End-to-End Communication

Goal: Interconnect multiple LANs.

Why?

- Diverse LANs speak different languages

  $\longrightarrow$ need to make them talk to each other

- Management flexibility

  $\longrightarrow$ global vs. local Internet

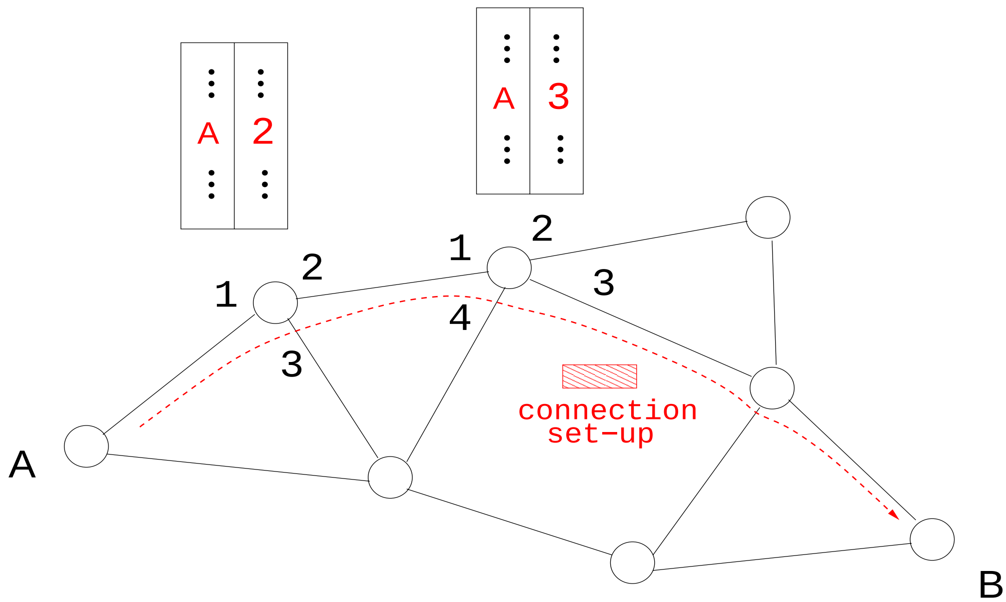
Problems:

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

  $\longrightarrow$ not too much, not too little

- How to provide service quality (QoS control)?

Route set-up: circuit-switched



- connection set-up message: signaling

  → route selection: routing subsystem

  → different from route set-up subsystem

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

  → on-demand look-up table

  → deletion upon termination

Route set-up: packet-switched

$\longrightarrow$ 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

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

$\longrightarrow$ Internet packet switching

$\longrightarrow$ any problems with 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

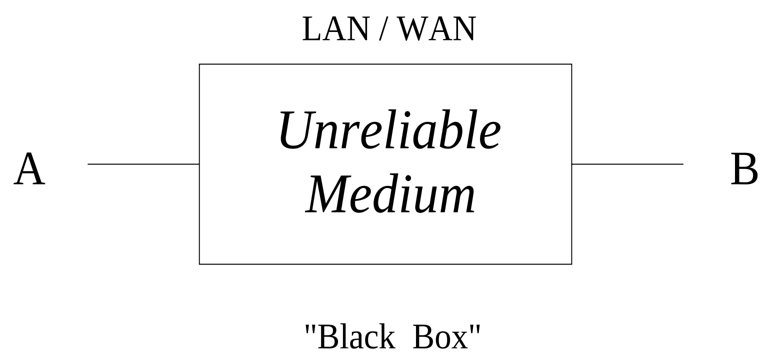
  → called end-to-end paradigm

Reliability over best-effort Internet:

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

  → implement ARQ at sender/receiver

  → router does not carry out ARQ



IP packet format:

4	4	8	16	
version	header length	TOS	total length	
fragmentation identifier			flags	fragment offset
TTL		protocol	header checksum	
source address				
destination address				
options (if any)				

