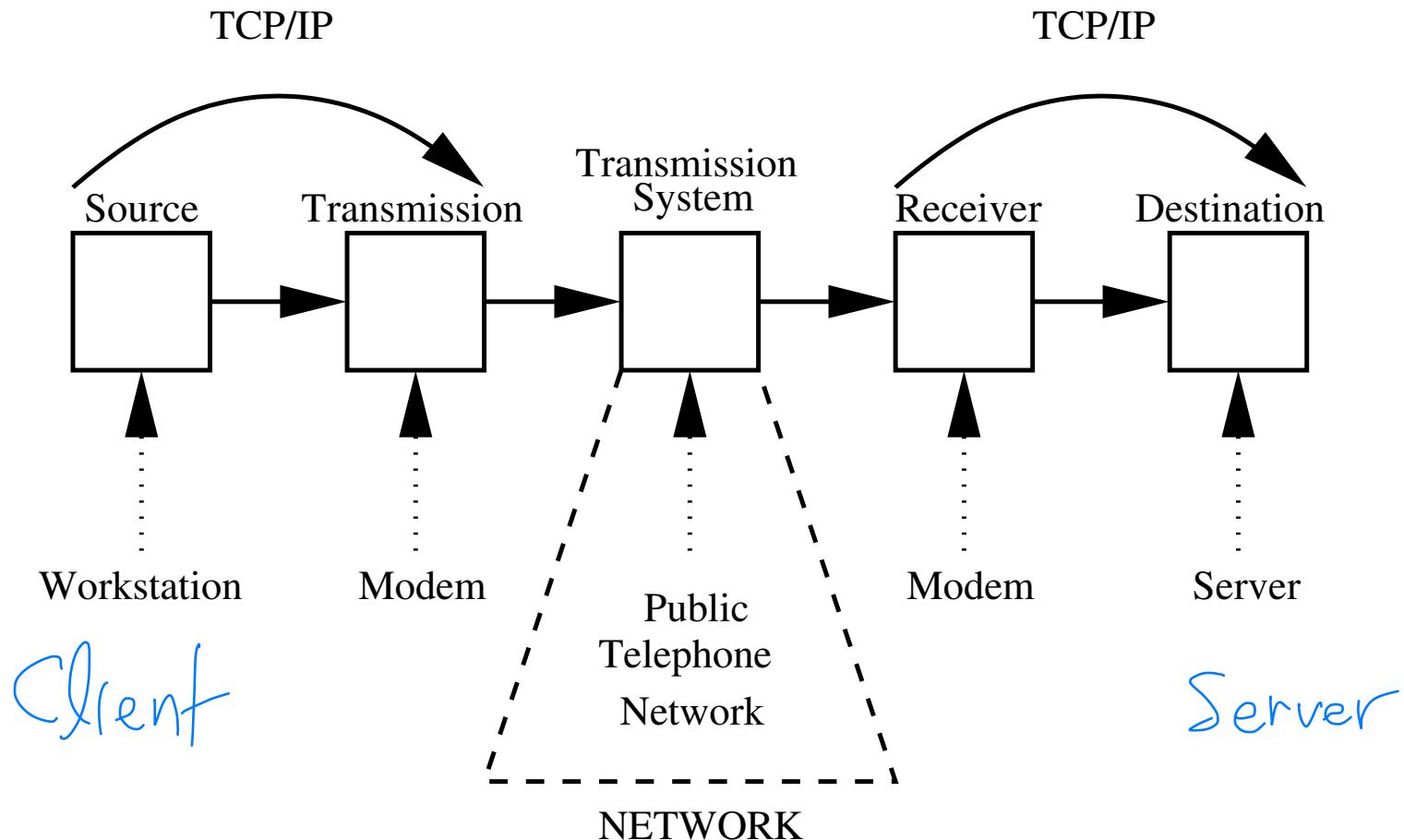
Outline

- Transmission
 - Conversions
 - Digital-to-Digital
 - Analog-to-Digital
 - Digital-to-Analog
 - Analog-to-Analog
 - Media
 - Appendix (Not Required)
- .

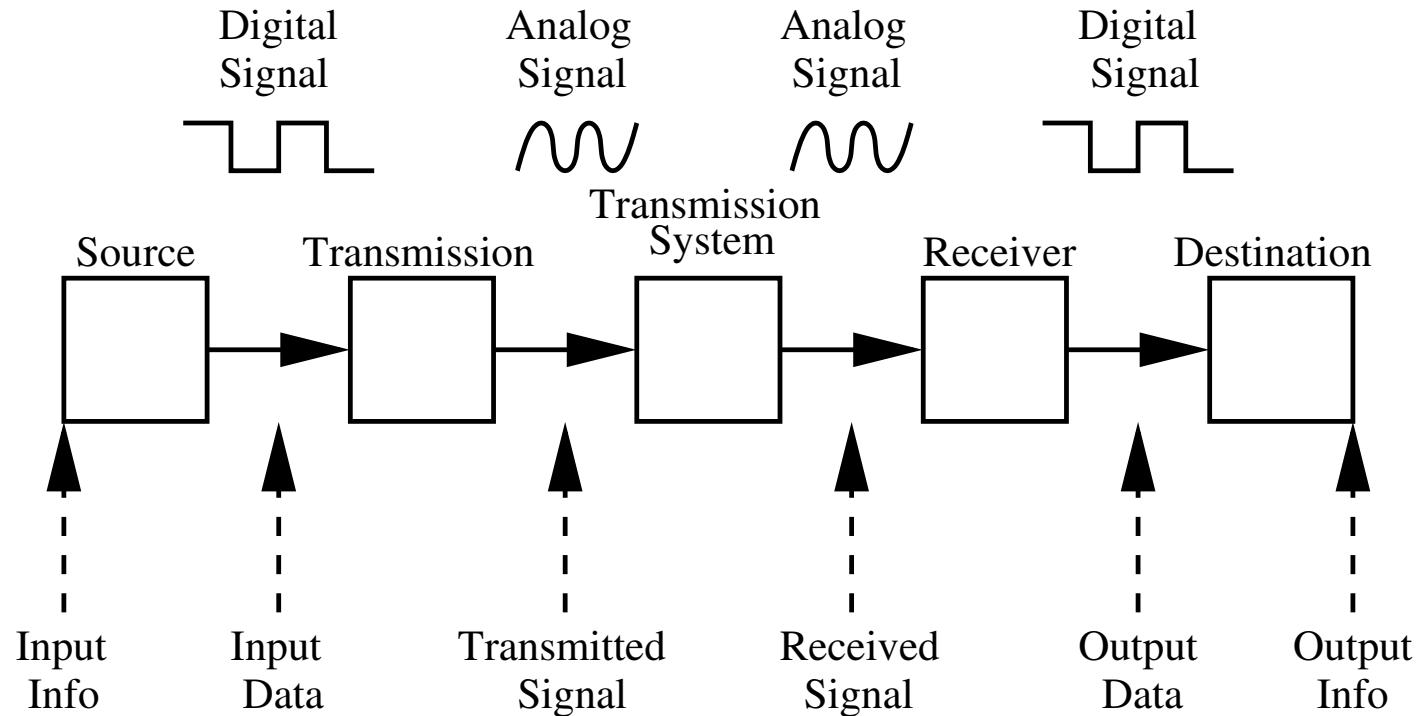
Computer-to-Computer

Digital-to-Digital

The Network Transmission Process



Transitions: Analog ↔ Digital



The network infrastructure

not only physical
but also protocols)

enables quality of service QoS

Encoding and Modulation

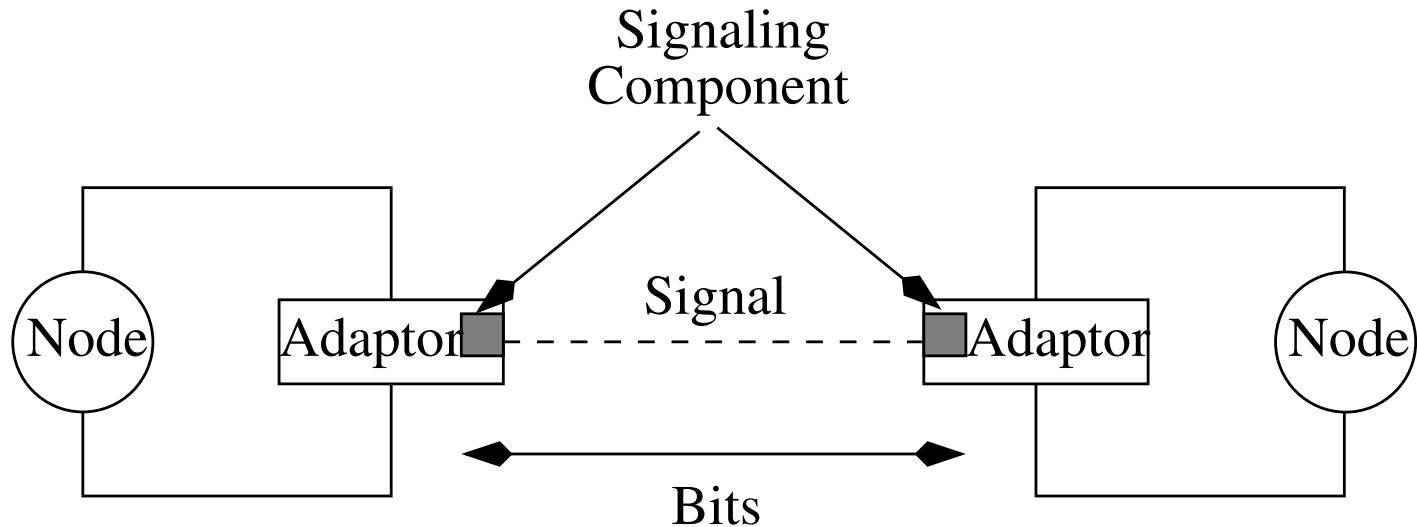
- **Digital-to-Digital Conversion:** Computer data are stored as 0s and 1s. To be carried from one place to the other (inside or outside the computer) they must be transformed from one representation to another (e.g. Binary Data to Gray Data).
- **Analog-to-Digital Conversion:** An analog signal (like voice) must be converted to digital (to decrease noise effects).
- **Digital-to-Analog Conversion:** A digital signal must be often converted to analog to carry it over a medium.
- **Analog-to-Analog Conversion:** This is required for multiplexing purposes, for carrying a signal over long distances, etc.

NB: Each of these “Conversions” has different signal representations, which we look at in the sequel.

Digital → Digital

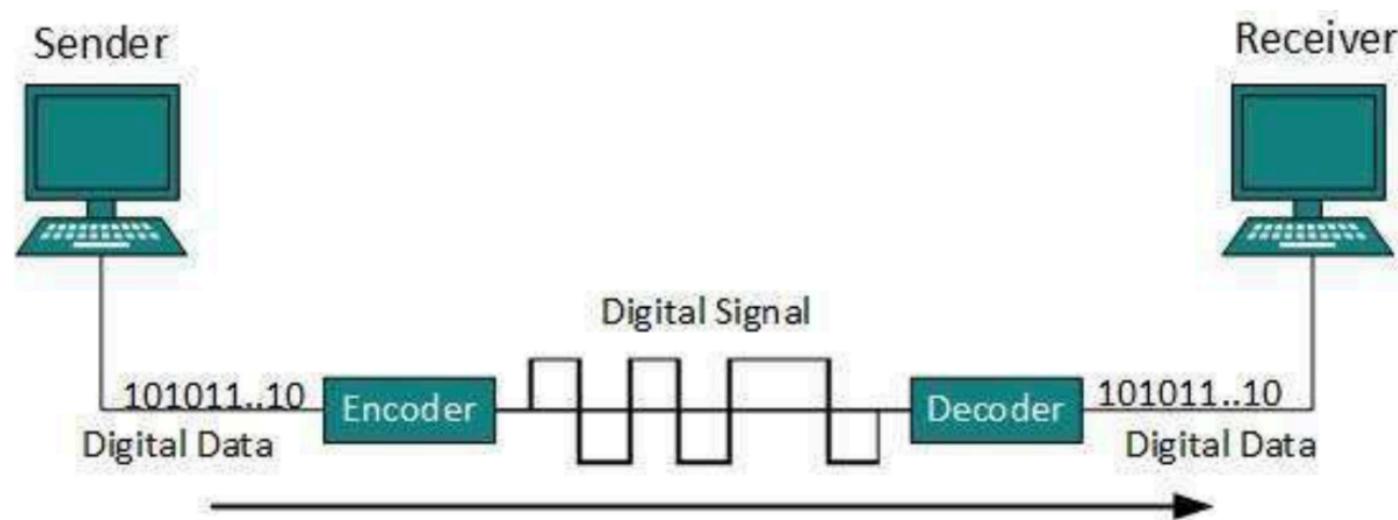
Sender and Receiver

- A signaling component is required to translate digital sequences between sender and receiver within a computer.



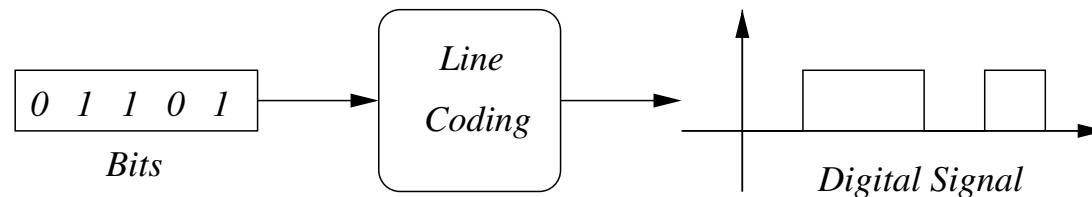
Line Coding

- Digital-to-digital encoding is the representation of digital information by a digital signal. When binary 1s and 0s generated by the computer are translated into a sequence of voltage pulses that can be propagated over a wire, this process is known as digital-to-digital encoding
- Line coding is the process of converting binary data (i.e., a sequence of bits) to a digital signal.

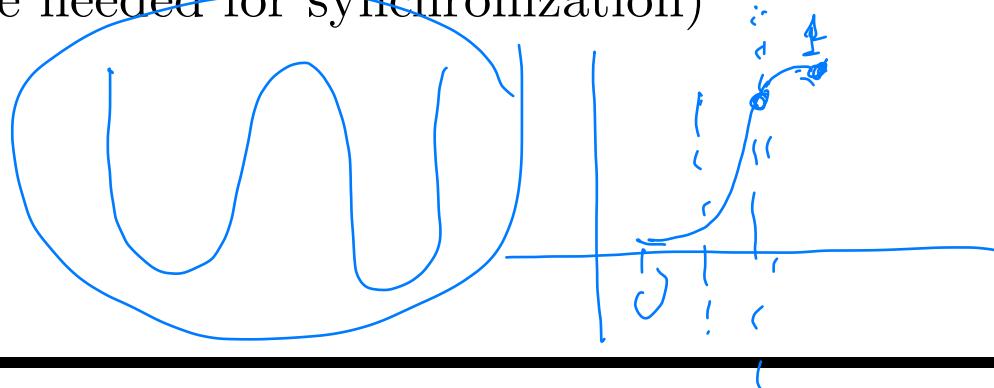


Line Coding

- Signals transmitted at various “levels” and “pulse rates” depending on the quality of the line to help differentiate.

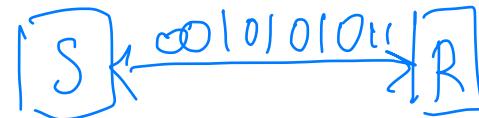


- Important issues taken into account include:
 - What should the signal level be? (Transmission Amplitude)
 - What should the pulse rate be? (How often you transmit)
 - How about (self-)synchronization between sender and receiver? (Change needed for synchronization)



On the Need for Encodings

- **Baseline wander:**
 - The receiver takes the average of what it has seen so far to distinguish between high and low!
 - Thus a long sequence of consecutive 1s (or 0s) changes this average and is hard to distinguish 0 and 1
- **Clock Recovery:**
 - Coding and decoding processes are driven by (separate) clocks which must be synchronized.
 - Frequent transitions from 0 to 1 and vice versa are necessary in order to enable sender and receiver to synchronize.
- This indicates a “need for change” while transmitting!

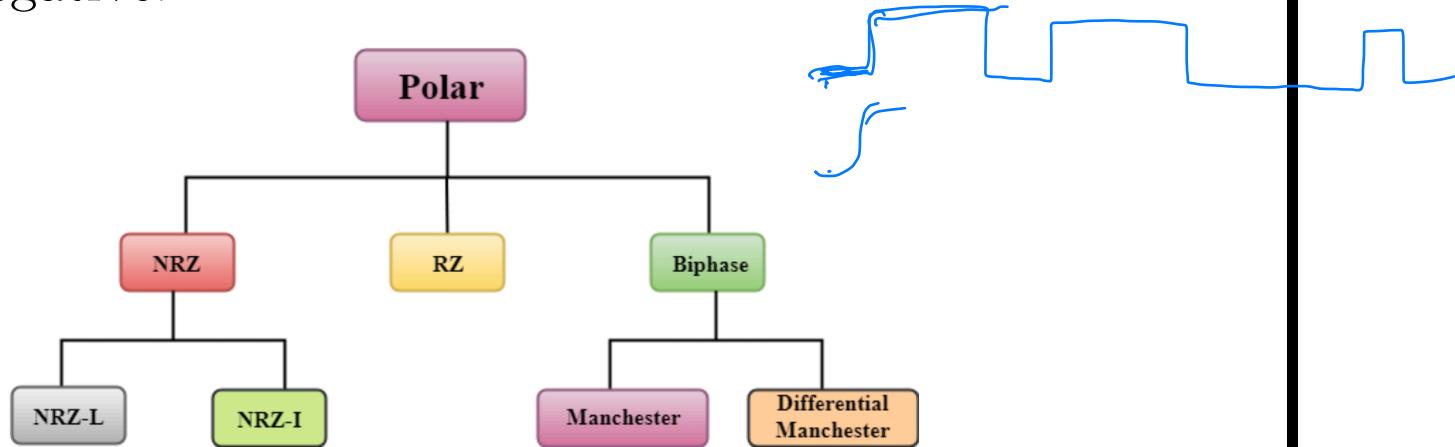


Parameters to Consider

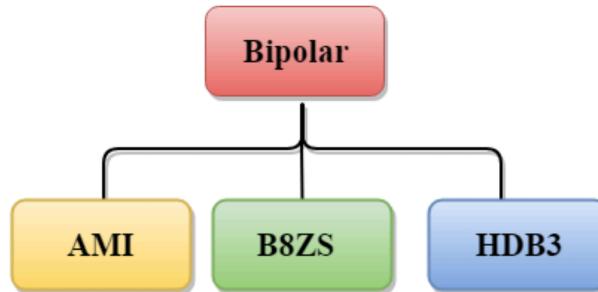
- Signals represented by voltages.
- What voltage is “high” and what voltage is “low”?
- How do you represent “high → low” and “low” → “high”?
- How often do you make transitions?

Polar and Bipolar Encodings

- Polar encodings use two voltage levels: one is positive, and another is negative.



- Bipolar encodings use three voltage levels: positive, negative, and zero



Two Additional Solutions

- 4B/5B
- Bit-Stuffing

4B/5B

- For every 4 bits of data a 5th bit is inserted (and transmitted) so that the resulting 5 bit sequence has no more than one leading 0 and no more than two trailing 0s.
- Thus, two such consecutive 5 bit sequences have no sequence of three consecutive 0s. The result is then transmitted with NRZ-I.
- **Example:** Arrange bits in groups of four and use the table

0011	1010	1111
↓	↓	↓
10101	10110	11101

Now you can apply a polar or bipolar encoding technique (e.e., NRZ-I) to the result.

Table: 4B/5B

4-Bit Data	5-Bit Code	4-Bit Data	5-Bit Code
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
0111	01111	1111	11101

Bit-Stuffing

- **Bit Stuffing**
 1. Fix a value of k (usually $k = 6$).
 2. Scan string left-to-right.
 3. Insert 1 (respectively, 0) after an occurrence of k consecutive 0s (respectively, 1s).
- **Example:** For $k = 4$, in the string

001110101111

we need only to insert a bit 0 at the end.

- **Mathematical Question:** For n random bits, what is the expected number of stuffed bits?

Bit stuffing $k=8$

1001100000000010100111111

t

1

n bits

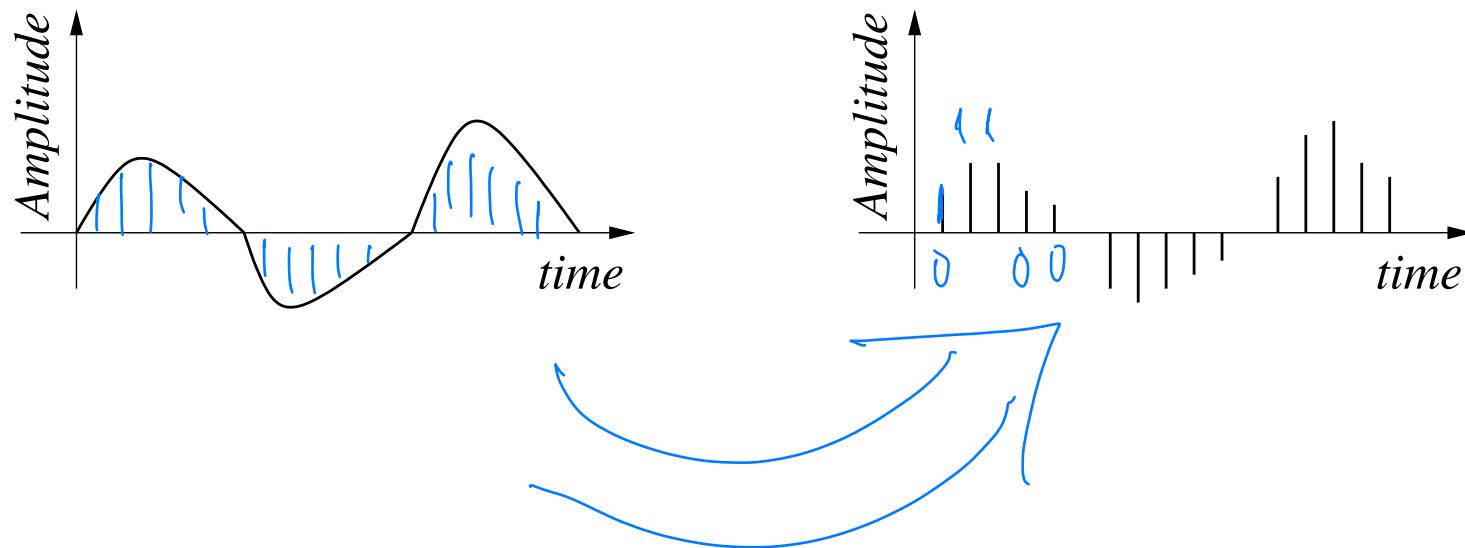
|-----|

$$n + n \cdot 2^{-k}$$

Analog → Digital

Analog-to-Digital: From Waveforms to Bits

- This requires reduction of an “infinite number” of values in an analog message to a digital stream with min loss of information.
- Analog signal converted to digital by sampling.
- In **Pulse Amplitude Modulation (PAM)** you sample the amplitude.



Sampling Rate

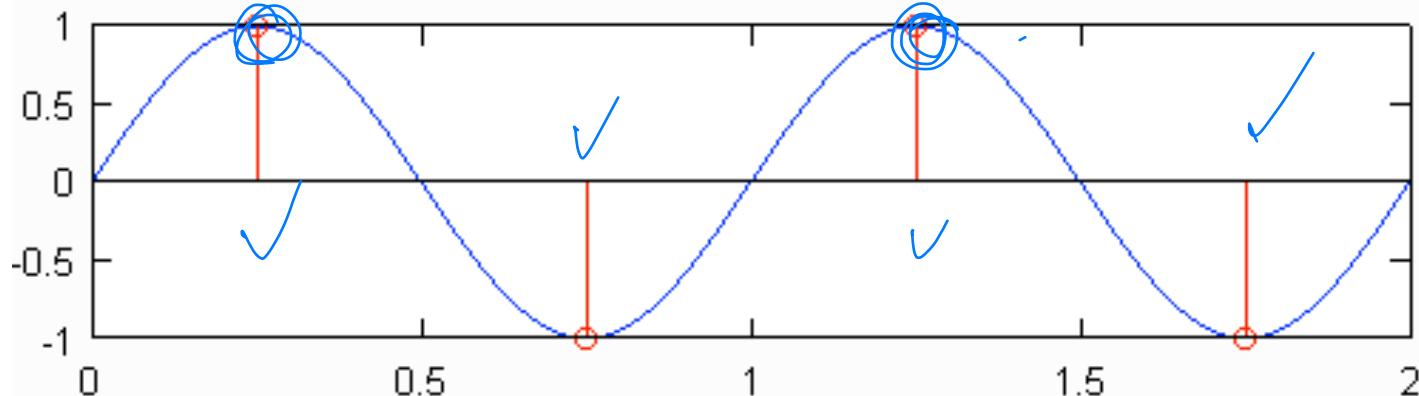
- **Sampling rate** plays an important role in the quality of the output.
- The more frequent the samples the more accurate the signal.
- **Main Question:**

What is the tradeoff between “sampling frequency” and “output accuracy”?

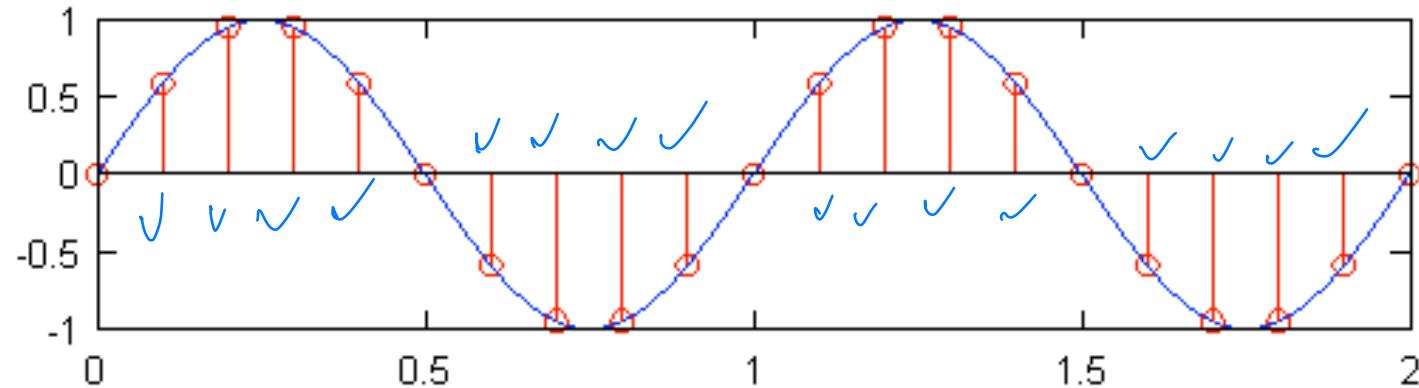
- Consider a noise-free channel.
- In this setting, the limitation on data rate is the bandwidth of the signal.

Nyquist Limits (1/2)

- Sampling at the limit (2 times the max frequency of the signal)

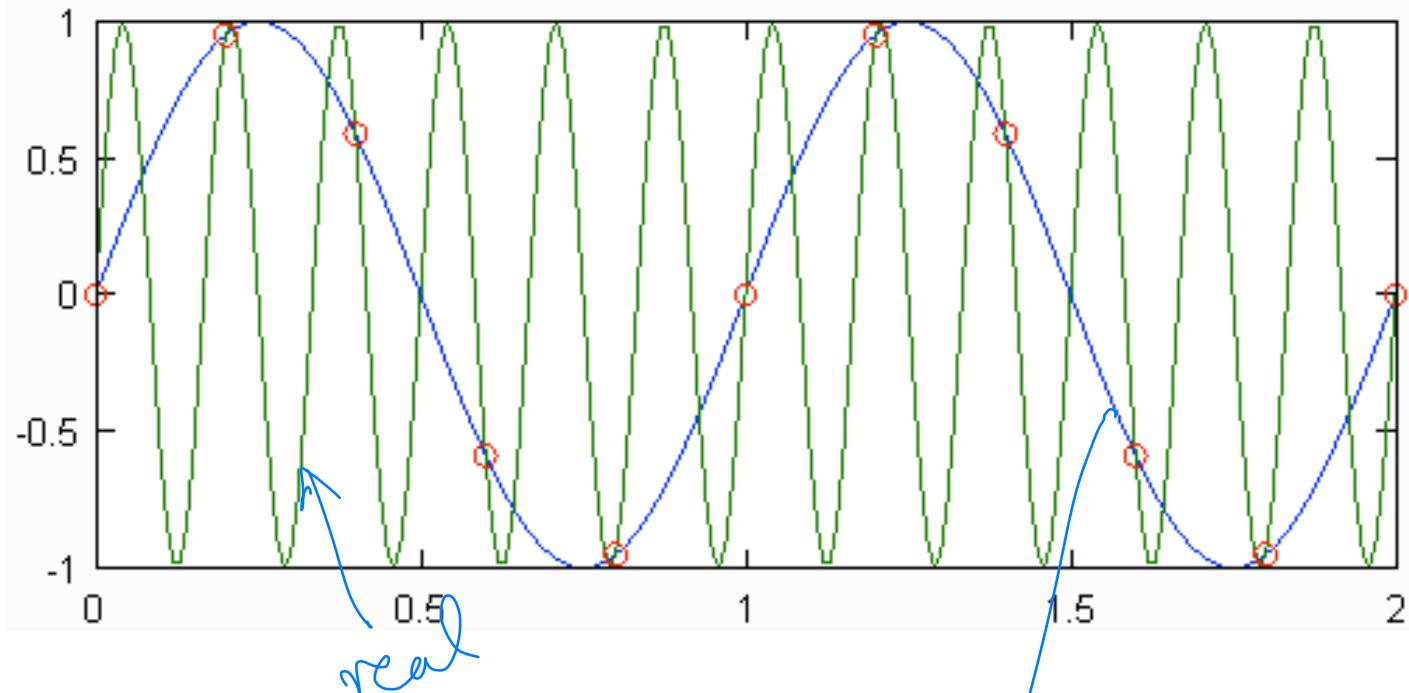


- Sampling above the limit (oversampling)



Nyquist Limits (2/2)

- Sampling below the limit (under sampling)



- Sampling is inadequate!

blue
sampling

Nyquist Limits (or Nyquist Frequencies)

- **Nyquist Theorem:**
 - If the rate of signal transmission is $2B$, then a signal with frequencies no greater than B is sufficient to carry the signal rate.
 - Also conversely: Given bandwidth B , the highest signal rate that can be carried is $2B$.
- This limitation is due to the effect of inter-symbol interference, such as is produced by delay distortion.

Nyquist Limits (or Nyquist Frequencies): Example

- To sample telephone voice with max frequency $4,000 \text{ Hz}$ we must sample at the rate $8,000$ per sec.
- If transmitted signals are binary (two voltage levels), then the data rate that can be supported by $B \text{ Hz}$ is $2B \text{ bps}$.
- However, we can also use signals with more than two levels: each signal element can represent more than one bit.

Nyquist Limits: Limitations

- For example,
 - if four possible voltage levels are used as signals, then each signal element can represent two bits.
 - With multilevel signaling, the Nyquist formulation becomes

$$C = 2B \log_2 M,$$

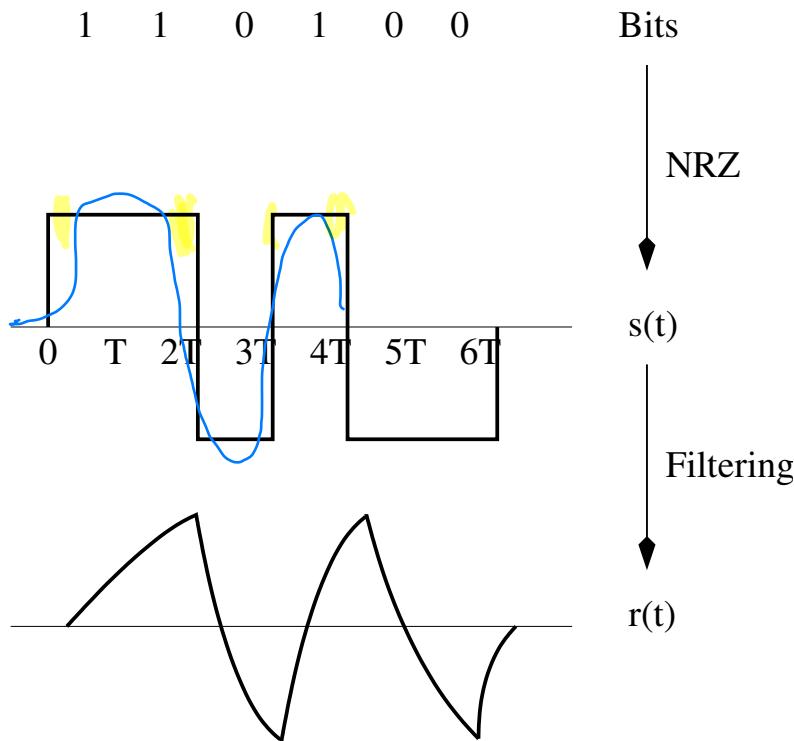
where M is the number of discrete signal or voltage levels.

- For a given bandwidth, the data rate can be increased by increasing the number of different signal elements.
- This increases burden of the receiver: Instead of distinguishing one of two possible signal elements during each signal time, it must distinguish one of M possible signal elements.
- Noise and other impairments on the transmission line will limit the practical value of M .

Digital → Analog

Digital-to-Analog: From Bits to Waveforms

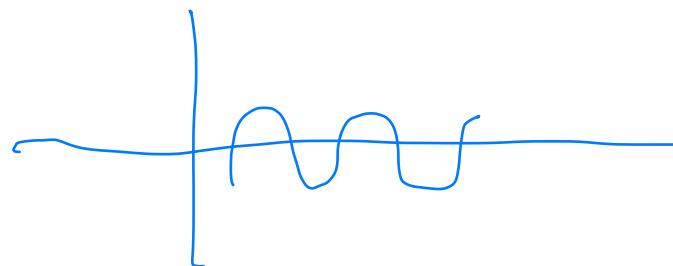
- Converts bits to signals and then filtering takes place.



- Usually: 1 goes to high voltage and 0 to low voltage.

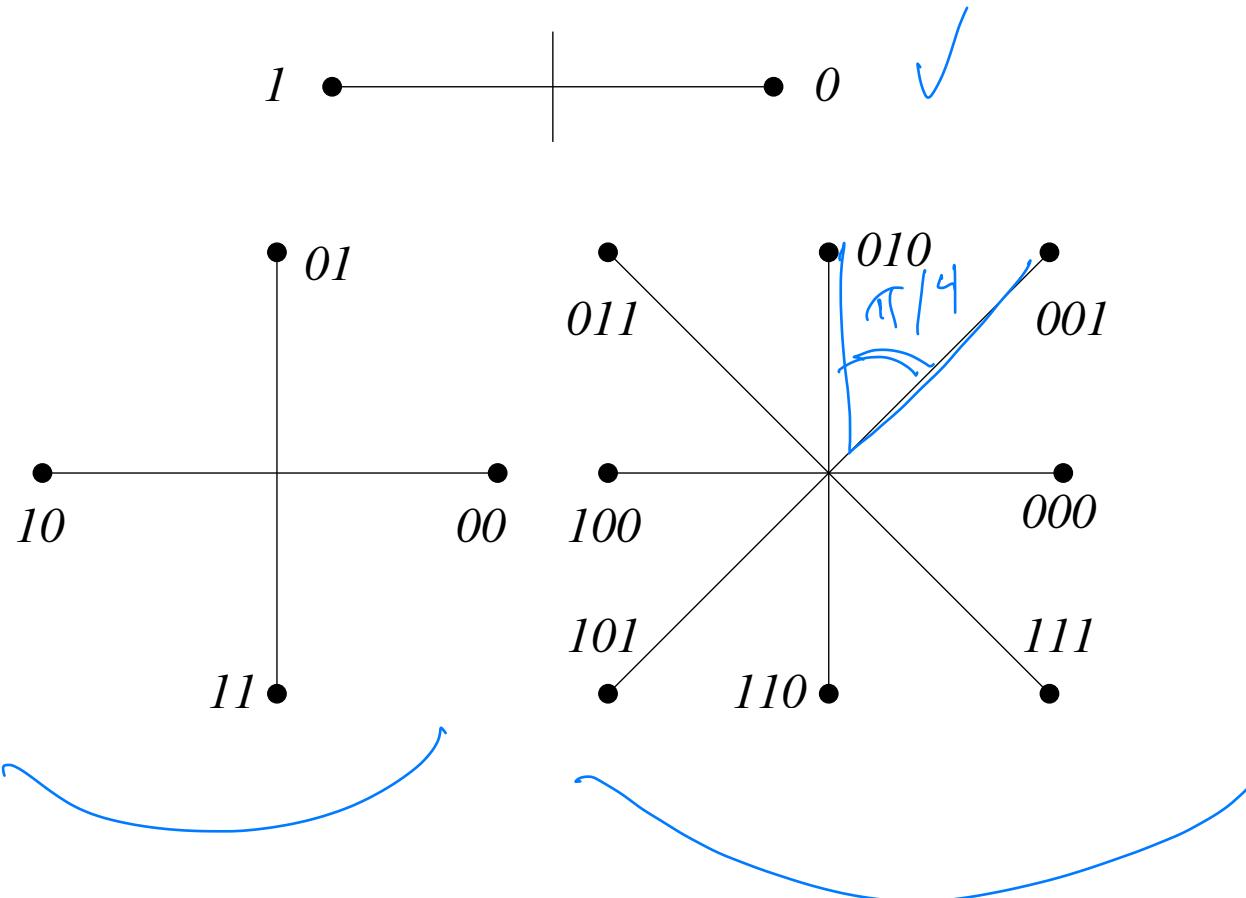
Digital-to-Analog Methods

- Waves are represented as combinations of sin, cos functions and as such are characterized by amplitude, frequency, and phase.
 - **Amplitude Shift Keying (ASK):** You vary the strength of the signal to represent binary 0s and 1s.
 - **Frequency Shift Keying (FSK):** You vary the frequency of the signal to represent binary 0s and 1s.
 - **Phase Shift Keying (PSK):** You vary the phase of the signal to represent binary 0s and 1s.
 - **Quadratic Amplitude Modulation (QAM):** Combines ASK and PSK in order to maximize contrast between bits.



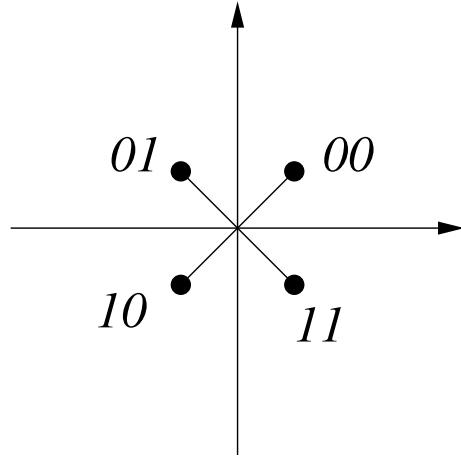
PSK: Examples

- Examples of PSK with (1) 0, 180, (2) 0, 90, 180, 270, and (3) 0, 45, 90, 135, 180, 225, 270, 315 degrees, respectively.

