Neso Academy Computer Networks: Study Notes

Free study notes for the free Computer Networks course from Neso Academy, one section per topic (chapters 1 to 3, the free playlist). Written by HoverNotes in our own words as a companion to the videos: watch the topic's videos, solve the problems yourself, then use these to review and test yourself. Not affiliated with Neso Academy.

Course guide, study plan and reviews: https://hovernotes.io/courses/neso-academy-computer-networks

Made with HoverNotes, the notes panel that sits beside any video. Version 2026-11-02.

---

1. What a network is

Chapter: 1. Introduction to Computer Networks · **Videos:** Introduction to Computer Networks (10 min) · Basic Characteristics (9 min)

Key ideas

- A **computer network** is a group of **nodes** joined by **communication links**. A node is any device that can send data, receive data, or both: PCs, servers, printers, IP cameras, and also switches and routers.
- A **link** only carries the signal. It is either **wired** (copper or fibre cable) or **wireless** (radio through the air, as between two phones or a laptop and a Wi-Fi router).
- **End devices** are where a communication starts or finishes (your laptop, the web server). **Intermediary devices** pass data along in between (switches, wireless routers, routers, the many boxes inside "the Internet" cloud).
- The main reason to build a network is **resource sharing**: four PCs can share one printer, one file server, one Internet connection.
- The course opens by saying who it is for: undergraduates, GATE aspirants, people preparing for networking interviews, and as groundwork before CCNA. It uses Cisco Packet Tracer for the practical side.
- **Four characteristics every network should have:**
  - **Fault tolerance:** keeps working when a link or device fails, by sending traffic another way (like taking a detour when your usual road home is blocked). Needs redundant paths.
  - **Scalability:** can grow, more users and devices, without performance falling apart. The Internet is the standard example.
  - **Quality of service (QoS):** decides what goes first when traffic competes. A router holding a VoIP packet and an email packet sends the voice first, because a one-second delay ruins a call and doesn't matter for an email.
  - **Security:** blocks forgery, misuse and access by people who shouldn't have it, and provides **confidentiality** (only the sender and receiver can read it), **integrity** (what arrives is what was sent) and **availability** (the service stays reachable).

Remember for the exam

- "Node" is broader than "computer": anything that sends or receives data counts, including routers and switches.
- End device vs intermediary device is about the device's **role in this communication**, not its brand. A PC is an end device; a router forwarding your packets is an intermediary.
- QoS exists because **real-time traffic** (voice, video calls) can't tolerate delay or loss, while email and file transfers can wait.
- The security goals are the CIA triad: **confidentiality, integrity, availability**. Defacing or flooding a web server attacks availability.

Check yourself

1. Is a network printer a node? *(Yes: it receives data, and often sends status back.)*
2. In a home with a laptop, a Wi-Fi router, a modem and a Netflix server, which are end devices? *(The laptop and the Netflix server; the router and modem pass data along.)*
3. A router drops a link and traffic still reaches the server through another router. Which characteristic is that? *(Fault tolerance.)*
4. Why does a router send a VoIP packet before an email packet that arrived at the same time? *(QoS: voice is real-time and can't wait; email can.)*

Do next

Draw a small network you use every day (home or college lab), label each device as end or intermediary, and mark the wired and wireless links. Then watch the two videos on protocols, which explain the rules those devices follow to talk.

---

2. Protocols and communication

Chapter: 1. Introduction to Computer Networks · **Videos:** Network protocols and communications, part 1 (12 min) · Network protocols and communications, part 2 (11 min)

Key ideas

- **Data communication** means two nodes passing data to each other over a **transmission medium** (a link). Every communication has a **sender**, a **receiver** and a **channel**.
- There are three **data flow** modes:
  - **Simplex:** one direction only. A keyboard sends to the computer and never receives.
  - **Half duplex:** both directions, but one at a time. A walkie-talkie works this way.
  - **Full duplex:** both directions at the same moment. A phone call works this way.
- A **protocol** is the rulebook both sides follow. It fixes **what** is communicated, **how** it is communicated and **when** it is communicated. Without shared rules, a fast sender swamps a slow receiver, or the two sides use formats the other cannot read.
- A protocol covers five elements:
  - **Message encoding:** turning data into **signals** (electrical or light) for a wired medium, or into **waves** for a wireless one. The receiver decodes it back.
  - **Message formatting and encapsulation:** both sides agree on a layout, and the sender wraps the data with control information such as source and destination addresses.
  - **Message size:** a big message is cut into smaller pieces that the link can carry. Each piece is numbered so the receiver can reassemble them in order and spot a missing one.
  - **Message timing: flow control** (the receiver tells the sender how much it can take), **response timeout** (how long the sender waits for an acknowledgement before sending again) and access method.
  - **Delivery options: unicast** (one sender, one receiver), **multicast** (one sender, a chosen group) and **broadcast** (one sender, every node on the network).
- **Peer-to-peer:** all nodes have equal standing and no machine is in charge, so it suits only small setups and is hard to scale or administer.
- In a **client-server** network a server holds the resources and clients send **requests**; the server sends **responses**. It scales well and is administered in one place, but the server is a single point of load and failure.

Remember for the exam

- Half duplex and full duplex both allow two-way traffic. The difference is **simultaneous or not**.
- Broadcast reaches **all** nodes, multicast reaches **only the subscribed group**. A common slip is to call any one-to-many send a broadcast.
- A protocol is **rules**, not software or hardware. Software and devices implement the rules.
- Encoding depends on the medium: signals on cable, waves in the air.
- Peer-to-peer: no dedicated server. Client-server: dedicated server and the request-response model.

Check yourself

1. A radio station transmits and listeners only tune in. Which data flow mode? *(Simplex.)*
2. A sender must send a 9,000-byte file over a link that carries 1,500 bytes at a time. Which protocol element handles it, and how many pieces result at least? *(Message size, with numbered pieces; at least 6 pieces.)*
3. A sender is faster than the receiver. Which element prevents data loss? *(Message timing, specifically flow control.)*
4. A server sends a software update to five of the 40 computers in a lab, and only to them. Unicast, multicast or broadcast? *(Multicast.)*

Do next

Write down one everyday conversation (a phone call, a WhatsApp chat) and name its sender, receiver, medium and the rules both sides follow. Next, look at the kinds of devices, media and network sizes that carry this communication.

---

3. Components and types of networks

Chapter: 1. Introduction to Computer Networks · **Videos:** Components of a computer network (10 min) · Classification of computer networks (9 min)

Key ideas

- Any network is built from **nodes** (the devices), **media** (the links between them) and **services** (what people use it for).
- **End nodes** start or finish a communication: computers, network printers, VoIP phones, security cameras, smartphones, tablets, barcode scanners. **Intermediary nodes** only pass data along: switches, routers, wireless access points, firewalls, hubs, repeaters, bridges.
- **Media** are the links. **Wired (guided)** media have a cable that guides the signal. **Wireless (unguided)** media use no cable and carry waves through air or space.
- Wired examples:
  - **Ethernet (twisted pair):** electrical signals. A **straight-through** cable joins unlike devices (PC to switch); a **crossover** cable joins like devices (switch to switch, PC to PC).
  - **Fibre optic:** light pulses; high speed and long distance.
  - **Coaxial:** electrical signals; used for cable TV and set-top boxes.
  - **USB:** short connection between a computer and a phone or peripheral.
- Wireless examples, from short range to long: **infrared** (TV remote), **radio waves** (Bluetooth, Wi-Fi), **microwaves** (cellular networks), **satellite** (GPS, long distance).
- **Services** are what a network is used for: email, file sharing, instant messaging, voice over IP, video conferencing, online games, cloud storage and the **World Wide Web**.
- A **LAN** (local area network) covers a limited area: a home, lab, office or campus. Wired LANs use **Ethernet** with hubs or switches; wireless LANs use **Wi-Fi**.
- A **MAN** (metropolitan area network) joins several LANs inside one city, using routers or bridges. For example, a retail chain linking its branches in one city.
- A **WAN** (wide area network) spans a large geography such as a country or continent, joining LANs and MANs through routers. The **Internet** is a network of networks, the biggest WAN.
- Trends: **bring your own device (BYOD)**, **online collaboration** and **cloud computing**, which is on-demand access to computing resources, especially storage, without managing them yourself.

Remember for the exam

- Classify by **geographic reach**: LAN < MAN < WAN. Ownership and technology are secondary clues.
- Straight-through for **different** device types, crossover for **same** type. Modern devices often detect this automatically, but exam questions still use the rule.
- A switch or hub builds a LAN; a router or bridge is needed to join LANs into a MAN or WAN.
- Fibre uses light and gives the highest speed of the wired media. Coaxial and twisted pair carry electrical signals.
- Wired is "guided" and wireless is "unguided".

Check yourself

1. A wireless access point: end or intermediary? *(Intermediary: it forwards traffic.)*
2. Which cable joins two switches in a classroom of one building? *(Crossover, since both devices are the same type.)*
3. A bank links its Mumbai branches into one network. LAN, MAN or WAN? *(MAN, as it is within one city.)*
4. Which medium type does a Bluetooth headset use? *(Wireless, radio waves.)*

Do next

List every device in your hostel or home network, mark each as end or intermediary, and say whether each link is guided or unguided. The next topic looks at the ways nodes can be arranged: bus, ring, star and mesh.

---

4. Network topology

Chapter: 1. Introduction to Computer Networks · **Videos:** Network topology (14 min) · Network topology, solved questions (12 min)

Key ideas

- **Topology** is the layout of a network. **Physical topology** is where devices and cables are placed. **Logical topology** is how data actually flows between them.
- **Bus:** all nodes attach to one shared cable (the backbone). A frame sent by one node reaches every node; only the addressed node keeps it.
  - Pros: cheap, easy for a temporary setup, a failed node does not stop the others.
  - Cons: the single cable is a single point of failure, cable length is limited, everyone hears everyone (poor privacy), and heavy load slows all.
- **Ring:** nodes form a closed loop; each connects to two neighbours. Traffic goes in one direction, and a **token** passed around decides who may send (token ring).
  - Pros: orderly access, equal rights for all nodes.
  - Cons: a node or link failure can break the whole ring, a packet may pass many nodes before it arrives, and adding nodes disturbs the ring.
- **Star:** every node connects to a central **hub** or **switch**. Traffic goes through the centre.
  - Pros: easy to build and manage, a failed cable affects only one node, easy to expand by adding another switch.
  - Cons: the central device is a single point of failure and a possible bottleneck, and it adds cost.
- **Mesh:** every node has a dedicated link to every other node. It is highly fault tolerant and has no shared-medium traffic, but the number of cables and ports grows fast, so it suits only small networks (or the core of a backbone).
- **Hybrid:** a combination of topologies, for example several stars linked by a bus.

Formulas and methods

For n devices, counting cables and ports:

Topology	Cables (links)	Ports per device	Total ports
Ring ($n \geq 3$)	n	2	2n
Star (n end devices + 1 switch)	n	1 on each end device	2n (n on devices + n on the switch)
Mesh	$n(n-1)/2$	$n-1$	$n(n-1)$
Bus	1 backbone + n drop cables	1	n

Method: draw the topology, count links, then count the port at each end of every link.

Worked example: 8 computers in a full mesh.

- Cables = $8 \times 7 / 2 = 28$.

- Ports per computer = 7.
- Total ports = 8 × 7 = 56, which is 2 × 28 because each cable ends in two ports.

Another: 12 PCs on one switch (star) need 12 cables, 12 PC ports and 12 switch ports, so 24 ports in all. A ring of 9 nodes needs 9 cables and 18 ports.

Remember for the exam

- A switch port counts too: the star total is **2n**, not n.
- Mesh links: **n(n – 1)/2**. Mesh ports in total: **n(n – 1)**. Do not mix them up.
- "Minimum traffic problem" and "most fault tolerant" both point to **mesh**. "Single point of failure" fits bus (the cable) and star (the centre).
- Ring is **unidirectional** and uses a token; bus is **bidirectional** and shared.
- Physical topology is layout; logical topology is data flow. A physical star can behave as a logical bus (a hub).

Check yourself

1. How many cables does a full mesh of 10 nodes need? *(10 × 9 / 2 = 45.)*
2. How many ports in total for a star with 20 computers and one switch? *(40.)*
3. In a bus network, one node fails. Does the network stop? *(No, the other nodes continue; a break in the backbone would stop it.)*
4. Which topology adds a new node most easily without touching existing connections? *(Star: connect the new node to the switch.)*

Do next

Work out the cable and port counts for 5, 6 and 7 nodes in each topology, and check that your table matches the formulas above. Then practise GATE previous-year questions on topologies. The next topic covers the three addresses used to find a device and process.

---

5. IP, MAC and port addresses

Chapter: 1. Introduction to Computer Networks · **Videos:** Basics of IP addressing (8 min) · Basics of MAC addressing (7 min) · Basics of port addressing (8 min)

Key ideas

- Three addresses work together to deliver data: **IP** finds the right **network**, **MAC** finds the right **device** on that network, and the **port** finds the right **process** on that device. A courier analogy: city, building, flat.
- An **IP address** (Internet Protocol address) is a **logical address**. It depends on where the device is connected and can change, manually or automatically. Move a laptop to another LAN and it gets an address from that LAN's range.
- An **IPv4 address** has **32 bits**, written as four decimal **octets** separated by dots, for example `192.168.1.10`. Every octet has a value from 0 to 255. IPv6 addresses are 128 bits.
- A **MAC address** (media access control address) is a **physical (hardware) address** of the network interface, assigned by the manufacturer and meant to be unique worldwide. It is **48 bits**, written as 12 hexadecimal digits with separators such as `-`, `:` or `.`.
- Who uses which: **routers** forward on IP addresses; **switches** forward on MAC addresses, using a MAC address table.
- A **port number** identifies a process (an application using the network). It is **16 bits**, so ports run from **0 to 65535**. Some are fixed for well-known services (80 for HTTP, 25 for SMTP). Others are **dynamic**: the operating system hands one to a browser when it opens a connection.
- Every packet carries a **source and destination port**, so the reply returns to the process that asked, not just to the machine.
- On Windows, `ipconfig` shows the IP address, subnet mask and default gateway; `ipconfig /all` also shows the MAC address (called "Physical Address"). The Resource Monitor shows which ports processes use.
- Before sending, a node attaches source and destination **IP**, **MAC** and **port** addresses. Each is read by a different part of the path: routers, switches and the operating system.

Formulas and methods

- IPv4 space: 2^32 = 4,294,967,296 addresses. MAC space: 2^48 addresses. Port space: 2^16 = 65,536 ports.
- Check an IPv4 address in three steps: exactly four octets, only decimal digits, each value 0 to 255.

Examples with our own addresses:

Address	Valid?	Why
`10.20.30.40`	Valid	four octets, all in range
`172.16.300.1`	Invalid	300 is above 255
`192.168.5`	Invalid	only three octets
`8.8.8.A`	Invalid	A is not a decimal digit

Address	Valid?	Why
`0.0.0.0` and `255.255.255.255`	Valid format	the lowest and highest 32-bit values

Remember for the exam

- IP is **logical** and changeable; MAC is **physical** and burned in. But a MAC can be changed in software, and a device with several interfaces has several MACs.
- IPv4 is 32 bits in decimal, MAC is 48 bits in hexadecimal, port is 16 bits.
- The IP header carries addresses end to end, while the MAC addresses are rewritten at each hop (covered in topic 8).
- Router uses IP, switch uses MAC, operating system uses the port.
- An octet above 255, more or fewer than four octets, or a letter in the address makes an IPv4 address invalid.

Check yourself

1. How many hexadecimal digits are in a MAC address? *(12, which is 48 bits.)*
2. Is `203.0.113.256` a valid IPv4 address? *(No, 256 exceeds 255.)*
3. A laptop has three browser tabs open to the same site. What lets each reply reach the right tab's connection? *(Different source port numbers.)*
4. What is the largest port number? *(65535.)*

Do next

Run `ipconfig /all` (or `ip addr` on Linux) and note your IPv4 address, MAC address and default gateway. Then practise GATE previous-year questions on IPv4 address validity. The next topic covers how data finds its way across a large network.

---

6. Switching techniques

Chapter: 1. Introduction to Computer Networks · **Videos:** Switching techniques in computer networks (13 min)

Key ideas

- **Switching** is how a large network moves data from source to destination when many paths exist. The result should feel like a direct connection between the two ends.
- The video compares **circuit**, **message** and **packet** switching, and splits packet switching into the **datagram** and **virtual circuit** approaches.
- **Circuit switching** reserves a **dedicated path** before any data moves. The telephone network is the standard example. It has three phases:
  - connection **setup**,
  - **data transfer** over the reserved path,
  - connection **teardown**. The path stays reserved even during silence, which wastes capacity, but delay during transfer is steady.
- **Message switching** sends the whole message as one unit. Each intermediate node **stores** the complete message and then **forwards** it (**store and forward**). It needs large buffers and adds long delays, so it is unsuitable for real-time use such as voice calls.
- **Packet switching** cuts the message into small **packets**, each with a header and a **sequence number**. Packets share links with other users' packets, so capacity is used well. The receiver reorders packets and, using acknowledgements and timeouts, asks for lost ones again.
- **Datagram approach (connectionless):** every packet carries the full destination address and each router chooses a route for it independently. Packets of one message can take different paths and arrive out of order. The Internet's IP works this way.
- **Virtual circuit approach (connection-oriented):** a route is planned first with a setup request and accept, and packets carry a short **virtual circuit identifier**. All packets of that connection follow the same path. Unlike circuit switching, the bandwidth is not reserved exclusively; it is only a logical path.

Formulas and methods

Ignoring propagation and queueing delay, with k links in the path (k – 1 routers), message length L, packet length p and link rate R:

- Message switching: total time = k × (L / R).
- Packet switching: total time = (k + n – 1) × (p / R), where n = L / p packets. The first packet crosses k links, and the remaining n – 1 packets follow one packet-time apart.
- Circuit switching: setup time + L / R + propagation delay.

Worked example: 12,000 bits over 3 links of 1 Mbps, packets of 2,000 bits.

- Message switching: 3 × (12,000 / 1,000,000) = 3 × 12 ms = 36 ms.
- Packet switching: n = 6, so (3 + 6 – 1) × 2 ms = 16 ms.

- Splitting the message lets links work in parallel, so packet switching is faster than store-and-forward of the whole message.

Remember for the exam

- Circuit switching: dedicated path, setup first, no reordering, resources reserved. Packet switching: shared links, no setup in datagram.
- Datagram packets can be **out of order and take different routes**; virtual circuit packets keep order on one route.
- Message and packet switching both use store and forward. The difference is the unit: **whole message** or **small packets**.
- Virtual circuit is still packet switching, not circuit switching.
- Delay formula for packet switching uses (k + n – 1), not k × n.

Check yourself

1. Which technique suits a voice call that needs steady delay? *(Circuit switching.)*
2. Does each datagram need the full destination address? *(Yes, because routers decide per packet.)*
3. 20,000 bits cross 4 links of 2 Mbps in 5,000-bit packets. Time for packet switching? *(n = 4, packet time 2.5 ms, so (4 + 4 – 1) × 2.5 = 17.5 ms.)*
4. Why can message switching be slow? *(Each node must receive the entire message before forwarding it.)*

Do next

Redo the worked example with 1, 3 and 12 packets and watch the total time fall as the packets get smaller. Then practise GATE previous-year questions on switching and packet delay. The next topic introduces layers, which split communication into separate jobs.

