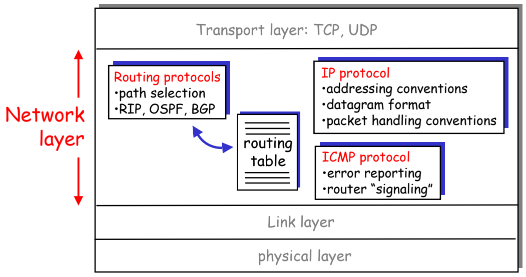
8: IP Basics

Last Modified:
3/5/2003 2:11:15 PM

The Internet Network layer

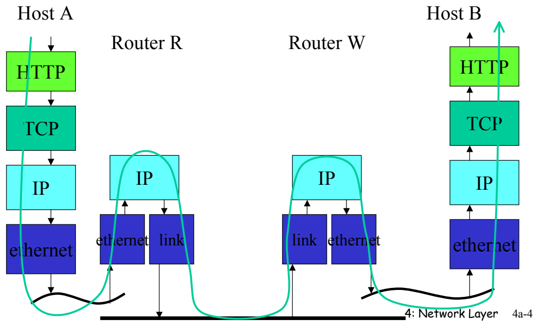
Host, router network layer functions:



Internet Protocol

- The Internet is a network of **heterogeneous** networks:
  - using different technologies (ex. different maximum packet sizes)
  - belonging to different administrative authorities (ex. Willing to accept packets from different addresses)
- Goal of IP: interconnect all these networks so can send end to end without any knowledge of the intermediate networks
- Routers: machines to forward packets between heterogeneous networks

Protocol stack: packet forwarding

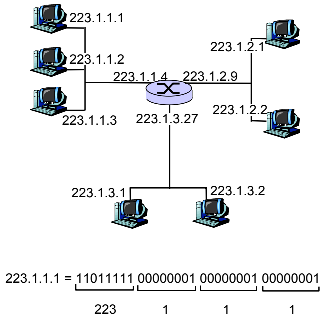


IP Addressing

- IP address:
  - 32 bits
  - network part (high order bits)
  - host part (low order bits)
  - Defined by class of IP address?
  - Defined by subnet mask?

IP Address Per Interface

- IP address: 32-bit identifier for host, router *interface*
- *interface:* connection between host and physical link
  - router's must have multiple interfaces
  - host may have multiple interfaces
  - IP addresses (unicast addresses) associated with interface, not host, router

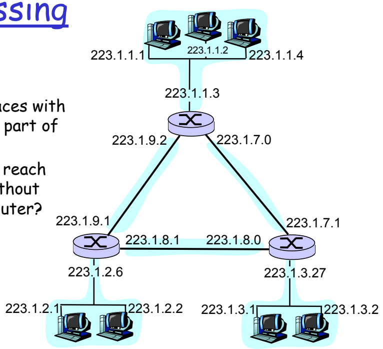


IP Addressing

How to find the networks?

- device interfaces with same network part of IP address?
- can physically reach each other without intervening router?

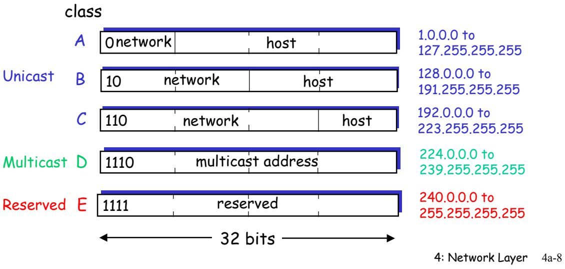
Interconnected system consisting of six "networks" or one network (223.1.*.*)?



