

## Summary of Last Lecture

- UDP
  - No connection management
  - No flow or error control
  - No guarantee in-order packet delivery
- Why UDP?
- UDP checksum
  - Error detection
  - No action taken

Network Layer 4-1

## Summary of Last Lecture

- TCP Frame Fields
  - Header bits
    - ACK
    - SYN
    - FIN
- TCP connection establishment
  - Three-way handshake
- TCP connection tear-down
  - Two double handshakes

Network Layer 4-2

## Summary of Last Lecture

- Timeout → TCP retransmission
- Round Trip Time (RTT)
- Retransmission Timeout Interval (RTO)
  - Smoothed RTT
  - Karn's algorithm
  - Jacobson/Karels Algorithm

Network Layer 4-3

## Summary of Last Lecture

- TCP flow control
  - Window size (sliding window, receiver window)
- TCP congestion control
  - Sliding window (receiver flow control)
  - Congestion window (sender flow control)
  - Threshold (sender's slow start vs. linear mode line)

Network Layer 4-4

## Summary of Last Lecture

- TCP slow start
  - Double the congestion window's volume each RTT
  - After each timeout, go back to slow start
- TCP congestion avoidance
  - Linear increase
  - Only increment by a fraction of MSS for every received ACK
  - For the whole RTT, increase 1 MSS to the congestion window

Network Layer 4-5

## Summary of Last Lecture

- TCP Fast Retransmission
  - Based on 3 duplicate ACKs
- TCP Fast Recovery
  - Don't do a slow start
  - Modified algorithm

Network Layer 4-6

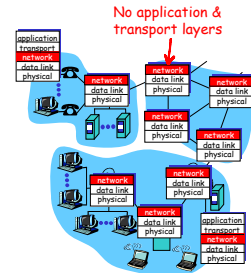
## Summary of Last Lecture

- Why is TCP fair?
  - Achieve efficiency and equal bandwidth
- TCP delay modeling
  - Fixed congestion window
  - With slow start

Network Layer 4-7

## Network layer

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



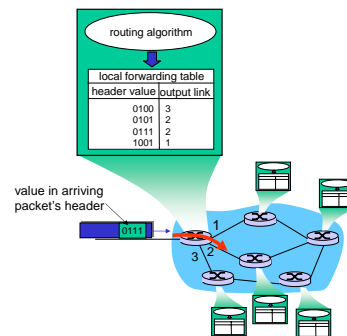
Network Layer 4-8

## 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 dest.
    - *routing algorithms*
- analogy:**
- *routing*: process of planning trip from source to dest
  - *forwarding*: process of getting through single interchange

Network Layer 4-9

## Interplay between routing and forwarding



Network Layer 4-10

## Connection setup

- 3<sup>rd</sup> important function in *some* network architectures:
  - ATM, frame relay, X.25
- before datagrams flow, two end hosts *and* intervening routers establish virtual connection
  - routers get involved
- network vs transport layer connection service:
  - **network**: between two hosts (may also involve intervening routers in case of VCs)
  - **transport**: between two processes

Network Layer 4-11

## Network service model

Q: What *service model* for "channel" transporting datagrams from sender to receiver?

- |  |   |
|--|---|
| <p><b>Example services for individual datagrams:</b></p> <ul style="list-style-type: none"> <li>□ guaranteed delivery</li> <li>□ guaranteed delivery with less than 40 msec delay</li> </ul> | <p><b>Example services for a flow of datagrams:</b></p> <ul style="list-style-type: none"> <li>□ in-order datagram delivery</li> <li>□ guaranteed minimum bandwidth to flow</li> <li>□ restrictions on changes in inter-packet spacing</li> </ul> |
|--|---|

Network Layer 4-12

## Network layer service models:

Network Architecture	Service Model	Guarantees ?				Congestion feedback
		Bandwidth	Loss	Order	Timing	
Internet	best effort	none	no	no	no	no (inferred via loss)
ATM	CBR	constant rate	yes	yes	yes	no congestion
ATM	VBR	guaranteed rate	yes	yes	yes	no congestion
ATM	ABR	guaranteed minimum	no	yes	no	yes
ATM	UBR	none	no	yes	no	no

Network Layer 4-13

## Network layer connection and connection-less service

- datagram network provides network-layer connectionless service
- VC network provides network-layer connection service
- analogous to the transport-layer services, but:
  - **service:** host-to-host
  - **no choice:** network provides one or the other
  - **implementation:** in network core

