Network Layer: Addressing

Lecture 20

http://www.cs.rutgers.edu/~sn624/352-F24

Srinivas Narayana

RUTGERS
UNIVERSITY | NEW BRUNSWICK

Software/hardware organization at hosts

Application: useful user-level functions

Transport: provide guarantees to apps

Network: best-effort global pkt delivery

Link: best-effort local pkt delivery

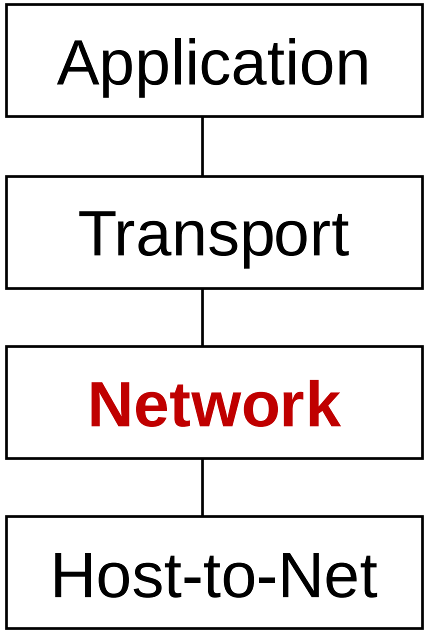
Communication functions broken up and “stacked”

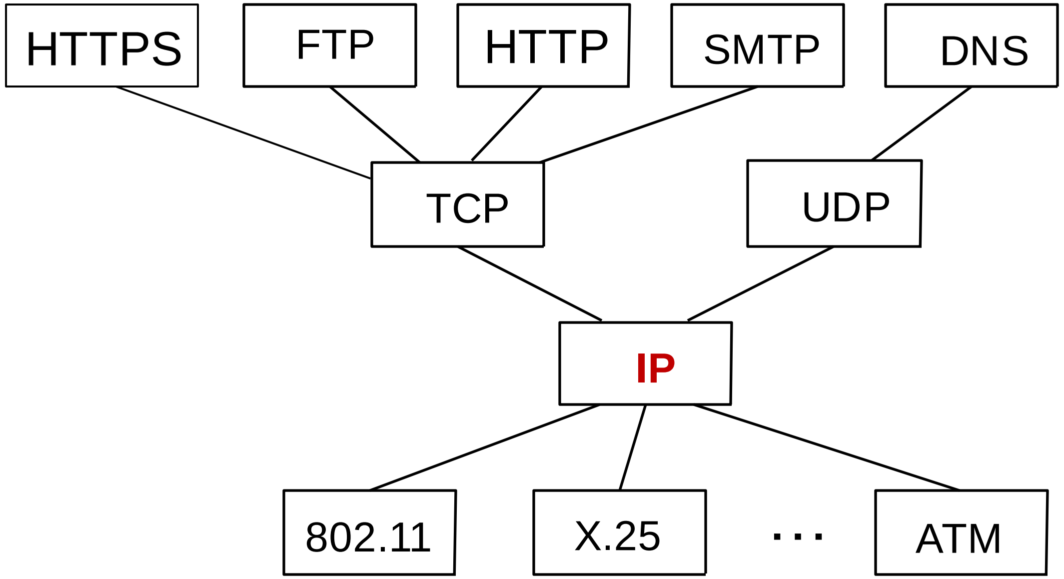
Each layer depends on the one below it.

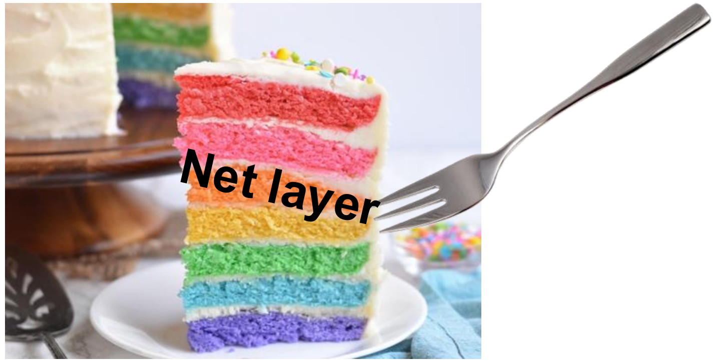
Each layer supports the one above it.