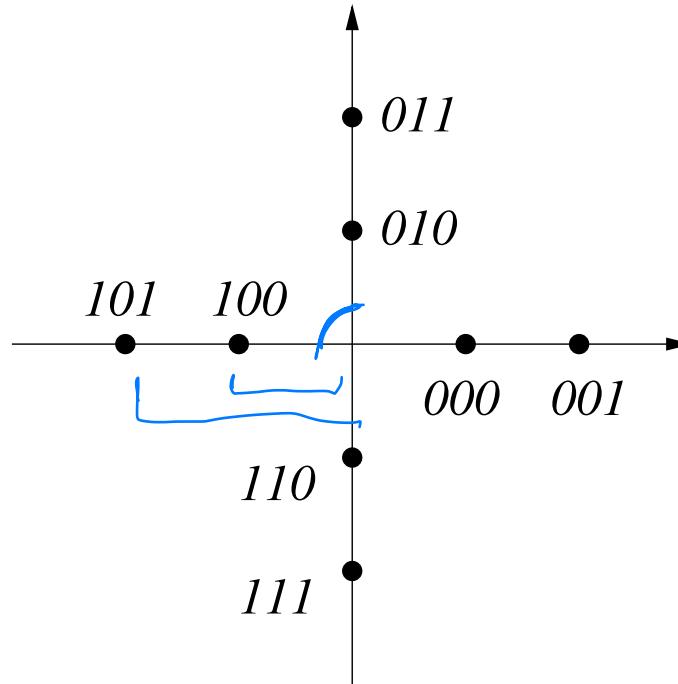


QAM: Examples

- QAM has numerous variants: (1) 4-QAM uses 1 amplitude and 4 phases, (2) 8-QAM uses 2 amplitudes and 4 phases.



4-QAM



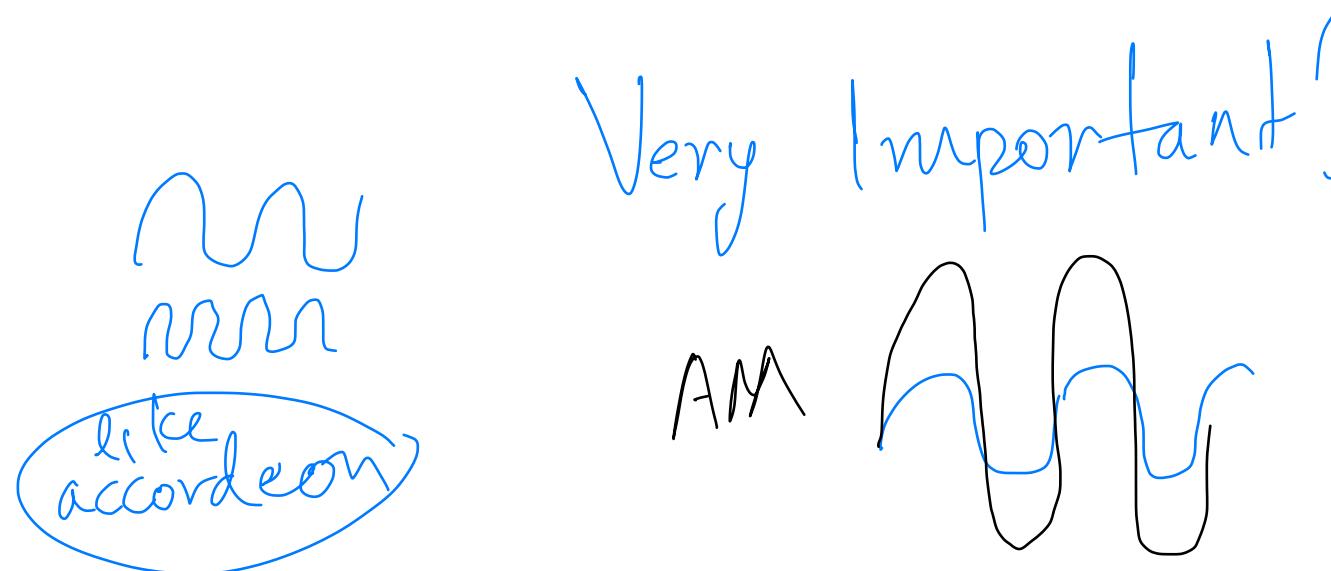
8-QAM

Analog → Analog

Analog-to-Analog: From Waveforms to Waveforms

Very useful for transferring from medium to medium, multiplexing, etc.

- **Amplitude Modulation (AM):** You modulate the amplitude.
- **Frequency Modulation (FM):** You modulate the frequency.
- **Phase Modulation (PM):** You modulate the phase.



Media

Bit Pipes

There are several kinds of bit pipes.

- **Synchronous:** The data is supplied synchronously.
 - If there is no data it must send dummy data.
- **Intermittent Synchronous:** It is the same as above except that if there is no data available, the modem sends no signal.
- **Asynchronous:** Data flows with no synchronicity whatsoever.
 - Bits within a character are sent at a fixed rate, but successive characters may be separated by a variable delay.
- **Total Asynchronous:** No restrictions on delay.

Transmission Lines and Signals

- Transmission lines are:
 - either simplex (only in one direction) 
 - or half-duplex (either direction but only one direction at a time) 
 - or (full-)duplex (either direction, both directions simultaneously) 
- Signals are:
 - either discrete or continuous
 - either periodic or aperiodic
 - represented either as functions either of time or frequency
- sin and cos are two fundamental periodic signals which are also “orthogonal”.

Propagation Media

- **Guided** (e.g., Twisted Pair, Coaxial Cable, Optical Fiber) and **Unguided** (Radio, Microwave, Satellite).

Medium	Data Rate
Twisted Pair	1 Mbps
Coaxial Cable	10 – 1,000 Mbps
Optical Fiber	> 1,000 Mbps
Radio	< 1,000 MHz
Microwave	> 1,000 MHz
Satellite	> 1,000 MHz

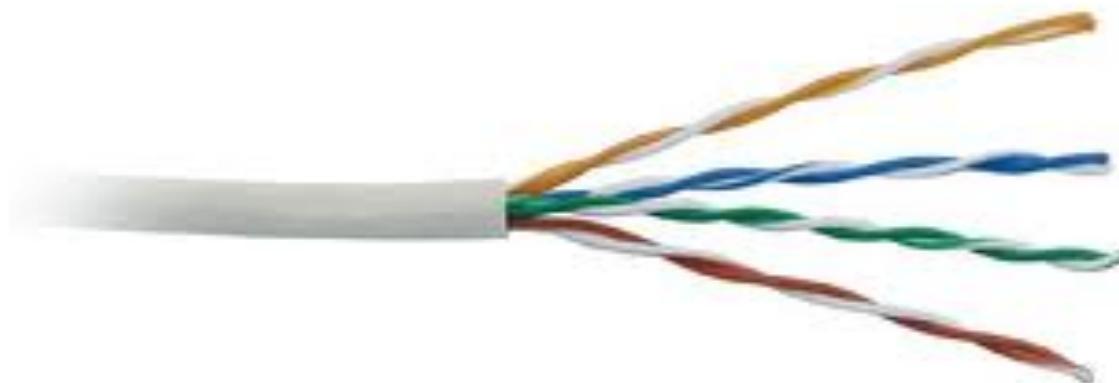
//

modulation

- They are unreliable bit pipes (with various degrees of unreliability) which makes important the study of **error-correction** and **error-detection**

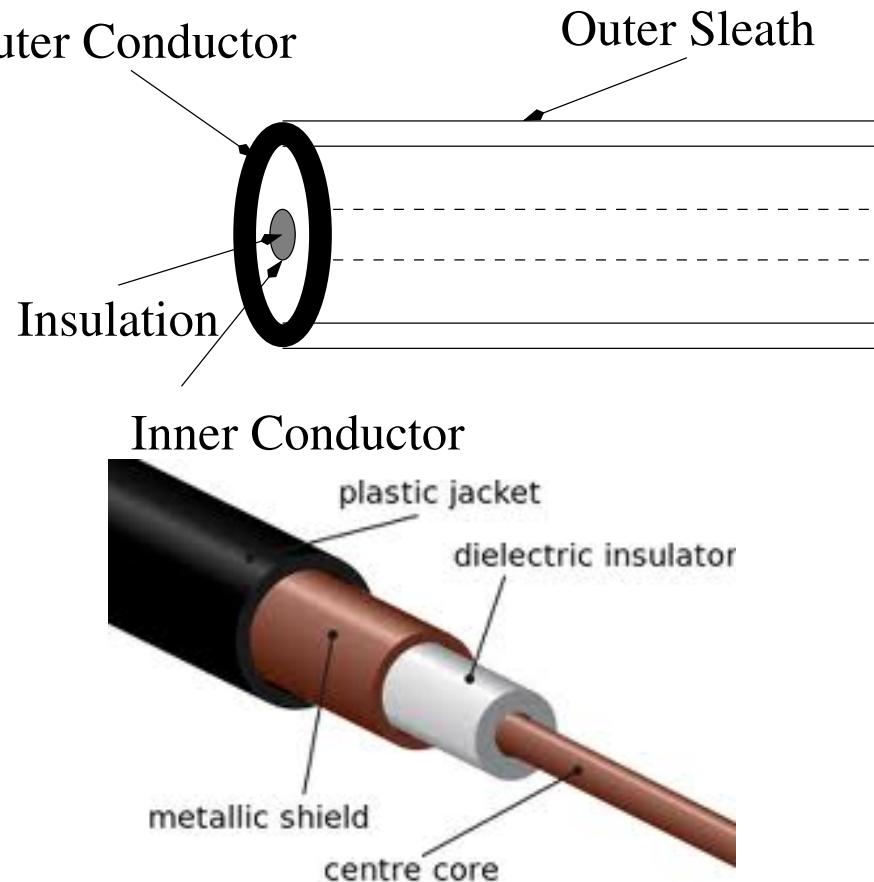
Propagation Media (Twisted Pair)

- Telephony for Subscribers
- Local Stations.
- ADSL (Assymmetric Digital Subscriber Line).



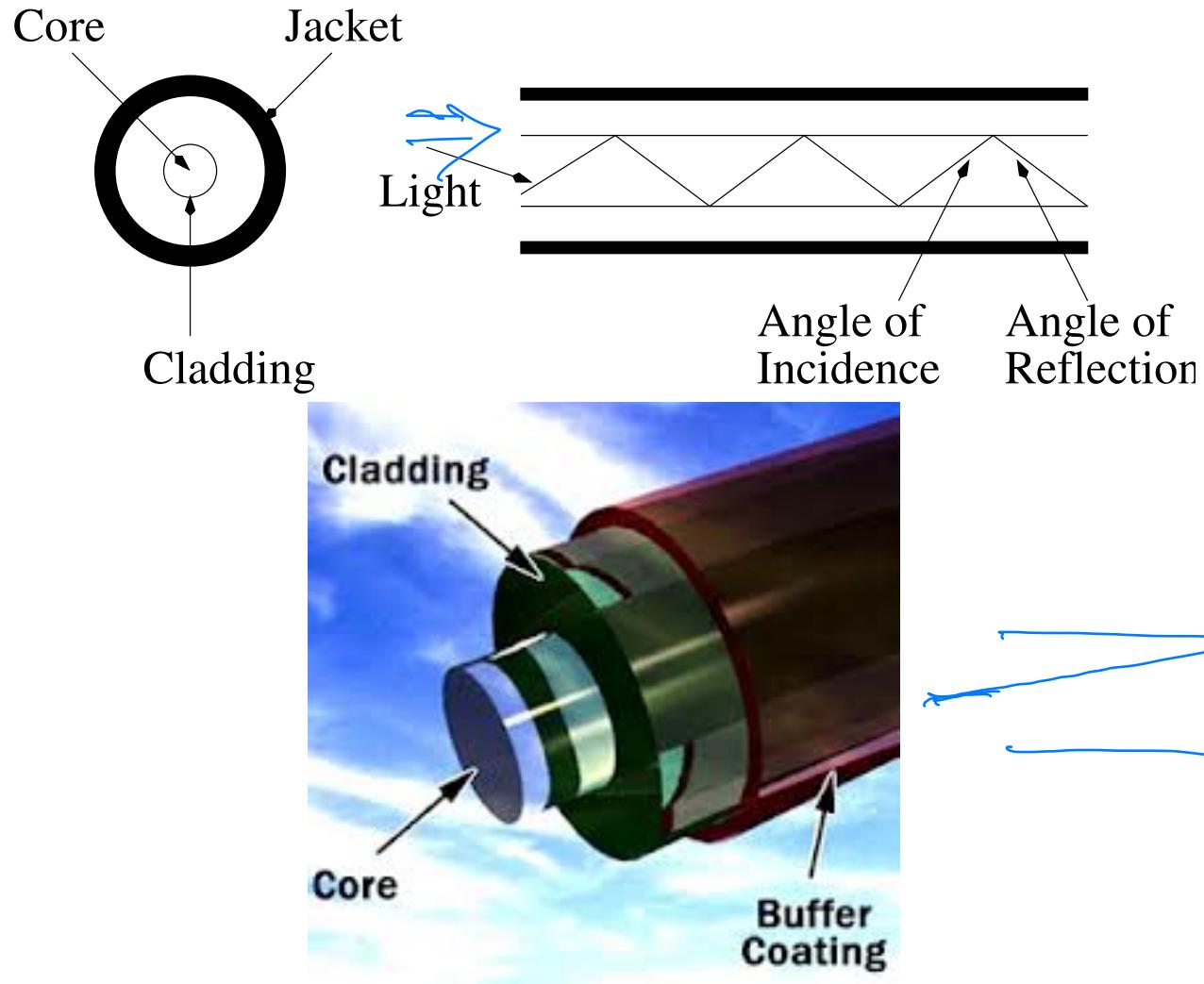
Propagation Media (Coaxial Cable)

- LANs, Cable TV, High-Speed point-to-point.



Propagation Media (Optical Fiber)

- Growing in importance for High-end applications.



Need for Repeaters

- For all these three types of media, propagated signal decays exponentially with distance and (many) repeaters are needed.
- In general, the rate of attenuation depends on the frequency.

Propagation Media (Radio: frequency < 1,000)

- **< 30 MHz:** Electromagnetic waves reflected by ionosphere.
 - Long distance of propagation is possible by reflection to the ionosphere.
 - The 3 to 30 MHz range is called HF band and is used by ham radio. This band is very noisy and achieves data rates of $\leq 2,400$ bps.
- **> 30 MHz:** Ionosphere transparent to these electromagnetic waves.
 - Propagation is on **line-of-sight** paths.
 - Antennas placed on towers, hills, etc. in order to increase length of line-of-sight paths.
 - Used for UHF and VHF TV broadcast, FM broadcast, and Packet Radio Networks.

Propagation Media (Microwave, Satellite: frequency > 1,000)

- **Microwave: frequency > 1,000**
 - Uses line-of-sight paths
 - Antennas used are highly directional
 - Typical lengths achieved are 10 – 200 km
 - Larger paths need repeaters
 - Can carry 1,000 Mbps
- **Satellites: frequency > 1,000**
 - Rates similar to Microwave
 - Satellites used as repeaters
 - Used for broadcast and multicast



Allocation Techniques (1/2)

- Allocation techniques depend on network type (wireline or wireless) and technology used.
- ALOHA (a dynamic allocation technique) is used in Ethernet.
- Several types of DMA (Division Multiple Access) techniques have been used in Wireless network design.
- The table below

Features	1G	2G	3G	4G	5G
Start/Development	1970/1984	1980/1999	1990/2002	2000/2010	2010/2015
Technology	AMPS, NMT, TACS	GSM	WCDMA	LTE, WiMax	MIMO, mm Waves
Frequency	30 KHz	1.8 Ghz	1.6 - 2 GHz	2 - 8 GHz	3 - 30 Ghz
Bandwidth	2 kbps	14.4 - 64 kbps	2 Mbps	2000 Mbps to 1 Gbps	1 Gbps and higher
Access System	FDMA	TDMA/CDMA	CDMA	CDMA	OFDM/BDMA
Core Network	PSTN	PSTN	Packet Network	Internet	Internet

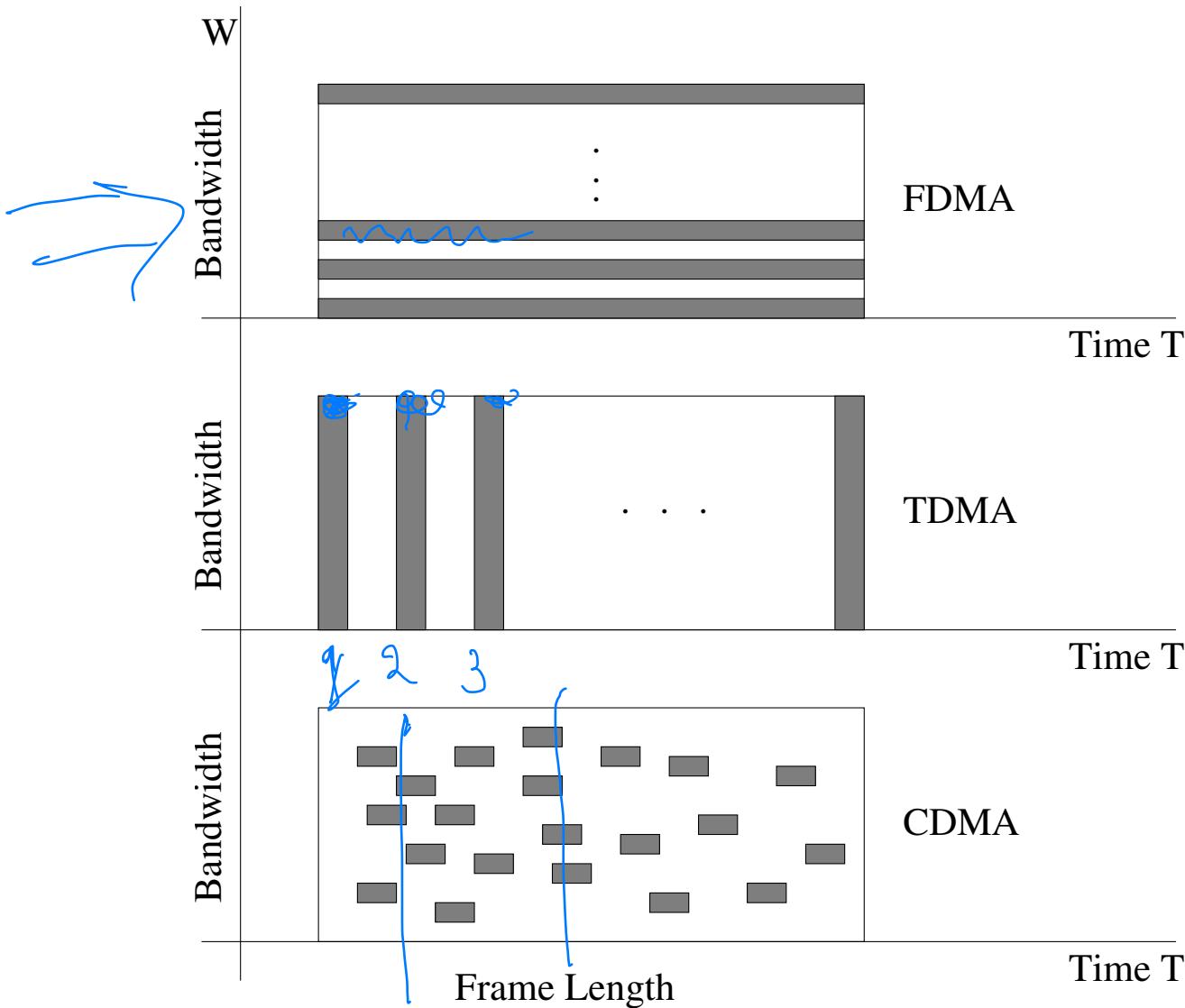
depicts the evolution of DMA techniques.

Allocation Techniques (2/2) $W = [f_L, f_H]$

- **Frequency DMA:** Splits channel into many little channels. If physical channel has bandwidth W herz then we split it into m equal subchannels each of bandwidth W/m .
- **T(ime)DMA:** Same bandwidth is used, however data is sent in frames with each frame having m slots (e.g T1 carrier).
- **C(ode)DMA:** Encodes 64 channels into 1.25 MHz of spectrum. Each channel uses an orthogonal code with all channels transmitting across the same bandwidth.
- **S(tatistical)DMA:** Channels are allocated at random. This is the most efficient but has a lot of overhead.

DMA = Division Multiple Access

Conceptual Characterization of Allocation Techniques



Coctail Party Paradigm

- Imagine a coctail party with many participating people in a large room.
 - **FDMA:** is when people break up into small independent groups each group holding conversation independently of the other groups.
 - **TDMA:** is when all people are in a single group with people taking turns talking.
 - **CDMA:** is when they are all talking at once but each pair of people using a different language.
- In certain cases we may use mixed DMA techniques, e.g., GSM uses FDMA and TDMA).

Exercises^a

1. Why do we need digital to digital encodings?
2. For n random bits, what is the expected number of stuffed bits?
3. What are advantages and disadvantages of using geostationary satellites in communication networks?
4. How Low Earth Orbit (LEO) satellites differ from geostationary satellites?
5. Why can't we simplify things and use exclusively wireless networks everywhere?
6. Why do we need physical to digital (and vice versa) conversions?
7. It is desired to send a sequence of computer screen images over an optical fiber. The screen is 480×640 pixels, each pixel

^aNot to hand in!

being 24 bits. There are 60 screen images per second. How much bandwidth is needed?

8. There are several DMA (Division Multiple Access) access techniques in the scientific literature. Investigate how the following work:
 - (a) Spatial Division Multiple Access (allocates space)
 - (b) Beam Division Multiple Access (BDMA) (uses multiple radiators)
 - (c) Orthogonal Frequency Division Multiple Access (OFDMA)
9. Sometimes combinations of the Division Multiple Access techniques discussed is also being used. Elaborate why this can create efficiencies. E.g., GSM uses a combination of both TDMA and FDMA techniques. The FDMA element divides the assigned frequency of 25 MHz bandwidth into 124 carrier frequencies, all spaced 200 kHz apart. The carriers are also

divided in time using TDMA. Different users of each RF channel are allocated different time slots (there are 8 time slots per channel).

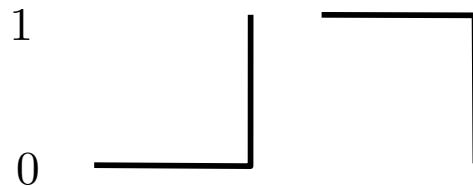
10. The uplink band in GSM has a total of 25 MHz of bandwidth and each radio channel has an assigned bandwidth of 200 kHz. What is the number of radio channels (FDMA)?
11. In practice GSM uses 124 channels (not 125). Each channel is divided into 8 time slots, so 8 users are allowed per radio channel (TDMA). What is the max number of users?
12. GSM also has a corresponding downlink band for sending signals to the mobile phone. For GSM in Europe the uplink and downlink frequency bands are 890 to 915 MHz and 935 to 960 MHz, respectively. How does this affect the number of users?

Appendix

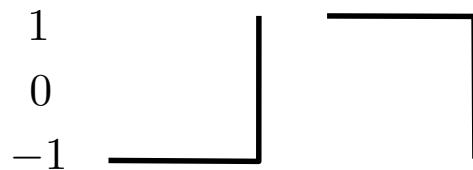
(Digital → Digital)
(Not Required)

Representing Transitions

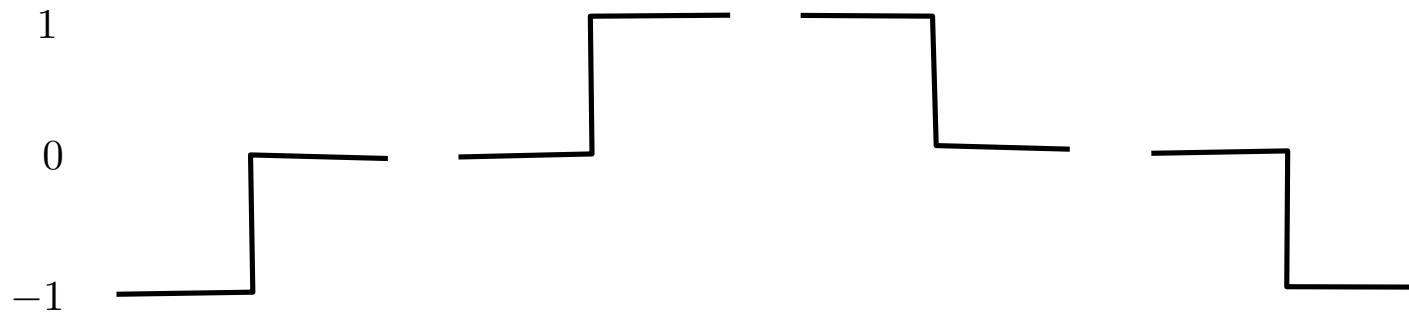
- Levels 0, 1 (Returns to 0)



- Levels $-1, 0, 1$ (Does not return to 0)



- Levels $-1, 0, 1$



Amplitude Levels

1. **Unipolar:** Very primitive (and almost obsolete) and uses only one level of value
2. **Polar:** Two levels (one positive and one negative) of amplitude are being used.
 - (a) NRZ (Non Return to Zero): Signal always either positive or negative (does not return to 0).
 - i. NRZ-L (Non Return to Zero Level): the level of the signal is dependent upon the state of the bit it represents.
 - ii. NRZ-I (Non Return to Zero Invert): voltage inversion represents the bit 1.
 - (b) RZ (Return to Zero): uses positive, negative and zero.
3. **Bipolar:** uses three levels (positive, negative and zero) but unlike RZ zero is used to represent binary 0.

1: Unipolar

- Uses only one voltage level (other than 0).
- Very primitive (and almost obsolete) and uses only one level of value.
- This type of encoding is also known as **unipolar** encoding because it uses only one level of value.

Problems with Unipolar

- **Baseline wander:**
 - The receiver takes the average of what it has seen so far to distinguish between high and low!
 - Thus a long sequence of consecutive 1s (or 0s) changes this average and is hard to distinguish 0 and 1
- **Clock Recovery:**
 - Coding and decoding processes are driven by (separate) clocks which must be synchronized.
 - Frequent transitions from 0 to 1 and vice versa are necessary in order to enable sender and receiver to synchronize.
- This indicates a “need for change” while transmitting!

2: Polar Encodings RZ and NRZ

- **NRZ-L:**^a Voltage is constant during the bit interval; the level of the signal depends on the state of the bit. Positive (resp., negative) voltage means bit is 0 (resp., 1).
- **NRZ-I (invert on ones):**^b Voltage is constant during the bit interval; the signal is inverted when a 1 is encountered and is left unchanged otherwise, at the beginning of the bit time.
- **Bipolar RZ:**^c Is a return to zero encoding (always returns to 0) so that three levels: negative, zero, and positive are used. .

^aNRZ-L used for short distances

^bNRZ-I commonly used with serial ports.

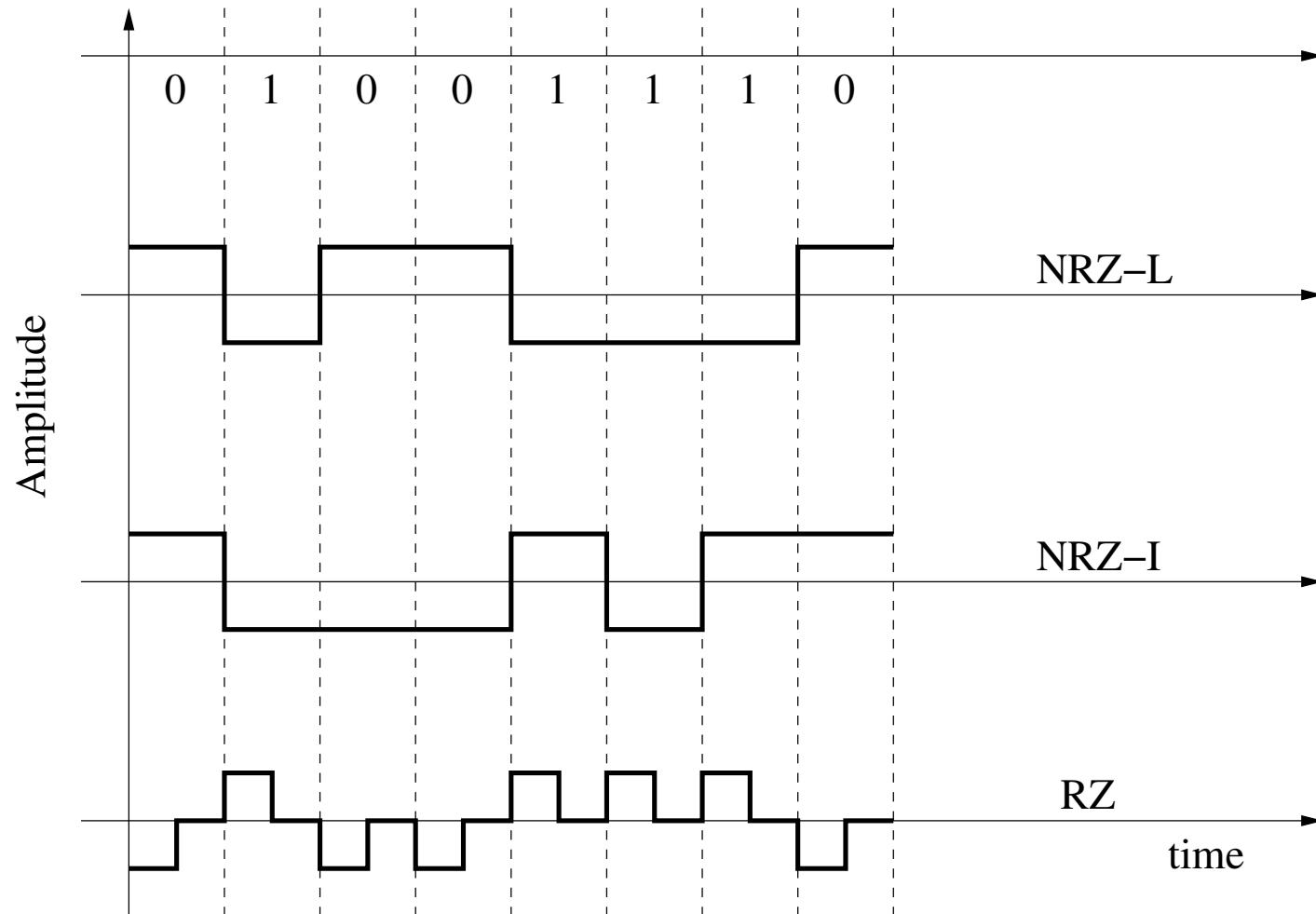
^cBipolar RZ encodings are designed to be DC-Direct Current balanced.

Solution: NRZ-I

- **NRZ-I** (i.e., NRZ Inverted):
- The sender does the following:
 - To encode a 1: makes transition from current signal
 - To encode a 0 stays at the current signal.
- **Example:**

0	0	1	1	1	0	1	0	1	1
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
0	0	1	0	1	1	0	0	1	0

Bipolar RZ and NRZ Polar Encodings



Polar Biphasic Manchester (1/2)

- **Manchester:**

Transmits $0 \rightarrow 01$ and $1 \rightarrow 10$.

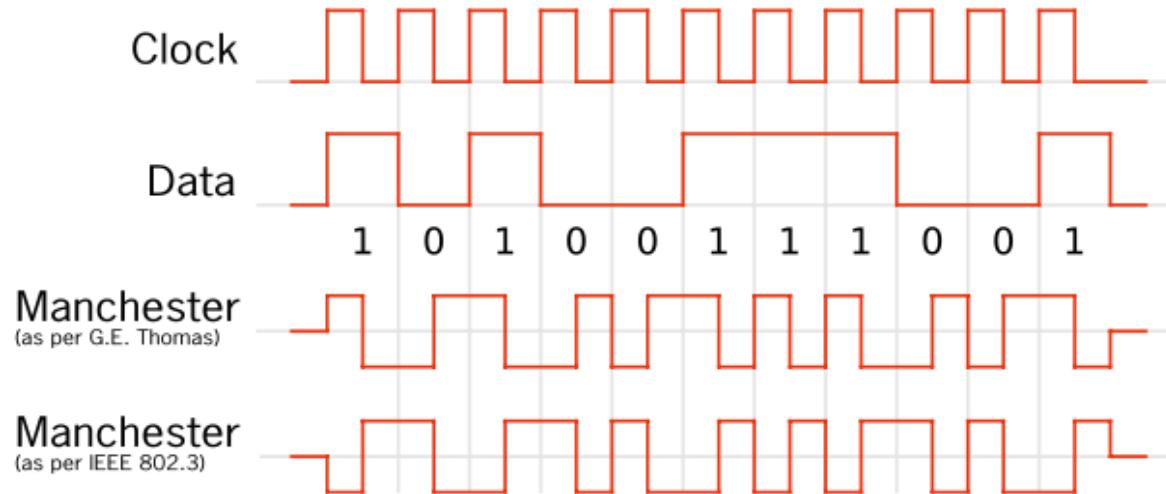
- **Example:**

0	0	1	1	1	0	1	0	1	1
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
01	01	10	10	10	01	10	01	10	10

- **Shortcomings:** Manchester code needs twice the bandwidth of asynchronous communications, and the signal spectrum is much wider. Most high-speed communication now uses encoding schemes with better coding performance.

Polar Biphasic Manchester (2/2)

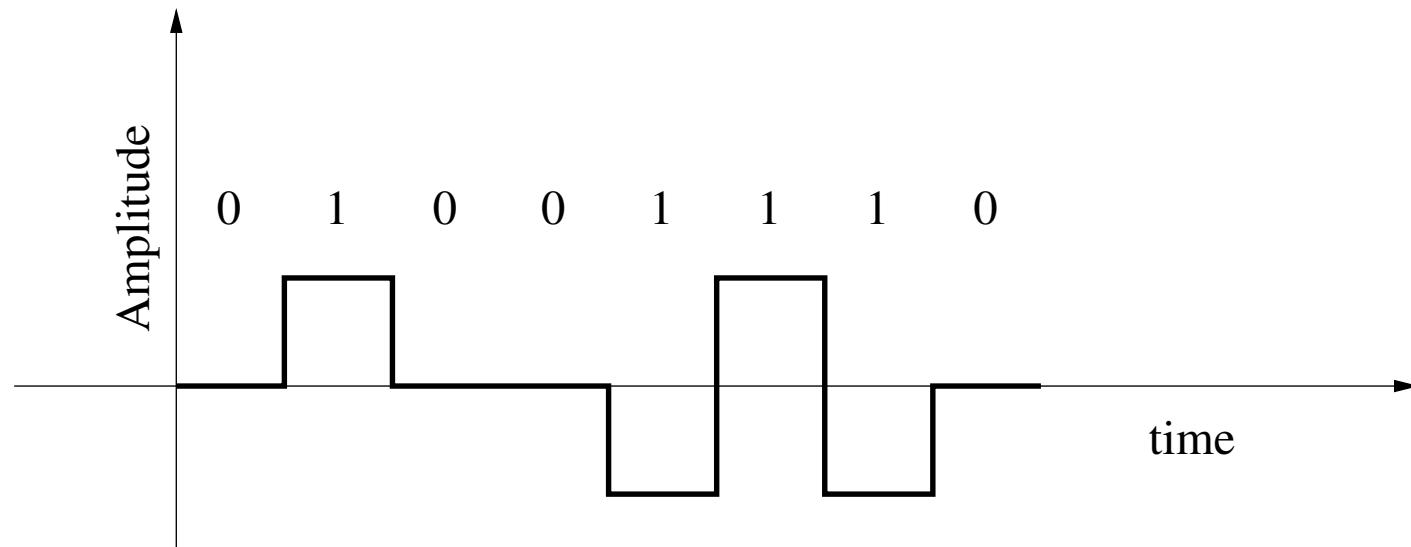
- **Manchester:** Transmits the XOR of NRZ with the clock.



- Important consideration is synchronization of the receiver to the transmitter. It might appear that a half bit period error would give an inverted output at the receiver, but for typical data this leads to code violations. The receiver can detect these violations and use this information to synchronize accurately.

Bipolar (Alternate “Mark” Inversion (AMI))

- Here the word “Mark” means “1”.
- AMI means “Alternate 1 Inversion”: i.e., invert 1s.



- Neutral voltage represented by 0 and 1s represented by alternating positive and negative voltages.

More Bipolar

- **Bipolar 8 Zero Substitution (B8ZS):**
 - Force artificial signal changes to AMI whenever eight consecutive 0s occur in the signal (similar to 4B/5B).
 - This system is used in North America.
- **Bipolar (High Density Bipolar 3 (HDB3)**
 - Force artificial signal changes to AMI whenever four consecutive 0s occur in the signal (similar to 4B/5B).
 - This system is used in Europe and Japan.

The diagram illustrates the Data Link Layer. It features two blue-outlined ovals side-by-side; the left oval contains the word "DATA" and the right one contains "LINK". Below these ovals, the word "LAYER" is written in a large, bold, black font.