Network Layer 4-14

## 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 (dedicated resources = predictable service)

Network Layer 4-15

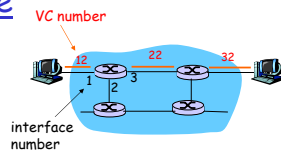
## VC implementation

a VC consists of:

1. path from source to destination
  2. VC numbers, one number for each link along path
  3. entries in forwarding tables in routers along path
- packet belonging to VC carries VC number (rather than dest address)
  - VC number can be changed on each link.
    - New VC number comes from forwarding table

Network Layer 4-16

## Forwarding table



Forwarding table in northwest router:

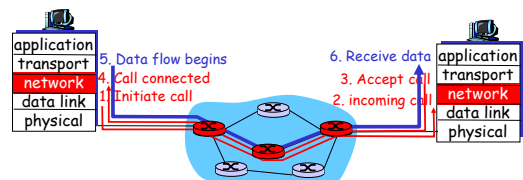
Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

Routers maintain connection state information!

Network Layer 4-17

## Virtual circuits: signaling protocols

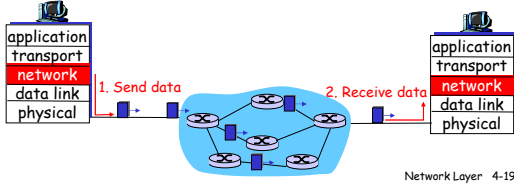
- used to setup, maintain, teardown VC
- used in ATM, frame-relay, X.25
- not used in today's Internet



Network Layer 4-18

## 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



## 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

Network Layer 4-20

## 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?

Network Layer 4-21

## Datagram or VC network: why?

### Internet (datagram)

- 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 (VC)

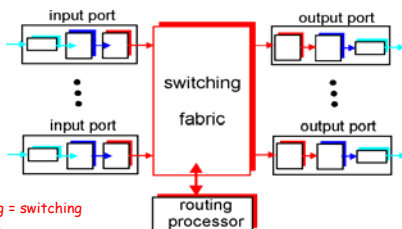
- evolved from telephony
- human conversation:
  - strict timing, reliability requirements
  - need for guaranteed service
- "dumb" end systems
  - telephones
  - complexity inside network

Network Layer 4-22

## Router Architecture Overview

### Two key router functions:

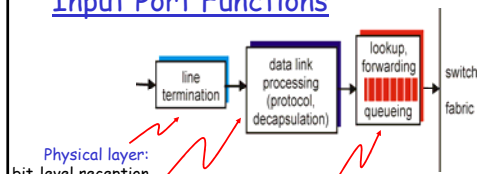
- run routing algorithms/protocol (RIP, OSPF, BGP)
- forwarding datagrams from incoming to outgoing link



Forwarding = switching (synonyms)

Network Layer 4-23

## Input Port Functions



Physical layer: bit-level reception

Data link layer: e.g., Ethernet see chapter 5

### Decentralized switching:

- given datagram dest., lookup output port using forwarding table in input port memory
- goal: complete input port processing at line speed
- queuing: if datagrams arrive faster than forwarding rate into switch fabric

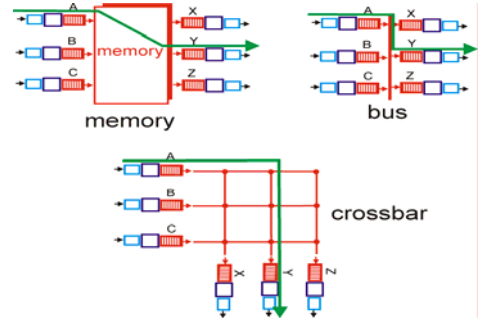
Network Layer 4-24

## High speed routers

- Input port processing at line speed (lookup is performed in less than the amount of time needed to receive a packet)
- OC48 link: 2.5 Gbps, 256 bytes packets -> 1000000 lookups per second.
- Forwarding table stored in a tree data structure
- Content addressable memory (CAM): for 32 bit IP address it returns the associated forwarding table entry in constant time
- Keep recently accessed forwarding table entries in a cache
- Table entry compression
- Log(# of bits) algorithm