---

7. Layering and the OSI model

Chapter: 1. Introduction to Computer Networks · **Videos:** Layering in computer networks (7 min) · The OSI reference model, part 1 (8 min) · The OSI reference model, part 2 (12 min) · The OSI reference model, part 3 (16 min) · The OSI reference model, part 4 (9 min)

Key ideas

- **Layering** splits the big problem of communication into smaller parts. It gives a **modular design** and makes **troubleshooting** easier, because a fault can be traced to one layer. The protocols in each layer handle that layer's job.
- The **OSI reference model** (Open Systems Interconnection, by ISO) is a guideline, not a protocol. It lets different systems communicate without changing their own hardware or software. It was never fully implemented.
- **TCP/IP** was developed before OSI, so its layers do not match OSI one to one.
- The seven OSI layers, top to bottom: **Application (7), Presentation (6), Session (5), Transport (4), Network (3), Data link (2), Physical (1)**. A bottom-up mnemonic: "Please Do Not Throw Sausage Pizza Away" (Physical, Data link, Network, Transport, Session, Presentation, Application).
- On sending, data moves down the layers and each layer adds its control information. On receiving, it moves up and each layer removes its own.
- **Application:** the interface to the user's programs: file transfer and access, mail, directory services.
- **Presentation:** syntax and meaning of the data: **translation**, **encryption** and **decryption**, **compression**.
- **Session:** opens, maintains and closes dialogues: **dialogue control** and **synchronisation** (checkpoints).
- **Transport:** process-to-process delivery of the whole message: **port addressing**, **segmentation and reassembly**, **connection control** (connection-oriented or connectionless), **end-to-end flow control** and **error control**.
- **Network:** source-to-destination delivery across networks: **logical (IP) addressing** and **routing**.
- **Data link:** node-to-node delivery on one link. It does **framing** and **physical (MAC) addressing**, controls who may use a shared link (**access control**), and handles flow and error control between neighbours. It adds a header and a **trailer**.
- **Physical:** moves raw bits on the medium: signal representation of bits, data rate, bit synchronisation, line configuration, physical topology and transmission mode.
- Each layer adds a header (H7 to H2) to what comes from above; the data link layer also adds a trailer. Each layer on the receiver reads only its own header.

Remember for the exam

- Remember the order and the numbers: Physical is layer 1, Application is layer 7.
- **Port** goes with transport, **IP** with network, **MAC** with data link.
- The OSI model is a reference, TCP/IP is what is actually used.
- Transport is end to end between processes; data link is hop by hop.

- Only the data link layer adds a **trailer**.
- Routers work up to the network layer; switches work at the data link layer.

Check yourself

1. Which layer converts a message into a form the other side can read, including encryption? *(Presentation.)*
2. Which layer cuts a large message into numbered pieces? *(Transport: segmentation and reassembly.)*
3. At which layer does a router decide where to forward a packet? *(Network layer, using the IP address.)*
4. Which layer adds an error-check trailer? *(Data link.)*
5. Name the layer that decides whether bits travel as electrical, light or radio signals. *(Physical.)*

Do next

Write the seven layers from memory, and next to each put one address or service it owns. Then trace a web request from your browser down through the layers and back up at the server. The next topic shows how the three addresses are added and changed on the way, and compares OSI with TCP/IP.

---

8. Addressing and the TCP/IP suite

Chapter: 1. Introduction to Computer Networks · **Videos:** Addressing in networking (15 min) · The TCP/IP protocol suite (10 min)

Key ideas

- Sending data from one process to another uses all three addresses. The **transport layer** adds the **source and destination port**. The **network layer** adds the **source and destination IP address**. The **data link layer** puts both **MAC addresses** (sender and next hop) in its header, plus an error-check trailer.
- A port is 16 bits. An IPv4 address is 32 bits and an IPv6 address is 128 bits. A MAC address is 48 bits.
- A host does not know the MAC address of a far-away machine. To leave its own LAN it sends the frame to its **default gateway**, the router interface on its LAN. The frame carries the sender's MAC as source and the **router's MAC** as destination, while the IP header still names the real destination.
- At each router the frame is opened, the data link header is discarded, and the router reads the destination IP address, finds the next hop and builds a **new frame** with new MAC addresses for the next link. So **IP addresses stay the same end to end** (ignoring NAT), but **MAC addresses change at every hop**.
- The final router puts the destination host's own MAC address on the last frame. The host's data link layer accepts it, the network layer checks the IP address, and the transport layer uses the port to pick the process.
- In its four-layer form the **TCP/IP model** is: **Network access** (OSI layers 1 and 2), **Internet** (OSI network layer), **Transport**, and **Application** (OSI layers 5, 6 and 7 merged). Some books split network access into physical and data link, giving five layers.
- Example protocols by layer:
  - Network access: Ethernet, PPP, frame relay.
  - Internet: IPv4, IPv6, ICMP, and routing protocols such as OSPF and BGP.
  - Transport: **TCP** (Transmission Control Protocol) and **UDP** (User Datagram Protocol).
  - Application: HTTP, DNS, DHCP, FTP, SMTP, POP3, IMAP.
- A **protocol data unit (PDU)** is the name of the data at each layer: **data** (application), **segment** (transport; a UDP unit is usually called a datagram), **packet** (internet), **frame** (network access or data link), **bits** (physical).
- A frame is medium dependent: an Ethernet frame differs from a Wi-Fi frame.

Formulas and methods

Encapsulation size arithmetic. Take a 100-byte application message, a 20-byte TCP header, a 20-byte IPv4 header, a 14-byte Ethernet header and a 4-byte Ethernet trailer.

Step	PDU	Size
Application	data	100 bytes
Transport adds 20	segment	120 bytes

Step	PDU	Size
Internet adds 20	packet	140 bytes
Data link adds 14 + 4	frame	158 bytes

Tracing one hop pair, with names we choose: host H1 (IP `P`, MAC `m1`) sends to server S (IP `Q`, MAC `m9`) through router R (LAN-side MAC `r1`, far-side MAC `r2`).

- Frame on the first link: IP P to Q; MAC m1 to r1.
- Frame on the last link: IP P to Q; MAC r2 to m9.
- Only the MAC pair changed.

Remember for the exam

- Router changes the MAC addresses and keeps the IP addresses; a switch changes neither.
- Port: transport layer. IP: network layer. MAC: data link layer.
- PDU names in order: data, segment, packet, frame, bits.
- TCP/IP application layer = OSI application + presentation + session. TCP/IP network access = OSI data link + physical.
- The destination MAC of the first frame is the **default gateway's**, not the final server's.
- TCP and UDP are transport protocols; IP and ICMP are internet-layer protocols.

Check yourself

1. In a TCP/IP stack, which PDU contains both a TCP header and an IP header, but no frame header? *(The packet.)*
2. A PC sends to a web server in another country. Whose MAC address is the destination in the first frame? *(The default gateway router's.)*
3. Which TCP/IP layer covers the jobs of OSI layers 5, 6 and 7? *(The application layer.)*
4. A segment is 540 bytes, IPv4 header 20 bytes. How large is the packet? *(560 bytes.)*

Do next

Draw the sender, one router and the receiver, and write the MAC and IP pair on each of the two frames. Then practise GATE previous-year questions on encapsulation and layer functions. The next video shows these addresses on a real machine through command-line tools.

---

9. Networking commands

Chapter: 1. Introduction to Computer Networks · **Videos:** Basic networking commands, part 1 (14 min)

Key ideas

Commands are shown for the Windows command prompt, with Linux and macOS equivalents where they differ.

- `ipconfig` shows each adapter's **IPv4 address**, **subnet mask** and **default gateway**. These are layer 3 details. The subnet mask tells the host which addresses are on its own network; anything else must go to the gateway.
- The **default gateway** is the first router a packet reaches when the destination is outside the local network. For traffic to another network, the host sends the data to the gateway and the router forwards it.
- `ipconfig /all` adds the **physical address** (MAC address) and more detail, such as DNS servers and whether DHCP is used. This is the layer 2 detail. Linux uses `ip addr` or `ifconfig`.
- `nslookup <name>` asks a **DNS server** for the IP address of a host name. Users type names, but packets need IP addresses, so DNS translates between them.
- `ping <address or name>` tests whether another host is **reachable**. It sends **ICMP echo request** messages and reports the replies, round-trip time and packet loss. By default Windows sends four, Linux keeps sending until stopped. A reply for every request means 0% loss; no replies means 100% loss.
- `tracert <address or name>` (Linux and macOS: `traceroute`) lists the **routers** on the path, one **hop** per line, with timing. Windows stops at 30 hops by default. Each line is a router the packet passed through; the first hop is normally the default gateway.
- `cls` clears the Windows screen (`clear` on Linux) and `Ctrl+C` stops a command that is running.

Formulas and methods

How to use the commands together when a website will not open:

1. `ipconfig`: check that you have an IP address and a default gateway. An address starting with `169.254` usually means DHCP failed.
2. `ping <gateway IP>`: can you reach your router?
3. `ping 8.8.8.8` (a public IP address): can you reach the Internet without DNS?
4. `ping example.com`: if step 3 works and this fails, DNS is the problem. Confirm with `nslookup example.com`.
5. `tracert <host>`: find where along the path replies stop.

How `tracert` works: it sends probes with a **TTL (time to live)** of 1, then 2, then 3 and so on. A router that reduces TTL to 0 drops the packet and returns an ICMP "time exceeded" message, which reveals that router. For example, a path with 9 routers needs a TTL of 10 to reach the destination host, so it prints about 10 lines.

Remember for the exam

- `ipconfig` = IP-layer information; `ipconfig /all` = the MAC address as well.
- `nslookup` = name to IP (DNS); `ping` = reachability and loss; `tracert` = path of routers.
- Ping uses **ICMP**, a network-layer protocol, not TCP or UDP.
- "Request timed out" can mean the host is down or that a firewall blocks ICMP. It does not always mean the host is off.
- A TTL of 1 expires at the first router, which is how `tracert` finds each hop.

Check yourself

1. Which command shows the MAC address on Windows? *(`ipconfig /all`.)*
2. You can `ping 8.8.8.8` but not `ping example.com`. What is likely wrong? *(DNS name resolution.)*
3. Which protocol does `ping` use? *(ICMP echo request and echo reply.)*
4. Which command lists the routers between you and a server? *(`tracert`, or `traceroute` on Linux and macOS.)*

Do next

Run all five commands on your own machine against a site you use, and write down your gateway, your MAC address and the hop count to the site. Next, Cisco Packet Tracer lets you build a small network and run these same commands inside it.

---

10. Packet Tracer: hubs and switches

Chapter: 1. Introduction to Computer Networks · **Videos:** Packet Tracer basics, part 1 (12 min) · Packet Tracer basics, part 2: hub (14 min) · Packet Tracer basics, part 3: switch (12 min)

Key ideas

- **Cisco Packet Tracer (CPT)** is a free simulator from Cisco's Networking Academy. You drag devices onto a workspace, cable them, configure them and test them without any hardware. It is the practical side of this course and useful for CCNA later.
- The device bar groups routers, switches, hubs, wireless devices, cables, end devices (PCs, laptops, servers, phones) and security devices. Two cable types matter first: **straight-through** (joins unlike devices, such as PC to hub or PC to switch) and **crossover** (joins like devices, such as PC to PC or switch to switch).
- Two PCs joined by a crossover cable make the smallest **peer-to-peer network**. Each PC needs an **IP address** (Desktop > IP Configuration) and they need to share a subnet. Subnet masks come later in the course.
- Useful commands in a PC's Command Prompt: `ipconfig` shows the IP address, `ipconfig /all` also shows the **MAC address** (a 48-bit physical address; Cisco writes it with dots, Windows with hyphens), and `ping <IP>` tests reachability. A default ping sends four echo requests; four replies and 0% loss means the path works.
- A **hub** works at the physical layer (layer 1) and has many ports. Whatever arrives on one port is copied out of every other port. Hubs make a **star topology** LAN.
- Because a hub repeats every bit everywhere, every PC receives every frame, and only the addressed PC keeps it. The rest discard it. This wastes bandwidth, lets any host capture others' traffic, and puts all ports in one **collision domain**.
- A **switch** is a layer 2 (data link layer) device. It keeps a **MAC address table** that maps each MAC address to a port number, so a frame goes only to the port of its destination. This is **unicast** forwarding. Broadcasts still go to all ports.
- A switch learns the table itself by reading the source MAC address of each incoming frame. In Packet Tracer a new switch link shows an amber light for a short time while it settles, then turns green.
- The switch never looks at IP addresses. The PCs still need them, because `ping` and other programs name the other host by its IP address.

Remember for the exam

- Hub: layer 1, no memory, broadcasts every frame, one collision domain. Switch: layer 2, has a MAC address table, forwards to the destination port only.
- Cable rule of thumb: **unlike devices use straight-through, like devices use crossover**. Many modern devices detect this automatically, but exam answers follow the rule.
- A hub is cheap but a poor choice when security or load matters. A switch is the normal LAN choice.
- Ports on a switch here are hardware interfaces. Do not confuse them with the transport layer's port numbers.

Check yourself

1. Which cable joins a PC to a switch? *(Straight-through, because they are different kinds of device.)*
2. A hub has 8 ports and PC1 sends a frame. How many copies leave the hub? *(Seven, one out of every port except the arrival port.)*
3. Why does a switch send a frame to one port only? *(It looked up the destination MAC address in its MAC address table and found the port.)*
4. Which command shows a PC's MAC address in Packet Tracer? *(`ipconfig /all`.)*

Do next

Build two small LANs in Packet Tracer, one with a hub and one with a switch, give each PC an address such as 192.168.1.x, and watch a single ping in Simulation mode on both. Count which PCs receive the frame. The next topic adds routers to join separate LANs.

---

11. Routers, repeaters and bridges

Chapter: 1. Introduction to Computer Networks · **Videos:** Basics of router (10 min) · Packet Tracer basics, part 4: router (15 min) · Packet Tracer basics, part 5: repeater (7 min) · Basics of bridge (5 min)

Key ideas

- A **router** is a layer 3 (network layer) device that forwards packets between different networks. A switch joins devices inside one network. To go from the 10.0.0.0 network to the 192.168.1.0 network you need a router.
- A router has at least two **interfaces**, each in a different network and each with its own IP address. Interfaces on a Cisco router are **shut down by default**, so you must give each an IP address and switch it on.
- A router keeps a **routing table** and decides by the destination **IP address**. A switch keeps a MAC address table and decides by MAC address.
- A host sends traffic for another network to its **default gateway**, which is the IP address of the router interface on its own LAN. Each LAN uses a different gateway address, the one on its own side.
- Packet Tracer router lab: build two LANs (each a switch with three PCs), join each switch to a router interface with a straight-through cable, set the interface addresses, and set each PC's default gateway. If you forget the gateway, pings across networks fail. Once it is set, the first ping often times out and the rest succeed, because the first packet is spent on address resolution.
- A **repeater** is a layer 1 device with two ports. It only regenerates a weakened signal so a cable run can be longer, for example between two buildings. It reads no addresses.
- A **bridge** is in effect a repeater that also reads **MAC addresses** (layer 2). It joins two LAN segments that use the same protocol, and it can filter frames by MAC address. A multiport bridge is what we now call a switch.
- Two bridge types: **transparent** (hosts do not know the bridge exists, so nothing to configure) and **source routing** (the sending station puts the route in the frame).
- Router versus bridge: a router joins networks that may use different protocols and works on IP addresses, and it rewrites the MAC addresses in the frame at each hop. A bridge joins segments of the same protocol and works on MAC addresses.

Remember for the exam

- Layers: repeater = 1, bridge = 2, switch = 2, router = 3.
- Switch and bridge forward by MAC address, router by IP address.
- A router connects different networks; a switch connects devices in the same network.
- A repeater adds no intelligence. It cleans and resends the signal and does not filter.
- Default gateway = the router interface address on the host's own subnet.

Check yourself

1. PC A (10.1.1.5) and PC B (172.16.0.9) hang off one switch. Can they talk? *(No. They are in different networks, so a router is needed.)*

2. A new router interface has the right IP address but traffic does not flow. What is the first thing to check? *(Whether the interface is turned on; it is shut down by default.)*
3. Which device would you add to extend a cable run that is too long without changing any addressing? *(A repeater.)*
4. What does a router read that a bridge does not? *(The destination IP address in the packet header.)*

Do next

Rebuild the two-LAN lab in Packet Tracer, then deliberately delete one PC's default gateway and see the ping fail. Then read the next topic, which puts all these devices side by side with solved questions.

---

12. Network devices compared

Chapter: 1. Introduction to Computer Networks · **Videos:** Network devices (5 min) · Network devices: solved questions (8 min)

Key ideas

- A **network interface card (NIC)** is the part inside a PC or server that connects it to the network. It is an end-device component, not an intermediary device.
- **Repeater** (layer 1): regenerates a signal within one LAN. It cannot read addresses.
- **Hub** (layer 1): a multiport repeater. It **broadcasts** every incoming signal to all other ports.
- **Switch** (layer 2): uses a **MAC address table** to **unicast** frames to the right port. It connects devices in one LAN.
- **Bridge** (layer 2): joins two LAN segments of the same protocol, filtering by MAC address.
- **Router** (layer 3): forwards by IP address and can join different networks and different link technologies, such as Ethernet on one side and Wi-Fi on the other.
- **Multilayer (layer 3) switch**: a switch that can also do routing between networks, so it is both layer 2 and layer 3.
- **Brouter**: a device that acts as a bridge for some traffic and a router for the rest.
- **Modem** (modulator-demodulator): puts digital data onto an analog carrier at the sender and recovers it at the receiver. The classic case is data over a telephone line.
- **Firewall**: filters incoming and outgoing traffic by IP address, port number or application data, to protect a network.

Device	Layer	Decides by	Sends to
Repeater	1	nothing	the other side
Hub	1	nothing	all ports
Bridge / switch	2	MAC address	one port (or all for broadcast)
Router	3	IP address	next hop

Formulas and methods

Counting MAC addresses on a path. The source and destination IP addresses stay the same from end to end, but a router makes a new frame on each hop with new source and destination MAC addresses. So:

`destination MAC addresses seen = number of routers on the path + 1` (one per link)

Example: host P sends to host Q through three routers in a line. The frame has four different destination MACs on the way: router 1's incoming interface, router 2's incoming interface, router 3's incoming interface, and finally Q's own MAC address.

Remember for the exam

- Layer 1: repeater, hub. Layer 2: bridge, switch. Layer 3: router. A trick question will offer "hub" as a layer 2 device. It is not.
- Hub = broadcast always; switch = unicast after learning. For security, pick a switch.
- Bridge joins same-protocol segments; router joins different protocols.
- IP addresses stay unchanged across a path (without NAT); MAC addresses change at every router.

Check yourself

1. Which devices among hub, repeater, bridge and switch are layer 2? *(Bridge and switch.)*
2. Which component inside a PC connects it to the network? *(The NIC.)*
3. A frame crosses two routers between two hosts. How many distinct destination MAC addresses appear? *(Three.)*
4. A client wants a medium-sized office LAN. Hub or switch? *(Switch. A hub would flood every host with all traffic.)*

Do next

Write the table above from memory with the layer and decision field for each device, then check it. Practise GATE previous-year questions on network devices and layers. Next up is the physical layer, where signals and cables are the topic.

