

Telecomms Cheat Sheet T3

MAC Protocols Cheat Sheet

♦ What is MAC?

- MAC is a sub-layer of the **Data Link Layer** in the OSI model.
 - It controls **how multiple devices share a communication channel**.
 - Goal: Prevent **collisions** and **coordinate access** efficiently.
-

1. ALOHA Protocol

➤ Pure ALOHA

- Any station can transmit **at any time**.
- If collision occurs, wait for a **random time** and retransmit.
- **Max throughput: 18.4%** ($S = G * e^{-2G}$)

➤ Slotted ALOHA

- Time is **divided into slots**; transmission only at slot start.
 - Reduces collision chances.
 - **Max throughput: 36.8%** ($S = G * e^{-G}$)
-

2. CSMA (Carrier Sense Multiple Access)

- Devices **listen to the channel** before transmitting.
- **Carrier sensing** helps avoid collisions.

➤ Types of CSMA:

Type	Description
1. 1-persistent CSMA	If channel is idle, send immediately. If busy, keep listening and send as soon as it becomes idle.
2. Non-persistent CSMA	Wait a random time if channel is busy. Less chance of collision.
3. p-persistent CSMA	Used in slotted channels. Send with probability p when idle, otherwise wait.

3. CSMA/CD (Collision Detection)

- Used in **wired LANs (like Ethernet)**.
 - After sending, device continues to **listen**:
 - If it **detects collision**, it stops, sends a **jamming signal**, and waits before retrying.
 - Efficient but **not usable in wireless** networks.
-

4. CSMA/CA (Collision Avoidance)

- Used in **wireless LANs (like Wi-Fi)**.
- Devices use **RTS/CTS** (Request to Send / Clear to Send) to avoid collisions.

- Since collision detection is **not possible** wirelessly, collisions are **prevented** instead.

♦ Collision Handling Techniques

Technique	Description
Backoff	After a collision, wait for a random time before retrying.
Exponential Backoff	Wait time doubles after each successive collision (used in Ethernet).

✅ Comparison Table

Protocol	Channel Sensing	Collision Handling	Throughput (Max)
Pure ALOHA	✗	Random wait	18.4%
Slotted ALOHA	✗	Slot-based resend	36.8%
CSMA	✅	Random/backoff	Higher than ALOHA
CSMA/CD	✅	Detect and jam	Efficient (wired)
CSMA/CA	✅	Avoid via RTS/CTS	Efficient (wireless)

Network Layer (OSI Layer 3)

Purpose:

- Responsible for **routing** and **forwarding** of packets across networks.
- It handles **logical addressing**, not physical.
- Ensures packets get from **source to destination** even across multiple networks.

◆ Core Functions of the Network Layer

Function	Description
Routing	Selecting the best path for data from source to destination across networks.
Forwarding	Moving packets from input to the appropriate output interface on a router.
Logical Addressing	Assigns IP addresses to uniquely identify each host in a network.

◆ IP (Internet Protocol)

- IP is a **protocol** that defines how data should be **packetized, addressed, transmitted, routed, and received**.
- It's **connectionless** and **best effort** (no guarantee of delivery).

➤ IPv4 (Internet Protocol version 4)

Feature	Value
Address Size	32 bits (e.g., 192.168.0.1)
Total Addresses	~4.3 billion
Address Format	Dotted Decimal
Header Size	20–60 bytes
Limitations	Limited addresses, no built-in security, no QoS

➤ IPv6 (Internet Protocol version 6)

Feature	Value
Address Size	128 bits (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334)

Total Addresses ~340 undecillion

Address Format Hexadecimal, separated by colons

Features Built-in security (IPsec), no need for NAT, simplified headers, QoS support

◆ NAT (Network Address Translation)

- Allows **multiple private IPs** to share a **single public IP**.
- Common in home/office networks behind routers.
- Maps **internal addresses** ↔ **external addresses**.

➤ Why NAT?

- To **save IP addresses** (especially in IPv4).
- To **hide internal network** structure for security.

➤ Types:

Type	Description
Static NAT	One-to-one mapping of private to public IP.
Dynamic NAT	Chooses from a pool of public IPs.
PAT (Port Address Translation)	Many-to-one: Uses ports to differentiate devices.

◆ ISP-NAT (Carrier-Grade NAT / Large-Scale NAT)

- Used by ISPs when they **don't have enough public IPv4 addresses**.
- Customers get private IPs even at the ISP level.

- Similar to regular NAT, but done at a **much larger scale**.

➤ **Problem:**

- Makes **end-to-end communication** harder (e.g., P2P, hosting servers).
 - Another reason to move to **IPv6**.
-

◆ **Forwarding Techniques**

➤ **1. Destination-Based Forwarding**

- Most common.
- Router looks at **destination IP** and forwards based on routing table.

