Switched Networks — overview

- **Switched network** = a network built from *nodes called switches* joined by links.
- A **switch** is a device (hardware/software) that establishes temporary or permanent connection paths through the network so end systems (computers, phones) can talk.
- Some switches are **edge switches** (connected to end systems); others are **core/route-only** switches used only for forwarding/routing.
- Example topology in your text: end systems labeled A–L, switches labeled I, II, III… — to send A → L the message must traverse a series of switches/links.

Why switches? they let the network set up paths (or routes) dynamically and forward traffic efficiently without every device being directly connected to every other device.

Classification (taxonomy) of switching techniques

High-level split:

1. **Circuit-switched networks**
2. **Packet-switched networks**
    - **Message-switched networks** (store-and-forward full messages)
    - **Datagram networks** (each packet routed independently)
    - **Virtual-circuit networks** (a logical circuit is set up, then packets follow it)

So packet switching breaks into datagram vs virtual-circuit; message switching is an older store-and-forward approach.

Circuit switching — definition & essence

- Circuit switching *reserves* a dedicated path and resources between sender and receiver for the entire session.
- **OSI layer:** conceptually at the *physical* layer (resource reservation of channels/time).
- **Resource reservation** can be:
    - **FDM (Frequency Division Multiplexing):** each conversation gets a fixed frequency band, or
    - **TDM (Time Division Multiplexing):** each conversation gets fixed time slots.
- Once a circuit is established, data flows continuously (not broken into network-layer packets in the classic sense).
- **Addressing** is needed during *setup* (to find the destination); after setup the path is known so per-packet addressing isn't required for the duration of the circuit.

Real world: Traditional telephone networks (PSTN) used circuit switching. Modern telephony (VoIP) uses packet switching.

Circuit switching — the three phases (detailed)

Every circuit-switched communication has **three phases**:

1. **Connection setup (circuit establishment)**
   - Source sends a *setup request* (containing destination address) to its local switch.
   - Each switch along the path allocates a free channel (frequency/time slot or internal switch resource) on the outgoing link toward the destination.
   - The setup message is forwarded hop-by-hop until it reaches the destination.
   - Destination replies with an *acknowledgment* back to the source (hop-by-hop), confirming the circuit is created.
   - Only after the source gets the ACK does data transfer begin.

   Notes: The signaling messages for setup/teardown can be in-band (over the same channel) or out-of-band (separate control channel). Telephone networks use sophisticated signaling (example: SS7).

2. **Data transfer**
   - The reserved circuit is used for continuous transmission.
   - No per-packet routing decisions are needed; the switch fabric forwards based on the established path.
   - If multiplexing (TDM/FDM) is used, each circuit has its reserved slot/band for the whole session.
3. **Connection teardown (circuit disconnect)**
   - When communication ends (call hangup / application end), a teardown signal is sent through the path to release all reserved resources so they can be reused.

Multiplexing: TDM vs FDM (what they are & how they look)

FDM (Frequency Division):

- Entire link bandwidth is split into different frequency bands; each circuit uses one band continuously.
- Analogy: many radio stations on different frequencies.
- Visual (conceptual):

```
|======|======|======|  frequency axis
  ch1    ch2    ch3
```

TDM (Time Division):

- Time is divided into frames; each circuit gets specific time slots in every frame.
- Analogy: many people take turns speaking in turn, each at fixed times.
- Visual (conceptual):

```
Frame 1: [slot1][slot2][slot3][slot4]
Frame 2: [slot1][slot2][slot3][slot4]
```

Key difference: FDM reserves frequency resources constantly; TDM reserves time slot cycles constantly.

Efficiency & resource usage (why circuit switching can be wasteful)

- **Resource reservation** means channels are tied up for the entire connection—even when no data is being sent. That leads to **poor utilization** if sources are bursty or idle much of the time.
- Circuit switching gives guaranteed bandwidth (good for steady, continuous flows like classic voice) but wastes capacity for bursty traffic (most computer traffic).
- **Packet switching** uses statistical multiplexing (on-demand sharing) and typically achieves much higher average utilization.

Delay components in circuit-switched networks — breakdown & formula

Your slides mention:

Total delay = setup delay + $3T$ + 3τ + teardown delay

Let's define and derive that precisely.

