

Computer Networks Quick Revision Notes

Exam-focused notes on the OSI and TCP/IP models, the physical, data link, network, transport and application layers, with 12 solved numericals checked by code and a last-minute checklist.

Free

Free PDF

7 pages

All levels

By [wins.solutions team](#)

Updated 2 Oct 2026

Short answer

For a computer networks exam, revise the OSI and TCP/IP models, then each layer from the bottom up: Nyquist and Shannon, framing, CRC, Hamming code, ARQ protocols and media access, IPv4 subnetting and routing, TCP versus UDP and congestion control, and the main application protocols and ports. Practise the standard numericals until you can do them by hand; 12 are solved here.

WINS.SOLUTIONS · COMPUTER NETWORKS

Computer Networks Quick Revision Notes

These notes follow the order most university syllabi and placement interviews use: models first, then one layer at a time from the bottom up. Each section ends with the points examiners and interviewers ask most. The solved numericals were checked by running code, so you can trust the final answers.

1. Network models

OSI vs TCP/IP

OSI layer	Job	TCP/IP layer	Examples
7 Application	Services for applications	Application	HTTP, DNS, SMTP, FTP
6 Presentation	Encoding, encryption, compression	Application	TLS, JPEG, UTF-8
5 Session	Opening, managing, closing sessions	Application	RPC, NetBIOS
4 Transport	Process-to-process delivery	Transport	TCP, UDP
3 Network	Logical addressing and routing	Internet	IP, ICMP, OSPF
2 Data link	Framing, MAC addressing, error detection	Link	Ethernet, Wi-Fi, PPP

OSI layer	Job	TCP/IP layer	Examples
1 Physical	Bits on the medium	Link	Cables, radio, hubs

Data units: bits (physical), frames (data link), packets (network), segments for TCP or datagrams for UDP (transport), messages (application).

Devices by layer: hub and repeater (1), switch and bridge (2), router (3). A gateway can work at any layer up to 7.

Encapsulation: each layer adds its header on the way down (and the data link layer also adds a trailer); the receiver strips them on the way up.

Key terms

- **Bandwidth:** the range of frequencies a channel carries, in Hz. In computing it's often used loosely for data rate in bits per second.
- **Throughput:** the data rate actually achieved.
- **Latency:** total delay = transmission + propagation + queuing + processing.
- **Transmission delay** = frame length ÷ data rate.
- **Propagation delay** = distance ÷ signal speed (about 2×10^8 m/s in cable and fibre).

2. Physical layer

- **Nyquist (noiseless channel):** maximum data rate $C = 2 \times B \times \log_2(L)$, where B is bandwidth in Hz and L the number of signal levels.
- **Shannon (noisy channel):** $C = B \times \log_2(1 + \text{SNR})$, with SNR as a plain ratio. Convert from decibels: $\text{SNR} = 10^{(\text{dB}/10)}$.
- **Guided media:** twisted pair, coaxial cable, optical fibre (highest bandwidth, immune to electromagnetic interference). **Unguided:** radio, microwave, infrared.
- **Switching:** circuit switching reserves a path for the whole call (old telephones). Packet switching sends independent packets that share links (the internet).

3. Data link layer

Framing and error control

- **Framing methods:** character count, byte stuffing with flag bytes, bit stuffing (insert a 0 after five consecutive 1s), physical-layer coding violations.
- **Parity** detects any odd number of bit errors, but misses an even number.
- **CRC:** append r zero bits (r = degree of the generator), divide by the generator using XOR, and append the remainder. The receiver divides again; remainder 0 means no error was detected.
- **Hamming code:** places parity bits at positions 1, 2, 4, 8 and so on. The syndrome gives the position of a single-bit error, so it can **correct** it.
- **Detecting d errors** needs a minimum Hamming distance of $d + 1$; **correcting d errors** needs $2d + 1$.

Flow control: ARQ protocols

Let a = propagation delay ÷ transmission delay.

Protocol	Efficiency	Window	Sequence bits needed for window N
Stop-and-wait	$1 \div (1 + 2a)$	1	1
Go-Back-N	$N \div (1 + 2a)$, capped at 100%	Sender N, receiver 1	$N \leq 2^k - 1$
Selective Repeat	$N \div (1 + 2a)$, capped at 100%	Sender N, receiver N	$N \leq 2^{k-1}$

Go-Back-N resends the lost frame **and every frame after it**. Selective Repeat resends only the lost frame, so it needs a buffer at the receiver.

Media access

- **Pure ALOHA**: transmit any time. Maximum throughput $1 \div (2e)$, about **18.4%**.
- **Slotted ALOHA**: transmit only at slot boundaries. Maximum $1 \div e$, about **36.8%**.
- **CSMA/CD (classic Ethernet)**: listen before sending, detect collisions, back off. The minimum frame must last at least one round trip: $L_{\min} = 2 \times \text{propagation delay} \times \text{data rate}$.
- **CSMA/CA (Wi-Fi)**: avoids collisions with random backoff and optional RTS/CTS, because wireless stations can't reliably detect collisions.