DATA **LINK**
LAYER

Outline

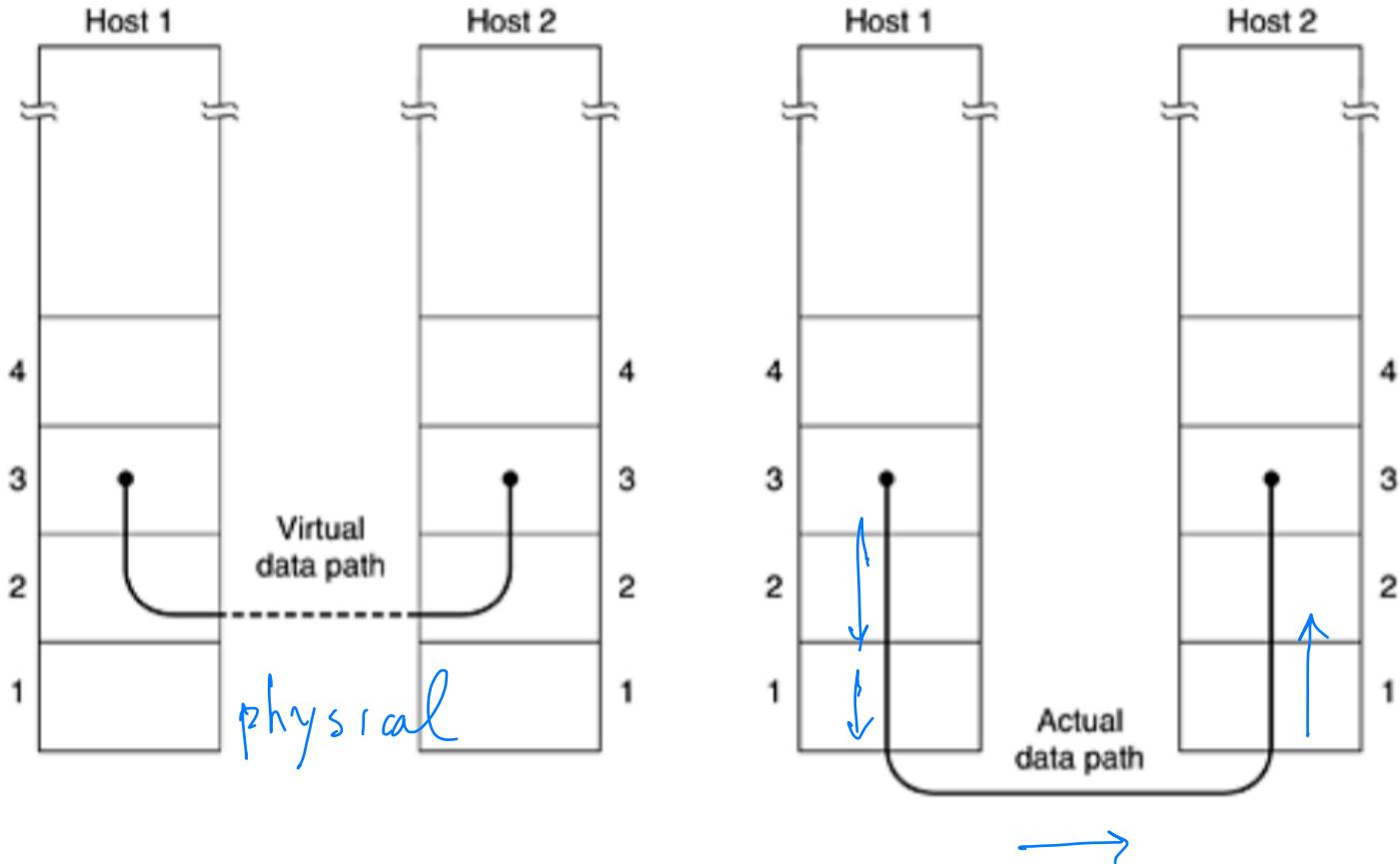
1. Frames
2. Framing Methods

Data Link Layer/Control

- This layer has a number of specific functions that include:
 1. Well-defined service interface to the network layer.
 2. Determining how bits from the physical layer are grouped into frames. *"delimit the bits"*
 3. Error correction and detection. *at the packet level*
 4. Regulating the flow of frames through retransmission strategies.
- The main abbreviations are:
 - Data Link Layer (DLL)
 - Data Link Control (DLC)

Virtual Connection

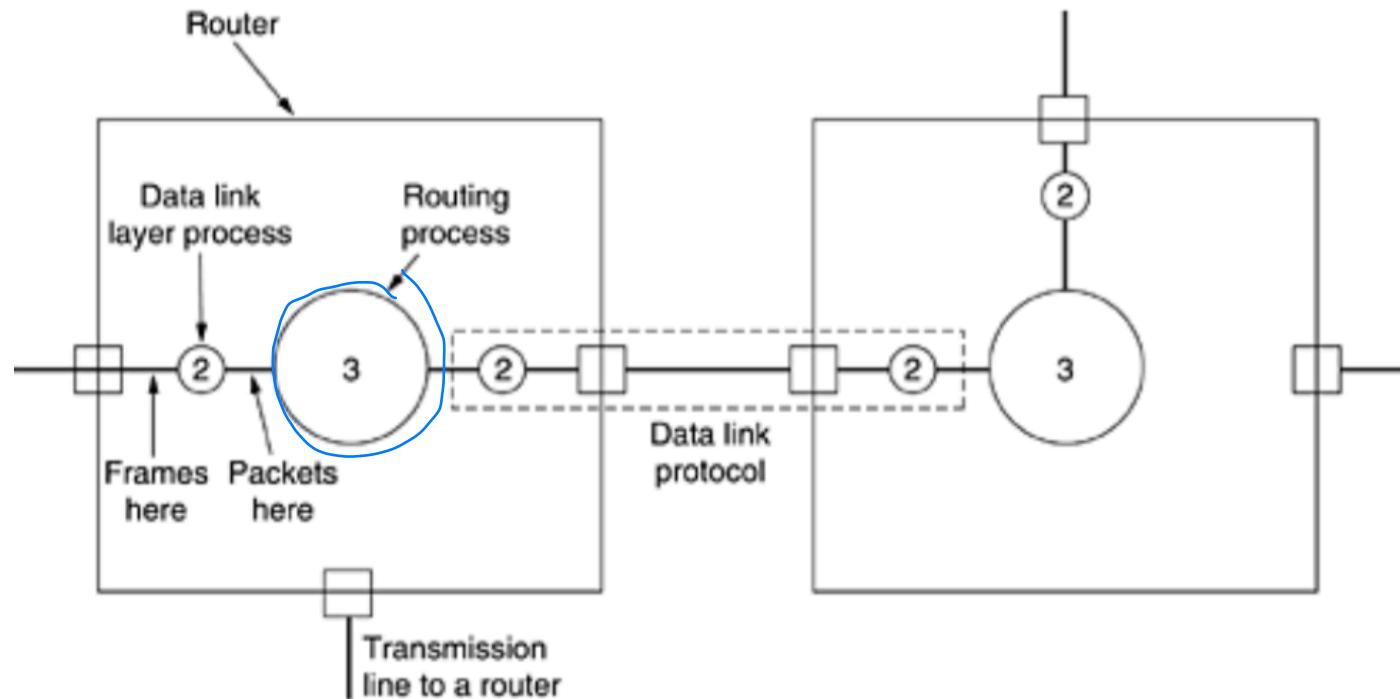
- Virtual communication in Data Link Layer



- versus actual communication.

Location with Respect to Router (1/2)

- When a frame arrives at a router, the hardware checks it for errors, then passes the frame to the data link layer software.



(which might be embedded in a chip on the network interface board)

NIC

Location with Respect to Router (2/2)

- The data link layer software checks to see if this is the frame expected, and if so, gives the packet contained in the payload field to the routing software.
- The routing software then chooses the appropriate outgoing line and passes the packet back down to the data link layer software, which then transmits it.

Frames

Frames and Pictures

- The idea of the terminology is as usual

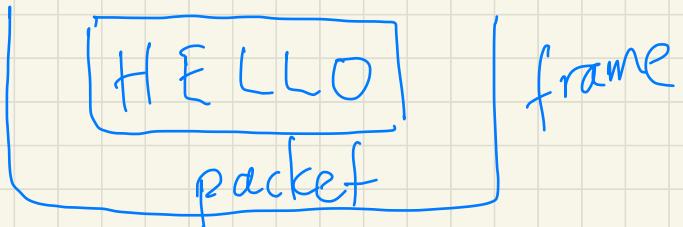


- There is a picture (packet) and is placed inside a frame.
- The idea is that the picture is “most important” and must be protected by a frame!



packet
at the DCC

- DLC puts the bits in packets
and frames



- As you move up the hierarchy of layers you add frames to the packet you receive

PhyLayer

DLC

Nef

TCP

Sec.

Appl

packet

frame (packet)

frame (frame (packet))

frame (frame (frame (packet)))

!

:

<

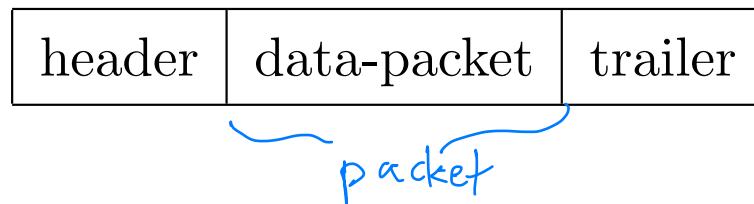
Frames and Packets

- Data is combined and organized in **frames** and **packets** for transmission purposes in the physical layer.

- **Problem 1:**

Decide at the receiving DLC (Data Link Control) where successive packets start and end.

- A typical frame structure contains the following

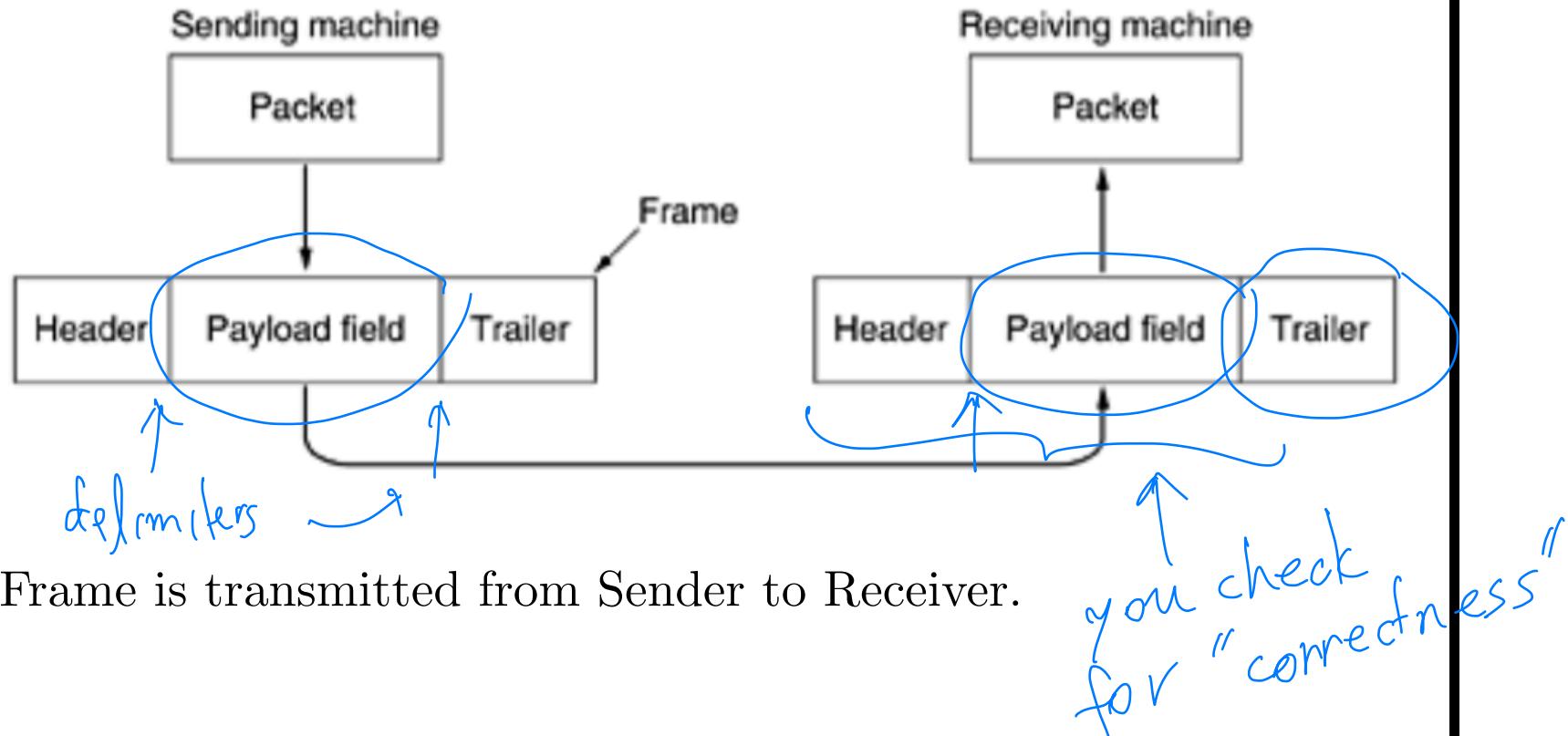


- **Problem 2:**

Transmission errors may occur and issues of interest include error correction, error detection.

Framing

- Each frame contains a frame header, a payload field for holding the packet, and a frame trailer



- Frame is transmitted from Sender to Receiver.

Framing

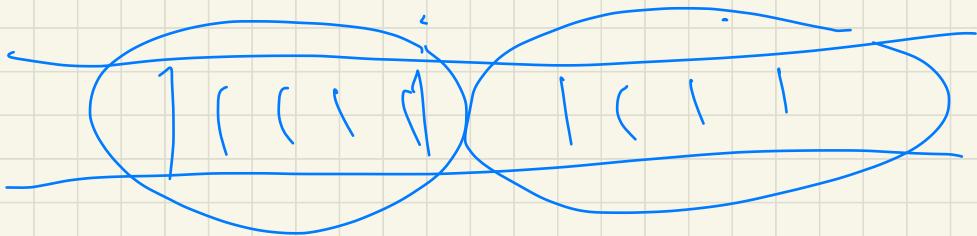
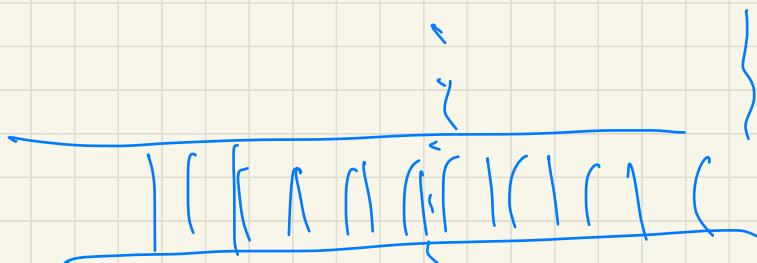
- While “Physical” layer produces virtual bit pipe with no concept of packets, “Data Link” layer has input/output packets
- DLC (Data Link Control) places
 - **header** bits at the beginning, and
 - **trailer** bits at the endof packets to form **frames**
- Purpose of header/trailer:
 - Error control (e.g., **CRC**)
 - Distinguishing one frame from the next and from idle

Cyclic Redundancy Code

Phys Layer; Stream of bits

D L C

Stream of frames



How do you identify Frames?

- First task is to delineate frames
 - Receiver needs to know when a frame starts and ends
 - Otherwise, errors from misinterpretation of data stream
- Several different alternatives
 - Fixed length (bits) frames
 - Explicitly delimited frames
 - Length-based framing
 - Sentinel-based framing (use markers)
- Fixed duration (seconds) frames

Framing Methods

Framing Methods

- Four methods used to distinguish beginning and end of frames:
 1. **Character-based framing**
uses special communication control characters.
 2. **Bit-oriented framing with flags**
uses flags (i.e., special strings of bits).
 3. **Clock-based framing**
 4. **Length fields**
frame length given in the field.
- The idea is to “enforce” simple presentation techniques to mitigate inconsistencies and reduce errors.

Lots of material in “white” pages

Misses 1 sec' in ~~17~~ billion yrs

We try to structure the models of framing so as to mitigate inconsistencies.

However, ~~us~~ something will still be missed.

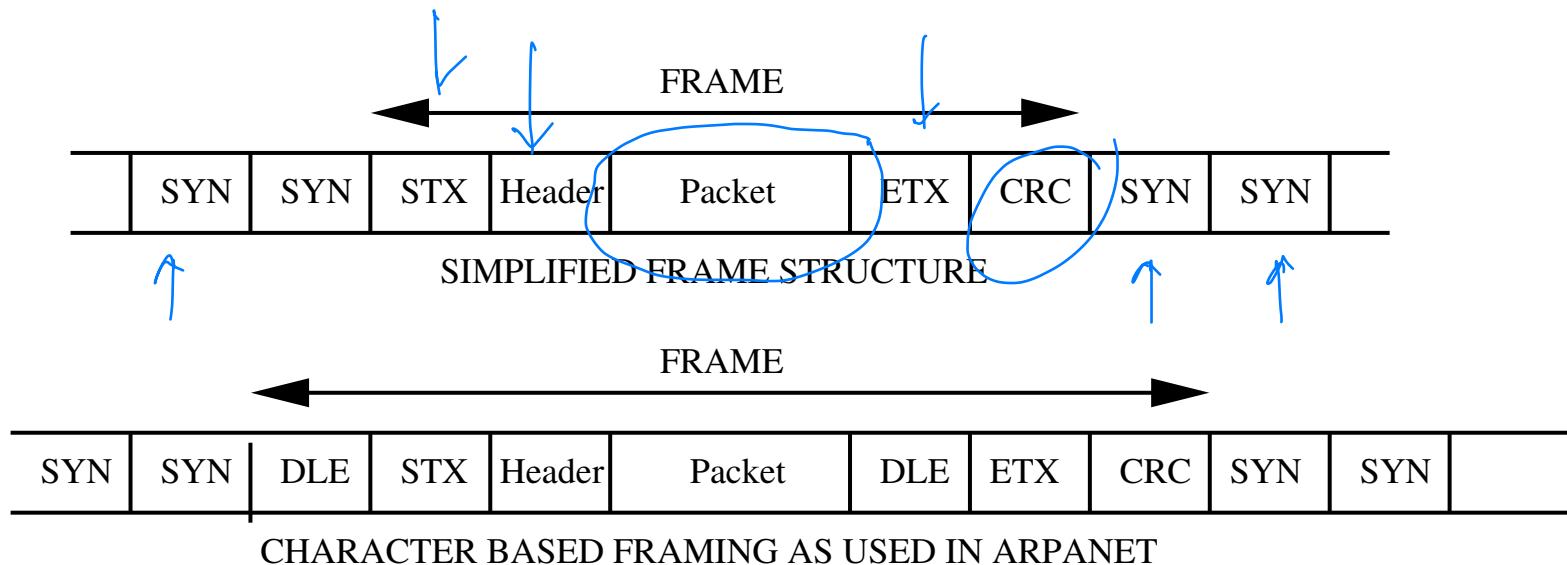
If you miss and cannot fix it you request retransmission

Character (or Byte) Based Framing (1/4)

- Developed for and used in character-oriented communication, e.g., sending ASCII
- Special characters set aside for
 - Synchronous idle (SYN): provides idle fill between frames, used within frames for synchronization, bridges delays in supplying characters.
 - Start of text (STX)
 - End of text (ETX)
 - Data Link Escape (DLE): used only in special modes of transmission to provide more transparency (e.g., international uses of communication characters).

Character (Byte) Based Framing

- Uses characters as the basis of frames.



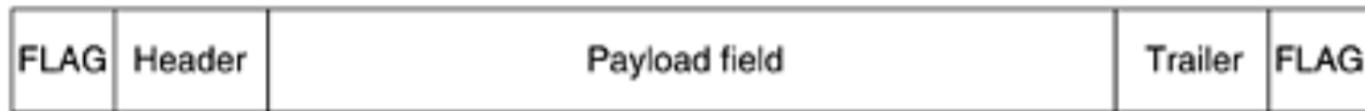
SYN = Synchronous Idle
STX = Start of Text

DLE = Data Link Escape
ETX = End of Text

- Certain combinations of characters are given special meaning.

Character (Byte) Based Framing

- This framing method gets around the problem of resynchronization after an error by having each frame start and end with special bytes.



- In the past, the starting and ending bytes were different, but in recent years most protocols have used the same byte, called a flag byte, as both the starting and ending delimiter, known as FLAG.
- If the receiver ever loses synchronization, it can just search for the flag byte to find the end of the current frame.
- Two consecutive flag bytes indicate the end of one frame and start of the next one.

Problems/Solutions: Byte Stuffing

- **Problem:** When sending arbitrary data (as opposed to characters like ASCII) control characters may appear in frame
 - **Solution:** introduce DLE (data link escape) character
 - Start of text indicated by DLE STX, end of text by DLE ETX
- **Problem:** What if DLE appears in packet as part of the data?
 - **Solution:** Use DLE DLE - the first DLE of every pair is stripped off at the receiving end
 - E.g., X DLE ETX means end of text but X DLE DLE ETX means data bits X DLE ETX

An alternative: DLE, STX, ETX

do not use these
for your data

Ω Ω

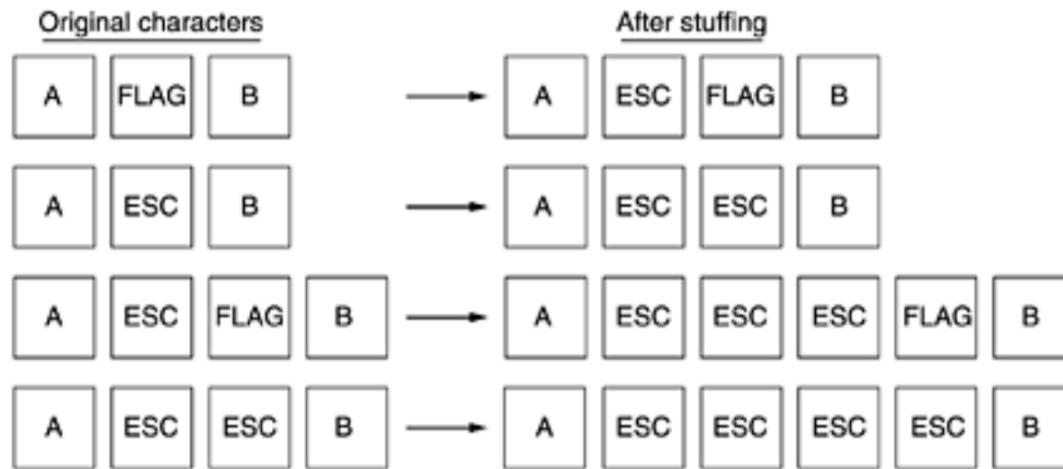
September 26, 2020

Ω X, Ω Y, -

X
Y
Z
Ω

Problems/Solutions: Byte Stuffing

- The sender's data link layer inserts a special escape byte (ESC) just before each “accidental” flag byte in the data and the data link layer on the receiving end removes the escape byte before the data are given to the network layer.



- This technique is called byte stuffing or character stuffing.
- Thus, a framing flag byte can be distinguished from one in the data by the absence or presence of an escape byte before it.

Errors

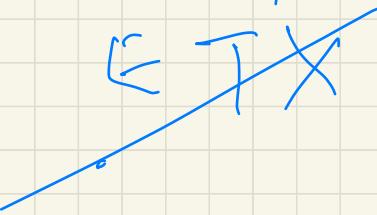
- Errors in DLE STX or DLE ETX may cause a frame to be missed
- Errors inside the frame will (hopefully) be caught by CRC
- An error could cause DLE STX or DLE ETX to appear in the middle of the frame
- In either case a portion of the frame is dropped and remainder is unlikely to be accepted as correct due to CRC

If you miss the end of the frame

ETX

≤

ETX



Errors

- What happens if an escape byte occurs in the middle of the data?
- The answer is that it, too, is stuffed with an escape byte.
- Thus, any single escape byte is part of an escape sequence, whereas a doubled one indicates that a single escape occurred naturally in the data.

Extensions

- A major disadvantage of using this framing method is that it is closely tied to the use of 8-bit characters.
- Not all character codes use 8-bit characters.
- For example. UNICODE uses 16-bit characters,
- As networks developed, the disadvantages of embedding the character code length in the framing mechanism became more and more obvious, so a new technique had to be developed to allow arbitrary sized characters.

Bit-Oriented Framing (2/4)

- This technique allows the data frames to contain an arbitrary number of bits and allows character codes with arbitrary number of bits per character.
- Similar to above but a *flag* (fixed bit string) is used to delineate start and finish of frame
- Flags that could be used are strings of the form

$$01^j 0 := 0 \overbrace{11 \cdots 1}^j 0$$

↑ consecutive 1s

- E.g., ISO - High-level Data Link Control (HDLC) uses

$$01^6 0 := 0 \overbrace{111111}^6 0$$

as flag string.

Bit-Oriented Framing and Bit Stuffing

- To avoid interpreting data as flag *bit stuffing* is used.

Sender:

- With the exception of the flag 01^60 , after every sequence of 5 1's a 0 is stuffed.

Receiver:

- The first 0 after every sequence of five consecutive 1s is deleted.

- **Idle:** 15 1's, i.e.. 1^{15} .
- **Abort:** is 7 1's in a row.

} 15 values depend
 } 7 on the ISO "j"

NB: *Read Ahead* is necessary to predict correctly whether it is idle or abort.



Stuffed Bits: Example

- Whenever the sender's data link layer encounters five consecutive 1s in the data, it automatically stuffs a 0 bit into the outgoing bit stream.

011011111111111110010
01101111011111011111010010
 ↑
 Stuffed bits
011011111111111110010

- When receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically destuffs (i.e., deletes) the 0 bit.
- If the user data contain the flag pattern, 01111110, this flag is transmitted as 011111010 but stored in the receiver's memory as 01111110.

Receiver and Bit-stuffing

- If five consecutive 1s arrive, the receiver makes its decision based on next_bit.
- If next_bit = 0 it must have been stuffed and is removed.
- If next_bit = 1 then either this is end of frame or error has been introduced.
- To distinguish look at the next bit (after next_bit).

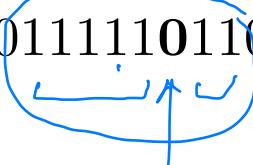
Example: Bit-Stuffing/Bit-Unstuffing

- Sender

00011111↓110011111↓01000

↓ stuffing 2 bits

00011111**011**0011111**00**1000



- Receiver

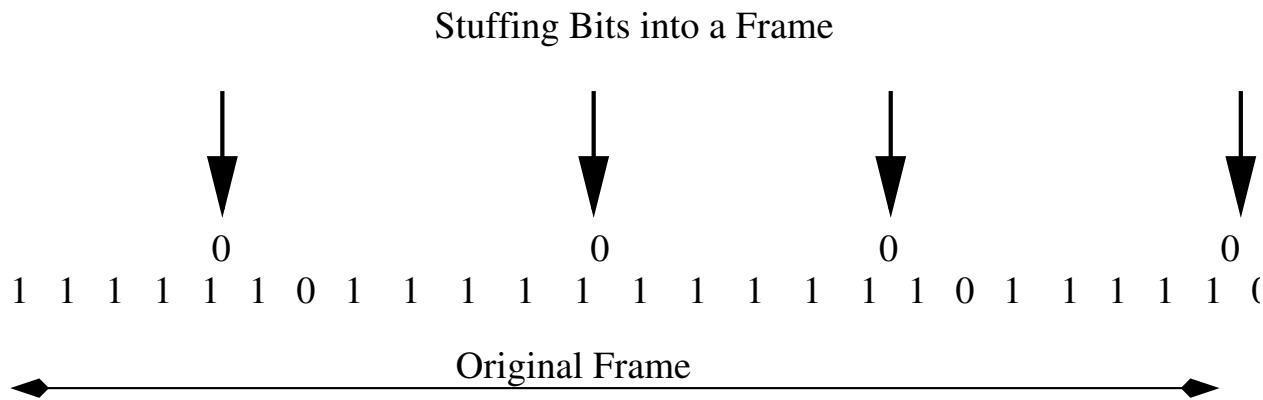
00011111**0**11001111**0**01000

↓ bit unstuffing 2 bits

000111111001111101000

Uses of Bit-stuffing

- Thus, 01^6 when followed by a
 - 0 indicates normal termination: it is the flag.
 - 1 indicates abnormal termination: abort
- Bit stuffing helps in synchronization because of short length sequences of 1s.
- **Example: Bit-Stuffing**

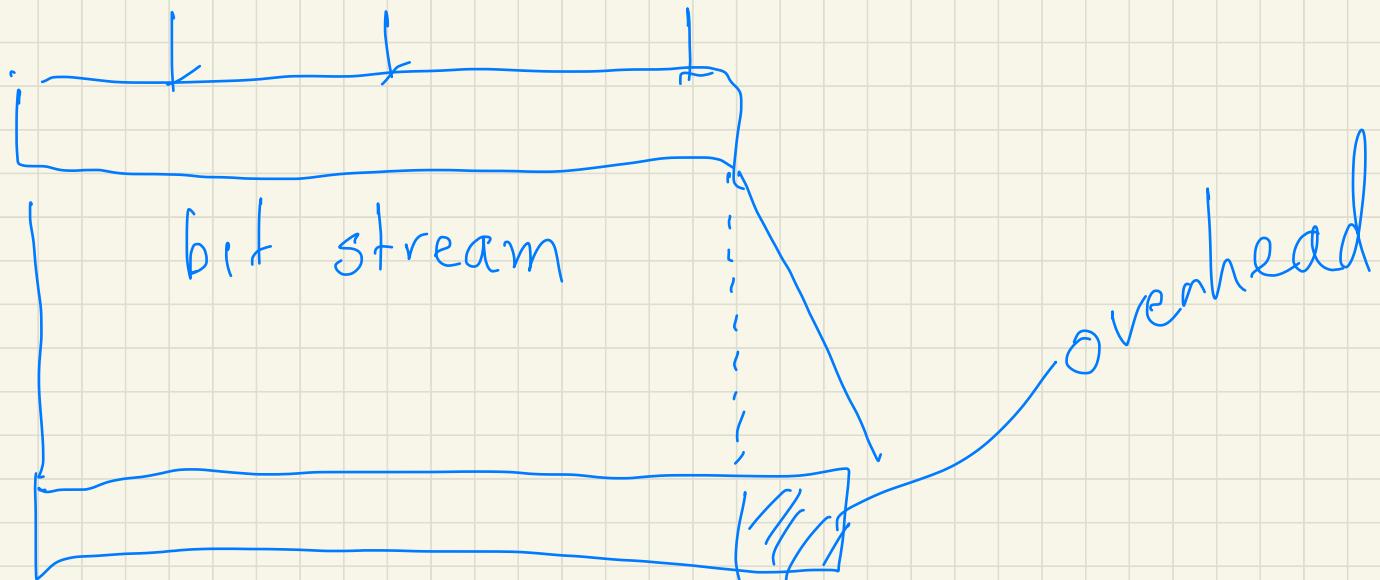


A 0 is stuffed after five consecutive 1s: 11111

A flag (six 1s surrounded by two 0s) 01111110 is sent at the end of frame

In bit stuffing ;

you are inserting bits

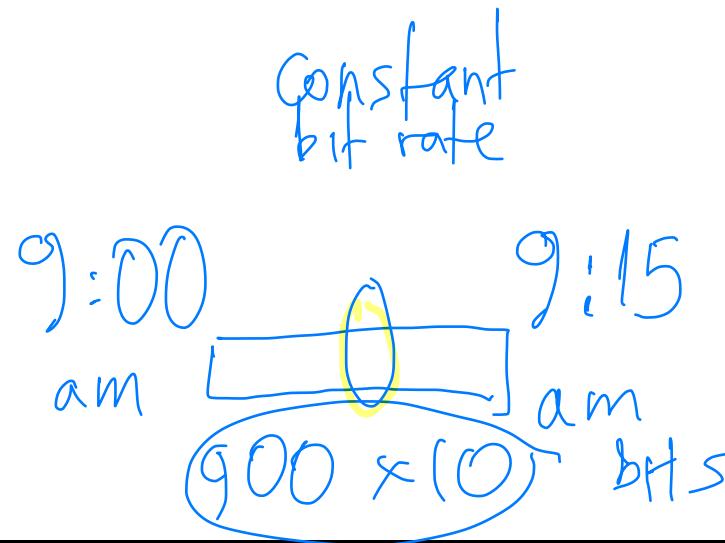


On Bit Stuffing

- With bit stuffing, the boundary between two frames can be unambiguously recognized by the flag pattern.
- If the receiver loses track of where it is, all it has to do is scan the input for flag sequences, since they can only occur at frame boundaries and never within the data.
- Bit-stuffing is only applicable to networks in which the encoding on the physical medium contains some redundancy.
 - Some LANs encode 1 bit of data by using 2 physical bits.
 - A 1 bit is a high-low pair and a 0 bit is a low-high pair.
 - This means that every data bit has a transition in the middle, making it easy for the receiver to locate the bit boundaries.
 - The combinations high-high and low-low are not used for data but are used for delimiting frames in some protocols.

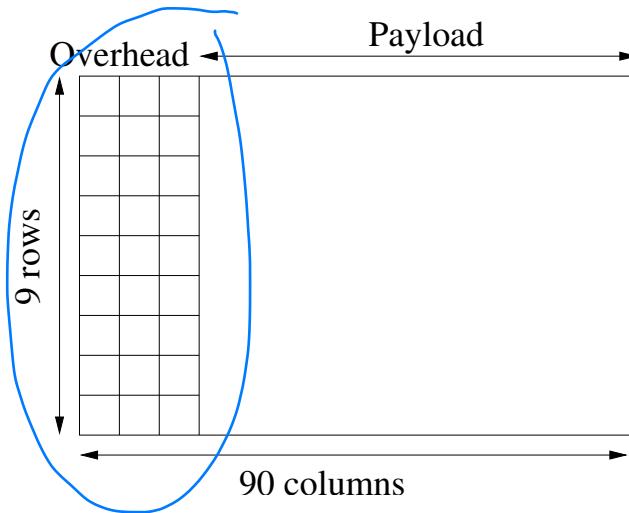
Clock-Based Framing (3/4)

- A series of repetitive pulses are used to maintain a constant bit rate and keep the digital bits aligned in the data stream.
 - Continuous stream of fixed-length frames
 - Clocks must remain synchronized
 - No bit or byte stuffing
 - Exemplified by SONET (Synchronous Optical Network) Standard.



SONET

- Standard has specific frame sizes.



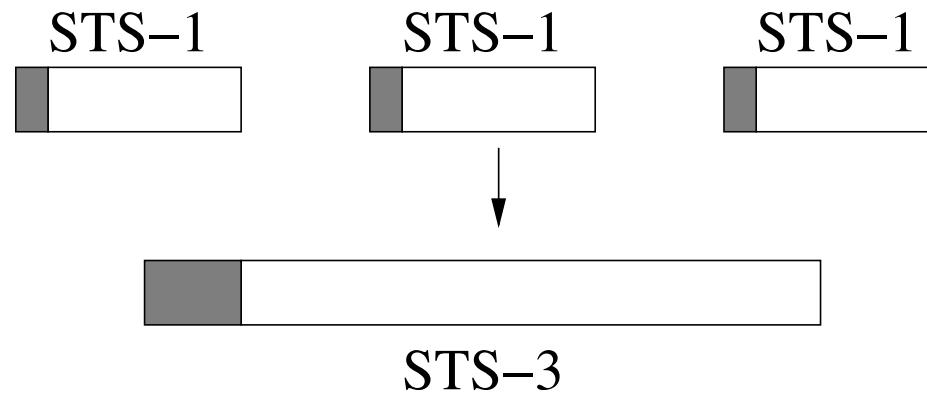
- Frame represented by nine rows, of ninety bytes each, with the first three bytes of each row overhead, i.e. total 810 bytes long.
- First 2 bytes of frame contain special bit pattern to help receiver determine start of frame.
 - Since this bit pattern may occur as part of payload receiver looks for special bit pattern every 810 bytes.

SONET Multiplexing

- SONET rates from STS-1 to STS-48: STS- n means n times STS-1.

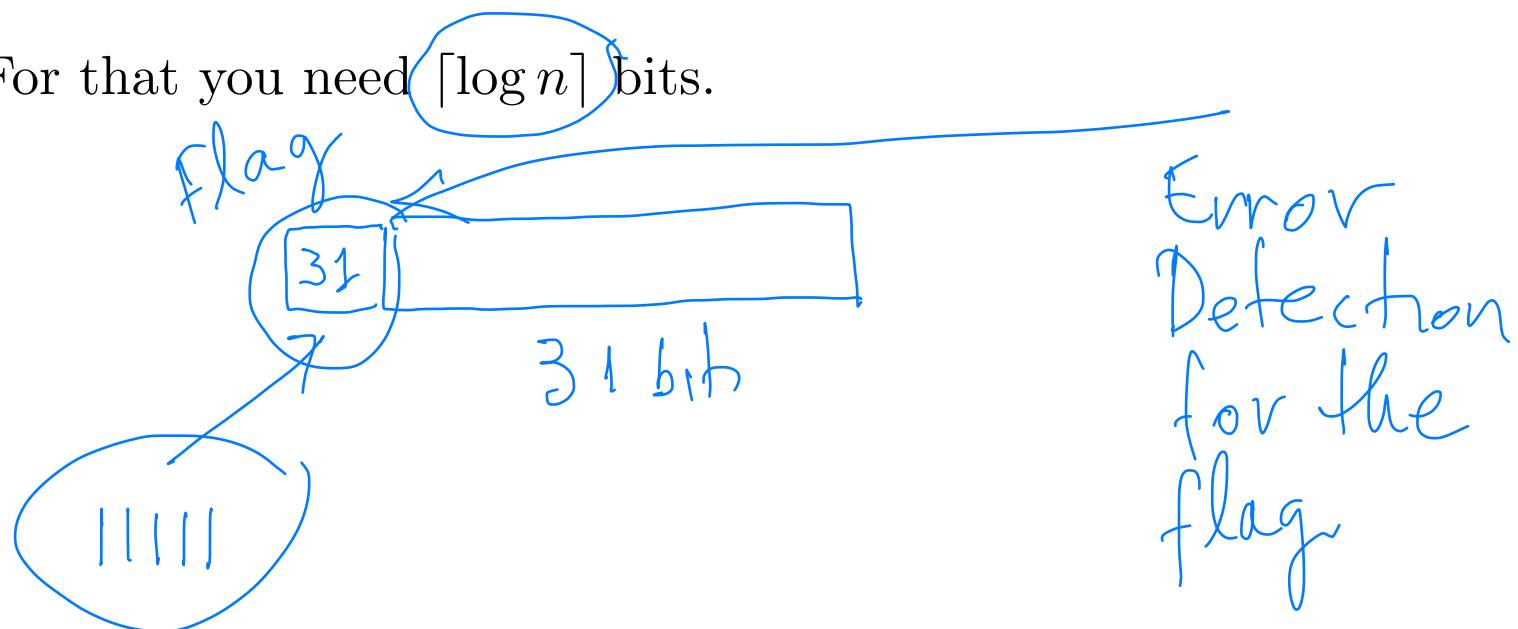
SONET	STS-1	STS-3	STS-12	STS-24	STS-48
Speed Mbps	51.84	155.25	622.08	1,244.16	2,488.32

- In multiplexing frames are concatenated to form new frames at different rates.



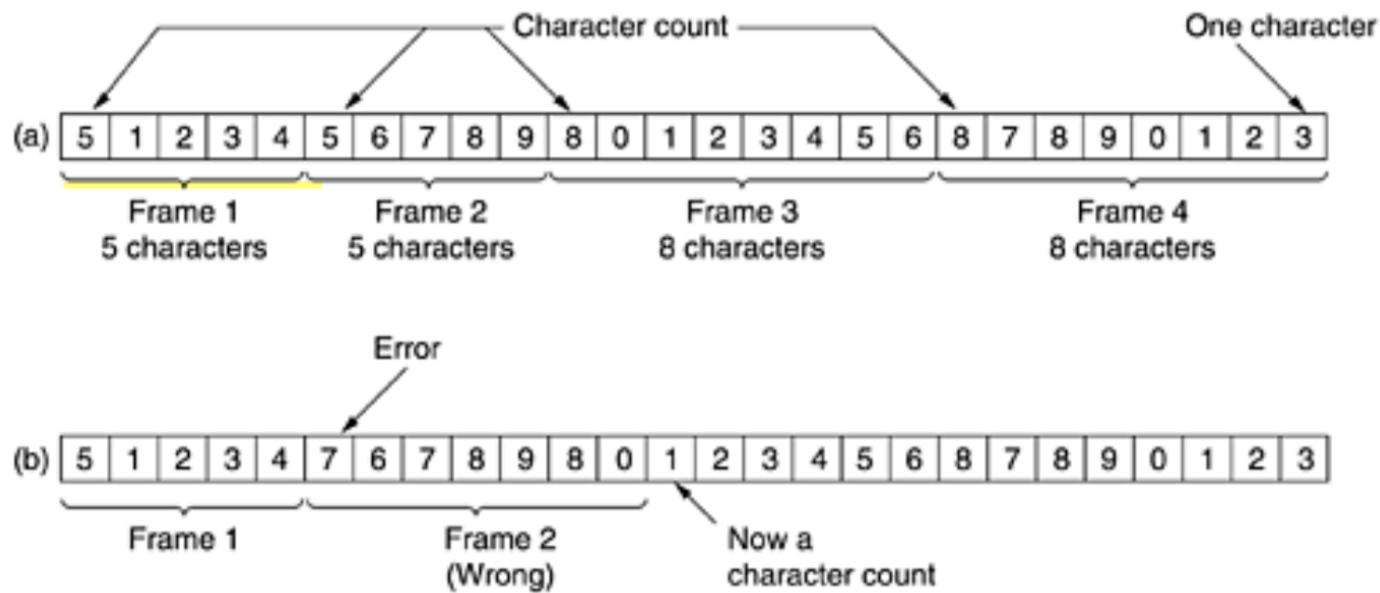
Length Fields (4/4)

- A special field is required.
- The field specifies the length (in bits) of data following the header.
- To specify precisely the length of n bits of data you must specify n !
- For that you need $\lceil \log n \rceil$ bits.