- 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

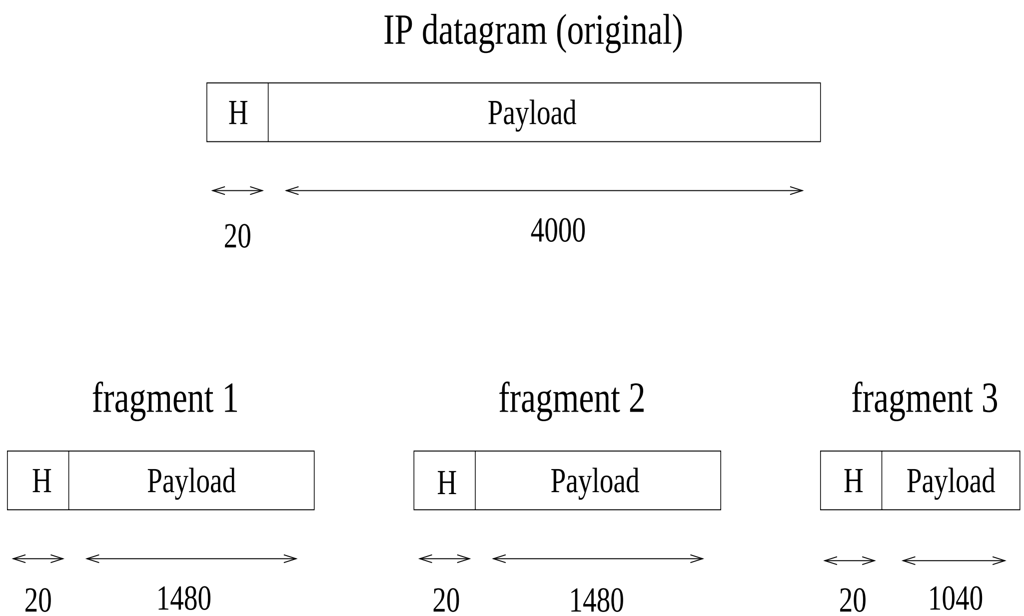
$\longrightarrow$ e.g., Ethernet 1500 B, WLAN 2313 B

- 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)



fragment ID: 900
flag bit (3rd): 1
fragment offset: 0

fragment ID: 900
flag bit (3rd): 1
fragment offset: 185

fragment ID: 900
flag bit (3rd): 0
fragment offset: 370

Note: Each fragment is an independent IP packet.

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

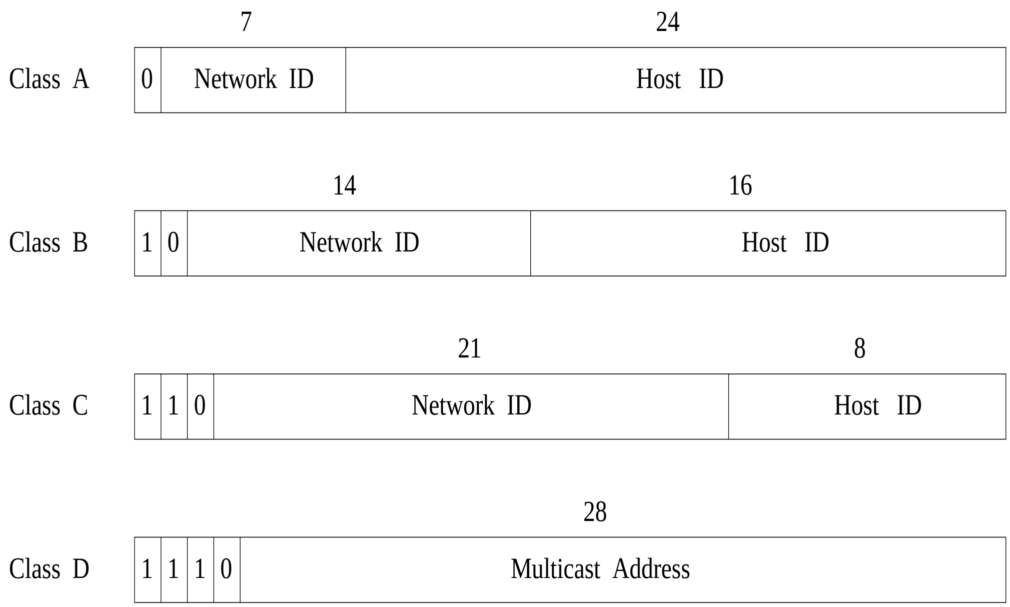
$\longrightarrow$ "all for one, one for all"

$\longrightarrow$ 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

$\longrightarrow$ network and host

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

$\longrightarrow$ single-homed vs. multi-homed

Running out of IP addresses ... or not?

$\longrightarrow$ should Purdue get a class B address?

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

$\longrightarrow$ what about Purdue's Biology Dept.?