➤ **2. Label-Based Forwarding**

- Used in **MPLS (Multiprotocol Label Switching)**.
- Uses **short path labels** instead of IP addresses.

➤ **3. Source-Based Forwarding**

- Uses **source address** and predetermined path.
 - Less common; can be used in specific networks or security setups.
-

◆ **Extra Concepts**

➤ **Subnetting**

- Dividing a network into smaller logical segments.
- Helps in efficient IP address management.

➤ CIDR (Classless Inter-Domain Routing)

- Eliminates fixed IP classes (A, B, C).
- Allows variable-length subnet masks (e.g., /24, /26).

✓ Summary Table

Concept	IPv4	IPv6
Address Size	32 bits	128 bits
NAT Needed	Yes	No
Number of Devices	~4.3 billion	Virtually unlimited
Format	Dotted decimal	Hexadecimal
Security	Optional (IPsec)	Built-in (IPsec)

Let me know if you want this in visual format (flowcharts/diagrams) or if you need help solving subnetting/NAT-related numerical questions.

You said:

also do subnetting

ChatGPT said:

Here's a **Subnetting Cheat Sheet** along with **step-by-step explanation** so you can understand how to subnet any given IP block.

◆ What is Subnetting?

Subnetting is the process of **dividing a large network** into **smaller sub-networks (subnets)** to:

- Improve **routing efficiency**
 - Enhance **network security**
 - Better utilize IP **address space**
-

◆ Key Concepts

Term	Description
IP Address	A 32-bit address in IPv4 used to identify a device.
Subnet Mask	Used to determine the network and host parts of an IP.
CIDR Notation	Shorthand like /24 meaning 24 bits are for the network.
Network Address	First address in a subnet (used to identify the subnet).
Broadcast Address	Last address in a subnet (used to send to all hosts).
Usable Hosts	Total IPs in the subnet minus 2 (network + broadcast).

◆ Important Formulas

1. **Number of subnets** = $2^{\text{borrowed bits}}$
 2. **Number of hosts per subnet** = $2^{\text{remaining bits}} - 2$
 3. **Block size** = $256 - \text{subnet mask octet}$
-

◆ Subnetting Table (Common CIDR Notations)

CIDR	Subnet Mask	# Subnets (from /24)	Hosts/Subnet
/24	255.255.255.0	1	254
/25	255.255.255.128	2	126
/26	255.255.255.192	4	62
/27	255.255.255.224	8	30
/28	255.255.255.240	16	14
/29	255.255.255.248	32	6
/30	255.255.255.252	64	2

◆ Step-by-Step Example

Q: You're given the IP block: **192.168.1.0/24**. You need to make 8 subnets.

Step 1: Find subnet mask

You want 8 subnets $\rightarrow 2^3 = 8 \rightarrow$ So, borrow 3 bits from host part.

- Original /24 (255.255.255.0)
- New subnet mask = **/27** \rightarrow **255.255.255.224**

Step 2: Block Size

Block size = **256** $-$ **224** = **32**

Step 3: Subnets

Subnet No.	Network Address	First Host	Last Host	Broadcast Address
Subnet 1	192.168.1.0	192.168.1.1	192.168.1.30	192.168.1.31
Subnet 2	192.168.1.32	192.168.1.33	192.168.1.62	192.168.1.63

Subnet 3	192.168.1.64	192.168.1.65	192.168.1.94	192.168.1.95
Subnet 4	192.168.1.96	192.168.1.97	192.168.1.126	192.168.1.127
...

You continue incrementing by 32 for each subnet.

◆ Quick Subnetting Tips

- **Power of 2s:** Always think in terms of 2^n when dealing with subnets or hosts.
- **Block size:** Helps you find the start of each subnet (use for addressing).
- **Subtract 2 for usable hosts.**
- Broadcast address is always the **last** address in the block.
- First usable host = network + 1
- Last usable host = broadcast – 1

◆ TCP (Transmission Control Protocol)

TCP is a reliable, connection-oriented protocol used when accurate data delivery is critical.

Key Features:

Feature	TCP
Connection	Connection-oriented (setup via 3-way handshake)
Reliability	Guaranteed delivery using ACKs, retransmission
Ordering	Maintains order of data packets
Error Checking	Yes (via checksum, sequence numbers)

Speed	Slower due to overhead
Use Cases	Web browsing (HTTP/HTTPS), Email (SMTP), File Transfer (FTP)

How It Works:

- **Handshake:** Establishes a connection (SYN → SYN-ACK → ACK)
- **Data Transfer:** Ensures all packets are delivered in order and correctly
- **Teardown:** Gracefully closes the connection

◆ UDP (User Datagram Protocol)

UDP is a fast, connectionless protocol used when speed is more important than reliability.

