

## INTERNETWORKING

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)?
- How to scale?

Routing: packet- vs. circuit-switched

Circuit-switched

→ establish path and resource before packet transmission

Packet-switched

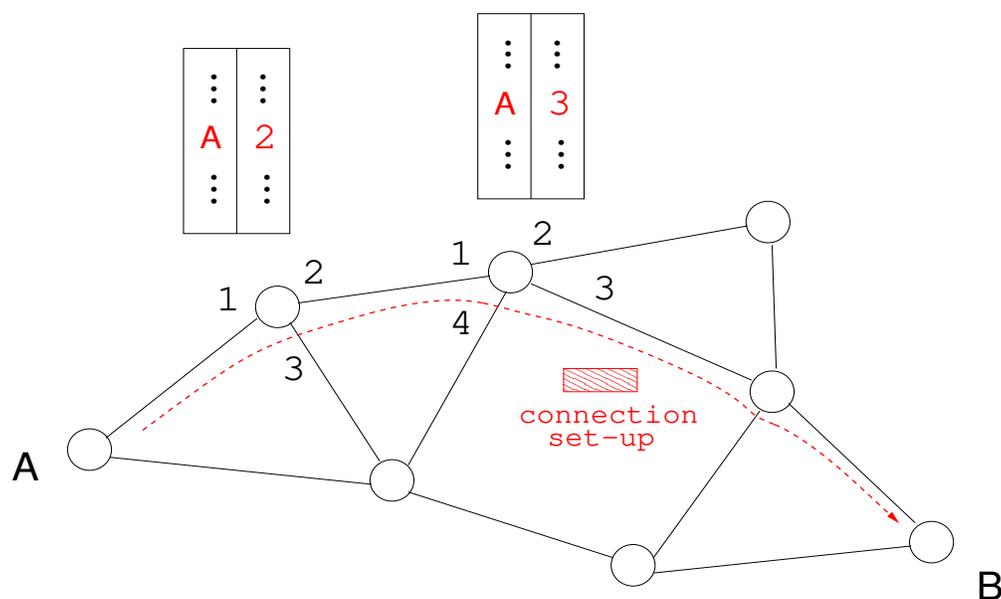
→ every packet is an autonomous entity

→ no prior set-up

Global Internet: packet-switched

Intranet of large ISPs: both packet- and circuit-switched

## Circuit-switched routing: set-up phase



- connection set-up message: signaling
  - route specification
- source tag “A” inserted into route look-up table
  - entry deletion upon session termination
  - in general: tag can be any label

Special case: packet switching with source routing

- packet itself contains path information
  - $\langle A, D, \dots, B \rangle$
- drawback: header length increases with path length
  - not good for fast packet handling
  - option still available in IP: may be used for management purposes

Destination-based forwarding:

- select output port based on destination address
- source address ignored

Negative impact stemming from ignoring source address

→ spoofing

→ source address is not actual source

By design: innocuous

→ e.g., Mobile IP, MAC address spoofing

Security vulnerability

→ tool utilized by hackers

Internet Protocol (IP):

Goals:

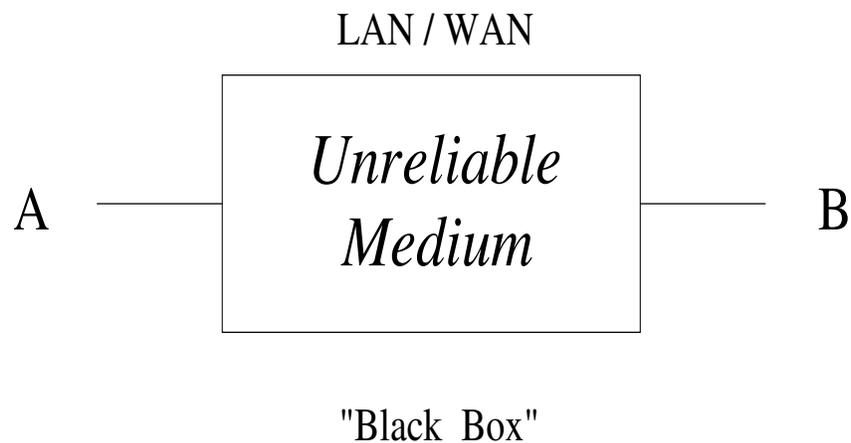
- interconnect diverse LANs into one logical entity
- implement best-effort service
  - no assurances (“what you get is what you get”)
  - simplicity is key

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
  - increases complexity of end systems (e.g., servers, laptops, handhelds)
- implement ARQ at sender/receiver



IPv4 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			
<i>options (if any)</i>			

- 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 2304 B

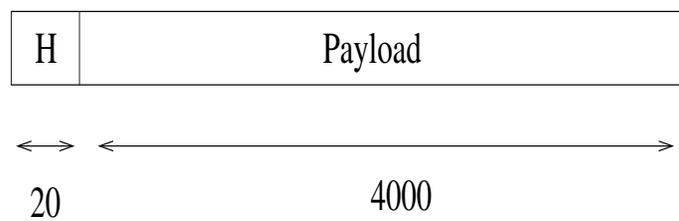
- potential size mismatch problem (IP 64 KB)
- may happen multiple times hopping from LAN to LAN

