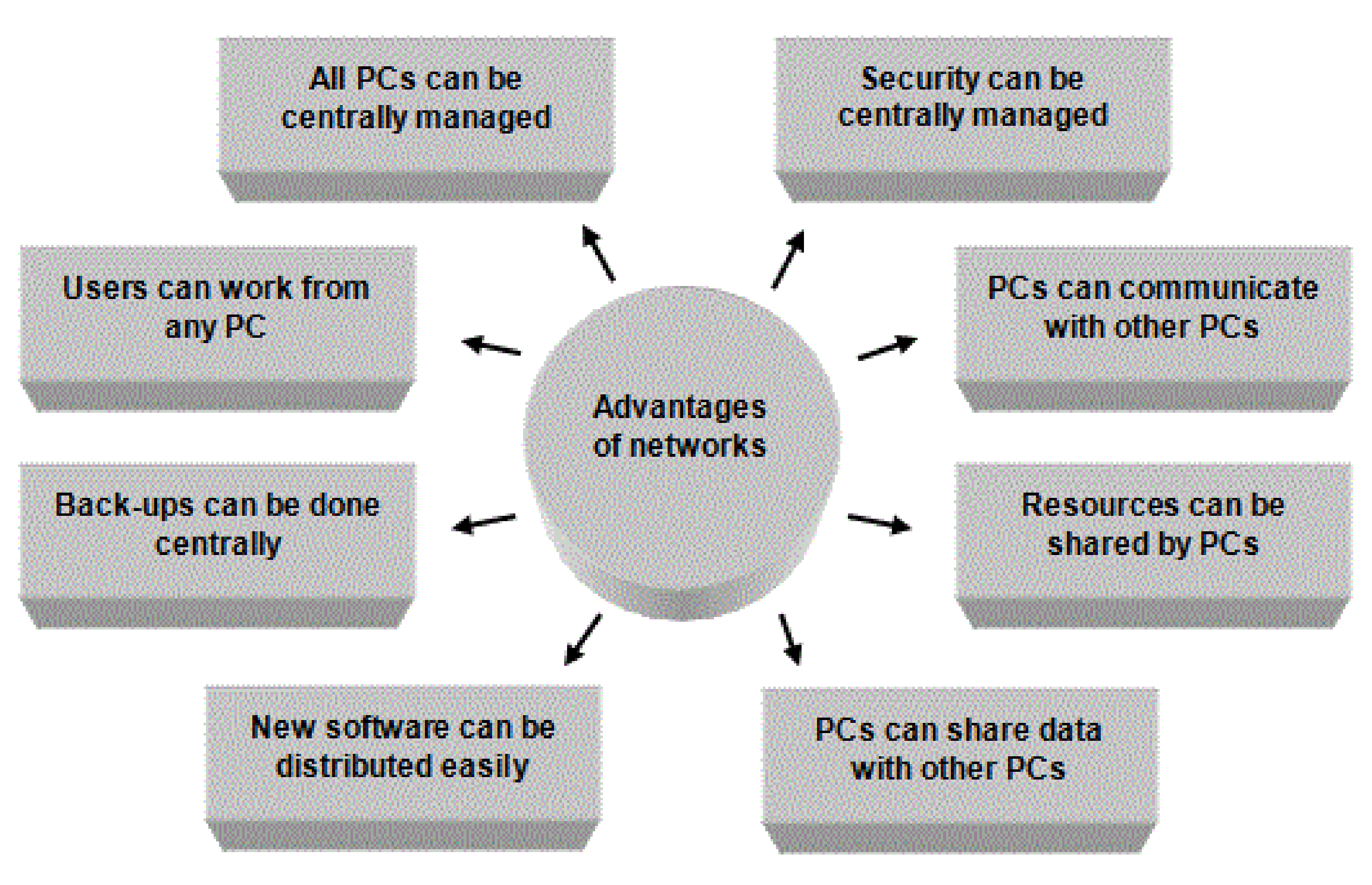
Networks

Learning Aims

❑Characteristics of networks
❑The importance of protocols and standards
❑Protocols and The TCP/IP stack
❑LANs and WANs
❑Packet v circuit switching
❑Data transmission
❑Network security and threats, and using firewalls, proxies and encryption
❑Network hardware
❑Client-server and peer-to-peer networks
❑Domain Name Servers

Key features of networked computers

The advantages of a network of computers rather than standalone machines include:



Advantages of networks

All PCs can be centrally managed

Security can be centrally managed

Users can work from any PC

PCs can communicate with other PCs

Back-ups can be done centrally

Resources can be shared by PCs

New software can be distributed easily

PCs can share data with other PCs

Advantages of networks.

Advantages

Benefits of a network	**Benefit to the Company and employees**
Users on a network can easily **share resources** such as printers and access to the internet.	A company doesn't have to buy so much equipment
Users on a network can **share files**	Team might be working on a project and they can all access and work on the same files easily,
New software need only be added once and **distributed to all other PCs automatically.**	Saves time and resources.
Users can retrieve and work on files from **any machine on a network.**	If one machine is being used or breaks down, you simply move to a different machine!
When data files are **backed up** they only need to be backed-up once, centrally, at the server.	If the files on the network are compromised or a user accidently deletes a file, it can be restored easiliy from the back up.
Networks can be **managed centrally**.	Network manager is able through the network software to control who can access the network, when they can access it, and what files and software and hardware they are allowed to use. Carry out an audit trail of each user.
Security can be centrally managed by the network manager.	They can add patches centrally, ensure virus patterns are up-to-date centrally and so on.

Drawbacks

- It **costs more money to build a network** than it does standalone machines.
  - This is because you have to buy network cards, interconnections, a server and a Network Operating System.
- Additional **support cost** that is not insignificant.
  - Networks are more complicated than sets of standalone machines. They need specialist knowledge to set them up and maintain them as well as time to maintain them. To do this, you would need to employ somebody with network management skills.
- Networks have machines in different locations, each of which potentially could be used to **gain access** to the server's hard disk and the data stored on it. This is a security headache and the need to protect data on a network from hackers adds an extra degree of complexity to a network that doesn't exist with standalone machines.
- If the **server on a client-server network goes down**, the whole network will be out of action. It is also possible for one faulty machine on a network to cause other machines on the network to stop working. In addition, if the cables on a network fail, problems on either individual machines or on the whole network may occur.

Remember this definition!

A protocol is a set of **rules and standards** that define how **data is transmitted** and communicated between **devices in** a network.

Common protocols

HTTP (Hypertext Transfer Protocol): used to send resources via the World Wide Web, such as web pages. It is the basis for web browsing and acts at the application layer.

HTTPS (Hypertext Transfer Protocol Secure): an HTTP extension that secures data transmission between a web browser and a web server using encryption (often TLS or SSL), protecting data confidentiality and integrity.

FTP (File Transfer Protocol): used to move files over a network between a client and a server. For safe file transfers, it includes two modes: FTP and FTPS (FTP Secure).

TCP (Transmission Control Protocol): A reliable and connection-oriented transport layer protocol responsible for ensuring that data sent from one computer to another arrives intact and in the correct order.

IP (Internet Protocol): a protocol at the network layer that provides the addressing and routing frameworks for transmitting data packets between devices via the internet or other IP-based networks.

SMTP (Simple Mail Transfer Protocol): between email clients and email servers for the purpose of sending email messages. The delivery of email messages to their intended recipients is outlined by SMTP.

POP3 (Post Office Protocol version 3): a technique for retrieving emails from mail servers that email clients can use. Typically, it is used to download messages to a local device while keeping a copy on the server.

IMAP (Internet Message Access Protocol): IMAP is a different email retrieval protocol that enables email clients to manage and access email messages kept on a distant mail server. It is made to synchronize email between many devices.

Rules

A communications protocol needs to specify a range of things before successful communication can take place.

These include:

- What baudrate will be used - It determines how fast data is sent and received.
- What error checking will be used - **Checksums** or **Parity Checking**
- Whether software or hardware 'handshaking' is to be used - Handshaking is the process of **establishing communication** between devices
- What character set is to be used. (**ASCII** or **UNICODE**)
- How many bits will be used for data.
- How many control bits will be used to control data transfer (**Start Bit** – Signals the beginning of a data packet or a **Stop Bit** – Marks the end of the data packet)

The importance of standards in communication between computers

Setting standards, rules that all manufacturers of hardware and software will follow, are important for a number of reasons:

- Standards describe **accurately** and unambiguously how information is transmitted.
- A manufacturer's products will work successfully with other manufacturer's products if they all follow the same standards.
- By defining a set of standards, you are providing a framework within which all manufacturers can design new, successful products.
- Standards break down complex ideas into smaller, methodical, easier-to-understand components.

TCP/IP - The Internet's Main Protocol

- **TCP/IP (Transmission Control Protocol / Internet Protocol)** is the most common protocol used on the Internet.
- It ensures information is transferred successfully between computers.
- Computers using different hardware or software can communicate as long as they follow TCP/IP.
- **IP (Internet Protocol)** handles addressing and routing data.
- **TCP (Transmission Control Protocol)** ensures data is delivered correctly and in order.

Other protocols rely on TCP/IP, such as:

- **SMTP (Simple Mail Transfer Protocol)** for emails.
- **HTTP (Hypertext Transfer Protocol)** for web pages.