Ethernet, switches and ARP

- A **MAC address** is 48 bits, written in hexadecimal, and identifies the network interface.
- **Switches** learn which MAC address sits on which port and forward frames only there. **Hubs** repeat every frame to every port.
- **ARP** finds the MAC address for an IPv4 address on the local network by broadcast.

4. Network layer

IPv4 addressing

Class	First octet	Default mask	Use
A	1–126	/8	Large networks
B	128–191	/16	Medium networks
C	192–223	/24	Small networks
D	224–239	none	Multicast
E	240–255	none	Reserved

127.x.x.x is loopback. **Private ranges**: 10.0.0.0/8, 172.16.0.0/12 and 192.168.0.0/16.

Subnetting: with a /n prefix there are $2^{(32 - n)}$ addresses per subnet and $2^{(32 - n)} - 2$ **usable hosts**, because the first address is the network address and the last is the broadcast address. **CIDR** drops classes and uses any prefix length.

Routing and related protocols

- **Distance vector (RIP):** each router shares its table with neighbours. Simple, but slow to converge and prone to count-to-infinity. Fixes include split horizon and poison reverse. RIP's maximum is 15 hops.
- **Link state (OSPF):** each router floods link information and computes shortest paths with **Dijkstra's algorithm**. Converges faster.
- **BGP:** routes between autonomous systems on the internet (path vector).
- **ICMP:** error and diagnostic messages; ping and traceroute use it.
- **NAT:** maps private addresses to a public address, saving IPv4 space.
- **DHCP:** assigns IP addresses automatically through Discover, Offer, Request, Acknowledge (DORA).
- **IPv6:** 128-bit addresses, a simpler fixed header, no broadcast (multicast and anycast instead), and no fragmentation by routers.

5. Transport layer

TCP vs UDP

	TCP	UDP
Connection	Connection-oriented (three-way handshake)	Connectionless
Reliability	Acknowledgements, retransmission, ordering	None
Flow and congestion control	Yes	No
Header size	20–60 bytes	8 bytes
Used by	HTTP/1.1 and HTTP/2, SMTP, FTP, SSH	DNS, video calls, online games, QUIC

- **Three-way handshake:** SYN, SYN-ACK, ACK. **Closing:** FIN and ACK in each direction.
- **Flow control** protects the receiver, using the advertised window.
- **Congestion control** protects the network: **slow start** doubles the congestion window each round trip until it reaches the threshold (ssthresh), then **congestion avoidance** adds 1 per round trip. On a **timeout**, ssthresh becomes half the current window and the window resets to 1. On **three duplicate ACKs**, fast retransmit and fast recovery halve the window instead of resetting it.
- **Ports:** 0–1023 are well-known (80 HTTP, 443 HTTPS, 22 SSH, 25 SMTP, 53 DNS). A **socket** is an IP address plus a port.

6. Application layer

Protocol	Port	Transport	One line
HTTP / HTTPS	80 / 443	TCP (HTTP/3 uses QUIC over UDP)	Request-response web protocol; HTTPS adds TLS

Protocol	Port	Transport	One line
DNS	53	UDP (TCP for large responses and zone transfers)	Turns names into IP addresses
SMTP	25 (587 for submission)	TCP	Sends email between servers
POP3 / IMAP	110 / 143	TCP	Fetch email; IMAP keeps it on the server
FTP	20 data, 21 control	TCP	File transfer with separate control and data connections
SSH	22	TCP	Encrypted remote login
DHCP	67 server, 68 client	UDP	Automatic address assignment

DNS resolution: your resolver asks a root server, then the top-level domain server (.com), then the authoritative server for the domain. **Recursive** means the resolver does the work for you; **iterative** means each server refers you to the next one.

HTTP methods: GET (read), POST (create), PUT (replace), PATCH (update part), DELETE. GET, PUT and DELETE are idempotent: repeating them has the same effect as doing them once. **Status codes:** 2xx success, 3xx redirect, 4xx client error, 5xx server error.

7. Security basics

- **Symmetric encryption** (AES) uses one shared key and is fast. **Asymmetric** (RSA, elliptic-curve) uses a public and private key pair and solves key exchange.
- **TLS** uses asymmetric cryptography to agree a session key, then symmetric encryption for the data. Certificates prove the server's identity.
- **Hashing** (SHA-256) gives a fixed-size fingerprint and can't be reversed. **Digital signatures** are a hash encrypted with the sender's private key.
- **Firewalls** filter traffic by rules. **VPNs** tunnel encrypted traffic over a public network.

Solved numericals

Every answer below was checked by running code.