IP Addresses (Classes)

given notion of "network", let's re-examine IP addresses:

"class-full" addressing



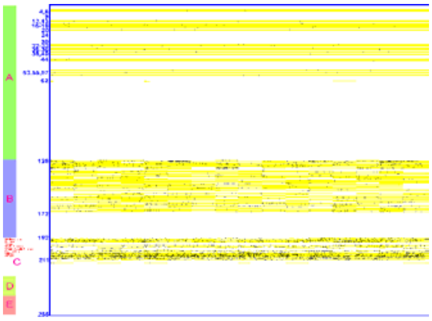
	class		
	A	0network · host	1.0.0.0 to 127.255.255.255
Unicast	B	10 · network · host	128.0.0.0 to 191.255.255.255
	C	110 · network · host	192.0.0.0 to 223.255.255.255
Multicast	D	1110 · multicast address	224.0.0.0 to 239.255.255.255
Reserved	E	1111 · reserved	240.0.0.0 to 255.255.255.255

32 bits

Hosts per Class

- Class A has ~2^{24} hosts (16777216)
- Class B has ~2^{16} hosts (65536)
- Class C has ~2^{8} hosts (256)
- What class do you think everyone wants?
  - Suppose you are a company/university etc. Do you expect to need 16777216 hosts? Do you expect to need more than 256?

IP Address Space Allocation



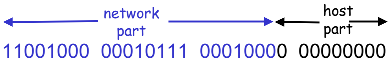
CAIDA 1998

Current Allocation

- Interesting to exam current IP address space allocation (who has class A's ? Etc)
  - Who has A's?
  - Computer companies around during initial allocation (IBM, Apple)
  - Universities (Stanford, MIT)
  - Have A and still use other IP address blocks?
- CAIDA has info on complete allocation

IP addressing: CIDR

- Class-full addressing:
  - inefficient use of address space, address space exhaustion
  - e.g., class B net allocated enough addresses for 65K hosts, even if only 2K hosts in that network
- CIDR: Classless InterDomain Routing
  - network portion of address of arbitrary length
  - address format: a.b.c.d/x, where x is # bits in network portion of address



11001000 00010111 00010000 00000000

200.23.16.0/23

Recall: How to get an IP Address?

- Answer 1: Normally, answer is get an IP address from your upstream provider
  - This is essential to maintain efficient routing!
- Answer 2: If you need lots of IP addresses then you can acquire your own block of them.
  - IP address space is a scarce resource - must prove you have fully utilized a small block before can ask for a larger one and pay $$ (Jan 2002 - $2250/year for /20 and $18000/year for a /14)

How to get lots of IP Addresses? Internet Registries

RIPE NCC (Riseaux IP Europiens Network Coordination Centre) for Europe, Middle-East, Africa

APNIC (Asia Pacific Network Information Centre) for Asia and Pacific

ARIN (American Registry for Internet Numbers) for the Americas, the Caribbean, sub-saharan Africa

Note: Once again regional distribution is important for efficient routing!

Can also get Autonomous System Numbers (ASNs) from these registries

Classful vs Classless

- Class A = /8
- Class B = /16
- Class C = /24

How to get a block of IP addresses?From upstream provider

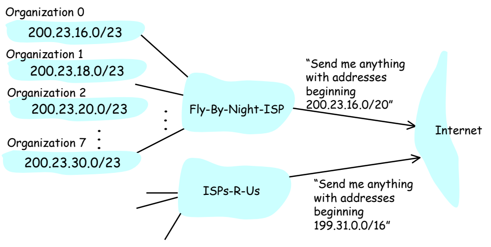
Network (network portion):

- get allocated portion of 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

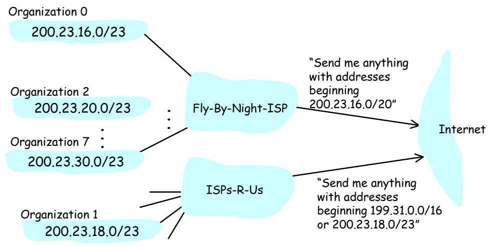
Hierarchical addressing: route aggregation

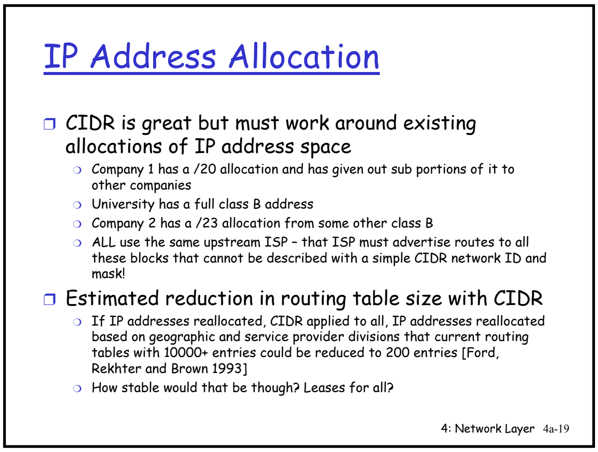
Hierarchical addressing allows efficient advertisement of routing information:



Hierarchical addressing: more specific routes

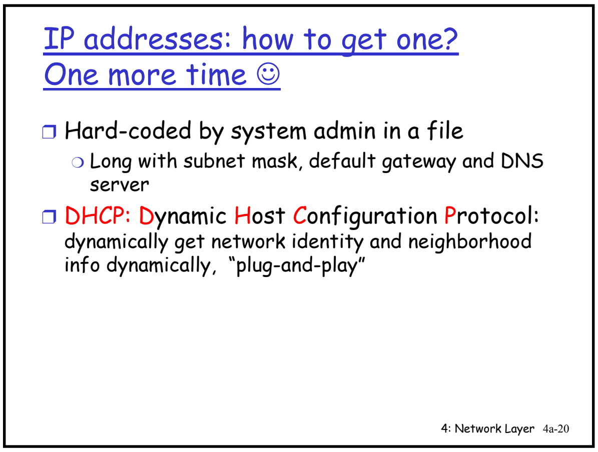
ISPs-R-Us has a more specific route to Organization 1





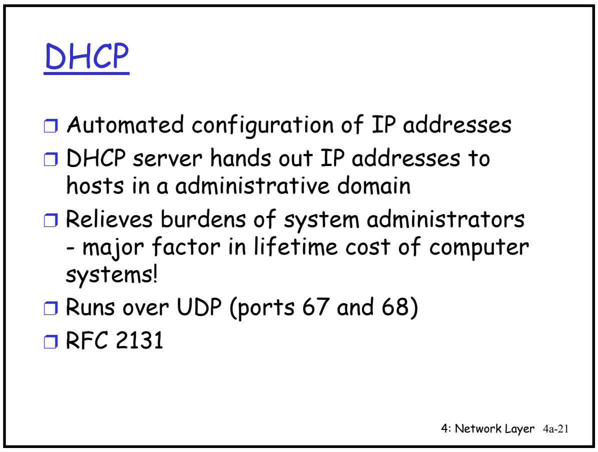
IP Address Allocation

- CIDR is great but must work around existing allocations of IP address space
  - Company 1 has a /20 allocation and has given out sub portions of it to other companies
  - University has a full class B address
  - Company 2 has a /23 allocation from some other class B
  - ALL use the same upstream ISP – that ISP must advertise routes to all these blocks that cannot be described with a simple CIDR network ID and mask!
- Estimated reduction in routing table size with CIDR
  - If IP addresses reallocated, CIDR applied to all, IP addresses reallocated based on geographic and service provider divisions that current routing tables with 10000+ entries could be reduced to 200 entries [Ford, Rekhter and Brown 1993]
  - How stable would that be though? Leases for all?



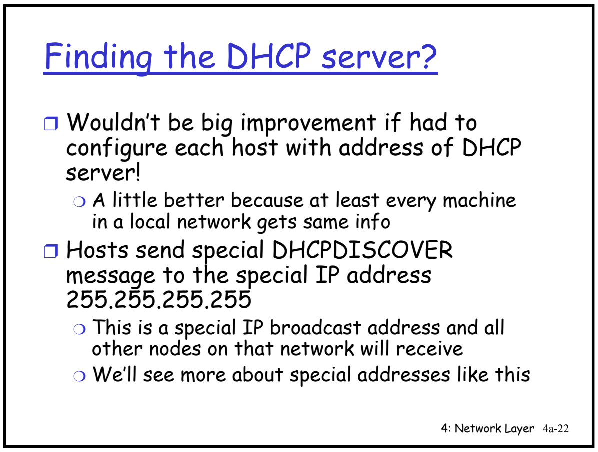
IP addresses: how to get one? One more time ☺

- Hard-coded by system admin in a file
  - Long with subnet mask, default gateway and DNS server
- DHCP: Dynamic Host Configuration Protocol: dynamically get network identity and neighborhood info dynamically, "plug-and-play"



