Unit 1

Digital Information

- (Computers represent data (e.g. text/images) through binary data (1s & 0s)

Data & Binary:

on ← bit → off

yes ↓ no

true false

1 0

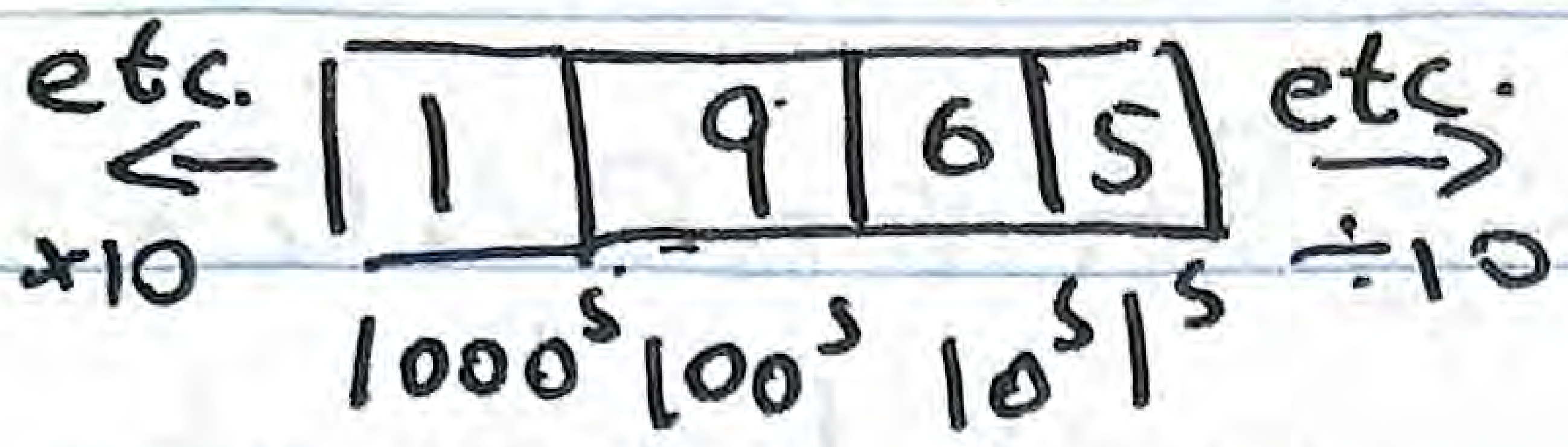
↓ Smallest piece of information that a computer can store

- More bits require more on/off wires and can be used to store more complex information

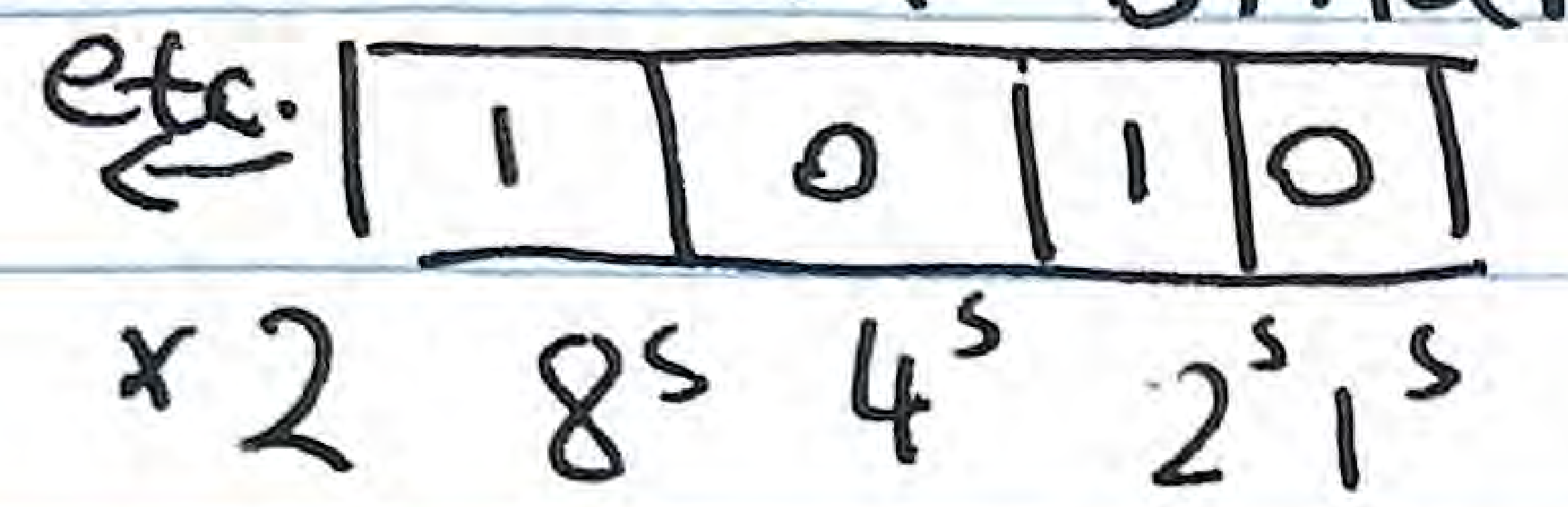
Binary Number System:

- Decimals require 10 digits (0,1,2,3,4,5,6,7,8,9)
- With Binary, only 2 · (0,1)
  - ↳ Still be able to represent any #

Decimal Number System:



Same with Binary:



e.g. Decimal 9 = Binary 1001

Thus, ANY NUMBER can be represented by 1s & 0s or wires that are on or off.

More wires/digits = bigger numbers.

To represent other data, e.g. text/images.

- For text, you can assign a letter to a number. E.g. ASCII or Unicode

A	1	00001
B	2	00010

- For images, you can represent its pixels and its colours with numbers.
- For sound, they have wave forms/frequencies, represented through a series of numbers

No.

Date

Binary & Data:

1s and 0s are the BACKBONES of how computers work (i.e. input, store, output, and process.

What fits in a bit?

- A single bit can only represent two different values. That's not very much, but that's still enough to represent any two-valued state

Sequences of Bits:

- Computers use multiple bits to represent more complex data:

Two bits can represent 2^2 amount of distinct values.

Three bits can do 2^3

x bits can do 2^x

Bytes are units of digital info.

1 Byte = 8 bits

Importance: Computers do process all data as bits, but they prefer to process bits in byte-sized groupings.

No.

Date

Binary Numbers

Decimal Numbers: How they work:

234 =	2 hundreds' 100 10^2	3 tens' 10 10^1	4 ones' 1 10^0	$= (2 \times 100) + (3 \times 10) + (4 \times 1)$

Binary Numbers: How they work:

0001 =	0 8 2^3	0 4 2^2	0 2 2^1	1 1 2^0	$= (0 \times 8) + (0 \times 4) + (0 \times 2) + (1 \times 1) =$ 1 decimal

E.g. $\underbrace{1}_{8}\ \underbrace{0}_{4}\ \underbrace{1}_{2}\ \underbrace{0}_{1}$ = 10 decimal

