Internetworking and End-to-End Communication

Goal: Interconnect multiple LANs

Why?

- Diverse LANs speak different languages

  → need to make them talk to each other

  → cannot use native LAN interconnection technology

- Management flexibility

Key problems:

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

  → not too fast, not too slow

- How to provide service quality (QoS control)?

Routing: circuit- vs. packet-switched

→ Internet: packet-switched dominant

→ every packet is on its own

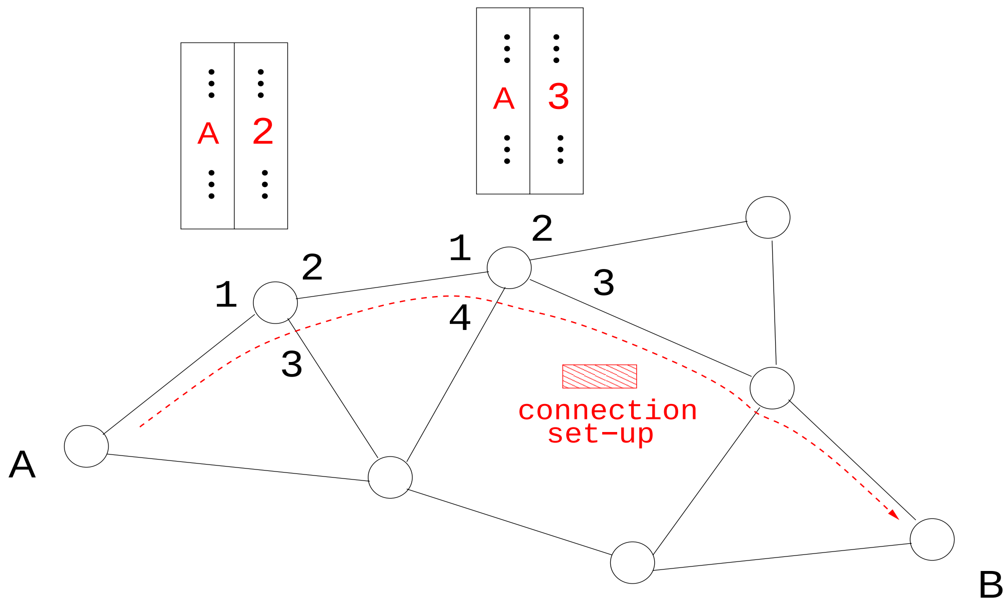
→ intranet of large ISPs: circuit- and packet-switched

→ fixed path/circuit is allocated to a group of packets

→ ex.: session: file download, VoIP (voice-over-IP), VoD (video-on-demand)

→ ex.: all packets from Atlanta to Seattle on an ISP's intranet

Circuit-switched routing: set-up



- connection set-up message: signaling

  → route selection: routing subsystem

  → different from route set-up subsystem

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

  → dynamic look-up table

  → entry deletion upon session termination

Packet-switched routing: set-up

$\longrightarrow$ no 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

  $\rightarrow$ option still available in IP: may be used for management purposes

Destination-based forwarding:

- determine output port by destination address
- source address ignored

→ same destination, same path

→ unless route tables change over time

→ e.g., shortest-path routing

→ any negative issues stemming from ignoring source address?

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 is key

IP represents:

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

  → technical definition of **I**nternet

- functionality and design philosophy

  → simple core / complex edge

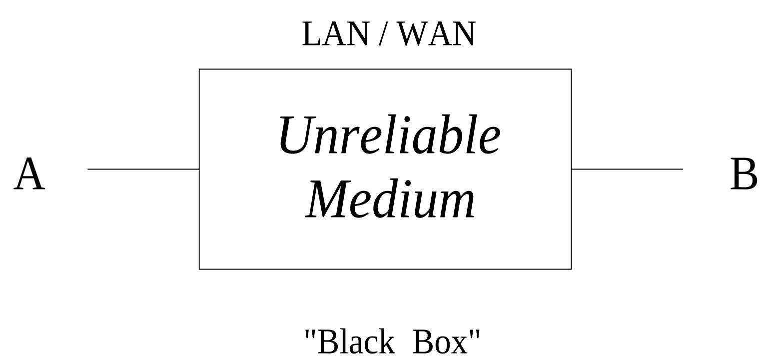
  → called end-to-end paradigm

Reliability over best-effort Internet:

- simplifies router design but increases complexity of end stations (e.g., servers, PCs, laptops, handhelds)

  → implement ARQ at sender/receiver

  → router does not carry out ARQ at IP layer or above



IPv4 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 (e.g., 64).
- Protocol: demultiplexing key (TCP 6, UDP 17).