---

13. Physical layer and media

Chapter: 1. Introduction to Computer Networks · **Videos:** Physical layer and media, part 1 (7 min) · Physical layer and media, part 2 (8 min) · Physical layer and media, part 3 (11 min) · Line configuration (6 min)

Key ideas

- The **physical layer** turns the bits of a frame into signals and puts them on a transmission medium, then turns signals back into bits at the receiver. A user sees an image or a voice; the medium only carries electromagnetic signals.
- On the way down, each layer wraps the data below it: application data, then **segment** (transport), **packet** (network), **frame** (data link), then plain **bits** at the physical layer.
- A **signal** shows how a physical quantity changes with time. An **analog signal** can take any value in a range, with infinitely many values (sound, temperature). A **digital signal** takes one of a finite set of values at discrete times.
- **Wired media:** copper (twisted pair and coaxial) carries electrical signals; **fibre optic** cable carries light pulses (light on is 1, off is 0). **Wireless media** use radio waves, with an access point, a wireless NIC and antennas.
- **UTP** (unshielded twisted pair) has no metal shield. **STP** (shielded twisted pair) has one. Shielding reduces **EMI/RFI** and so reduces **crosstalk** (a signal leaking between nearby wire pairs), but costs more. UTP fights crosstalk with twists instead: each pair is twisted at a different rate. Here STP means shielded twisted pair, not the Spanning Tree Protocol.
- **Single-mode fibre** carries one light path; **multimode fibre** carries many paths and suffers more dispersion, so it suits shorter distances.
- Wireless weak points: limited **coverage**, **interference** between radio signals, and weaker **security**, because anyone in range can receive the signal.
- **Line configuration** is how two nodes share a link. In **point-to-point**, a dedicated link joins exactly two devices and the whole capacity is theirs. In **multipoint** (multidrop), many devices share one link and its capacity. Sharing can be **spatial** (several devices use it at once, so collisions are possible) or **temporal** (devices take turns).

	Copper (UTP)	**Fibre optic**
Signal	electrical	light
Typical max run (Ethernet)	about 100 m	kilometres to tens of km
Immunity to EMI/RFI	low	very high (essentially immune)
Cost and installation skill	low	higher

Remember for the exam

- Physical layer unit: **bits**. Data link: **frames**. Network: **packets**. Transport: **segments**.
- UTP cheaper but crosstalk-prone; STP better shielded but dearer; fibre best for distance, speed and immunity.

- Point-to-point has a dedicated link; multipoint shares one, so it needs a rule for taking turns.
- Real-life signals (sound, light, temperature) are analog; computers work with digital data.

Check yourself

1. Which layer handles bit-to-signal conversion? *(The physical layer.)*
2. How do you reduce crosstalk without buying STP? *(Twist each pair more tightly, and at a different rate from its neighbours.)*
3. A link between two routers is used by those two only. What is its line configuration? *(Point-to-point.)*
4. Why is fibre immune to EMI? *(It carries light, not electric current.)*

Do next

Compare a UTP cable and an optical fibre datasheet from any vendor and fill in the table above with real figures. The next chapter starts the data link layer, which builds frames from the bits this layer sends.

---

14. Data link layer services

Chapter: 2. Data Link Layer · **Videos:** Link layer services (9 min) · Sub-layers of the data link layer (7 min)

Key ideas

- The **data link layer** is layer 2 of the OSI model, also called the **link layer**. It takes a packet from the network layer, adds a **header and trailer** to make a **frame**, and moves that frame from one node to the next node on the same link.
- The layer offers five services: **framing**, **MAC addressing** (physical addresses), **access control**, **flow control** and **error control**.
- **Framing:** packing bits into frames so the receiver can tell where one ends and the next begins. A letter in an envelope is the usual analogy: the envelope marks the boundary.
- **Physical addressing:** the header carries the source and destination **MAC addresses**, so the frame reaches the right node on the link. IP addresses are added one layer up.
- **Flow control:** a speed-matching rule so a fast sender does not overrun a slow receiver. The sender may send only as much as the receiver can take before an acknowledgement comes back. In the data link layer this works between neighbouring nodes on a link (hop to hop), not between the two end hosts. End-to-end flow control is the transport layer's job.
- **Error control:** protects frames against corruption in transit. The receiver can **detect** errors and discard the frame, or **correct** them. Detection is simpler than correction.
- **Access control (media access control):** decides who may transmit when many devices share one medium, so frames do not collide.
- The layer has two **sub-layers**. The **logical link control (LLC)** sub-layer, also called **data link control (DLC)**, talks to the network layer above and adds the control information used for flow control. The **MAC** sub-layer sits at the bottom, next to the physical layer, and is implemented in the **NIC**.
- The MAC sub-layer does **data encapsulation** (frame assembly before sending, disassembly on receipt) and **media access control**: placing frames on the medium and taking them off.
- In this course's split, the LLC handles flow control and the MAC sub-layer handles the other four (framing, MAC addressing, error and access control). Some textbooks put framing and error control under the LLC/DLC too, so use the split your syllabus gives.

Remember for the exam

- Data link PDU = **frame**; unit of address = **MAC address**; node-to-node delivery is its job.
- Name all five services: framing, MAC addressing, access control, flow control, error control.
- Flow control here is hop to hop; transport-layer flow control is end to end.
- The MAC sub-layer is the one that touches the physical layer and lives in the NIC.

Check yourself

1. What is added to a packet to make a frame? *(A data link header and trailer.)*
2. Which service stops a fast sender from flooding a slow receiver? *(Flow control.)*

3. Where is the MAC sub-layer implemented? *(In hardware, typically on the network interface card.)*
4. Which service stops two devices on a shared cable from transmitting at once? *(Access control.)*

Do next

Draw a frame around a packet and label each of the five services against the part of the frame or the mechanism that provides it. Then continue to framing, the first service worked through in detail.

---

15. Framing

Chapter: 2. Data Link Layer · **Videos:** Framing, part 1 (7 min) · Framing, part 2 (6 min) · Framing, part 3 (4 min)

Key ideas

- The physical layer sends a stream of bits and knows nothing about frames. The receiver's **adapter** must work out which bits form one frame. **Framing** is how the data link layer marks those boundaries. Between adapters you see a bit stream; between nodes you see frames.
- The sender's adapter is told to transmit a frame held in the node's memory. The receiving adapter collects the incoming bits and stores the matching frame in memory.
- **Fixed-size framing:** every frame has the same length, so the length itself acts as the delimiter. No start or end marker is needed. After the agreed number of bits, one frame ends and the next begins.
- **Variable-size framing:** frame lengths differ, so the receiver needs something else to find the boundaries, such as start and end markers or a length field.
- The sender and receiver agree on a **protocol**, for example "a frame starts and ends with the pattern 11011". The sender adds that pattern in the header and trailer.
- The **framing error**: if the payload (which comes from upper layers and can be anything) contains the same bit pattern as the delimiter, the receiver sees a false start or end and cuts the frame in the wrong place. Solving this is called **stuffing**, covered in the next topics.
- Three framing approaches:
  - **Bit-oriented:** the frame is a sequence of bits, so any content (text, audio, video) can be carried. Example: **HDLC**, which grew out of IBM's SDLC.
  - **Byte-oriented** (character-oriented): the frame is a sequence of bytes or characters. Examples: **BISYNC**, **PPP**, **DDCMP**.
  - **Clock-based:** frames are positioned by timing in a synchronous signal, mainly in optical networks. Example: **SONET** (Synchronous Optical Network).

Approach	Delimiting idea	Examples
Bit-oriented	special bit pattern	HDLC
Byte-oriented	special characters, or a byte count	BISYNC, PPP, DDCMP
Clock-based	fixed timing of a synchronous signal	SONET

Remember for the exam

- Fixed-size frames need no delimiter; variable-size frames do.
- A framing error happens when the delimiter pattern appears inside the data.
- Bit-oriented = bits (HDLC); byte-oriented = characters (BISYNC, PPP, DDCMP); clock-based = SONET.
- Bits travel between adapters; frames are exchanged between nodes.

Check yourself

1. Frames are always 200 bits. Do you need an end marker? *(No. Count 200 bits and the frame is complete.)*
2. The delimiter is 0110 and the payload holds 0110. What goes wrong? *(The receiver may end the frame early, a framing error.)*
3. Which approach does SONET use? *(Clock-based framing.)*
4. Name the three byte-oriented protocols. *(BISYNC, PPP and DDCMP.)*

Do next

Take any 12-bit payload, pick a 4-bit delimiter that occurs inside it, and show the receiver cutting the frame wrongly. Then study HDLC and bit stuffing, the standard cure for the bit-oriented case.

---

16. HDLC and bit stuffing

Chapter: 2. Data Link Layer · **Videos:** High-level data link control (HDLC) (6 min) · Bit stuffing (5 min)

Key ideas

- A **bit-oriented protocol** treats a frame as a plain sequence of bits. **HDLC** (High-level Data Link Control) is the main one. It was standardised by ISO from IBM's **SDLC**, and many later link protocols copy its frame layout.
- HDLC frame, left to right: **beginning flag** (8 bits), **header** (address and control, 16 bits in all), **body/payload** (variable length), **CRC** (16 bits, for error detection), **ending flag** (8 bits).
- The **flag** at both ends is `01111110`, a 0, six 1s and a 0. It is also sent while the link is idle, which keeps the two clocks in step.
- The header's two fields are **address** and **control**. The control field tells which kind of frame it is:
  - **I-frame** (information): control starts with `0`. Carries user data.
  - **S-frame** (supervisory): control starts with `10`. Used for flow and error control, such as acknowledgements.
  - **U-frame** (unnumbered): control starts with `11`. Used for link management.
- **Problem:** if the payload contains `01111110`, the receiver takes it for the end flag and splits the frame.
- **Bit stuffing:** after any five consecutive 1s in the data, the sender inserts a 0. Therefore six 1s can never appear inside a frame body, so the flag stays unique. The receiver reverses this: whenever it sees five 1s followed by a 0, it deletes that 0.

Formulas and methods

Bit stuffing method.

1. Scan the data bits from the left. Count consecutive 1s.
2. When the count reaches five, insert a 0 straight after them and reset the count to zero. Do this even if the next data bit is already 0.
3. Continue to the end of the data.

Unstuffing: scan the received bits and, after five consecutive 1s, drop the next bit (it will be the stuffed 0).

Worked example (our own numbers):

- Data: `01111110111110` (14 bits)
- First run: six 1s. After the fifth 1 insert a 0.
- Second run: five 1s, followed by a 0 in the data. Insert a 0 after the fifth 1 anyway.
- Stuffed: `0111110101111100` (16 bits)
- Check by unstuffing: removing the bit after each run of five 1s gives back `01111110111110`.

Overhead of one HDLC frame, ignoring stuffing: 8 + 16 + 16 + 8 = 48 bits. A 1000-bit payload gives a 1048-bit frame, and each stuffed bit adds one more.

Remember for the exam

- Flag = `01111110`. Stuff a 0 after **five** consecutive 1s (not six).
- Stuffing happens at the sender; unstuffing at the receiver. The frame is longer on the wire than the original data.
- Control field: first bit 0 = I; first two bits 10 = S; first two bits 11 = U.
- The stuffed 0 after five 1s is inserted whether or not the next data bit is 0.

Check yourself

1. Stuff this data: `1111101111111`. *(`111110011111011`. Split:* `11111` *+ stuffed* `0` *+ data* `0` *+* `11111` *+ stuffed* `0` *+* `11`*.)*
2. Control field begins `10`. Which frame? *(Supervisory.)*
3. How many bits does the HDLC header have in this course's format? *(16: address plus control.)*
4. Why can the receiver not confuse data with the flag after stuffing? *(No six consecutive 1s occur in the stuffed body.)*

Do next

Stuff and then unstuff five random 20-bit strings by hand, including one with an exact run of five 1s at the end. Practise GATE previous-year questions on bit stuffing. Next, the byte-oriented protocols that solve the same problem with characters.

---

17. BISYNC, PPP and DDCMP

Chapter: 2. Data Link Layer · **Videos:** BISYNC (6 min) · Point-to-Point Protocol (PPP) (6 min) · DDCMP (5 min)

Key ideas

- A **byte-oriented** (character-oriented) protocol sees a frame as a sequence of bytes. Three standard examples: **BISYNC**, **PPP** and **DDCMP**.
- **BISYNC** (binary synchronous communications, also BSC) was developed by IBM. It uses the **sentinel** approach: special characters mark where parts of the frame begin and end. Frame order: two **SYN** characters (8 bits each, mark the start), **SOH** (start of header), the header, **STX** (start of text), the body, **ETX** (end of text), and a 16-bit **CRC**. There is no end-of-frame character after the CRC. ETX closes the body.
- If STX or ETX appears inside the body, the receiver would misread it. **Character (byte) stuffing** fixes this: the sender puts a **DLE** (data link escape) character before such a character in the body, and the receiver removes the DLE. A DLE that occurs in the data is escaped too.
- **PPP** (Point-to-Point Protocol) is a WAN protocol used on direct links such as two routers, and for broadband and dial-up access. It carries data of several network protocols.
- PPP frame: **flag** (1 byte, `01111110`), **address** (1 byte, normally all 1s, the broadcast value), **control** (1 byte, constant `00000011`), **protocol** (1 or 2 bytes, says what the payload holds), **payload** (variable; default maximum 1500 bytes, can be negotiated), **checksum** (2 bytes, or 4), and a closing **flag**.
- The PPP trailer is a frame check sequence, a CRC-based check. Other error-detection methods come in a later topic.
- PPP uses **byte stuffing**: if the flag byte appears in the payload, an escape byte is added so the receiver does not take it for the end of the frame.
- **DDCMP** (Digital Data Communications Message Protocol) comes from Digital Equipment Corporation. It uses **byte counting** instead of a trailing marker. Frame: two SYN bytes, **class** (1 byte), **count** (14 bits, the number of bytes in the body), a header, the body and a 16-bit CRC.
- Weakness of byte counting: if a transmission error damages the **count**, the receiver finds the wrong end of the frame and every frame after it is misaligned. This is a **framing error**.

Formulas and methods

Largest body DDCMP can describe: the count has 14 bits, so `max body = 2^14 − 1 = 16383 bytes`.

Byte stuffing in PPP. Reserve one byte as an escape. In common PPP implementations the flag `0x7E` in the data is sent as the escape `0x7D` followed by `0x7E XOR 0x20 = 0x5E`. The receiver reverses the step.

Count-error example (our own numbers). Frames A, B and C have 3-, 5- and 4-byte bodies. If A's count is damaged from 3 to 6, the receiver takes 6 bytes as A's body. That covers A's real body plus the first bytes of B, so it treats B's header bytes as data, then reads a wrong count from the middle of B and stays out of step for the following frames.

Remember for the exam

- Byte-oriented: BISYNC (sentinel with DLE stuffing), PPP (flag with byte stuffing), DDCMP (byte count). Bit-oriented: HDLC.
- BISYNC and PPP use delimiters and so need stuffing. DDCMP uses a count and does not, but a damaged count breaks framing.
- PPP's extra field compared with HDLC: **protocol**.
- DDCMP count field = **14 bits**.

Check yourself

1. Which BISYNC character is used to escape a special character in the body? *(DLE.)*
2. What does PPP's protocol field tell the receiver? *(What type of data is in the payload.)*
3. What is the largest body a 14-bit count can describe? *(16383 bytes.)*
4. Why can a single bit error in DDCMP's count damage several frames? *(The receiver mislocates the end of this frame, so the following frames start in the wrong place.)*

Do next

Write the field order of BISYNC, PPP and DDCMP side by side on one page and mark which method (sentinel, flag, count) each uses. Practise GATE previous-year questions on framing and stuffing. Next in this chapter come the other data link services, starting with error control.

---

18. Error detection: parity, LRC, checksum

Chapter: 2. Data Link Layer · **Videos:** Error Detection (8 min) · Vertical Redundancy Check (VRC) (7 min) · Longitudinal Redundancy Check (LRC) (7 min) · Checksum (10 min)

Key ideas

- A **transmission error** means the bits that arrive differ from the bits that left. Error control is a service of the data link layer and the transport layer.
- A **single-bit error** changes exactly one bit in a data unit. A **burst error** changes two or more bits, and its **length** is measured from the first corrupted bit to the last corrupted bit, including unchanged bits in between.
- The receiver has no copy of the original message, so the sender adds **redundant bits** computed from the data. The receiver recomputes them and compares. A match means accept, a mismatch means discard.
- **Detection** only says "good or bad". **Correction** must also find or repair the damage, either by asking for **retransmission** or by using an **error-correcting code**. Both need redundancy.
- **VRC (parity check):** one parity bit per data unit. With **even parity** the sender sets the bit so that the total number of 1s, parity bit included, is even. It detects every odd number of flipped bits and misses every even number.
- **LRC (two-dimensional parity):** the data units are stacked as rows, and one parity bit is computed for each **column**. The block of column parities is sent as an extra row. It catches burst errors far better than VRC, but fails when two bits in the same column positions of two rows flip together.
- **Checksum:** split the message into k blocks of n bits, add them with **wrap-around carry** (**one's complement addition**), then take the **one's complement** of the sum. That n-bit result is the checksum. The receiver adds all blocks plus the checksum and accepts only if the result is all 1s.
- The checksum catches every single-bit error and most others, but it misses errors that cancel out in the sum: for example, the same bit position going from 0 to 1 in one block and from 1 to 0 in another.

Formulas and methods

Even parity (VRC). Count the 1s in the data. If the count is odd, parity bit = 1. If even, parity bit = 0.

Example: data `1011001` has four 1s, so the parity bit is 0 and the frame is `10110010`. If one bit flips, the receiver counts an odd number of 1s and rejects it. If two bits flip, the count stays even and the error goes unnoticed.

LRC. Four rows of six bits:

```
101100
011010
110001
001111
------
001000   <- column parities (even parity)
```

Column 1 has two 1s (parity 0), column 3 has three 1s (parity 1), and the other columns have two 1s each. The sender transmits the four rows followed by `001000`.

Checksum (4-bit blocks). Blocks `1011`, `1110`, `0111`.

1. `1011 + 1110 = 11001`. The carry 1 wraps around: `1001 + 1 = 1010`.
2. `1010 + 0111 = 10001`. Wrap the carry: `0001 + 1 = 0010`.
3. One's complement of `0010` is `1101`. This is the checksum.

Receiver: `1011 + 1110 + 0111 + 1101` gives `1111` after wrap-around, so the data is accepted.

Remember for the exam

- The checksum is the **complement** of the wrapped sum. At the receiver, the sum of everything including the checksum is all 1s (its complement is 0).
- VRC and LRC use **parity**; the checksum uses **addition**. Do not mix up which one works on columns: LRC does.
- A burst error of length 8 does not mean 8 flipped bits. It means 8 bits from first to last damaged bit.
- VRC cannot detect an even number of bit errors, however large the burst.
- Checksum and CRC are the two techniques used in real protocols (the checksum in IP, TCP and UDP; CRC in link-layer frames).

Check yourself

1. What parity bit does even parity give for `1101110`? *(Five 1s, an odd count, so the parity bit is 1.)*
2. Three bits of one frame flip and the frame uses VRC. Is the error detected? *(Yes: an odd number of flipped bits changes the parity.)*
3. In LRC, which two-bit pattern of errors slips through? *(Two bits in the same position of two different rows, because both column parities are unchanged.)*
4. A checksum of 4-bit blocks `0110` and `1001` is computed. What is it? *(Sum is `1111`, no carry, complement is `0000`.)*

Do next

Take any eight-bit values you like, compute VRC, LRC and a checksum by hand, then flip bits to see which flips each method misses. Next comes CRC, which is stronger than all three and is built on binary division.