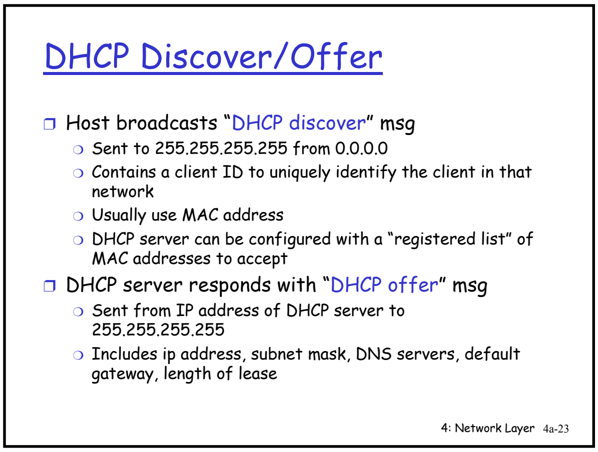
DHCP

- Automated configuration of IP addresses
- DHCP server hands out IP addresses to hosts in a administrative domain
- Relieves burdens of system administrators - major factor in lifetime cost of computer systems!
- Runs over UDP (ports 67 and 68)
- RFC 2131



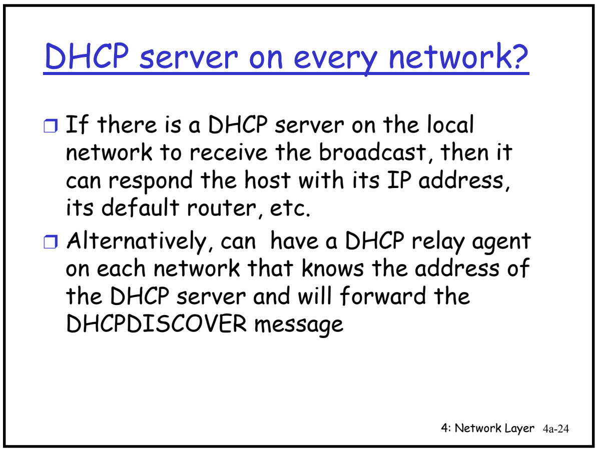
Finding the DHCP server?

- Wouldn't be big improvement if had to configure each host with address of DHCP server!
  - A little better because at least every machine in a local network gets same info
- Hosts send special DHCPDISCOVER message to the special IP address 255.255.255.255
  - This is a special IP broadcast address and all other nodes on that network will receive
  - We'll see more about special addresses like this



DHCP Discover/Offer

- Host broadcasts "DHCP discover" msg
  - Sent to 255.255.255.255 from 0.0.0.0
  - Contains a client ID to uniquely identify the client in that network
  - Usually use MAC address
  - DHCP server can be configured with a "registered list" of MAC addresses to accept
- DHCP server responds with "DHCP offer" msg
  - Sent from IP address of DHCP server to 255.255.255.255
  - Includes ip address, subnet mask, DNS servers, default gateway, length of lease



DHCP server on every network?

- If there is a DHCP server on the local network to receive the broadcast, then it can respond the host with its IP address, its default router, etc.
- Alternatively, can have a DHCP relay agent on each network that knows the address of the DHCP server and will forward the DHCPDISCOVER message

Leases

- DHCP doesn't *give* each client an IP address, just lease them for a while
  - IP addresses aren't reserved by clients not currently connected to the network
- Clients can keep their address by renewing their leases
  - Some DHCP servers will hand out new address each time (especially to prevent customers running servers)
  - Dynamic DNS?
- Clients typically start renewing ½ way through lease period
- Internet wide DHCP to enforce efficient CIDR?

Renew lease

- Host requests IP address: "DHCP request" msg
  - Once know the IP address of the DHCP server then just send a request message directly to them
- DHCP server sends address: "DHCP ack" msg
  - Contains same info as an offer

DHCP other

- DHCP servers and DNS
- DHCP servers and routers
  - What if some machine choose their own IP address in the proper range? Doesn't stop data from flowing to them (see ARP later)
- DHCP and BOOTP
- Security problem – attach to network and act like DHCP server – give out all duplicate IPs