Length Fields: Example

- Uses a field in the header to specify the number of characters in the frame.



- When the data link layer at the destination sees the character count, it knows how many characters follow and hence where the end of the frame is.

Error in flag

Length Fields: Example

- This technique is shown for four frames of sizes 5, 5, 8, and 8 characters, respectively.
- The count can be garbled by a transmission error.
 - For example, if the character count *of 5* in the second frame becomes a 7, the destination will get out of synchronization and will be unable to locate the start of the next frame.

Length Fields: Example

- Some kind of error control is required!
- Even if the checksum is incorrect so the destination knows that the frame is bad, it still has no way of telling where the next frame starts.
- Sending a frame back to the source asking for a retransmission does not help either, since the destination does not know how many characters to skip over to get to the start of the retransmission.
- For this reason, the character count method is rarely used anymore.

You will put an error
detection / correction
mechanism in all schemes

No guarantee that you
will mitigate all errors!

You resolve errors at
a higher layer!

Request packet to
be sent again!

You make Sender/Receiver
employ ~~high~~ buffers.

ERROR DETECTION

DLC layer

A decision has to be made
about the length of the packet
to be used by the DLC

Outline

1. Error Discovery vs Recovery
2. Types of Errors
3. Vertical Redundancy (Parity)
4. Longitudinal Redundancy (2D-Parity)
5. CRC Codes
6. Checksums



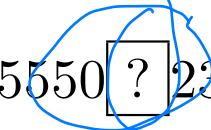
Discovery vs Recovery

Errors

- Physical layer produces a virtual bit pipe.
- Nyquist's theorem gives the signal frequencies which are sufficient to carry the signal rate.
- Shannon's theorem gives the optimal channel capacity under random noise but gives no idea on how to achieve it.
- Errors still occur due to other sources such as switching effects, cross talk, lightning, etc.
- Error recovery occurs at physical through transport layer but mainly at the data link layer

Transmitted Error Problem

- After meeting a new friend at a party, you want to get his/her cell phone number: ten digits.
- Your friend hands you the telephone number on a scrap of paper as

617 5550  ? 23 *binary*
617 5550 *  23 "location" has error
 we can correct if

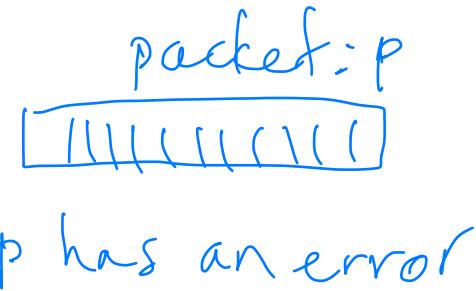
where ? means that the digit is unrecognizable, while * means digit is misrecognizable.

- How do you figure out the missing digit?
 - Call all possible ten numbers!
 - Ask for the number again!
 - Correct it yourself!

overhead
overhead
"overhead"

Error Recovery

- Error recovery takes on two forms as Error Detection and Error Correction, depending on the requirements of the application.
- **Error Detection:**
 - must be followed by retransmission:
 - most often used at higher levels, and
 - leads to retransmission strategies discussed later
- **Error Correction:**
 - most often used at the physical layer to produce bit pipe with low error rate



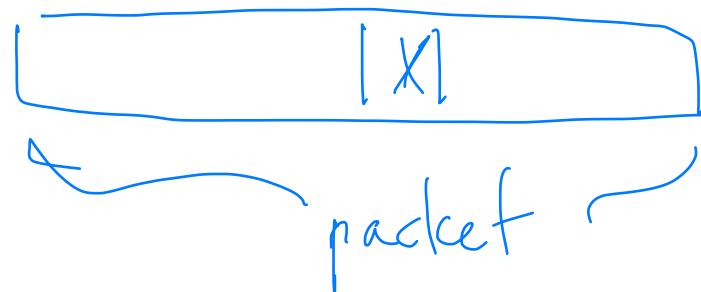
satellite communications
use "error correction"

Types of Errors

Types of Errors: Single bit error

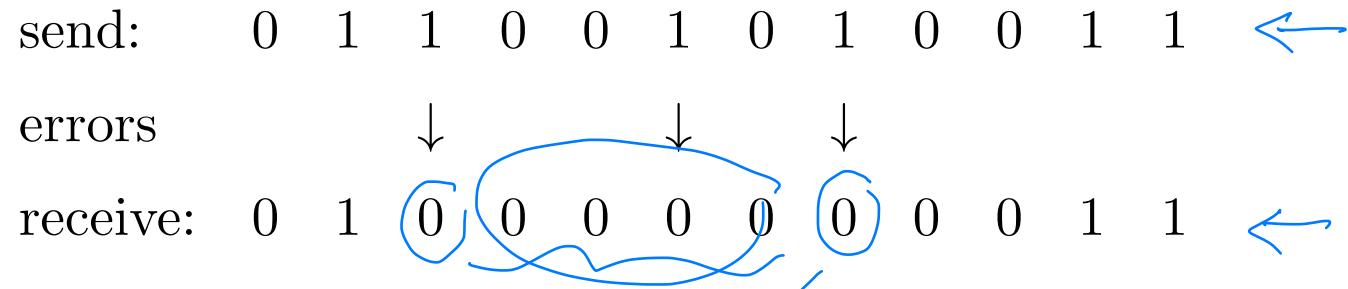
- **Single bit error:**

- Just a single bit changed:
from 0 to 1 or from 1 to 0.
- A single bit error may affect a larger block of data bits.
- They are more likely in parallel (because in parallel more bits are sent at the same time) than serial transmissions!

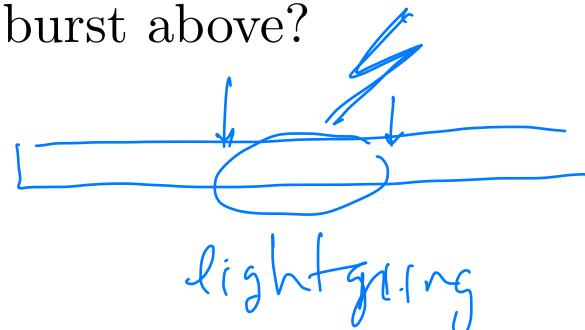


Types of Errors: Burst error

- **Burst Error:** is a contiguous block of at least two error bits (starting with an error bit and ending with an error bit).



- In a burst not all bits between two endpoints are in error!
- Length of a burst error is measured as the *distance* from the first corrupted to the last corrupted bit in this burst.
 - What is the length of the burst above?



Modeling errors

- Errors are notoriously difficult to model.
- Usually we look at the frequency of errors in an application.
 - **Error rate:** probability a bit is in error (bits are often assumed to be independent).
- **Typical error rates**
 - Wireless links: 10^{-4}
 - ISDN line: 10^{-6}
 - Optical fiber: 10^{-10}

bit
error
rate is $\frac{1}{10}$

Example

- A network channel has bit error rate p .
- How many errors do you expect in a packet of length n ?

$$pn$$



- Assume errors in a packet are independent of each other.
- What is the probability a packet of length n has an error?
 - The probability that a given bit is correct is $1 - p$.
 - The probability that all bits are correct is $(1 - p)^n$.

$$\Pr[\text{Packet has an error}] = 1 - \Pr[\text{Packet has no error}]$$

$$= 1 - (1 - p)^n$$

$$\approx 1 - 1/e, \approx 2/3$$

$$\left(1 - \frac{1}{n}\right)^n \underset{n \approx 2}{\approx} \frac{1}{e}$$

$\approx 2/3$

provided that $p = 1/n$, where e is Euler's number.

Example

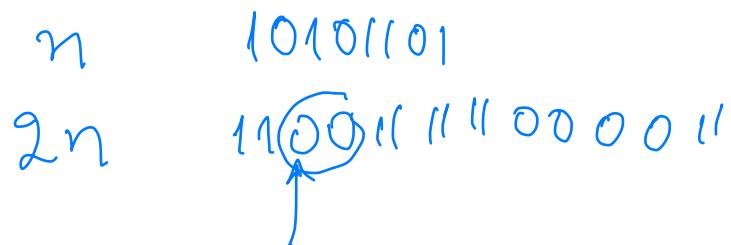
- So, if the bit error rate is $1/n$ then a packet of length n will have error with a “non-negligible” probability $\sim 1 - 1/e$.
- Following packet lengths for network types
 - Wireless links: packet length $n = 10^4$ bits
 - ISDN line: packet length $n = 10^6$ bits
 - Optical fiber: packet length $n = 10^{10}$ bits
- That's no good!

Different technology
→ "packet lengths"

Concept of Redundancy

- Error detection uses the concept of redundancy:
 - this means adding extra bits at the source in order to detect errors at the destination.
- **Example:** repeat every bit twice.
 - Receiver will do a bit-by-bit comparison.
 - Error detection system is good but it is not efficient.
- Rather than repeat the whole string twice a shorter stream of bits could be appended.

n 10101101
 $2n$ 11001111000011



Modelling errors

- A typical error detection/correction algorithm performs an operation

word → code.

transforming the original word into a code word (or code for short).

- We measure efficiency with the **Redundancy**

$$\text{Redundancy} = \frac{\text{length of code}}{\text{length of word}}$$

Code = "word" plus "redundant"

Parity

XOR or mod2

- If $b, b' \in \{0, 1\}$ are bits

$$b \oplus b' = \begin{cases} 0 & \text{if } b = b' = 0 \text{ or } b = b' = 1 \\ 1 & \text{otherwise} \end{cases}$$

$$\textcircled{1} \oplus \textcircled{1} = \textcircled{1}$$

$$1 \oplus 1 = 0$$

- Sometimes we also write $b \oplus b' \bmod 2$.
 - Sometimes we also use it for sequences of bits

$$\underbrace{(x_1 x_2 \cdots x_n)}_{\text{Product}} \oplus (y_1 y_2 \cdots y_n) = (x_1 \oplus y_1)(x_2 \oplus y_2) \cdots (x_n \oplus y_n)$$

- Example:

$$(01101) \oplus (10011) = 11110$$

VRC: Vertical Redundancy (or Parity) Check

- Based on bit XORing a sequence $x_1 x_2 \cdots x_n$.
- Single bit equal to the exclusive-or of the bits is added,

$$\text{Parity } (x_1 x_2 \cdots x_n) = x_1 \oplus x_2 \oplus \cdots \oplus x_n$$

Sum is invariant
under two errors

$$\left(\sum_{i=1}^n x_i \right) \text{ mod } 2.$$

- If number of 1 bits is even result is 0, otherwise 1
- Given string x of length n , the codeword of x is a string of length $n + 1$ defined by $C(x) = \underbrace{x}_{\sim} \text{Parity}(x)$
- Detects a single error

$$x' \text{ Parity}(x')$$

VRC Check: Example

- Example:

word: 00101010

check bit: 1

code: 001010101

- When the receiver “receives the code” transmitted through the channel, it checks whether or not the sum of the bits is equal to 0 modulo 2.

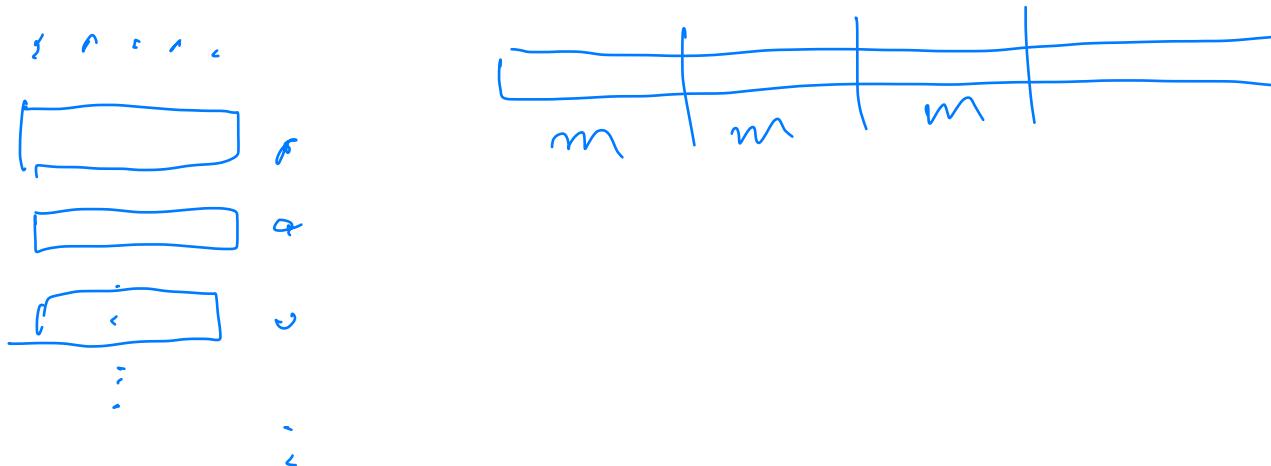
$$n \rightarrow n+1$$

$$\frac{n+1}{n}$$

2D Parity

LRC: Longitudinal Redundancy (or 2d-Parity) Check

- Bits placed in $m \times n$ array with $m, n \geq 2$.
Rows are sent one after the other.
One parity bit used for each row and column (total of $m + n$ bits).
- Detects up to 2 errors
- Corrects a single error
- Often used with ASCII stream ($m = 8$)



LRC Check: Example

- Consider a sequence of 35 bits:

10010100111010111000110001110011001

- Arrange the sequence as a 5×7 matrix and add the check bits (one for each row and column).

1	0	0	1	0	1	0	1
0	1	1	1	1	0	1	0
1	1	1	0	0	0	1	0
1	0	0	0	1	1	1	0
0	0	1	1	1	0	1	1
<hr/>							1
1	0	1	1	1	1	1	1

$5 \times 7 + 5 + 7$
 5×7

- Transmit as a sequence.

LRC Check: Example

- The receiver arranges the sequence into a matrix

1	0	0	1	0	1	0	1
0	1	1	1	0	1	0	0
1	1	0	0	0	0	1	0
1	0	0	0	1	1	1	0
0	0	1	1	0	0	1	1
<hr/>							
1	0	1	1	1	1	1	1

A blue arrow points from the left towards the third column, indicating the column being checked. A blue circle highlights the '0' in the third column of the fourth row. A blue arrow points from the right towards the fourth column, labeled 'error'. A blue arrow points upwards from the bottom of the fourth column, labeled 'error'.

and checks the condition.

- There is a single error! Can you locate the error?

LRC Check: Example

- The receiver arranges the sequence into a matrix

1	0	0	1	0	1	0	1
0	1	0	0	0	1	0	0
1	1	1	0	0	0	1	0
1	0	0	0	1	1	1	0
0	0	1	1	0	0	1	1
<hr/>							
1	0	1	1	1	1	1	1

and checks the condition.



- Here there are two errors!
- Can you locate the errors? No!
- You can only detect that two errors occurred!

LRC Check: Example

- It is even possible you will not notice any error!!!

1	0	0	1	0	1	0	1
0	1	1	1	0	1	0	0
1	1	1	0	0	0	1	0
1	0	0	0	1	1	1	0
0	0	1	1	0	0	1	1
<hr/>							
1	0	1	1	1	1	1	1

- If all four bits in “boxes” are in error you will not notice anything!

CRC

Cyclic Redundancy Check Codes

- Based on the theory of cyclic error-correcting codes.
- Using cyclic codes, encode messages by adding a fixed-length check value, for the purpose of error detection in communication networks.
 - First proposed by W. Wesley Peterson in 1961.
- Most commonly used error detection scheme in use are Cyclic Redundancy Check (CRC) codes.

Polynomials in Z_2

$$Z_2 = \{0, 1\}$$

- A polynomial is an expression that can be built from constants and symbols called variables by means of addition, multiplication and exponentiation to a non-negative power.
variable
- A polynomial in a single indeterminate x can always be written (or rewritten) in the form

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0,$$

where a_0, \dots, a_n are constants and x is the variable.

- This can be expressed more concisely by using summation notation:

$$\sum_{i=0}^n a_i x^i$$

$$3x^2 + 2x^{-2} \\ x^5 - 4x^{-3}$$

Polynomials in Z_2

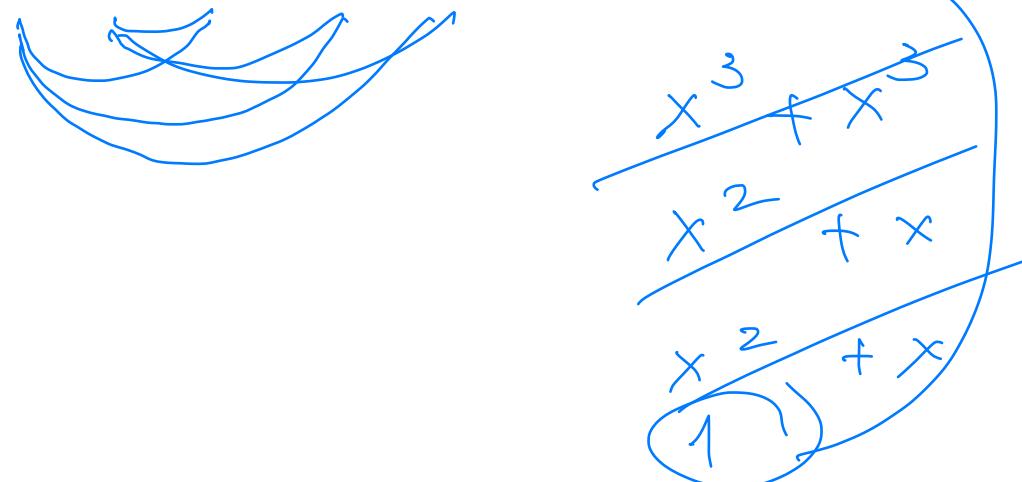
- Two such polynomial expressions may be added, multiplied, divided, etc
- Example: Addition

$$(x^3 + x + 1) + (x^2 + x + 1) = x^3 + x^2$$

In Z_2 $2 = 0$
 $3 = 1$
 $4 = 0$

- Example: Multiplication

$$(x^3 + x + 1) \cdot (x^2 + x + 1) = x^5 + x^4 + 1$$



Cyclic Redundancy Check Codes: Definitions

- L : length of check bits. *check bits*
- K : length of data bits. *data*
- Bit strings are treated as polynomials over Z_2 .
- A bit string $s_{K-1}s_{K-2}\cdots s_1s_0$ is encoded as a polynomial in Z_2 :

$$s(x) = s_{K-1}x^{K-1} + s_{K-2}x^{K-2} + \cdots + s_1x + s_0$$

- Addition and multiplication of polynomials is done mod 2

Polynomials and Bit Sequences

- In the transformation between polynomials and bit sequences “missing coefficients of the polynomial” must be included in the bit sequence as 0s.

not

$$\begin{array}{c} x^7 + x^5 + x^2 + x + 1 \\ \downarrow \\ x^7 + 0x^6 + x^5 + 0x^4 + 0x^3 + x^2 + x + 1 \\ \downarrow \\ 10100111 \end{array}$$

- Conversely, 0s in the bit sequence are missing coefficients in the polynomial.

Polynomials

- A monomial is a number times a power of x : ax^n
 - $3x^2, 2x^7, 8$ are all monomials.
- A polynomial is a sum or difference of monomials
 - $4x^5 - 3x^2 - 1, 4x^2, 2$ are all polynomials
- When we write $P(x) = 3x^3 - 2x^2 + 1$ we say “ P of x ”
- To add or subtract polynomials, we just collect like terms:
 - Example: $P(x) = x^2 + 3x + 5$ and $Q(x) = 4x^3 - 2x^2 + 3x - 2$
- How do we multiply polynomials?
 - Example: $P(x) = 3x + 5$ and $Q(x) = 4x^3 + 3x - 2$

Please
practice!

Polynomials in mod2 Arithmetic

Lets try to do all previous examples mod2

- A monomial is a number times a power of x :
 - $3x^2 = x^2, 2x^7 = 0, 8 = 0$ are all monomials.
- A polynomial is a sum or difference of monomials
 - $4x^5 - 3x^2 - 1 = x^2 - 1, 4x^2 = 0, 2 = 0$ are all polynomials
- To add or subtract polynomials, we just collect like terms:
 - Example: $P(x) = x^2 + 3x + 5 = x^2 + x + 1$ and
 $Q(x) = 4x^3 - 2x^2 + 3x - 2 = x$
- How do we multiply polynomials?
 - Example: $P(x) = 3x + 5 = x + 1$ and
 $Q(x) = 4x^3 + 3x - 2 = x$

Please practice!

Somehow, things get simpler mod2!

CRC Algorithm

1. **Input:** sequence of data bits of length K and a generator polynomial of degree L .

2. Append L bits (also called CRC bits) to the K data bits



in such a way that the resulting sequence of bits gives rise to a polynomial that is divisible by the generator polynomial.

3. Send these $K + L$ bits.

condition used for error detection

4. At the receiving end compute the data bits and check the error condition.



Computing CRCs

- **Data bits:** $s(x) = s_{K-1}x^{K-1} + \cdots + s_1x + s_0$

- Use a specially chosen function for generating check bits:

$$g(x) = x^L + g_{L-1}x^{L-1} + \cdots + g_1x + 1$$

Note: $g_L = g_0 = 1$

- Divide $s(x)x^L$ by $g(x)$ and set

$$c(x) = \text{Remainder in division } \left[\frac{s(x)x^L}{g(x)} \right]$$

$$\deg(c(x)) < L = \deg(g(x))$$

- **Check bits:** $c(x) = c_{L-1}x^{L-1} + \cdots + c_1x + c_0$

- **Codeword:**

$$y(x) = s(x)x^L + c(x)$$

$$= s_{K-1}x^{L+K-1} + \cdots + s_0x^L + c_{L-1}x^{L-1} + \cdots + c_0$$

$$g_0 = 1$$

If $g_0 = 0$ then
then is
divisible
by x
Hence if
is not for
a general

Example: CRC (1/2)

data $\downarrow 101$

$k = 2$

- $s(x) = x^2 + 1, g(x) = x^3 + x^2 + 1$
- $c(x) = \text{Remainder in division}$ $\left[\frac{(x^2+1)x^3}{x^3+x^2+1} \right]$
- Elementary division gives that $c(x) = x^2 + x.$

generator $L = 3$

Euclidean Algorithm

• Codeword is $y(x) = s(x)x^3 + c(x) = x^5 + x^3 + x^2 + x$

CRC $\boxed{101110}$

Example CRC (2/2)

		110 (Quotient)
Generator	Message	
$1101 (= x^3 + x^2 + 1)$	$101000 (= x^5 + x^3)$	
Addition mod2 →	1101	
	1110	
Addition mod2 →	1101	
	0110 (Remainder)	
	Check bits	

Another Example: Polynomial Division (1/2)

- For $D = 1010001101$ we have

$$D(X) = X^9 + X^7 + X^3 + X^2 + 1$$

- For $P = 110101$ we have

$$P(X) = X^5 + X^4 + X^2 + 1$$

- When we divide $D(X)$ by $P(X)$ we should end up with $R = 01110$, which corresponds to $R(X) = X^3 + X^2 + X$
- Lets check why this is true.

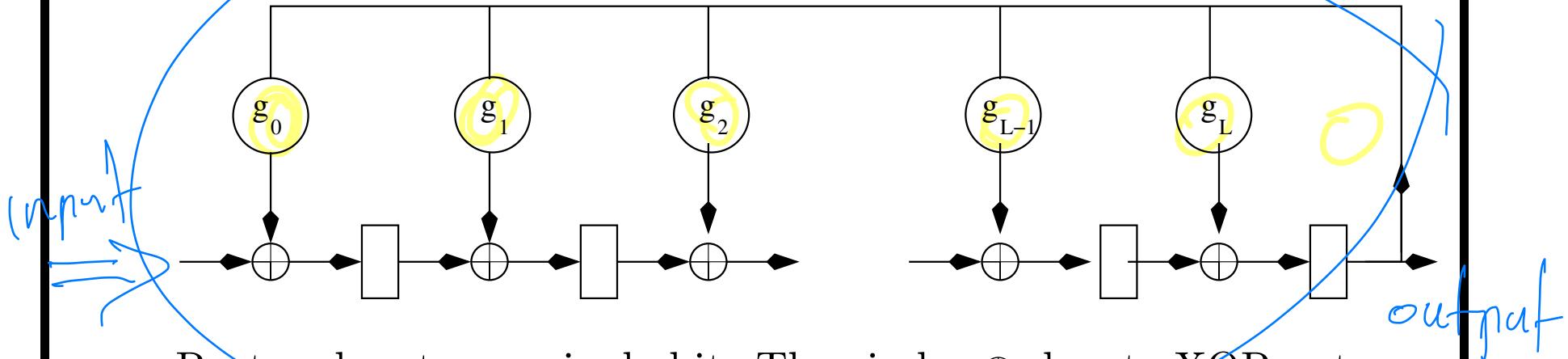
Another Example: Polynomial Division (2/2)

- Let $D(X) = X^9 + X^7 + X^3 + X^2 + 1$, $P(X) = X^5 + X^4 + X^2 + 1$
- Hence, $X^5 D(X) = X^{14} + X^{12} + X^8 + X^7 + X^5$
- Then the division $\frac{X^5 D(X)}{P(X)} = Q(X) + \frac{R(X)}{P(X)}$ yields
generator $\frac{X^5 D(X)}{P(X)}$

$$\begin{array}{r}
 \begin{array}{c}
 X^9 + X^8 + X^6 + X^4 + X^2 + X \\
 \hline
 X^{14} & X^{12} & & X^8 + X^7 + & X^5 \\
 \hline
 X^{14} + X^{13} + & X^{11} + & X^9 \\
 \hline
 X^{13} + X^{12} + X^{11} + & & X^9 + X^8 \\
 \hline
 X^{13} + X^{12} + & X^{10} + & X^8 \\
 \hline
 X^{11} + X^{10} + X^9 + & & X^7 \\
 \hline
 X^{11} + X^{10} + & X^8 + & X^6 \\
 \hline
 & X^9 + X^8 + X^7 + X^6 + X^5 \\
 \hline
 X^9 + X^8 + & X^6 + & X^4 \\
 \hline
 X^7 + & X^5 + X^4 \\
 \hline
 X^7 + X^6 + & X^4 + & X^2 \\
 \hline
 X^6 + X^5 + & & X^2 \\
 \hline
 X^6 + X^5 + & X^3 + & X \\
 \hline
 & X^3 + X^2 + X
 \end{array} \\
 \leftarrow Q(X) \\
 \leftarrow X^5 D(X)
 \end{array}$$

Shift-Register

- Division easily computed in hardware using shift register circuit



- Rectangles store a single bit. The circles \oplus denote XOR gates. Big circles indicate multiplication by g_i .
- Register loaded with L bits: $s_{K-L+1} \cdots s_{K-1}s_K$, with s_K first.
- At each clock pulse a new bit of $s(x)$ comes in at the left. Register reads in corresponding mod2 sum of feedback plus contents of previous stage. After K shifts the switch to the right moves to the horizontal position and the CRC is read out.

Properties of CRC

Why are all code words divisible by $g(x)$, and vice versa?

- **sender** computes the code $c(x)$ from $s(x)$ and transmits
 $y(x) = s(x)x^L + c(x)$

- Computation is done as follows: Let $z(x)$ be quotient of $s(x)x^L/g(x)$, i.e., $s(x)x^L = g(x)z(x) + c(x)$

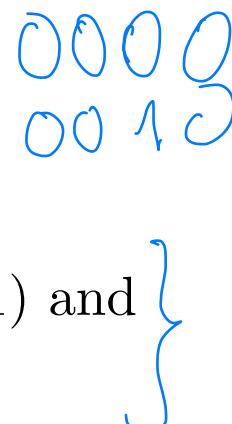
- Since subtraction is the same as addition mod 2 we get mod2

$$\begin{aligned}
 y(x) &= s(x)x^L + c(x) && \leftarrow && c(x) = -c(x) \\
 &= s(x)x^L - c(x) && \leftarrow && \\
 &= g(x)z(x) && \leftarrow && 15 = 5 \cdot 3
 \end{aligned}$$

Hence, $g(x)$ divides $y(x)$.

- **Recall:** divisibility by $g(x)$ was our error detection condition!

Detecting Single Errors

- Assume receiver gets $w(x) = \underbrace{y(x)}_{\text{Code word}} + \underbrace{e(x)}_{\text{errors' polynomial}}$, where $e(x)$ represents the errors' polynomial.
- Receiver calculates remainder and if result is zero then accepts, otherwise detects error.
- Can a single error be undetected?
 - Code word $y(x)$ is divisible by $g(x)$.
 - Undetected means $e(x)$ divisible by $g(x)$.
 - Single error implies $e(x) = x^i$ for some i
 - But $g(x)$ has at least two non-zero terms $(x^L, 1)$ and therefore so must $e(x)$.
- It follows that single errors are detected!

Selection of CRCs

- Most important part of implementing the CRC algorithm is the selection of generator polynomial.
- Polynomial is chosen so as to maximize the error-detecting capabilities while minimizing overall collision probabilities.
- Most important attribute is length (largest degree(exponent) +1 of any one term in the polynomial), because of its direct influence on the length of the computed check value.
- Design of the CRC polynomial depends on
 - max length of block to be protected (data + CRC bits),
 - desired error protection features,
 - type of resources for implementing the CRC, and
 - desired performance

Summary of CRCs

Several CRC polynomials have been adopted by the International Telecommunication Union (ITU) and the Consultative Committee for International Telephony and Telegraphy (CCITT).

- CRC-8 (Used in ATM headers): $x^8 + x^2 + x + 1$
- CRC-16 (Used in HDLC): $g(x) = x^{16} + x^{15} + x^2 + 1$
- CRC-CCITT: $g(x) = x^{16} + x^{12} + x^5 + 1$
- CRC-32 (Used in LANs):

$$\begin{aligned} g(x) &= x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + \\ &\quad x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + 1 \end{aligned}$$

generators

Checksums

Checksum

Sender:

1. Divide data $\boxed{B_1 \cdots B_k}$ into k blocks B_1, \dots, B_k each of a fixed size n bits (usually $n = 16$).
2. Append to $B_1 \cdots B_k$ the sum S (**checksum**) modulo $(2^n - 1)$.
3. send $\boxed{B_1 \cdots B_k \mid S}$

Receiver:

1. Detach blocks B_1, \dots, B_k and S from $\boxed{B_1 \cdots B_k \mid S}$
2. Check that

$$\sum_{i=1}^k B_i \equiv S \bmod (2^n - 1)$$

Checksum: Example

Let the input have twenty bits and let the block size be $n = 5$.

Input: 01001110110101111110

1. Break into 4 Blocks: 01001 11011 01011 11110

2. Convert each block to integers: 9 27 11 30

3. Take the sum mod($2^5 - 1$):

$$9 + 27 + 11 + 30 \equiv 77 \equiv 15 \text{ mod } (2^5 - 1)$$

4. Convert to binary (block of 5 bits): 01111

Output (append value to input):

01001 11011 01011 11110 01111

This is the output transmitted by the sender.

ERROR CORRECTION

Outline

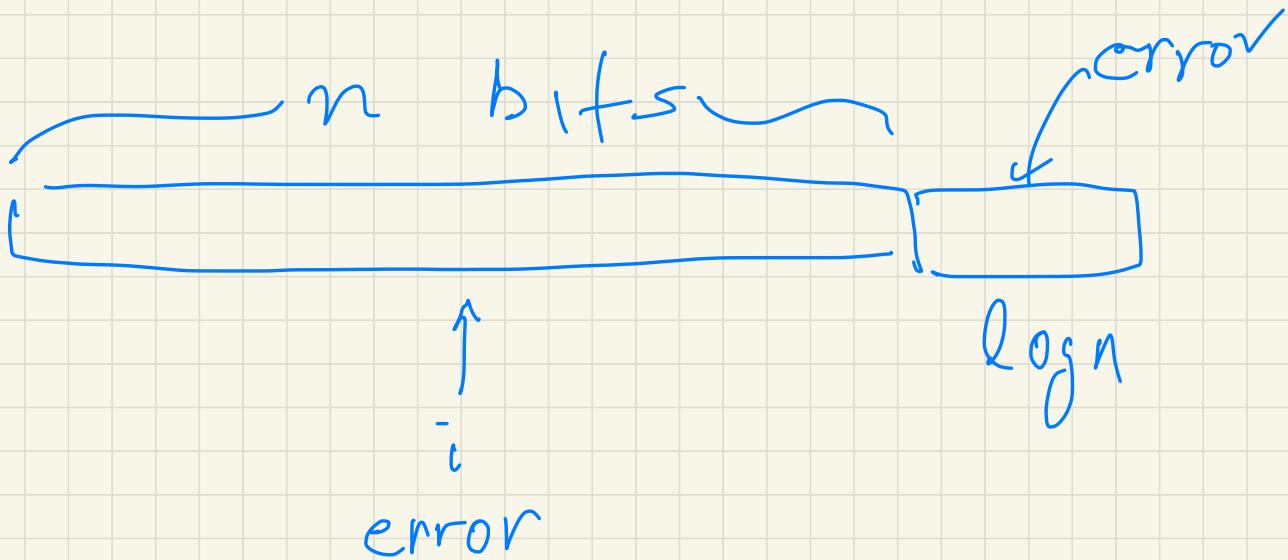
1. Error Correction vs Error Detection (syndrome)
2. Message Geometry (Hypercube)
3. Hamming Distance
4. Redundancy Bits (syndrome)
5. Hamming Code (Not Required Material)

Correction vs Detection

Error Correction

- Error correction, like error detection, uses the idea of redundancy.
 - However, this time more redundancy is needed! Why?
- Because error correction is more difficult than error detection!
 - Not only you must detect that an error occurred!
 - You have to correct it, as well!
- To correct a bit-error it is enough to locate it! Why?
- Because
 - If you can locate the error, then “flip” the bit at that location!
 - So, this is ok because we use binary data representation.

packet



You need $\log n$ to identify
location of i^2 error

Message Space

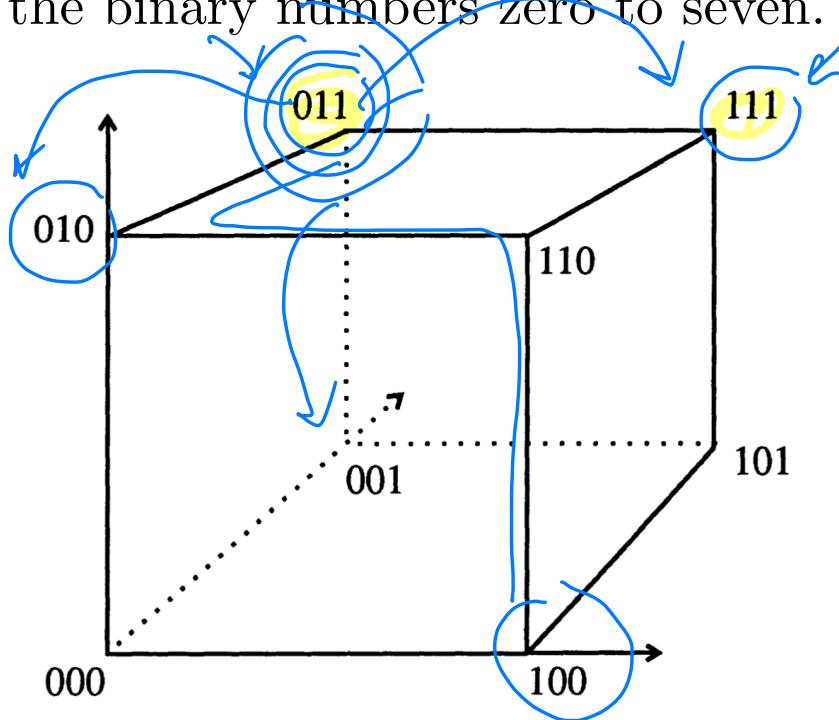
Messages

- Message space is made up of the messages that we want to transmit (also known as words).
- We are used to thinking of a space as something which can be many-dimensional, either continuous or discrete, and whose points can be labeled by coordinates.
- Message space is a multidimensional discrete space, some or all of whose points correspond to messages.
- To make matters a little more concrete, consider a three-bit binary code, with acceptable words:

000, 001, 010, 011, 100, 101, 110, 111

Space

- These are just the binary numbers zero to seven.



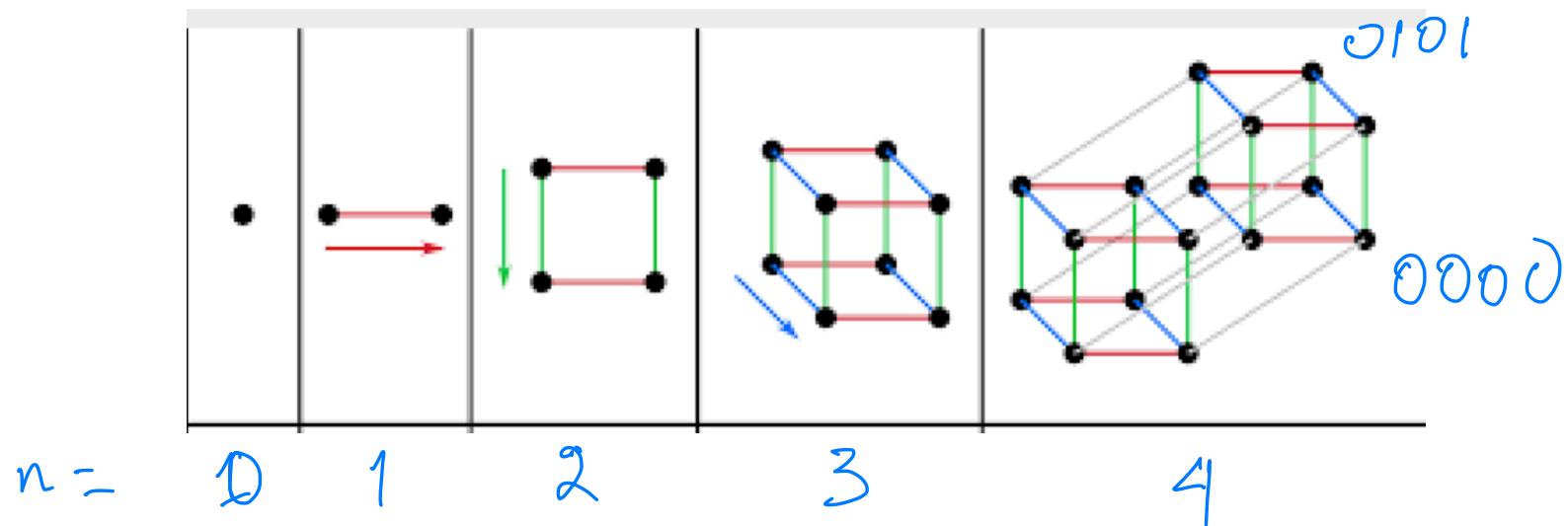
3-d(m)
hypercube

n-dim
 2^n corners

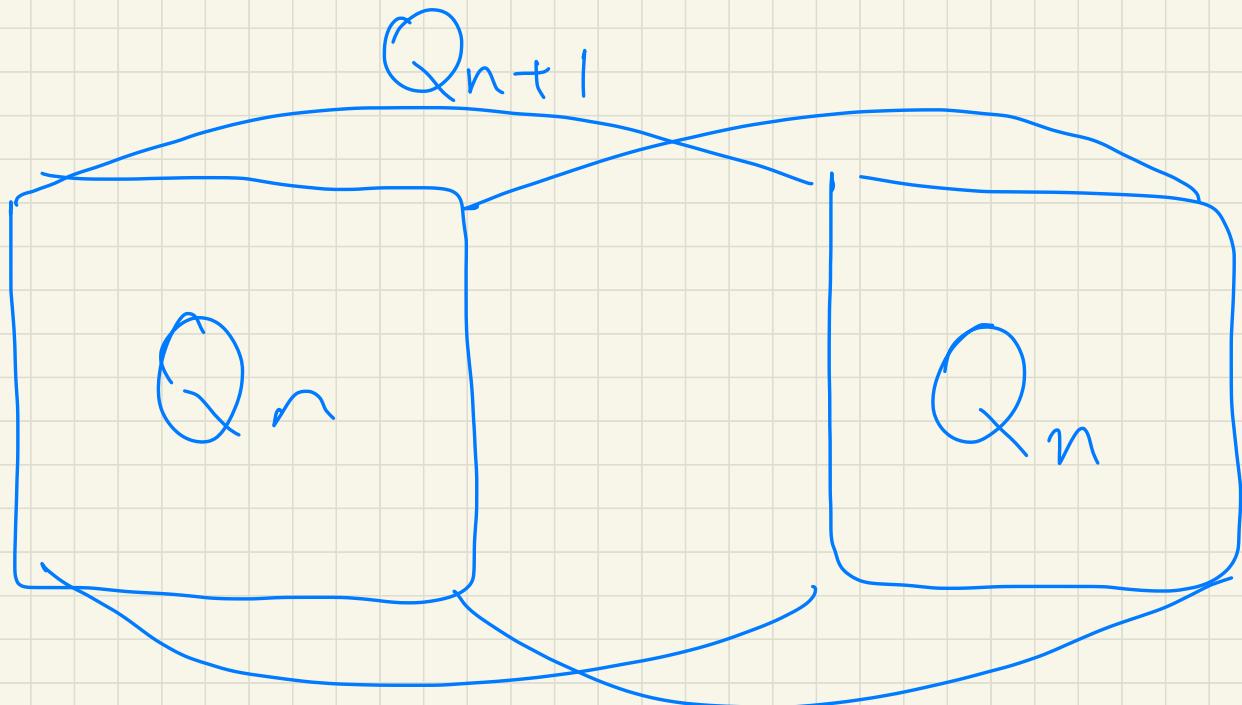
- We can consider these numbers to be the coordinates of the vertices of a cube in three-dimensional space

Space

- This cube is the message space corresponding to the three-bit messages.
- The only points in this space are the vertices of the cube - the space between them in the diagram, the edges, and whatnot are not part of it.
- There is also the n -cube corresponding to n -bit messages: it is also called the hypercube of dimension n .