---

19. Cyclic redundancy check

Chapter: 2. Data Link Layer · **Videos:** Cyclic Redundancy Check (CRC) - Part 1 (9 min) · Cyclic Redundancy Check (CRC) - Part 2 (6 min) · Cyclic Redundancy Check (Solved problem) (9 min)

Key ideas

- **CRC** is the strongest of the four detection methods in this chapter. The redundant bits it produces are called the **CRC** (or FCS, frame check sequence).
- Sender and receiver agree beforehand on a **divisor** (the **generator**). The same divisor is used to generate and to check.
- The divisor is a bit string of length L, and it is also written as a polynomial. For example `1011` is `x^3 + x + 1`. Good divisors begin and end with 1.
- The arithmetic is **modulo-2 division**: there are no borrows and no carries, and subtraction is the same as addition, which is the same as **XOR**.
- **Generation (sender):** append L – 1 zeros to the data, divide by the divisor with modulo-2 division, and keep the **remainder** (L – 1 bits). Replace the appended zeros with the remainder and send the result (the **codeword**).
- **Checking (receiver):** divide the received codeword by the same divisor. A remainder of all zeros means accept. Any nonzero remainder means the frame is corrupted.
- The codeword is always an exact multiple of the divisor, which is why a clean frame leaves remainder zero.
- A well-chosen divisor detects all single-bit errors, all burst errors no longer than L – 1 bits, and (with the factor `x + 1`) all errors with an odd number of bits. Real links use standard generators such as CRC-16 and CRC-32.

Formulas and methods

1. Let the divisor have L bits. Append L – 1 zeros to the data.
2. Divide with XOR: if the leading bit of the current window is 1, XOR it with the divisor; if it is 0, XOR with all zeros (just shift). Bring down the next bit and repeat.
3. The final L – 1 bit remainder is the CRC.
4. Transmit data followed by the CRC.

Worked example. Data `110101`, divisor `1011` (L = 4), so append three zeros: `110101000`.

```
1101 XOR 1011 = 0110  -> 110, bring down 0 -> 1100
1100 XOR 1011 = 0111  -> 111, bring down 1 -> 1111
1111 XOR 1011 = 0100  -> 100, bring down 0 -> 1000
1000 XOR 1011 = 0011  -> 011, bring down 0 -> 0110
0110 : leading 0, shift  -> 110, bring down 0 -> 1100
1100 XOR 1011 = 0111  -> remainder 111
```

The CRC is `111`, so the sender transmits `110101111`. The receiver divides this by `1011` and gets remainder `000`. If the last bit flips in transit, the receiver divides `110101110` and gets remainder `001`, which is nonzero, so the frame is rejected.

Check with polynomials: `x^8 + x^7 + x^5 + x^3` reduced modulo `x^3 + x + 1` leaves `x^2 + x + 1`, which is `111`.

Remember for the exam

- Number of zeros appended = **divisor length – 1** (degree of the polynomial), not the divisor length.
- The remainder always has exactly L – 1 bits. Pad with leading zeros if the division gives fewer.
- Modulo-2 arithmetic: `1 + 1 = 0`, no carry. Addition and subtraction are both XOR.
- The receiver succeeds when the remainder is **zero**; do not confuse this with the checksum rule (all 1s).
- Divisor `11` (`x + 1`) gives the same result as even parity.
- CRC is computed by hardware shift registers in real network cards, which is why it is fast.

Check yourself

1. How many zeros are appended for the divisor `10011`? *(Four, because the divisor has 5 bits.)*
2. Data `1001`, divisor `11`. What is the CRC? *(Append one zero:* `10010` *divided by* `11` *leaves remainder 0, since there are two 1s in the data. The CRC is* `0`*.)*
3. The received codeword leaves remainder `010`. Accept or reject? *(Reject: the remainder is not zero.)*
4. What is the generator `x^4 + x + 1` as a bit string? *(*`10011`*.)*

Do next

Divide `100110` by `1101` by hand with three appended zeros, then check that dividing your codeword leaves zero. Next, the course turns from errors to speed: delay, bandwidth and round trip time, the numbers that flow control needs.

---

20. Delay, bandwidth and RTT

Chapter: 2. Data Link Layer · **Videos:** Network Performance (7 min) · Latency (6 min) · Latency (Simulation) (5 min) · Bandwidth Delay Product (5 min) · Delay (Solved problem) (4 min) · Round Trip Time (RTT) (4 min)

Key ideas

- Network performance is measured by **bandwidth**, **throughput** and **latency** (delay).
- **Bandwidth** is the maximum rate a link can carry, in **bits per second**. For wireless channels the word also means a range of frequencies, measured in **hertz**.
- **Throughput** is the rate actually achieved. It is always less than or equal to bandwidth. Bandwidth is the capacity of a road; throughput is how many cars really get through in traffic.
- **Latency** is the total time for data to get from source to destination. It has four parts:
  - **Transmission delay** (Tt): time to push all bits of the packet onto the link.
  - **Propagation delay** (Tp): time for one bit to travel the length of the link.
  - **Queuing delay**: time spent waiting in a router or switch buffer.
  - **Processing delay**: time to examine the header and decide where to send the packet.
- Transmission delay depends on packet size and bandwidth. Propagation delay depends on distance and signal speed. Making the link faster reduces Tt but never Tp.
- The simulation video shows both delays together: a bigger message or slower rate lengthens the time the sender takes to finish putting bits on the wire, while a longer cable lengthens the time before the first bit appears at the other end. Queuing and processing are ignored in the textbook problems.
- **Round trip time (RTT)** is the time for a signal to go to the receiver and for the reply to come back. In the usual exam simplification, RTT = 2 × Tp, ignoring transmission, queuing and processing. In practice (for example `ping`), measured RTT also includes those delays, so it is larger.
- The **bandwidth-delay product (BDP)** is the number of bits that fit on the link at one time. Think of a pipe: bandwidth is its cross-section, delay is its length, BDP is its volume.

Formulas and methods

Quantity	Formula
Transmission delay	Tt = L / B (bits divided by bits per second)
Propagation delay	Tp = d / v (metres divided by metres per second)
Total latency	Tt + Tp + queuing + processing
RTT (exam form)	2 × Tp
Bandwidth-delay product	B × delay (one-way delay, or RTT for bits "in flight" over a full round trip)
Common signal speeds	about 2 × 10^8 m/s in cable or fibre, 3 × 10^8 m/s in vacuum

Worked example. A 1500-byte packet goes over a 2000 km link at 10 Mbps, with signal speed 2 × 10^8 m/s. Ignore queuing and processing.

- L = 1500 × 8 = 12,000 bits, so Tt = 12,000 / 10^7 = 1.2 ms.

- Tp = 2,000,000 m / (2 × 10^8) = 10 ms.
- Latency = 1.2 + 10 = 11.2 ms. RTT = 2 × 10 = 20 ms.
- BDP with one-way delay = 10^7 × 0.010 = 100,000 bits (12,500 bytes). With RTT it is 200,000 bits.

Watch units: convert km to m, kilobits to bits, ms to s before dividing.

Remember for the exam

- Tt is about the **packet and the link rate**; Tp is about the **distance and the medium**. They are independent.
- Throughput can never exceed bandwidth, and bandwidth in bits per second is not the same as speed of the signal.
- When a question says "ignore queuing and processing", latency = Tt + Tp.
- Check which delay a BDP question uses (one-way or RTT) from the wording. It matters for window-size questions later.
- For a very long link with a small packet, Tp dominates. For a short link with a big packet, Tt dominates.

Check yourself

1. A 8000-bit packet is sent at 1 Mbps. What is Tt? *(8000 / 10^6 = 8 ms.)*
2. A 3000 km cable carries signals at 2 × 10^8 m/s. What are Tp and RTT? *(Tp = 15 ms, RTT = 30 ms.)*
3. Upgrading the link from 10 Mbps to 100 Mbps changes which delay? *(Only Tt, which becomes ten times smaller; Tp is unchanged.)*
4. What is the BDP of a 100 Mbps link with RTT 20 ms? *(10^8 × 0.02 = 2,000,000 bits, or 250,000 bytes.)*

Do next

Pick two cities, estimate the fibre distance, and compute Tp and RTT; compare with a real `ping` result and explain the gap. The next topic uses Tt and Tp (the ratio a = Tp / Tt) to measure how well stop-and-wait uses a link.

---

21. Flow control and stop-and-wait

Chapter: 2. Data Link Layer · **Videos:** Flow Control (5 min) · Stop-and-Wait Protocol (7 min) · Stop-and-Wait ARQ Protocol (6 min)

Key ideas

- **Flow control** stops a fast sender from overwhelming a slower receiver whose buffer or processing capacity is limited. It is a data link layer job (and also a transport layer job).
- **Error control** is a separate job: detect damaged or lost frames and get them resent. **ARQ** (automatic repeat request) protocols combine flow control and error control.
- **Stop-and-wait** is the simplest flow control scheme. The sender transmits one frame, then waits for an **acknowledgement (ACK)** before sending the next.
- Over an error-free channel it works, but the sender is idle for most of each cycle when the link is long or fast.
- **Stop-and-wait ARQ** adds the machinery needed for real channels:
  - the sender starts a **timeout timer** after every frame and keeps a copy;
  - if no ACK arrives before the timeout, it **retransmits** the frame;
  - frames carry a **sequence number**, 0 and 1 alternately, so the receiver can recognise duplicates.
- Cases: a **lost frame** is fixed by the timeout. A **damaged frame** is discarded by the receiver (failed error check), and the timeout fixes it. A **lost ACK** makes the sender resend a frame the receiver already has; the sequence number lets the receiver discard the duplicate and ACK again.
- The ACK normally names the **next frame expected**: after receiving frame 0, the receiver sends ACK 1.
- The timeout must be longer than the round trip time, otherwise the sender resends frames that were fine and the link carries needless duplicates.

Formulas and methods

Let Tt be the frame transmission time and Tp the one-way propagation delay. Ignore the ACK's transmission time, queuing and processing.

- Cycle time for one frame = Tt + 2·Tp (send the frame, wait a round trip for the ACK).
- **Efficiency (utilisation)** η = Tt / (Tt + 2·Tp) = 1 / (1 + 2a), where a = Tp / Tt.
- **Throughput** = η × bandwidth.
- If each frame is lost or damaged with probability p, effective efficiency falls to about η × (1 – p).

Worked example. Bandwidth 2 Mbps, frame 4000 bits, distance 600 km, speed 2 × 10^8 m/s.

- Tt = 4000 / (2 × 10^6) = 2 ms. Tp = 600,000 / (2 × 10^8) = 3 ms. So a = 1.5.
- η = 1 / (1 + 2 × 1.5) = 1/4 = 25%.
- Throughput = 0.25 × 2 Mbps = 0.5 Mbps, which is one 4000-bit frame every 8 ms (2 + 6).

Remember for the exam

- Stop-and-wait efficiency is poor when **a is large** (long or fast links, small frames). Making the frame larger raises efficiency.
- Only **two sequence numbers** (0 and 1) are needed. Sender window size is 1, receiver window size is 1.
- Plain stop-and-wait does not handle loss. A timer and sequence numbers turn it into stop-and-wait ARQ.
- Efficiency does not depend on bandwidth directly, only through a (a grows when Tt shrinks).
- Utilisation formula assumes the ACK has negligible transmission time unless the question states otherwise.

Check yourself

1. What do the timer and the sequence number each do in stop-and-wait ARQ? *(The timer triggers retransmission after a lost frame or lost ACK; the sequence number lets the receiver spot and discard duplicates.)*
2. If Tt = 1 ms and Tp = 0.25 ms, what is the efficiency? *(a = 0.25, η = 1/1.5 = 66.7%.)*
3. A frame arrives intact but the ACK is lost. What does the receiver do when the frame is resent? *(Sees the old sequence number, discards the duplicate, and sends the ACK again.)*
4. Why is the timeout set above the RTT? *(An early timeout resends frames that were already received correctly.)*

Do next

Compute stop-and-wait efficiency for a 1500-byte frame over a 1 Gbps link of 100 km, and see how low it is. That gap is why the next topic sends many frames before waiting, using a sliding window.

---

22. Sliding window and Go-Back-N

Chapter: 2. Data Link Layer · **Videos:** Sliding Window Protocol (8 min) · Go-Back-N ARQ (11 min) · Go-Back-N ARQ (Solved problem 1) (15 min) · Go-Back-N ARQ (Solved problem 2) (8 min)

Key ideas

- Stop-and-wait wastes the link while waiting for each ACK. **Sliding window** protocols let the sender transmit several frames before the first ACK returns, keeping the link busy. This is **pipelining**.
- The **window size** is the number of frames that may be outstanding (sent but not yet acknowledged). Frames carry **sequence numbers**, and the window **slides** forward as ACKs arrive.
- With **k-bit sequence numbers**, numbers run from 0 to 2^k – 1 and then wrap around.
- **Go-Back-N (GBN) ARQ** has a sender window of N frames and a receiver window of **1**: the receiver accepts frames only in order.
- If a frame is lost or damaged, the receiver **discards** it and every later frame (they arrive out of order). The sender's timer expires and it **goes back** to the lost frame and resends it and all frames after it.
- GBN uses **cumulative acknowledgement**: an ACK that names frame n (or "next expected n") confirms everything before it, so a lost ACK is often covered by a later one.
- The **maximum sender window is 2^k – 1**. With window 2^k, a lost-ACK situation would let a retransmitted frame look like a new one.
- In a GBN timeline, the sender window's left edge moves only when an ACK arrives, and the sender stops when the window is full.
- The cost of GBN is wasted bandwidth: one lost frame in a window of N may cause up to N frames to be resent, even though most were fine.

Formulas and methods

- Stop-and-wait is the special case N = 1.
- **Efficiency** with window N (error-free): $\eta = \min(1, N / (1 + 2a))$, $a = T_p / T_t$.
- **Window needed for 100% utilisation:** $N \geq 1 + 2a$, rounded up.
- **Sequence bits needed** for GBN: $2^k \geq N + 1$, so $k = \text{ceil}(\log_2(N + 1))$.

Worked example. T_t = 1 ms, T_p = 9 ms, so $a = 9$ and $1 + 2a = 19$.

- With N = 7: $\eta = 7/19$, about 36.8%.
- To keep the link full, N = 19. Then $2^k \geq 20$, so k = 5 (numbers 0 to 31; maximum GBN window with 5 bits is 31, enough).
- Loss scenario: the sender transmits frames 0 to 5 and frame 3 is lost. The receiver accepts 0, 1, 2, then discards 4 and 5. After the timeout the sender resends 3, 4 and 5: three transmissions to repair one loss.

Why N ≤ 2^k – 1. Take k = 2 and N = 4. The sender sends 0, 1, 2, 3 and all arrive, but every ACK is lost. After the timeout it resends 0. The receiver is now expecting 0 as a new frame, so it accepts the old frame as new data. With N = 3, the receiver after frames 0, 1, 2 expects 3, so a resent 0 is recognised as old and discarded.

Remember for the exam

- GBN: sender window N, receiver window 1. Max N = 2^k – 1.
- The receiver in GBN does **not** buffer out-of-order frames.
- Retransmission in GBN covers the lost frame and everything sent after it, not just the lost one.
- "Go back 3" means the sender window is 3.
- The same formula N / (1 + 2a) is used for GBN and selective repeat when there are no errors.
- Sequence number bits k is decided by the window size, not the other way round, in most questions: window N needs k = ceil(log2(N + 1)).

Check yourself

1. How many sequence-number bits does a GBN sender need for window 15? *(2^k ≥ 16, so k = 4.)*
2. What is the largest GBN window with 3-bit sequence numbers? *(7.)*
3. Tt = 2 ms, Tp = 11 ms, N = 6. What is the efficiency, and what window gives 100%? *(a = 5.5, 1 + 2a = 12; η = 6/12 = 50%; window 12.)*
4. The sender sent frames 0 to 4 and frame 1 was lost. Which frames are resent? *(1, 2, 3 and 4.)*

Do next

Draw a Go-Back-N timeline for window 4 with one lost frame and one lost ACK, then count the transmissions. The next topic fixes GBN's waste by letting the receiver buffer frames and asking for only the missing one.

---

23. Selective repeat and window sizes

Chapter: 2. Data Link Layer · **Videos:** Selective Repeat ARQ (5 min) · Selective Repeat ARQ (Solved problem 1) (5 min) · Selective Repeat ARQ (Solved problem 2) (4 min) · Sliding Window Protocol (Solved problem 1) (4 min) · Sliding Window Protocol (Solved problem 2) (4 min)

Key ideas

- **Selective Repeat (SR) ARQ** removes Go-Back-N's waste. The receiver has a window of size N too, accepts frames that arrive **out of order** inside that window, and **buffers** them.
- Each frame is acknowledged **individually**. The sender keeps a **timer per frame** and, when one expires, resends **only that frame**.
- The receiver delivers data to the upper layer in order: it waits until the missing frame arrives, then passes up the whole run of buffered frames.
- A **negative acknowledgement (NAK)** may be used to ask for a missing frame earlier, before the sender's timer runs out.
- SR trades extra work for efficiency: the receiver needs **buffer space and sorting logic**, and the sender needs **several timers**. It uses the link better than GBN when errors are common.
- **Window limit:** with k-bit sequence numbers, there are 2^k distinct numbers. The sender window plus the receiver window must fit without overlap, so **Ws + Wr ≤ 2^k**. With equal windows, **N ≤ 2^(k–1)**.
- The reason: if the windows were larger, after the receiver slides forward, a **retransmitted old frame** could carry a number that falls inside the new window and would be accepted as new data.
- Compare: GBN needs sender window ≤ 2^k – 1 with receiver window 1, so Ws + Wr = 2^k.
- The sliding window problems in this topic ask for (a) minimum sequence bits for a given window, (b) maximum window for given bits, or (c) the window or efficiency for a given link.

Formulas and methods

	Go-Back-N	Selective Repeat
Sender window	up to 2^k – 1	up to 2^(k–1)
Receiver window	1	same as sender
Sequence numbers needed for window N	N + 1	2N
Bits needed	ceil(log2(N + 1))	ceil(log2(2N))
Ack type	cumulative	individual
On loss	resend lost frame and all after	resend only lost frame

Efficiency without errors for both is η = min(1, N / (1 + 2a)), a = Tp / Tt.

Worked example. Tt = 0.5 ms, Tp = 4.75 ms, so a = 9.5 and 1 + 2a = 20.

- A window of 20 gives 100% utilisation. A window of 10 gives 10/20 = 50%.
- SR with N = 20 needs 2N = 40 sequence numbers: 2^5 = 32 is too few, so k = 6.
- GBN with N = 20 needs N + 1 = 21 numbers, so k = 5 is enough (32 ≥ 21).

Same link, same utilisation, but SR needs one more header bit and buys better recovery from losses.

Remember for the exam

- Maximum windows with k bits: **GBN 2^k – 1, SR 2^(k–1)**. Do not swap them.
- If a question gives the sender and receiver windows separately, use Ws + Wr ≤ 2^k for SR.
- GBN has no receiver buffering beyond one frame; SR buffers out-of-order frames.
- With no errors, GBN and SR have the same efficiency. Their difference appears only when frames are lost.
- If a question says "sequence numbers 0 to 15", then k = 4.

Check yourself

1. Largest SR window with 4-bit sequence numbers? *(2^3 = 8.)*
2. How many bits does SR need for window 12? *(2N = 24 numbers, so k = 5.)*
3. In SR, frame 2 of frames 0 to 5 is lost. How many frames are resent? *(One, frame 2; the receiver buffered 3, 4, 5.)*
4. Tt = 1 ms, Tp = 7 ms, N = 5. Efficiency? *(a = 7, 1 + 2a = 15, η = 5/15 = 33.3%.)*

Do next

Work out, for three different links, the window needed for 100% utilisation and the sequence bits needed by GBN and by SR; practise GATE previous-year questions on sliding window. The next topic moves to links shared by many stations, where the question is who gets to send.

