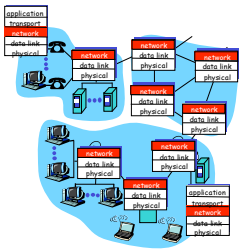
# Network Layer Overview and IP



## Network layer

- transport segment from sending to receiving host
- on sending side encapsulates segments into datagrams
- on rcving side, delivers segments to transport layer
- network layer protocols in *every* host, router
- Router examines header fields in all IP datagrams passing through it



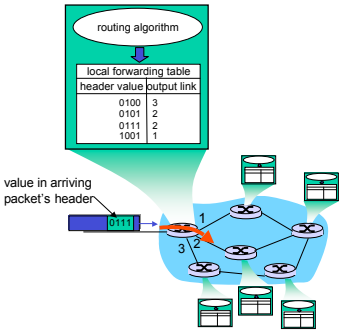



## Key Network-Layer Functions

- *forwarding:* move packets from router's input to appropriate router output
- *routing:* determine route taken by packets from source to dest.
  - *Routing algorithms*



## Interplay between routing and forwarding






## Network service model

Example services for individual datagrams:

- guaranteed delivery
- Guaranteed delivery with less than 40 msec delay

Example services for a flow of datagrams:

- In-order datagram delivery
- Guaranteed minimum bandwidth to flow
- Restrictions on changes in inter-packet spacing



## Virtual circuits

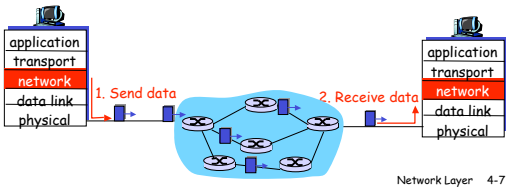
"source-to-dest path behaves much like telephone circuit"

- performance-wise
- network actions along source-to-dest path

- call setup, teardown for each call *before* data can flow
- each packet carries VC identifier (not destination host address)
- *every* router on source-dest path maintains "state" for each passing connection
- link, router resources (bandwidth, buffers) may be *allocated* to VC

## Datagram networks

- no call setup at network layer
- routers: no state about end-to-end connections
  - no network-level concept of "connection"
- packets forwarded using destination host address
  - packets between same source-dest pair may take different paths
- Why is this OK for the Internet?






## Forwarding table

4 billion possible entries

| Destination Address Range | Link Interface |
|---|---|
| 11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111 | 0 |
| 11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111 | 1 |
| 11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111 | 2 |
| otherwise | 3 |



## Longest prefix matching

| Prefix Match | Link Interface |
|---|---|
| 11001000 00010111 00010 | 0 |
| 11001000 00010111 00011000 | 1 |
| 11001000 00010111 00011 | 2 |
| otherwise | 3 |

Examples

DA: 11001000 00010111 00010110 10100001 Which interface?

DA: 11001000 00010111 00011000 10101010 Which interface?



## Datagram or VC network: why?

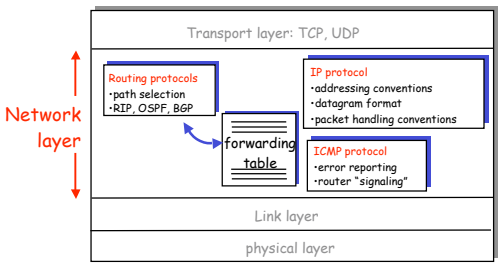
Internet
- data exchange among computers
  - "elastic" service, no strict timing req.
- "smart" end systems (computers)
  - can adapt, perform control, error recovery
  - simple inside network, complexity at "edge"
- many link types
  - different characteristics
  - uniform service difficult

ATM
- evolved from telephony
- human conversation:
  - strict timing, reliability requirements
  - need for guaranteed service
- "dumb" end systems
  - telephones
  - complexity inside network