The role of a mail server in retrieving and sending email

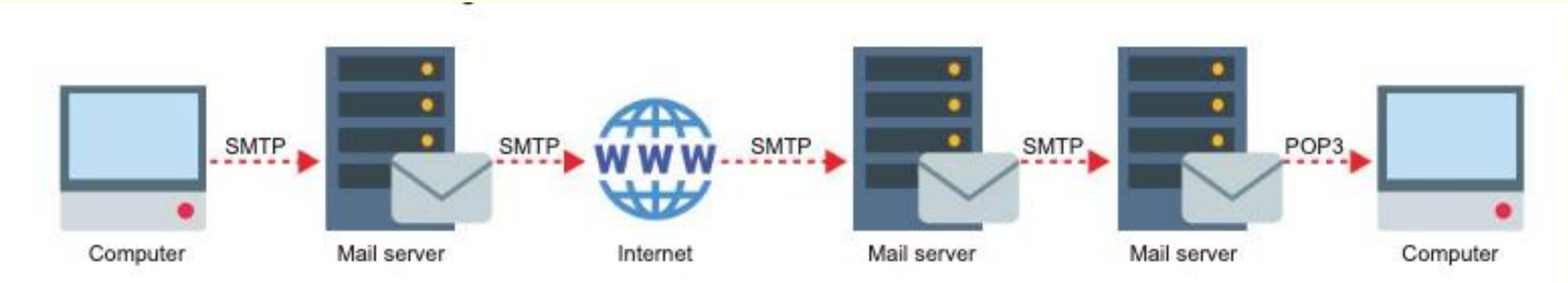
A mail server functions like a virtual post office, managing incoming and outgoing emails.

It routes messages based on its user database and stores them until retrieved.

POP3 retrieves emails from the server and deletes them after transfer, preventing synchronisation across devices.

IMAP keeps emails on the server, ensuring all devices stay synchronised.

SMTP handles the transfer of outgoing emails between servers or from an email client to the server.



TCP/IP Stack and Protocol Layers

TCP/IP (Transmission Control Protocol / Internet Protocol) is a set of networking protocols that work together to send data across networks.

It is divided into four layers:

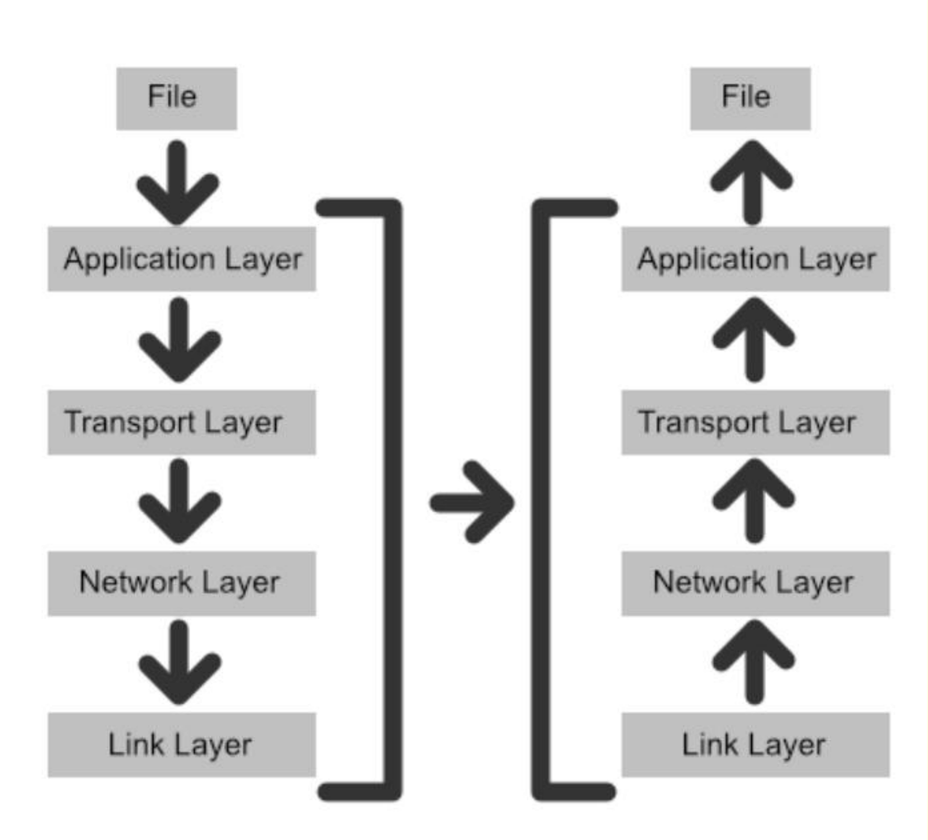
Application Layer	The top layer that determines the correct protocol based on the application (e.g., HTTP for web browsing, FTP for file transfers).
Transport Layer	• Uses **TCP** to establish a connection between sender and receiver. • Splits data into packets, labels them with numbers and port details. • Requests retransmission if packets are lost.
Network Layer	• Adds **IP addresses** of the sender and recipient to ensure correct delivery. • Routers use these IPs to forward packets. • The combination of **IP address + port number** is called a **socket address**.
Link Layer	• Handles the physical connection between devices. • Adds **MAC addresses** to identify the sender and recipient's network devices. • If the recipient is on a different network, the packet goes to the router's MAC address first.

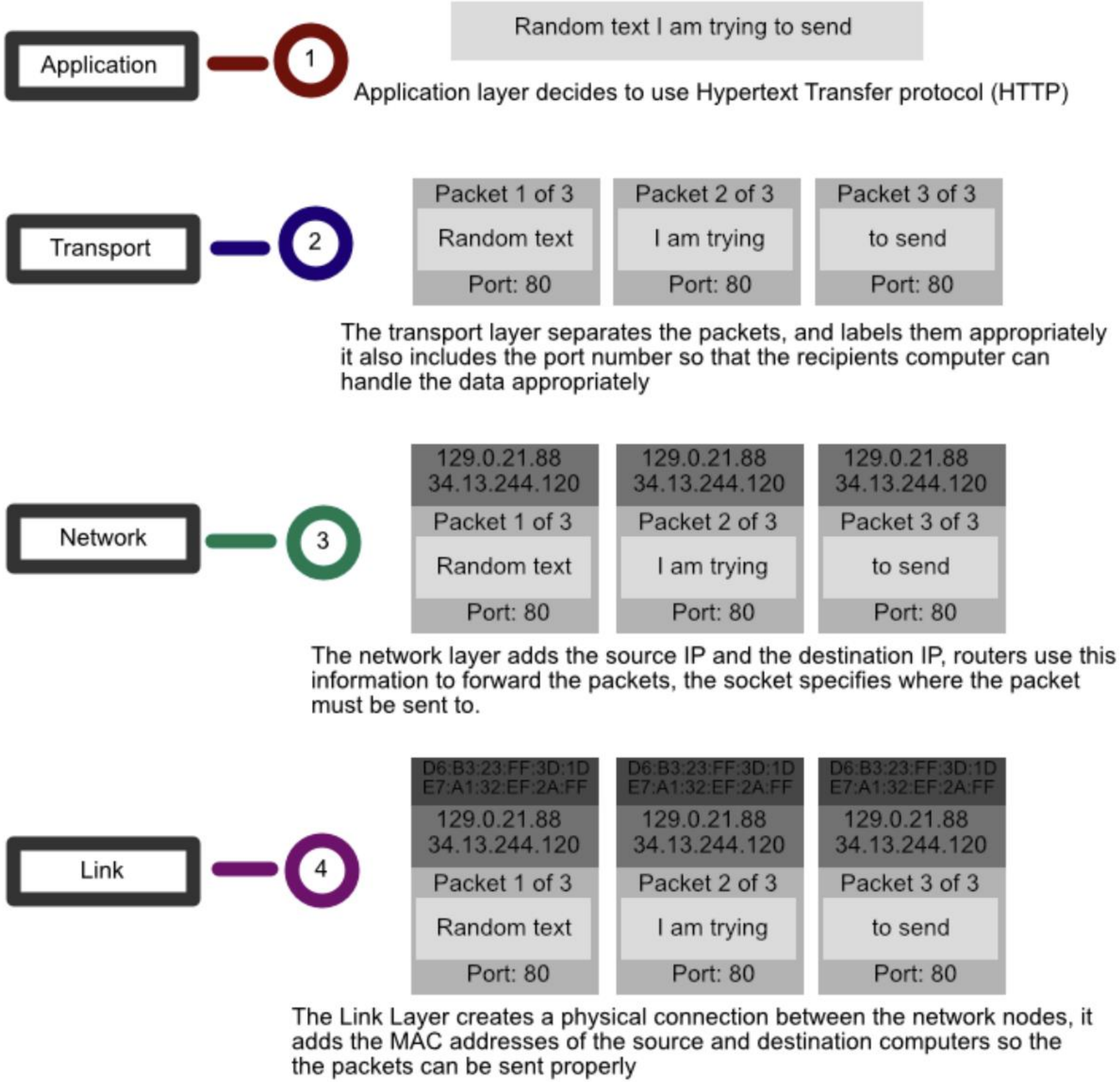
It’s important to realise that this is a stack.

On the recipient’s computer these layers are looked at from bottom to top.

Once the destination has been reached, the MAC address is removed by the link layer, then the IP addresses are removed by the Network Layer, then the transport layers remove the port number and reassemble the packets.

Finally, the application layer presents the data to the recipient in the form it was requested in.