---

24. Multiple access and ALOHA

Chapter: 2. Data Link Layer · **Videos:** Multiple Access Protocols (8 min) · Pure Aloha (12 min) · Slotted Aloha (9 min)

Key ideas

- On a dedicated point-to-point link, the data link layer needs no sharing rules. On a **shared medium** many stations compete, and if two transmit together their frames **collide** and are lost. **Multiple access protocols** decide who may send and when.
- Three families:
  - **Random access:** no station is in charge and all have equal priority; any station may send when it has data, and collisions are possible. ALOHA and CSMA belong here.
  - **Controlled access:** stations take turns under some arrangement, such as reservation, polling or token passing.
  - **Channelization:** the channel is split among stations by frequency (FDMA), time (TDMA) or code (CDMA).
- **Pure ALOHA:** a station sends whenever it has a frame. It waits for an acknowledgement; if none arrives within a timeout, it assumes a collision, waits a **random backoff time**, and sends again. The random wait prevents the same stations colliding again and again.
- **Vulnerable time** is the window during which another transmission would destroy a frame. In pure ALOHA a frame of duration Tfr is hit by any other frame started in the Tfr before or the Tfr after it, so vulnerable time = **2 × Tfr**.
- **Slotted ALOHA:** time is divided into slots equal to one frame time, and stations may start only at a slot boundary. Frames either collide completely or not at all, so vulnerable time = **Tfr**. This halves the collision window but needs clock synchronisation.
- **G** is the average number of frames generated by all stations per frame time (offered load). **S** is the throughput: the average number of successful frames per frame time.
- Pure ALOHA's best throughput is about 18.4%. Slotted ALOHA's best is about 36.8%.

Formulas and methods

- Pure ALOHA: **S = G · e^(–2G)**, maximum at G = 0.5, S = 1/(2e) ≈ 0.184.
- Slotted ALOHA: **S = G · e^(–G)**, maximum at G = 1, S = 1/e ≈ 0.368.
- Frames per second successfully delivered = S × (1 / Tfr).

Worked example. Frame time Tfr = 2 ms, so 500 frame times per second.

- Pure ALOHA at G = 0.5: S = 0.5 × e^(–1) = 0.184. Successful frames = 0.184 × 500 = about 92 per second.
- Slotted ALOHA at G = 2: S = 2 × e^(–2) = 0.271, about 135 frames per second. Pushing load past G = 1 makes slotted ALOHA worse, not better.
- Pure ALOHA at G = 1: S = e^(–2) = 0.135.

Remember for the exam

- Vulnerable time: pure **2Tfr**, slotted **Tfr**.
- Maximum efficiency: pure **18.4%** (G = 0.5), slotted **36.8%** (G = 1).
- Both are random access, with no carrier sensing: stations do not listen before transmitting.
- Slotted ALOHA's gain comes from synchronisation; collisions are still possible when two stations pick the same slot.
- Throughput rises with load up to the maximum, then falls as collisions dominate.
- Controlled access and channelization avoid collisions altogether but cost coordination or capacity.

Check yourself

1. Vulnerable time for a pure ALOHA frame of 5 ms? *(10 ms.)*
2. Pure ALOHA with G = 1.5: what is S? *(1.5 × e^(–3) = 1.5 × 0.0498 = 0.075.)*
3. Why does slotted ALOHA double the maximum throughput? *(Frames can only overlap completely, so the vulnerable time halves.)*
4. Name two controlled-access methods. *(Any two of reservation, polling, token passing.)*

Do next

Plot S against G for both protocols using a spreadsheet to see the two peaks. Next, CSMA reduces collisions by making stations listen before they send.

---

25. CSMA, CSMA/CD and CSMA/CA

Chapter: 2. Data Link Layer · **Videos:** Carrier Sense Multiple Access (CSMA) – Part 1 (13 min) · Carrier Sense Multiple Access (CSMA) – Part 2 (13 min)

Key ideas

- **CSMA (carrier sense multiple access)** follows the rule "listen before you talk": a station senses the channel and sends only if it looks idle. This lowers collisions compared with ALOHA.
- Collisions still happen because of **propagation delay**: a station that has just started sending may not yet be heard at the far end, so another station senses idle and starts too. The vulnerable time is therefore the propagation time, Tp.
- The three CSMA behaviours differ in what a station does after sensing:
  - **1-persistent:** keep sensing a busy channel and send **immediately** (probability 1) once it goes idle. Simple, but if two stations are waiting, they collide for certain.
  - **Non-persistent:** if the channel is busy, do not keep sensing; **wait a random time** and sense again. Fewer collisions, but longer delay and some idle time on the channel.
  - **p-persistent:** for **slotted** channels. When idle, send with probability p; with probability 1 – p wait for the next slot and sense again.
- **CSMA/CD (collision detection)** is used on wired Ethernet. The station keeps listening while sending. If it detects a collision, it stops, sends a short **jam signal** so everyone knows, then waits a random time chosen by **binary exponential backoff** and tries again.
- For CD to work, the sender must still be transmitting when the collision signal returns, so the frame must be long enough: **Tt ≥ 2 × Tp**.
- A CSMA/CD channel cycles through **transmission**, **contention** (stations colliding and backing off) and **idle** periods. The video's efficiency formula (given without derivation) shows why it suits LANs: efficiency falls as the cable gets longer and rises as frames get longer.
- **CSMA/CA (collision avoidance)** is used on wireless LANs. Radios cannot listen while sending, and a distant station may be out of range (hidden terminal). So the station waits an interframe space, counts down a random backoff, and relies on **acknowledgements** (optionally RTS/CTS) to confirm delivery.

Formulas and methods

- Collision detection condition: **Tt ≥ 2 × Tp**, where Tt = L / B.
- Minimum frame length: **L_min = 2 × Tp × B**.
- CSMA/CD efficiency: **η = 1 / (1 + 6.44a)**, where a = Tp / Tt. For a = 0.1, η = 1 / 1.644, about 61%.
- Binary exponential backoff: after the n-th collision, pick K at random from 0 to 2^n – 1 and wait K slot times. In Ethernet, n stops growing at 10, and the frame is dropped after 16 attempts.

Worked example. Bandwidth 100 Mbps, cable 2 km, speed 2 × 10^8 m/s.

- Tp = 2000 / (2 × 10^8) = 10 microseconds. 2Tp = 20 microseconds.
- L_min = 100 × 10^6 × 20 × 10^-6 = 2000 bits (250 bytes).

- If the cable length is doubled to 4 km, minimum frame size doubles to 4000 bits. If the bandwidth rises tenfold, the minimum also rises tenfold.

Classic 10 Mbps Ethernet uses a 51.2 microsecond slot time, which gives the familiar 512-bit (64-byte) minimum frame.

Remember for the exam

- CSMA reduces collisions but cannot eliminate them; propagation delay is the cause.
- Persistence summary: 1-persistent sends at once when idle, non-persistent backs off when busy, p-persistent sends with probability p in slots.
- CSMA/CD: wired, detects collisions. CSMA/CA: wireless, avoids them.
- Min frame = 2·Tp·B. Longer cable or faster link means a longer minimum frame.
- CSMA/CD efficiency 1 / (1 + 6.44a) drops as distance (Tp) grows, so CSMA/CD is a LAN method, not a WAN one.
- Backoff window doubles after each collision: 0 to 1, then 0 to 3, then 0 to 7, and so on.

Check yourself

1. Which CSMA variant makes two waiting stations collide for sure? *(1-persistent.)*
2. Why can't wireless LANs use collision detection? *(A radio cannot sense a collision while transmitting, and hidden stations are out of range.)*
3. 10 Mbps, 1 km, speed 2 × 10^8 m/s. Minimum frame? *(Tp = 5 microseconds, 2Tp = 10 microseconds, L = 100 bits.)*
4. After the 3rd collision, what range does K come from? *(0 to 7.)*

Do next

Recompute the minimum frame size for three different cable lengths and bandwidths and note the pattern; practise GATE previous-year questions on CSMA/CD frame size and ALOHA throughput. The next topic builds on these ideas with the link-layer technologies that use them.

---

26. Controlled access and channelization

Chapter: 2. Data Link Layer · **Videos:** Reservation (5 min) · Polling (8 min) · Token passing (7 min) · Channelization protocols (11 min)

Key ideas

- Multiple access protocols fall into three families: **random access** (ALOHA, CSMA), **controlled access** and **channelization**. Controlled access and channelization are designed so that stations do not collide.
- In **controlled access**, a station may send only when it has been given permission. The three methods are reservation, polling and token passing.
- **Reservation:** time is cut into intervals. Each interval starts with a **reservation frame** that has one **mini-slot per station**, so N stations means N mini-slots. A station that wants to send sets a 1 in its own mini-slot. After the reservation frame, the stations that reserved send their data frames in order. The next interval starts with a new reservation frame.
- **Polling:** one station is the **primary** (master) and the others are **secondaries**. The primary invites each secondary in turn to send up to a set number of frames. Two functions are used: **poll** (primary asks "do you have anything to send?") and **select** (primary tells a secondary "get ready to receive").
- Polling has two drawbacks: the **polling delay** before each station gets a turn, and a **single point of failure**, because the whole channel stops if the primary fails.
- **Token passing:** stations form a **logical ring**. A small control frame called the **token** circulates. Only the station holding the token may transmit. When it is done (or its holding time runs out), it passes the token to the next station. The standard examples are Token Ring (IEEE 802.5) and Token Bus (IEEE 802.4). Practical problems are a lost token, a duplicated token and a failed station breaking the ring.
- **Channelization** shares the link's bandwidth among stations by frequency, time or code: **FDMA**, **TDMA** and **CDMA**.
- **FDMA** gives each station its own frequency band, with unused **guard bands** between bands to stop overlap. All stations transmit at the same time, each in its own band.
- **TDMA** gives the whole bandwidth to one station at a time, in turns (time slots).
- **CDMA** lets all stations use the whole bandwidth at the same time. Each station has a **code** (a chip sequence). Codes are chosen to be **orthogonal**: the product of two different codes is 0, and the product of a code with itself equals N, the number of stations.

Formulas and methods

Polling efficiency = Tt / (Tt + Tpoll), where Tt is the transmission time of the data and Tpoll is the polling overhead per turn.

Token passing efficiency (token released right after the frame is sent, one frame per turn) = N·Tt / (N·Tt + Tring), where Tring is the time for the token to go once around the ring.

Worked example, polling: a secondary sends one 2 ms frame, and each poll costs 0.5 ms. Efficiency = 2 / (2 + 0.5) = 0.8, or 80%.

Worked example, token passing: N = 10 stations, Tt = 1 ms, Tring = 2 ms. Efficiency = 10 / (10 + 2) = 0.833.

FDMA bandwidth: user band = (total bandwidth – guard bands) / N. For 1 MHz shared by 4 stations with a 20 kHz guard band between neighbours (3 guards): (1000 – 60) / 4 = 235 kHz each.

CDMA with four stations, codes c1 = [+1 +1 +1 +1], c2 = [+1 −1 +1 −1], c3 = [+1 +1 −1 −1], c4 = [+1 −1 −1 +1]. Let station 1 send +1, station 2 send −1, station 3 stay silent (0), station 4 send +1. The channel carries the sum: [1 1 1 1] + [−1 1 −1 1] + [0 0 0 0] + [1 −1 −1 1] = [1 1 −1 3]. To recover station 2's bit, take the inner product with c2: 1·1 + 1·(−1) + (−1)·1 + 3·(−1) = −4, and −4 / 4 = −1. Station 3 gives 0 (silent). The receiver divides by N.

Remember for the exam

- Reservation has **N mini-slots for N stations**, and the reservation frame comes before the data of every interval.
- Polling and token passing are collision-free but add overhead; polling fails if the master fails, and a ring fails if the token is lost.
- FDMA wastes some bandwidth on guard bands. TDMA shares by time. CDMA shares both time and frequency, and separates users by code.
- Orthogonal codes: different codes give 0, a code with itself gives N.
- FDM and TDM are physical-layer multiplexing techniques; FDMA, TDMA and CDMA are the MAC-layer access methods built on the same ideas.

Check yourself

1. A network has 8 stations. How many mini-slots are in each reservation frame? *(8, one per station.)*
2. A secondary sends a 5 ms frame and each poll takes 1 ms. What is the polling efficiency? *(5 / 6 = 0.833.)*
3. What happens to a token ring if the token is lost? *(No station may transmit, so the ring needs a monitor or a recovery procedure to create a new token.)*
4. A 600 kHz channel is divided into 5 FDMA bands with 4 guard bands of 10 kHz each. Width of one band? *((600 – 40) / 5 = 112 kHz.)*

Do next

Work out two more CDMA decodings by hand with the codes above, including one where two stations send at once. Then watch the Ethernet videos, which return to random access and apply CSMA/CD on real wired LANs. Practise GATE previous-year questions on polling and token ring efficiency.

---

27. Ethernet

Chapter: 2. Data Link Layer · **Videos:** Ethernet (13 min) · Solved questions, part 1 (14 min) · Solved questions, part 2 (8 min)

Key ideas

- **Ethernet** is the most widely used wired LAN technology. It covers the physical layer and the data link layer, and is defined by the IEEE 802.3 family (with 802.2 for the LLC sub-layer). At the data link layer it uses both the LLC and MAC sub-layers.
- Speeds grew over time: **standard Ethernet** 10 Mbps, **Fast Ethernet** 100 Mbps, **Gigabit Ethernet** 1 Gbps, then 10, 40 and 100 Gbps.
- **Frame format** (in order): **preamble** 7 bytes (alternating 1s and 0s, for clock synchronisation), **SFD** 1 byte (10101011, marks the start of the frame), **destination address** 6 bytes, **source address** 6 bytes, **type/length** 2 bytes, **data and padding** 46 to 1500 bytes, **CRC** 4 bytes (error detection).
- The preamble and SFD are added by the physical layer, so frame-size calculations normally leave them out.
- **Payload range:** minimum 46 bytes, maximum 1500 bytes (the MTU). If the upper layer gives less than 46 bytes, the data link layer **pads** it.
- **Minimum frame** = 46 + 6 + 6 + 2 + 4 = **64 bytes (512 bits)**. **Maximum frame** = 1500 + 18 = **1518 bytes (12,144 bits)**. The 18 bytes are the two addresses, type/length and CRC.
- An **Ethernet (MAC) address** is 48 bits, written as 6 bytes or 12 hex digits, usually separated by colons.
- The lowest-order bit of byte 1 tells the type: 0 means **unicast**, 1 means **multicast**. All 48 bits set to 1 (`FF:FF:FF:FF:FF:FF`) is **broadcast**; a switch floods it to every port in the LAN.
- A **multicast or broadcast address can only be a destination**. A frame that carries one in the source field is invalid and is discarded.
- Each byte of an address is put on the wire **least significant bit first**.
- The solved questions practise these rules: converting hex to binary, classifying addresses, working out bit order on the wire, and spotting invalid addresses. Frame-size, transmission-time and efficiency problems use the sizes above.

Formulas and methods

Hex to binary: each hex digit is 4 bits (8-4-2-1). 3C is 0011 1100.

Classify a destination address: write the first byte in binary and look at its last bit. Then check if every digit is F.

- `3C:...`: first byte 0011 1100, last bit 0, so unicast.
- `5D:...`: first byte 0101 1101, last bit 1, so multicast.
- `FF:FF:FF:FF:FF:FF`: broadcast.

Bit order on the wire: reverse the 8 bits of each byte. For `1A` = 0001 1010 the line carries 0 1 0 1 1 0 0 0. Bytes still go left to right.

Frame size and payload efficiency (counting preamble and SFD as 8 bytes of overhead): efficiency = payload / (payload + 18 + 8). For a 1000-byte payload: 1000 / 1026 = 0.975, about 97.5%. For a 20-byte payload, padding raises it to 46 bytes, so the frame is 64 bytes and only 20 of 72 bytes on the wire are useful.

Transmission time = bits / bandwidth. A maximum 1518-byte frame at 100 Mbps: 12,144 / 10^8 = 121.44 microseconds.

Remember for the exam

- Minimum frame **64 bytes**, minimum payload **46 bytes**, maximum frame **1518 bytes**, maximum payload **1500 bytes**. Know which numbers include the 18 bytes of header and trailer and which include the 8 bytes of preamble and SFD.
- Unicast versus multicast is the **last bit of the first byte**, not the first bit you read on the page.
- Broadcast is all ones; multicast and broadcast never appear as a source.
- The type/length field is 2 bytes. Values up to 1500 mean length; larger values (0x0600 and up) mean the type of the upper-layer protocol.
- Ethernet gives no acknowledgements at this layer; it detects errors with the CRC and drops bad frames.

Check yourself

1. How many bytes of padding go into a frame carrying a 30-byte payload? *(16, to reach 46.)*
2. Is `A3:...` unicast or multicast? *(A3 = 1010 0011, last bit 1: multicast.)*
3. What is the maximum payload efficiency of Ethernet, counting preamble and SFD? *(1500 / 1526 = 98.3%.)*
4. How many bits is a minimum-size frame, and how long does it take at 10 Mbps? *(512 bits; 51.2 microseconds.)*

Do next

Take any three MAC addresses from your own devices (`ipconfig /all` or `ip link`) and classify them. Then continue with how Ethernet shares the cable: transmission rules, collisions and the minimum frame size. Practise GATE previous-year questions on Ethernet frame size and MAC addresses.

---

28. Ethernet backoff and solved questions

Chapter: 2. Data Link Layer · **Videos:** Ethernet transmitter algorithm (12 min) · Experience with Ethernet (6 min) · Solved question 1 (3 min) · Solved question 2 (5 min) · Solved question 3 (4 min)

Key ideas

- The Ethernet MAC algorithm runs in the hardware of the **network adapter** (NIC). The access method is **CSMA/CD** (collision detection) and the classic line code is **Manchester encoding**.
- **Transmit rule:** if the line is idle, send at once. If it is busy, wait until it becomes idle and then send immediately. Because the adapter sends with probability 1 as soon as the line frees up, Ethernet is **1-persistent**.
- There is no central control, so two adapters can start together, and then their frames **collide**.
- When an adapter detects a collision, it sends a **32-bit jam signal** so that everybody notices, then stops. A transmitter therefore sends at least 64 bits of preamble plus 32 bits of jam, which is **96 bits**, in a collision.
- A **runt frame** is a frame shorter than 64 bytes. Collisions are the usual cause; a faulty NIC, a buffer under-run or a duplex mismatch (one end half duplex, the other full duplex) can also make them.
- **Worst-case detection time:** host A sends at time t. The signal reaches B, at the far end, at t + Tp. Just before it arrives, B starts sending and a collision happens at B. B's damaged signal reaches A at t + 2Tp. So A must keep transmitting for at least **2Tp** to be sure of seeing any collision. This is why the minimum frame size exists: **Tt >= 2Tp**.
- **Binary exponential backoff:** after the n-th collision of the same frame, the adapter picks K at random from **0 to 2^n – 1** and waits **K slot times**. A slot time is **512 bit times**: 51.2 microseconds at 10 Mbps. The exponent stops growing at 10, and the adapter gives up (reports an error) after **16 attempts**. The random range doubling keeps the collision chance falling as load rises.
- **Strengths of Ethernet:** cheap, simple to maintain, all stations are equal, tolerant of noise, and fast (10 Gbps and more).
- **Weaknesses:** collisions waste capacity under heavy load; no priorities, so it is poor for real-time traffic; padding small payloads to 46 bytes wastes bandwidth; no link-layer acknowledgement.
- **Cable categories** (twisted pair, 100 m per segment): Cat 3 10 Mbps, Cat 5 100 Mbps, Cat 5e 1 Gbps, Cat 6 1 Gbps (10 Gbps over short runs), Cat 6a and Cat 7 10 Gbps, Cat 8 up to 25 or 40 Gbps over short runs.
- The solved questions test statements about CSMA/CD (sensing, jam signal, stopping after a collision, backoff) and the **maximum cable length for a given frame size, speed and signal speed**.

Formulas and methods

Collision detection condition: Tt >= 2Tp, which gives L / B >= 2d / v.