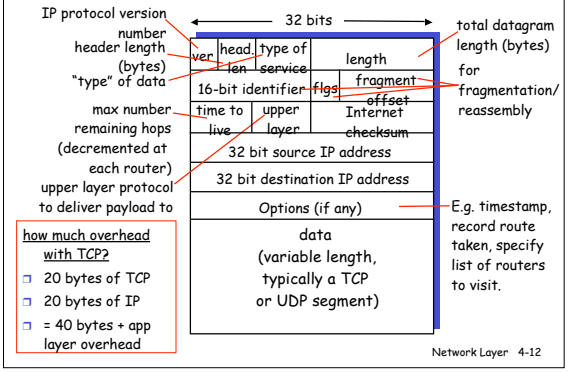
## The Internet Network layer

Host, router network layer functions:





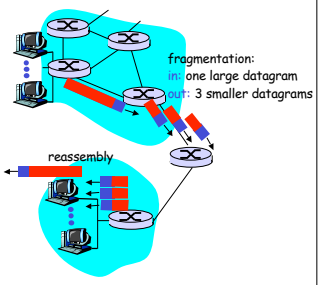

## IP datagram format




how much overhead with TCP?
- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

## IP Fragmentation & Reassembly

- network links have MTU (max.transfer size) - largest possible link-level frame.
  - different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify, order related fragments

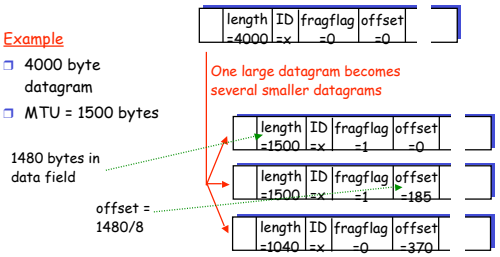





## IP Fragmentation and Reassembly

Example

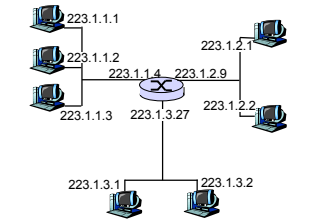
- 4000 byte datagram
- MTU = 1500 bytes






## IP Addressing: introduction

- IP address: 32-bit identifier for host, router *interface*
- *interface:* connection between host/router and physical link
  - router's typically have multiple interfaces
  - host may have multiple interfaces
  - IP addresses associated with each interface



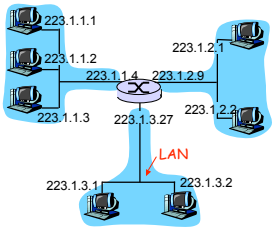

223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1



## Subnets

- IP address:
  - subnet part (high order bits)
  - host part (low order bits)
- *What's a subnet ?*
  - device interfaces with same subnet part of IP address
  - can physically reach each other without intervening router



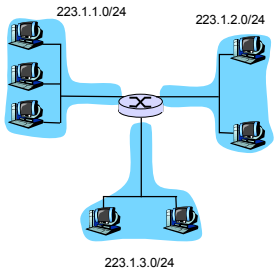

network consisting of 3 subnets



## Subnets

Recipe

- To determine the subnets, detach each interface from its host or router, creating islands of isolated networks. Each isolated network is called a subnet.

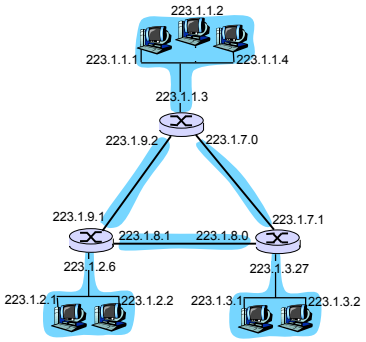



Subnet mask: /24

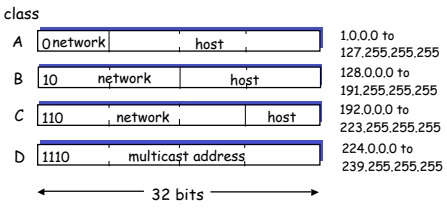


## Subnets

How many?