The interfaces between layers are well-defined and standardized.

Network

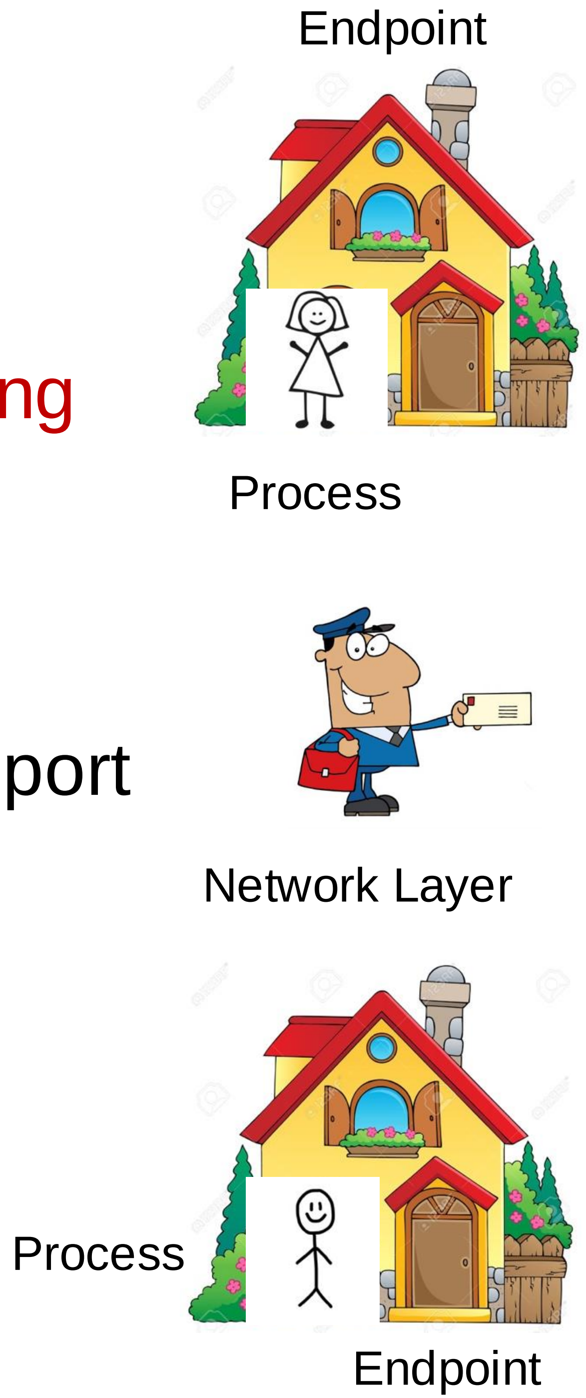

Application
Transport
Network
Host-to-Net


HTTPS
FTP
HTTP
SMTP
DNS
TCP
UDP
IP
802.11
X.25
…
ATM


Net layer

The network layer

- Main function: Move data from sending to receiving endpoint
- on sending endpoint: encapsulate transport segments into datagrams
- on receiving endpoint: deliver datagrams to transport layer
- The network layer also runs in every router
  - Implication: Often challenging to evolve network layer
- Routers examine headers on all packets passing through them

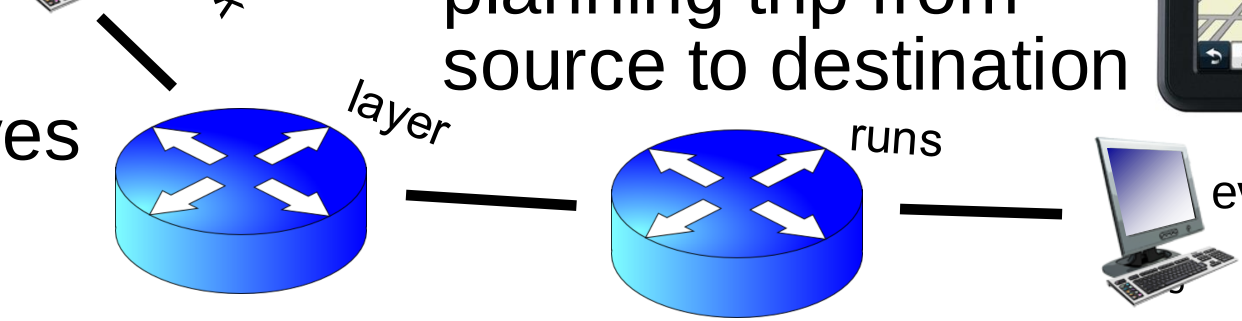


Two key network-layer functions

- Forwarding: move packets from router's input to appropriate router output

- Routing: determine route taken by packets from source to destination
  - routing algorithms

- The network layer solves the routing problem.