- Minimum frame length: L >= 2 · B · d / v.
- Maximum distance: d <= L · v / (2B).

Worked example, maximum length: frame 12,000 bits, bandwidth 100 Mbps, signal speed 2 x 10^8 m/s. Tt = 12,000 / 10^8 = 120 microseconds, so Tp <= 60 microseconds, so d <= 2 x 10^8 x 60 x 10^-6 =

12,000 m = **12 km**.

Worked example, minimum frame: 100 Mbps, a 500 m segment, speed 2 x 10^8 m/s. Tp = 500 / (2 x 10^8) = 2.5 microseconds. L >= 2 x 2.5 x 10^-6 x 10^8 = **500 bits**.

Backoff range after n collisions: K in {0, ..., 2^n – 1}.

- After the 3rd collision: K from 0 to 7, eight values, longest wait 7 x 51.2 = 358.4 microseconds at 10 Mbps.
- If two stations collide for the third time, the chance they pick the same K again is 1/8.

Remember for the exam

- **CSMA/CD needs Tt >= 2Tp.** Use the round-trip time (2Tp), not Tp alone, and check the units (bits, bytes, km, m).
- The jam signal is for **telling others** a collision happened, not to pad short frames.
- A station that detects a collision **stops** sending data; it does not continue.
- Backoff makes collisions on retransmission **less likely**, not impossible. K is chosen from 0 to 2^n – 1 and the wait is K slot times.
- Ethernet is 1-persistent, has no link-layer acknowledgement and works at both layer 1 and layer 2.

Check yourself

1. After the 5th collision, how many values can K take? *(32, from 0 to 31.)*
2. A 10 Mbps Ethernet has a 1 km cable and signal speed 2 x 10^8 m/s. Minimum frame in bits? *(Tp = 5 microseconds, 2Tp = 10 microseconds, L = 100 bits; the real standard asks for 512 bits for far larger networks.)*
3. What is the maximum cable length for 8,000-bit frames, 200 Mbps, 2 x 10^8 m/s? *(Tt = 40 microseconds, Tp <= 20 microseconds, d <= 4 km.)*
4. Why does a station send a jam signal? *(So all stations notice the collision and discard the damaged frame.)*

Do next

Solve three distance and frame-size problems like the examples above, changing one number at a time. The next topic moves to the wireless LAN, where collisions cannot be detected and CSMA/CA is used instead. Practise GATE previous-year questions on CSMA/CD and Ethernet.

---

29. Wi-Fi (IEEE 802.11)

Chapter: 2. Data Link Layer · **Videos:** IEEE 802.11 Wi-Fi (11 min) · Distribution system (15 min) · Wi-Fi frame format (9 min)

Key ideas

- **IEEE 802.11 (Wi-Fi)** is the wireless LAN standard for homes, offices and campuses. Wired LAN siblings are Ethernet and Token Ring. The medium is the air, so anyone in range can hear a transmission, which is a security concern.
- A typical setup: client devices reach an **access point (AP)**, the AP connects by wire to a switch and router, and the router leads to the ISP. A **wireless router** combines AP, switch and router. Some APs use **Power over Ethernet (PoE)**, so one cable carries data and power.
- Wi-Fi cannot detect collisions while transmitting, so it uses **CSMA/CA** (collision avoidance), where Ethernet uses CSMA/CD.
- **Two modes: infrastructure mode**, where every client talks through an AP (centralised), and **ad hoc mode**, where devices talk to each other directly with no fixed infrastructure.
- **Bands:** 2.4 GHz has only **three non-overlapping channels** (1, 6 and 11); 5 GHz has many more non-overlapping channels. Standards: 802.11b is 2.4 GHz at 11 Mbps, 802.11a is 5 GHz at 54 Mbps, 802.11g is 2.4 GHz at 54 Mbps, 802.11n works on either band, 802.11ac uses 5 GHz, and 802.11ax (Wi-Fi 6) uses both bands. The quoted speeds are theoretical maximums.
- **Distribution system (DS):** the wired backbone that links the access points. Each AP serves a **cell**, like a base station in a mobile network. Nodes can move and the set of nodes they can reach changes over time.
- To send between two cells, a node sends to its own AP, the AP passes the frame over the DS to the destination's AP, and that AP transmits it to the destination.
- **Joining an AP is called scanning.** In **active scanning** the node sends a **probe frame**; APs in range answer with a **probe response**; the node picks one AP, sends an **association request**, and gets an **association response**. In **passive scanning** the APs periodically send **beacon frames** advertising themselves and their rates, and the node answers with an association request.
- When a node moves and a better AP appears, it joins the new AP (a **reassociation**), and the new AP tells the old AP through the DS.
- **Frame format:** frame control (2 bytes), duration (2), address 1 (6), address 2 (6), address 3 (6), sequence control (2), address 4 (6, only in AP-to-AP frames), data (0 to 2312), and FCS/checksum (4).
- **Frame control** has 11 subfields: version (2 bits, set to 0), type (2 bits: management, control or data), subtype (4 bits, for example RTS, CTS or ACK), **To DS**, **From DS**, **more fragments**, **retry**, **power management**, **more data**, **WEP/protected** and **order** (1 bit each).
- **Sequence control** is 16 bits: 4 bits fragment number and 12 bits sequence number. It detects duplicates and orders frames.

Formulas and methods

Address fields depend on To DS and From DS:

To DS	From DS	Address 1	Address 2	Address 3	Address 4
0	0	destination	source	BSSID	not used
0	1	destination	AP (BSSID)	source	not used
1	0	AP (BSSID)	source	destination	not used
1	1	receiving AP	sending AP	destination	source

Worked example: a laptop sends 1,400 bytes to a server on the wired network through its AP. To DS = 1, From DS = 0, so the addresses are: AP, laptop, server. The header is 2 + 2 + 6 + 6 + 6 + 2 = 24 bytes and the FCS is 4 bytes, so the frame is 1,400 + 28 = **1,428 bytes**. With four addresses the header is 30 bytes, and the largest possible MAC frame is 30 + 2312 + 4 = 2346 bytes.

Remember for the exam

- **Ethernet uses CSMA/CD; Wi-Fi uses CSMA/CA.** Wireless stations cannot hear a collision while sending.
- Only **three non-overlapping channels** exist in the 2.4 GHz band.
- Frame control is 2 bytes and sequence control is 2 bytes; maximum data field **2312 bytes**; the 802.11 address fields are 48-bit MAC addresses.
- **Address meaning changes with To DS and From DS**: memorise the table.
- Active scanning means the node probes; passive scanning means the AP sends beacons.

Check yourself

1. Which frame does a client send first in active scanning? *(A probe frame.)*
2. A frame goes from a phone to its AP. What are To DS and From DS? *(To DS = 1, From DS = 0.)*
3. How many bits are in the frame control field, and how many subfields? *(16 bits, 11 subfields.)*
4. How many bytes is the header of a three-address frame? *(24 bytes, plus 4 bytes of FCS.)*

Do next

Sketch a building with three APs, label the cells and the DS, and trace one frame between two phones in different cells. The next topic explains why Wi-Fi needs the RTS/CTS exchange, through the hidden and exposed terminal problems.

---

30. Hidden and exposed terminals, MACA

Chapter: 2. Data Link Layer · **Videos:** Hidden terminal problem (4 min) · Exposed terminal problem (4 min) · MACA (10 min) · Wi-Fi solved question (4 min)

Key ideas

- A wireless node reaches only the nodes inside its **coverage range**. Two nodes can both be in range of a third node while being out of range of each other. This causes two problems that carrier sensing alone cannot fix.
- **Hidden terminal problem:** A and C cannot hear each other, but B is in range of both. A and C each sense an idle channel and send to B at the same time, and the frames **collide at B**. Neither sender can detect it. A and C are "hidden" from each other.
- **Exposed terminal problem:** B is sending to A. C is in range of B, so C hears B and wrongly decides it must stay silent, even though C wants to send to D, which is out of B's range. C's transmission would not disturb A's reception. The result is wasted capacity, not a collision.
- Both problems are also called hidden/exposed node or station problems and happen only in wireless networks.
- **MACA** (Multiple Access with Collision Avoidance) is the idea behind 802.11's handshake. Before sending data, sender and receiver exchange short **control frames**, so that every nearby node learns a transmission is coming.
- The sender sends **RTS** (request to send), which carries a field telling how long it will hold the medium. If the receiver agrees, it replies with **CTS** (clear to send), which repeats that duration.
- **Rules for bystanders:**
  - A node that hears the **CTS** is near the receiver, so it must stay silent for the stated time.
  - A node that hears the **RTS but not the CTS** is near the sender only, so it is free to transmit.
  - A node that hears both stays silent.
- **Hidden terminal solved:** C cannot hear A's RTS, but it hears B's CTS and defers. The handshake prevents the collision at B.
- **Exposed terminal helped:** C hears B's RTS but not A's CTS, so it knows it can send to D. In practice, this part works less reliably than the hidden-terminal fix.
- **MACAW** (MACA for wireless) adds an **acknowledgement** after each frame, and nodes wait for it before sending again. In 802.11 a sender whose RTS gets no CTS assumes a collision and backs off using the same **exponential backoff** idea as Ethernet.
- The 802.11 mechanism that remembers the reserved time is the **network allocation vector (NAV)**, set from the duration field of overheard frames.

Formulas and methods

Duration field of an RTS = 3 x SIFS + CTS + data + ACK (the whole exchange after the RTS).

Worked example (assumed times): SIFS = 10 microseconds, CTS = 30 microseconds, data = 200 microseconds, ACK = 30 microseconds. Duration = 30 + 30 + 200 + 30 = **290 microseconds**. A bystander that hears the RTS or CTS sets its NAV to this value and keeps silent until it expires.

Method for a topology question: (1) draw each node's range as a circle; (2) list who can hear whom; (3) for any sender/receiver pair, mark nodes that hear the RTS only, the CTS only, both, or neither; (4) apply the bystander rules above.

Remember for the exam

- **Hidden terminal: collision at the receiver. Exposed terminal: wasted opportunity, no collision.**
- RTS/CTS is a **collision avoidance** mechanism, not collision detection.
- 802.11 uses CSMA/CA with acknowledgements for **unicast** frames; broadcast and multicast frames are not acknowledged.
- On a statement-style question about 802.11: the 2.4 GHz band has three non-overlapping channels, RTS/CTS avoids collisions, and unicast frames are acknowledged.
- Ethernet uses CSMA/CD and so cannot be used as the model for Wi-Fi: a radio cannot listen while it transmits.

Check yourself

1. A, B, C are in a line and A and C are 150 m apart, out of each other's range. Who are hidden from each other? *(A and C, with B as the receiver in the middle.)*
2. In the exposed terminal case, which control frame does the exposed node hear? *(The RTS from the sender but not the CTS from its receiver.)*
3. What does a node do if it hears a CTS addressed to someone else? *(It stays silent for the duration in the CTS.)*
4. How does a sender know its RTS collided? *(No CTS comes back, so it backs off and retries.)*

Do next

Draw the hidden and exposed layouts on paper and walk through the RTS/CTS exchange for each. Then continue to Bluetooth, a short-range wireless technology with a very different, master-controlled access method. Practise GATE previous-year questions on Wi-Fi and CSMA/CA.

---

31. Bluetooth

Chapter: 2. Data Link Layer · **Videos:** IEEE 802.15.1 Bluetooth (10 min) · Bluetooth protocol stack (7 min)

Key ideas

- **Bluetooth** (IEEE 802.15.1) is a short-range wireless technology for a **personal area network (PAN)**. It sends data and voice between fixed and mobile devices without the Internet. It uses the **2.4 GHz ISM band** (industrial, scientific and medical), about 2.400 to 2.485 GHz.
- It was first imagined as a wireless replacement for RS-232 serial cables. The name comes from the Danish king Harald Bluetooth, who united tribes.
- The **Bluetooth SIG** (special interest group) defines the whole protocol suite, up to the application level. The application-level specifications are called **profiles** (for example, one to synchronise a PDA with a PC, one that connects a laptop to a wired LAN).
- The basic network is a **piconet**: **one master and up to seven active slaves**. All communication goes between the master and a slave; **slaves never talk directly to each other**.
- A slave can be **parked**, which is an inactive low-power state, to save battery.
- Several piconets that overlap form a **scatternet**. One device can be the master in one piconet and a slave in another, which links them.
- Radio detail: Bluetooth uses **frequency hopping** across 79 channels of 1 MHz (classic Bluetooth) at 1600 hops per second, with time slots of 625 microseconds. The master sends in even slots and slaves reply in odd slots.
- **Advantages:** cheap, easy, sets up an ad hoc connection quickly, works through walls, carries voice and data. **Disadvantages:** short range (about 10 m for the common class, up to about 100 m for the high-power class), low data rate, weaker security.
- **Bluesnarfing** is an attack that copies information (contacts, calendar, messages) from a device through an unauthorised Bluetooth connection.
- **Protocol stack:**
  - **Radio** layer: modulation and demodulation, the transceiver's physical characteristics.
  - **Baseband** (link) layer: sets up connections inside a piconet and defines the physical links (connectionless and connection-oriented).
  - **Link manager protocol (LMP):** manages existing links, authentication and encryption.
  - **L2CAP** (logical link control and adaptation): the layer between upper and lower layers; it adapts packets, segments and reassembles them, and multiplexes. The video calls it the heart of the stack.
  - Upper protocols: **RFCOMM** (serial-port emulation, used by WAP and OBEX), **TCS** (telephony control), **SDP** (service discovery), then the **applications/profiles**.

Formulas and methods

Hop rate = 1 / slot time. With 625 microsecond slots: 1 / 625 x 10^-6 = **1600 hops per second**.

Master-slave exchange time: the master sends in one slot and the slave answers in the next, so one round trip uses 2 slots = 2 x 625 = **1.25 ms**.

Piconet size: 1 master + 7 active slaves = **8 active devices**. Parked devices can be many more but cannot transmit until unparked.

Worked example: two piconets are full (each has 1 master and 7 slaves) and one phone is a slave in both. The phone links them into one scatternet. Counting devices: 2 x 8 – 1 = **15 distinct devices**, because the phone is counted once.

Remember for the exam

- A piconet has **one master and at most seven active slaves**.
- Slaves **cannot communicate directly**; the master relays.
- A device can be master of one piconet only, but may be a slave in another; the shared device ties piconets into a scatternet.
- Bluetooth is a PAN; it works at 2.4 GHz and uses frequency hopping.
- L2CAP sits above the baseband and link manager and below the upper protocols.

Check yourself

1. How many devices can transmit in one piconet at a time (master included)? *(8 active devices: master plus 7 slaves.)*
2. Can two slaves in the same piconet exchange data directly? *(No; they go through the master.)*
3. Which layer handles authentication and encryption? *(The link manager protocol.)*
4. What is the name of the attack that steals contacts over Bluetooth? *(Bluesnarfing.)*

Do next

List every Bluetooth device you own and draw the piconet each pair would form. The next topic returns to switched LANs and shows how one switch can be split into separate logical networks (VLANs).

---

32. VLANs

Chapter: 2. Data Link Layer · **Videos:** Virtual LAN (VLAN) (11 min)

Key ideas

- Without VLANs, five separate LANs need five switches. A **VLAN (virtual LAN)** divides one physical switch into several **logical** LANs by grouping its ports. Think of it as splitting the switch in software, not by cutting it.
- Each VLAN is its own **broadcast domain**, normally with its own IP subnet. A broadcast sent in one VLAN reaches only the ports of that VLAN.
- Ports that you do not assign belong to the **default VLAN** (VLAN 1 on many switches).
- Different VLANs are isolated at layer 2. For hosts in different VLANs to talk, traffic must pass through a **router** (or a layer 3 switch). Different VLANs use different IP networks, so this is routing, not switching.
- The hosts do not know which VLAN they are in. Only the switch keeps the port-to-VLAN mapping.
- A **trunk port** (trunk link) carries the traffic of several VLANs, for example between two switches or from a switch to a router. Ports that belong to one VLAN only are **access ports**.
- **Benefits:** better security (broadcasts stay inside the VLAN), lower cost (fewer switches), better performance (smaller broadcast domains), flexible grouping of users, and simpler management.
- **Types of VLAN** named in the course: data, default, native, management and voice VLANs.
- **Frame tagging:** on a trunk, the switch must know which VLAN a frame belongs to, so it inserts a **tag** into the Ethernet frame. The tag goes between the source address and the type/length field. The standard is **IEEE 802.1Q**. The switch adds the tag when it puts the frame on a trunk and removes it before sending the frame out of an access port. A tagged frame can cross any number of switches and still reach the right VLAN.

Formulas and methods

802.1Q tag = 4 bytes:

Field	Size	Purpose
TPID	16 bits	value 0x8100, marks the frame as tagged
Priority (PCP)	3 bits	class of service
DEI	1 bit	drop eligibility
VLAN ID (VID)	12 bits	which VLAN

With 12 bits there are 2^12 = 4096 values. IDs 0 and 4095 are reserved, so **4094 VLANs** are usable. A maximum tagged Ethernet frame is 1518 + 4 = **1522 bytes**.

Worked example: a 24-port switch has VLAN 10 on ports 1 to 8, VLAN 20 on ports 9 to 16, and ports 17 to 24 unassigned (default VLAN). A host on port 3 sends a broadcast. It is delivered to the other 7 ports of VLAN 10 (ports 1, 2, 4 to 8), and to none of the ports in VLAN 20 or the default VLAN. The switch therefore has 3 broadcast domains. To let a host on port 3 reach one on port 12, add a router link (a trunk, or one access port per VLAN) so a router can move packets between the two subnets.

Remember for the exam

- A VLAN is a **layer 2** broadcast domain; **one VLAN = one broadcast domain = usually one IP subnet**.
- **Inter-VLAN traffic needs a layer 3 device.**
- **802.1Q** tag: 4 bytes, VID of 12 bits, so 4094 usable VLANs.
- Tags exist only on **trunk links**; they are stripped at access ports.
- Splitting a network into VLANs **increases** the number of broadcast domains and so reduces the size of each.

Check yourself

1. Two hosts in VLAN 10 and VLAN 20 on the same switch need to communicate. What is required? *(A router or layer 3 switch.)*
2. How many bits does the VLAN ID field have, and how many VLANs are usable? *(12 bits; 4094.)*
3. A switch has 12 ports: 4 in VLAN 5, 5 in VLAN 6, 3 unassigned. How many broadcast domains? *(3: VLAN 5, VLAN 6 and the default VLAN.)*
4. How long is a maximum 802.1Q-tagged frame? *(1522 bytes.)*

Do next

Draw two switches joined by a trunk, with two VLANs on each, and mark which ports are access ports and which are trunk. Then read about the Spanning Tree Protocol, which stops loops when switches are joined by more than one link.