Application

1

Random text I am trying to send

Application layer decides to use Hypertext Transfer protocol (HTTP)

Transport

2

Packet 1 of 3	Packet 2 of 3	Packet 3 of 3
Random text	I am trying	to send
Port: 80	Port: 80	Port: 80

The transport layer separates the packets, and labels them appropriately it also includes the port number so that the recipients computer can handle the data appropriately

Network

3

129.0.21.88 34.13.244.120	129.0.21.88 34.13.244.120	129.0.21.88 34.13.244.120
Packet 1 of 3	Packet 2 of 3	Packet 3 of 3
Random text	I am trying	to send
Port: 80	Port: 80	Port: 80

The network layer adds the source IP and the destination IP, routers use this information to forward the packets, the socket specifies where the packet must be sent to.

Link

4

D6:B3:23:FF:3D:1D E7:A1:32:EF:2A:FF	D6:B3:23:FF:3D:1D E7:A1:32:EF:2A:FF	D6:B3:23:FF:3D:1D E7:A1:32:EF:2A:FF
129.0.21.88 34.13.244.120	129.0.21.88 34.13.244.120	129.0.21.88 34.13.244.120
Packet 1 of 3	Packet 2 of 3	Packet 3 of 3
Random text	I am trying	to send
Port: 80	Port: 80	Port: 80

The Link Layer creates a physical connection between the network nodes, it adds the MAC addresses of the source and destination computers so the the packets can be sent properly

Benefits of Layering:

Layering means grouping **protocols and standards** into **layers** makes communication more **organised and efficient**.

- Each layer can be **designed and tested separately**.
- Each layer **builds on the one below** and provides data for the next.
- As long as **standards are followed**, different manufacturers' products will work together.
- **Problems are easier to find** because each layer is independent.
- Before the OSI model, different devices **struggled to communicate**, but layering **solves this problem**.

LANs and WANs

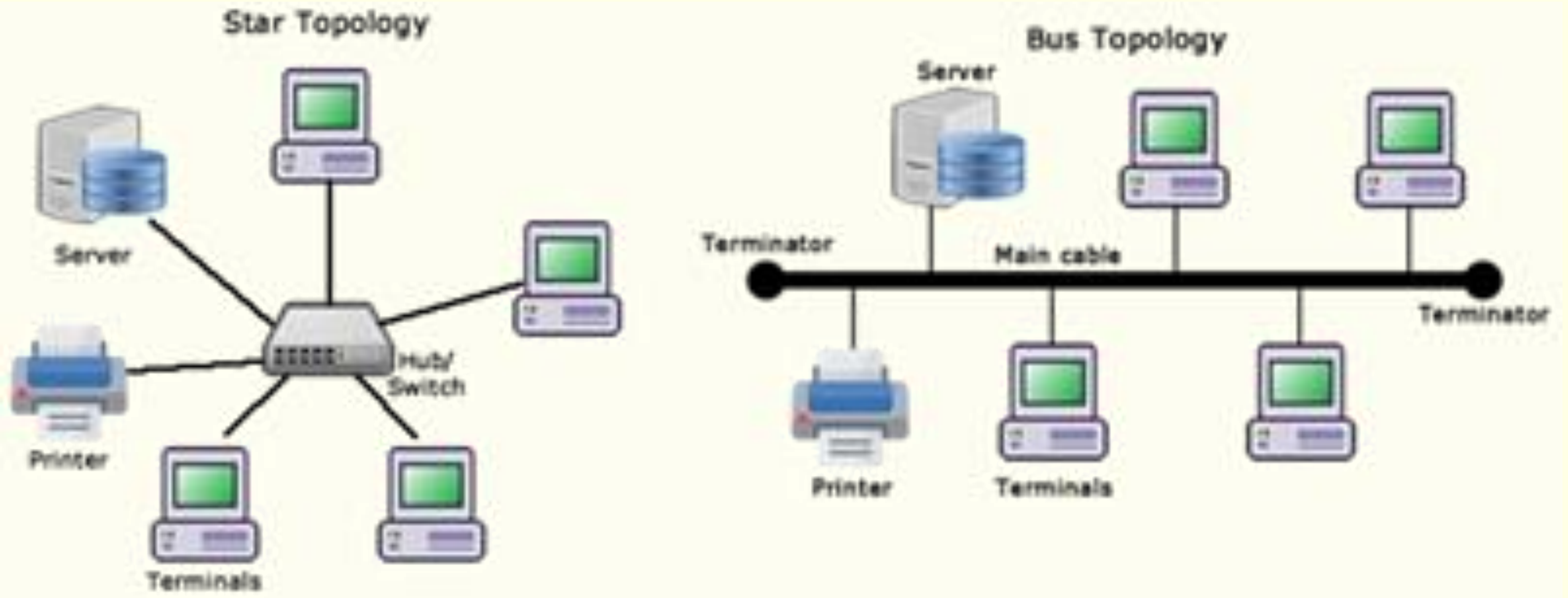
Local Area Networks are networks that are made up of computers connected up to each other and are geographically close to each other, for example, in a room, or in a building, or in a number of close buildings.

Wide Area Networks, are made up of computers connected up to each other over a wide geographical area. The Internet is a good example of a WAN. Companies who have offices scattered over the country may have a Wide Area Network.

Network topologies

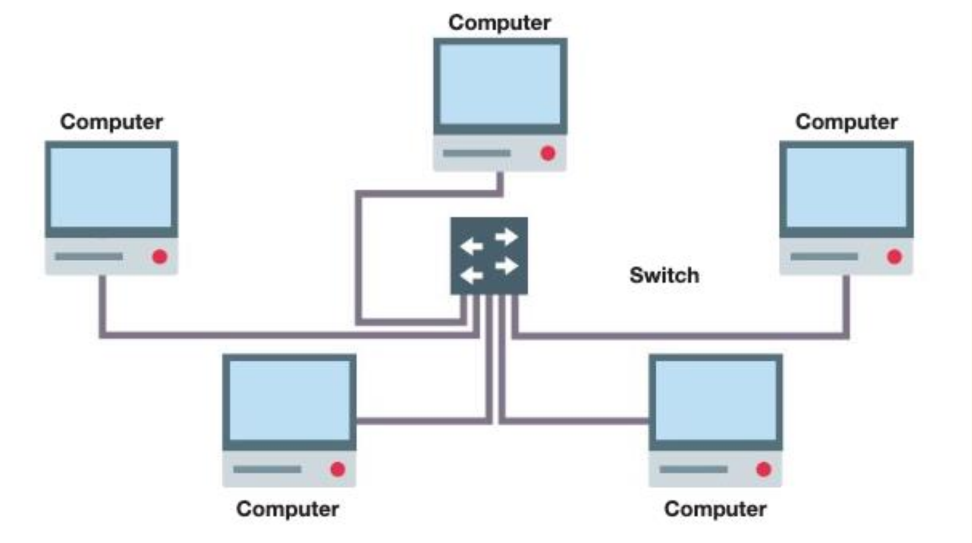
Computers that are going to be connected together can be connected in different ways.

The way that the computers on a network are connected together is known as the topology of the network.



Star network

- A star network has a central node, which may be a switch or computer which acts as a router to transmit messages.
- A switch keeps a record of the unique MAC address of each device on the network and can identify which particular computer on the network it should send the data to.



Advantages of a star topology

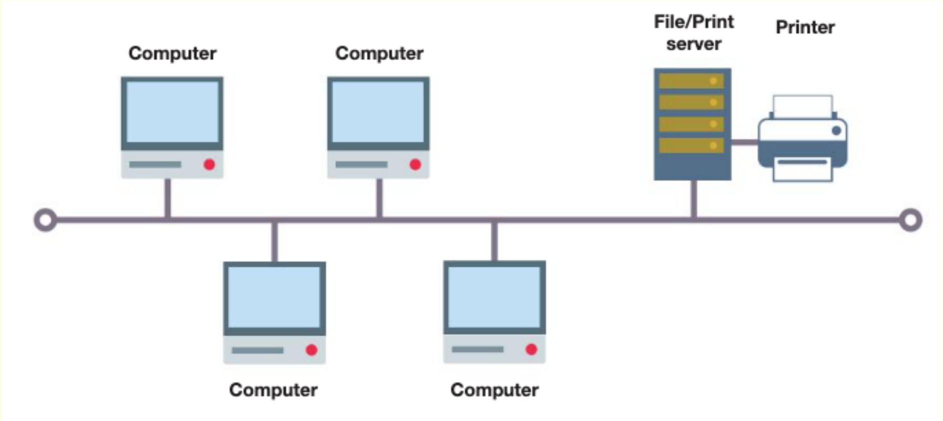
- If one cable fails, only one station is affected, so it is simple to isolate faults
- Consistent performance even when the network is being heavily used
- Higher transmission speeds can give better performance than a bus network
- No problems with 'collisions' of data since each station has its own cable to the server
- The system is more secure as messages are sent directly to the central computer and cannot be intercepted by other stations
- Easy to add new stations without disrupting the network

Disadvantages of a star network

- May be costly to install because of the length of cable required
- If the central device goes down, network data can no longer be transmitted to any of the nodes

Bus networks

- A LAN can use different layouts or topologies. In a bus topology, all computers are connected to a single cable. The ends of the cable are plugged into a terminator.