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

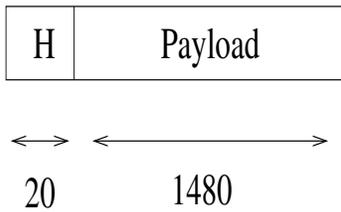
- assign unique fragmentation ID
- set 3rd flag bit if fragmentation in progress
- sequence fragments using offset in units of 8 bytes

## Example: IP fragmentation (Ethernet MTU)

### IP datagram (original)

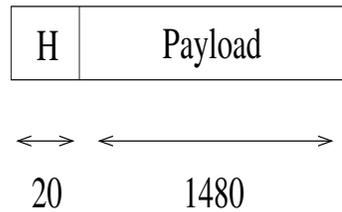


### fragment 1



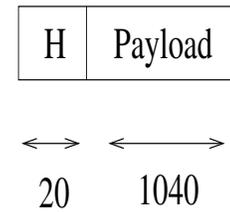
fragment ID: 900  
 flag bit (3rd): 1  
 fragment offset: 0

### fragment 2



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

### fragment 3



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.

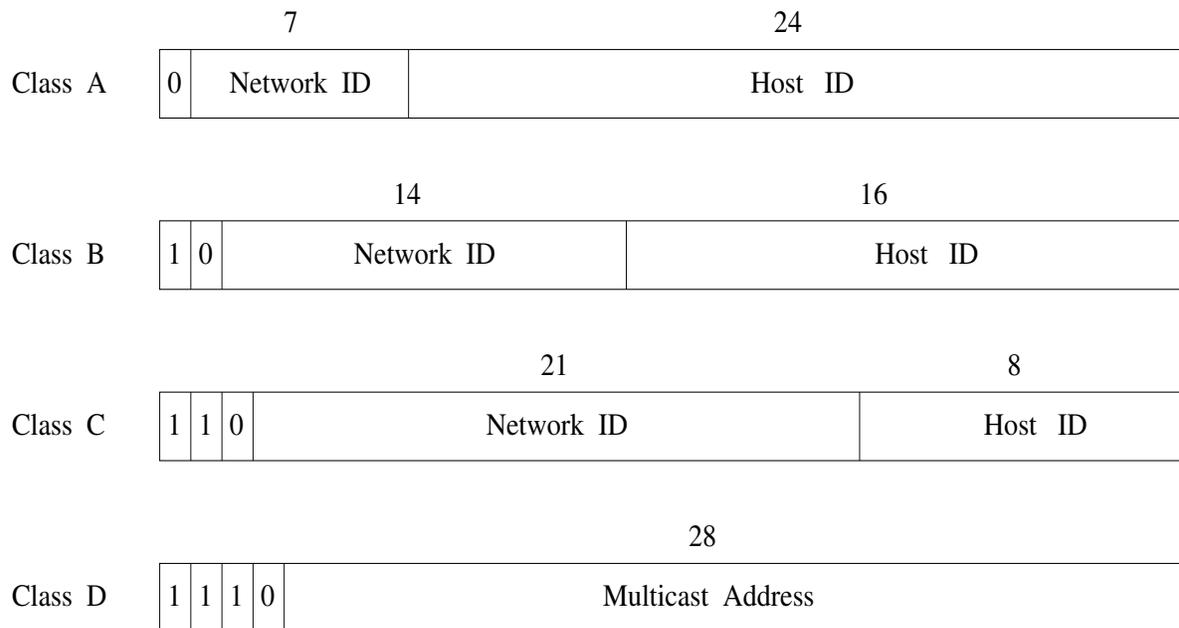
→ “all for one, one for all”

→ set 2nd flag bit to disable fragmentation

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

→ 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.

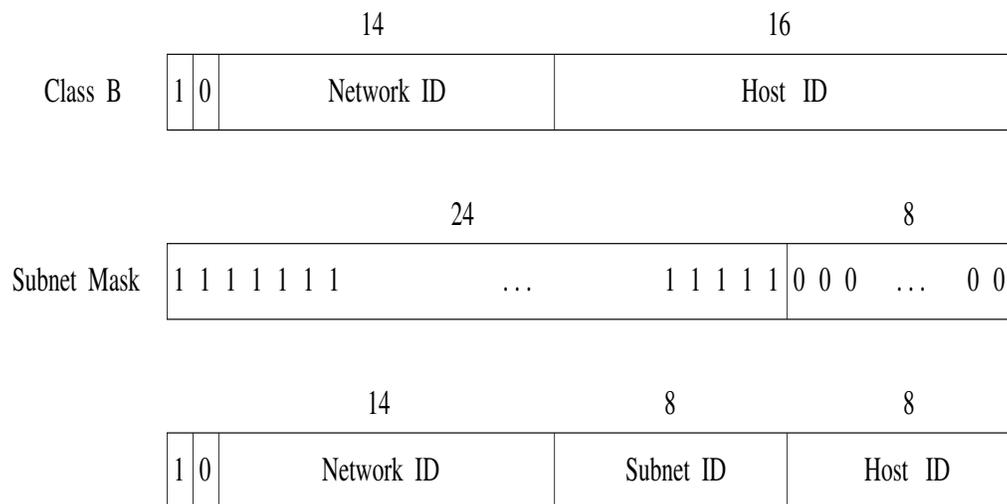
→ single-homed vs. multi-homed

Running out of IP addresses ... or not?