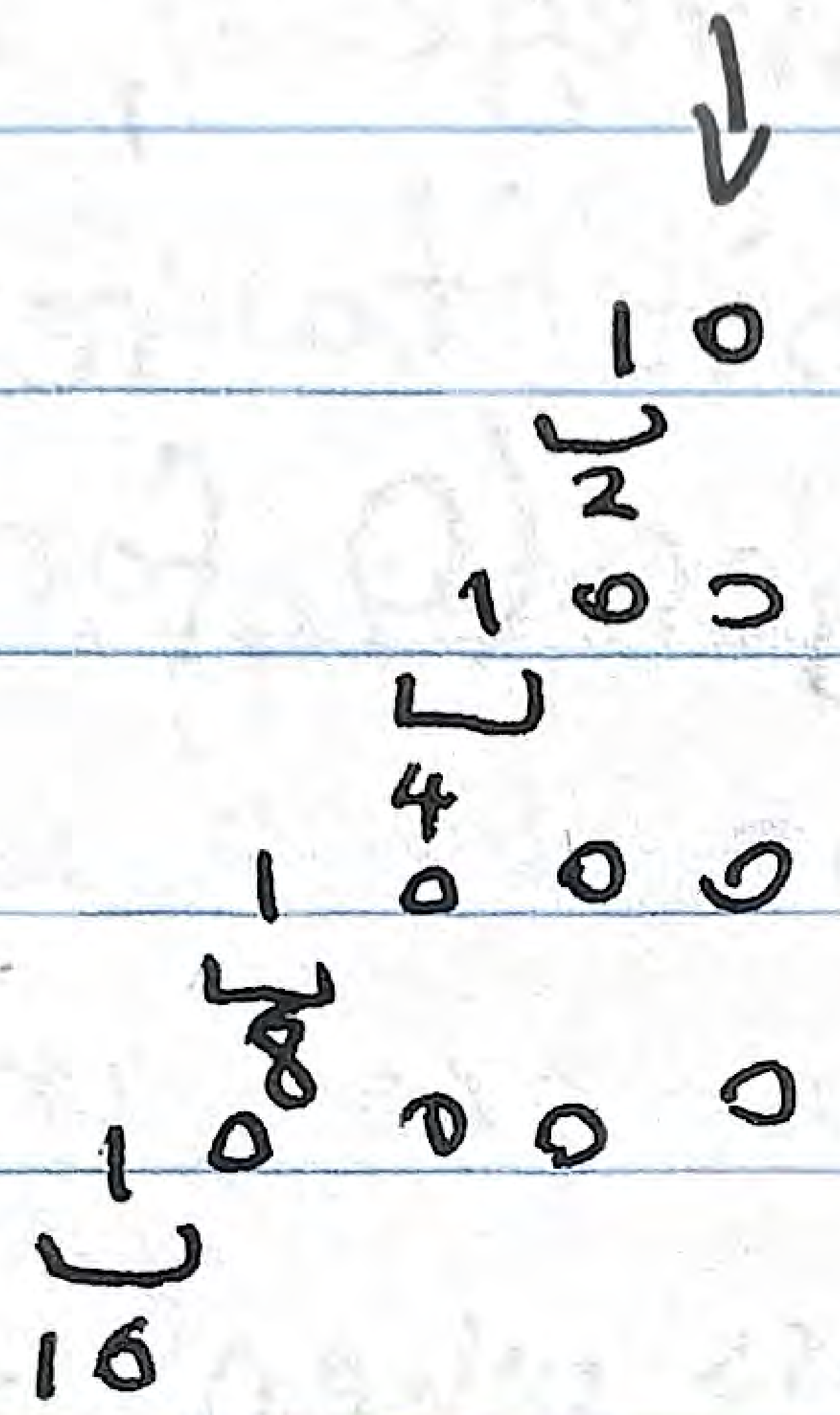
8 + 2 = 10

Patterns in binary numbers:

Decimal	Binary
3	0011
5	0101
7	0111
9	1001

All are 1s when odd

Decimal	Binary	
1	1	2 - 1 = 1
3	11	4 - 1 = 3
7	111	8 - 1 = 7
15	1111	16 - 1 = 15



Limitations of storing numbers

Integer Representation: An integer is any number without a fractional component (e.g. 120, 10, 0, -20)

To represent integers, computers usually use the first bit to represent the sign of the integer (0 for +, 1 for -)

Overflow

An error that occurs when a number exceeds the amount of bits or resources the computer provides

↳ May report an "overflow error".

↳ May display a message like "number is too large"

May just truncate the number or wrap the number around (restart from 1)

E.g. 7 → 8 for 4 bits (where one is for ±)

So...

it may store as 1 or just stops at 7

Floating-point Representation: To represent or store fractions and irrational numbers.

↳ Also sometimes used for scientific integers

↳ is similar to "scientific notation"

In floating-point representation, a number is multiplied by a base raised to an exponent

↳ Since computers are using the Binary System, the base is 2.

Eg. $128 = 1 \times 2^7$ $\quad 0.50 = 1 \times 2^{-1}$

$160 = 1.25 \times 2^7$ $\quad 0.750 = 1.5 \times 2^{-1}$

Once the computer determines the floating point representation for a number, it stores that in bits.

Modern Computers use a 64-bit system where 1 is for the sign, 11 for the exponent, and 52 for the number in front

No.

Date

E.g. π

Roundoff errors - Errors that occur when numbers require too many bits to be stored.