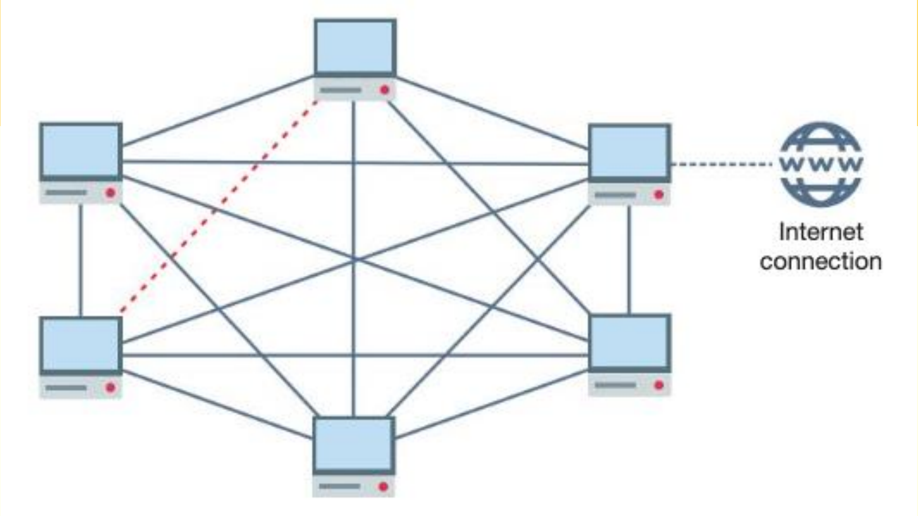
Advantage of a bus topology

- Inexpensive to install as it requires less cable than a star topology and does not require any additional hardware

Disadvantages of a bus topology

- If the main cable fails, network data can no longer be transmitted to any of the nodes
- Performance degrades with heavy traffic
- Low security – all computers on the network can see all data transmissions

Mesh Networks



Mesh networks are becoming more common with the widespread use of wireless technology.

Each node in a mesh network has a connection to every other node, by transmitting data across any intermediate nodes.

Only one node requires a connection to the Internet and all others can share this connection.

Mesh networks can quickly become big enough to cover entire cities.

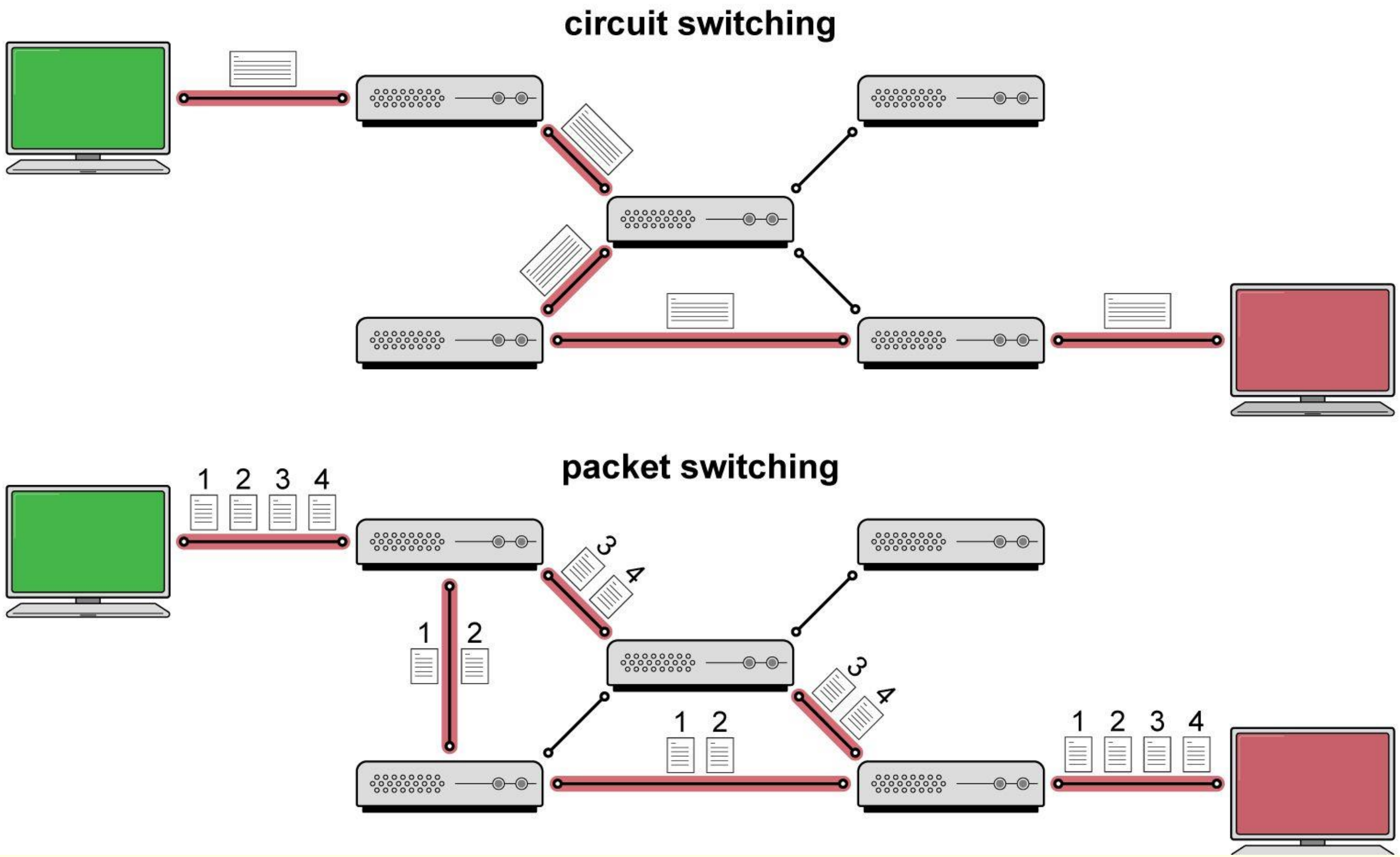
Advantages of a wireless mesh network

- The advantages of a mesh network include:
- No cabling costs
- The more nodes that are installed, the faster and more reliable the network becomes, since one blocked or broken connection (as shown above) can easily be circumvented by another route. In this respect, the mesh topology can be described as 'self healing'.
- New nodes are automatically incorporated into the network
- Faster communication since data packets do not need to travel via a central switch

Packet formation

- When a message is too large to be sent as a single unit, it's divided into smaller packets
- Each packet is typically composed of a header, payload (actual data), and a footer (or trailer)
- **Use of headers**
- Headers are important because they contain information necessary for the packet's delivery
- Typical information in a header includes:
    - Source **IP address**: identifies the sender of the packet
    - Destination IP address: identifies the intended recipient of the packet
    - Sequence Number: helps in reassembling the packets back into the original message at the receiving end
    - **Protocol**: identifies the transport protocol (TCP, UDP, etc.)
    - Packet Length: indicates the size of the packet
    - Checksum: a value used for error-checking
- **Packet transmission**
- After being packetised and **encapsulated** with headers (and trailers), packets are transmitted individually across the network
- Packets might take different routes to reach their destination
- **Packet reassembly**
- When the packets reach their destination, they are reassembled back into the original message using information in the headers

Switching networks



circuit switching

packet switching

Packet switching

In A Level Computer Science, packet switching is a **networking communication method** that **breaks down data** (large files, emails) into **smaller packets**

It sends these packets separately along different routes, and then reassembles them at their destination

Benefits	Drawbacks
Efficient use of network resources as packets can follow different paths to the destination, using more of the available bandwidth	Not ideal for real-time services like video calling or VoIP, which require a steady stream of data without delays
More reliable, as if a single packet fails to reach its destination, only that packet needs to be resent, not the entire data stream	Packets can arrive out of order, requiring reassembly and error-checking
Lower cost due to shared network resources	Potential for congestion in the network

Circuit switching

Circuit switching is a communication method where a **dedicated communication** path is established between two devices for the duration of their conversation (like a phone call), and **all packets** are sent along the **same route**

Benefits	**Drawbacks**
Ideal for real-time services, with a constant and steady data transmission rate	Less efficient, as resources remain allocated during the whole conversation, even when no data is being sent
No delays as a dedicated path is established	It is more costly due to the dedicated line requirement
Data arrives in order as it follows the same path	Less flexible and scalable, as adding new devices can be complex

Packet switching vs circuit switching comparison table

Packet Switching	Circuit Switching
Benefits	
Efficient use of network resources as packets can follow different paths to the destination, using more of the available bandwidth	Ideal for real-time services, with a constant and steady data transmission rate
More reliable, as if a single packet fails to reach its destination, only that packet needs to be resent, not the entire data stream	No delays as a dedicated path is established
	Data arrives in order as it follows the same path
Drawbacks	
Not ideal for real-time services like video calling or VoIP, which require a steady stream of data without delays	Less efficient, as resources remain allocated during the whole conversation, even when no data is being sent
Packets can arrive out of order, requiring reassembly and error-checking	More costly due to the dedicated line requirement
Network congestion can lead to packet loss	Less flexible and scalable as adding new devices can be complex

Examiner Tips and Tricks

- Avoid talking about the speed of data transmission in an answer to a question on packet or circuit switching.
- This will not get you a mark in the exam and, in some questions, is explicitly stated as not worthy of a mark.
- It is better to talk about higher bit rates or **bandwidth** (the number of bits sent per second) or the efficiency of the transmission

Data Transmission and Error Detection

A **parity bit** is an extra bit added to a group of data bits to help detect errors during transmission. It is a simple form of **error detection** used in digital communication systems.