Fragmentation and reassembly:

LAN has maximum transmission unit (MTU):

→ maximum frame size

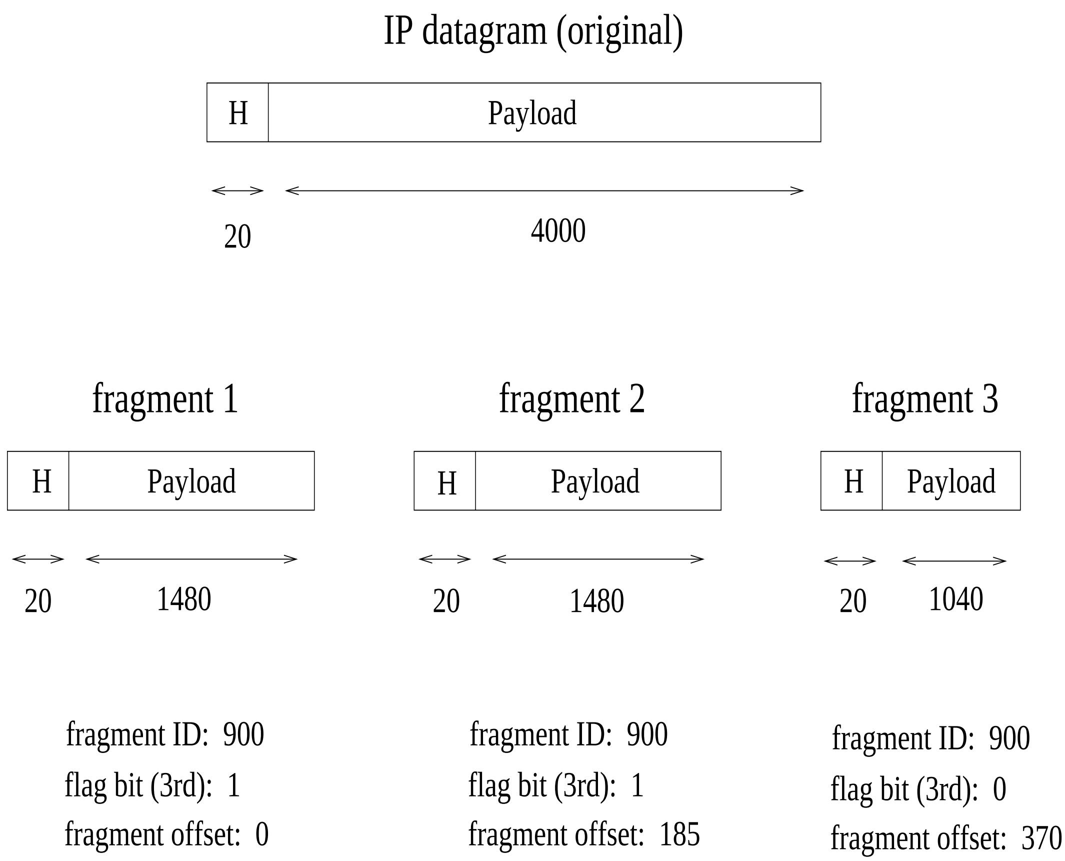
→ e.g., Ethernet 1500 B, WLAN 2313 B

- 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.

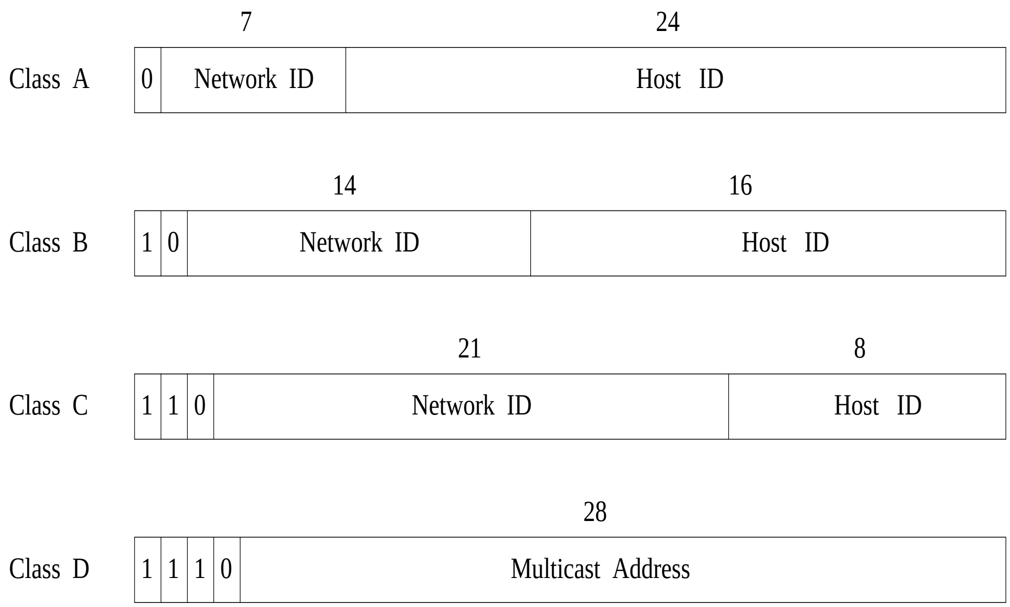
$\longrightarrow$ "all for one, one for all"