E.g. $\frac{1}{3} = 1.\overline{3} \times 2^{-1}$

So... in binary, $0.\overline{3}$ is an infinitely repeating sequence meaning a computer has to end the number eventually due to it being impossible to store an infinite sequence in a computer

More bits = more precise numbers and calculations will be.

Less bits used in calculations make this roundoff error noticeable

Storing text in binary

To store text or any other symbol, we use encodings, which maps one character to a binary number counterpart.

How encodings work:

- ->When a program needs to store a specific character, it instead stores the encoded binary number instead.
- ->When a program needs to display the encoded binary number, it displays the decoded character instead.
- ->When a program needs to store several characters, they can string each character's encoding together.

ASCII encoding (one of first standardised encodings)

- ↳ Encoded in Binary using 7 bits
- ↳ First 32 codes represent "control characters" which executes a command other than printing a letter.

No.

Date

Problems with ASCII:

- ↳ Only includes letters from English
- ↳ Had a limited set of symbols
- ↳ No cross-language compatibility
- ↳ ASCII uses 7 bits whilst computers use 1 byte or 8 bits

Unicode (1987, solved problems with ASCII)

- ↳ Assigns each a "code point" (hexadecimal #) and a name to each character
- ↳ Saved space by unifying characters across languages
- ↳ Started with 7,129 in 1991
- ↳ Grown to 137,929 in 2019
- ↳ Includes over 1,200 emoji symbols

However... Unicode is a character set and not an encoding

UFT-8 (1992, compatible with ASCII and solved its problems)

↳ Can describe every character from Unicode using 1~4 bytes (adaptive)

↳ A computer knows how many bytes represent the next character based on how many 1 bits it finds at the beginning of the byte.

No. of bytes	Byte 1	Byte 2	Byte 3	Byte 4
1	0xxxxxxx			
2	110xxxxx 2	10xxxxxx		
3	1110xxxx 3	10xxxxxx	10xxxxxx	
4	11110xxx 4	10xxxxxx	10xxxxxx	10xxxxxx

If there are no 1 bits in prefix (i.e. first bit is 0), it means character can be represented by a single byte.

↳ Remaining 7 bits used for ASCII

2 bytes - Latin-script languages + others (e.g. Greek, Arabic)

3 bytes - Most Asian languages

4 bytes - everything else (historical scripts, emojis)

No

Date

Analog data

→ Continuous stream of varying data

→ Infinite amount of . . . information

↳ Analog data is infinitely detailed

Converting to Binary Data

1. Sampling ↗ product is a table of values

- Sample or record data at regular intervals
- Inverse of Sampling Interval is sampling rate (# of samples in a second)
- Nyquist Shannon sampling theorem:

↳ Sufficient sampling rate is anything larger than the highest frequency in the signal.

↳ Cycles per second (Hertz - Hz)

2. Quantization

↳ Reduces the continuous amplitude domain into discrete levels. (values approximated)

3. Binary encoding

↳ Stores much smaller value that represents the quantized y value

Abstraction - Process of reducing complexity by focusing on the main idea

E.g. Analog real-world to digital

Bits are grouped to represent abstractions

Compression - Algorithm that shrinks the file size needed to represent a file (less storage space)

fully reconstr.

→ lossless - Reduces size without losing any information in the file (can reconstruct)

→ Lossy - Reduce size by discarding less important info (can only be reconstructed as an approximation)

Compression Algorithm - Finding repeated, redundant information and replacing it with shorter representations

E.g. "To be or not to be"

⇒ "θα or not θα"

↳ A dictionary will be created to store the actual representation. (lossless representation)

No.

Date

Image Compression - Run-length encoding (RLE)

↳Always start with # of white pixels

Compression algorithm is used where the computer replaces each row with numbers that say how many consecutive pixels are the sme colour.

-RLE is useful with icons of limited color palettes

Huffman coding algorithm - By assigning a shorter bit for a letter/symbol that is more repetitive, you can also save space

Lossless vs **Lossy**

Lossless

↳When the loss of text or numbers can change info.

- Exe files
- Text
- Spreadsheet

Lossy

↳When partial removal of data has little or no effect to the representation of it

- Images (Graphics)
- Audio
- Video

No

Date

Which type of compression?