How Parity Bits Work:

1. **Even Parity:** The parity bit is set so that the total number of 1s in the byte (including the parity bit) is even.
2. **Odd Parity:** The parity bit is set so that the total number of 1s in the byte (including the parity bit) is odd.

Example:

- Suppose we have a **7-bit data**: 1011001
- Using **even parity**, we count the number of 1s (which is 4). Since it's already even, the parity bit is **0**.
- The transmitted byte becomes: **10110010**
- If an error occurs during transmission (e.g., a bit flips), the receiver counts the 1s and checks if the parity is still even. If not, an error is detected.

Limitations:

- Parity bits can detect **single-bit errors** but cannot correct them.
- They fail to detect errors when two bits flip (even number of errors).
- Parity checking is a simple but effective way to detect errors in data transmission systems like serial communication, RAM storage, and networking protocols.

Checksum and Sending Bytes in Data Transmission

A **checksum** is an error-detection method used to ensure that data sent over a network or storage system is received correctly. It works by calculating a **sum** of all the bytes before transmission and sending this sum (checksum) along with the data. The receiver then performs the same calculation and checks if the result matches the sent checksum. If they match, the data is correct; if not, an error has occurred.

How Checksum Works with Sending Bytes

Step 1: Sender Computes the Checksum

The sender takes all the data bytes and adds them together.

Given Bytes (8-bit each):

01101101 (109 in decimal)
10000001 (129 in decimal)
10001000 (136 in decimal)

109 + 129 + 136 = 374

To calculate the checksum:
Add the bytes and ignore the carry to keep the checksum as 8 bits
(1)01110110
The sender sends:
01101101 10000001 10001000 01110110 (checksum)
The checksum value is then sent along with the data.

Step 2: Data and Checksum are Sent

- The data bytes and the checksum are transmitted to the receiver.

Step 3: Receiver Verifies the Checksum

- The same sum is done at the receiving end and the results compared

Why Use Checksum?

+ Detects errors in data transmission.
+ Simple and efficient for detecting single-bit and some multi-bit errors.
+ Cannot correct errors—only detects them.

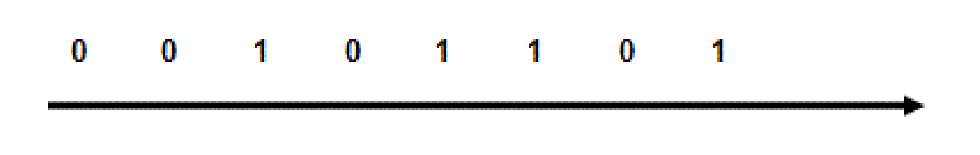
Checksum is widely used in networking (TCP/IP), file transfers, and storage systems to ensure reliable data communication.

Serial vs. Parallel Data Transmission

Serial Transmission – Sends data **one bit at a time** using **one wire**.

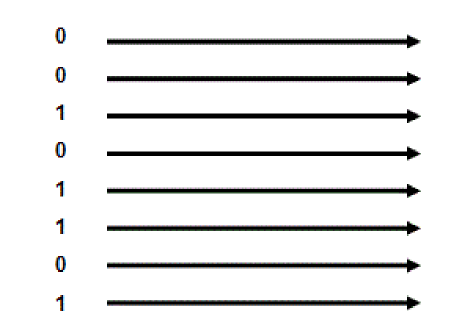
Example: Sending **10111010** as **1 → 0 → 1 → 1 → 1 → 0 → 1 → 0**

Slower but reliable for long distances.



Parallel Transmission – Sends **multiple bits at the same time** using **multiple wires**.

- **Example:** Sending **10111010** all at once through **8 wires**.
- **Faster** but can cause **errors over long distances** (over 10m).
- Used for **printers & scanners** that need fast data transfer but are close to the computer.



Simplex, Half-Duplex, and Full-Duplex Transmission

Basis for Comparison	**Simplex**	**Half Duplex**	**Full Duplex**
Direction of Communication	Unidirectional	Two-directional, one at a time	Two-directional, simultaneously
Send / Receive	The sender can only send data	The sender can send and receive data, but one a time	The sender can send and receive data simultaneously
Performance	Worst performing mode of transmission	Better than Simplex	Best performing mode of transmission
Example	Keyboard and monitor	Walkie-talkie	Telephone

Network security and threats, and using firewalls, proxies and encryption

Comparison of Viruses, Worms, and Trojan Horses

Type	**How It Spreads**	**What It Does**	**Example**	**Prevention**
Virus	Attaches to files & programs	Corrupts data, spreads via emails & USB drives	Infected email attachment	Use **antivirus**, avoid unknown attachments
Worm	Spreads through networks	Slows systems, consumes bandwidth	**MSBlaster (2003)**	Use a **firewall**, keep system updated
Trojan Horse	Hidden inside fake software	Steals data, allows hackers access	Fake security software	Download software only from **trusted sources**

Comparison of Spyware and Adware

Feature	**Spyware**	**Adware**
What It Does	Secretly collects personal data & tracks activity	Displays unwanted ads & pop-ups
How It Spreads	Bundled with free software, malicious websites	Comes with free software, often as an optional install
Effects	Slows down PC, steals data, changes settings	Annoying ads, slows browsing experience
Risk Level	**High** – Can steal sensitive information	**Low to Medium** – Mostly irritating but not dangerous
Prevention	Avoid unknown downloads, use **anti-spyware software**	Read install screens carefully, uncheck unnecessary options
Removal Tools	**Malwarebytes, Windows Defender**	**Adaware, Malwarebytes**

Phishing

- This is a term used to describe when criminals try to get hold of your credit card details or other personal information by pretending to be someone they are not over the Internet.

- They do this by sending out bogus emails e.g. pretending to be from a bank and asking you to confirm passwords for security reasons or by setting up a web site that looks like it is a legitimate business and luring you into entering personal data, perhaps by advertising very cheap prices for goods.

- Despite numerous warnings that organisations never ask for personal details by email, and reminding people that if an offer is too good to be true it probably is, people fall victim to

- Phishing attacks regularly and can suffer huge financial loses.

Cookies

- A **cookie** is a small text file saved on your computer by a website you visit. It helps the site remember your preferences, login details, or past activity.

How it Works:

1. You visit a website.
2. The site stores a cookie on your device.
3. Next time you visit, the cookie helps load personalized content.

Are Cookies Dangerous?

- **Not a threat**, but some people block them for privacy reasons.
- Websites **must ask for permission** before using cookies.

Hackers and hacking

Hackers try to get unauthorised access to your computer by 'hacking' into it

Firewalls are an excellent way of preventing hackers from getting into a network and most companies and individuals set one up on their system.

Keeping Data Secure

The **Data Protection Act 1998** requires organisations to keep data safe. Security measures include **firewalls, proxy servers, encryption, and authentication.**

Firewall 🛡️

A firewall **blocks unauthorised access** to a system. It allows only approved users to access certain data while keeping others out.

Proxy Server 🌐

A **proxy server** sits between users and the main server. It checks user permissions before granting access to data, adding an extra layer of security.

How it Works:

1. A user requests data.
2. The **firewall** checks if they have permission.
3. If valid, the **proxy server** retrieves the data and sends it back.
4. Users cannot access the main server directly.

Firewalls & User Authorisation

Many networks allow users to **remotely access files and resources.** To prevent unauthorized access, a **firewall** on a **proxy server** is used.

How it Works:

1. The user logs in with a **User ID & Password.**
2. The **firewall** checks this along with the user's **IP address.**
3. If valid, access is **granted**—but only through the **proxy server,** not directly to the network.

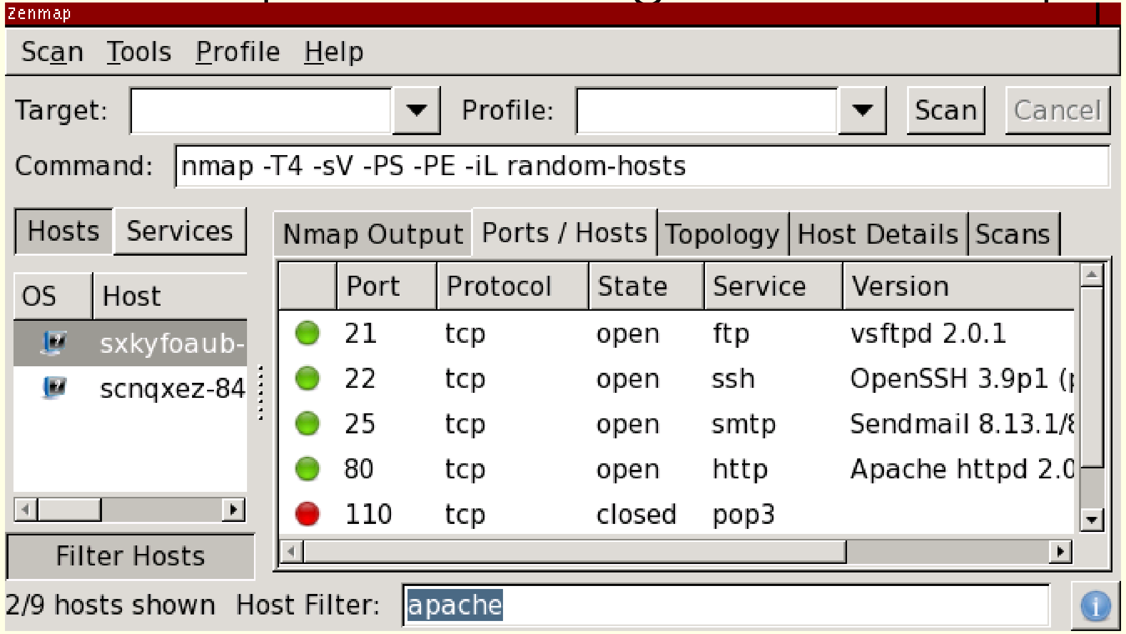
Packet filtering

Packet filtering, also referred to as static filtering, controls network access according to network administrator rules and policies by examining the source and destination IP addresses in packet headers.