Unicast vs Broadcast vs Multicast

- Unicast Addresses
  - IP Datagram destined for single host
  - Type of IP address you normally think of
  - Class A-C + some special IP addresses
- Broadcast
  - IP Datagram sent to all hosts on a given network
  - Some unicast network id + special host id
  - Some part of reserved E class
- Multicast
  - IP Datagram sent to a set of hosts belonging to a "multicast" group or group of interested receivers
  - Class D
  - We will return to IP multicast later

Broadcast

- All ones like *
- Limited Broadcast
  - 255.255.255.255
  - *.*.*.*
  - Not forwarded!
- Net-directed Broadcast
  - netid.*
  - All bits in host portion 1's
  - 128.1.2.255 is a subnet-directed broadcast with subnet mask 255.255.255.0 but not with 255.255.254.0

Other special addresses

- Loopback
- Specify this host

Note: -1 = all 1's

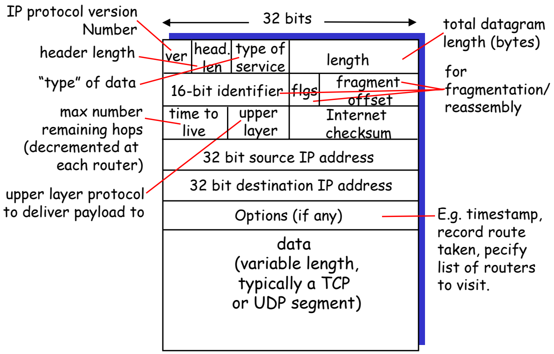
Special Address Summary

net ID	Subnet ID	Host ID	Can be source?	Can be dest?	Description
-1		-1	N	Y	255.255.255.255 Limited broadcast (do not forward!)
Netid		-1	N	Y	netid.255.255.255 Net directed broadcast to netid
Netid	Subnetid	-1	N	Y	Subnet directed broadcast to netid, subnetid
Netid	-1	-1	N	Y	All subnets directed broadcast to netid
0		0	Y	N	This host on this net
0		Hostid	Y	N	Specified host on this net
127		Any	Y	Y	Loopback

Note

- Broadcast and multicast make sense for UDP and not for TCP
  - telnet 255.255.255.255 doesn't make sense
-

IP datagram format



IP protocol version Number

header length

"type" of data

max number remaining hops (decremented at each router)

upper layer protocol to deliver payload to

32 bits

ver	head. len	type of service	length	
16-bit identifier			flgs	fragment offset
time to live		upper layer	Internet checksum	
32 bit source IP address				
32 bit destination IP address				
Options (if any)				
data (variable length, typically a TCP or UDP segment)				

total datagram length (bytes)

for fragmentation/ reassembly

E.g. timestamp, record route taken, pecify list of routers to visit.

IP Header: Version and Header Length

- Version number (4-bit)
  - 4 for IPv4, 6 for IPv6
  - Fields that follow can vary based on this number
- Header length (4-bit)
  - Number of 32 bit words (2^4-1 32 bits = 60 bytes)
  - Includes length of options (40 bytes max)

IP Header: TOS

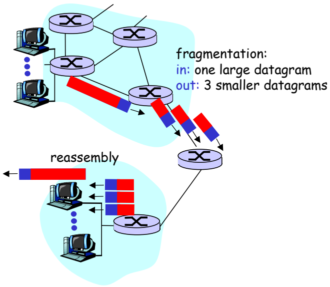
- Type-of-service (TOS) field (8 bits)
  - 3 Bit precedence field
  - 4 TOS bits (only one may be turned on)
    - Minimize delay
    - Maximize throughput
    - Maximize reliability
    - Minimize monetary cost
  - 1 unused bit
- Many implementations ignore; most implementations don't allow application to set this to indicate preference anyway

IP Header

- Total length field (16 bits)
  - Length in bytes
  - Max Total length 2^{16}-1 = 65535 bytes
  - Max Data = 65535 –Header Length
- Can you really send that much?
  - Link layer might not be enough to handle that much; Various link layer technologies have different limits
  - As pass over various link layers, IP datagram will be fragmented if necessary
  - Total length field will change when fragmented

IP Fragmentation & Reassembly

- network links have MTU (max.transfer size) - largest possible link-level frame.
  - different link types, different MTUs
  - Ex. Ethernet maximum is 1500 bytes, FDDI maximum is 4500 bytes
- large IP datagram divided ("fragmented") within net
  - "reassembled" only at final destination (even if pass over other links that could handle larger datagram)
  - May be fragmented multiple times
  - One fragment dropped => entire datagram dropped