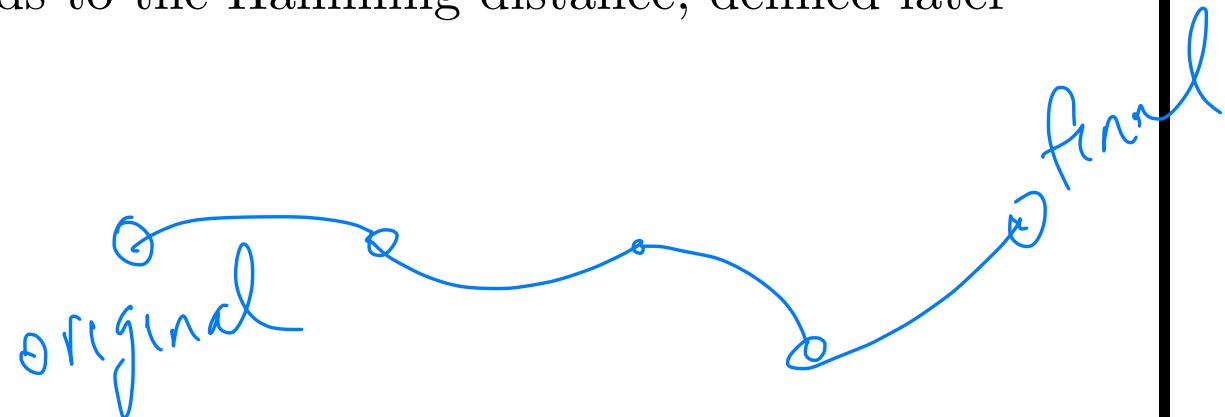


n - Hypercube : Q_n



Errors in Space

- What happens if there is an error in transmission?
- This will change the bits in the sent message, and correspond to moving us to some other point in the message space.
- Intuitively, it makes sense to think that the more errors there are, the “further” we move in message space;
- in the diagram above, (111) is “further” from (000) than is (001) or (100).
- This of course leads to the Hamming distance, defined later



Hamming Distance

How Do We Handle Errors?

- Normally, a code (also called frame) consists of m data (i.e., message) bits and r redundant, or check, bits.
- Let the total length be n (i.e., $n = m + r$): message plus redundancy bits.
- An n -bit unit containing data and check bits is often referred to as an n -bit codeword (also known as code).
- Given any two codewords, say, 10001001 and 10110001 , it is possible to determine how many corresponding bits differ.
 - In this case, 3 bits differ.
- To determine how many bits differ, just XOR the two codewords and count the number of 1 bits in the result.

$$(10001001) \oplus (10110001) = 00111000$$

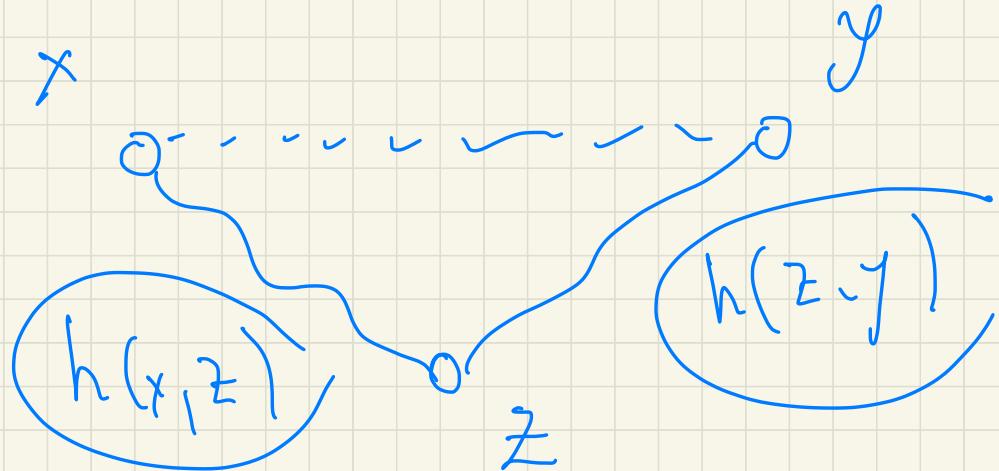
Hamming Distance

$h(x, y)$ = Hamming distance
of x, y .

$$h(x, y) = 0 \iff x = y$$

$$h(x, y) = h(y, x)$$

$$h(x, y) \leq h(x, z) + h(z, y)$$



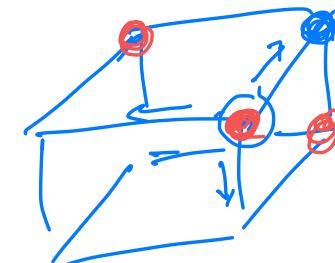
$$h(x, y) \leq h(x, z) + h(z, y)$$

Number of Different Bits

- The number of bit positions in which two codewords differ is called the Hamming distance (Hamming, 1950).
 - 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.
- The Hamming distance of 10001001 and 10110001 is 3.
- **NB.** 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 ($= 2^{m+r}$) possible codewords are used.

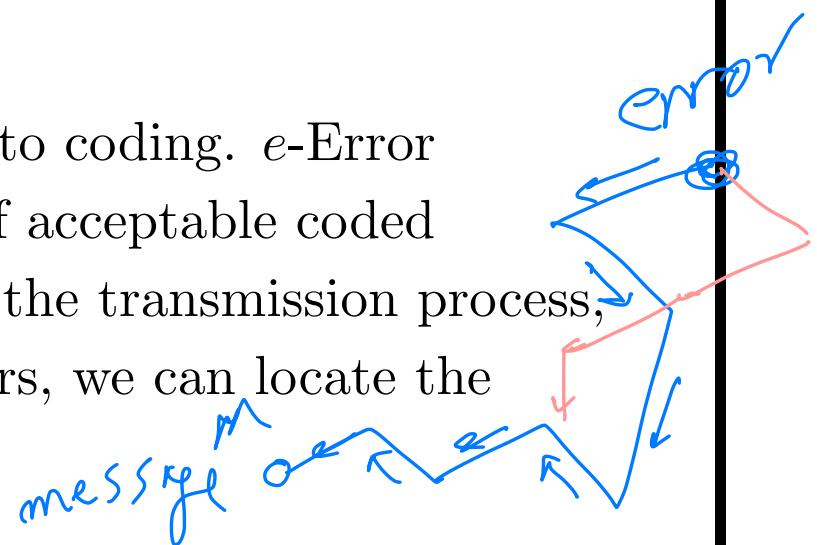
Distance and Space

- The notion of distance is useful for discussing errors.
- Clearly, a single error moves us from one point in message space to another a Hamming distance of one away; a double error puts us a Hamming distance of two away, and so on.
- For a given number of errors e we can draw about each point in our hypercubic message space a “sphere of error”, of radius e , which is such that it encloses all of the other points in message space which could be reached from that point as a result of up to e errors occurring.
- This gives us a nice geometrical way of thinking about the coding process.

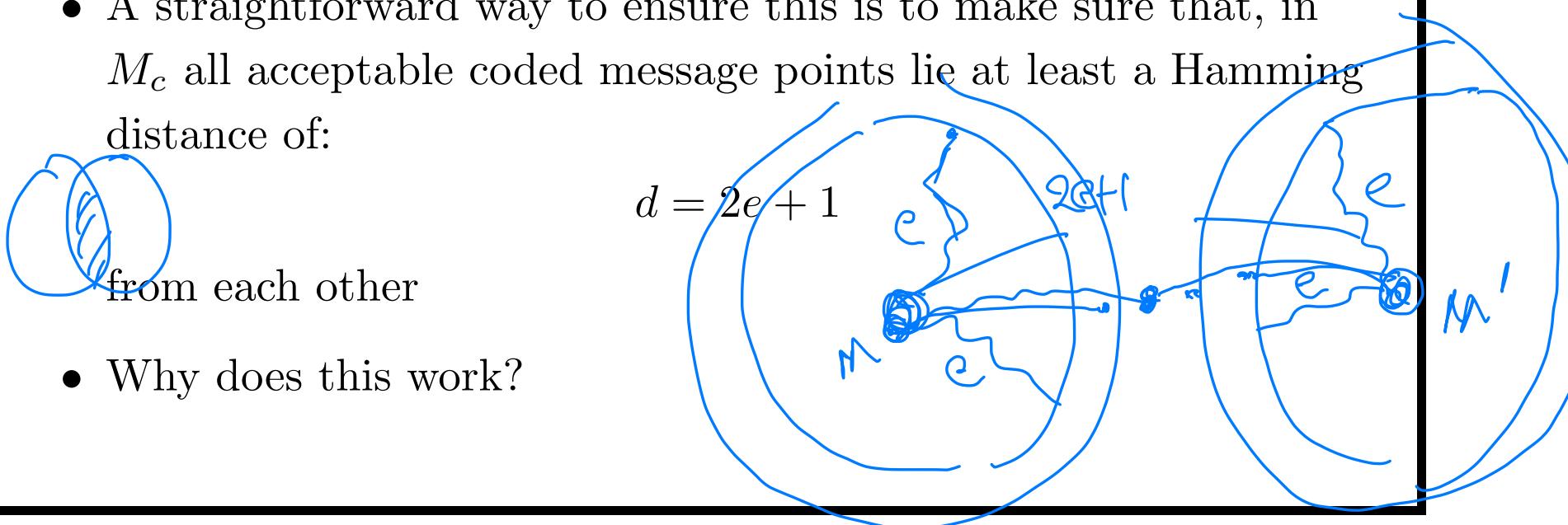


Distance and Space

- Whenever we code a message M , we rewrite it into a longer message M_e .
- We can build a message space for M_e just as we can for M ; of course, the space for M_e will be bigger, having more dimensions and points.
- Clearly, not every point within this space can be associated one-on-one with points in the M -space; there is some redundancy.
- This redundancy is actually central to coding. e -Error correction involves designing a set of acceptable coded messages in M_e such that if, during the transmission process, any of them develops at most e errors, we can locate the original message with certainty.



Distance and Space

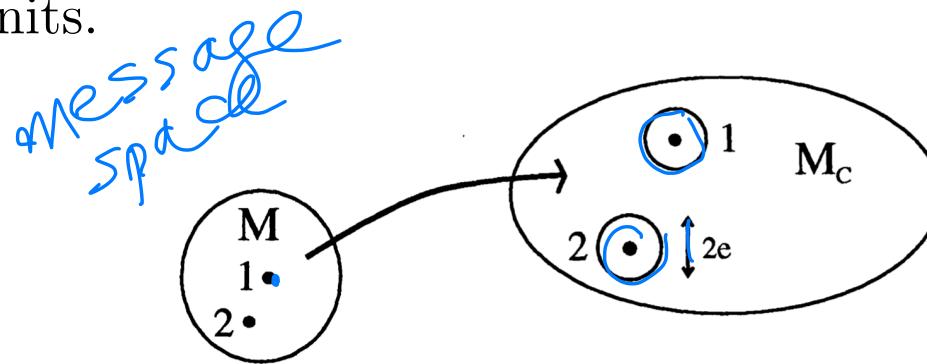
- In our geometrical picture, acceptable messages correspond to certain points within the message space of M_c
- Errors make us move to other points, and to have error correction we must ensure that if we find ourselves at a point which does not correspond to an acceptable message, we must be able to backtrack, uniquely, to one that does.
- A straightforward way to ensure this is to make sure that, in M_c all acceptable coded message points lie at least a Hamming distance of:

from each other
- Why does this work?

Distance and Space

- Suppose we send an acceptable message M , and allow e errors to occur in transmission.
- The received message M' will lie at a point in M_e which is e units away from the original.
- How do we get back?
- Because of the separation of $d = 2e + 1$ we have demanded, M is the closest acceptable message to M' : all other acceptable messages must lie at a Hamming distance $\geq e + 1$ from M' .
- Note that we can have simple error detection more cheaply; in this case, we can allow acceptable points in M_e to be within $2e$ of one another.
- The demand that points be $(2e + 1)$ apart enables us to either correct e errors or detect $2e$.

Distance and Space

- Each element of M is associated with a point in M_e such that no other acceptable points lie within a Hamming distance of $2e + 1$ units.



- We can envisage the space for M_e as built out of packed spheres, each of radius e units, centered on acceptable coded message points
- If we find our received message to lie anywhere within one of these spheres, we know exactly which point corresponds to the original message.

Example: Error Detection

- Consider a code in which a single parity bit is appended to the data.
- The parity bit is chosen so that the number of 1 bits in the codeword is even (or odd).
 - For example,
 - * in even parity, when 1011010 is sent a bit is added to the end to make it 10110100
 - * in odd parity, when 1011000 is sent a bit is added to the end to make it 10110001.
- A code with a single parity bit has a distance 2, since any single-bit error produces a codeword with the wrong parity. It can be used to detect single errors.

Redundancy Bits

lets put everything together
to see how many redundancy
bits we should add
per message

Error Correction: # of Redundancy Bits

- If we have m bits of data and r bits of redundancy (this is called the *syndrome*), $m + r$ bits are transmitted.
- If we decide that a vanishing syndrome is to represent no error, that leaves at most $(2^r - 1)$ message error positions that can be coded. However, errors can occur in the syndrome as well as the original message we are sending. $m + r$
- We need to determine the location of any of these $m + r$ bits:
 - these r bits must be able to indicate any of the $m + r$ positions!

m bit locations

r bit locations

Basic Idea
of Hamming

- Hence, r must be chosen so that we have

$$2^r - 1 \geq m + r$$

Requirements of Error Correction: Condition $2^r > m + r$

# Data Bits m	# Redundancy Bits r	# Total Bits $m + r$
1	2	$3 \leftarrow 2^r > 1+r$
2	3	$5 \leftarrow 2^r > 2+r$
3	3	$6 \leftarrow$
4	3	7
5	4	9
6	4	10
7	4	11
16	5	$21 \leftarrow 2^r > 16+r$
32	6	38
\vdots	\vdots	

Example

- If we wanted to send a message 11 bits in length, we would have to include a syndrome of at least four bits, making the full message fifteen bits long.
- This does not seem particularly efficient
 - efficiency = $11/15$ or about 70%.
- However, if the original message was, say, 1000 bits long, we would only need ten bits in our syndrome ($2^{10} = 1024$) which is a considerable improvement!

Hamming Codes

(Not Required)

will not
be covered

Idea of Hamming Code

- Use extra parity bits to allow the identification of a single error.
- Create the code word as follows:
 1. Mark all bit positions that are powers of two as parity bits.
 - positions: 1, 2, 4, 8, 16, 32, 64, etc.
 2. All other bit positions are for the data (message) to be encoded.
 - positions: 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17, etc.
 3. Each parity bit calculates the parity for some of the bits in the code word.
 4. The position of the parity bit determines the sequence of bits that it alternately checks and skips.

$$7 \rightarrow 7 + 4$$

Error Correction: Example of ASCII Characters

- ASCII characters consist of 7 bits.
- To correct an error on an ASCII character the algorithm must determine which of the seven bits has changed.
- An ASCII character has 7 bits.
 - Three redundancy bits are enough to identify any one of seven positions.
- What if an error occurs in a redundancy bit?

Hamming Code

- The **Hamming code** provides a practical solution to the error correction problem.
- For simplicity, in the sequel we discuss the case of ASCII character bit strings (these are bit strings of length 7).
 - By the previous discussion we must choose r so that $2^r > 7 + r$.
 - The minimum possible such value of r is 4 and so we must use four redundancy bits.
- It can be designed to work to data units of any given length.

Hamming Code for ASCII Characters

- Where do you locate the four redundancy bits?
- To form the eleven bits of the Hamming code redundancy bits are placed in positions $1 = 2^0, 2 = 2^1, 4 = 2^2, 8 = 2^3$:

input		d		d	d		d	d	d
insertions	2^0	2^1		2^2			2^3		
position	1	2	3	4	5	6	7	8	9
	↓	↓		↓			↓		
type of bit	r_1	r_2	d	r_4	d	d	d	r_8	d
									d

- r with subscripts indicate the redundancy bit, and d the data bit.

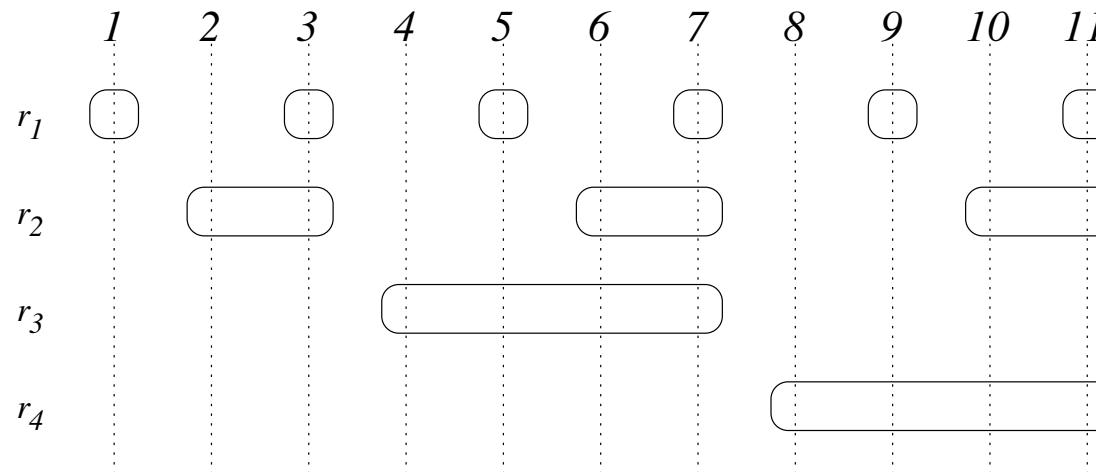
Hamming Code for ASCII Characters: Example

- For the sequence 1010110 of seven bits we have the following Hamming code

		1		0	1	0		1	1	0
r_1	r_2	1	r_4	0	1	0	r_8	1	1	0

Hamming Code for ASCII Characters

- The redundancy bits are essentially parity bits computed in a special way.
- What are the values of the redundancy bits?
- Take as parity bit the XOR of the bits in positions indicated below!



Hamming Code for ASCII Characters

redundancy bit	positions used for parity
r_1	parity check bits of $\boxed{1}, 3, 5, 7, 9, 11$
r_2	parity check bits of $\boxed{2}, 3, 6, 7, 10, 11$
r_4	parity check bits of $\boxed{4}, 5, 6, 7$
r_8	parity check bits of $\boxed{8}, 9, 10, 11$

Hamming Code for ASCII Characters: Example

- For example for the sequence $r_1 \boxed{r_2} 1 \boxed{r_4} 010 \boxed{r_8} 110$ we obtain the following equations

$$0 = r_1 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0$$

$$0 = r_2 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0$$

$$0 = r_4 \oplus 0 \oplus 1 \oplus 0$$

$$0 = r_8 \oplus 1 \oplus 1 \oplus 0$$

- Solving these equations we obtain

$$r_1 = 0, r_2 = 1, r_4 = 1, r_8 = 0$$

Example of Hamming Code

- Consider the sequence of bits 1011001

1	2	3	4	5	6	7	8	9	10	11
↓	↓	↓					↓			
r	r	1	r	0	1	1	r	0	0	1
0	r	1	r	0	1	1	r	0	0	1
0	1	1	r	0	1	1	r	0	0	1
0	1	1	1	0	1	1	r	0	0	1
0	1	1	1	0	1	1	0	0	0	1

- So the sender transmits the sequence 01110110001
- By independence of vectors of positions it is possible to locate error! How is this done?

Locating Errors in a Hamming Code

- In previous example, suppose 7th bit of message is in error, i.e., receiver receives 011101 $\boxed{0}$ 0001
- Receiver does not know there is an error, but recalculates the values r_1, r_2, r_4, r_8 .

1	2	3	4	5	6	7	8	9	10	11
↓	↓	↓				↓				
0	1	1	1	0	1	0	0	0	0	1

$$\rightarrow r_1 = 0$$

$$\rightarrow r_2 = 1$$

$$\rightarrow r_4 = 1$$

$$\rightarrow r_8 = 1$$

Locating Errors in a Hamming Code

- Look at locations 1, 2, 4, 8 and compare value received with value calculated: if equal: bit-value is 0, and if different: bit-value is 1.

Bit Positions :	1	2	4	8
Value Received :	1	0	0	1
Value Calculated :	0	1	1	1
Difference :	1	1	1	0

- The last sequence of bits gives the location of the error

$$1 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 = 7$$

- The 7th bit is in error!!! Fix this bit and you are done!!!
- **Correcting Bursts:** To detect bursts of a certain length an even higher redundancy is required! We won't cover this here!

Sender

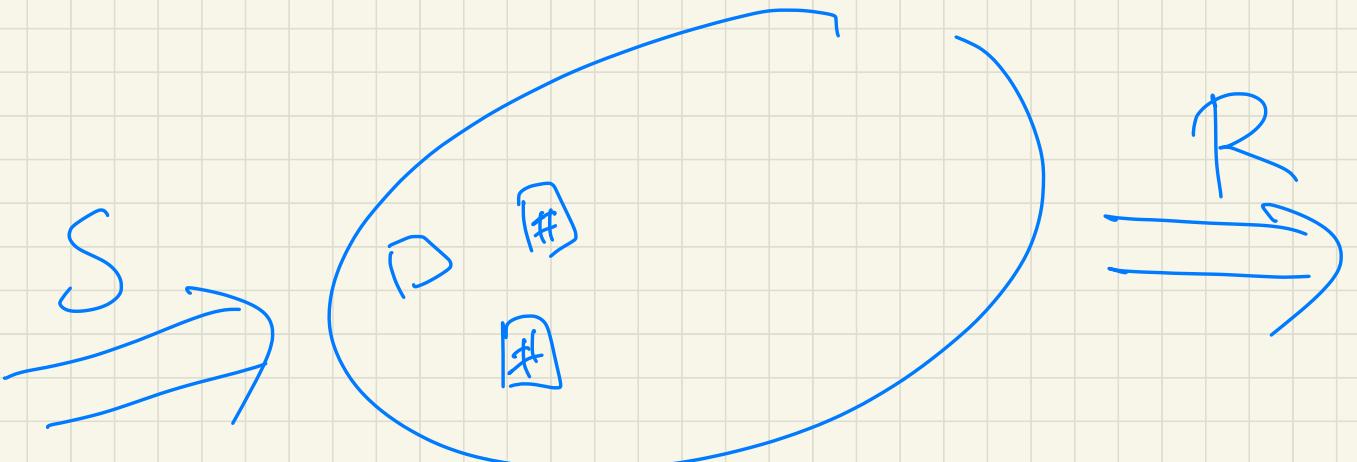
receiver

PEER-TO-PEER CONNECTIVITY

Service Models

- There are two types of service models.
 - **Connection-oriented:** A connection setup procedure precedes the transfer of information. This initializes state information between the two peer nodes. In the data transfer phase the state information provides a context used by the peer processes in the exchange of data.
 - **Connectionless:** There is no connection setup procedure. Instead, self-contained blocks of information are transmitted and delivered using address information in each packet. In its simplest case, no ACK is provided. So this is more appropriate when transfer is reliable.
- QoS parameters may also be added, e.g., reliability in terms of probability of errors, probability of loss, etc.

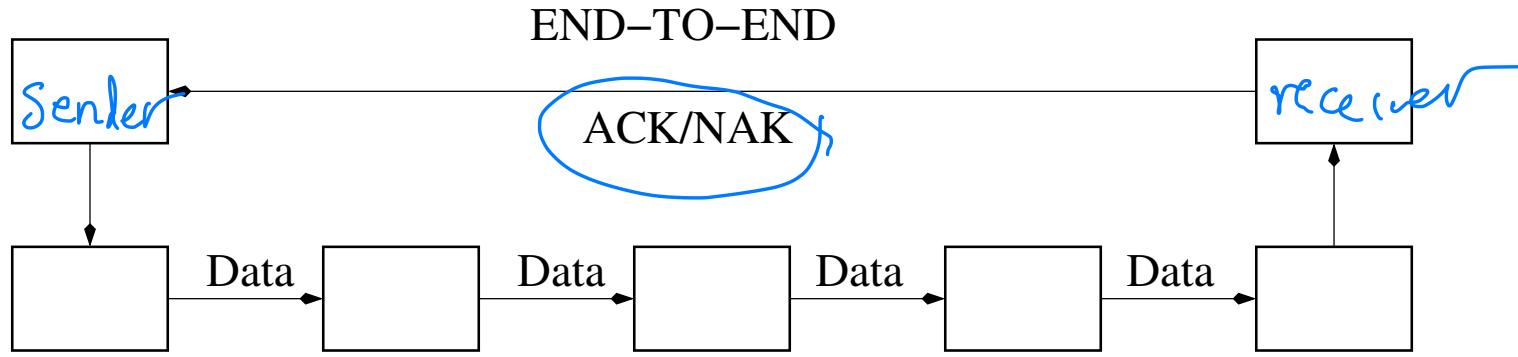




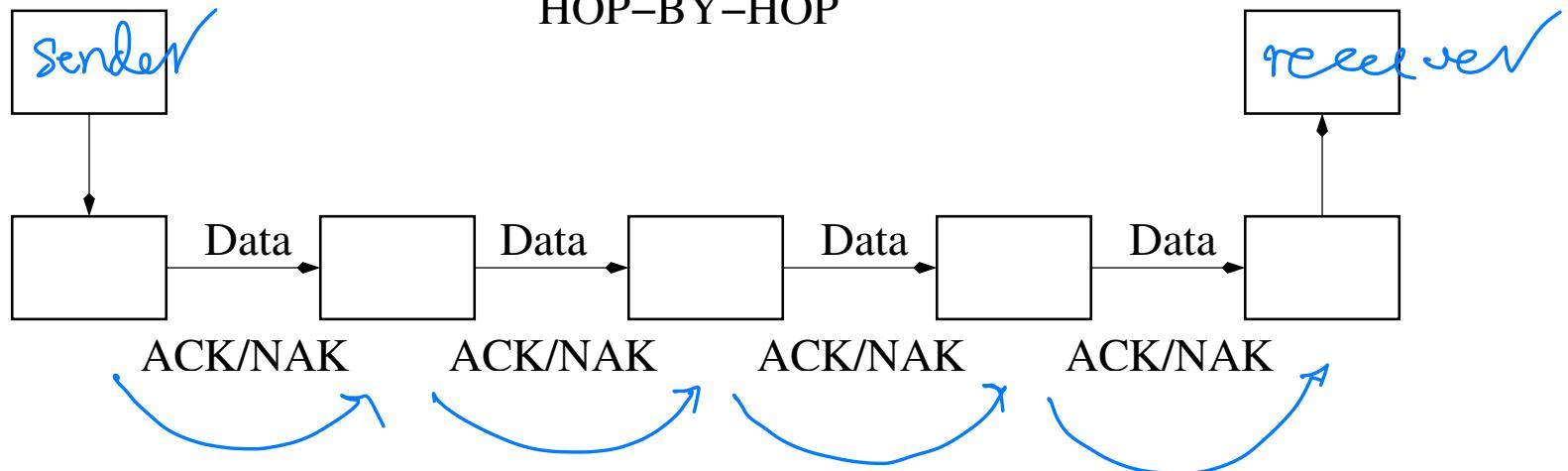
End-to-End Requirements

1. Arbitrary message size.
2. Reliability and sequencing.
3. Pacing and flow control.
4. Timing.
5. Addressing.
6. Security features (Privacy, authentication, integrity).

End-to-End and Hop-by-Hop



HOP-BY-HOP



Need for Coordination

- Three issues must be addressed:

1. **Line Discipline:**

Coordinates the link systems. Typically we have half-duplex transmission between two communicating peers.

2. **Flow Control:**

Coordinates the amount of data that can be sent and received.

3. **Error Control:**

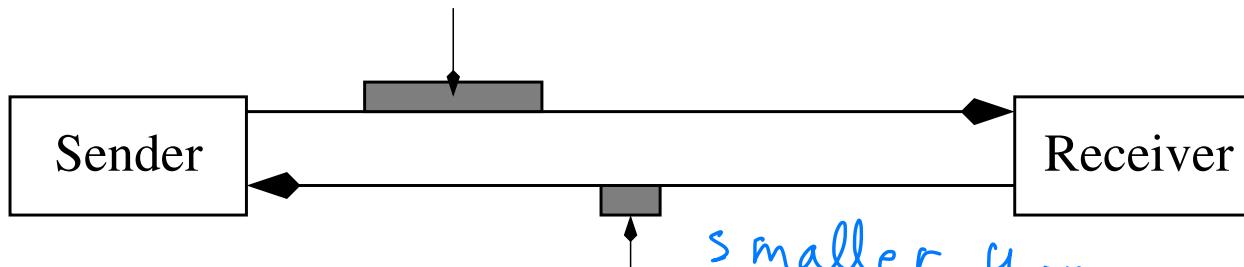
Receiver must inform sender of lost and/or damaged packets. This is usually handled by ARQs (Automatic Repeat reQuests).

- We examine all three issues in the sequel.

Peer-to-Peer Strategies

- Peer-to-Peer considerations:

Information Packets

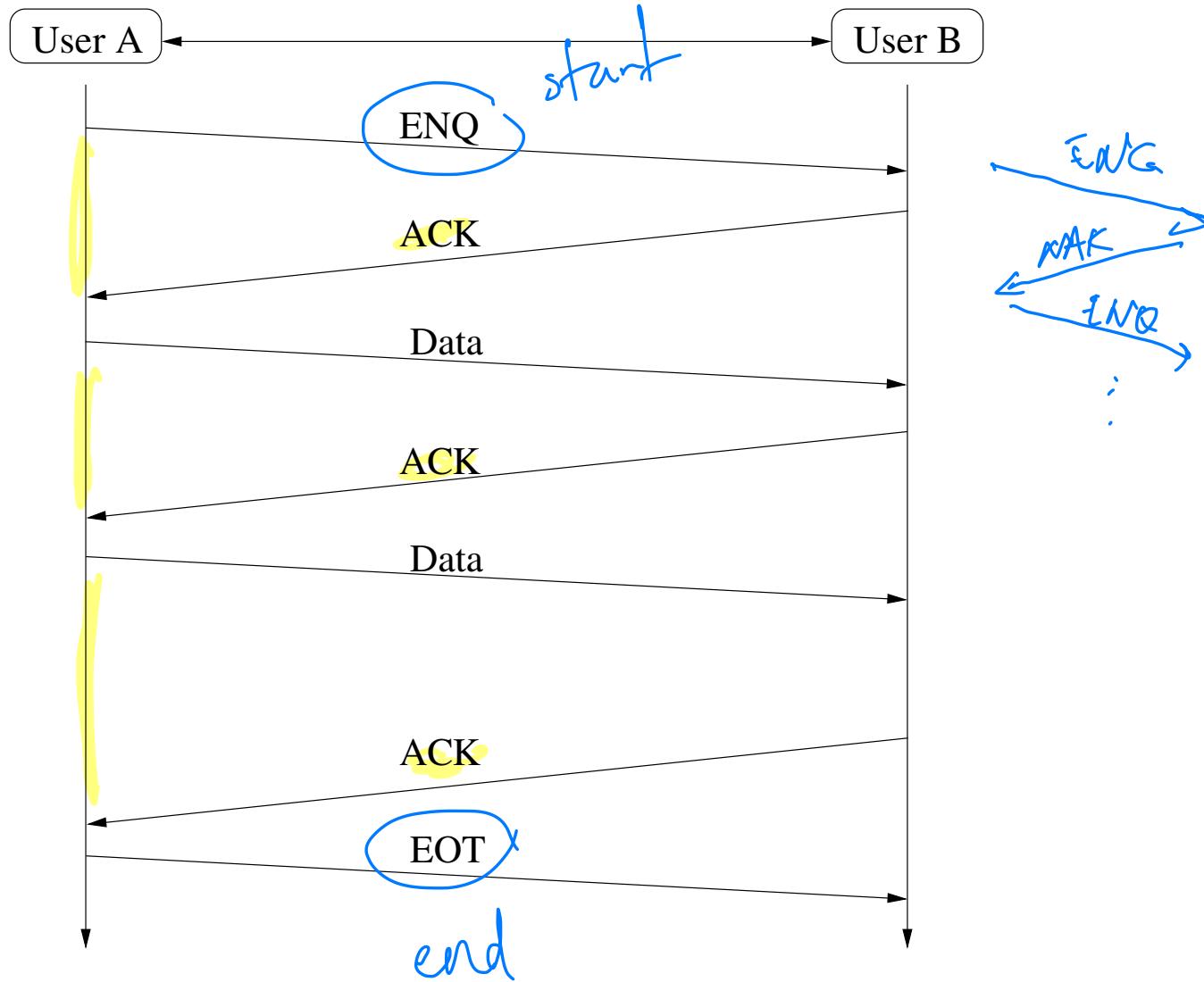


Control Packets

*smaller than
information packet*

1. Who should start?
2. Is the other peer ready?
3. How much should I send?
4. When do I report problems?

ENQ/ACK Line Discipline

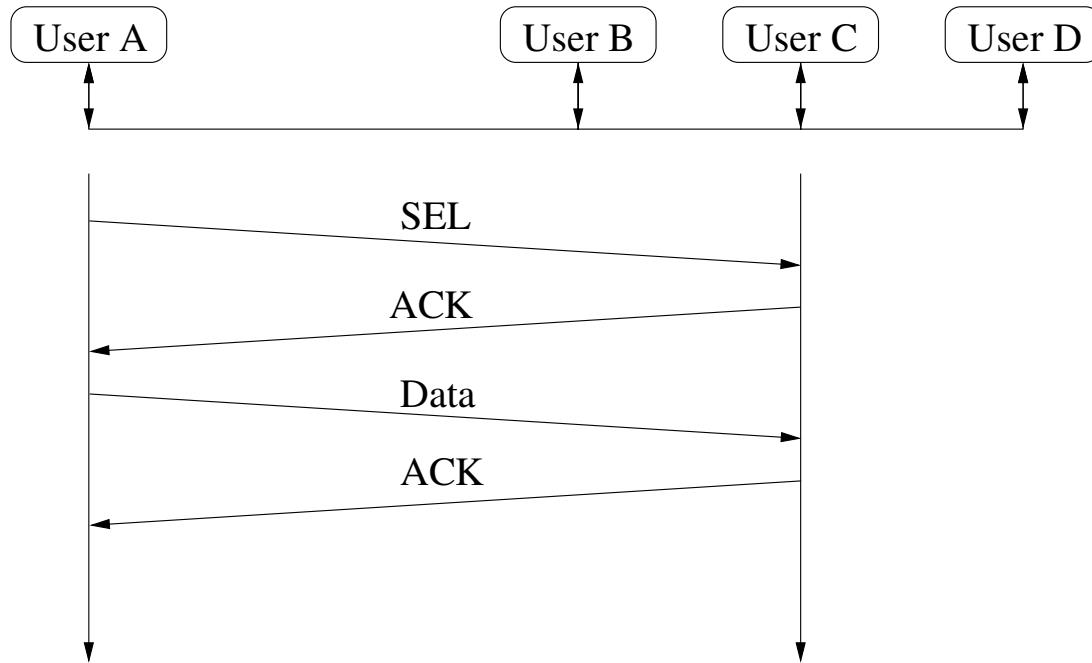


ENQ/ACK Line Discipline

1. Sender sends packet ENQ to determine if receiver is available.
2. Receiver must answer either with ACK (acknowledgement) or NAK (Negative ACK).
 - (a) If sender receives neither ACK nor NAK it must assume frame ENQ was lost and sends replacement (usually makes three attempts).
 - (b) If sender receives a NAK to ENQ each time after three times it gives up and tries later.
3. After receiving an ACK sender sends Data which are “periodically” acknowledged by the receiver.
4. Once all Data has been sent the sender sends EOT (End of Transmission).



Poll/Select Line Discipline



1. In SEL User A selects the user it wants to talk to.
2. One of Users B, C, D must respond with ACK.
3. Poll is used by User A to solicit transmissions.

LINK (FLOW) CONTROL MECHANISMS

- How is the flow controlled?
- Several flow control mechanisms are in use.
- We discuss two:
 - Stop-and-Wait
 - Sliding Window



STOP-AND-WAIT

- Basic Algorithm:

- **Source:**

- Transmits frame

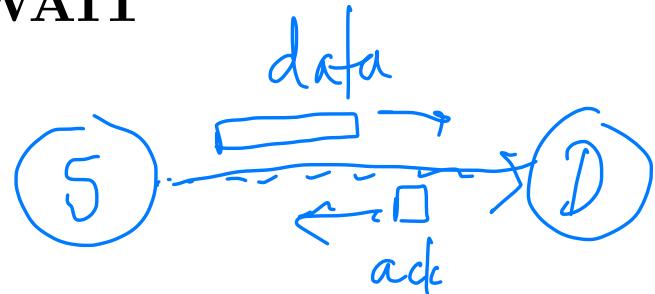
- **Destination:**

- * receives frame
 - * indicates willingness to accept another frame by sending back ACK to the frame just received.

