Internet Protocol (IP): Addressing and Forwarding

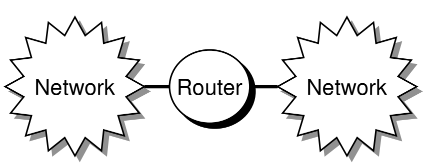
Shivkumar Kalyanaraman
Rensselaer Polytechnic Institute
shivkuma@ecse.rpi.edu
http://www.ecse.rpi.edu/Homepages/shivkuma

- **Internetworking**
- **IP Address format**
- **IP data forwarding**

Ref: RFC 791, Chap 1, 3

Internetworking

- **Internetwork = Collection of networks**
  Connected via routers



Internet = Virtual Network

- Any computer can talk to any other computer

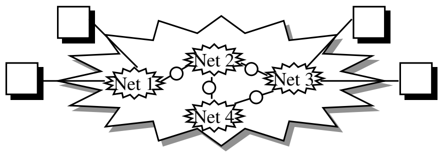


Fig 13.3

How does IP forwarding work ?

- A) *Source & Destination in same network (fig 3.3 in text)*
  - Recognize that destination IP address is on same network. [1]
  - Find the destination LAN address. [2]
  - Send IP packet encapsulated in LAN frame directly to the destination LAN address.
    - Encapsulation => source/destination IP addresses don't change

IP forwarding (contd)

- B) *Source & Destination in different networks (fig 3.4 in text)*
  - Recognize that destination IP address is not on same network. [1]
  - Look up destination IP address in a (routing) table to find a match, called the next hop router IP address.
  - Send packet encapsulated in a LAN frame to the LAN address corresponding to the IP address of the next-hop router. [2]

Addressing & Resolution

- *[1] How to find if destination is in the same network ?*
  - IP address = network ID + host ID. *Source and destination network IDs match* => *same network*
  - Splitting address into multiple parts is called hierarchical addressing
- *[2]: How to find the LAN address corresponding to an IP address ?*
  - Address Resolution Problem.
  - Solution: ARP, RARP (next chapter)

Route Table Lookup

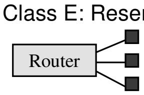
- Intermediate routers lookup only the destination network-ID,
  - Deliver datagrams to next-hop and finally to destination network, not to host directly
  - Hierarchical forwarding: routing tables scale.

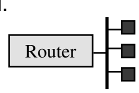
Table at R2:

Destination	Next Hop
Net 1	Forward to R1
Net 2	Deliver Direct
Net 3	Deliver Direct
Net 4	Forward to R3

IP Address Formats

- Class A: | 0 | Network | Host | — 1, 7, 24 bits
- Class B: | 10 | Network | Host | — 2, 14, 16 bits
- Class C: | 110 | Network | Host | — 3, 21, 8 bits
- Class D: | 1110 | Multicast Group addresses | — 4, 28 bits
  - Class E: Reserved.





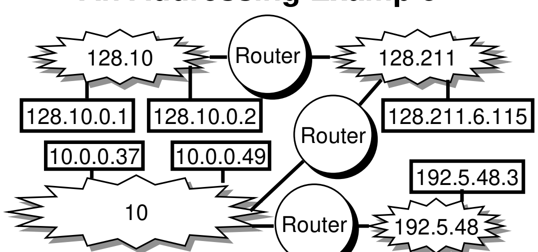
Dotted Decimal Notation

- **Binary: 11000000 00000101 00110000 00000011**
  Hex Colon: C0:05:30:03
  Dotted Decimal: 192.5.48.3

Class	Range
A	0 through 127
B	128 through 191
C	192 through 223
D	224 through 239
E	240 through 255

Also see fig 1.6, 1.7 in text

An Addressing Example



- **All hosts on a network have the same network prefix (I.e. network ID)**

Some special IP addresses

- **All-0s ⇒ This computer**
- **All-1s ⇒ All hosts on this net (*limited broadcast*: *don't forward out of this net)***
- **All-0 *host suffix* ⇒ Network Address ('*0'* means '*this'*)**
- **All-1 *host suffix* ⇒ All hosts on the destination net (directed broadcast).**
- **127.*.*.* ⇒ Loopback through IP layer**
- **Further classification in fig 3.9 of text**