Analogy: taking a road trip

- Forwarding: process of getting through single interchange

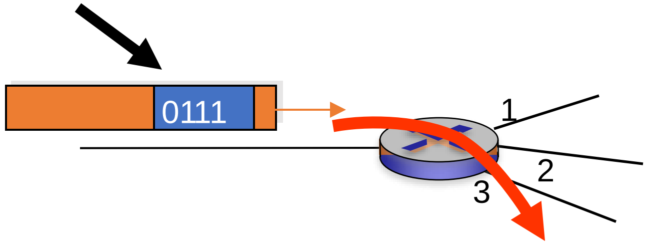
- Routing: process of planning trip from source to destination

Data plane and Control Plane

Data plane = Forwarding

- local, per-router function
- determines how datagram arriving on router input port is forwarded to router output port

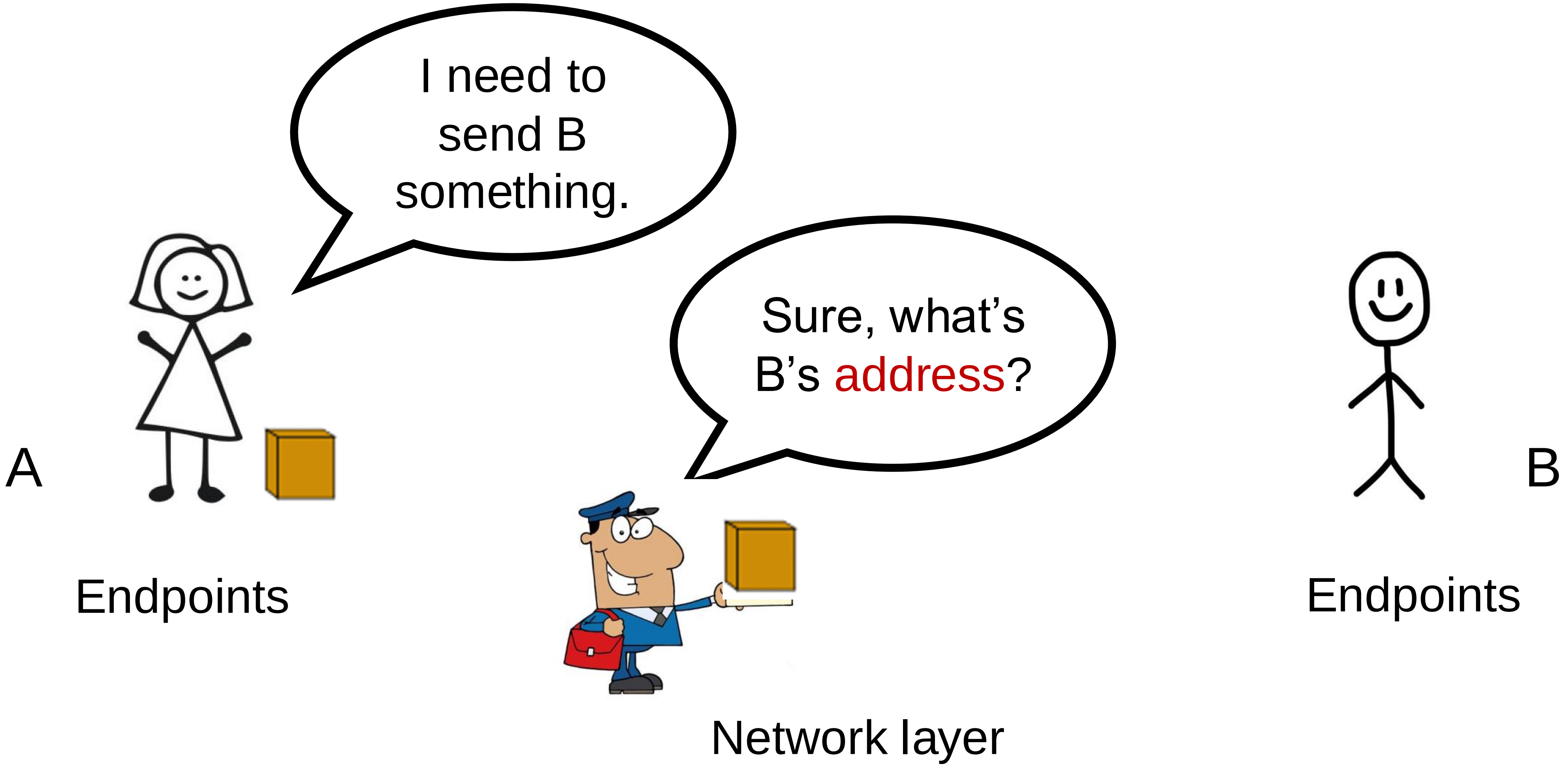
values in arriving packet header



Control plane = Routing

- network-wide logic
- determines how datagram is routed along end-to-end path from source to destination endpoint
- two control-plane approaches:
  - Distributed routing algorithm running on each router
  - Centralized routing algorithm running on a (logically) centralized server

This lecture: Addresses in the Internet



Internet Addressing

The Internet needs addresses

- Addresses are a precursor to communication.
  - Allow endpoints to locate each other
- Internet addresses are neither endpoint names nor identify them
- Addresses help routers determine how to move a packet
  - Like street address for the postal system
- Network layer addresses are designed to help routers perform the forwarding and routing functions efficiently
  - Specifically, we’ll look at Internet Protocol (IP) addresses.
  - Most popular: IP version 4 or IPv4. (later: IPv6)

IPv4 Addresses

- 32 bits long
- Identifier for a network interface
- An IP address corresponds to the point of attachment of an endpoint to the network.
- An IP address is NOT an identifier for the endpoint
  - Changes when endpoint moves
