End-to-End Communication

Goal: Interconnect multiple LANs.

Why?

- Diverse LANs speak different languages

  → need to make them talk to each other

- Need management flexibility

  → global vs. local Internet

  → administrative policy barriers

Problems:

- How to choose paths (routing)?
- How to regulate flow (congestion control)?

  → not too much, not too little

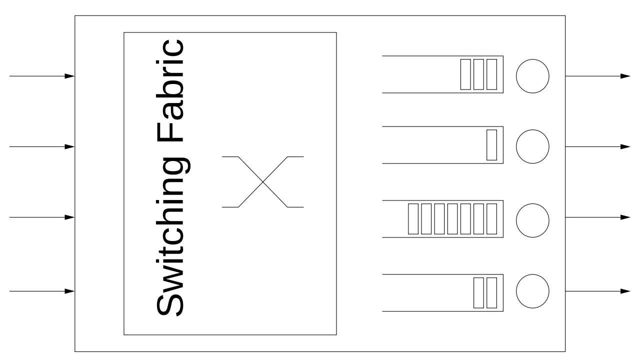
- How to provide service quality (QoS control)?

Packet Switching vs. Circuit Switching

Router/switch design:

- Hardware: ASIC
- Software: fast PC as router or gateway
- Hybrid: network processor

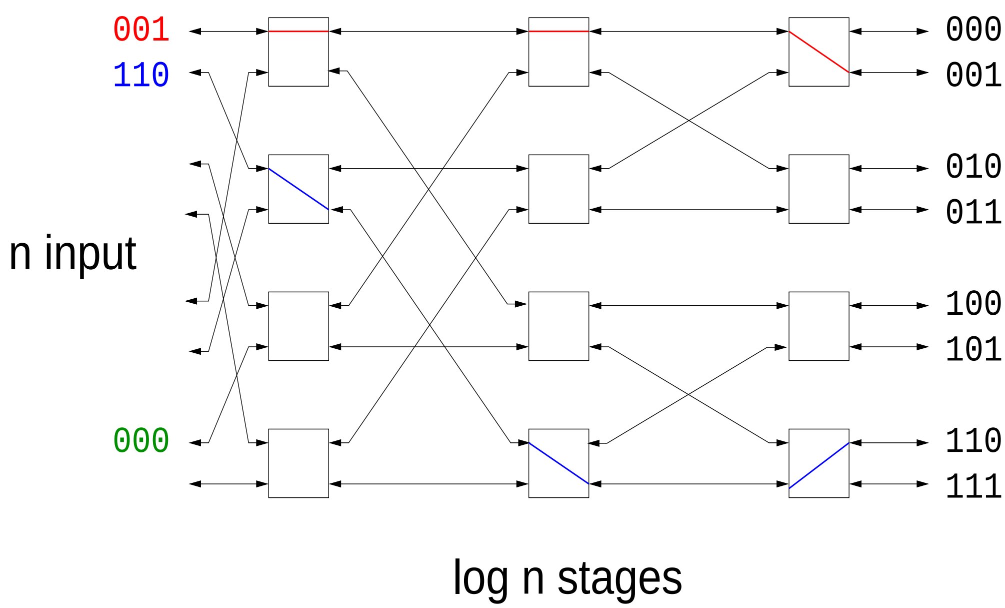
  → programmable



⟶ fast vs. slow forwarding path

⟶ interconnection network

Ex.: Butterfly interconnection network



- self-routing

  → enter top: if 0 go through else cross

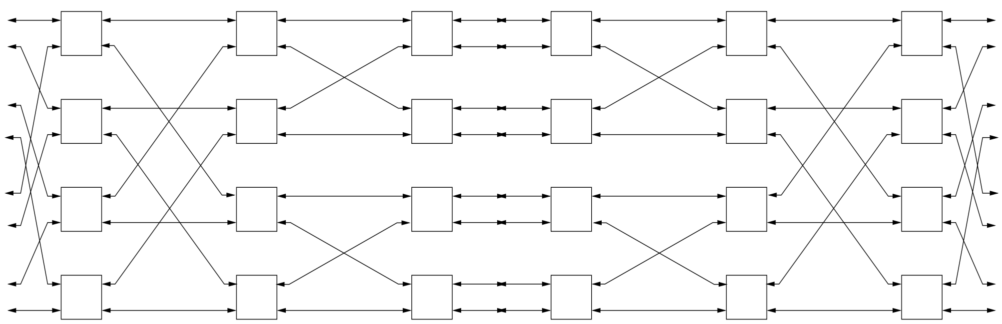
  → enter bottom: opposite

- blocking

  → some permutations cause collision

  → e.g., 001 and 000 collide at second stage

Back-to-back butterfly:



2 log n stages

$\longrightarrow$ non-blocking but cannot use self-routing

$\longrightarrow$ cost: $O(n \log n)$

Cross-bar: everyone-to-everyone else

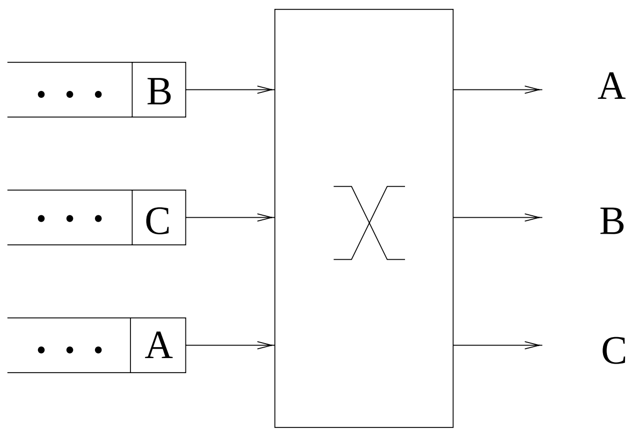
$\longrightarrow$ non-blocking

$\longrightarrow$ cost: $O(n^2)$

$\longrightarrow$ doesn't scale; ok for small n (in practice)

Problem with input-buffered switch design:

$\longrightarrow$ head-of-line blocking



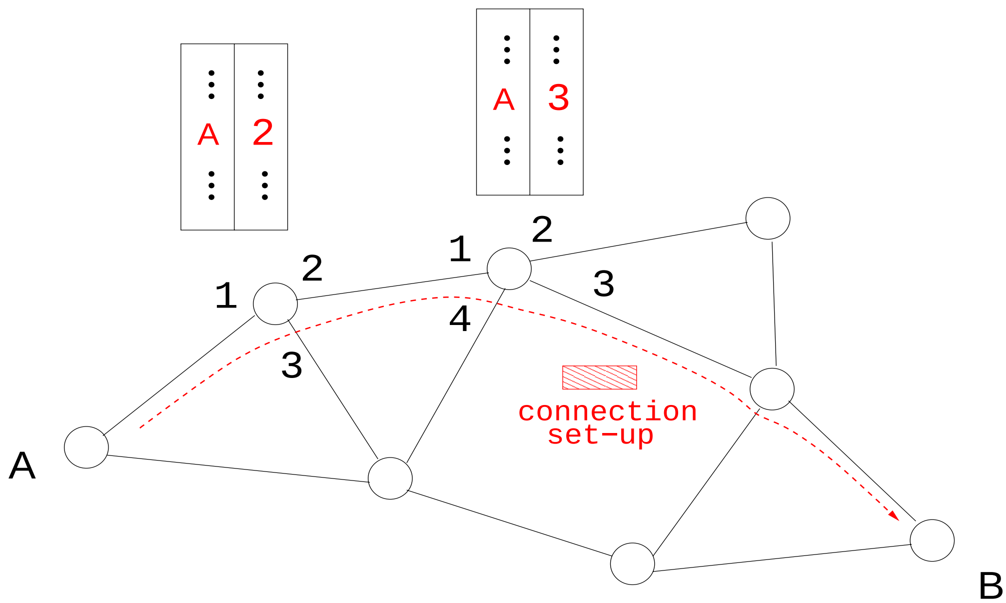
Most routers: output-buffered design

$\longrightarrow$ get to output port fast: routing

$\longrightarrow$ not done yet: output port may be crowded

$\longrightarrow$ packet scheduling

Circuit-switched forwarding:



- connection set-up message: signaling

  → how to: routing subsystem

  → different from forwarding subsystem

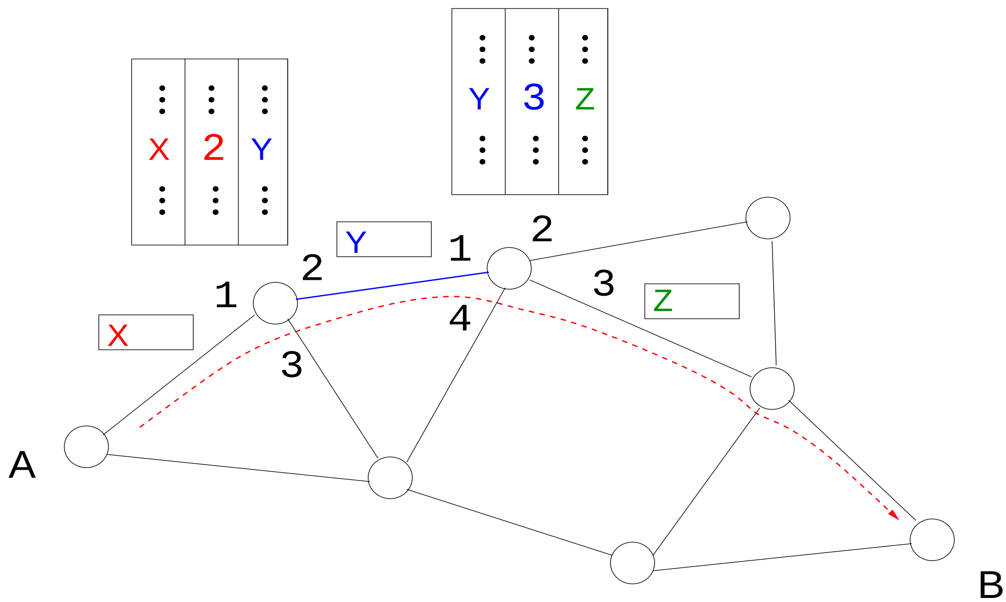
- source tag "A" inserted into look-up table

  → on-demand, compact look-up table

  → deletion upon termination

  → tag: VPI (virtual path identifier)

VCI (virtual channel identifier):



⟶ keep switching tag (hop-to-hop)

⟶ bundling effect on heavily shared link/channel

⟶ e.g., share same tag X on blue link

⟶ sufficient?

⟶ need VCI/VPI pair

Packet-switched forwarding:

$\longrightarrow$ dispense with connection set-up signaling

$\longrightarrow$ each packet: autonomous entity

Source routing:

- packet contains path information

  $\rightarrow \langle A, C, \ldots, B \rangle$
- drawback: header length increases with path length

  $\rightarrow$ not good for fast packet handling

Destination-based forwarding:

- determine output port by destination address
- source address ignored

$\longrightarrow$ destination address: postfix VCI

$\longrightarrow$ same destination, same path: at any node

$\longrightarrow$ Internet packet switching

Internet Protocol (IP)

Goals:

- interconnect diverse LANs into one logical entity
- implement best-effort service

  → no assurances ("what you get is what you get")

  → simplicity

Represents:

- common language for carrying out non-LAN-specific conversations

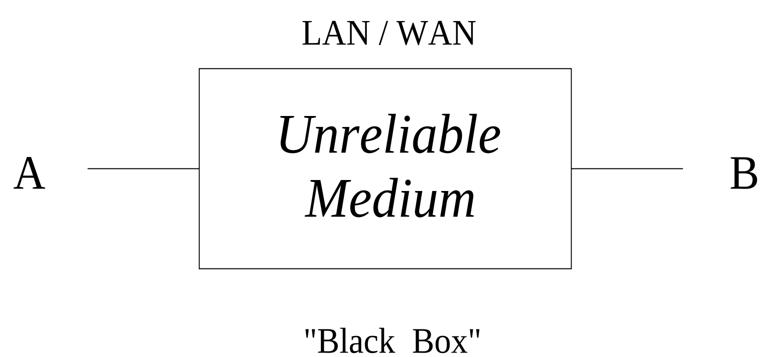
  → technical definition of **I**nternet
- functionality and design philosophy

  → simple core / complex edge

  → end-to-end paradigm

Best-effort vs. guaranteed service:

- easier to implement best-effort service; no resource reservation/leasing
- simplifies router design but increases complexity of end stations
- necessitates higher-up functional layer to achieve reliable transmission over unreliable medium



IP packet format:

<table>
<tr><th>4</th><th>4</th><th>8</th><th colspan="2">16</th></tr>
<tr><td>version</td><td>header length</td><td>TOS</td><td colspan="2">total length</td></tr>
<tr><td colspan="3">fragmentation identifier</td><td>flags</td><td>fragment offset</td></tr>
<tr><td colspan="2">TTL</td><td>protocol</td><td colspan="2">header checksum</td></tr>
<tr><td colspan="5">source address</td></tr>
<tr><td colspan="5">destination address</td></tr>
<tr><td colspan="5">options (if any)</td></tr>
</table>

- Header length: in 4 byte (word) units.
- TOS (type-of-service): Partially used.
- 4 bytes used for fragmentation.
- TTL (time-to-live): Prevent cycling (default 64).
- Protocol: demultiplexing key (TCP 6, UDP 17).