## Classful Addressing






## IP addressing: CIDR

CIDR: Classless InterDomain Routing

- subnet portion of address of arbitrary length
- address format: a.b.c.d/x, where x is # bits in subnet portion of address


200.23.16.0/23



## IP addresses: how to get one?

Q: How does *host* get IP address?

- hard-coded by system admin in a file
  - /etc/hosts
- DHCP: Dynamic Host Configuration Protocol: dynamically get address from as server
  - "plug-and-play"

(more in next chapter)



## IP addresses: how to get one?

Q: How does *network* get subnet part of IP addr?

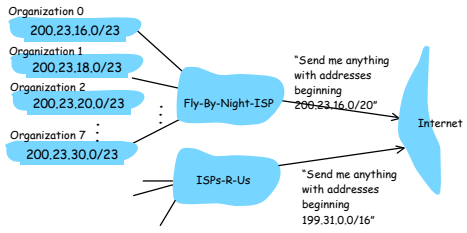
A: gets allocated portion of its provider ISP's address space

| | | |
|---|---|---|
| ISP's block | 11001000 00010111 00010000 00000000 | 200.23.16.0/20 |
| Organization 0 | 11001000 00010111 00010000 00000000 | 200.23.16.0/23 |
| Organization 1 | 11001000 00010111 00010010 00000000 | 200.23.18.0/23 |
| Organization 2 | 11001000 00010111 00010100 00000000 | 200.23.20.0/23 |
| ... | .... | .... |
| Organization 7 | 11001000 00010111 00011110 00000000 | 200.23.30.0/23 |



## Hierarchical addressing: route aggregation

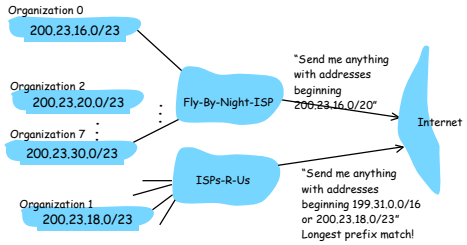
Hierarchical addressing allows efficient advertisement of routing information:






## Hierarchical addressing: more specific routes

ISPs-R-Us has a more specific route to Organization 1



## IP addressing: the last word...

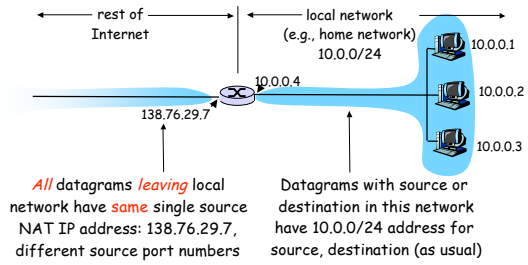
Q: How does an ISP get block of addresses?

A: ICANN: Internet Corporation for Assigned Names and Numbers

- allocates addresses
- manages DNS
- assigns domain names, resolves disputes



## NAT: Network Address Translation






## NAT: Network Address Translation

- Motivation: local network uses just one IP address as far as outside word is concerned:
  - no need to be allocated range of addresses from ISP: - just one IP address is used for all devices
  - can change addresses of devices in local network without notifying outside world
  - can change ISP without changing addresses of devices in local network
  - devices inside local net not explicitly addressable, visible by outside world (a security plus).



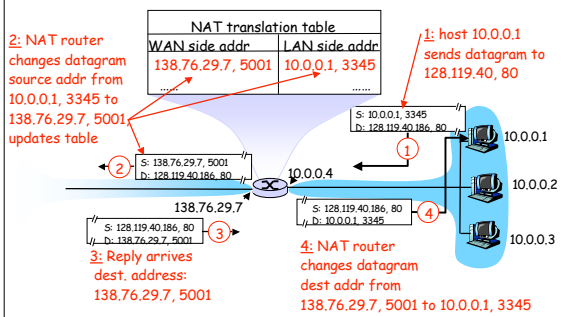
## NAT: Network Address Translation

Implementation: NAT router must:

- *outgoing datagrams: replace* (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
  . . . remote clients/servers will respond using (NAT IP address, new port #) as destination addr.
- *remember (in NAT translation table)* every (source IP address, port #) to (NAT IP address, new port #) translation pair
- *incoming datagrams: replace* (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table



## NAT: Network Address Translation






## NAT: Network Address Translation

- 16-bit port-number field:
  - 60,000 simultaneous connections with a single LAN-side address!
- NAT is controversial:
  - routers should only process up to layer 3
  - violates end-to-end argument
    - NAT possibility must be taken into account by app designers, eg, P2P applications
  - address shortage should instead be solved by IPv6

## ICMP: Internet Control Message Protocol

- used by hosts & routers to communicate network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- network-layer "above" IP:
  - ICMP msgs carried in IP datagrams
- ICMP message: type, code plus first 8 bytes of IP datagram causing error

| Type | Code | description |
|---|---|---|
| 0 | 0 | echo reply (ping) |
| 3 | 0 | dest. network unreachable |
| 3 | 1 | dest host unreachable |
| 3 | 2 | dest protocol unreachable |
| 3 | 3 | dest port unreachable |
| 3 | 6 | dest network unknown |
| 3 | 7 | dest host unknown |
| 4 | 0 | source quench (congestion control - not used) |
| 8 | 0 | echo request (ping) |
| 9 | 0 | route advertisement |
| 10 | 0 | router discovery |
| 11 | 0 | TTL expired |
| 12 | 0 | bad IP header |



## Traceroute and ICMP

- Source sends series of UDP segments to dest
  - First has TTL =1
  - Second has TTL=2, etc.
  - Unlikely port number
- When nth datagram arrives to nth router:
  - Router discards datagram
  - And sends to source an ICMP message (type 11, code 0)
  - Message includes name of router& IP address
- When ICMP message arrives, source calculates RTT
- Traceroute does this 3 times

Stopping criterion

- UDP segment eventually arrives at destination host
- Destination returns ICMP "host unreachable" packet (type 3, code 3)
- When source gets this ICMP, stops.



## IPv6

- Initial motivation: 32-bit address space soon to be completely allocated.
- Additional motivation:
  - header format helps speed processing/forwarding
  - header changes to facilitate QoS

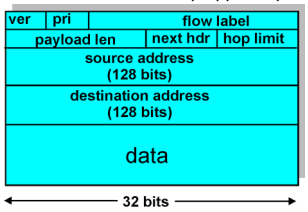
IPv6 datagram format:

- fixed-length 40 byte header
- no fragmentation allowed



## IPv6 Header (Cont)

*Priority:* identify priority among datagrams in flow
*Flow Label:* identify datagrams in same "flow."
(concept of"flow" not well defined).
*Next header:* identify upper layer protocol for data






## Other Changes from IPv4

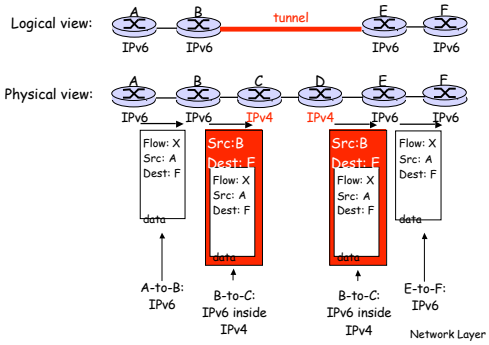
- *Checksum:* removed entirely to reduce processing time at each hop
- *Options:* allowed, but outside of header, indicated by "Next Header" field
- *ICMPv6:* new version of ICMP
  - additional message types, e.g. "Packet Too Big"
  - multicast group management functions



## Transition From IPv4 To IPv6

- Not all routers can be upgraded simultaneous
  - no "flag days"
  - How will the network operate with mixed IPv4 and IPv6 routers?
- Dual-stack
- *Tunneling:* IPv6 carried as payload in IPv4 datagram among IPv4 routers

# Tunneling




Network Layer 4-37