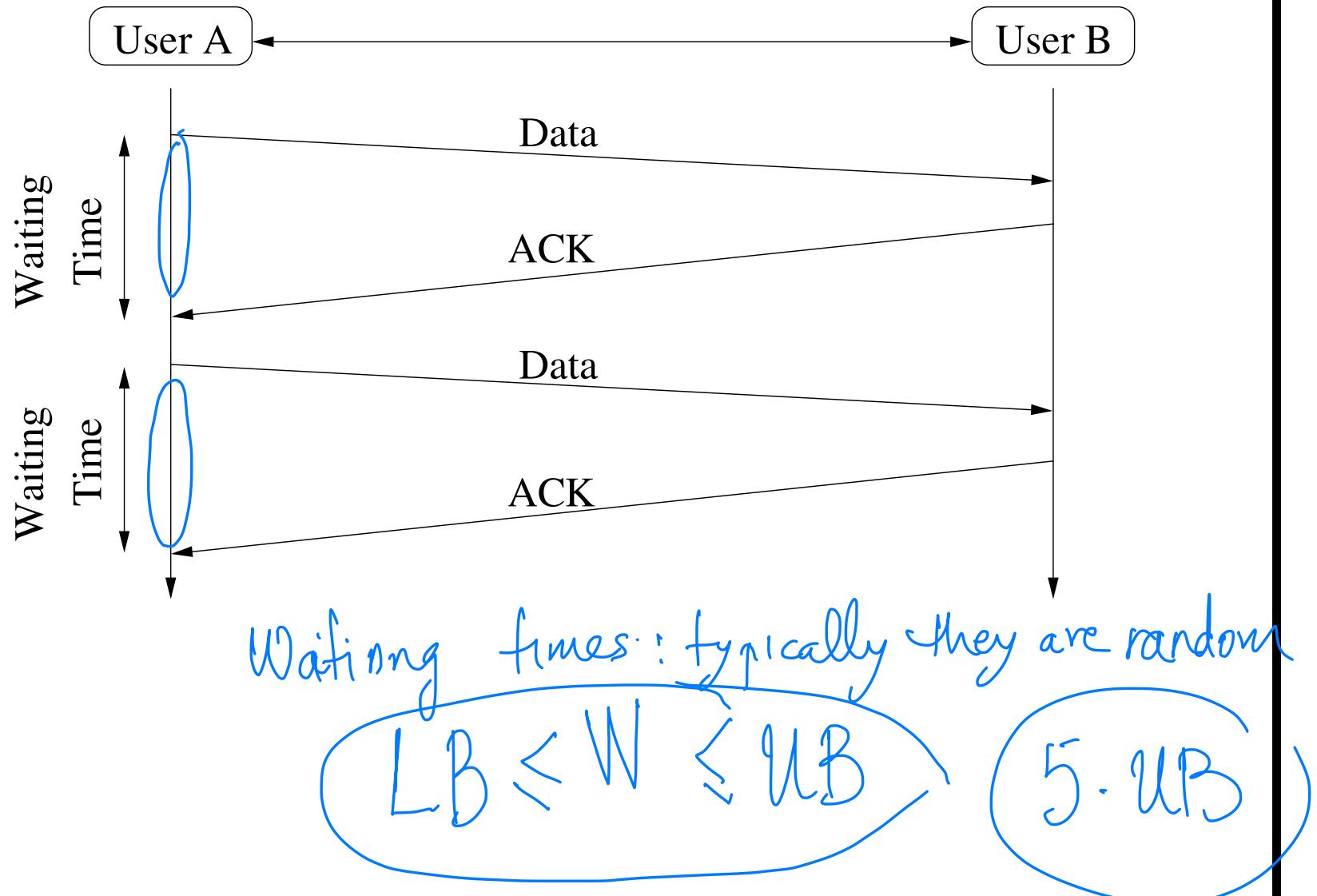
- **Source:**

- Must wait until it receives ACK before it sends new frame.

- Thus by withholding ACK the destination can stop the flow of data.

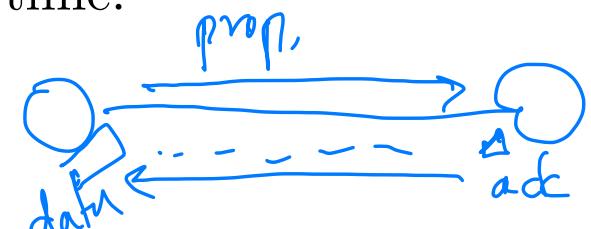


STOP-AND-WAIT



INEFFICIENCIES

- Stop-and-Wait rarely used in practice for data transfers
- Only one frame at time is in transit
- If propagation time is long relative to the transmission time then the line will be idle for most of the time.
- For example,
 - the frame rate is

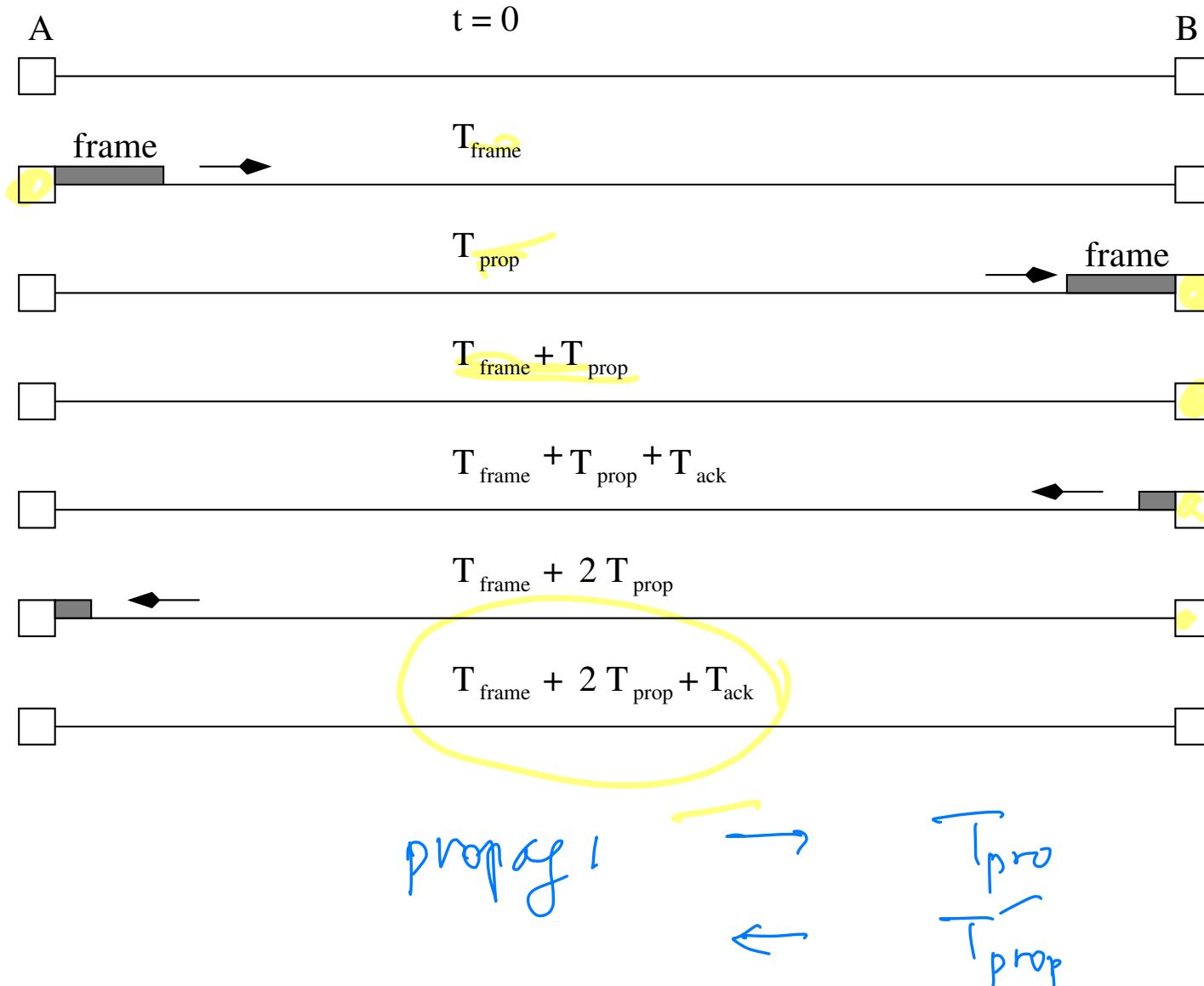


$$\frac{1}{T_{frame}}$$

- but in fact the actual rate achieved is reduced to

$$\frac{1}{T_{frame} + 2 \cdot T_{prop} + T_{ack}}$$

Utilization of Links in Stop-and-Wait



T_{prop} : is measured from
the time the last bit of
the frame "left" A to the
time the first bit of the
frame arrived at B



Efficiency of stop and wait

$$\frac{T_{frame} + 2T_{prop} + T_{ack}}{T_{frame}}$$

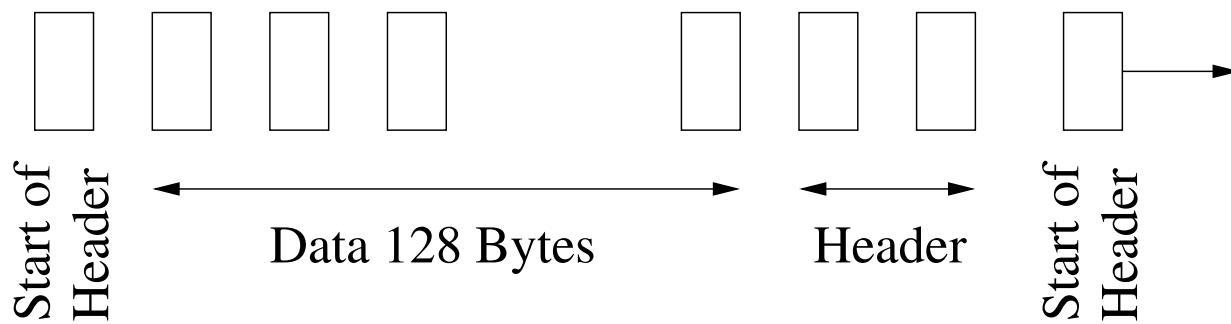
Asynchronous Protocols (Usually Slow!)

Based on Stop-and-Wait.

- XMODEM
- YMODEM
- ZMODEM
- BLAST
- KERMIT

XMODEM: Example^a

- The XMODEM frame is depicted below.
- Start of Header is 1 Byte, Header is 2 Bytes, and Data is 128 Bytes.



- The header was followed by the 128 bytes of data, and then a single-byte checksum.
- The complete packet was thus 132 bytes long, containing 128 bytes of payload data, for a total channel efficiency of about 97%

^aDeveloped in 1977

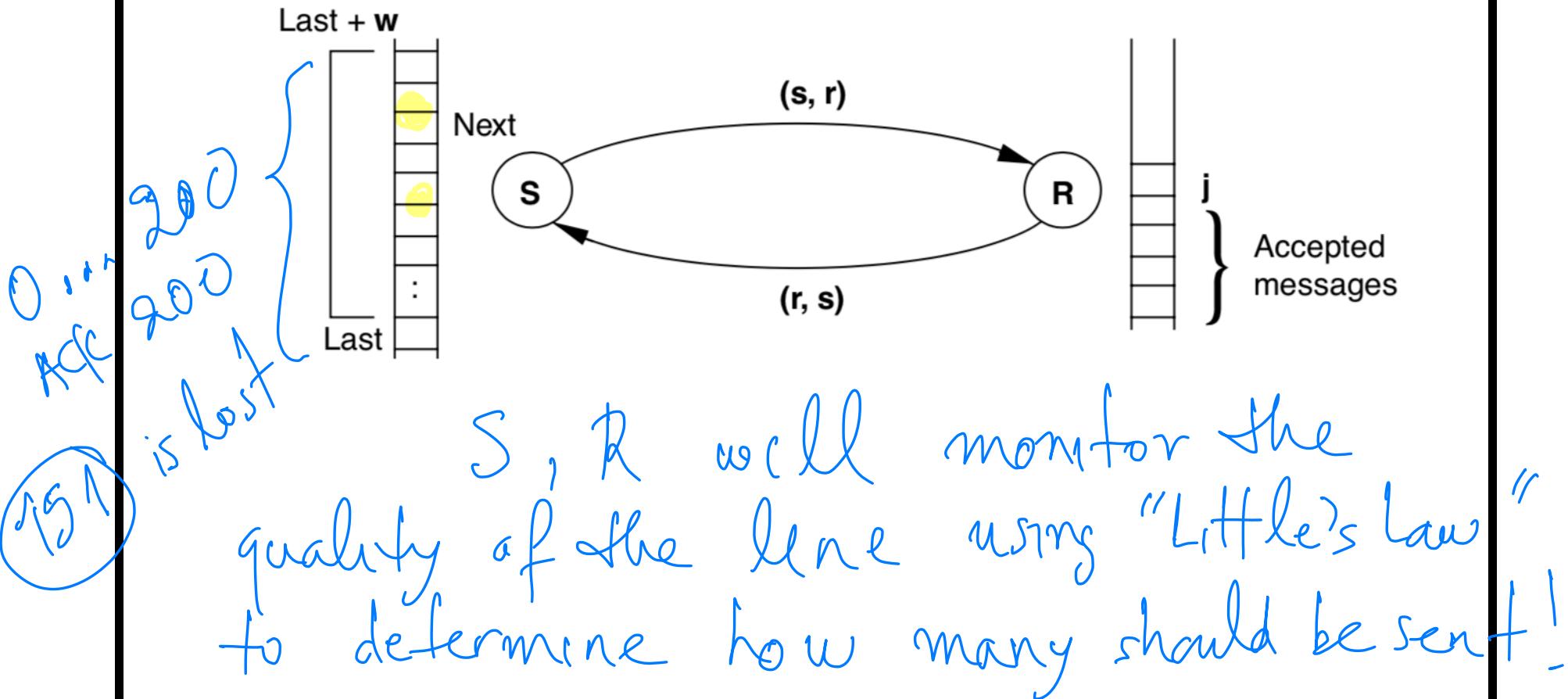
Example: Xmodem ARQ^a

- Xmodem used to be a popular modem transfer protocol.
- Information transmitted in fixed-length blocks consisting of a 3-byte header, 128 bytes of data and a 1-byte checksum.
- Header has a Start-of-header character, a 1-byte sequence number and a 2s complement of the sequence number.
- **Example: Bitsync ARQ**
 - Bitsync is IBM's Binary Synchronous Communications protocol.
 - It is character-oriented and uses the ASCII character set.

^aARQ stands for Automatic Repeat Request

Idea of Sliding Window

- Sliding window protocol is a widely used ~~transport layer~~ protocol that implements a reliable channel between a pair of processes: S (sender) and R (receiver).



Sliding Window: What Does It Do?

- It handles both omission failures and message reordering caused by an unreliable channel.
- It detects the loss or reordering of messages (and acknowledgments) using timeouts (with limited accuracy since message propagation delays are arbitrarily large but finite) and resolves it by retransmissions.
- It has a mechanism to improve the transmission rate, and restore the message order at the receiving end without overflowing its buffer space.

Sliding Window: How Does it Work?

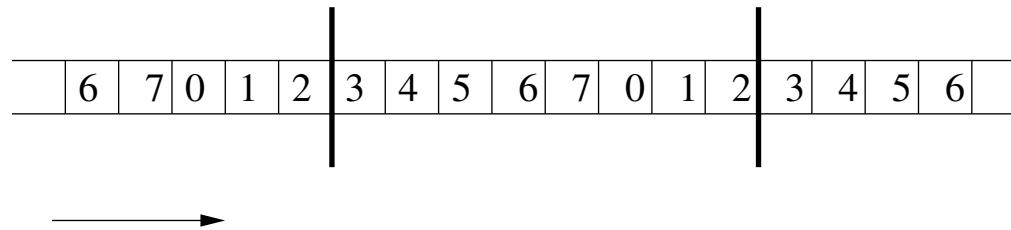
- The sender continues the send action without receiving the acknowledgments of at most w outstanding messages ($w > 0$), where w is called the window size. If no acknowledgment to the previous w messages is received within an expected period of time, then the sender resends those w messages.
- The receiver anticipates the sequence number j of the next incoming message. If the anticipated message is received, then R accepts it, sends the corresponding acknowledgment back to S , and increments j . Otherwise, R sends out an acknowledgment corresponding to the sequence number $j - 1$ of the previous message accepted by it.

SLIDING WINDOW (General Concept)

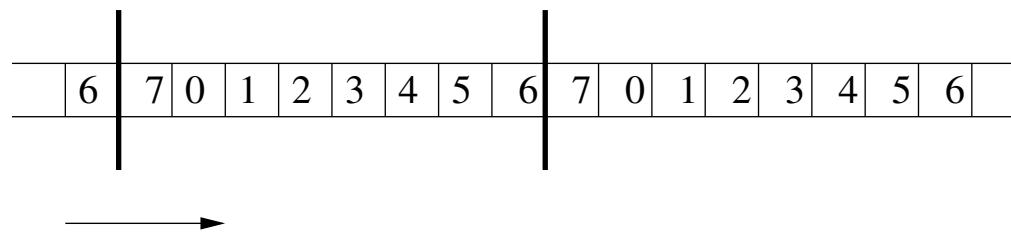
- Main problem in Stop-and-Wait is that only one frame at a time can be in transit.
- Assume we have two stations A, B .
 - B has buffer space for n frames
 - A is allowed to send n frames without waiting for ACK
 - Each frame is labeled with a sequence number
 - B acknowledges a frame by sending an ACK that includes a sequence number announcing it is ready to receive next n frames beginning with number specified
 - sequence numbers occupy a field in the frame; for a k -bit field the range is $0..2^k - 1$ and frames are numbered modulo 2^k .

Example of Sliding Window of Size 8

- Sender:



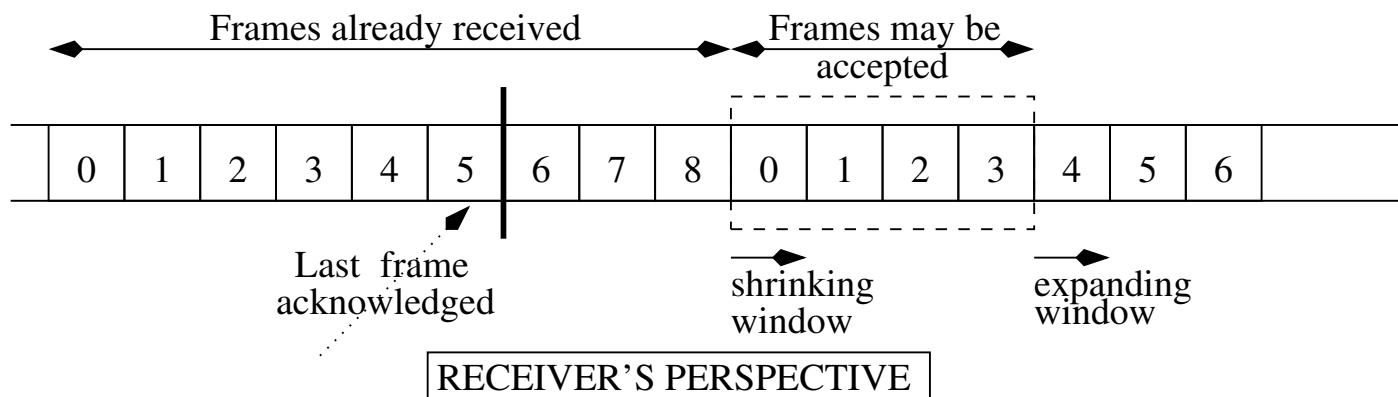
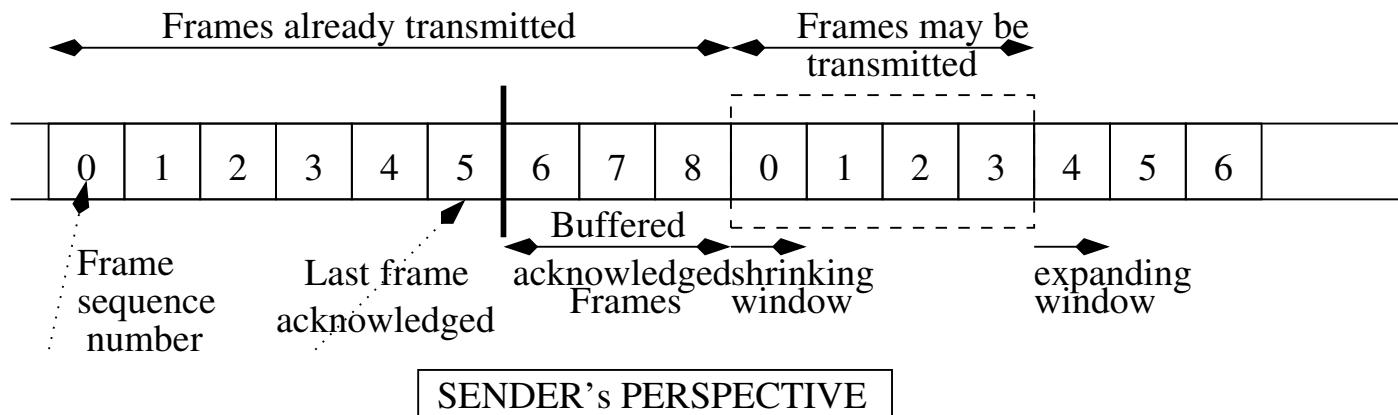
- Receiver:



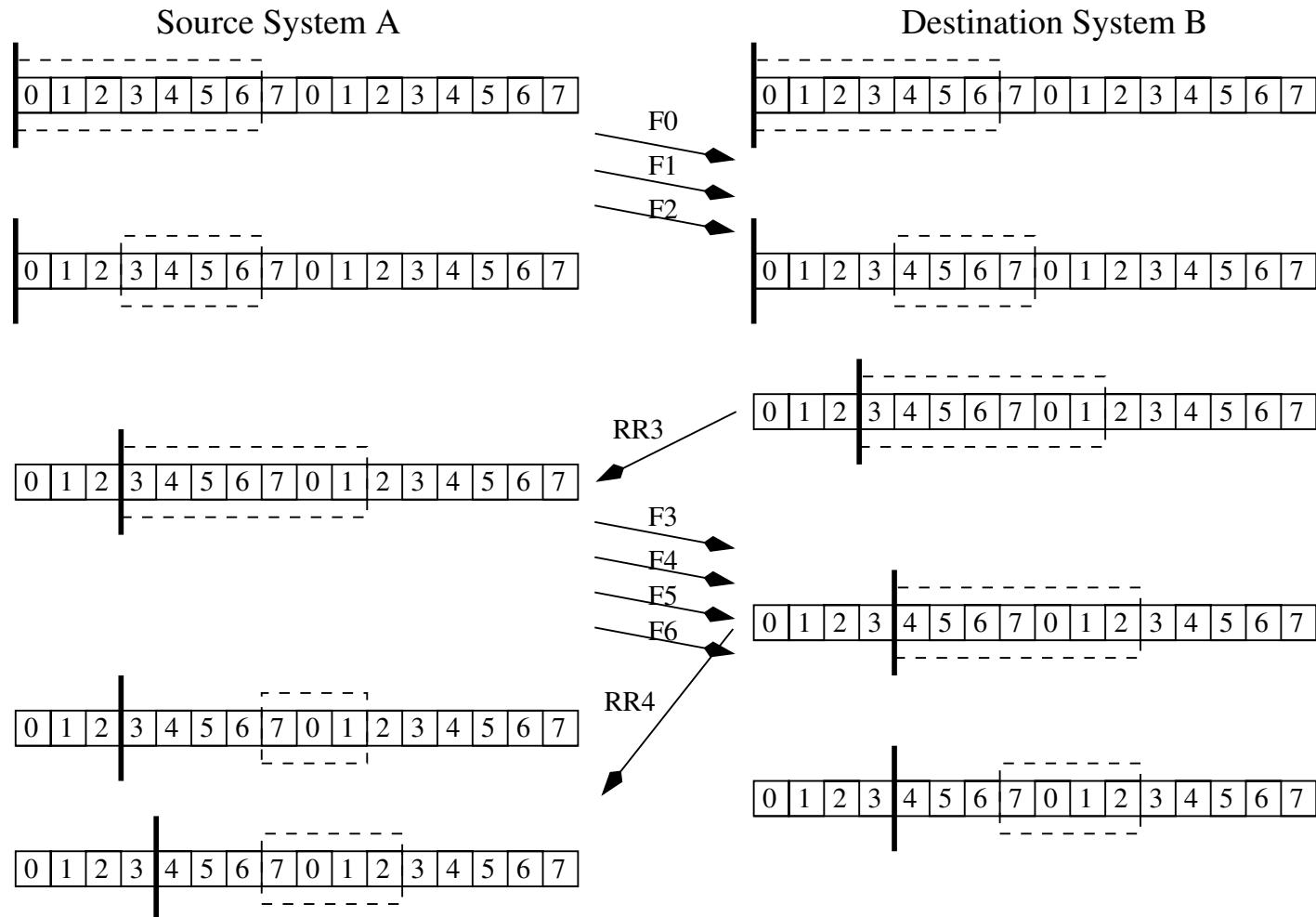
Example of Sliding Window

- B could receive frames labeled 2, 3, 4.
- It does not have to acknowledge frames received so far until frame labeled 4 has arrived.
- It then returns an ACK[5].
- A maintains a list of numbers that it is allowed to send.
- B has a separate list of numbers that it is prepared to receive.

Sliding Window Perspectives



Example of Sliding Window Protocol



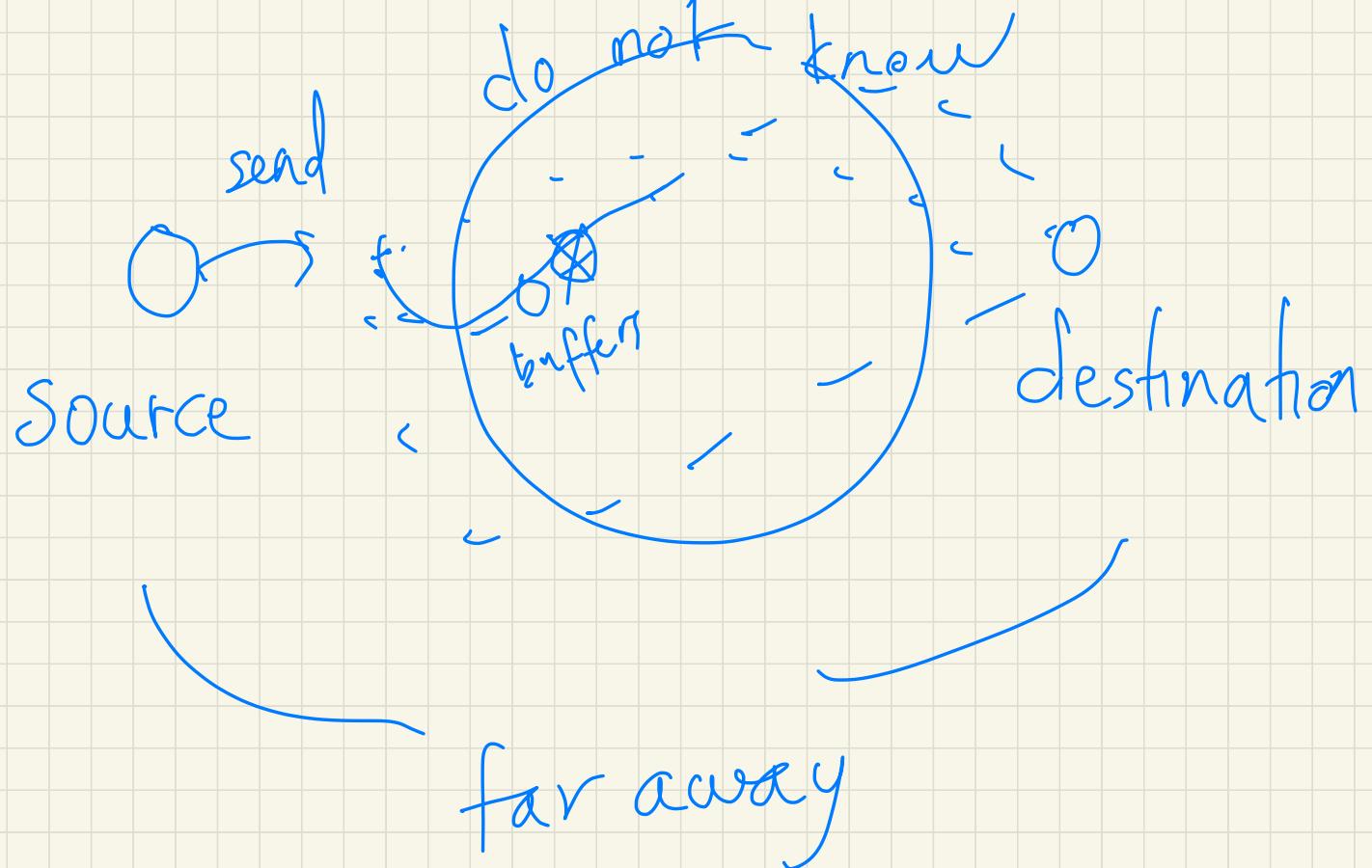
Little's Law

Little is somebody's
name

Little's is invisible in the
OSI hierarchy above DLL.
We have created the packets
and now we must deliver
them.
It provides "Control" mechanisms

Network Performance

- Development of efficient network algorithms is influenced by transmission delays of packets from source to destination.
 - Which network protocol gives the best delay-throughput characteristics under specified conditions?
 - What size buffers must be employed by a network's users in order to keep the probability of buffer overflow below a particular value? 
 - What is the maximum number of voice calls that can be accepted by a network in order to keep the voice packet transfer delay to a minimum?
 - How many users can a satellite link support and still maintain a reasonable response time?



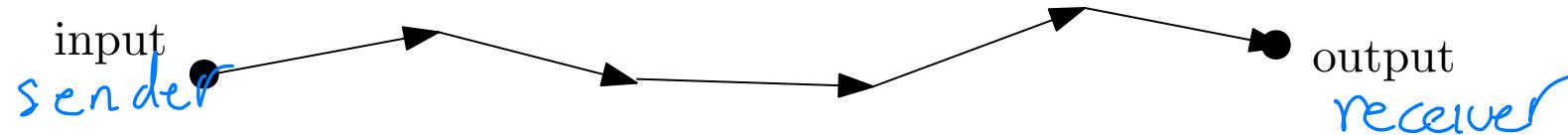
Types of Delay

- They are all measured in time units.
 - **Processing:** Delay between time packet is correctly received and the time it is correctly assigned to an outgoing link.
 - **Queuing:** Delay between time packet is assigned to a queue for transmission and the time it starts being transmitted.
 - **Transmission:** Delay between time that the first and the last bits of the packet are transmitted.
 - **Propagation:** Delay between the time that the last bit is transmitted and the bit is received.

- Is there a general principle underlying the various types of delay?

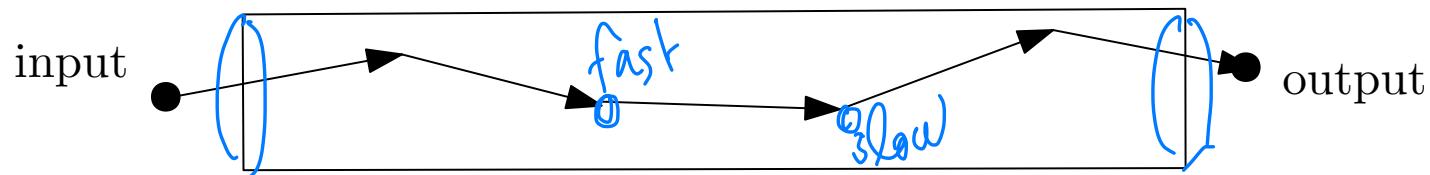
Packet Pipes

- The transmission of packets from source (input) to destination (output) ...



... resembles a pipe of packets

pipe of packets



in that you can observe only the input and the output.

- You don't know precisely what is going on inside the pipe!
- Can observations of the input and output teach us something about the performance of the system?

Example

Restaurant Paradigm

- You are a spy from Burger King trying to figure out how many people are inside MacDonald's.



Buffer
Site

- You cannot sit inside MacDonald's all day;
- You must derive the answer based only on observing traffic.

Example

- It's like having a wire with packets entering from the left and exiting from the right.



- You can count how many packets enter the wire in a given time interval: of course the count will be on the average!
- You can count how long a packet stays in the wire before exiting: of course for many customers the measurement will be on the average!
- But you cannot see inside the wire!
- This is observed in many network traffic applications.

Example

- Back to the restaurant example:
- You observe that on the average 40 customers per hour go into the restaurant.
- You observe that on the average a customer stays 15 minutes.
- Any given time there are, on the average, 10 customers inside the restaurant, because

$$40 \text{ customers per hr} \times 1/4 \text{ of an hour} = 10$$

- This sounds like a fundamental principle in networking!.

Modeling Delay

- In typical queueing systems customers (i.e. packets) arrive at random times to obtain service.
- If L = packet length in bits, C = link transmission capacity in bits/sec Then:

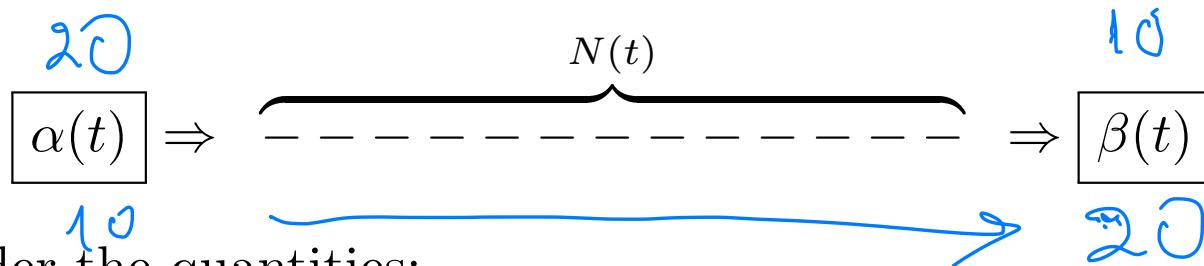
$$\text{service time} = \frac{L}{C}$$

- We ignore the distinction between frame and packet.
- We will be interested in estimating:
 - Average # of customers in the system (either waiting in queue or undergoing service);
 - Average delay per customer.
- These will be estimated in terms of customer arrival and service rates.

Little's Theorem

- Little's theorem concerns time averages in the limit.
- Suppose we observe a sample history of a system from the starting time $t = 0$,

MacDonalds



- Consider the quantities:

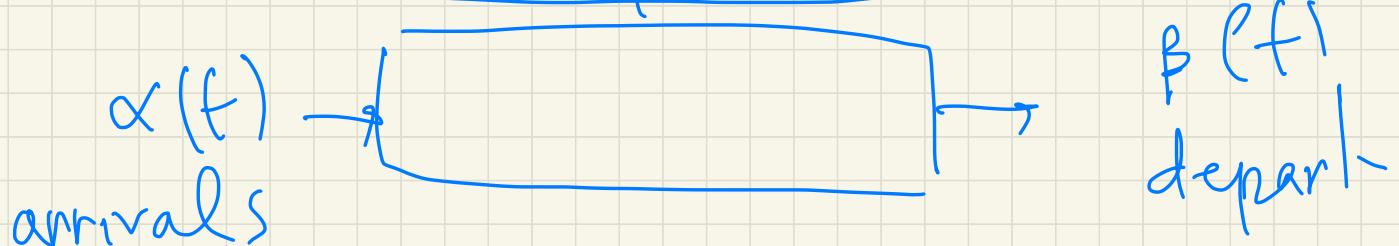
$\left\{ \begin{array}{l} N(t) = \# \text{ of customers in the system at time } t; \\ \alpha(t) = \# \text{ of customers who arrived in the interval } [0, t]; \\ \beta(t) = \# \text{ of customers who departed in the interval } [0, t]; \\ T(i) = \text{time spent in the system by } i\text{-th customer.} \end{array} \right.$

$$\frac{\alpha(t)}{t}$$

$$\frac{\beta(t)}{t}$$

- How are these quantities related?

On the average there is
equilibrium



$$\begin{array}{r} 11 \\ 9 \\ 20 \\ \hline 46 \\ 4 \\ \hline 6 \\ \hline 46 \end{array}$$

$$\begin{array}{r} 12 \\ 10 \\ 15 \\ \hline 47 \\ 4 \\ \hline 10 \\ \hline 47 \end{array}$$

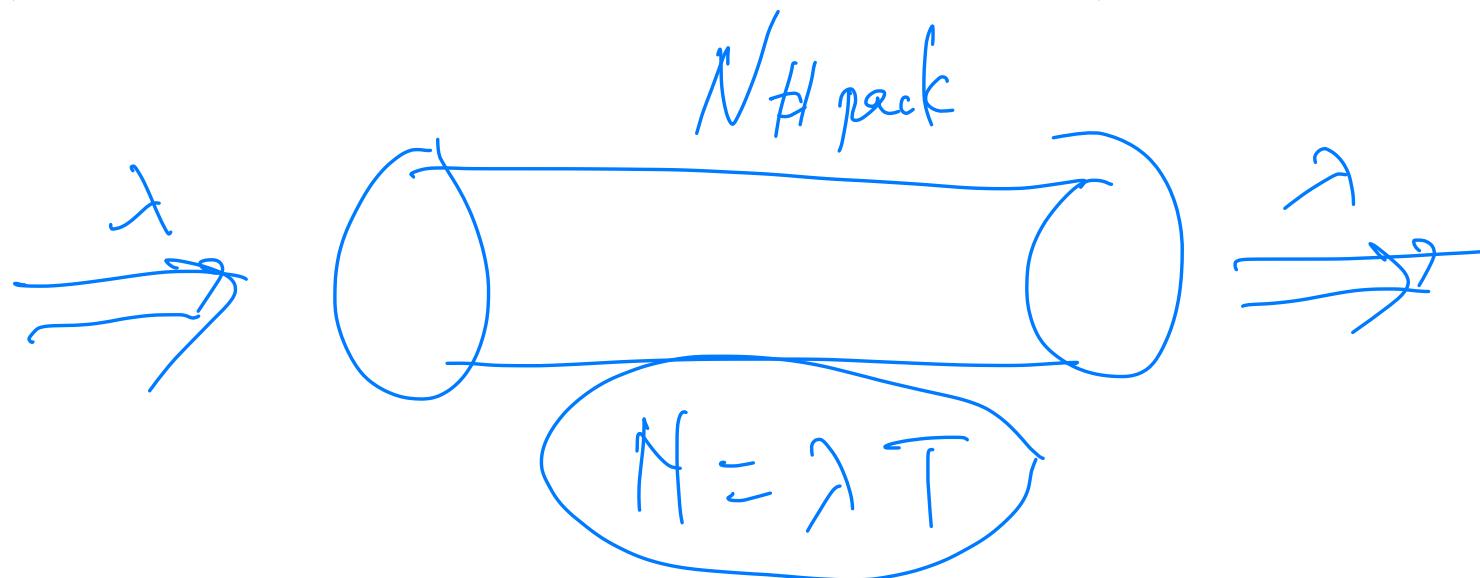
Little's Theorem

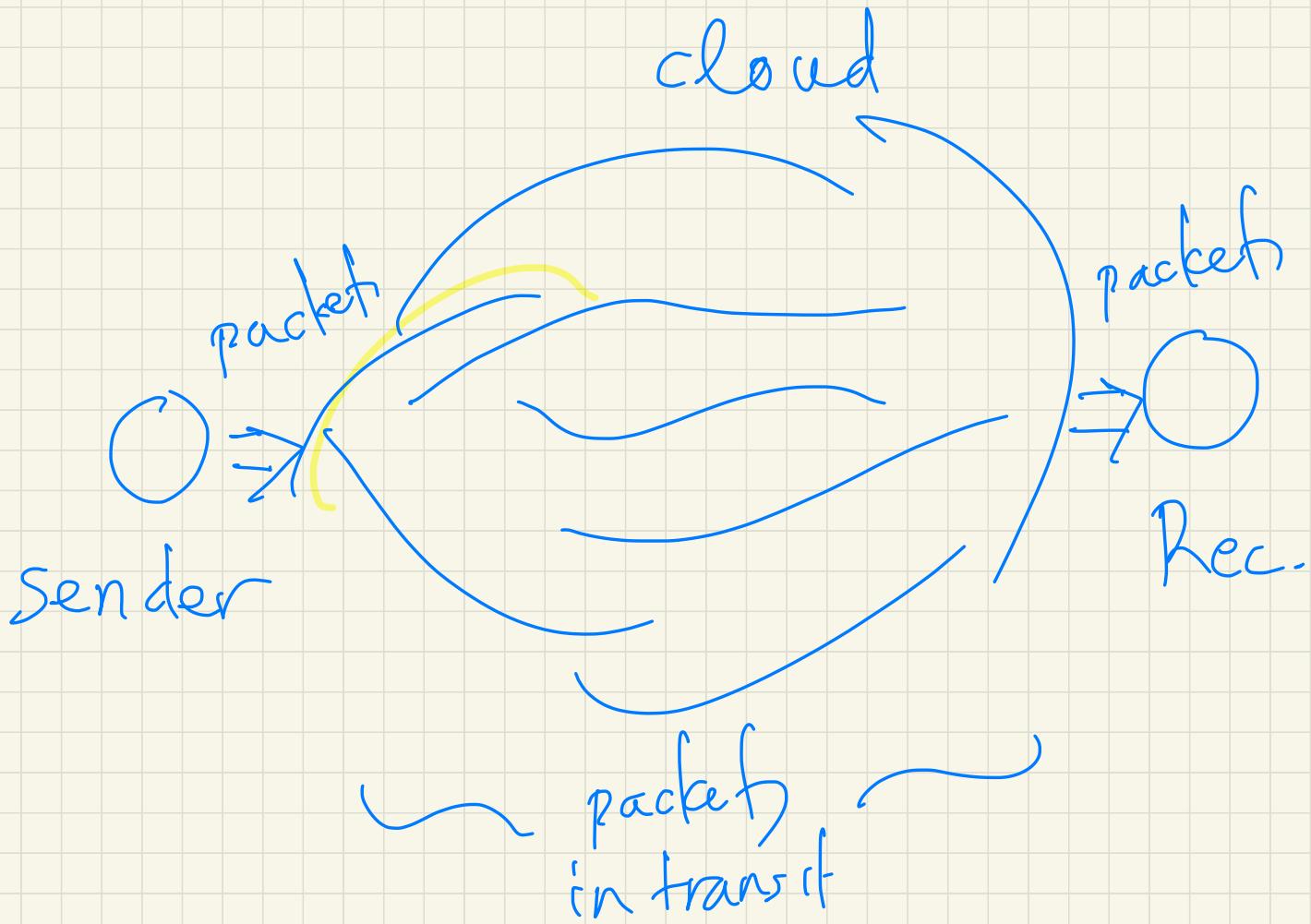
- **Theorem 1 (Little's theorem)** Assuming a system with steady state behavior, i.e., the rate of arrival and departure are the same (in the limit), we have that

$$N = \lambda T$$

*they are
all averages*

where N, T, λ are the averages of the quantities defined before (and will be defined in the course of the proof).