Network Layer 4-25

## Three types of switching fabrics

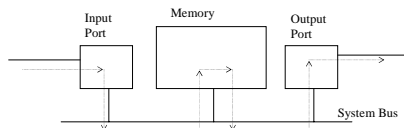


Network Layer 4-26

## Switching Via Memory

### First generation routers:

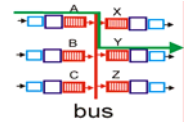
- traditional computers with switching under direct control of CPU
- packet copied to system's memory
- speed limited by memory bandwidth (2 bus crossings per datagram)



E.g. Cisco Catalyst 8500, BayNetworks Accelar 1200

Network Layer 4-27

## Switching Via a Bus



- datagram from input port memory to output port memory via a shared bus
- bus contention:** switching speed limited by bus bandwidth
- 1 Gbps bus, Cisco 1900: sufficient speed for access and enterprise routers (not regional or backbone)

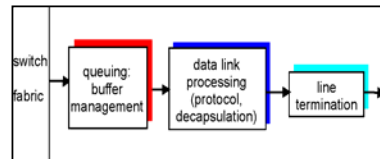
Network Layer 4-28

## Switching Via An Interconnection Network

- overcome bus bandwidth limitations
- Banyan networks, other interconnection nets initially developed to connect processors in multiprocessor
- Advanced design: fragmenting datagram into fixed length cells, switch cells through the fabric.
- Cisco 12000: switches Gbps through the interconnection network

Network Layer 4-29

## Output Ports

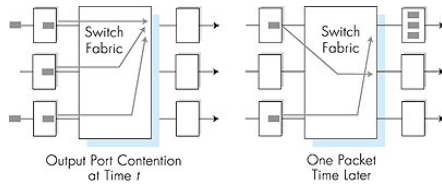


- Buffering** required when datagrams arrive from fabric faster than the transmission rate
- Scheduling discipline** chooses among queued datagrams for transmission

First come first served (FCFS), weighted fair queuing (WFQ)

Network Layer 4-30

## Output port queuing

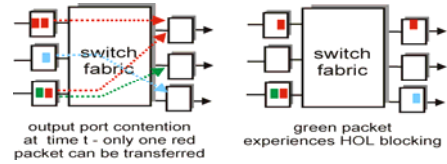


- buffering when arrival rate via switch exceeds output line speed
- *queuing (delay) and loss due to output port buffer overflow!*

Network Layer 4-31

## Input Port Queuing

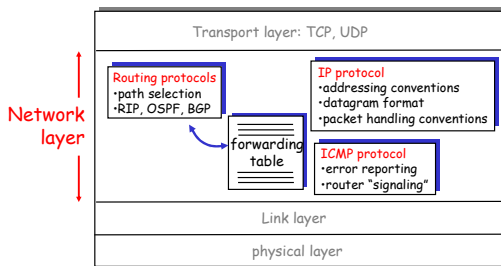
- Fabric slower than input ports combined → queuing may occur at input queues
- **Head-of-the-Line (HOL) blocking:** queued datagram at front of queue prevents others in queue from moving forward
- *queuing delay and loss due to input buffer overflow!*



Network Layer 4-32

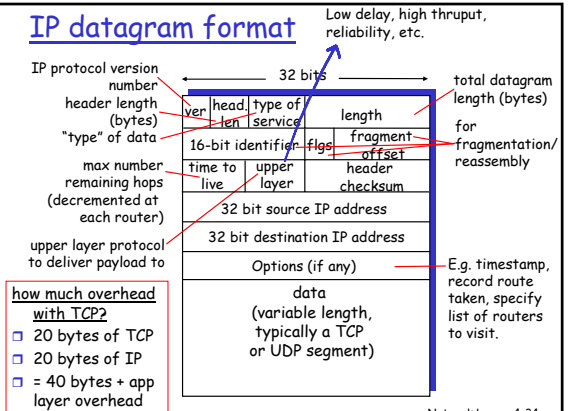
## The Internet Network layer

Host, router network layer functions:



Network Layer 4-33

## IP datagram format



Network Layer 4-34

## IP Packet Fields

- Version
  - The IP version number (currently 4)
- IHL
  - IP Header Length in 32-bit words
- Type of Service
  - Contains priority information, rarely used
- Total Length
  - The total length of the datagram in bytes
  - Includes header

Network Layer 4-35

## IP Packet Fields (cont'd)

- Identification
  - When an IP packet is segmented into multiple fragments, each fragment is given the same identification
  - This field is used to reassembly fragments
- DF
  - Don't Fragment
- MF
  - More Fragments
  - When a packet is fragmented, all fragments except the last one have this bit set