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

TCP: Negotiate at start-up TCP segment (packet) size based on MTU

$\longrightarrow$ tries to prevent fragmentation

IP address format:



Dotted decimal notation: 10000000 00001011 00000011 00011111 $\leftrightarrow$ 128.11.3.31

Symbolic name to IP address translation: domain name server (DNS).

Hierarchical organization: 2-level

→ network and host

Each interface (NIC) has an IP address; single host can have multiple IP addresses.

→ single-homed vs. multi-homed

Running out of IP addresses . . . or not?

→ note: IANA gave out last block to regional registries

→ should Purdue get a class B address?

→ how about your start-up company?

→ what about Purdue's Biology Dept.?

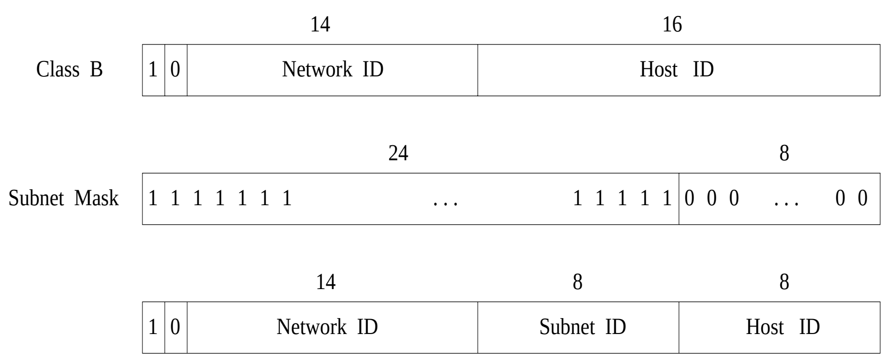
→ what about the Math Dept.?

Waste of address space:

⟶ typical organization: network of networks

⟶ not too many hosts (class B: 64K)

Solution: subnetting—subdivide host ID into subnetwork ID and host ID



To determine subnet ID:

- AND IP address and subnet mask

  → already know if class A, B, C, or D

- 3-level hierarchy

Forwarding and address resolution:

Subnet ID	Subnet Mask	Next Hop
128.10.2.0	255.255.255.0	Interface 0
128.10.3.0	255.255.255.0	Interface 1
128.10.4.0	255.255.255.0	128.10.4.250

Either destination host is connected on a shared LAN, or not (additional IP hop needed).

$\longrightarrow$ reachable by LAN address forwarding

$\longrightarrow$ if not, network address (IP) forwarding

Table look-up I ("where to"):

- For each entry, compute *SubnetID* = *DestAddr* AND *SubnetMask*.
- Compare *SubnetID* with *SubnetID*.
- Take forwarding action (LAN or IP).

Remaining task: translate destination or next hop IP address into LAN address

$\longrightarrow$ must be done in either case

$\longrightarrow$ address resolution protocol (ARP)

Table look-up II ("what's your LAN name"):

- If ARP table contains entry, using LAN address link layer can take over forwarding task.

  → ultimately everything is LAN

  → network layer: virtual

- If ARP table does not contain entry, broadcast ARP Request packet with destination IP address.

  → e.g., Ethernet broadcast address (all 1's)

- Upon receiving ARP response, update ARP table.

Dynamically maintain ARP table: use timer for each entry (15 min) to invalidate entries.

$\longrightarrow$ aging (standard caching technique)

Subnetting only goes so far.

$\longrightarrow$ depts. within Purdue share same class B address

$\longrightarrow$ what about your start-up company?

$\longrightarrow$ only 2^{21} class C addresses available

Other approaches to solve address depletion problem:

• IPv6

→ 128-bit addresses

→ who wants it (or doesn't want it)?

→ not used much: will it change?

• Classless (vs. classful) IP addressing

→ variable length subnetting

→ that is, $a.b.c.d/x$ (x: mask length)

→ e.g., 128.10.0.0/16, 128.210.0.0/16, 204.52.32.0/20

→ used in inter-domain routing

→ CIDR (classless inter-domain routing)

→ de facto global Internet addressing standard