- Need to recover original: Lossless
- Data size/transmission time: Lossy

Data

- Collection: Consider Source & Tools to analyse data
- Processing: How much info can we get?
  Do we need parallel processing?
- Bias: Intentional (Who, for what?)
  Unintentional (Who, how is data collected?)
- Data Cleaning: Identifying and replacing, modifying or deleting incomplete, corrupt, duplicate or inaccurate reccords
  ↳ Be careful when modifying or deleting

Metadata - It's the data about data
(e.g. Author, Date, length, size)

- Patterns: Data allows us to look for trends and patterns, answer questions
  ↳ Some can be misleading
  - Correlation does not imply causation

Copyright Laws

↳ Laws that protect the creative works of an author

- Only ideas that were expressed, not ideas

·Rights: Reproduce work
Create derivative works
Distribute copies by sale/rental
Public display of work
Public perform of work

Copyright laws expire after a specific duration after death.

↳ They then enter public domain (use by anyone without restriction).

Fair Use - Allows the limited use of copyright materials for purposes like criticism, comment, news reporting, teaching, or research.

Copyright is especially important in the digital age where anyone can reproduce anything without realizing it.

Digital Rights Management

- A tool that restricts where and how a user can use copyrighted material.

Digital Millennium Copyright Act (DMCA)

- Criminalizes production and distribution of technology that tries to circumvent DRM
  - ↳ Copyright owner can send takedown notices if copyrighted works of owner are violated

No.