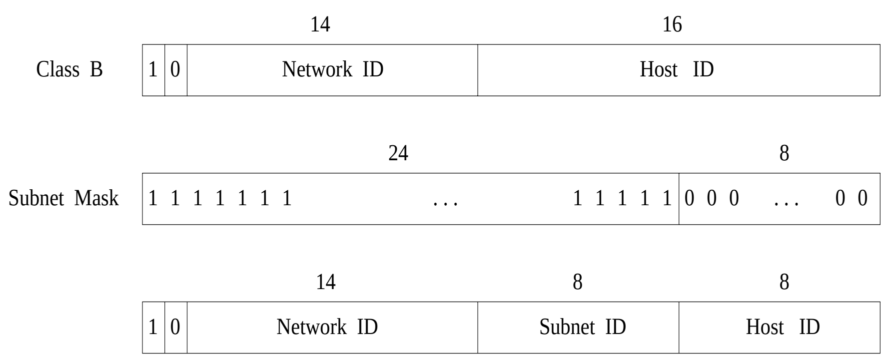
$\longrightarrow$ what about the Math Dept.?

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.

⟶ aging (standard caching technique)

Subnetting only goes so far.

- ⟶ depts. within Purdue share same class B address
- ⟶ what about your start-up company?
- ⟶ only 2^{21} class C addresses available

Other approaches to solve address depletion problem:

- IPv6

→ 128 bits (who wants it?)

→ not used much

- 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

- Dynamically assigned IP addresses

→ share an IP address pool

→ reusable

→ e.g., DHCP (dynamic host configuration protocol)

→ used by access ISPs, enterprises, etc.

- Network address translation (NAT)

→ dynamically assigned + address translation

→ private vs. public IP address

→ private: Internet routers discard them

→ e.g., 192.168.0.0 is private

→ useful for home networks, small businesses

- NATP

→ variant of NAT: use src port numbers as address bits

→ e.g.: 192.168.10.10 and 192.168.10.11 map to 128.10.27.10

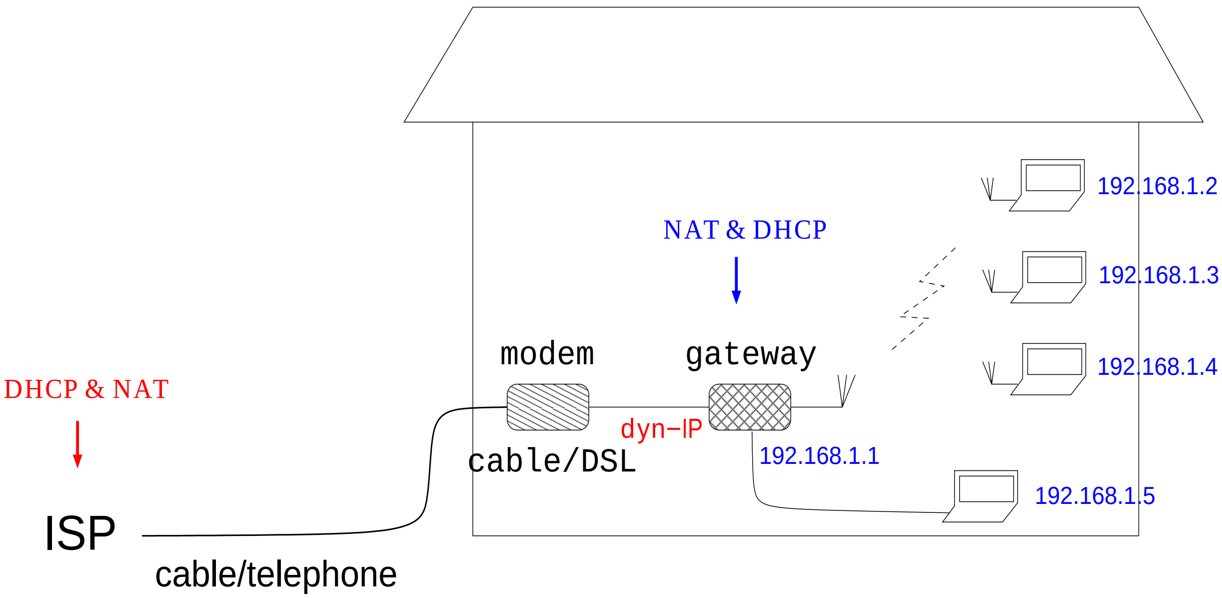
→ but 192.168.10.10 maps to 128.10.26.10:6001

→ and 192.168.10.11 maps to 128.10.26.10:6002

→ what about source port numbers?

Ex.: SOHO (small office/home office)

$\longrightarrow$ now: home networking



- dynamic IP address provided by ISP is shared through NAT
- IANA (Internet Assigned Numbers Authority)

  $\rightarrow$ non-routable: e.g., 192.168.0.0/16, 10.0.0.0/8

CIDR and dynamically assigned IP addresses with NAPT

$\longrightarrow$ significant increase of Internet's effective address space

$\longrightarrow$ saved the day (for now)!

$\longrightarrow$ perception: no pressing need for IPv6

$\longrightarrow$ can one run a web server behind at NATP box?

$\longrightarrow$ other limitations?