Proof of Little's Theorem

- We will be interested in finding a relation between these parameters. We define the time average

- arrival rate over interval $[0, t]$: $\lambda_t = \alpha(t)/t$
- of the customer delay up to time t :

$$T_t = \frac{1}{\alpha(t)} \sum_{i=1}^{\alpha(t)} T(i)$$

$$\frac{\alpha(t)}{t}$$

$\alpha(t)$
 $\underline{\alpha(t)}$
 $\underline{[0, t]}$

- of the number of customers up to time t :

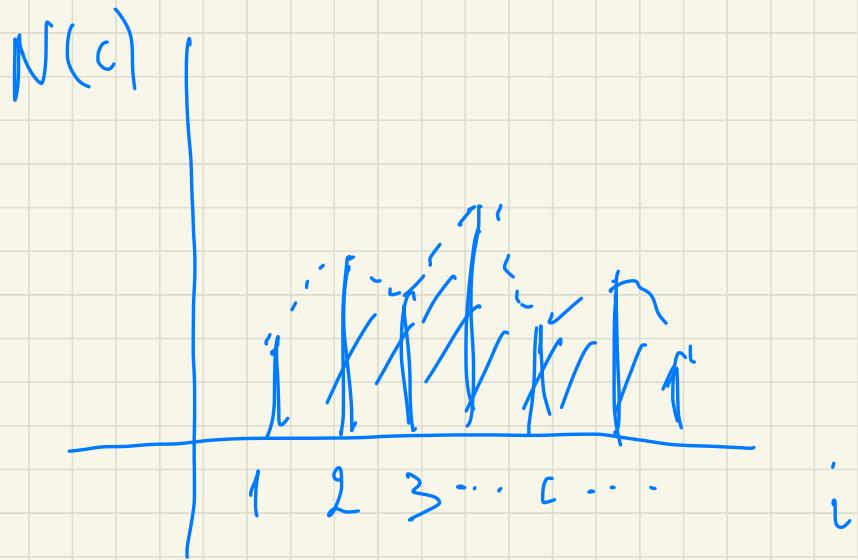
$$N_t = \frac{1}{t} \sum_{i=1}^t N(i) \approx \frac{1}{t} \int_0^t N(i) di$$

$T(c) = \text{true}$
 $\text{customer } c$
 stays in
 system

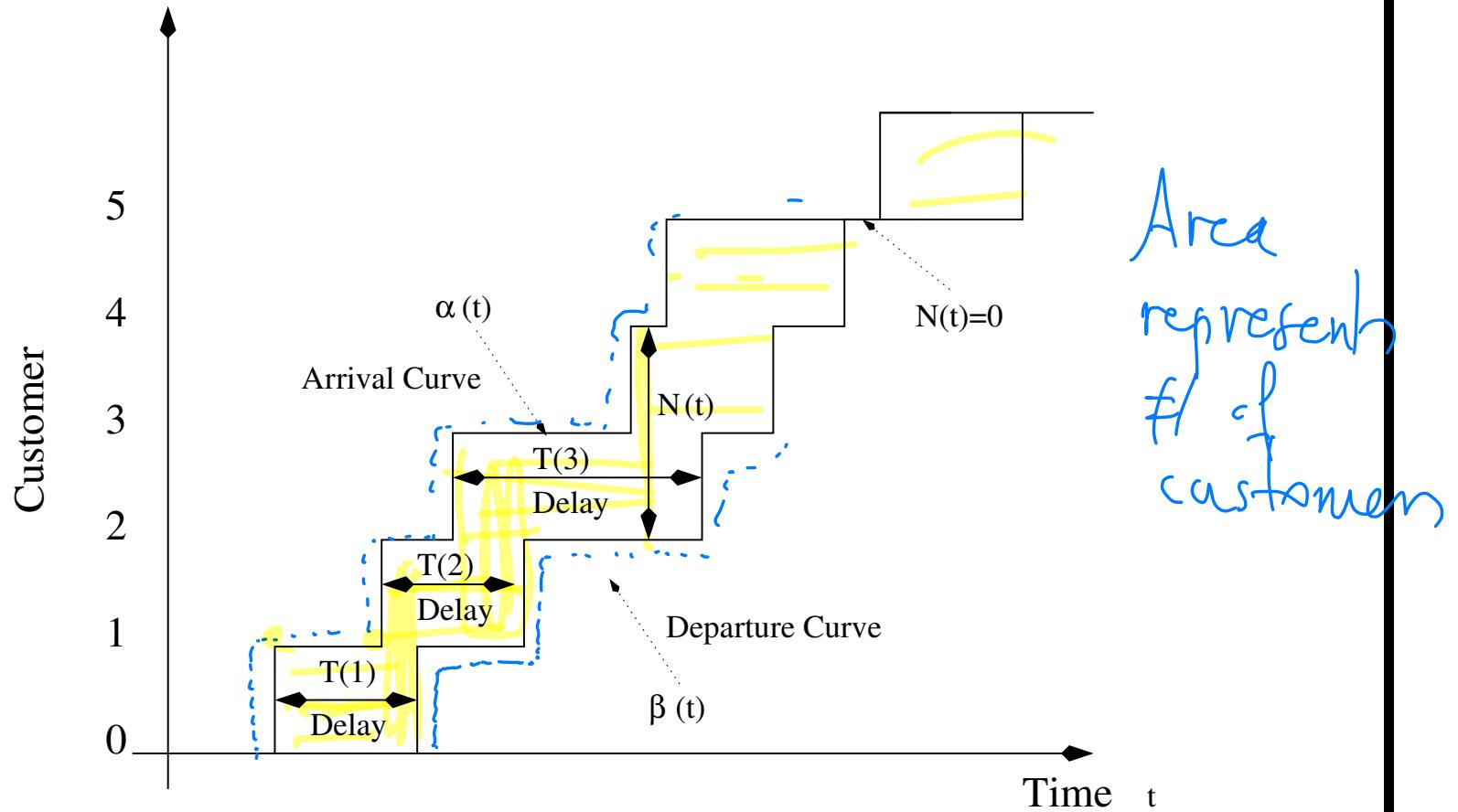
$$N(c)$$

- In many systems of interest, these quantities tend to a steady state:

$$\lambda := \lim_{t \rightarrow \infty} \lambda_t, T := \lim_{t \rightarrow \infty} T_t, N := \lim_{t \rightarrow \infty} N_t$$



Proof of Little's Theorem



Illustrated are the functions $\beta(t)$, $\alpha(t)$ and $\beta(t) \leq \alpha(t)$, $T(i)$, is the delay of customer i .

Proof of Little's Theorem

- $\alpha(t), \beta(t)$ is the number of arrivals and departures up to time t .
- Their difference $\alpha(t) - \beta(t)$ is the number $N(t)$ in the system at time t .
If $\alpha(t) > \beta(t)$ then what can you conclude?
- The area between the arrival and departure curves $\alpha(t), \beta(t)$ is equal to

$$\int_0^t N(\tau) d\tau$$

- If $N(t) = 0$ then the area between the arrival and departure curves is also equal to

$$\sum_{i=1}^{\alpha(t)} T(i).$$

Proof of Little's Theorem

- From the picture $\sum_{i=1}^{\beta(t)} T(i) \leq \int_0^t N(\tau) d\tau \leq \sum_{i=1}^{\alpha(t)} T(i)$.
Therefore

$$\begin{aligned}
 \lambda_t T_t &= \frac{\cancel{\alpha(t)}}{t} \frac{1}{\cancel{\alpha(t)}} \sum_{i=1}^{\alpha(t)} T(i) \quad \text{Diagram: } \text{A yellow hand with fingers pointing right, each finger has a blue arrow pointing towards the center.} \\
 &= \frac{1}{t} \sum_{i=1}^{\alpha(t)} T(i) \\
 &\geq \circled{>} \quad \frac{1}{t} \int_0^t N(\tau) d\tau \quad \text{Diagram: } \text{A yellow circle containing a blue arrow pointing right, followed by a blue arrow pointing down, then a blue arrow pointing right again.} \\
 &\geq \frac{1}{t} \sum_{i=1}^{\beta(t)} T(i) \\
 &= \frac{\cancel{\beta(t)}}{t} \frac{1}{\cancel{\beta(t)}} \sum_{i=1}^{\beta(t)} T(i) \quad \text{Diagram: } \text{A yellow hand with fingers pointing left, each finger has a blue arrow pointing away from the center.}
 \end{aligned}$$

Proof of Little's Theorem

- Hence:

$$\text{dep. rate} \quad \leftarrow \frac{\beta(t)}{t} \cdot \frac{\sum_{i=1}^{\beta(t)} T(i)}{\beta(t)} \leq N_t \leq \frac{\alpha(t)}{t} \cdot \frac{\sum_{i=1}^{\alpha(t)} T(i)}{\alpha(t)}$$

arrival rate

- Taking the limit we have that

$$\lambda T \leq N \leq \lambda T$$

- Which proves, $N = \lambda T$, i.e., Little's Theorem.



Remarks

- Note in the proof we used the fact that

$$\lambda = \lim_{t \rightarrow \infty} \frac{\beta(t)}{t} = \lim_{t \rightarrow \infty} \frac{\alpha(t)}{t}$$

$$T = \lim_{t \rightarrow \infty} \frac{\sum_{i=1}^{\beta(t)} T(i)}{\beta(t)} = \frac{\sum_{i=1}^{\alpha(t)} T(i)}{\alpha(t)}$$

$$N = \lim_{t \rightarrow \infty} N_t = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{i=1}^t N(i)$$

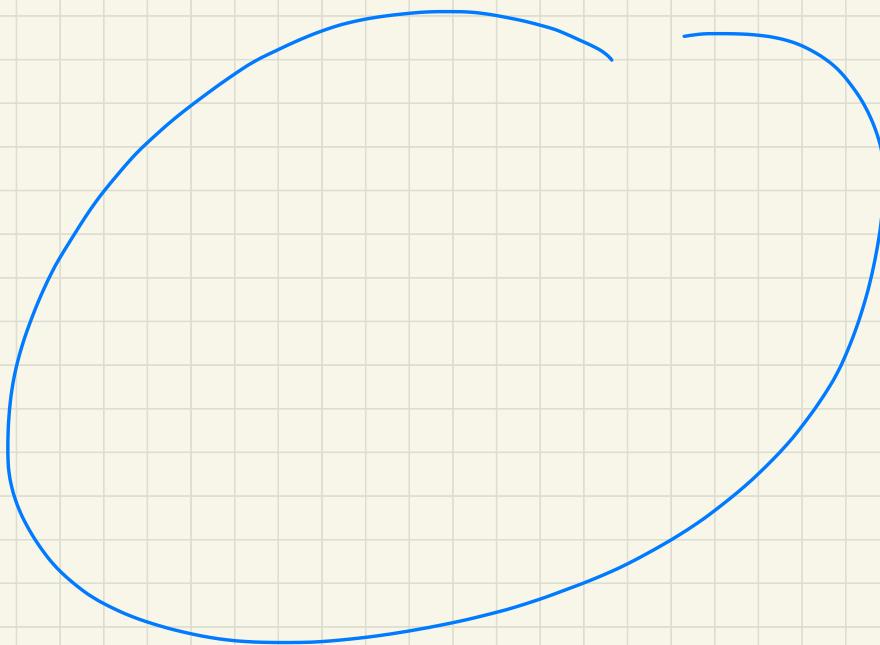
System
in steady
state
satisfies
equilibrium
conditions

- In a way, these are stability conditions!
- The significance of Little's result is that it holds for any system that reaches a steady state.
- System need not consist of a single queue provided that the terms N, λ, T are properly interpreted.

134

~~134~~
not received

Sender
→
→



Receiver
→
→

TCP / IP

Example 1

- Suppose we have a closed full system of K servers and N customers, $N \geq K$ (closed means departing customers are always replaced).
- Say average customer service time is \bar{X} ; we want to find the average customer time T in the system. Apply Little's Theorem on the whole system: $N = \lambda T$.
- Apply Little's Theorem on the service portion: $K = \lambda \bar{X}$ since all K servers are always busy
- It follows that:

$$\frac{N}{T} = \frac{K}{\bar{X}}$$

- Hence:

$$T = N \frac{\bar{X}}{K}$$

Example 2 (Complex)

- Consider now the system under the assumption that customers arrive at a rate λ and are lost (or blocked) if they find the system full.
- In this case the number of busy servers may be less than K . Let \bar{K} be the average number of busy servers, β the proportion of customers that are blocked from entering the system. From Little's theorem we derive that

$$\bar{K} = (1 - \beta)\lambda\bar{X},$$

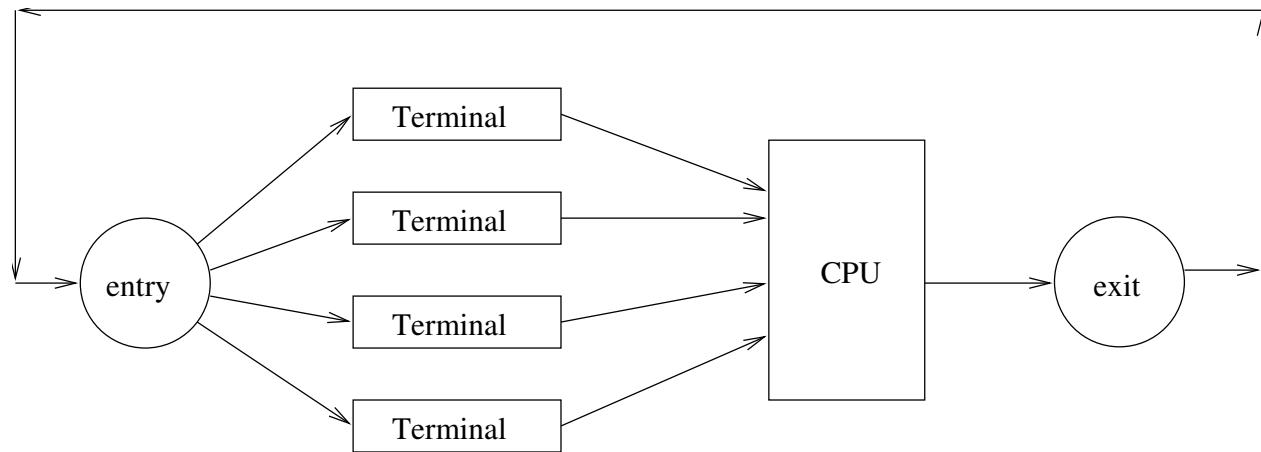
$$\beta = 1 - \frac{\bar{K}}{\lambda\bar{X}}.$$

Since $\bar{K} \leq K$ we obtain the lower bound

$$\beta \geq 1 - \frac{K}{\lambda\bar{X}}.$$

Example 3 (1/3)

- Suppose a system consisting of N terminals connected to a single CPU. Users login through a terminal.



- After reflection R , a user submits a job requiring average processing time P .
- Applying Little's Theorem between entry and exit portion of the system, we have: $N = \lambda T$, where T is average time a user spends in the system, and λ the attainable system throughput.

Example 3 (2/3)

- However, $T = R + D$, where D is the average delay between the time a job is submitted and the time its execution is completed. Clearly D may vary.

$$\begin{array}{ccc}
 P & \leq & NP \\
 \uparrow & & \uparrow \\
 \text{case of no other} & & \text{waiting for other} \\
 \text{job submitted} & & \text{job to be com-} \\
 & & \text{pleted}
 \end{array}$$

$$R + P \leq T \leq R + NP$$

- Hence: $\frac{N}{R+NP} \leq \lambda = \frac{N}{T} \leq \frac{N}{R+P}$
- However, λ is also bounded above by the processing capacity of the computer. Since the CPU can not process more than one terminal per P time units.

Example 3 (3/3)

- We have: $\lambda \leq \frac{1}{P}$

- Hence:

$$\frac{N}{R+NP} \leq \lambda \leq \min \left\{ \frac{1}{P}, \frac{N}{R+P} \right\}$$

\downarrow

$$(N \rightarrow \infty)$$

\downarrow

$$\frac{1}{P} \leq \lambda \leq \frac{1}{P}$$

which means that in the limit $\lambda = 1/P$.

- By using:

$$T = \frac{N}{\lambda}$$

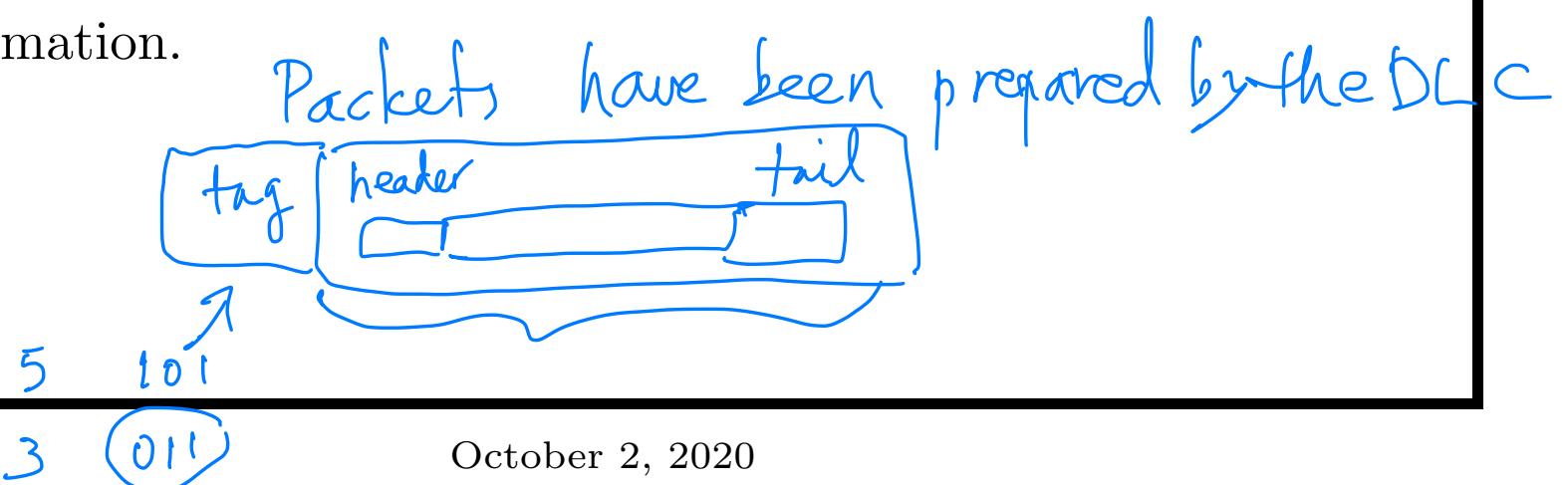
- We obtain:

$$\max\{R + P, NP\} \leq T \leq R + NP$$

AUTOMATIC REPEAT REQUEST (ARQ)

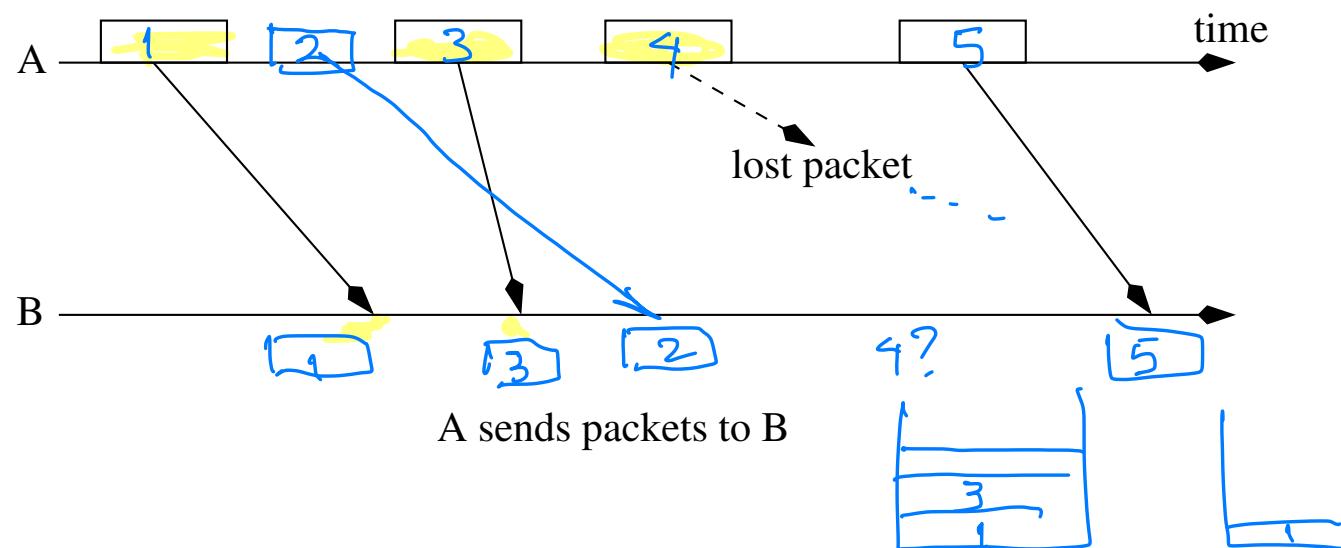
AUTOMATIC REPEAT REQUESTs

- This technique is used to ensure the data is delivered accurately. *automatically*
- Assuming framing works, a strategy is required to handle frames with errors detected by the CRC (or whatever “error” detection technique is being used).
 - Such strategies are called **Automatic Repeat reQuest** (ARQ) strategies.
- Their purpose is to detect frames with errors at the receiving DLC and then to request the transmitting DLC to repeat the information.



CHARACTERISTICS OF STRATEGIES

- A ARQ strategy is characterized by its
correctness: Is each packet released once and only once without errors from the DLC?
efficiency: How much of the bit-transmitting capability is wasted by unnecessary waiting and by sending unnecessary retransmissions?
- Consider a flow from A to B .



ASSUMPTIONS IN ARQ STRATEGIES

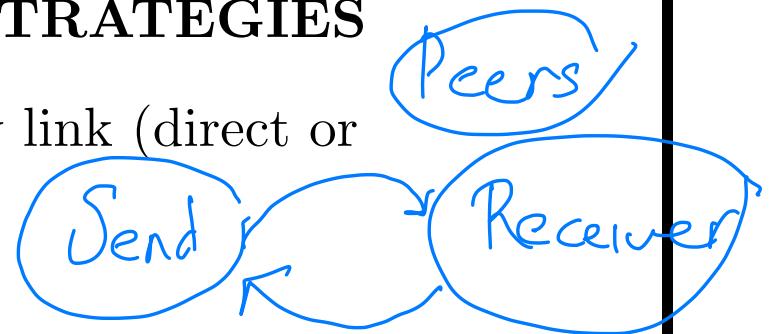
1. Framing provides the begin and end of frame information for the receiving DLC.
2. A CRC (or other method) may be used for detecting errors.
3. All frames containing transmission errors are detected.
 - as a matter of fact actual probability for CRC error detection is $1 - 2^{-L}$, where L is the length of the CRC.^a
4. The bit pipe delivers frames in order (e.g., FIFO).
5. Each transmitted frame is delayed for an arbitrary and variable time before arriving at the receiving DLC.
6. The bit pipe may lose some frames.

$$d_1, l_1, \dots, \underbrace{N-1}_{N=?}, d_2, l_2, \dots, \underbrace{N-1}_{N=?}, q_1, \dots$$

^aWe will not discuss this here in more detail.

ASSUMPTIONS IN ARQ STRATEGIES

- Assume two end systems connected by link (direct or otherwise).
 1. Source wishes to send messages
 2. Message broken into blocks which are sent individually.
 3. The reasons for this might be
 - limited buffer size
 - the longer the transmission the bigger the error probability
 - in LANs it is best not to have a single node occupy the medium for an extended period of time
- Question: What ARQ strategies can we use?
- Main Issue: the pipes may deliver in order but the queues may not be not be FIFO.



$$\frac{1}{100} \xrightarrow{\text{---}} \text{Little's law}$$

ARQ STRATEGIES

- Several strategies are in use.
- Most important ones are
 - Stop-and-Wait
 - Sliding Window (this comes in two basic forms)
 1. Go-Back- N
 2. Selective Reject

Numerous Sliding Window

- Wireless
- Satellite
- Fiber Optic
- Wireline

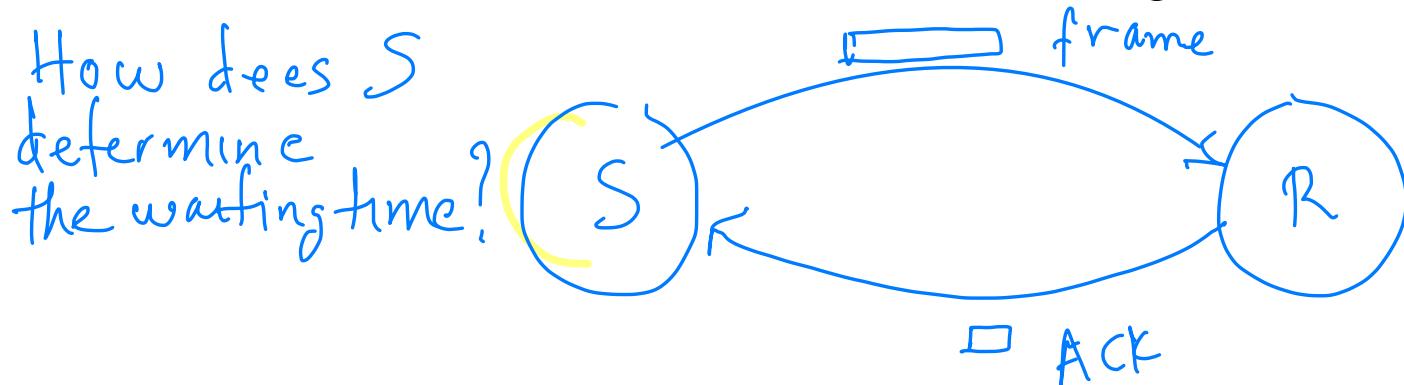
ERRORS IN STOP-AND-WAIT

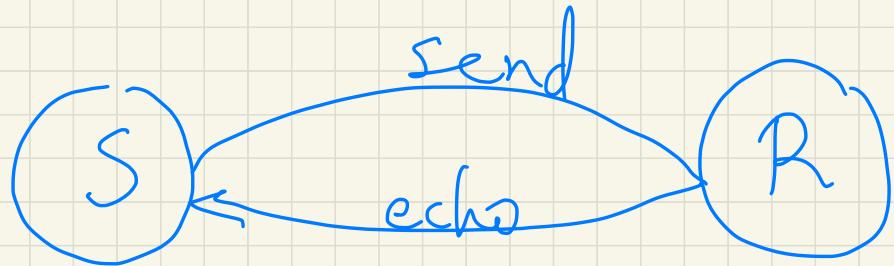
- There are two types of errors: Frame- and ACK-errors.

1. FRAME ERRORS

Frame that arrives in destination is damaged, i.e. one or more bits have been altered.

- if error is detected (e.g. via a frame check sequence based on CRC codes) the receiver discards frame.
- after frame is transmitted source waits for ACK within prespecified time frame (using a timer). If no ACK is received then the same frame is sent again.





- Can compute the RTT (Round Trip Time)
- ping
- The RTT depends on network circumstances

ERRORS IN STOP-AND-WAIT

2. ACK ERRORS

Frame is received correctly but ACK is damaged.

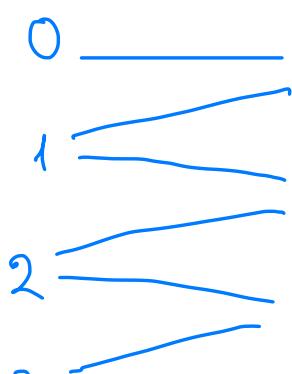
Thus sender resends message and receiver accepts same message twice. *You need to "add" a header*

For this reason we use a labeling mechanism

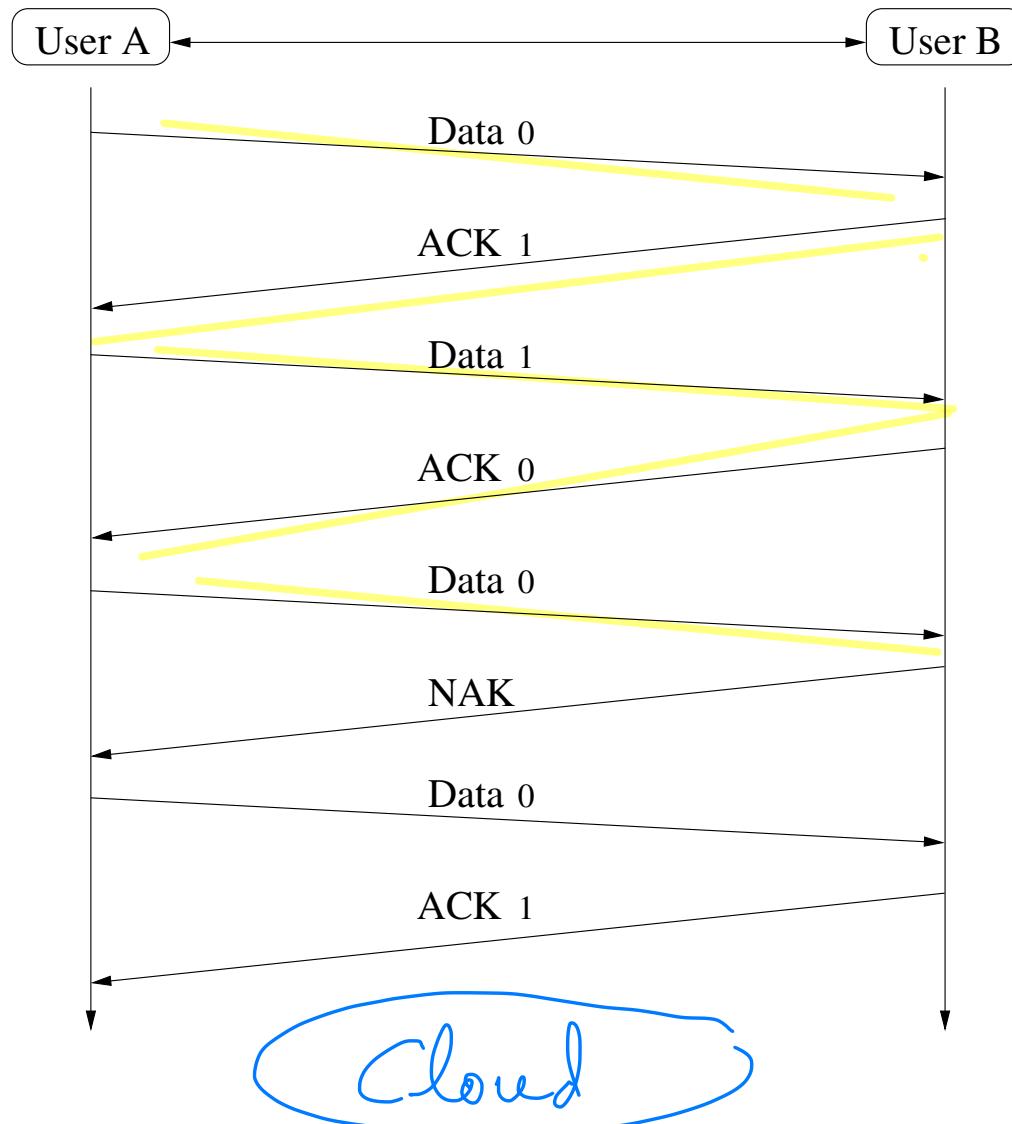
- Frames are labeled with a bit $b \in \{0, 1\}$.
- ACKs are also labeled with a bit $b \in \{0, 1\}$.
- $\text{ACK}[b]$ acknowledges frame labeled $b + 1 \bmod 2$ and indicates that the receiver is ready for frame labeled b , where $b = 0, 1$.