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

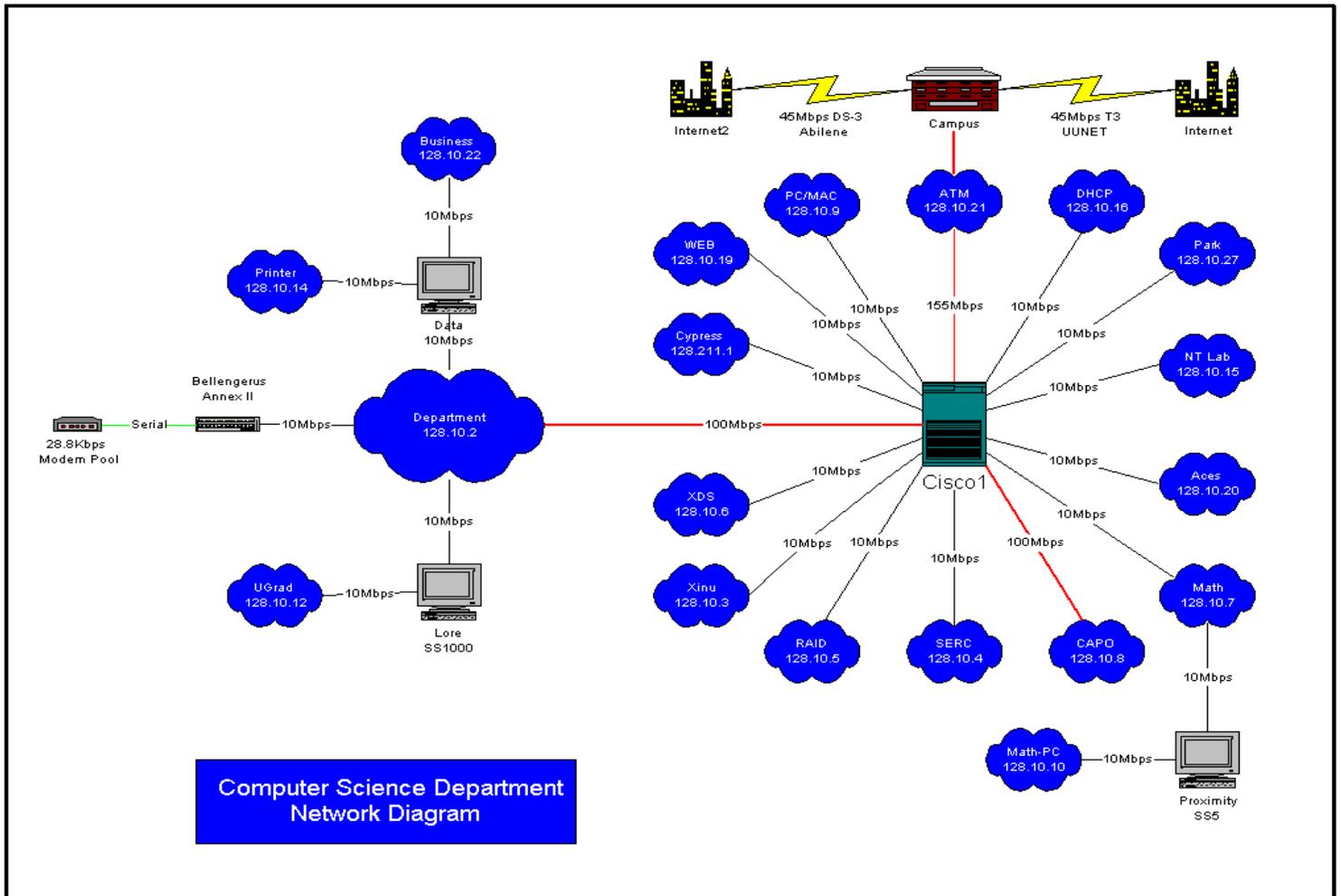


To determine subnet ID:

- AND IP address and subnet mask
  - already know if class A, B, C, or D
- 3-level hierarchy

# Example: Purdue CS network

→ a few years back



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).

- reachable by LAN address forwarding
- if not, network address (IP) forwarding

Table look-up I (“where to”):

- For each entry, compute  $SubnetID = DestAddr \text{ AND } SubnetMask$ .
- Compare  $SubnetID$  with  $SubnetID$ .
- Take forwarding action (LAN or IP).

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

→ must be done in either case

→ 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