Notation & definitions

- Suppose there are **n links (hops)** between source and destination (e.g., Host1 → Switch1 → Switch2 → Host2 gives $n = 3$ links).
- For each link:

    - TTT = **transmission time** (also called service time) for the data unit on the link = L/RL/RL/R where LLL = number of bits to send, RRR = link bandwidth (bits/sec).
    - τ\tauτ (tau) = **propagation delay** for that link (distance / propagation speed).
- **Setup delay** = time to send the setup request from source to destination and for the destination's acknowledgement to return. That equals a sum of propagation and processing times for the setup round-trip — more on components below.
- **Teardown delay** = time to send the teardown signal through path and confirm release, analogous to setup.

Transfer delay

- For continuous circuit transfer across n links, the data must be transmitted across each link; total is roughly n(T+τ)n(T + \tau)n(T+τ). If the data is a single block/packet and pipeline effects are considered, the expression can change slightly; but slides used 3T+3τ3T + 3\tau3T+3τ because n = 3 in their example.

Setup delay components (more detail)

Setup delay = (i) propagation time of the setup request along the path (sum of propagation times across all links), + (ii) processing and queuing delays at intermediate switches for the setup message, + (iii) propagation + processing of the acknowledgment back. In simple form:

- Setup ≈ sum of propagation times (forward) + sum of processing times (forward) + sum of propagation times (back) + sum of processing times (back).

Complete formula (generalized)

If nnn links:

Total delay≈setup_delay+n(T+τ)+teardown_delay\text{Total delay} \approx \text{setup_delay} + n(T + \tau) + \text{teardown_delay}Total delay≈setup_delay+n(T+τ)+teardown_delay

For the 3-link example they gave:

Total delay=setup_delay+3T+3τ+teardown_delay\text{Total delay} = \text{setup_delay} + 3T + 3\tau + \text{teardown_delay}Total delay=setup_delay+3T+3τ+teardown_delay

Numeric example

- Suppose message size L=1,000,000L = 1{,}000{,}000L=1,000,000 bits, bandwidth R=1,000,000R = 1{,}000{,}000R=1,000,000 bps → T=L/R=1T = L/R = 1T=L/R=1 s.
- Propagation per link τ=0.05\tau = 0.05τ=0.05 s.
- n = 3 → transfer part = 3(1+0.05)=3.153(1 + 0.05) = 3.153(1+0.05)=3.15 s.
- If setup delay = 0.2 s and teardown = 0.1 s, total ≈ 0.2 + 3.15 + 0.1 = **3.45 s**.

Important note: The slide's expression uses 3T + 3τ because it assumes 3 links; in general substitute n.

Addressing & forwarding in circuit switching

- **During setup:** the setup message carries the address of the destination so each switch can forward the request toward destination.
- **After setup:** each switch has reserved a channel and local forwarding table state saying "traffic on this incoming channel → go out on this outgoing channel." So per-packet addresses aren't needed; switches just forward based on the established circuit.
- In **virtual-circuit packet switching**, packets still carry a small VC identifier (not full addresses), so the per-packet overhead is less than a full IP header but still present.

Advantages & disadvantages (summary)

Advantages

- Guaranteed dedicated bandwidth and consistent latency during connection (good for real-time voice in classic systems).
- No (or minimal) per-packet routing/processing overhead once circuit is set up.
- Predictable performance (if circuit established).

Disadvantages

- **Inefficient** for bursty traffic; resources are locked even when idle.
- **Long setup delay** before data can be sent (bad for short transactions).
- **Inflexible** — once path set you cannot easily reroute mid-session for better performance.
- No built-in priority/classes in classic circuit switching (everything equal), though modern networks add QoS on top.
- Scaling & resource management is harder in highly dynamic traffic environments.

Packet switching — quick contrast (because your slide contrasted it)

- **No resource reservation.** Resources are allocated on demand, per packet.
- **Two main flavors:**
  - **Datagram networks:** each packet routed independently (like IP). Packets may take different paths and can arrive out of order. Must include full addressing info in each packet.

    - **Virtual-circuit networks:** a logical circuit is established (signalling or soft state); then packets use small VC identifiers and follow that path (e.g., ATM, Frame Relay — historically). Setup exists but is lighter than classical circuit setup; packets still routed per-VC.
    - **Message switching:** older: the entire message is stored at intermediate node then forwarded (store-and-forward of whole messages). Causes large delays and requires storage.
- **Advantages of packet switching:** efficient multiplexing, better utilization, more robust in presence of failures (packets can be rerouted).
- **Disadvantages:** variable delay, possible packet loss, out-of-order arrival, jitter — needs buffering, retransmission protocols, QoS mechanisms.