IP Header: Fragmentation

- Fields to manage fragmentation
  - Identification (16 bits)
    - "Unique ID" for datagram
    - Original spec said transport layer would set
    - Usually set to value of variable in IP layer that is incremented by one for each datagram sent from that host (regardless of destination)
    - Would wrap every 65535 datagrams
  - Flags (3 bits)
    - 1 bit used to say whether there are more fragments following this one in the original datagram
    - 1 bit used to say "do not fragment" (drop and send error message back to source if need to fragment)
  - Fragment Offset (13 bits)
    - Give offset of data in this fragment into original datagram

IP Fragmentation and Reassembly

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

One large datagram becomes
several smaller datagrams each with their own IP header!!

length	ID	fragflag	offset
=1500	=x	=1	=0

length	ID	fragflag	offset
=1500	=x	=1	=1480

length	ID	fragflag	offset
=1040	=x	=0	=2960

Note: TCP/UDP header with port numbers etc only in 1st fragment

Alternatives to Fragmentation?

- IP wants to be able to run anywhere: Make packet size as small as the minimum packet size anywhere along a route
  - Detection? Min Increases after detection? Least common denominator?
- Look before you leap?
  - Path MTU Discovery = To avoid overhead of fragmentation and reassembly in network, hosts often send a series of probe packets to determine the smallest MTU along a route

Path MTU Discovery in TCP

- If doing Path MTU Discovery, start with minimum of receiver's specified MSS or local MTU and set the Don't Fragment Bit
- If ICMP message received indicating that fragmentation was required, then segment size will be reduced
- Periodically (every 10 min or so), TCP will try a higher segment size up to the receiver's MSS to see if new route is being used that would allow larger segments
- Not all implementations support this

Path MTU Discovery in UDP

- Not like TCP where sender sends stream in chunks as they see fit and receiver reads in chunks as they see fit
- With UDP, the size of the UDP packet is much more visible to the application
- May send with DF bit off
- May send with DF bit on and if get ICMP messages then IP on host may fragment before sent but not exposed to application layer to encourage smaller amounts of data sent
- Again not all implementations support

Exercise

- Trace a TCP connection and look at the IP headers
- Trace a UDP traffic
  - Hint: DNS or DHCP good sources of UDP traffic
- Questions
  - Is the Do Not Fragment Bit on?
  - Look at the Identification flag of subsequent IP datagrams

Path MTU Discovery

- Look in Ethereal at TCP segments, will see Do Not Fragment Bit is set
  - If on Ethernet, don't usually see adjustment
  - Ethernet has one of the smaller MTUs so never get an ICMP error saying needs to be smaller
  - If sent from local network with larger MTU might see this activity

IP Header: TTL

- Time-to-live field / TTL (8 bits)
  - Initialized by sender; decremented at each hop
  - If reaches zero, datagram dropped
  - Limits total number of hops from source to destination (2^8-1 = 255)
  - Prevents things like infinite routing loops
  - Usually set to 32 or 64
    - Look at TTL field in Ethereal for incoming traffic and for outgoing traffic
  - Used by traceroute (more later)

IP Header: Protocol

- Identifies which upper layer protocol to which IP should pass the data
- 8 bits: 2^8-1 = 255 max number protocols
  - 1= ICMP
  - 2= IGMP
  - 6 = TCP
  - 17 = UDP
  - 135-254: Unassigned
- Who do you think assigns these numbers?
  - http://www.iana.org/assignments/protocol-numbers

IP Header

- Header Checksum
  - Calculated over IP header
  - 16-bit one's complement
  - When change TTL, checksum updated
- Source and destination IP addresses
- Options: variable length
  - Security options
  - Record route/timestamp (alternative to traceroute)
  - Loose (strict?)source routing - source can say path it would datagram to take; routers need not support
  - Options must end on 32 bit boundary - pad of 0 if necessary

ICMP: Internet Control Message Protocol

- used by hosts, routers, gateways to communication network-level information like error notification or querying network conditions
- network-layer "above" IP:
  - ICMP msgs carried in IP datagrams
- ICMP message: type, code, checksum plus typically first 8 bytes of IP datagram causing error