- IPv4 dotted quad notation: each byte is written in decimal in MSB order, separated by dots. Example:

10000000	11000011	00000001	01010000
128 .	195 .	1 .	80

Grouping IP addresses by prefixes

- IP addresses can be grouped based on a shared prefix of a specified length

- Example: consider two IP addresses:
  - 128.95.1.80 and 128.95.1.4
  - The addresses share a prefix of (bit) length 24: 128.95.1
  - The addresses have different suffixes of (bit) length 8

- IP addresses: prefix corresponds to the network component and the suffix to an endpoint (host) component of the address

IP addresses use hierarchy to scale routing

- IP addresses of endpoint interfaces in a network (e.g., Rutgers Busch campus) share a prefix of some length
- Each interface/endpoint has a different suffix, and hence a different 32-bit IP address
- Using prefixes reduces the amount of information needed to forward packets over the Internet
- IP prefixes are like zip codes: routers don't need to store info for each endpoint, just each prefix
- Prefixes allow IP addresses to move from one network to another (more on this later)

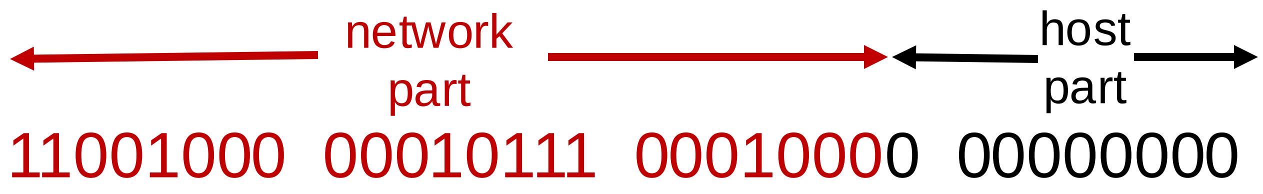
IP addresses use hierarchy to scale routing

- Postal envelopes should show clearly delineated zip codes.
- Q: How to identify the prefix from a 32-bit IP address?
  - Two methods:
- Old: Classful addressing. IP address itself is formatted to denote the IP prefix
- New: Classless addressing (also called classless inter-domain routing, or CIDR).
  - Each router independently identifies prefix from IP