Where to use which?

- **Circuit switching**: used historically for voice (steady continuous stream with strict latency needs).
- **Packet switching**: used for general data traffic, Internet, and increasingly for voice (VoIP) with QoS to handle jitter and latency. Modern telecom is converging on packet switching (MPLS, IP) with virtual circuits and QoS features that emulate some circuit properties.

Practical examples & technologies

- **Circuit-switched example:** PSTN (public switched telephone network), ISDN (earlier digital circuit-switched).
- **Virtual circuit packet tech:** ATM, Frame Relay, MPLS (Label Switched Paths can approximate virtual circuits).
- **Datagram:** Internet Protocol (IP) — each IP packet is independent.
- **Message switching:** historical message relay networks or store-and-forward email-like systems.

Exam-style comparison table (concise)

Feature	Circuit switching	Datagram packet switching	Virtual-circuit packet switching
Resource reservation	Yes (dedicated)	No	Setup reserves VC state (but not fixed channel in same sense)
Setup delay	High (for circuit setup)	Low/none (send immediately)	Moderate (VC setup)

Feature	Circuit switching	Datagram packet switching	Virtual-circuit packet switching
Per-packet addressing	No (after setup)	Yes (full address)	Small VC id
Efficiency for bursty traffic	Poor	Good (statistical multiplexing)	Good
Order of arrival	In-order	Can be out of order	In-order (VC)
Suitable for	Continuous steady streams	IP/data, bursty flows	Telecoms needing VC semantics, QoS

Extra clarifications (common pitfalls / subtle points)

- **“Circuit switching happens at physical layer”** — the concept of reserving a physical channel is a physical-layer idea, but actual signalling and control often uses control-plane protocols at higher layers (signalling networks operate outside the raw physical channel).
- **Switch vs router vs hub:**
  - A **switch** (in the switched network context) is the node that establishes and maintains dedicated connection state or forwarding entries. In Ethernet terms, a switch is a Layer-2 forwarding device (not the same as circuit switches), but conceptually both "switch" types forward frames/packets.
  - A **router** is a Layer-3 device making packet-by-packet routing decisions (datagram world).
  - A **hub** is a dumb repeater — not used in switched network designs for routing.
- **Virtual circuit does not mean physical dedicated circuit.** It’s a logical concept: packets carry a small label/VC id, switches forward based on VC id; bandwidth may still be shared.

1. Datagram Networks (Packet-Switched Network)

Also called **Connectionless Networks**.

Key Characteristics

1. **Create Packets**
   - Before transmission, the message is broken into **packets**.
   - Size of packets may be **fixed** or **variable**, depending on the network and protocol.
   - Typical size: ~1000 bytes.
   - Each packet contains:
     - **User data** (actual message content)
     - **Control information** (headers, destination address, error check info, etc.)
2. **No Resource Allocation**

   - No dedicated path is reserved before sending data.
   - **No reserved bandwidth** → packets share network capacity dynamically.
   - Packets use **first-come, first-served** access.
   - This is unlike circuit switching, where resources are locked for the whole duration.
3. **Store-and-Forward**
   - Packets move from **node to node** (nodes = routers/switches).
   - At each node, packets are **stored briefly** in a buffer before forwarding.
4. **Delay**
   - If many packets are in the buffer, each new packet **waits**.
   - Causes **queuing delay** at busy switches.
5. **Packets are Independent Entities**
   - Each packet is **routed separately**.
   - Even packets from the same message may take **different paths**.
6. **Different Routes Possible**
   - If one path is congested, packets may take an alternate route.
   - Ensures fault tolerance, but can cause **out-of-order delivery**.
7. **Variable Delay**
   - Packets taking different paths → arrive at different times.
   - May also get **lost** or **duplicated**.
8. **Reordering by Upper Layers**
   - Transport layer protocols like **TCP** reorder packets or request retransmission of lost packets.
9. **Connectionless Operation**
   - No **setup** or **teardown** phases.
   - Switches (routers) do **not** store connection state information.
10. **Layer of Operation**
    - Implemented at **network layer** (e.g., Internet Protocol - IP).

Role of Routing Table

- Each router maintains a **routing table**.
- Table maps **destination addresses → output ports**.
- When a packet arrives:
  1. Router reads **destination address** from packet header.
  2. Looks up the port in its routing table.
  3. Forwards packet via that port.

