CS 352
Network Layer: Intro

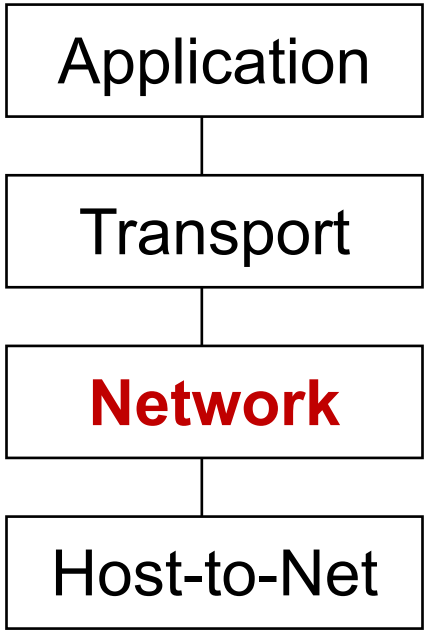
CS 352, Lecture 14.1

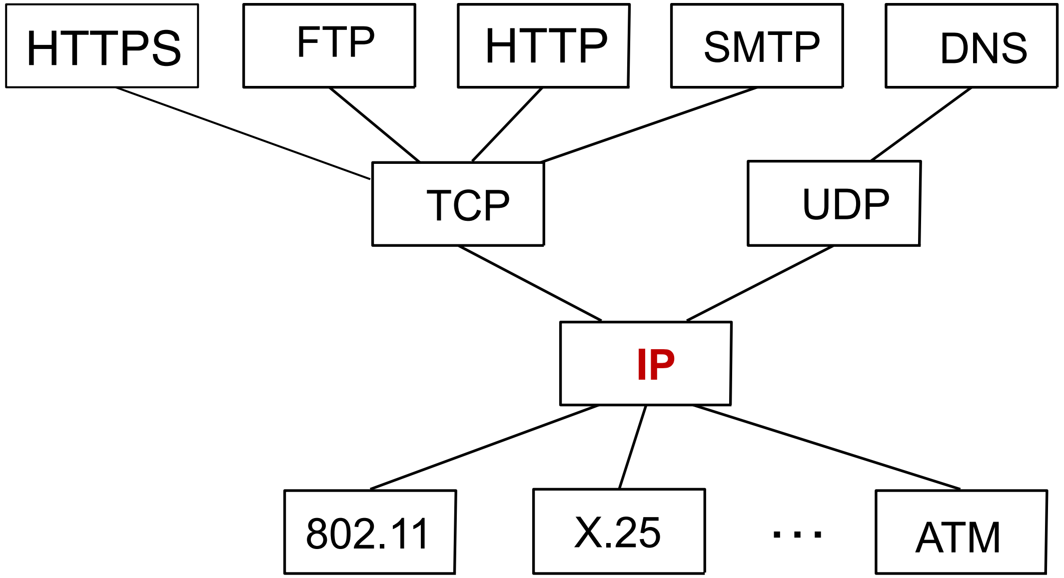
http://www.cs.rutgers.edu/~sn624/352

Srinivas Narayana

RUTGERS
UNIVERSITY | NEW BRUNSWICK

Transport


Application
Transport
Network
Host-to-Net


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

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
- The router examines header fields in all network-layer datagrams passing through it

Endpoint

Process

Network Layer

Process

Endpoint

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

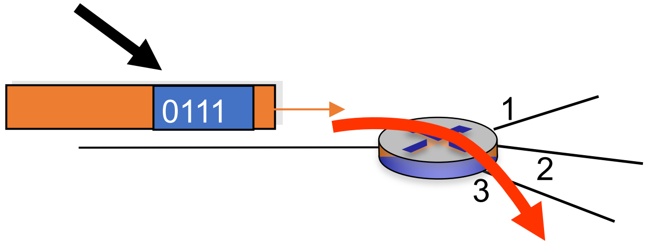
- Forwarding: process of getting through single interchange
- Routing: process of planning trip from source to destination