Classless IPv4 addressing (CIDR)

Classless IPv4 addressing

- Also called classless inter-domain routing (CIDR)
- Key idea: Network component of the address (ie: prefix) can have any length (usually from 8—32)
- Address format: a.b.c.d/x, where x is the prefix length
  - Customary to use 0s for all suffix bits

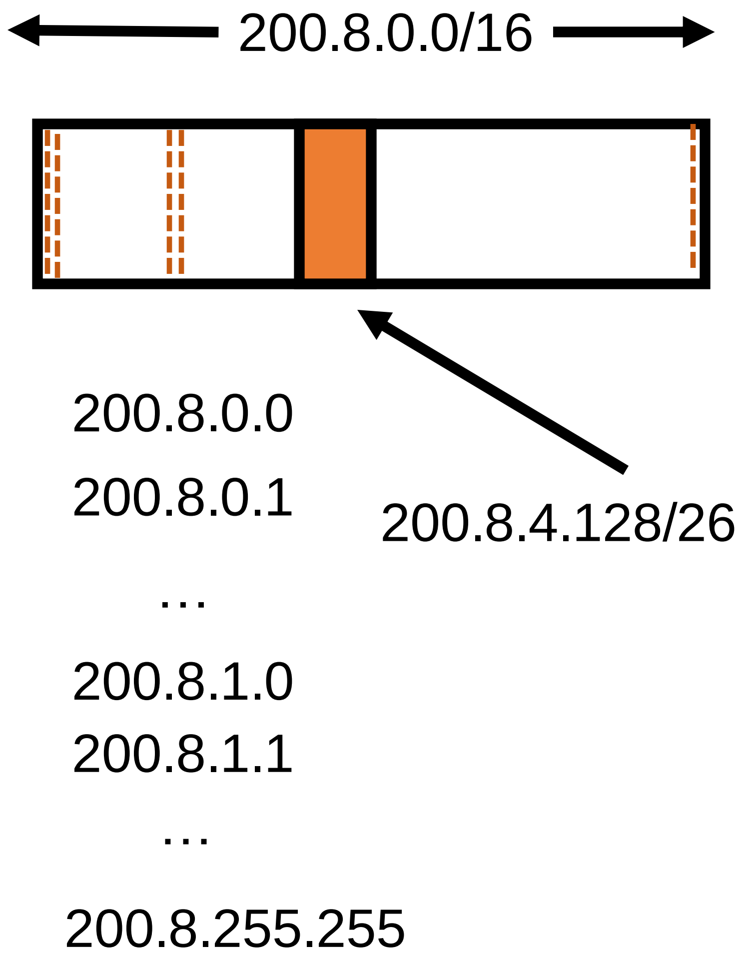


200.23.16.0/23

https://www.internetsociety.org/resources/deploy360/2015/short-guide-ip-addressing/

CIDR

- An ISP can obtain a block of addresses and partition this further to its customers
- Say an ISP has 200.8.0.0/16 address (65K addresses).
- The ISP has customer who needs only 64 addresses starting from 200.8.4.128
- Then that block can be specified as 200.8.4.128/26
- 200.8.4.128/26 is “inside” 200.8.0.0/16



Netmask (or subnet mask)

- An alternative to denote the IP prefix length of an organization
- 32 bits: a 1-bit denotes a prefix bit position. 0 denotes host bit.

network part | host part

11001000 00010111 00010000 00000000

200.23.16.0/23

network part of mask | host part of mask

11111111 11111111 11111110 00000000

Netmask: 255.255.254.0

Detecting addresses from same network

- Given IP addresses A and B, and netmask M.
    1. Compute logical AND (A & M).
    2. Compute logical AND (B & M).
    3. If (A & M) == (B & M) then A and B are on the same network.

- Ex: A = 165.230.82.52, B = 165.230.24.93, M = 255.255.128.0

- A and B are in the same network according to the netmask
- A & M == B & M == 165.230.0.0

Finding your own IP address(es)

- The old way (still works today on Mac and Linux):
  - `ifconfig -a`

- The new way using "iproute2" tools on Linux:
  - `ip link`
  - `ip addr`

- What else do you see in these outputs?