If the IP addresses match those recorded on the administrator’s ‘permitted’ list, they are accepted.

Static filtering can also block packets based on the protocols being used and the port numbers they are trying to access.

Certain protocols use particular ports.



Command: nmap -T4 -sV -PS -PE -iL random-hosts

OS	Host
	sxkyfoaub-
	scnqxez-84

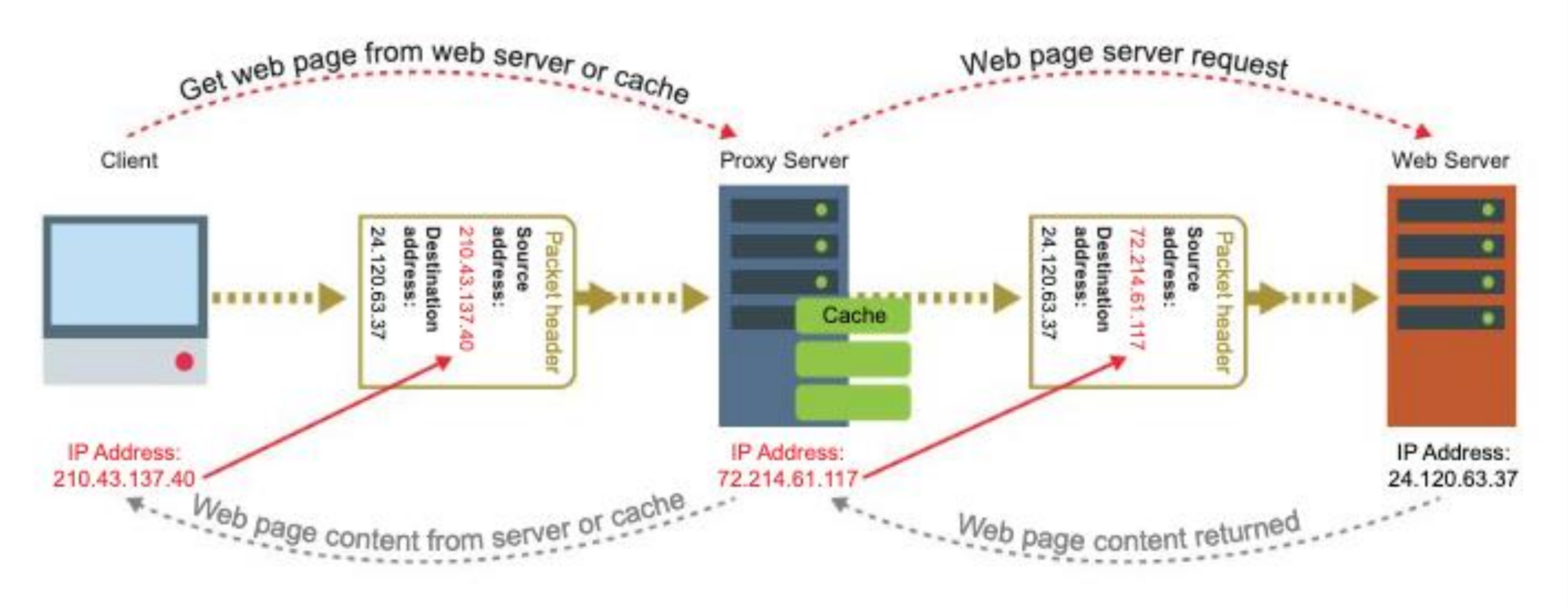
	Port	Protocol	State	Service	Version
🟢	21	tcp	open	ftp	vsftpd 2.0.1
🟢	22	tcp	open	ssh	OpenSSH 3.9p1 (
🟢	25	tcp	open	smtp	Sendmail 8.13.1/
🟢	80	tcp	open	http	Apache httpd 2.0
🔴	110	tcp	closed	pop3	

2/9 hosts shown Host Filter: apache

Proxy servers

- A proxy server intercepts all packets entering and leaving a network, hiding the true network addresses of the source from the recipient.
- This enables privacy and anonymous surfing.
- A proxy can also maintain a cache of websites commonly visited and return the web page data to the user immediately without the need to reconnect to the Internet and re-request the page from the website server.
- This speeds up user access to web page data and reduces web traffic.
- If a web page is not in the cache, then the proxy will make a request of its own on behalf of the user to the web server using its own IP address and forward the returned data to the user, adding the page to its cache for other users going through the same proxy server to access.
- A proxy server may serve hundreds, if not thousands of users.
- Proxy servers are often used to filter requests providing administrative control over the content that users may demand.
- A common example is a school web-proxy that filters undesirable or potentially unsafe online content in accordance with the school usage policies. Such proxies may also log user data withtheir requests.

Proxy servers



Encryption & Digital Security

Encryption ensures data is **secure** when sent over a network.

Digital Signatures verify the sender's identity:

1. The sender signs the email using their **Private Key.**
2. The receiver checks the signature with the sender's **Public Key.**
3. If the keys don't match, the email may be **fake or altered.**

Digital Certificates prove a website is **legitimate** for secure transactions. They are issued only after **strict security checks.**

http and https

- When you visit a website, your browser uses **HTTP** to request and load pages. However, **HTTP is not secure**, meaning hackers could **intercept** your data.

- If you're sending **personal details, passwords, or financial info**, this poses a **security risk** (e.g., identity theft or bank fraud).

- To **protect** sensitive data, websites use **HTTPS (HyperText Transfer Protocol Secure).**

Benefits of HTTPS:

- Ensures you're connecting to the **right website**
- **Encrypts** all communication, making it unreadable to hackers

Always check for "HTTPS" and a padlock 🔒 in the address bar when entering sensitive information!

Dangers of Public Wi-Fi 🚨

- **Public networks are NOT secure** – assume **someone is watching** your screen and data.
- **Hackers can steal** your information using free software that captures data packets.
- **Fake hotspots** set up by criminals can trick you into connecting, exposing all your activity.

Stay Safe:

✓ **Avoid public Wi-Fi** for sensitive tasks.
✓ **Use HTTPS** for secure browsing.
✓ **Use a VPN** to encrypt your data.

- ✓ **Never enter passwords or financial info** on public networks.

Virtual Private Networks (VPNs)

Public Wi-Fi is risky, so **encrypt your communications** for safety.

Best solution: Use a VPN (Virtual Private Network).
✓ Encrypts all your internet activity.
✓ Hides your IP address (useful for accessing geo-restricted content).
✓ Costs just a few pounds per month.

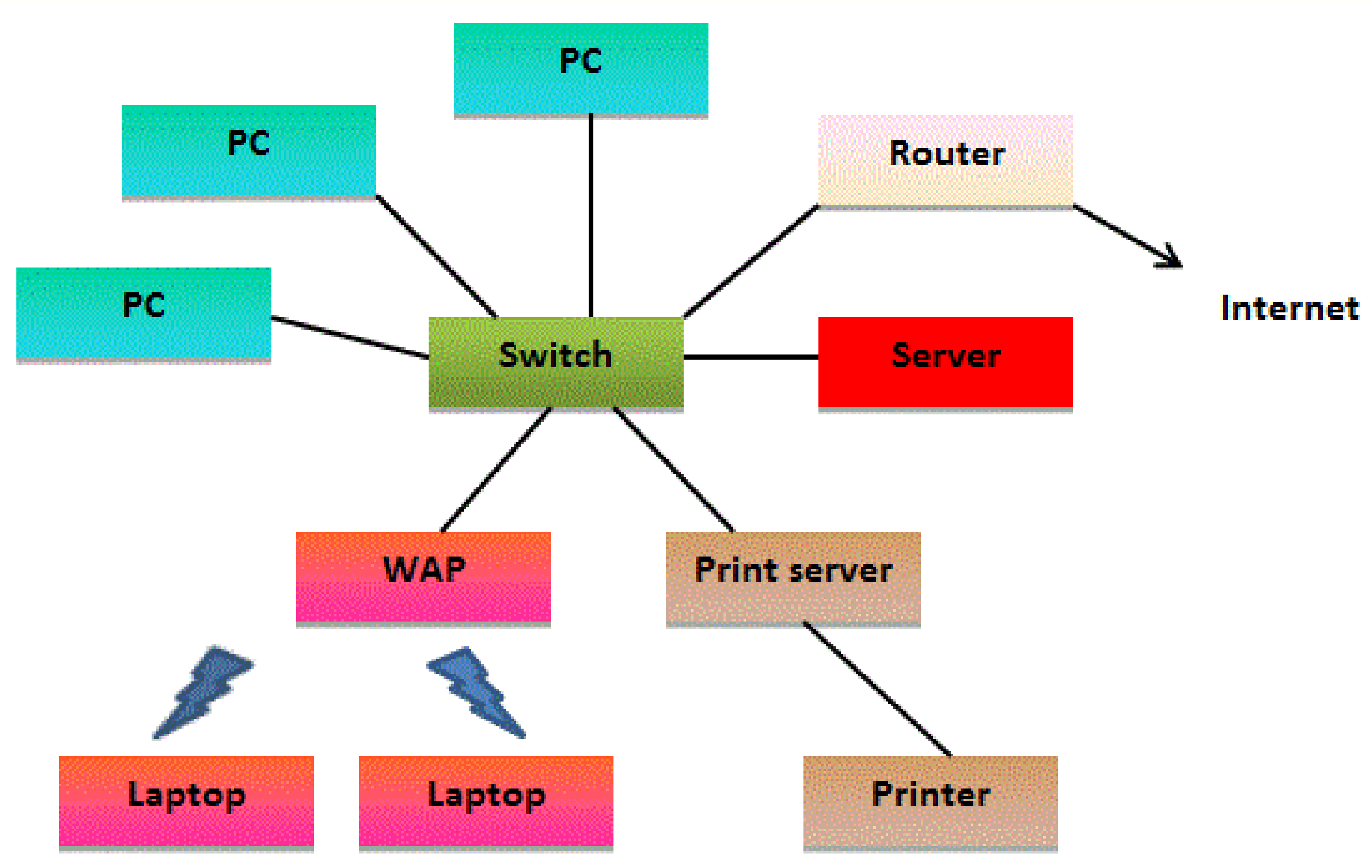
How it works:

1. Install VPN software.
2. Connect to a public network.
3. VPN verifies your identity and encrypts data.

No system is 100% secure!
✓ Check VPN terms—some log data.
✓ Employees at VPN companies could still pose risks.

Network hardware

What typical hardware is used on a network? If you had a group of stand-alone PCs, what hardware and software would you need to connect them all together to make a network?



Network Hardware

Physical Connections

- Computers in a network are linked using cables.
- **Cat 5 Cable:** Cheap, flexible, reduces interference, carries high bandwidth.
- **Fibre Optic Cable:** More expensive, immune to electrical interference, uses light signals.
- **Coaxial Cable:** Strong, protects against interference, also used for TV signals.

Switches & Hubs

- **Switch:** Manages multiple connections, directs data efficiently.
- **Hub (Older tech):** Slower, replaced by switches.

Wireless Connections

- **WAP (Wireless Access Point):** Allows wireless devices to connect to a network.
- **Hotspots:** Can be open (public) or require login, but pose security risks.
- **Always assume public networks are unsafe for sensitive data.**

Network Adapters (Network Cards)

- Required for a computer to connect to a network.
- Functions:
- Provides a physical connection.
- Splits & transmits data packets.
- Uses a unique MAC address for identification.

Servers

- **File Server:** Stores data & applications for multiple users.
- **Print Server:** Manages print jobs from multiple computers.

Routers & Modems

- **Router:** Connects a local network (LAN) to the Internet.
- **Modem:** Converts digital signals to analog (for telephone lines) and vice versa.

Network Operating System (NOS)

- Manages network communications & user access.
- Requires a login with a user ID & password.

ISP Account

- Needed to access the Internet.
- ISP provides the connection via dial-up or broadband.

Bandwidth

Definition: Amount of data a network can transmit.

- **High bandwidth activities:** Streaming video/music.
- **Low bandwidth activities:** Sending text files.

Not about speed, but the volume of data transmitted.

Client-server

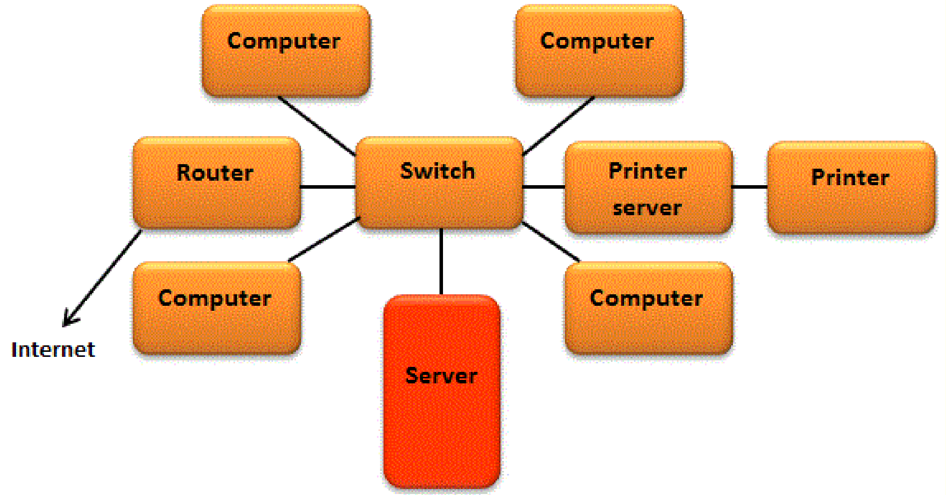
One computer acts as a **server** (file server).

Server:

- Manages **network security**.
- Stores **user login details** and **access rights**.
- Controls access to **software, files, and hardware** (e.g., printers).
- Has **large storage (hard disk)** and **high RAM** for multiple tasks.

Clients:

- Are the **computers used by authorised users**.
- Users **log in** to access files and applications.
- Users can usually log in from **any client** on the network.
- Access rights determine what a user can do (e.g., teachers vs. students).



Peer-to-peer network

- No **server**—all computers have **equal status** (peers).
- Each computer can:
- **Share files** with others.
- **Send print jobs** to another computer with a printer.

- **Pros**

✓ **Simple to set up**—best for small networks (10 computers or fewer).

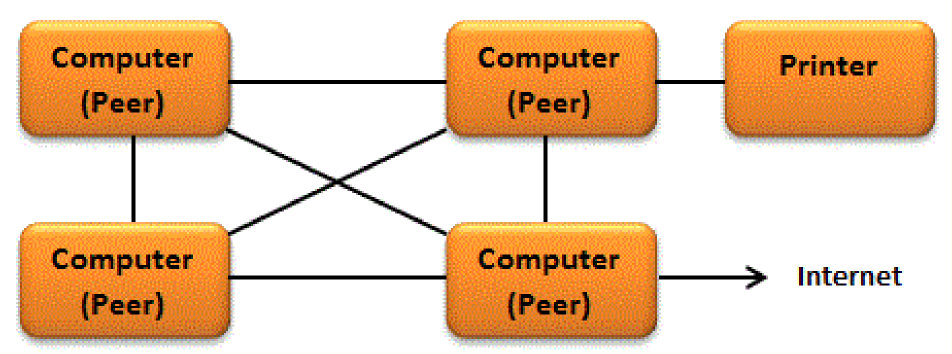
✓ **No need for a dedicated server.**

Cons

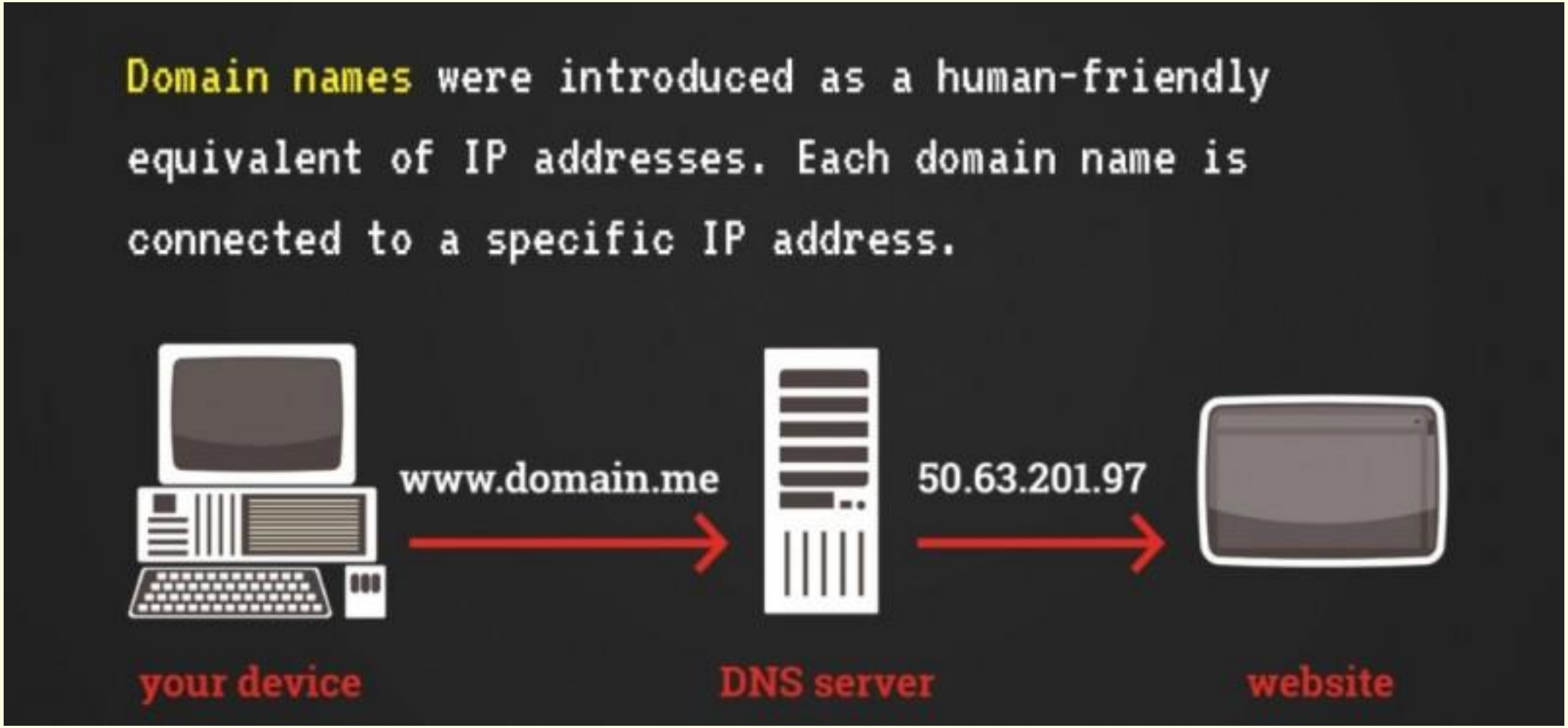
✕ **Less security**—each computer must manage its own security.