Network Layer 4-36

## IP Packet Fields (cont'd)

- Fragment offset
  - The fragment's position within the original packet
- Time to Live
  - Hop count, decremented each time the packet reaches a new router
  - When hop count = 0, packet is discarded
- Protocol
  - Identifies which transport layer protocol is being used for this packet
- Header Checksum
  - Verifies the contents of the IP header
  - Not polynomial-based

Network Layer 4-37

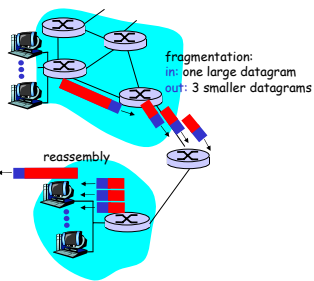
## IP Packet Fields (cont'd)

- Source and Destination Addresses
  - Uniquely identify sender and receiver of the packet
- Options
  - Up to 40 bytes in length
  - Used to extend functionality of IP
  - Examples: source routing, security, record route

Network Layer 4-38

## 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



Network Layer 4-39

## IP Fragmentation and Reassembly

### Example

- 4000 byte datagram
- MTU = 1500 bytes

length	ID	fragflag	offset
=4000	=x	=0	=0

One large datagram becomes several smaller datagrams

1480 bytes in data field  
offset = 1480/8

length	ID	fragflag	offset
=1500	=x	=1	=0
length	ID	fragflag	offset
=1500	=x	=1	=185
length	ID	fragflag	offset
=1040	=x	=0	=370

Network Layer 4-40

## IP Addresses

- 32 bits long
- Notation:
  - Each byte is written in decimal in MSB order, separated by decimals
  - Example: 128.195.1.80
- Special Address
  - Loopback (to self) address is 127.0.0.1
  - Broadcast is all 1's (255.255.255.255)

Network Layer 4-41

## IP Address Classes

- Class A:
  - For very large organizations
  - 16 million hosts allowed
- Class B:
  - For large organizations
  - 65 thousand hosts allowed
- Class C:
  - For small organizations
  - 255 hosts allowed
- Class D:
  - Multicast addresses
  - No network/host hierarchy

Network Layer 4-42

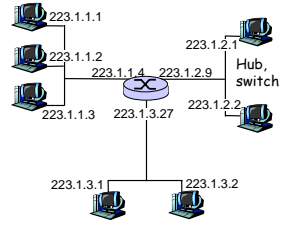
## IP Address Hierarchy

- Note that Class A, Class B, and Class C addresses only support two levels of hierarchy
- Each address contains a network and a host portion, meaning two levels of hierarchy
- However, the host portion can be further split into "subnets" by the address class owner
- This allows for more than 2 levels of hierarchy

Network Layer 4-43

## IP Addressing

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

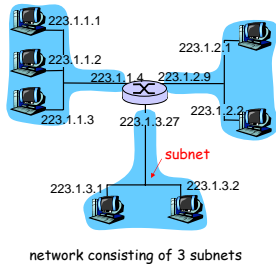


223.1.1.1 =  $\underbrace{11011111}_{223} \underbrace{00000001}_1 \underbrace{00000001}_1 \underbrace{00000001}_1$

Network Layer 4-44

## 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

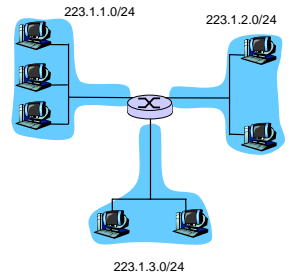


Network Layer 4-45

## Subnets

### Recipe

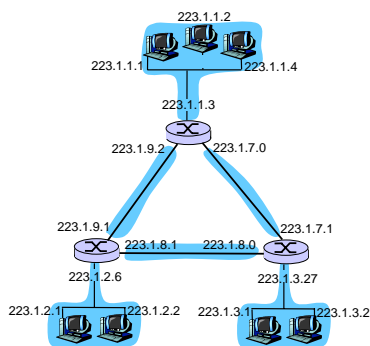
- To determine the subnets, detach each interface from its host or router, creating islands of isolated networks. Each isolated network is called a **subnet**.



Network Layer 4-46

## Subnets

How many?