---

33. Spanning Tree Protocol

Chapter: 2. Data Link Layer · **Videos:** Spanning Tree Protocol (STP) (20 min) · STP solved question (9 min)

Key ideas

- **Redundancy** (extra links between switches) improves reliability and availability: if one path fails, another remains. A single link is a single point of failure.
- Redundancy in a switched LAN creates **loops**, and loops are dangerous. A switch floods broadcasts out of every port except the one they arrived on. In a loop, copies chase each other round and round, and the network fills with traffic. This is a **broadcast storm**. Frames can also be duplicated and MAC tables become unstable.
- The **Spanning Tree Protocol (STP, IEEE 802.1D)** keeps the physical redundancy but keeps the logical topology loop-free. It **blocks** some ports so exactly one active path exists between any two switches. If an active link fails, STP recalculates and unblocks a backup port.
- Switches talk using **BPDUs** (bridge protocol data units), sent every 2 seconds by default. A blocked port still receives BPDUs but does not forward user data.
- The **bridge ID** has a **priority** (default 32768) and the switch's **MAC address**. The root bridge is the switch whose bridge ID is lowest. All have the same default priority, so the lowest MAC address wins unless an administrator lowers the priority of the preferred switch.
- Every other switch finds its cheapest path to the root. The cost depends on link speed, using the classic values: 10 Mbps = 100, 100 Mbps = 19, 1 Gbps = 4, 10 Gbps = 2. Costs along the path add up to the **root path cost**.
- **Port roles:**
  - **Root port:** on each non-root switch, the one port with the lowest root path cost.
  - **Designated port:** on each link (segment), the port that offers the lowest cost to the root. It forwards traffic. Every port of the root bridge is designated.
  - **Blocked (non-designated) port:** every other port; it does not forward data.
- **Tie-breakers**, in order: lowest root path cost, lowest sender bridge ID, lowest sender port ID.
- Classic 802.1D port states: blocking, listening, learning, forwarding (plus disabled). Convergence takes tens of seconds; **RSTP (802.1w)** is faster.
- STP never has to find each loop itself: electing one root and blocking every port that isn't a root or designated port leaves a tree, which has no loops.
- The solved question is of the usual exam type: given a diagram of LANs and numbered bridges, find the root, the root and designated ports, build the tree, then answer a question about it (for example a traversal order).

Formulas and methods

Steps:

1. Pick the root: lowest bridge ID.

2. For each other bridge, add link costs along each path to the root and choose the lowest total. That port is the root port.
3. On each link, the port with the lowest cost to the root is designated.
4. Block the remaining ports.

Worked example: root R, switches S1 and S2. R-S1 is 100 Mbps (cost 19), R-S2 is 10 Mbps (cost 100), S1-S2 is 100 Mbps (cost 19).

- S1: direct path to R costs 19, so its root port faces R.
- S2: direct path costs 100; path through S1 costs 19 + 19 = 38. So S2's root port faces S1.
- Link R-S2: R is on the root, so R's port is designated and S2's port on that link is **blocked**.
- Link S1-S2: S1's cost to root (19) beats S2's (38), so S1's port is designated; S2's is its root port (forwarding). The result is the loop-free tree R, S1, S2 with the 10 Mbps link held in reserve.

Remember for the exam

- Root bridge: **lowest priority, then lowest MAC**.
- Each non-root bridge has **exactly one root port**; each link has **exactly one designated port**.
- Every port of the root bridge is a designated port.
- Blocked ports still listen to BPDUs.
- When costs tie, use lower sender bridge ID, then lower port number. Exam questions often say "ties go to the lower number".
- STP runs at layer 2 and prevents loops; it does not do routing.

Check yourself

1. Why does a loop in a switched network cause a broadcast storm? *(Switches flood broadcasts out of all ports, so copies keep circulating through the loop.)*
2. Three switches have priority 32768 and MACs ending 0A, 0C and 0B. Which is the root? *(The one ending 0A, the lowest MAC.)*
3. A bridge reaches the root through a 1 Gbps link then a 100 Mbps link. Root path cost? *(4 + 19 = 23.)*
4. How many root ports does the root bridge have? *(None.)*

Do next

Draw a ring of four switches, give them bridge IDs, mark root, root, designated and blocked ports, and check there is exactly one blocked port. Practise GATE previous-year questions on spanning tree. The next topic starts the network layer with IPv4 addresses.

---

34. IPv4 addresses

Chapter: 3. Logical Addressing & Subnetting · **Videos:** IPv4 address, part 1 (10 min) · IPv4 address, part 2 (9 min)

Key ideas

- **IPv4 addresses** are **logical addresses** of 32 bits, used at the network layer. It identifies a device's connection to the Internet. Each packet carries a source and a destination IP address. The MAC address identifies the node on its local link and the port number identifies the application.
- Because it has 32 bits, the address space is 2^32 = **4,294,967,296**, a little over 4 billion addresses. IPv6 uses 128 bits.
- A public address belongs to one device at a time on the Internet. Inside a local network, a duplicate address causes a conflict. **Private addresses** can be reused in many networks, and **NAT** (network address translation) maps them to unique public addresses. NAT is also how billions of devices share about 4 billion addresses.
- Two notations: **binary** (32 bits, in four groups of 8) and **dotted decimal** (four numbers separated by dots). Each group is an **octet** (one byte).
- Every octet is in the range **0 to 255**, since 8 bits give 2^8 = 256 values. The first address is `0.0.0.0` and the last is `255.255.255.255`.
- **A valid address** has exactly four octets, each from 0 to 255, in plain decimal.
- **Invalid examples and the reason:** five octets; an octet above 255 (such as 256); a leading zero (such as `045`, because it is ambiguous and some systems read it as octal); a mix of binary and decimal.
- **Conversion uses the eight place values** 128, 64, 32, 16, 8, 4, 2, 1 (2^7 down to 2^0). Binary to decimal: add the values wherever there is a 1. Decimal to binary: take the largest place value that fits, subtract, and repeat.
- Classes, private and public ranges and loopback addresses come in the next videos of the course.

Formulas and methods

Decimal to binary for 192.168.10.37:

- 192 = 128 + 64 gives 11000000
- 168 = 128 + 32 + 8 gives 10101000
- 10 = 8 + 2 gives 00001010
- 37 = 32 + 4 + 1 gives 00100101

Result: `11000000.10101000.00001010.00100101`.

Binary to decimal for `10110010.00000111.01100100.11111110`:

- 10110010 = 128 + 32 + 16 + 2 = 178
- 00000111 = 4 + 2 + 1 = 7
- 01100100 = 64 + 32 + 4 = 100
- 11111110 = 255 − 1 = 254

Result: `178.7.100.254`.

Shortcut: an octet of all ones is 255. If k low bits are zero, the value is 255 minus (2^k – 1). So 11110000 = 255 – 15 = 240.

Number of addresses in a range with n host bits: 2^n.

Remember for the exam

- IPv4 = **32 bits = 4 octets = 4 bytes**; range per octet **0 to 255**.
- Address space is **2^32**, about 4.29 billion, not 2^31 or 2^16.
- Check validity in this order: number of octets, value range, leading zeros, notation.
- Memorise the powers of two up to 128; most conversion questions are quick once you do.
- IP addresses are logical and can change; MAC addresses are physical.

Check yourself

1. Convert 172.20.5.9 to binary. *(10101100.00010100.00000101.00001001.)*
2. Convert 11000000.00000000.00000010.00000001 to dotted decimal. *(192.0.2.1.)*
3. Is 10.256.3.1 valid? *(No; 256 is outside 0 to 255.)*
4. Is 192.168.1.01 valid? *(No; it has a leading zero.)*
5. How many addresses exist in total? *(2^32 = 4,294,967,296.)*

Do next

Convert five addresses from your own network in both directions until it takes under a minute each. Next, the course divides addresses into classes and then subnets. Practise GATE previous-year questions on IPv4 addressing and subnetting.

---

35. Classful addressing

Chapter: 3. Logical Addressing & Subnetting · **Videos:** Classful addressing, part 1 (10 min) · Classful addressing, part 2 (15 min) · Classful addressing, part 3 (6 min) · Classful addressing, CPT activity (9 min)

Key ideas

- An **IPv4 address** is 32 bits, written as four 8-bit **octets** in **dotted decimal** (for example 172.20.45.9) or in binary. Every address has a **network part** (which network) and a **host part** (which device on it).
- In **classful addressing** the first few bits of the first octet decide the class, and the class decides how many bits belong to the network. There are five classes, A to E.
- **Class A** starts with bit `0` (first octet 0 to 127). **Class B** starts with `10` (128 to 191). **Class C** starts with `110` (192 to 223). **Class D** starts with `1110` (224 to 239). **Class E** starts with `1111` (240 to 255).
- Classes A, B and C are for ordinary hosts. **Class D** is for multicast and **class E** is reserved for experiments.
- The **default mask** shows where the network part ends: A is 255.0.0.0 (/8), B is 255.255.0.0 (/16), C is 255.255.255.0 (/24).
- Fewer network bits means more hosts per network. A is few huge networks, C is many small ones.
- The **network address** is the one with all host bits 0. The **broadcast address** is the one with all host bits 1. Neither can be given to a host, so usable hosts = 2^h – 2, where h is the number of host bits.
- Packet Tracer activity: when you type an address into a PC, the mask field fills itself with the default mask of that address's class. Use this to check your own class identification.

Formulas and methods

Class	First octet	Network bits	Host bits	Networks	Usable hosts per network
A	0 to 127	8	24	2^7 = 128 (126 usable)	2^24 – 2 = 16,777,214
B	128 to 191	16	16	2^14 = 16,384	2^16 – 2 = 65,534
C	192 to 223	24	8	2^21 = 2,097,152	2^8 – 2 = 254

Class A has only 126 usable networks because 0.0.0.0/8 and 127.0.0.0/8 (loopback) are reserved.

Method. Look at the first octet, find the class, then keep the network octets and zero the host octets.

Worked example: `172.20.45.9`.

1. 172 is between 128 and 191, so it is class B.
2. Default mask 255.255.0.0: octets 1 and 2 identify the network.
3. Network address: 172.20.0.0. Broadcast address: 172.20.255.255.
4. Usable hosts: 172.20.0.1 to 172.20.255.254, which is 65,534 addresses.

Remember for the exam

- Learn the class boundaries as numbers: 127, 191, 223, 239. The first octet 127 is class A by the bit rule but is reserved for loopback.

- Count network bits including the class prefix bits. Class B has 14 free network bits (not 16) because `10` is fixed, so 2^14 networks. Class C has 21 free bits, so 2^21 networks.
- Hosts per network always lose 2 addresses; networks per class do not lose 2 (except the reserved class A cases).
- Classes D and E have no network/host split and no default mask.
- Class A takes half the address space, B a quarter, C an eighth, and D and E a sixteenth each.

Check yourself

1. Which class is 99.4.4.4, and what is its network address? *(Class A, default mask 255.0.0.0, network 99.0.0.0.)*
2. Which class is 191.255.0.1, and which class is 192.0.0.1? *(Class B and class C: 191 is the last class B value, 192 the first class C value.)*
3. How many usable hosts can a class C network hold? *(254.)*
4. How many class B networks exist? *(2^14 = 16,384.)*
5. A first octet is `11101001` in binary. Which class? *(Class D, since it starts with 1110; the value is 233.)*

Do next

Take ten addresses of your own, write each one's class, default mask, network address and broadcast address, then confirm the mask in Packet Tracer. The next topic covers how a packet is sent to one host, many hosts or all hosts, and applies this class table to solved questions.

---

36. Unicast, multicast, broadcast; class questions

Chapter: 3. Logical Addressing & Subnetting · **Videos:** IPv4 unicast, multicast and broadcast (7 min) · Classful addressing, solved question 1 (7 min) · Classful addressing, solved question 2 (5 min) · Classful addressing, solved question 3 (7 min)

Key ideas

- An IPv4 packet can reach its destination in three ways: **unicast** (one sender to one receiver), **broadcast** (one sender to every host on a network) and **multicast** (one sender to a chosen group).
- A **unicast** destination is an ordinary host address from class A, B or C. This is how almost all web, mail and file traffic travels.
- A **broadcast** reaches every host in a network. A **directed broadcast** uses the broadcast address of a particular network (for example 192.168.1.255 for 192.168.1.0/24). A **limited broadcast** uses 255.255.255.255 and stays inside the sender's own network.
- Routers do not forward limited broadcasts, so a 255.255.255.255 packet never leaves the network it started in.
- A **multicast** packet goes only to hosts that have joined the group. Multicast addresses are the **class D** block, from 224.0.0.0 up to 239.255.255.255. Routing protocols use it, for example 224.0.0.5 for OSPF routers and 224.0.0.9 for RIPv2.
- The solved questions all ask for the same three things about a block: the **first address** (network address), the **last address** (broadcast address) and the **number of usable hosts**.
- The recipe: identify the class from the first octet, keep the network octets, set the host octets to all 0 for the first address and all 255 for the last.

Formulas and methods

- Total addresses in the block = 2^h, where h is the number of host bits. Usable hosts = 2^h – 2.
- Class A: 2^24 addresses. Class B: 2^16 = 65,536. Class C: 2^8 = 256.

Worked examples with our own addresses:

Given address	Class	Network (first)	Broadcast (last)	Usable hosts
210.15.8.130	C	210.15.8.0	210.15.8.255	254
150.60.200.10	B	150.60.0.0	150.60.255.255	65,534
60.9.9.9	A	60.0.0.0	60.255.255.255	16,777,214

Any two hosts sharing the network octets belong to one network: in class A, 60.1.1.1 and 60.200.3.4 are on the same network, and 61.1.1.1 is not.

Remember for the exam

- Directed broadcast (all host bits 1 in one particular network) is different from limited broadcast (255.255.255.255). A router drops the limited one.

- The first and last address of a block are never assigned to hosts, so subtract 2 for hosts but not for "number of addresses".
- Multicast addresses only ever appear as destinations, never as the source of a packet.
- A class D address has no network/host split, so asking for its "network address" makes no sense.
- Read the question for what it asks: total addresses, usable hosts, first address, last address. They differ by 0 or 2.

Check yourself

1. Give the first and last address of 85.22.9.14. *(Class A: 85.0.0.0 and 85.255.255.255.)*
2. How many usable hosts does the block containing 133.4.4.4 have? *(Class B: 65,534.)*
3. Is 239.1.1.1 a unicast address? *(No, it is class D, a multicast address.)*
4. A host sends to 255.255.255.255. How far does the packet travel? *(Every host on its own network, no further.)*
5. Do 200.9.5.1 and 200.9.6.1 share a network? *(No: class C, so 200.9.5.0 and 200.9.6.0 are different networks.)*

Do next

Write three addresses, one per class, and fill in the table above without looking back. Then continue to public and private addresses, which explain why most addresses on your own devices never appear on the Internet.

---

37. Public and private addresses

Chapter: 3. Logical Addressing & Subnetting · **Videos:** Public and private IP addresses (12 min)

Key ideas

- A **public IPv4 address** is globally unique and routable on the Internet. Allocation is controlled, so a public address cannot be picked freely.
- A **private IPv4 address** can be used by anyone inside their own network without permission. Many different organisations reuse the same private addresses at the same time.
- Private addresses were created to save public addresses. Machines that never need direct Internet contact do not need a globally unique address.
- Internet routers do not forward packets with private addresses, so a private host cannot reach the Internet on its own.
- **NAT (network address translation)** solves this. A NAT router replaces the private source address of an outgoing packet with its own public address and keeps a **translation table**. When the reply returns, it looks up the table and forwards the reply to the right private host.
- The three private ranges:
  - 10.0.0.0 to 10.255.255.255 (10.0.0.0/8, in class A)
  - 172.16.0.0 to 172.31.255.255 (172.16.0.0/12, in class B)
  - 192.168.0.0 to 192.168.255.255 (192.168.0.0/16, in class C)
- **Special-use addresses** you should recognise:
  - **Loopback**, 127.0.0.0/8 (usually 127.0.0.1): a host sends traffic to itself. `ping 127.0.0.1` tests the local TCP/IP stack.
  - **Link-local**, 169.254.0.0/16: self-assigned by a host when it finds no DHCP server.
  - **TEST-NET**, 192.0.2.0/24: reserved for documentation and examples.
  - **Class E**, 240.0.0.0 to 255.255.255.254: reserved. 255.255.255.255 is the limited broadcast.
  - The first and last address of every network (network and broadcast address) cannot be given to a host.

Formulas and methods

Check whether an address is private by comparing it with the three ranges.

- First octet 10: private.
- First octet 172 and second octet 16 to 31: private. Everything else starting with 172 is public.
- First octet 192 and second octet 168: private. Everything else starting with 192 is public.

Size of each block (host bits h): 10/8 has 2^24 = 16,777,216 addresses, 172.16/12 has 2^20 = 1,048,576, and 192.168/16 has 2^16 = 65,536.

Remember for the exam

- The class B private range is 172.16 to 172.31, not "all of 172". So 172.32.1.1 and 172.15.1.1 are public.

- 192.168.0.0/16 is private, but 192.169.x.x is not.
- Private addresses are not blocked by the host, but dropped by Internet routers. Without NAT, replies never come back.
- Loopback addresses never leave the host. Link-local addresses never leave the local link.
- NAT keeps a table, so it works as a many-to-one mapping. In practice the table also stores port numbers, which lets many private hosts share one public address.

Check yourself

1. Is 172.31.255.254 private? *(Yes, it is inside 172.16.0.0/12.)*
2. Is 172.32.0.1 private? *(No, it is just past the range.)*
3. Why can two companies both use 10.0.0.5 without conflict? *(Private addresses are only meaningful inside each company's network; NAT hides them from the Internet.)*
4. A PC shows 169.254.37.9. What probably happened? *(It could not get an address from a DHCP server and gave itself a link-local one.)*
5. What does a NAT router store so that replies reach the right host? *(A translation table mapping private addresses to the public address.)*

Do next

Run `ipconfig` or `ip addr` on your own machine, note which range your address belongs to, and look up your public address from a website to see the difference. Next, classless addressing explains why /12 appears in the middle of the private ranges.

---

38. Classless addressing (CIDR)

Chapter: 3. Logical Addressing & Subnetting · **Videos:** Classless addressing, part 1 (6 min) · Classless addressing, part 2 (10 min)

Key ideas

- Classful addressing offers only three block sizes: 256, 65,536 and 16,777,216 addresses. An organisation that needs 300 hosts cannot use a class C (254 hosts), so it takes a class B and leaves about 65,000 addresses unused.
- Three drawbacks of classful addressing: **wasted address space**, **no flexibility** inside a block, and **large routing tables**, because every network needs its own entry.
- **Classless addressing** (**CIDR**, classless inter-domain routing) removes the fixed class boundaries. The network part can be any length from 1 to 32 bits.
- The length is written as a **prefix length** after a slash: `a.b.c.d/n` means the first n bits are the network part and the other 32 – n bits are the host part.
- The **subnet mask** has n ones followed by 32 – n zeros. For example /20 is 255.255.240.0.
- A **valid mask** is a run of ones followed by a run of zeros, with no mixing. In dotted decimal, each octet must be one of 0, 128, 192, 224, 240, 248, 252, 254 or 255, and once an octet is below 255 every later octet must be 0.
- A CIDR block has 2^(32 – n) addresses, always a power of 2. Its first address must be a multiple of the block size.
- CIDR also allows **route aggregation**: several contiguous blocks can be advertised as one shorter prefix, which keeps routing tables small.

Formulas and methods

- Addresses in block = 2^(32 – n). Usable hosts = 2^(32 – n) – 2.
- Hosts needed H: find the smallest h with 2^h – 2 ≥ H. Then the prefix is n = 32 – h.

Worked example: find the range of `12.34.70.9/20`.

1. n = 20, so host bits h = 12. The block holds 2^12 = 4096 addresses; usable hosts 4094.
2. Mask: 255.255.240.0.
3. The third octet decides: 70 = 01000110, AND with 240 (11110000) gives 01000000 = 64.
4. Network address: 12.34.64.0. Block size in the third octet is 16, so the last address is 12.34.79.255.

Prefix for 700 hosts: h = 10 gives 1022 ≥ 700 (h = 9 gives only 510), so n = 22, mask 255.255.252.0. A class B would waste about 64,800 addresses.

Remember for the exam

- /8, /16 and /24 are just the old default masks. CIDR notation is the same arithmetic, with any n allowed.
- 255.255.0.255 and 255.255.255.100 are invalid masks, because their ones are not contiguous.