✕ **Difficult backups**—files must be backed up **individually** on each computer.

✕ **Lacks advanced features** found in client-server networks.



IP addresses and domain names



Domain names were introduced as a human-friendly equivalent of IP addresses. Each domain name is connected to a specific IP address.

Domain Name Servers (DNS)

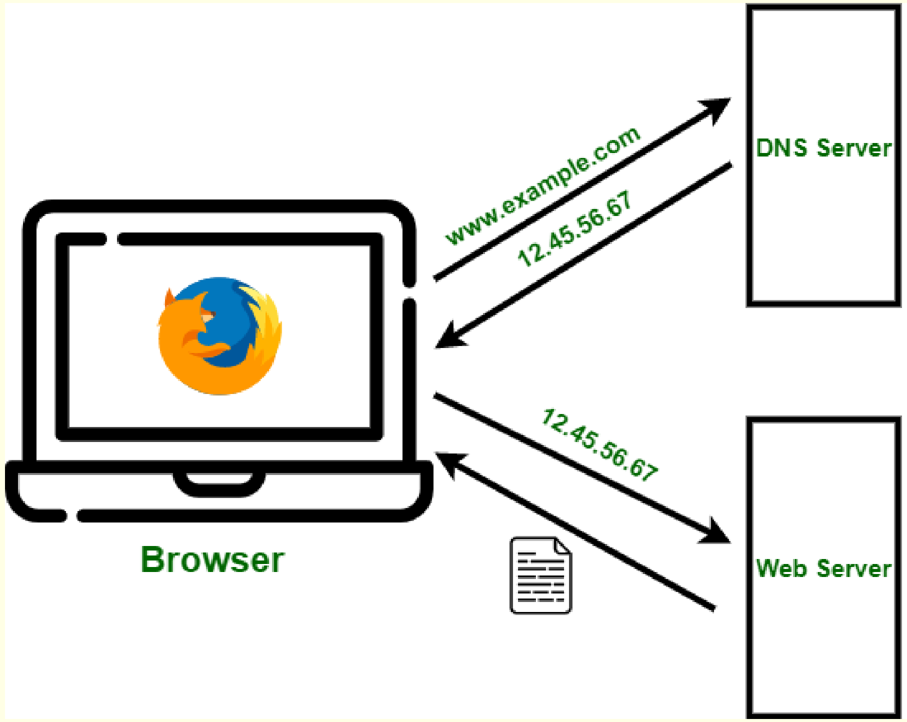
1. When you want to go to the web page for the BBC, you type the domain name into the URL search area on your web browser

2. The domain name is intercepted by a 'Domain Name Server' or DNS that your Internet Service Provider has.

3. The job of the domain name server is to look up in its database the domain name you have typed into your web browser and find the matching IP address.

4. It then forwards the request onwards, using this address. If it cannot find the IP address in its own database, it then contacts other Domain Name Servers until it finds it, or if it can't find it anywhere, it displays a 'web site not found' message to you.

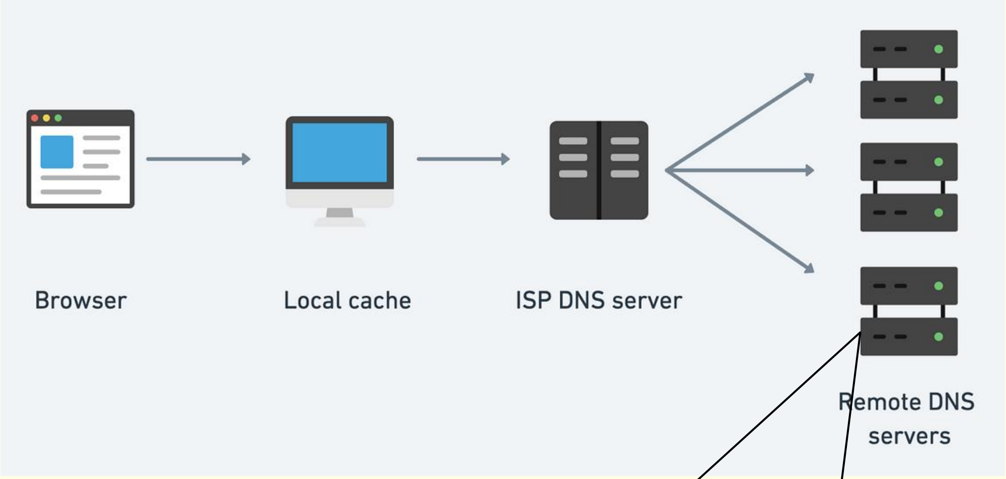
5. When the request reaches the destination, the pages are sent back.



More Advanced understanding for A-level

Step	Action
1. User enters URL	The browser sends a request to resolve the domain name to an IP address.
2. DNS Resolver Checks Cache	If the IP is found in the local cache, it's returned immediately. Otherwise, the request continues.
3. Query to TLD Name Server	If the resolver lacks the IP, it queries the TLD Name Server (e.g., .com, .org).
4. Query to Authoritative Name Server	If necessary, the request is forwarded to the Authoritative Name Server holding the actual IP.
5. IP Address Returned	The authoritative server responds with the correct IP address.
6. Browser Connects to Website	The browser uses the IP to establish a connection and load the web page.
7. Error Handling	If no match is found, an error is returned (e.g., "Site Not Found").

TLD Name Server (e.g., .com, .org).



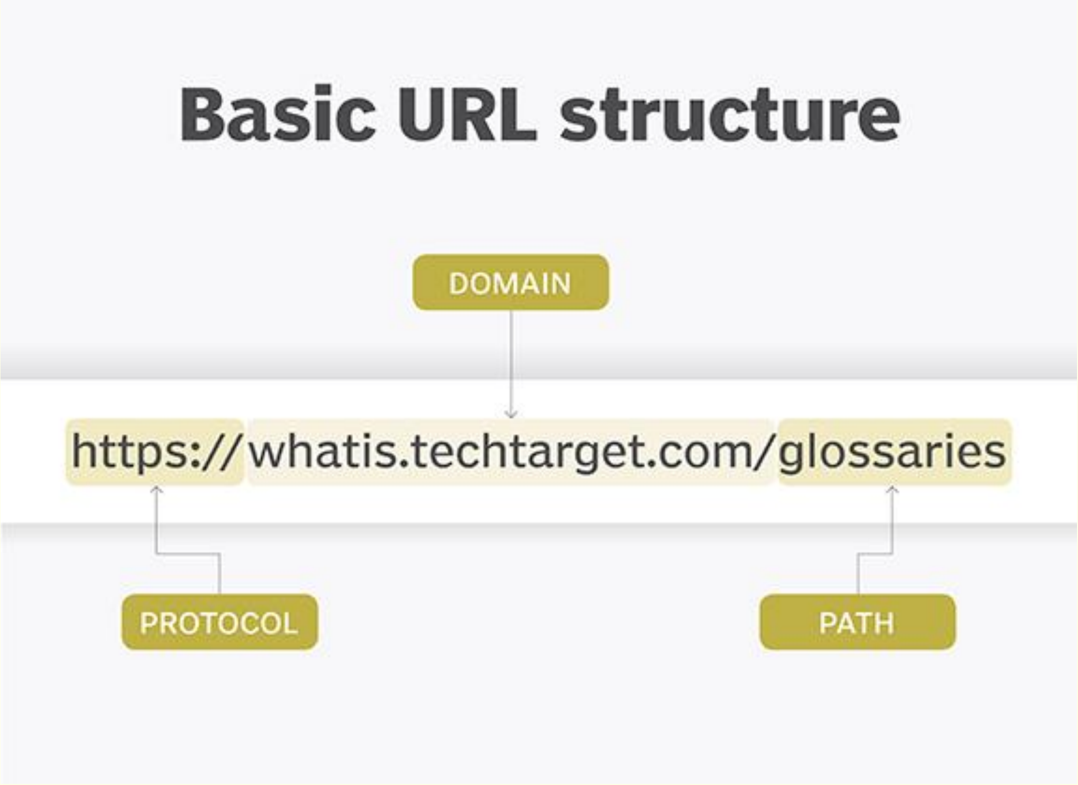
Authoritative Name Server holding the actual IP.

Uniform Resource Locator (URL)

The 'URL' is the fancy name for a full website address.

Three pieces of information are contained in a URL:

1. Protocol
2. Domain name
3. File to display.


Basic URL structure
DOMAIN
https://whatis.techtarget.com/glossaries
PROTOCOL
PATH