Network Layer 4-47

## Subnetting

Example: Class B address with 8-bit subnetting

	16 bits	8 bits	8 bits
	Network id	Subnet id	Host id
Example Address:	165.230	.24	.8

Network Layer 4-48

## Subnet Masks

Subnet masks allow hosts to determine if another IP address is on the same subnet or the same network

	16 bits	8 bits	8 bits
	Network id	Subnet id	Host id
Mask:	1111111111111111	11111111	00000000
	255.255	.255	.0

Network Layer 4-49

## Subnet Masks (cont'd)

Assume IP addresses A and B share subnet mask M.  
Are IP addresses A and B on the same subnet?

1. Compute (A and M).
2. Compute (B and M).
3. If (A and M) = (B and M) then A and B are on the same subnet.

Example: A and B are class B addresses

A = 165.230.82.52  
B = 165.230.24.93  
M = 255.255.255.0

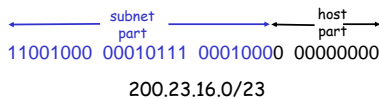
Same network?  
Same subnet?

Network Layer 4-50

## IP addressing: CIDR

### CIDR: Classless InterDomain Routing

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



Network Layer 4-51

## IP addresses: how to get one?

**Q:** How does *host* get IP address?

- o hard-coded by system admin in a file
    - o Wintel: control-panel->network->configuration->tcp/ip->properties
    - o UNIX: /etc/rc.config
  - o **DHCP:** Dynamic Host Configuration Protocol: dynamically get address from as server
    - o "plug-and-play"
- (more in next chapter)

Network Layer 4-52

## IP addresses: how to get one?

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

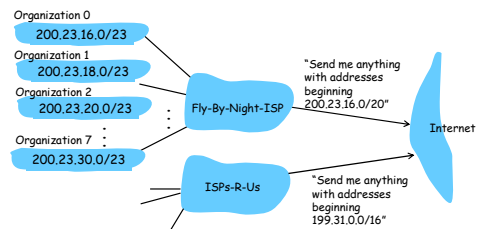
**A:** gets allocated portion of its provider ISP's address space

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

Network Layer 4-53

## Hierarchical addressing: route aggregation

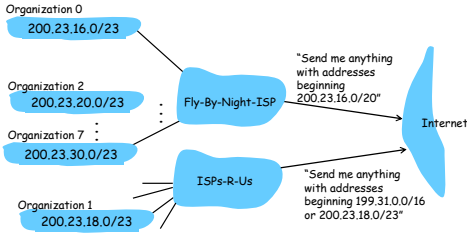
Hierarchical addressing allows efficient advertisement of routing information:



Network Layer 4-54

## Hierarchical addressing: more specific routes

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



Network Layer 4-55

## IP addressing: the last word...

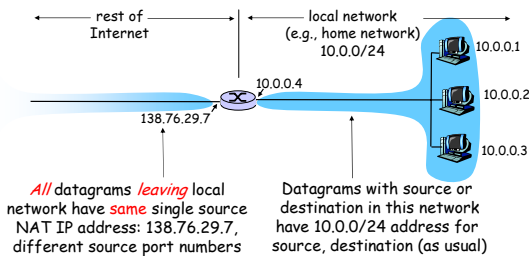
**Q:** How does an ISP get block of addresses?

**A:** **ICANN:** Internet Corporation for Assigned Names and Numbers

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

Network Layer 4-56

## NAT: Network Address Translation



Network Layer 4-57

## NAT: Network Address Translation

- **Motivation:** local network uses just one IP address as far as outside world is concerned:
  - range of addresses not needed from ISP: just one IP address 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).

Network Layer 4-58

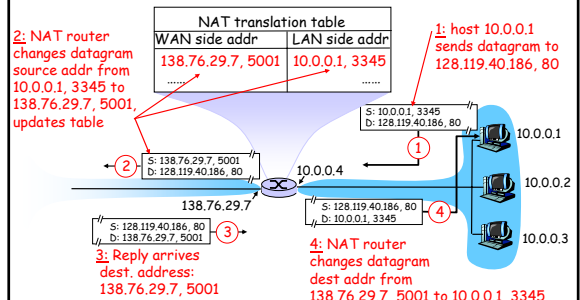
## NAT: Network Address Translation

**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

Network Layer 4-59

## NAT: Network Address Translation



Network Layer 4-60

## 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

Network Layer 4-61

## ICMP: Internet Control Message Protocol

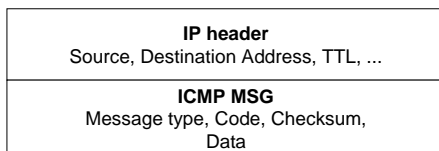
- used by hosts & routers to communicate network-level information
 