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

CS 352
Internet Addressing

CS 352, Lecture 14.2

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

Srinivas Narayana

RUTGERS
UNIVERSITY | NEW BRUNSWICK

The Internet needs addresses

- Addresses allow endpoints to identify, and hence talk to each other
  - E.g., like people have names
- Addresses allow routers to determine how to move a packet
  - E.g., like 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. (Coming up 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
- Dotted quad notation: each byte is written in decimal in MSB order, separated by dots. Example:

10000000 11000011 00000001 01010000

128 . 95 . 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 also allow IP addresses to be delegated from one network to another (more on this later)

IP addresses use hierarchy to scale routing

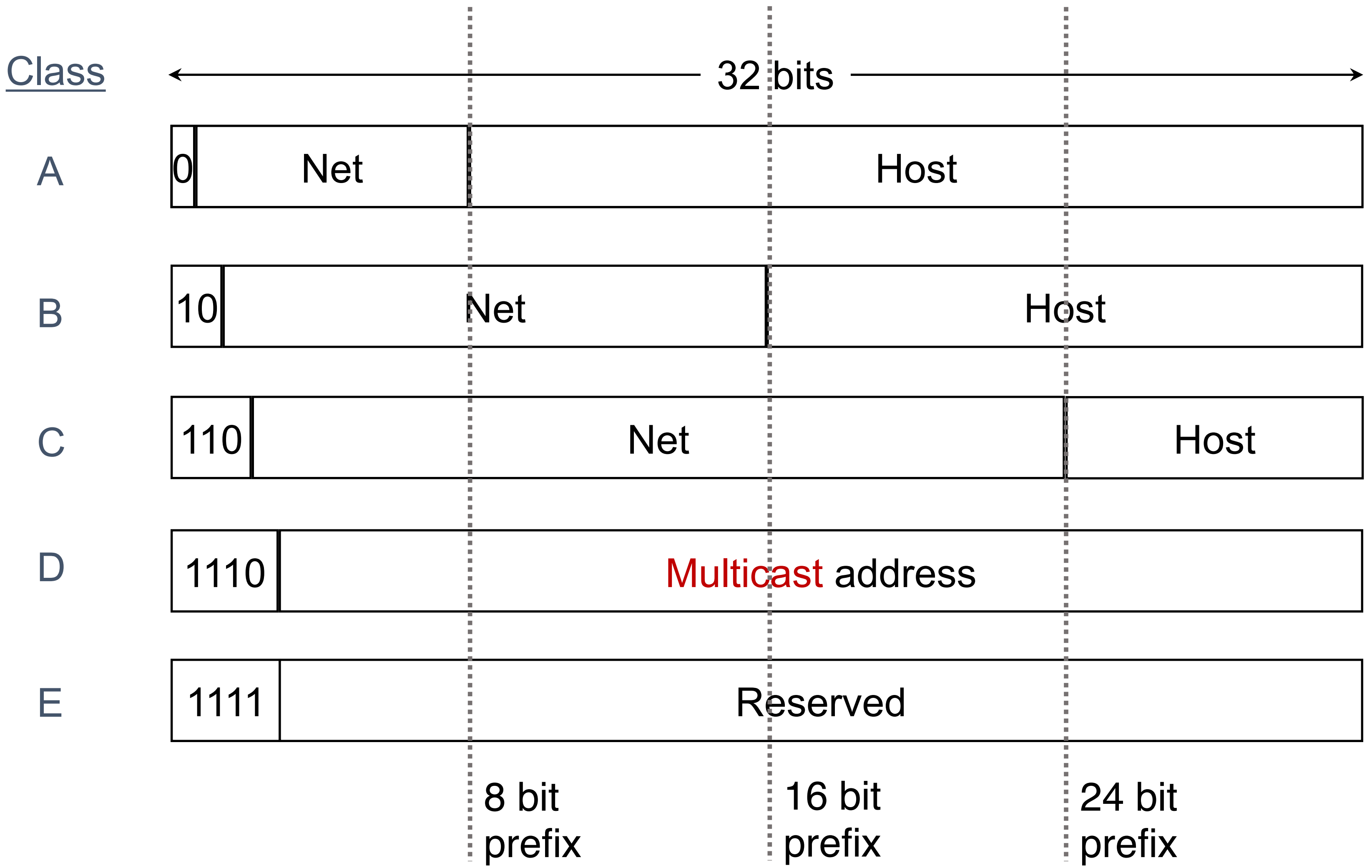
- Postal envelopes should show clearly delineated zip codes.