Key Features:

Feature	UDP
Connection	Connectionless (no handshake)
Reliability	No guarantees, no ACKs or retransmission
Ordering	No guarantee of order
Error Checking	Basic checksum only
Speed	Faster , less overhead
Use Cases	Online gaming, VoIP, Streaming, DNS

How It Works:

- **No setup:** Sends data immediately without establishing a connection
- **Unreliable but fast:** Great for real-time data where occasional loss is okay

◆ Side-by-Side Summary

Feature	TCP	UDP
Reliability	High (with error correction)	Low (no correction)
Speed	Slower	Faster
Order Guarantee	Yes	No
Use Cases	Email, Web, File Transfer	Video Calls, Games, DNS
Connection Type	Connection-oriented	Connectionless
Overhead	High	Low

◆ Congestion in Networking

? What is Congestion?

Congestion happens **when too much data is sent into the network**, and routers/switches can't handle it all. This causes:

- **Packet delay**
- **Packet loss**
- **Retransmissions**, which make congestion even worse

Analogy:

Think of a highway during rush hour: too many cars (data packets), not enough lanes (bandwidth). Traffic slows down, and some cars get stuck or take alternate routes.

Causes of Congestion:

- Too many packets at once

- Slow or busy links
- Insufficient buffer memory in routers
- High traffic from multiple users at the same time

Effects of Congestion:

- **High latency (delay)**
- **Jitter (variation in delay)**
- **Dropped packets**
- **Reduced throughput**

Congestion Control Techniques:

Technique	Description
Open Loop	Tries to prevent congestion (e.g., traffic shaping, admission control)
Closed Loop	Detects and reacts to congestion (e.g., window resizing in TCP)
Avoidance Mechanisms	Predict and avoid congestion (e.g., RED – Random Early Detection)

◆ **Quality of Service (QoS)**

What is QoS?

QoS is a set of techniques to manage network resources and ensure a certain level of performance for critical applications like video calls, VoIP, or streaming.

Why is QoS Important?

Because not all data is equal. For example:

- Video calls need low **delay**
- Streaming needs high **bandwidth**
- Emails can tolerate some delay

Key QoS Parameters:

Parameter	Description
Bandwidth	Amount of data that can be sent per second
Delay	Time it takes for a packet to reach its destination
Jitter	Variation in delay
Packet Loss	% of packets lost in transmission

QoS Mechanisms:

Mechanism	Description
Classification	Tagging traffic by type (e.g., voice, video, data)
Queuing	Prioritize some traffic over others (e.g., voice first)
Traffic Shaping	Control traffic flow rate (smooth out bursts)
Resource Reservation	Reserve bandwidth for important traffic (e.g., RSVP protocol)
Policing	Drop or mark excess traffic if it exceeds limits

Relation Between Congestion & QoS:

- Congestion negatively affects QoS
- QoS techniques help **mitigate congestion effects** by prioritizing and managing traffic

◆ CONGESTION CONTROL METHODS

1. Open-Loop Control

Goal: Prevent congestion **before** it happens (proactive).

Method	Explanation
Traffic Shaping	Regulates data entering the network to avoid bursts (e.g., Leaky Bucket, Token Bucket).
Admission Control	Only allow new connections if network has enough resources.
Load Shedding	If system is overloaded, low-priority packets are discarded.

2. Closed-Loop Control

Goal: React to congestion **after it is detected** (reactive).

Method	Explanation
Backpressure	Congested node tells previous node to stop sending.
Choke Packet	A control packet is sent back to the source to slow down transmission.
Implicit Signaling	Source detects congestion based on increased delay or packet loss.
Explicit Signaling	Routers explicitly notify the sender about congestion (e.g., ECN in IP).

3. Congestion Avoidance (Preventive)

Goal: Keep the network operating just below the congestion point.

Method	Explanation
Random Early Detection (RED)	Routers drop packets <i>before</i> queue is full to signal congestion.
TCP Congestion Control	Uses slow start , congestion avoidance , fast retransmit , and fast recovery .

◆ QUALITY OF SERVICE (QoS) METHODS

QoS is about **guaranteeing performance** for different types of traffic (e.g., giving priority to video over email).

1. Classification & Marking

Tag packets based on type (e.g., VoIP, HTTP) so routers can treat them accordingly.
Example: Differentiated Services Code Point (DSCP) field in IPv4 header.

2. Queuing Techniques

Decide **which packet goes first** when multiple packets compete.

Method	Description
FIFO (First In First Out)	Basic method, no priority.
Priority Queuing (PQ)	High-priority packets (like VoIP) are always sent first.
Weighted Fair Queuing (WFQ)	Assigns weights to flows; higher weight = more bandwidth.

3. Traffic Shaping

Regulates the traffic flow from the source to **smooth out bursts**.

Method	Description
Leaky Bucket	Converts bursty traffic to steady stream.
Token Bucket	Allows bursts if enough tokens are available. More flexible than Leaky Bucket.