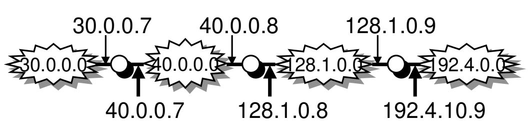
Subnet Addressing

- Classful addressing inefficient: Everyone wants class B addresses
- Can we split class A, B addresses spaces and accommodate more networks ?
  - Need another level of hierarchy. Defined by "subnet mask" (eg: fig 3.5)
  - External routers send to "network" specified by the "network ID" and have smaller routing tables

Subnet Addressing (Contd)

- Internal routers & hosts use subnet mask to identify "subnet ID" and route packets between "subnets" within the "network".
- Eg: Mask: 255.255.255.0 => subnet ID = 8 bits with upto 62 hosts/subnet
- Route table lookup:
  - IF ((Mask[i] & Destination Addr) = = Destination[i])
    Forward to NextHop[i]

Route Table Lookup: Example



Destination	Mask	Next Hop
30.0.0.0	255.0.0.0	40.0.0.7
40.0.0.0	255.0.0.0	Deliver direct
128.1.0.0	255.255.0.0	Deliver direct
192.4.10.0	255.255.255.0	128.1.0.9

Summary

- Addressing:
  - Unique IP address per interface
  - Classful (A,B,C) => address allocation not efficient
  - Hierarchical => smaller routing tables
  - Provision for broadcast, multicast, loopback addresses
  - Subnet masks allow "subnets" within a "network" => improved address allocation efficiency
  - Problem: Host moves between networks => IP address changes.

Summary (contd)

- Forwarding:
  - Simple *"next-hop"* forwarding.
  - Last hop forwards directly to destination
  - *Best-effort delivery :* No error reporting. Delay, out-of-order, corruption, and loss possible => problem of higher layers!
  - Forwarding vs routing: Routing tables setup by separate algorithm (s)

UNIX networking commands

*Section 1, tools and commands. (* denotes most relevant)*

- telnet telnet (1) - user interface to a remote system using the TELNET protocol
- ftp ftp (1) - file transfer program
- tftp tftp (1) - trivial file transfer program
- *ifconfig ifconfig (1m) - configure network interface parameters
- in.rdisc in.rdisc (1m) - network router discovery daemon
- *in.routed in.routed (1m) - network routing daemon

UNIX Networking commands

- *netstat netstat (1m) - show network status
- *ping ping (1m) - send ICMP ECHO_REQUEST packets to network hosts
- snoop snoop (1m) - capture and inspect network packets
- nslookup nslookup (1m) - query name servers interactively
- *tcpdump tcpdump (1) - dump traffic on a network
- *route route (1m) - manually manipulate the routing tables
- in.named in.named (1m) - Internet domain name server

UNIX networking commands

- in.tftpd in.tftpd (1m) - Internet Trivial File Transfer Protocol server
- tftpd in.tftpd (1m) - Internet Trivial File Transfer Protocol server
- whois whois (1) - Internet user name directory service
- *inetd inetd (1m) - Internet services daemon
- *arp arp (1m) - address resolution display and control

Section 4, related configuration files (All are pretty relevant)

- netconfig netconfig (4) - network configuration database

UNIX Networking commands

- netmasks netmasks (4) - network mask database
- networks networks (4) - network name database
- routing routing (4) - system support for packet network routing
- services services (4) - Internet services and aliases
- protocols protocols (4) - protocol name database
- inetd.conf inetd.conf (4) - Internet servers database
- nsswitch.conf nsswitch.conf (4) - configuration file for the name service switch

UNIX Networking commands

- bootparams: bootparams (4) - boot parameter data base

Section 7, protocol description. (All are pretty relevant)

- *ICMP icmp (7p) - Internet Control Message Protocol
- *IP ip (7p) - Internet Protocol
- *TCP tcp (7p) - Internet Transmission Control Protocol
- *UDP udp (7p) - Internet User Datagram Protocol
- *if_tcp if_tcp (7p) - general properties of Internet Protocol network interfaces
- *inet inet (7p) - Internet protocol family
- *arp arp (7p) - Address Resolution Protocol