*If is sufficient to use
as label a single bit $\in \{0, 1\}$*

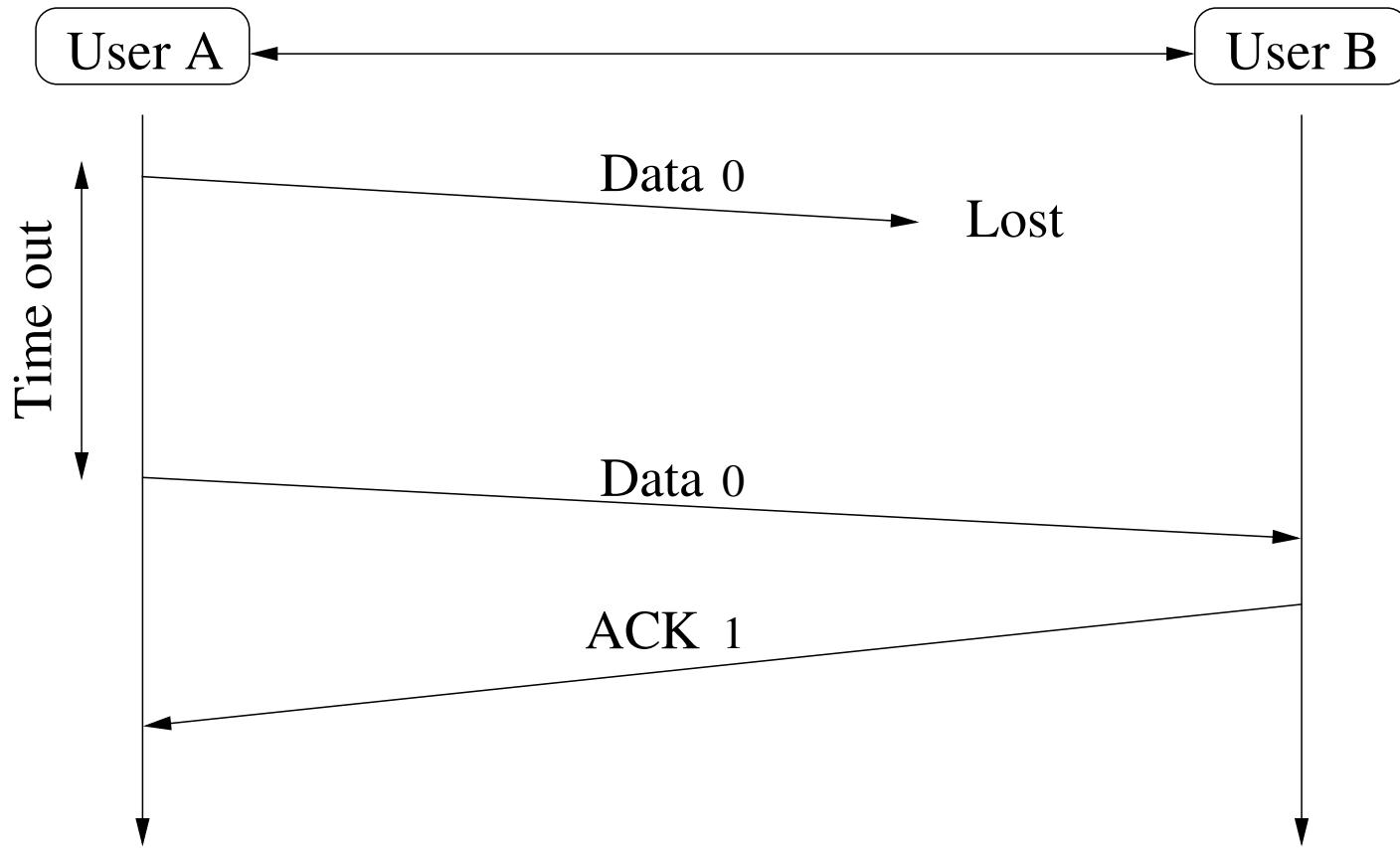
$$\begin{aligned} 0 + 1 &= 1 \\ 1 + 1 &= 0 \end{aligned}$$



Stop-and-Wait ARQ

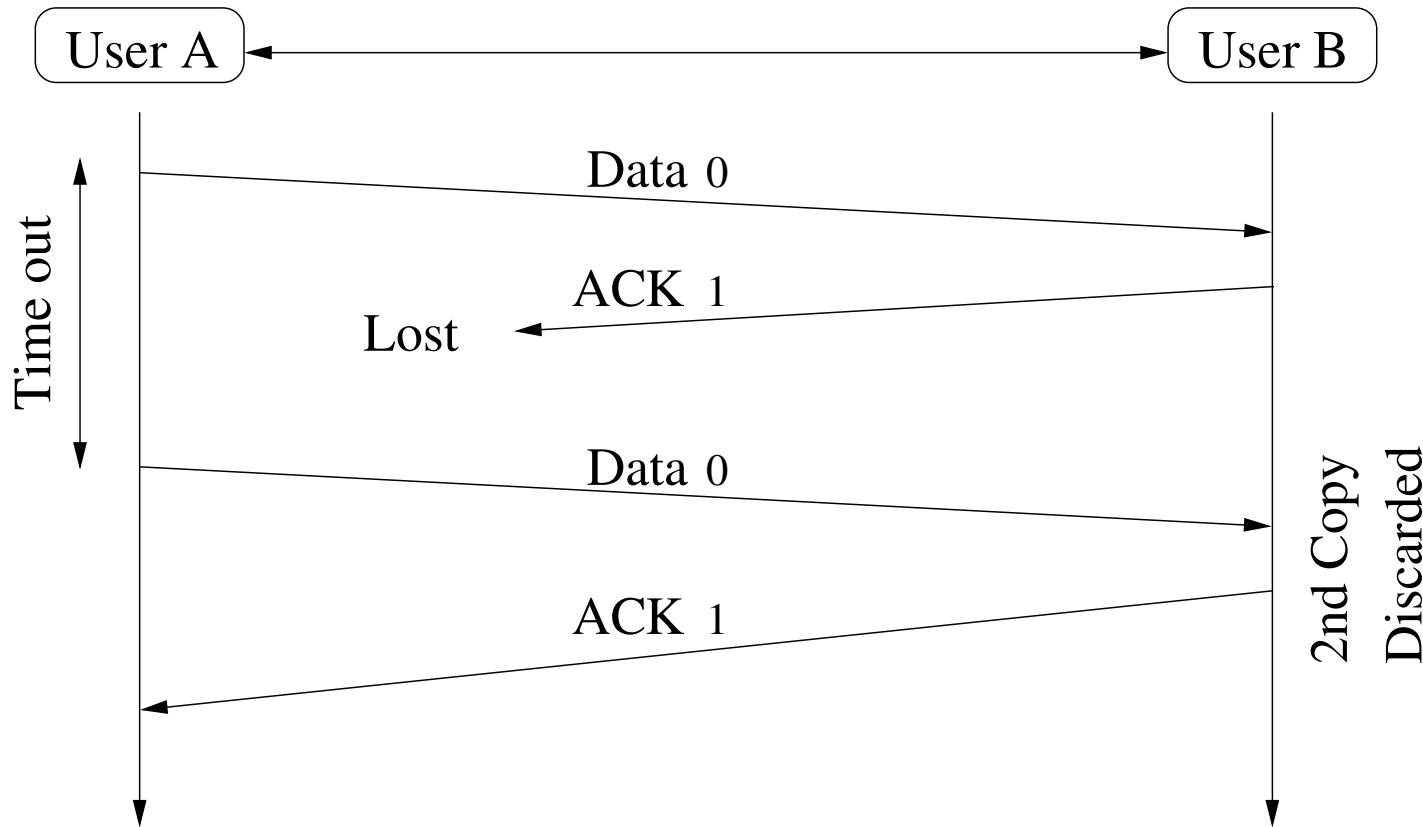


Stop-and-Wait ARQ: Lost Frame

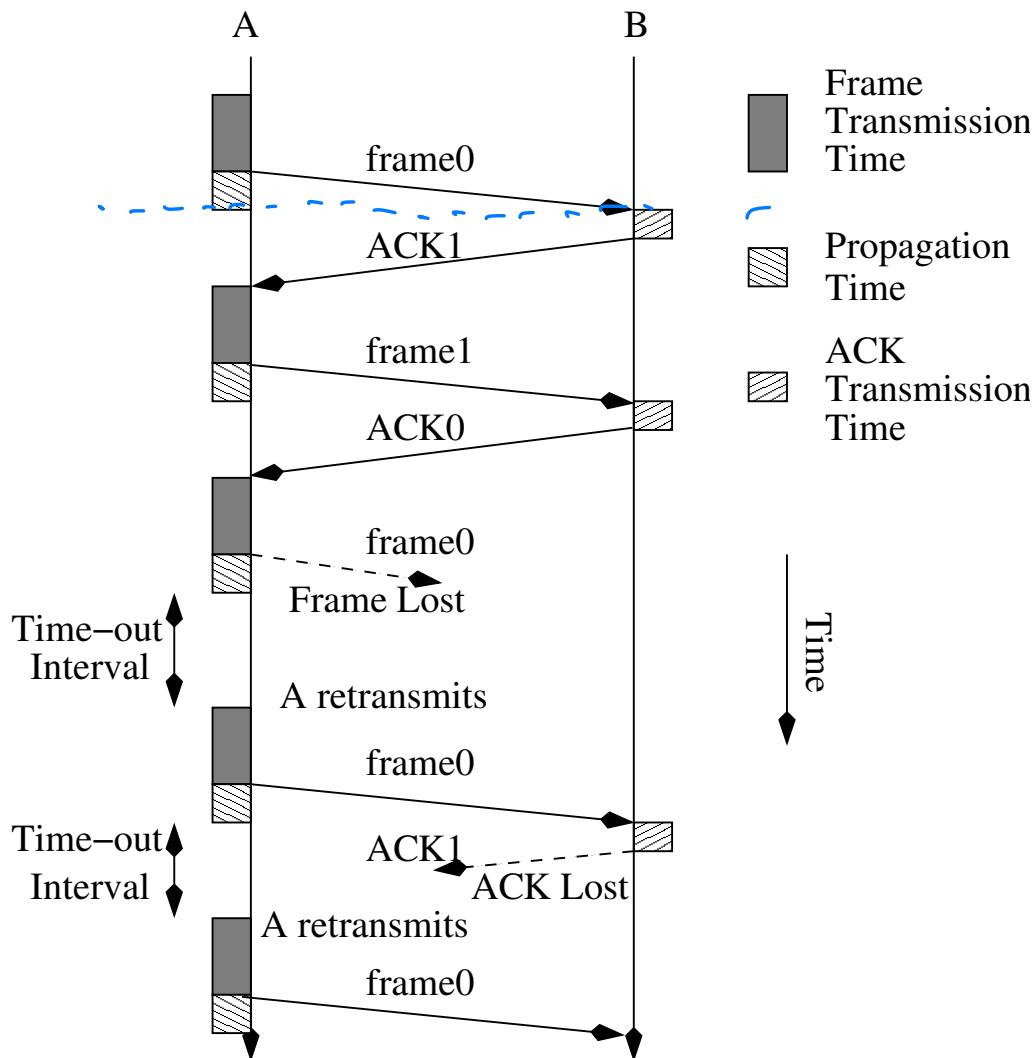


Time Out will be
within set limits.

Stop-and-Wait ARQ: Lost ACK



Stop-and-Wait ARQ



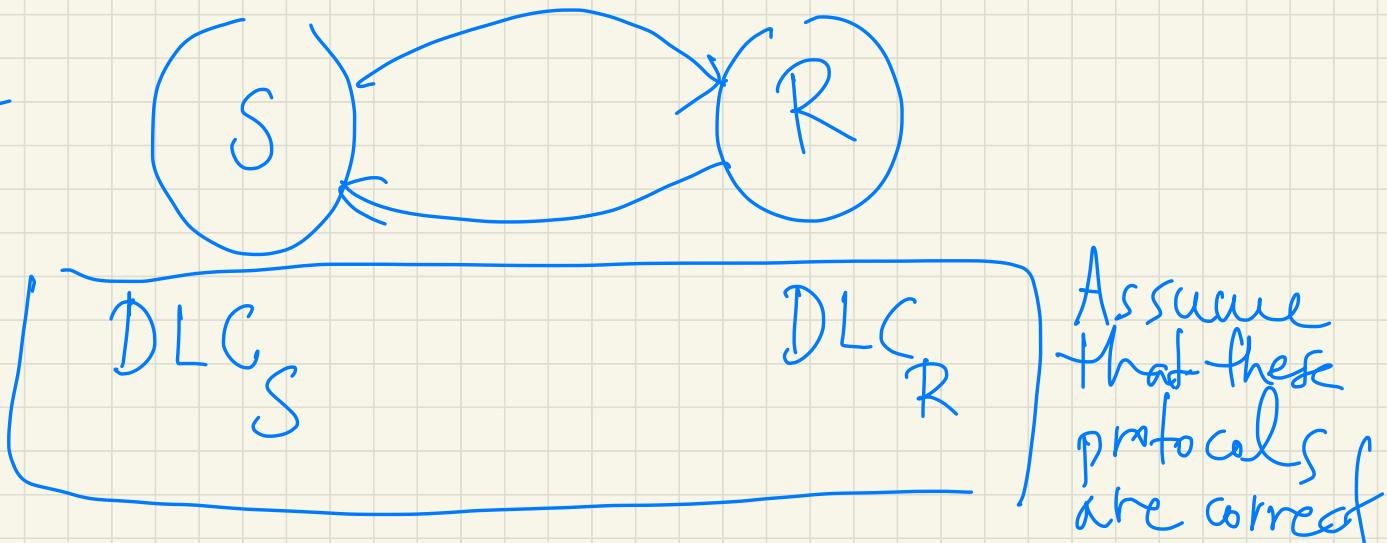
Correctness of Stop-and-Wait

- Can we show that a never ending stream of packets can be accepted from the higher layer at A to the higher layer at B in order and without repetitions and deletions?
You
- Initial assumptions:
 - 1. all error frames detected by CRC,
 - 2. for some $p > 0$ all frames are received error free with probability at least p ,
 - 3. link is initially empty,
 - 4. first packet from A has label $b = 0$,
 - 5. B is awaiting a packet with label $b = 0$.

Semantics: Program Verification

When you try to prove correctness

network
layer



Safety and Liveness of Stop-and-Wait

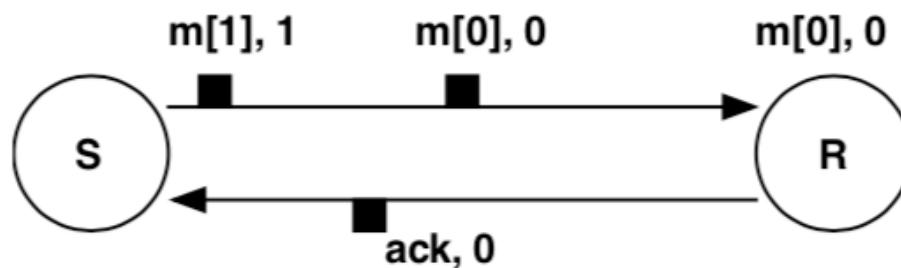
- Stop and Wait satisfies the following two important properties^a

- **Safety:** *Stop Wait*
Algorithm never produces an incorrect result.
In this case it means algorithm never releases a packet emanating from A to the layer at B out of the correct order.
 - **Liveness:**
Algorithm never enters a deadlock condition from which no further progress is possible.
In this case it means algorithm continues to accept for ever new packets at A and release them at B , and vice versa.

^aWe won't give a formal proof of these claims.

Alternating Bit Protocol (ABP)

- Stop and Wait is also referred to as Alternating Bit Protocol.
- ABP can be thought of a special version of the sliding window protocol, with window size one. Works only when the channels are FIFO, which rules out message reordering. It is a suitable candidate for application in the data-link layer.
- With FIFO channels, the alternating bit protocol overcomes this problem by appending only a one bit sequence number to the body of the message.
- Global state of ABP



ABP: Sender/Receiver

- Sender

```
define sent, b : 0 or 1; next : integer;
initially next=0, sent=1, b=0, and both channels are empty;
do   sent≠b           → send (m[next], b); next:=next+1;
      sent := b
      □ (ack, j) is received → if j = b → b := 1-b
          □ j ≠ b → skip
          fi
      □ timeout (R,S)       → send (m[next-1], b)
```

- Receiver

```
define j : 0 or 1;
initially j = 0;
do   (m[next], b) is received →
        if j = b → accept the message;
            send (ack, j);
            j := 1 - j
        □ j ≠ b → send (ack, 1-j)
        fi
od
```

GO-BACK-N ARQ

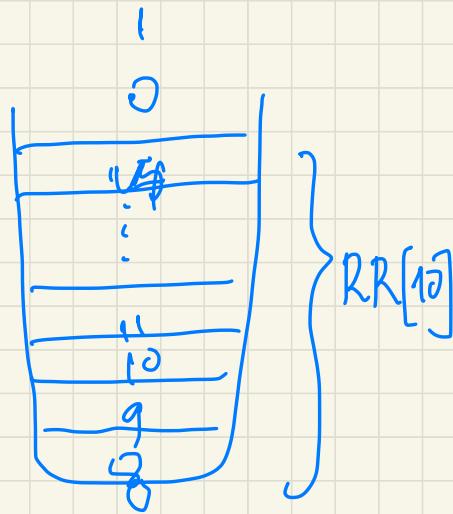
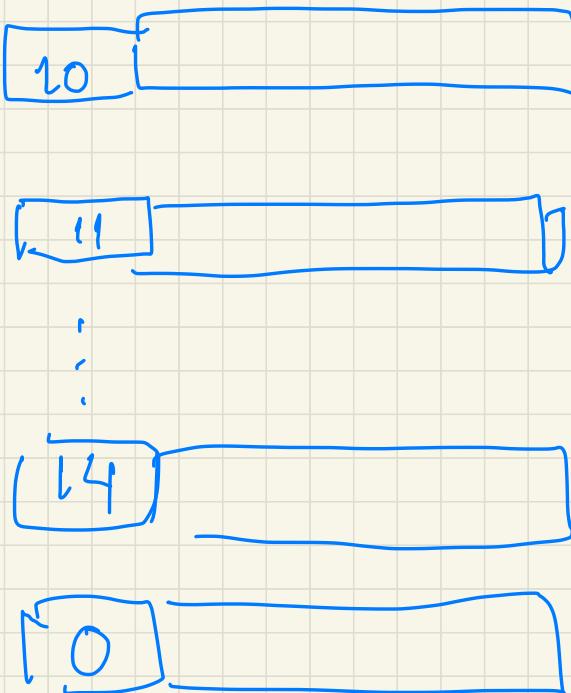
Go-Back-N is the most commonly used sliding window protocol.

- sender sends frames sequentially numbered modulo some max value
 - if receiver detects no error, an RR (Receive Ready) ACK is used for acknowledgement
 - if receiver detects error in a frame it sends REJ for this frame. This destination station will discard this and all future frames until the frame in error is correctly received.
- $N - 1$ frames may be sent before an ACK is received.
- Assuming stations A (sender) B (receiver) we have several cases to consider.

Unbounded sequence numbers is a major hurdle in implementing the sliding window protocols on **non-FIFO** channels.

$N = 15$:

Sequentially numbered modulo N

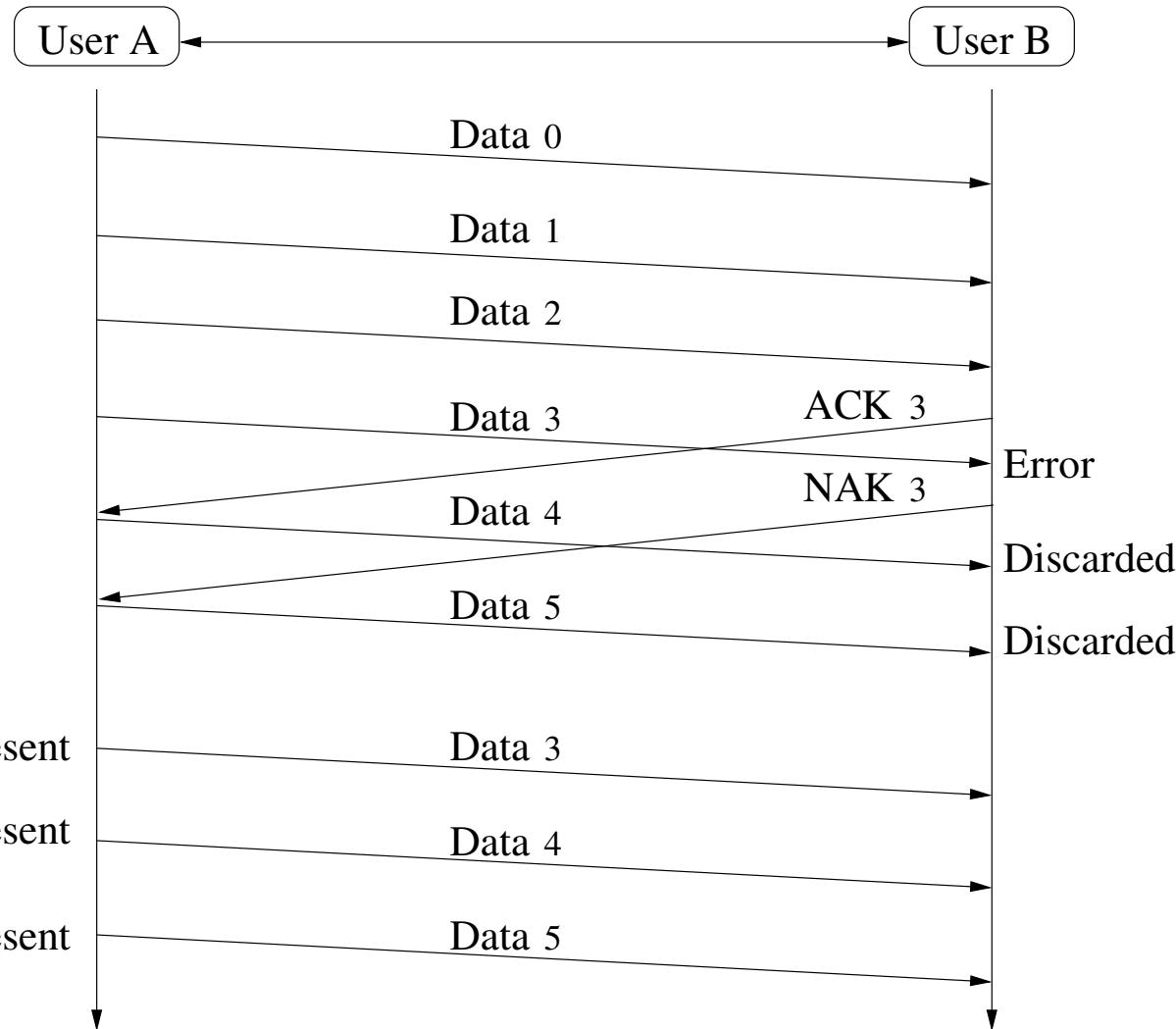


Buffer

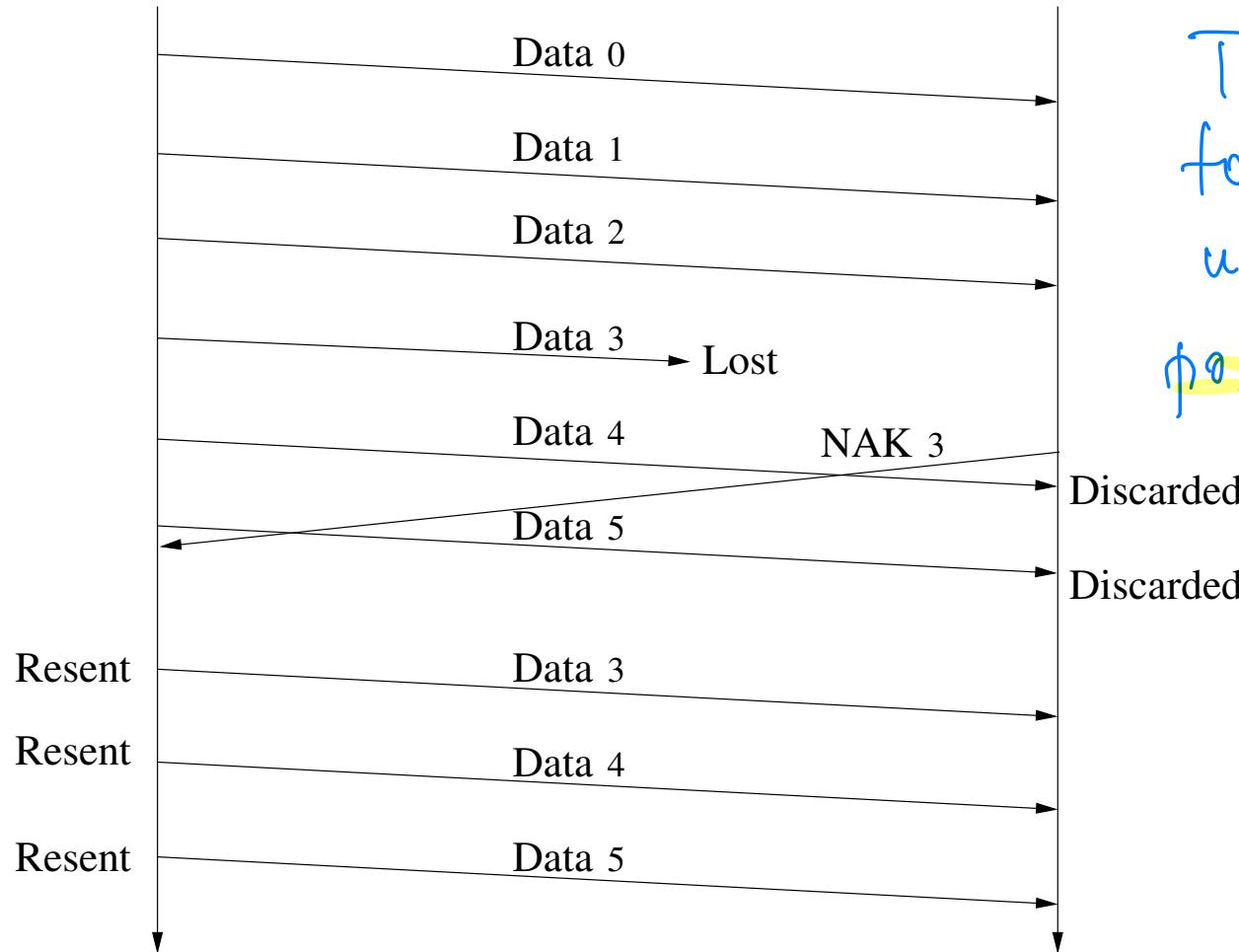
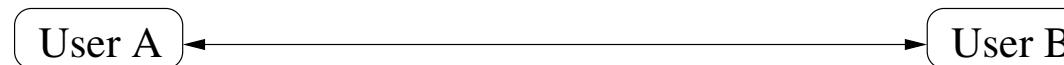
GO-BACK-N ARQ

- Sender keeps copies of all transmitted frames until they have been acknowledged. *where in its buffer*
- Receiver has the option to send either ACK or NAK if data frame is damaged. Because of the “continuity” of transmission both ACKs and NAKs are numbered. *also places seq numbers on ACKs*
- Sender is equipped with a timer in order to handle lost ACKs. Remember, $N - 1$ frames may be sent before an ACK is received. If $N - 1$ frames are awaiting acknowledgement the sender starts a timer and waits before sending any more!
- If allotted time runs out with no ACK received, the sender must assume frames were not received by receiver! *RTT*

GO-BACK-N ARQ: Damaged Frame

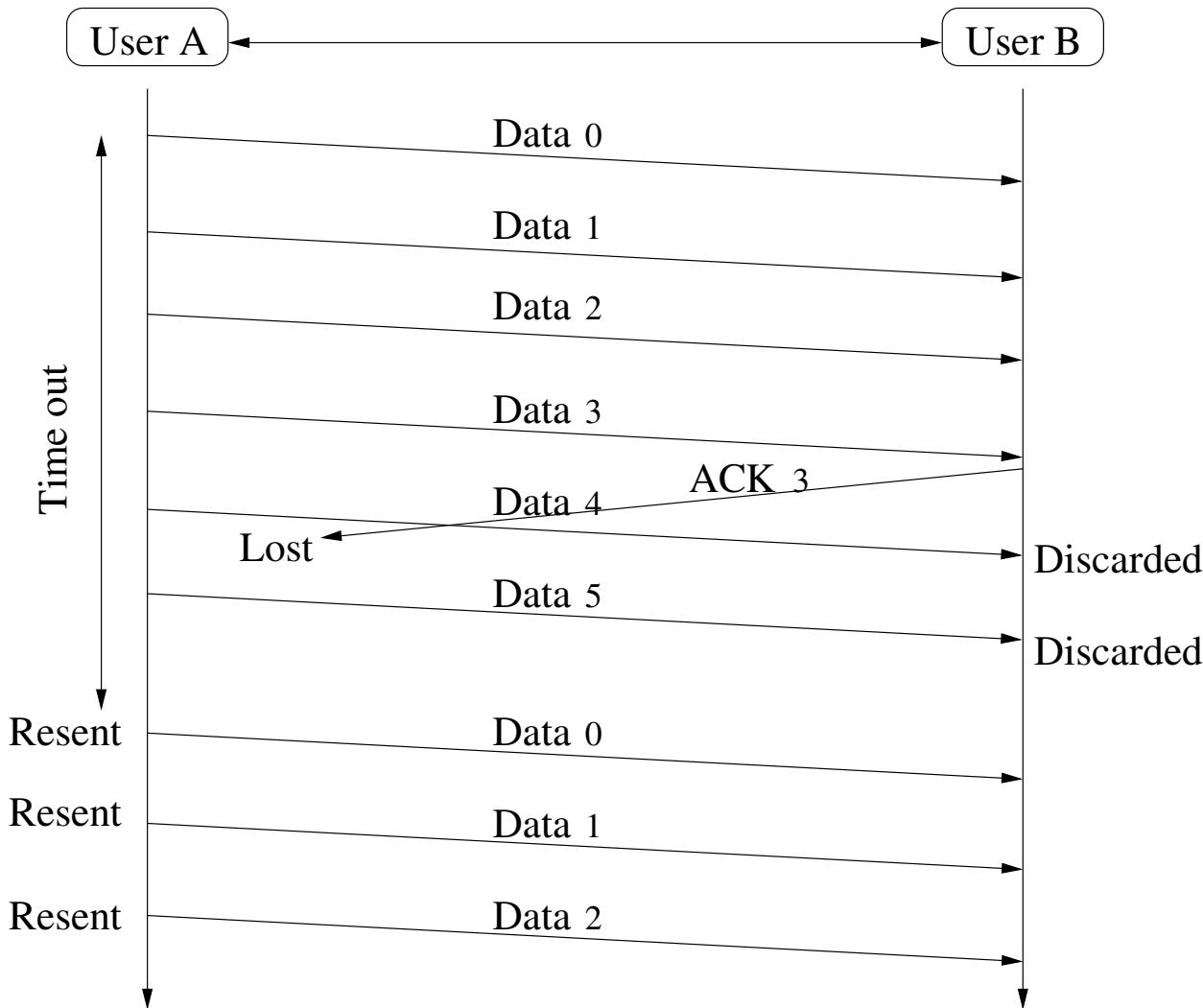


GO-BACK-N ARQ: Lost Frame



TCP likes to "say" if uses only positive ACKs .

GO-BACK-N ARQ: Lost ACK



Damaged Frame: Interpretations 1.



- **Case a.** B detects an error and has previously received successfully $frame[i - 1]$. B sends $REJ[i]$. (AE[$i-1$])
When A receives it must retransmit $frame[i]$ and all frames previously transmitted since original transmission of $frame[i]$.
- **Case b.** $frame[i]$ is lost in transit.
 A subsequently sends $frame[i + 1]$ which is received by B out of order, and hence B sends $REJ[i]$. A must transmit $frame[i]$ and all subsequent frames.
- **Case c.** $frame[i]$ is lost in transit and A does not soon send additional frames.
 B receives nothing: sends neither RR nor REJ . When A 's timer expires it transmits an RR frame and a P -bit set to 1; B interprets the RR frame with a P -bit 1 as a command to be acknowledged by sending an RR indicating the next frame that it expects. When A receives RR it retransmits $frame[i]$.

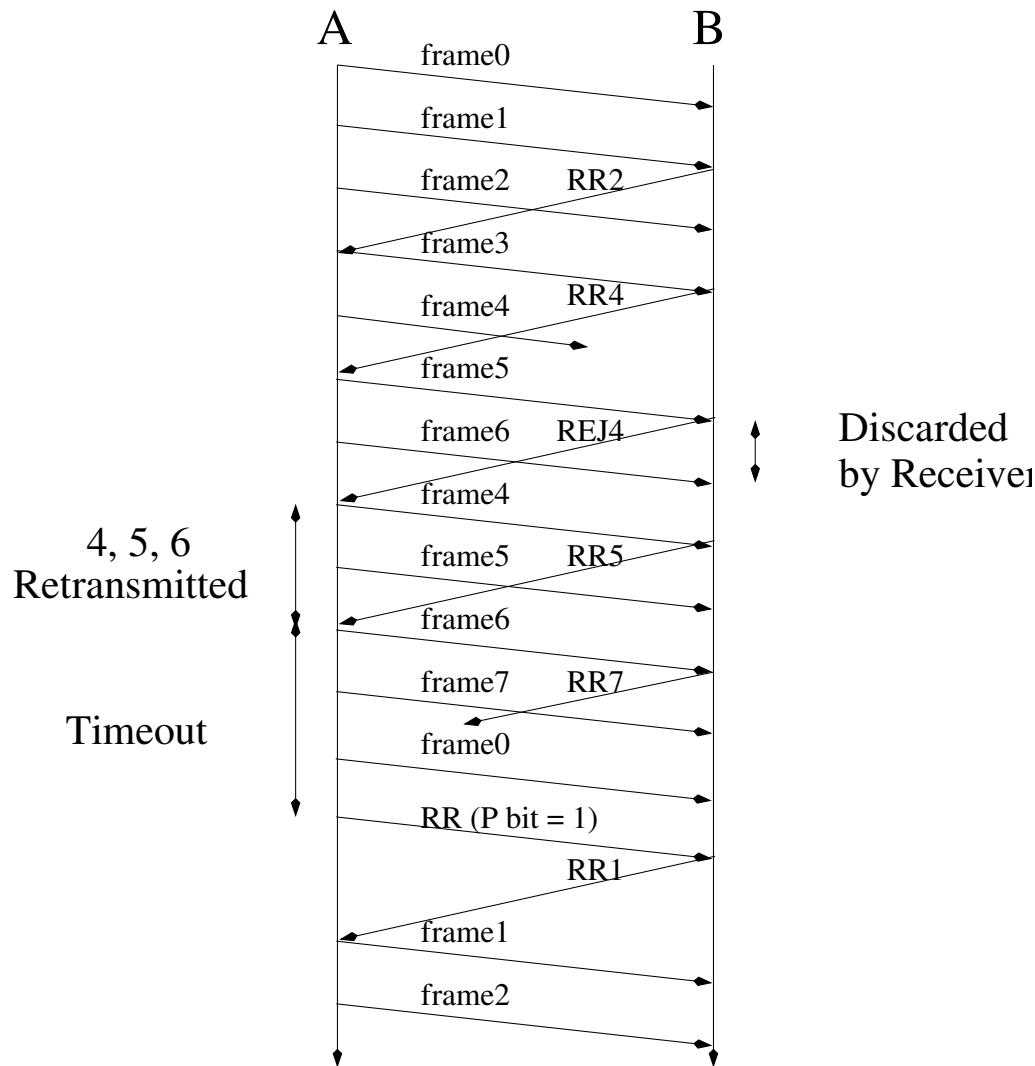
Cumulative

Damaged *RR* and *REJ*: Interpretations 2.

- a. B receives $frame[i]$ and sends $RR[i + 1]$ which is lost in transit. Because ACKs are cumulative (e.g. $RR[i]$ means all frames through B are acknowledged) it may be that A will receive a subsequent RR to a subsequent frame and that it will arrive before the timer associated to $frame[i]$ expires.
- b. If A 's timer expires it retransmits an RR command as in case 1.c. It sets another timer, called P -bit timer. If B fails to respond to the RR -command or if its response is damaged then A 's P -bit timer will expire. At this point A will try again by issuing a new RR -command and restarting the P -bit timer. This procedure is tried some maximum number of attempts. Failure of all these initiates a reset procedure.

Damaged **REJ** is similar to case 1c.

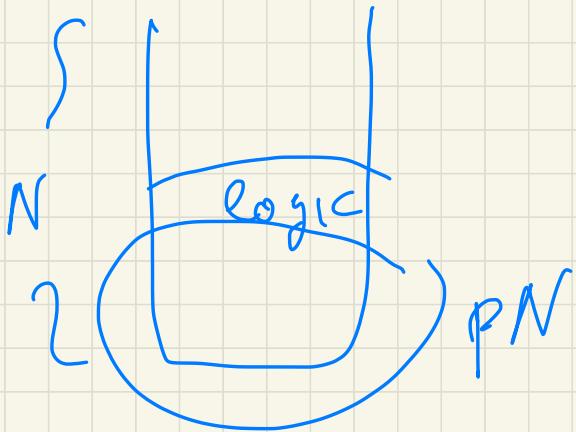
Go Back N ARQ



SELECTIVE REJECT: SREJ

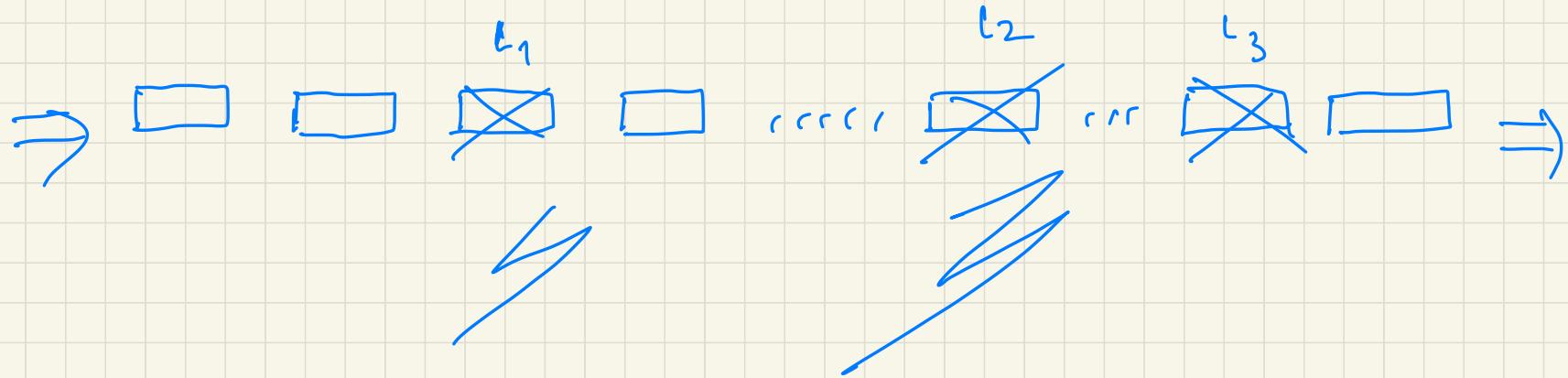
- Unlike Go-Back- N ARQ, here retransmission is not cumulative.
- Only the specific lost frame is retransmitted (out of order)!
 - The receiving device must contain *sorting logic* to be able to reorder frames, and must be able to store frames received after a NAK.
 - The receiving device must contain a searching mechanism to be able to find only the requested frame.
- In general, it requires smaller window size than Go-Back- N .

It works well in cases
where you have few errors



Errors w p p
p N

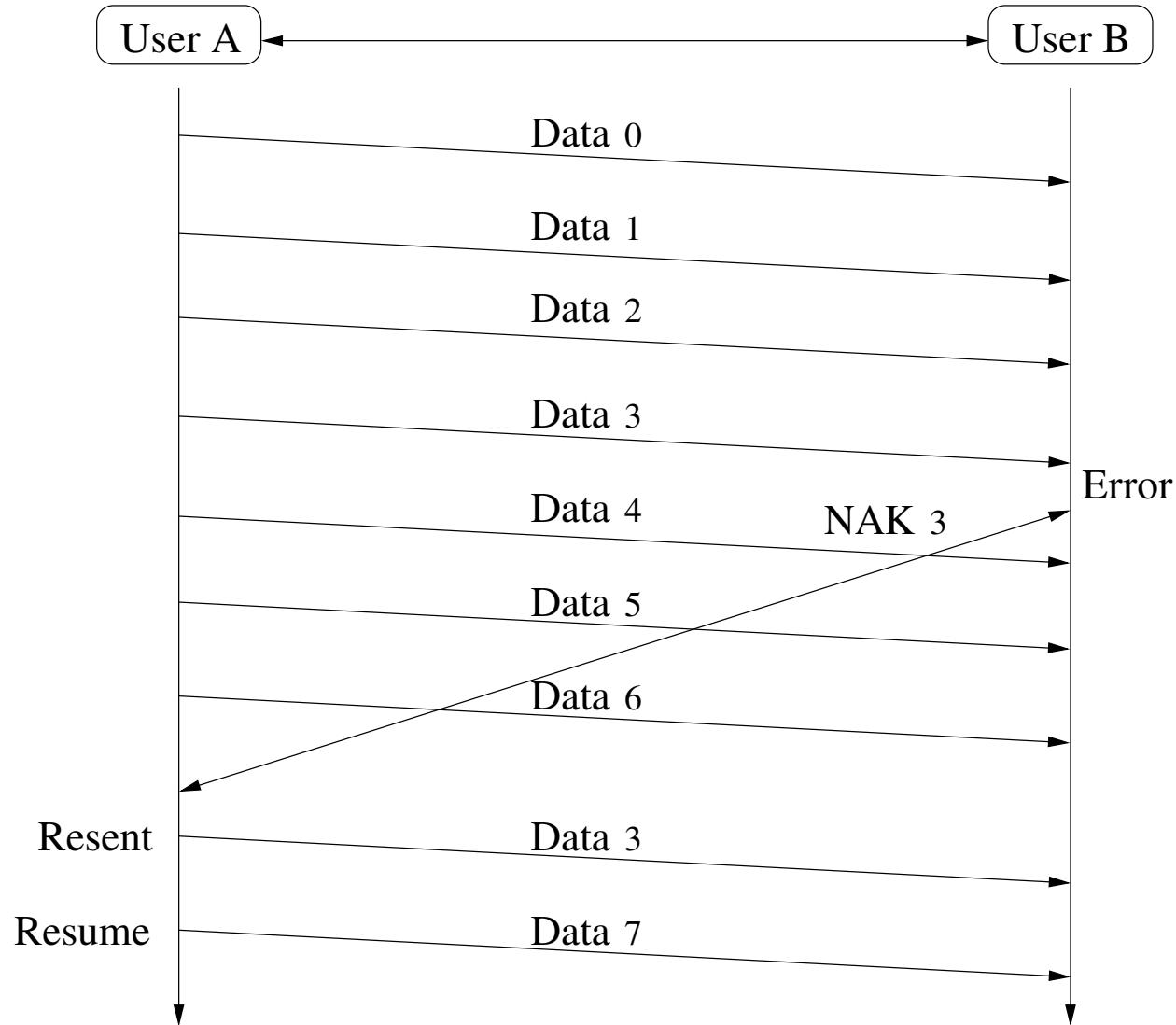
Problem in SREJ



$SREJ[c_1]$, $SREJ[c_2]$, $SREJ[c_3]$..

If creates overhead
"logic"

SELECTIVE REJECT: SREJ



Estimates on Overhead

Go - Back-N

$$\text{prob } p \\ \boxed{p N} = N_p$$

In the worst-case you may have to sort them.

Sorting Alg: $N_p \log N_p$

Logic Alg. L_p

Transm. Cost $\overline{T_p}$

$$(G_{st_{sw}} + N_p \log N_p + L_p + \overline{T_p}) / G_{st_{sw}}$$