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

  - 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

Network Layer 4-62

## ICMP MSG



Message type examples (Figure 6.3 in Stevens book):

- 0 (8) echo request (reply)
- 3 destination unreachable
- 4 source quench
- 11 time exceeded

Network Layer 4-63

## Specific uses of ICMP

- Echo request reply
  - Can be used to check if a host is alive
- Address mask request/reply
  - Learn the subnet mask
- Destination unreachable
  - Invalid address and/or port
- TTL expired
  - Routing loops, or too far away

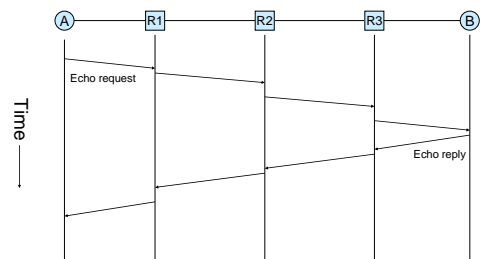
Network Layer 4-64

## Ping

- Uses ICMP echo request/reply
- Source sends ICMP echo request message to the destination address
  - Echo request packet contains sequence number and timestamp
- Destination replies with an ICMP echo reply message containing the data in the original echo request message
- Source can calculate round trip time (RTT) of packets
- If no echo reply comes back then the destination is unreachable

Network Layer 4-65

## Ping (cont'd)



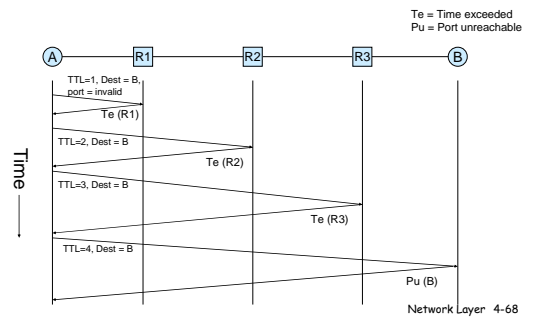
Network Layer 4-66

## Traceroute

- Traceroute records the route that packets take
- A clever use of the TTL field
- When a router receives a packet, it decrements TTL
- If TTL=0, it sends an ICMP time exceeded message back to the sender
- To determine the route, progressively increase TTL
  - Every time an ICMP time exceeded message is received, record the sender's (router's) address
  - Repeat until the destination host is reached or an error message occurs

Network Layer 4-67

## Traceroute (cont'd)



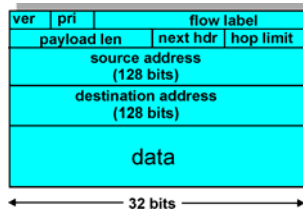
## IPv6

- **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

Network Layer 4-69

## 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



Network Layer 4-70

## IPv6 fields

- Version = 6
- Traffic class: Similar to TOS
- Flow label (20 bit) identifies flow of datagrams
- Payload length (16bit)
- Next header: identifies upper protocol (TCP, UDP, etc)
- Hop limit: similar to time to live

Network Layer 4-71

## New:

- No fragmentation (instead: packet is too big)
- No header Checksum
- No options (fixed length)

Network Layer 4-72

## 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

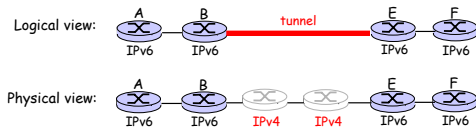
Network Layer 4-73

## 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?
- ❑ **Tunneling:** IPv6 carried as payload in IPv4 datagram among IPv4 routers

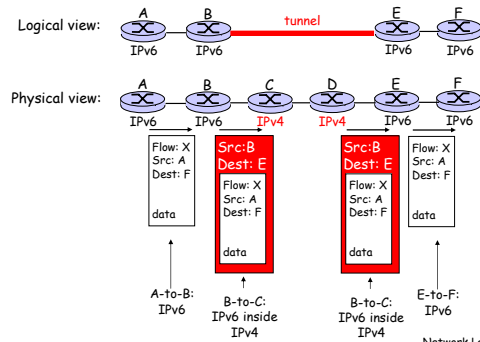
Network Layer 4-74

## Tunneling



Network Layer 4-75

## Tunneling



Network Layer 4-76