- Q: How to identify the prefix from a 32-bit IP address?

- Two methods:
  - Old: Classful addressing
  - New: Classless addressing (also called classless inter-domain routing, or CIDR)

Classful IPv4 addressing

Classful IPv4 addressing



0.x.x.x – 127.x.x.x

Unicast: single endpoint dest

128.x.x.x – 191.x.x.x

Unicast: single endpoint dest

192.x.x.x – 223.x.x.x

Unicast: single endpoint dest

224.x.x.x – 239.x.x.x

Destination is a group of hosts

240.x.x.x – 255.x.x.x

First octet of IP address gives you the prefix length.

Classful IPv4 addressing

- Class A:
  - For very large organizations
  - 2^{24} = 16 million hosts allowed
- Class B:
  - For large organizations
  - 2^{16} = 65 thousand hosts allowed
- Class C
  - For small organizations
  - 2^{8} = 255 hosts allowed
- Class D
  - Multicast addresses
  - No network/host hierarchy

Problems with classful addressing

- IP prefixes are allocated to organizations (e.g., Rutgers) by Internet Registry organizations (e.g., ARIN, in North America)
- Many organizations required something bigger than class C address, but smaller than a class A (or even B) address
- However, the Internet was running out of class B addresses
- Too many networks required multiple class C addresses
- Not enough nets in class A for large organizations
- Key issue: Classful addressing is too coarse-grained: The addressing strategy must allow for greater diversity of network sizes

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

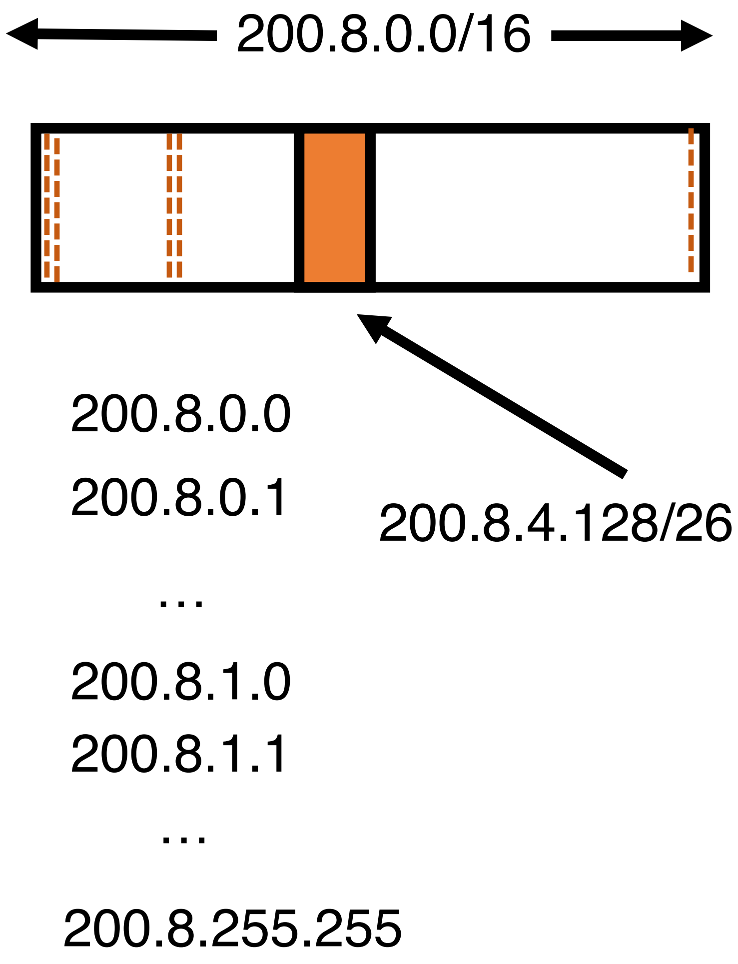
network part | host part

11001000 00010111 00010000 00000000

200.23.16.0/23

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 is the host part.

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 subnet.

- 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)

- A small demo