- "Slash number" = count of ones in the mask, not the count of hosts.
- The address after the slash is a length, so /24 is not "24 hosts".
- A block address such as 12.34.70.0/20 is not a valid network address, because 70 is not a multiple of 16. The network address is 12.34.64.0.

Check yourself

1. Write /26 as a mask and give the number of usable hosts. *(255.255.255.192; 2^6 – 2 = 62.)*
2. Is 255.255.250.0 a valid mask? *(No: 250 is 11111010, not a run of ones then zeros.)*
3. Which prefix suits a department of 3000 hosts? *(2^12 – 2 = 4094 ≥ 3000, so /20.)*
4. How many addresses are in a /29? *(2^3 = 8.)*
5. Name two drawbacks of classful addressing. *(Wasted addresses and large routing tables.)*

Do next

Convert ten prefixes from /17 to /30 to dotted masks and back until it is automatic. The next topic applies these masks to divide one network into several subnets.

---

39. Subnetting

Chapter: 3. Logical Addressing & Subnetting · **Videos:** Subnetting (10 min) · Subnetting, solved problem 1 (13 min) · Verifying the subnetting (CPT activity) (8 min)

Key ideas

- **Subnetting** divides one IP network into two or more smaller **subnets**. Each subnet is a separate network with its own network address, broadcast address and range of hosts.
- Why subnet: a class C block gives 254 hosts, but if you need only 10 hosts per network, a whole class C each is wasteful. Subnetting matches the block to the real need. It also keeps broadcast traffic inside smaller groups.
- Subnetting works by **borrowing bits** from the host part. These bits become a **subnet number**, so the mask gets longer.
- All hosts in one subnet share the same leading bits (the network and subnet bits). A host decides whether a destination is local by comparing those bits with its own.
- Subnetting can start from **host requirements** (hosts per subnet) or from **network requirements** (number of subnets). This topic works from hosts.
- The video's five-step procedure:
  1. Identify the class and note the default mask.
  2. Write the default mask in binary.
  3. From the hosts needed, find the number of host bits and the **subnet generator** (block size) and its octet.
  4. Write the new mask.
  5. Generate the subnets by stepping through the block size.
- In Packet Tracer you verify by building one subnet as a switch with PCs, giving every PC an address from that subnet's usable range and the new mask, and pinging. Addresses equal to the network or broadcast address are not accepted as host addresses.

Formulas and methods

- Host bits: smallest h with 2^h – 2 ≥ hosts needed.
- New prefix: 32 – h. Borrowed bits s = new prefix – default prefix. Subnets = 2^s.
- Block size (subnet generator) = 256 – mask value in the changing octet = 2^(host bits in that octet).
- Per subnet: network = start of block, broadcast = start + block size – 1 (in that octet), usable = between them.

Worked example: subnet `200.50.10.0` so that each subnet holds 40 hosts.

1. Class C, default mask 255.255.255.0 (/24).
2. h: 2^5 – 2 = 30 is too small, 2^6 – 2 = 62 ≥ 40, so h = 6.
3. New prefix /26, mask 255.255.255.192. Borrowed bits s = 2, so 2^2 = 4 subnets. Block size 256 – 192 = 64.

4. Subnets:

Subnet	Network	Usable range	Broadcast
1	200.50.10.0	.1 to .62	200.50.10.63
2	200.50.10.64	.65 to .126	200.50.10.127
3	200.50.10.128	.129 to .190	200.50.10.191
4	200.50.10.192	.193 to .254	200.50.10.255

Remember for the exam

- The mask gets longer when you subnet. More subnets means fewer hosts in each.
- Hosts needed + 2 (network and broadcast) must fit in the block. 62 hosts fit a /26 only because 62 = 64 – 2.
- Every subnet has the same size when the same mask is used for all of them (the fixed-length case).
- Subnet blocks always start at a multiple of the block size.
- Subnetting does not change the address, it only changes how much of it is read as network.

Check yourself

1. Subnet a class C network into subnets with 25 hosts each. What mask and how many subnets? *(h = 5, /27, 255.255.255.224, 8 subnets.)*
2. For `200.50.10.0/26`, what is the broadcast address of subnet 2? *(200.50.10.127.)*
3. What is the block size of mask 255.255.255.240? *(16.)*
4. Why is 200.50.10.63/26 not valid for a PC? *(It's subnet 1's broadcast address.)*

Do next

Build the four-subnet design above in Packet Tracer with one switch per subnet and ping inside each. The next topic repeats the method with class B and class A networks and with subnets that span more than one octet.

---

40. Subnetting practice, part 1

Chapter: 3. Logical Addressing & Subnetting · **Videos:** Subnetting, solved problem 2 (11 min) · Subnetting, solved problem 3 (13 min) · Subnetting, CPT activity (10 min) · Subnetting, solved problem 4 (10 min)

Key ideas

- These videos apply the same five-step method to a class C, a class B and a class A network, each given by hosts needed per subnet.
- Each problem asks for three things: the **subnets** (their ranges), the **number of subnets** and the **hosts per subnet**.
- In class C, subnet bits are borrowed from the last octet only. In class B, the host part is the last two octets, so the subnet bits may spill into the third octet. In class A, there are three host octets, so the number of subnets can be very large.
- The octet where the mask stops being 255 is the **interesting octet**. The block size is read from that octet, and the subnets are generated by counting in steps of that size.
- When the subnet boundary sits in the third octet, each subnet covers several consecutive third-octet values, with the last octet running 0 to 255 inside it.
- Packet Tracer activity: connect each subnet's LAN (switch plus PCs) to a router interface, give the interface an address in that subnet (it becomes the **default gateway**), set the new mask on every PC, then ping between PCs in different subnets to prove **inter-subnet communication**. Hosts in the same subnet reach each other directly.

Formulas and methods

- h = smallest value with 2^h – 2 ≥ hosts needed.
- Subnets = 2^(new prefix – default prefix). Usable hosts = 2^h – 2.
- Block size in the interesting octet = 256 – mask value.

Worked example (class B): subnet `140.30.0.0` with 1000 hosts per subnet.

1. Class B, default mask 255.255.0.0 (/16).
2. 2^10 – 2 = 1022 ≥ 1000, so h = 10. New prefix /22.
3. Mask: 255.255.252.0. Borrowed bits 6, so 2^6 = 64 subnets. Block size in the third octet is 256 – 252 = 4.
4. Subnets: 140.30.0.0, 140.30.4.0, 140.30.8.0, 140.30.12.0, and so on up to 140.30.252.0.
5. Subnet 3 is 140.30.8.0: usable 140.30.8.1 to 140.30.11.254, broadcast 140.30.11.255.

Class A example: `20.0.0.0` with 3000 hosts per subnet. h = 12 (4094 ≥ 3000), prefix /20, mask 255.255.240.0, borrowed bits 12, so 4096 subnets, block size 16 in the third octet.

Remember for the exam

- The subnet mask has to extend from the default mask, so a class B network cannot end up with a prefix shorter than /16 by subnetting.
- Subnet count comes from borrowed bits, not from host bits. Do not confuse the two exponents.
- In class A and B questions the broadcast address ends in .255 on the last octet, and the third octet is start + block size – 1.
- The first usable address of a subnet is network + 1, and the last is broadcast – 1.
- The number of subnets can be huge in class A. That is expected, not a mistake.

Check yourself

1. Subnet 150.15.0.0 into subnets of 500 hosts each. Mask and number of subnets? *(h = 9, /23, 255.255.254.0, 2^7 = 128 subnets.)*
2. In the 140.30.0.0 example above, what is the broadcast address of subnet 2 (140.30.4.0)? *(140.30.7.255.)*
3. Subnet 20.0.0.0 with 3000 hosts per subnet. Usable hosts per subnet? *(4094.)*
4. Which block size matches mask 255.255.248.0, and in which octet? *(8, in the third octet.)*

Do next

Solve one class A, one class B and one class C problem of your own and verify each with Packet Tracer. The next topic covers troubleshooting and finding a host's network and broadcast address from an address and mask.

---

41. Subnetting troubleshooting and practice, part 2

Chapter: 3. Logical Addressing & Subnetting · **Videos:** Subnetting, troubleshooting 1 (6 min) · Subnetting, troubleshooting 2 (8 min) · Subnetting, solved problem 5 (8 min) · Subnetting, solved problem 6 (7 min) · Subnetting, solved problem 7 (5 min) · Subnetting, solved problem 8 (4 min)

Key ideas

- **Troubleshooting 1** is about a host that cannot reach its neighbours because its address or mask is wrong. One cause is an **invalid subnet mask** (a mask whose ones are not contiguous, such as 255.255.255.2). Another is an address that is the **network** or **broadcast address** of its subnet, which a host cannot use.
- **Troubleshooting 2** is about a host that cannot reach its **default gateway**, although both have the same mask. The cause is that the host and the gateway interface sit in **different subnets**. A host sends to its gateway directly, so they must share one subnet.
- Checklist for any connectivity fault: valid mask, usable host address (not first or last of the block), gateway in the same subnet, same mask on both sides.
- **Problems 5 and 6** give one address and a mask and ask for the **first** (network) and **last** (broadcast) address of its block. Two methods are shown: bitwise AND with the mask, and the shortcut using the block size in the interesting octet.
- **Problem 7** gives a network requirement (a number of departments) and asks for the mask: borrow enough bits for the subnet count.
- **Problem 8** gives a mask and asks for hosts per subnet: count the zero bits in the mask and compute 2^h – 2.

Formulas and methods

- Network address = IP address AND mask. Broadcast address = network address OR (NOT mask).
- Shortcut: in the interesting octet, block size B = 256 – mask value, network = largest multiple of B not above the octet value, broadcast = network + B – 1; later octets are 0 (network) or 255 (broadcast).
- Subnets from network requirement: borrowed bits s = smallest with 2^s ≥ number of subnets.
- Hosts per subnet = 2^(zero bits in mask) – 2.

Worked example: `130.145.77.9/20`.

1. /20 is 255.255.240.0, so the third octet is the interesting one. B = 256 – 240 = 16.
2. Multiples of 16 up to 77: 64 (since 80 is more than 77). Network = 130.145.64.0.
3. Broadcast = 130.145.(64 + 16 – 1).255 = 130.145.79.255.
4. Binary check: 77 = 01001101, AND 11110000 gives 01000000 = 64.

Mask for a class B with 20 departments: 2^5 = 32 ≥ 20, so s = 5, prefix /21, mask 255.255.248.0. Hosts per subnet: 11 host bits, 2^11 – 2 = 2046.

Fault example: host 10.1.1.200 with mask 255.255.255.192 and gateway 10.1.1.65. The host is in 10.1.1.192 to 10.1.1.255. The gateway is in 10.1.1.64 to 10.1.1.127. They differ, so the gateway must be moved into 10.1.1.193 to 10.1.1.254.

Remember for the exam

- Never assume a block from the address alone. The mask decides it, even when the address looks like another class.
- A host address equal to the subnet's network or broadcast address is invalid (for example 192.168.5.63 with /26).
- Class B with a "subnet mask 255.255.248.0" has 11 host bits, giving 2046 hosts, not 2048.
- When a question gives a network requirement, round the number of subnets up to the next power of 2, then derive the mask.
- If two machines both look configured correctly but cannot talk, compare network addresses after ANDing.

Check yourself

1. Find the network and broadcast address of 192.168.40.200/27. *(Mask .224, block 32; 200 is in 192 to 223; network 192.168.40.192, broadcast 192.168.40.223.)*
2. Hosts per subnet for mask 255.255.255.224? *(5 host bits, 30.)*
3. A class C network must be split into 6 subnets. Which mask? *(s = 3, /27, 255.255.255.224.)*
4. Why is 255.255.255.2 not a usable mask? *(Its ones are not a single run: 2 is 00000010.)*

Do next

Write the AND steps for five addresses with different prefixes and check each with the shortcut. The next topic applies the same ideas to real GATE-style questions.

---

42. Subnetting practice, part 3

Chapter: 3. Logical Addressing & Subnetting · **Videos:** Subnetting, solved problem 9 (5 min) · Subnetting, solved problem 10 (9 min) · Subnetting, solved problem 11 (11 min) · Subnetting, solved problem 12 (4 min)

Key ideas

- These four videos solve GATE-style questions, each testing a different part of subnetting.
- **Problem 9** gives a fixed number of subnet bits on a class B network and asks for the maximum number of subnets and hosts per subnet. Subnets come from the subnet bits, hosts from what is left.
- **Problem 10** gives two computers with different IP addresses and different masks and asks how each one sees the other. Each computer works alone: it ANDs the destination address with its own mask and compares the result with its own network address. If they match, it believes the destination is local and sends directly. If they differ, it sends via the gateway. The two answers can disagree.
- **Problem 11** names two hosts that share a mask and asks which mask would put them in different networks. Two hosts are in one network only when their network parts are equal after the AND.
- **Problem 12** is a classful count, from GATE CS 2012 according to the video: how many networks class C allows. Class C has 3 fixed prefix bits, leaving 21 free network bits.
- A trap: some textbooks subtract 2 from the subnet count (all-zero and all-one subnets), while CIDR allows them. Read the question, and use the convention it implies.

Formulas and methods

- Subnets = 2^(subnet bits), possibly minus 2 in the old convention. Hosts = 2^(host bits) – 2.
- Class B with s subnet bits: host bits = 16 – s.
- Two hosts X and Y share a network under mask M only if (X AND M) = (Y AND M). Equivalently, the first n bits of X and Y match, where n is the number of ones in M.

Worked example for problem 9's pattern: class B, 5-bit subnet number. Host bits 16 – 5 = 11, so hosts = 2^11 – 2 = 2046. Subnets = 2^5 = 32 (30 under the old subtract-2 convention).

Worked example for problem 10's pattern. C1 = 150.20.70.10 with mask 255.255.192.0; C2 = 150.20.30.5 with mask 255.255.128.0. Only the third octet matters.

- C1's view: own 70 AND 192 = 64; C2's 30 AND 192 = 0. Different, so C1 thinks C2 is remote.
- C2's view: own 30 AND 128 = 0; C1's 70 AND 128 = 0. Same, so C2 thinks C1 is local.

Worked example for problem 11's pattern: hosts 172.16.5.66 and 172.16.5.130. In binary the last octets are 01000010 and 10000010. They first differ at the 25th bit. So masks up to /24 keep them together, and /25 or longer separates them. For example 255.255.255.128 would split them.

Remember for the exam

- With different masks, each host decides on its own: one can treat the other as local and send directly while the other sends through its gateway.

- For "which mask should not be used so both stay on one network", find the first differing bit and exclude every mask longer than that.
- Class A, B, C network counts: 2^7 (126 usable), 2^14, 2^21.
- Always check the question for "maximum number of hosts": subtract 2 for network and broadcast.
- Do not forget the fixed prefix bits when counting classful networks.

Check yourself

1. Class C with a 3-bit subnet number: how many hosts per subnet? *(5 host bits, 30.)*
2. Hosts 192.168.10.37 and 192.168.10.75 share one mask. Which of /26 and /25 keeps them together? *(/25 does: both lie in 0 to 127. /26 does not: 37 is in 0 to 63, 75 in 64 to 127.)*
3. How many class B networks are there? *(2^14.)*
4. Why can host A send directly to B while B sends to A through its gateway? *(Each uses its own mask; A's AND says local, B's AND says remote.)*

Do next

Practise GATE previous-year questions on subnetting and classful counts, writing the AND on paper for each. The next topic moves from equal-sized subnets to unequal ones.

---

43. FLSM and VLSM

Chapter: 3. Logical Addressing & Subnetting · **Videos:** Fixed length subnet masking (FLSM) (11 min) · VLSM, solved problem 1 (13 min) · VLSM, solved problem 2 (12 min)

Key ideas

- **FLSM (fixed length subnet masking)** uses one mask for every subnet, so all subnets have the same size. The mask must be long enough to give the number of subnets needed and short enough for the biggest subnet.
- **VLSM (variable length subnet masking)** uses different masks for different subnets, so each subnet is sized for what it needs. It is classless addressing used on purpose, inside one organisation.
- Count the networks first. Every LAN is a network, and every router-to-router link is also a network. A scenario with three LANs and three routers linked in a chain has more networks than the LANs alone.
- FLSM wastes addresses. A link between two routers needs only 2 usable addresses, but if the common mask is sized for 100 hosts, that link wastes the rest of a /25 block.
- **VLSM procedure:**
  1. List every network and its host requirement.
  2. Sort them from largest to smallest.
  3. Give the largest the first free block of the right size.
  4. Give each next network the next free block, with no overlap.
  5. Leave the remaining space free for later.
- Point-to-point links use /30 (2 usable addresses).
- **Problem 2** is an address-allocation question (GATE CS 2012, as the video notes): given an ISP block as a prefix, give some fractions of it to organisations. Work with block sizes in addresses: half of a /20 is a /21, a quarter is a /22.
- Each allocated block must start at a multiple of its own size and must not overlap another block.

Formulas and methods

- Block size = 2^(32 – prefix). Half of a /n block is /(n + 1). A quarter is /(n + 2).
- Hosts needed H gives h with 2^h – 2 ≥ H and prefix 32 – h.

Worked example (VLSM): divide `192.168.20.0/24` into five networks: LAN A with 100 hosts, LAN B with 50, LAN C with 20, and two router-to-router links.

Network	Need	Block	Network address	Usable range	Broadcast
A	100	/25 (126)	192.168.20.0	.1 to .126	.127
B	50	/26 (62)	192.168.20.128	.129 to .190	.191
C	20	/27 (30)	192.168.20.192	.193 to .222	.223
Link 1	2	/30 (2)	192.168.20.224	.225 to .226	.227
Link 2	2	/30 (2)	192.168.20.228	.229 to .230	.231

Used: 128 + 64 + 32 + 4 + 4 = 232 of 256 addresses. The range .232 to .255 stays free. With FLSM the same problem fails: the 100-host LAN forces /25, which gives only two subnets, not five.

Worked example (allocation): ISP block `90.12.32.0/20` covers 90.12.32.0 to 90.12.47.255 (4096 addresses). Give half to X and a quarter to Y. X = 90.12.32.0/21 (to 90.12.39.255). Y = 90.12.40.0/22 (to 90.12.43.255). Valid, because 32 is a multiple of 8 and 40 is a multiple of 4. Putting Y at 90.12.36.0/22 would overlap X.

Remember for the exam

- Always allocate the largest requirement first. Otherwise a small block can break the alignment of a larger one.
- Count point-to-point links as networks, each using a /30.
- To check a CIDR allocation, check three things: the size fits the fraction, the start address is a multiple of the block size, and the blocks do not overlap.
- In the third octet, a /21 spans 8 values and a /22 spans 4.
- FLSM is simpler, VLSM is efficient. Routing protocols that carry the mask (RIPv2, OSPF, EIGRP) support VLSM; classful RIPv1 does not.

Check yourself

1. What prefix serves a LAN of 28 hosts? *(2^5 – 2 = 30, so /27.)*
2. How many addresses does a /22 hold, and how many third-octet values does it span? *(1024; 4.)*
3. Give the usable addresses of 192.168.20.228/30. *(.229 and .230.)*
4. Why does VLSM allocate the largest requirement first? *(Large blocks need alignment at multiples of their size; small blocks fit in the leftover space.)*
5. Is 90.12.36.0/21 a valid block? *(No: 36 is not a multiple of 8.)*

Do next

Design a VLSM plan for a campus with four LANs of different sizes and two router links, then configure and ping it in Packet Tracer. Practise GATE previous-year questions on CIDR allocation and VLSM. This is the last topic in the free playlist: routing, NAT, TCP and the application layer come next, in Neso's paid chapters or in Kurose & Ross's free lectures.

Made with HoverNotes · https://hovernotes.io/courses/neso-academy-computer-networks