ICMP Message Format

8-bit type	8-bit code	16-bit checksum
Contents depends on type		

Error Conditions

- Some error conditions flagged by ICMP
  - unreachable host, network, port, protocol
  - Need to fragment and can't; TTL expired
  - Source Quench
    - Intended for congestion control but not used

ICMP also for Query/Response

- ICMP also used to request IP related info
- Query/Response
  - Address mask request/reply
  - Timestamp request/reply
- Reply echos identifier from query so they can be matched

Ping uses ICMP

- Sends ICMP echo request to a host and looks for ICMP echo reply (like locating ship with sonar)
- Used to measure RTT
- Most implementations support ping directly in the kernel; no "ping server"

Experiment

- Try tracing ping with Ethereal
  - Ping host on same LAN
  - Ping host across Internet (drops, duplication, larger variance in RTT)
  - Ping loopback vs ethernet interface
    - Does this work?
  - Ping of broadcast addresses
    - Does this work?

ICMP Router Discovery

- Instead of configuring machine to know default gateway can use ICMP to find routers
- Broadcast an ICMP router solicitation request
- Routers that hear respond with ICMP router advertisements
  - Advertisements contain the IP address(es) of available routers
- Routers also send periodic router advertisement

ICMP Redirection

- ICMP used to tell source that it sent a datagram through an inefficient path
  - If router sends datagram out the same interface, that it can in on then inefficient routing
  - Send ICMP redirect error
- Simple dynamic routing (more on dynamic routing soon)

ICMP Type/Code List

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

Cascading ICMP messages?

- To avoid “broadcast storms” of ICMP messages
- Do not send an ICMP message in response to:
  - Datagram sent to special IP addresses (broadcast, multicast, loopback,..)
  - Fragment other than the first
  - Other ICMP error messages

traceroute /tracert

- Uses UDP, Relies on TTL + ICMP
- Send series of IP datagrams with increasing TTL fields
  - Source sets TTL =1, will get an ICMP TTL expires message from the first hop
  - Source sets TTL=2, will get an ICMP TTL expires message from second hop
- Essential tool for exploring network core
- Trace route servers;Triangulating the Internet (?)
- Limitations
  - Subsequent datagrams may take different paths
- Ping –R (record route IP option)

Experiment

- Try tracing traceroute with Ethereal
  - Look for the increasing TTL fields
  - Look for the ICMP error messages

Roadmap

- Done with IP basics
  - IP addressing
  - IP datagram format
  - ICMP
- Next major topic: routing
- But first…IP’s datagram model is not the only choice for a network layer model

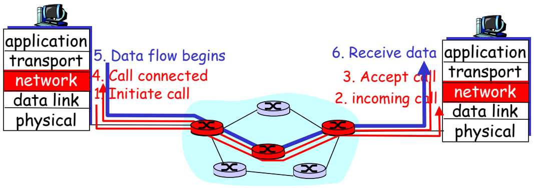
Virtual circuits

“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; associates VC identifier with the path
- each packet carries VC identifier (not destination host OD)
- *every* router on source-dest path s maintain “state” for each passing connection
  - transport-layer connection only involved two end systems
- link, router resources (bandwidth, buffers) may be *allocated* to VC
  - to get circuit-like performance

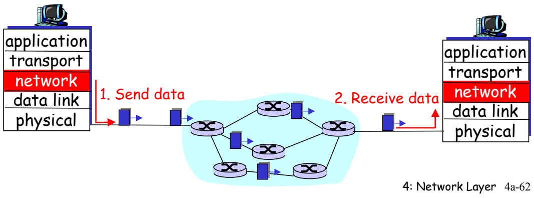
Virtual circuits: signaling protocols

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



Datagram networks: the Internet model

- no call setup at network layer
- routers: no state about end-to-end connections
  - no network-level concept of "connection"
- packets typically routed using destination host ID
  - packets between same source-dest pair may take different paths
- Best effort



Best Effort

What can happen to datagrams?

- Corrupted at the physical level
- Datagrams dropped because of full buffers
- Destination unreachable
- Routing loops

Datagram or VC network: why?

Datagram

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

Virtual Circuit

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