1. Transmission and propagation delay

A 1,500-byte frame is sent at 10 Mbps over a 2,000 km link (signal speed 2×10^8 m/s). Find both delays.

Transmission delay = $(1,500 \times 8) \div (10 \times 10^6) = \mathbf{1.2\ ms}$. Propagation delay = $(2,000 \times 10^3) \div (2 \times 10^8) = \mathbf{10\ ms}$.

2. Stop-and-wait efficiency

For the link in example 1, find the efficiency of stop-and-wait.

$a = 10 \div 1.2 = 8.33$. Efficiency = $1 \div (1 + 2 \times 8.33) = 1 \div 17.67 = \mathbf{5.66\%}$.

3. Go-Back-N with window 7

Same link, sender window 7. Efficiency = $7 \div 17.67 = \mathbf{39.6\%}$.

4. Window and sequence bits for 100% efficiency

The window must be at least $1 + 2a = 17.67$, so **18 frames**. Go-Back-N needs $2^k - 1 \geq 18$, so **k = 5 bits**. Selective Repeat needs $2^{k-1} \geq 18$, so **k = 6 bits**.

5. Nyquist

A noiseless 3,000 Hz channel uses 4 signal levels. $C = 2 \times 3,000 \times \log_2 4 = \mathbf{12,000 \text{ bps}}$.

6. Shannon

A 3,000 Hz channel has an SNR of 30 dB. $\text{SNR} = 10^3 = 1,000$. $C = 3,000 \times \log_2(1,001) \approx \mathbf{29,902 \text{ bps}}$, about 30 kbps.

7. CSMA/CD minimum frame size

10 Mbps Ethernet, maximum cable length 2,500 m, signal speed $2 \times 10^8 \text{ m/s}$. Propagation delay = $12.5 \mu\text{s}$. $L_{\min} = 2 \times 12.5 \times 10^{-6} \times 10 \times 10^6 = \mathbf{250 \text{ bits}}$.

8. CRC

Data 1101011011, generator 10011 ($x^4 + x + 1$). Append four zeros and divide by XOR: the remainder is **1110**, so the transmitted frame is **11010110111110**. Dividing the received frame by 10011 leaves 0000, so no error is detected.

9. Hamming (7,4), even parity

Encode data 1011, with bits in the order p1 p2 d1 p4 d2 d3 d4. p1 checks positions 1, 3, 5, 7; p2 checks 2, 3, 6, 7; p4 checks 4, 5, 6, 7. The codeword is **0110011**. If bit 6 flips (received 0110001), recomputing the checks gives a syndrome of **6**, which points at the error.

10. Subnetting a /24 into four

Split 192.168.10.0/24 into 4 equal subnets. Borrow 2 bits, so each is a **/26** with 64 addresses and **62 usable hosts**.

Subnet	Usable range	Broadcast
192.168.10.0/26	.1 to .62	192.168.10.63
192.168.10.64/26	.65 to .126	192.168.10.127
192.168.10.128/26	.129 to .190	192.168.10.191
192.168.10.192/26	.193 to .254	192.168.10.255

11. Network and broadcast address from a host

For 172.16.45.200/20: the mask is 255.255.240.0. The third octet 45 falls in the block 32–47, so the network is **172.16.32.0**, the broadcast is **172.16.47.255**, the usable hosts run from **172.16.32.1 to 172.16.47.254**, and there are $2^{12} - 2 = 4,094$ of them.

12. TCP congestion window

ssthresh = 16 segments, the window starts at 1, and a timeout happens in the round where the window reaches 20. Window per round: **1, 2, 4, 8, 16** (slow start), then **17, 18, 19, 20** (congestion avoidance). After the timeout, ssthresh = $20 \div 2 = 10$ and the window restarts: **1, 2, 4, 8, 10, 11**.

Bandwidth–delay product: a 100 Mbps link with a 40 ms round trip holds $100 \times 10^6 \times 0.04 = 4,000,000$ bits (**500 KB**) in flight. That's the window needed to keep the link full.

Last-minute checklist

- ☐ Draw the OSI and TCP/IP models side by side, with one protocol per layer.
- ☐ Write the formulas for transmission delay, propagation delay, Nyquist, Shannon and ARQ efficiency from memory.
- ☐ Do one CRC division and one Hamming encoding by hand.
- ☐ Find the network, broadcast and host range for any IP with a prefix.
- ☐ Explain the difference between Go-Back-N and Selective Repeat in two sentences.
- ☐ Trace the TCP window through slow start, congestion avoidance and a timeout.
- ☐ List the ports for HTTP, HTTPS, DNS, SMTP, SSH and FTP.
- ☐ Explain what happens when you type a URL: DNS, TCP, TLS, HTTP request, response.

Revising other subjects too? See the **DBMS quick revision notes** and the **operating systems important questions**.