Date

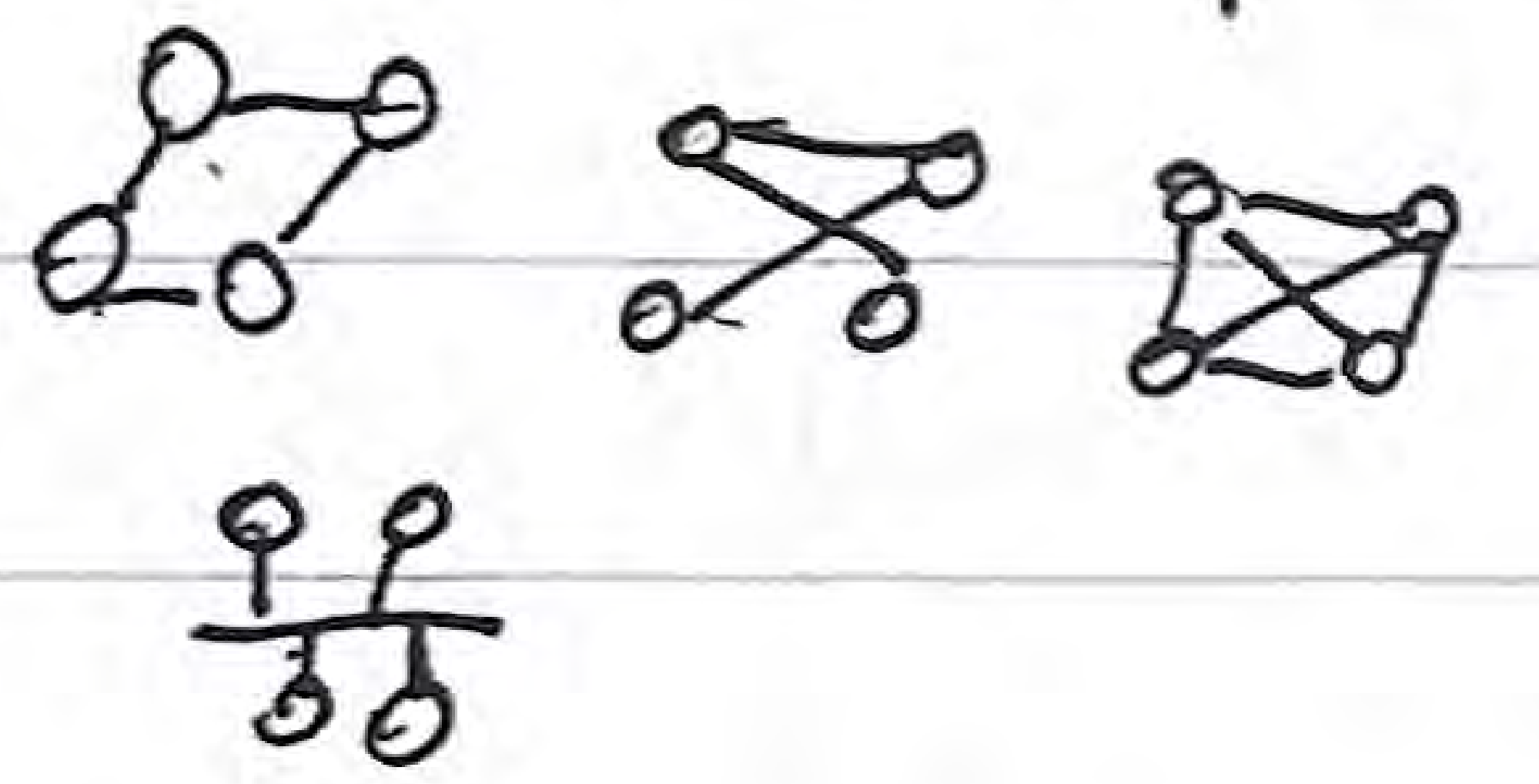
Computer Networks

- The internet is the world's largest computer network

- Computer netwook - Any group of interconnected computing devices (capable of send/receive)

- Computing Device - Any device that can run a program

- Network Topology - The different ways to connect devices together in a computer network.



Local Area Network (LAN) - Covers a limited area like School or house

Most common type

Wide Area Network (WAN) - Extends over a large geographic area (composed of many LANs)

largest type

Data Center Network (DCN) - Network for data centers (little delay)

No.

Date

Protocols - Allows computing devices to communicate with each other in a network.

↳ E.g. Devices must use the Internet networking protocols if they want to communicate over the internet.

The internet

- Global network of computing devices

Bit Rate

- Speed at which the network connection sends every second
(bits each/per second)

Bandwith

- Maximum bit rate of a system

Latency

- Time between the sending of a data message and the receiving of that message
- Limiting Factor: Nothing can travel faster than the speed of light

No.

Date

IP Address

The internet protocol (IP) is one of the core protocols in the layers of the internet.
- To handle addressing and routing

IP address uniquely identifies internet-connected devices

IPv4 addresses
- First version ever used on the Internet
- 4, 3 digit numbers (values 0-255)
  - Each number can represent 2^8 values (8 bits)

IPv6 addresses
- Uses Hexadecimal numbers

Hierarchy (IP Address)
- Makes it easier to route data

E.g. $\underbrace{24.147}_{\text{Comcast network}}.\underbrace{242.217}_{\text{Home computer}}$

Subnets - IP addresses can be further broken into subnets

IP Packets

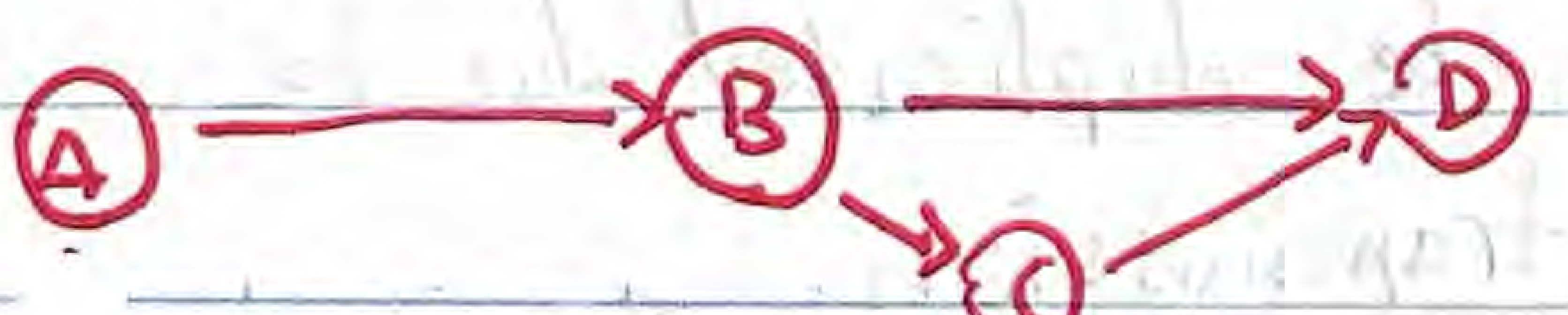
Due to limitations as to how much data can be transmitted at once, networking protocols split each message into multiple small packets.