Fragmentation and reassembly:

LAN has maximum transmission unit (MTU): maximum frame size

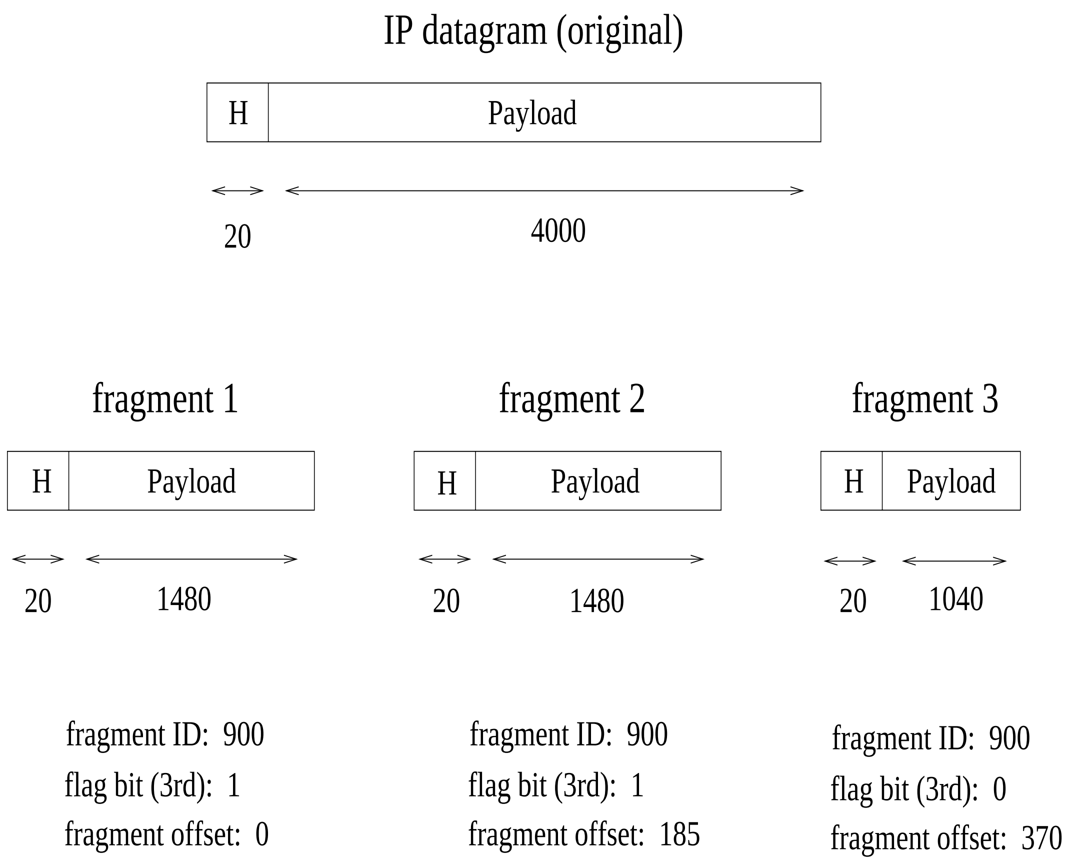
$\longrightarrow$ e.g., Ethernet 1500 B, FDDI 4500 B

- potential size mismatch problem (IP 64 kB)
- may happen multiple times hopping from LAN to LAN

Solution: fragment IP packet when needed, maintain sequencing information, then reassemble at destination.

- assign unique fragmentation ID
- set 3rd flag bit if fragmentation in progress
- sequence fragments using offset in units of 8 bytes

Example: IP fragmentation (Ethernet MTU)



Note: Each fragment is an independent IP packet.

Destination discards all fragments of an IP packet if one is lost.

$\longrightarrow$ "all for one, one for all"

$\longrightarrow$ set 2nd flag bit to disable fragmentation

TCP: Negotiate at start-up TCP segment (packet) size based on MTU; 1 kB or 512 B are common.

$\longrightarrow$ tries to prevent fragmentation