Efficiency

- More efficient than circuit switching:
  - Resources used **only when** there is data to send.
  - Idle resources are used by other packets.

Delay in Datagram Network

For **one packet**:
Total delay = $3T + 3\tau + W_1 + W_2$
Where:

- **T** = Transmission delay (time to push bits into the link)
- **τ** = Propagation delay (signal travel time in the medium)
- **W_1, W_2** = Queuing delays at intermediate nodes.

Advantages

- Cost-effective for short transmissions.
- Fault-tolerant: even if one route fails, others can be used.
- Can handle traffic even when network is busy.
- Can support **priority** packets.

---

2. Virtual-Circuit Networks

A **hybrid** between circuit switching and datagram switching.

Features from Circuit Switching

- **Setup & Teardown phases** (connection-oriented).
- Resources can be **reserved** during setup.
- All packets follow **same path**.

Features from Datagram Switching

- Data is **packetized**.
- Each packet carries **address information**.
- Can allocate resources **on demand**.

Layer of Implementation

- Usually implemented at **Data Link Layer** (e.g., Frame Relay, ATM).
- Circuit switching = Physical Layer; Datagram = Network Layer.

Addressing

- **Global Address** → unique across the network. Used during setup.
- **Virtual-Circuit Identifier (VCI)** → short number **local to each switch**:

- Incoming frame from one link may have VCI = 14,
  when sent out next link → VCI changes to 22.
- VCI changes **at every hop**.

Three Phases

1. **Setup Phase**
   - Source sends **setup request** to destination via switches.
   - Each switch assigns an incoming port, incoming VCI, and outgoing port.
   - Outgoing VCI decided during acknowledgment.
2. **Data Transfer Phase**
   - All packets follow same path.
   - Each switch uses its table to change the **VCI** and forward.
3. **Teardown Phase**
   - Source sends **teardown request**.
   - Destination replies with **teardown acknowledgment**.
   - Switches delete virtual-circuit entries.

Delay

Total delay = $3T + 3\tau$ + setup delay + teardown delay

- Setup delay is **extra** compared to datagram networks.

---

3. Circuit Switching vs Packet Switching

Feature	Circuit Switching	Packet Switching
Setup Phase	Required	Not required (for datagram)
Resource Reservation	Yes	No
Data Path	Fixed	Can change per packet
Reliability	High	Lower (handled by upper layers)
Delay	Uniform	Variable
Efficiency	Low (resources unused when idle)	High (shared resources)
Implementation Layer	Physical	Data Link & Network
Store-and-Forward	No	Yes
Charging Basis	Time & distance	Bytes sent & time connected

---

4. Sockets and Network Software

What is a Socket?

- **Endpoint** for two-way communication between two programs.
- Defined by: **IP Address + Port Number**.
- Can be used:
  - Between processes on the same machine.
  - Between processes on different machines.

Socket Attributes

1. **Domain**
   - `AF_INET` → Internet (TCP/UDP)
   - `AF_UNIX` → Local interprocess communication
2. **Type**
   - **SOCK_STREAM** → TCP (connection-oriented, reliable, ordered)
   - **SOCK_DGRAM** → UDP (connectionless, unreliable, unordered)
3. **Protocol**
   - Chosen based on type and domain.

Server-Client Workflow

Server:

1. Create socket.
2. Bind to IP and port.
3. Listen for incoming connections.
4. Accept connections.
5. Communicate.
6. Close connection.

Client:

1. Create socket.
2. Connect to server IP and port.
3. Communicate.
4. Close connection.

---

5. Transmission Impairments

What is it?

- Imperfections in transmission medium cause **signal degradation**.
- Received signal $\neq$ transmitted signal.

Causes

1. **Attenuation** (loss of signal strength)
   - Due to resistance of medium.
   - Measured in **decibels (dB)**:
     - Using power: $\text{Attenuation} = 10 \log_{10}(P_2 / P_1)$
     - Using voltage: $\text{Attenuation} = 20 \log_{10}(V_2 / V_1)$
     - Negative dB → loss; Positive dB → gain.
2. **Distortion**
   - Different frequencies in signal travel at different speeds.
   - Alters waveform shape.
3. **Noise**
   - Unwanted signals interfere with original data.
   - Types: Thermal noise, Induced noise, Crosstalk, Impulse noise.