- IP Protocol describes the structure of the packet
- IP Packet contains a header (20-24 bytes) and data (variable length)
  - -> Header: Source, destination
  - -> Data: Content

Internet Protocol (IP) describes how to route messages from one computer to another.

Redundancy in Routing

- -> There are a lot of redundant paths in networks to increase the # of possible routes a packet can take to reach its destination
- -> This allows the network to continue even if one segment is unavailable



2 different ways from A to D

No.

Date

Fault Tolerance

-> Fault tolerance describes how a system can experience failures in its components without malfunctioning.

-> Biggest contributor is the redundancy in network routing paths.

↳ No single point of failure

Problems with Packets

- IP does not handle all the consequences of packets, however.
  - A computer might send multiple messages to a destination, and the destination needs to identify which packets belong to which message.
  - Packets can arrive out of order
  - Packets can be corrupted (received ≠ sent)
  - Packets can be lost due to problems in the physical layer or in routes-fowarding tables
  - Packets might be duplicated due to accidental retransmission

Protocols

- There are protocols designed to solve problems with Packets

1. User Datagram Protocol (UDP)

↳ Fast but only solves corrupt data in packets.

↳ Each IP packet data portion is formatted as a UDP segment

- Contains 8-byte header and variable length data
- Uses Checksum to check for data corruption. (Adds data in binary, stores it, Computes check after received, check if result is same)

2. Transmission Control Protocol (TCP)

→ Solves corrupt, out of data, duplicate, lost packets

→ Data portion formatted as TCP segment

↳ Using TCP, two computers must establish a 3-way handshake connection (Synchronise, Acknowledge, Acknowledge)

↳ Then data is sent, recipient returns Acknowledge (also sends Sequence # and Acknowledgement # to avoid errors)

- TcP detects lost data with timeouts (resends if not received)
- Out of Order Packets are fixed by comparing sequence #'s.

The World Wide Web

- Massive network of webpages, programs, files accessible with URLs
- Uses protocols to load webpages

1. Domain Name System (DNS) protocol
   - Domain name to IP address
   - Domain name anatomy

     [Third-Level-Domain].[Second-Level-Domain].[Top-Level-Domain]

     ⇓ m . wikipedia . org ⇐

     → Computer checks local cache (for commonly visited websites), then ISP Cache (for commonly visited websites by internet service provider), and finally name servers

     ↳ Root Name → TLD Name → Host Name

2. Hypertext Transfer Protocol (HTTP)
   - Download visited page from another computer somewhere on the internet
   - Browser requests HTTP and receives response with headers and HTML file to render

Scalability of the Internet

- What increases the scalability?
  - → Any computing device can send data around the Internet if it follows protocols
  - → IPv6 can uniquely address a trillion times # of devices
  - → Routing is dynamic, new routers can join anytime to help move packets.
- What hinders scalability?
  - → Network connections have limited bandwith (delays or dropped packets)
  - → Routers have limited throughput (10 GBPS)
  - → Wireless routers have limitation in # of devices connected

To avoid possible outages engineering teams can prepare for spikes with load testing

The internet protocol suite

Many protocols power the internet, operating at different layers that build functionality on top of the layer below it.

Application Layer	HTTP	TLS	DNS
Transport Layer	TCP	UDP	
Network Layer	IP(v4, v6)		
Link Layer	Ethernet	Wireless LAN	

Open Protocol Development

- standardization Importance
  - → Without standardisation, computing devices may interpret messages differently and thus communication fails
- Importance of being open (non-proprietary)
  - → No central unit controlled the internet, allowing it to grow organically.

The global digital divide

Different countries/regions around the world have varying levels of access to the internet.

-> Some have more users and higher internet speeds, others do not.

↳ Due to higher infrastructure costs, etc.

-> Difference in access is referred to as the digital divide.

Digital Literacy

- Basic digital literacy includes the ability to use input and output devices, an understanding of the structure of the digital environment, and the ability to interact with digital information.

-> There are large differences in how effectively people can use digital technology, known as the digital use divide.