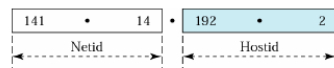
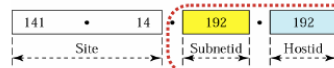


Subnets

- ❖ Network divided into several smaller subnetworks each having its own subnetwork address
 - Internally, each subnetwork is recognized by its subnetwork address; to the rest of the Internet all subnetworks still appear as a single network
- ❖ Organization of address space in a subnetted network
 - A number of HostID bits are borrowed for subnet identification
 - With m borrowed bits, 2^m subnets can be created
 - Number of hosts in each subnet: $2^{\text{Hostid}-m}$



a. Without subnetting

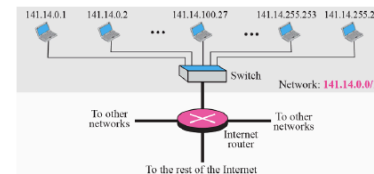


b. With subnetting

hostid is divided into 2 parts:
1) subnet number
2) host number on that subnet

-48

Benefits of subnets

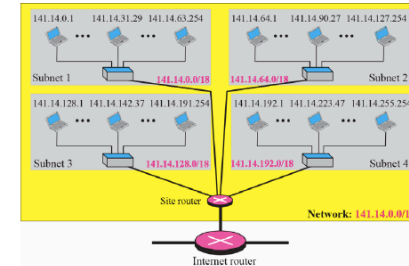


Class B network before subnetting:
each packet flooded through entire network. 2-level routing hierarchy:

- (1) deliver to the network
- (2) deliver to the host

Class B network after subnetting:
packets routed only to appropriate network segment. 3-level routing hierarchy:

- (1) deliver to the network
- (2) deliver to the subnetwork
- (3) deliver to the host

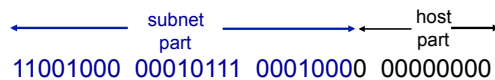


Network Layer 4-49

Classless 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

Network Layer 4-50

IP addresses: how to get one?

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

- ❖ hard-coded by system admin in a file
 - Windows: control-panel->network->configuration->tcp/ip->properties
 - UNIX: /etc/rc.config
- ❖ **DHCP: Dynamic Host Configuration Protocol:** dynamically get address from as server
 - “plug-and-play”

Network Layer 4-51

DHCP: Dynamic Host Configuration Protocol

goal: allow host to *dynamically* obtain its IP address from network server when it joins network

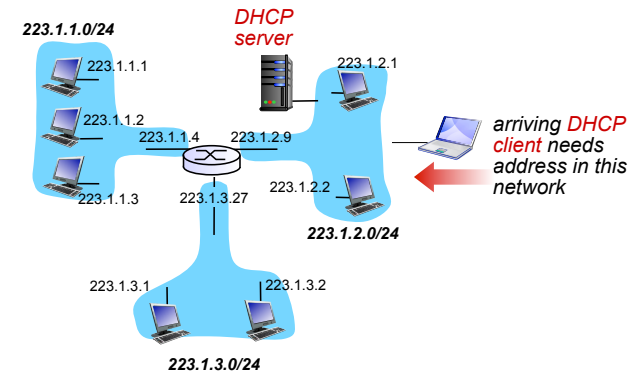
- can renew its lease on address in use
- allows reuse of addresses (only hold address while connected/"on")
- support for mobile users who want to join network (more shortly)

DHCP overview:

- host broadcasts "DHCP discover" msg [optional]
- DHCP server responds with "DHCP offer" msg [optional]
- host requests IP address: "DHCP request" msg
- DHCP server sends address: "DHCP ack" msg

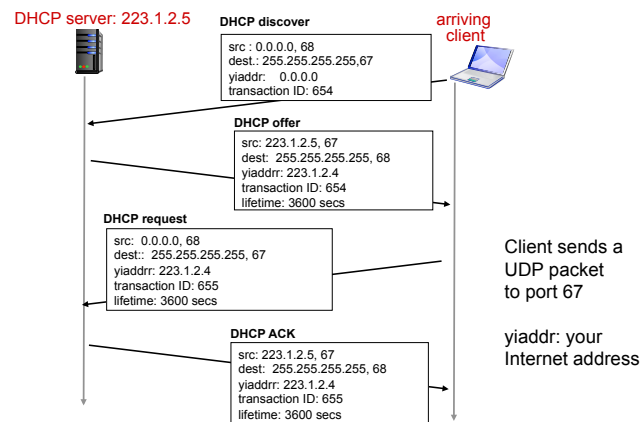
Network Layer 4-52

DHCP client-server scenario



Network Layer 4-53

DHCP client-server scenario



Network Layer 4-54

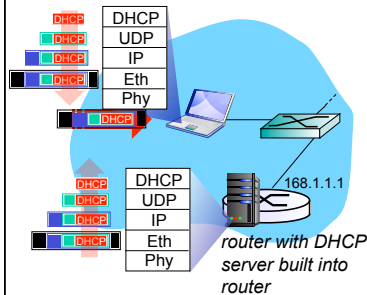
DHCP: more than IP addresses

DHCP can return more than just allocated IP address on subnet:

- address of first-hop router for client
- name and IP address of DNS server
- network mask (indicating network versus host portion of address)

Network Layer 4-55

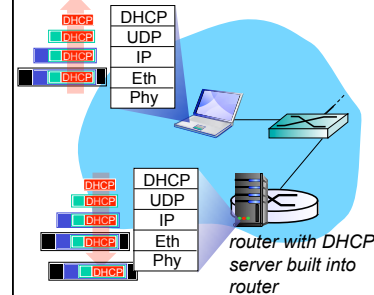
DHCP: example



- ❖ connecting laptop needs its IP address, addr of first-hop router, addr of DNS server: use DHCP
- ❖ DHCP request encapsulated in UDP, encapsulated in IP, encapsulated in 802.1 Ethernet
- ❖ Ethernet frame broadcast (dest: FFFFFFFF) on LAN, received at router running DHCP server
- ❖ Ethernet demuxed to IP demuxed, UDP demuxed to DHCP

Network Layer 4-56

DHCP: example



- ❖ DHCP server formulates DHCP ACK containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server
- ❖ encapsulation of DHCP server, frame forwarded to client, demuxing up to DHCP at client
- ❖ client now knows its IP address, name and IP address of DNS server, IP address of its first-hop router

Network Layer 4-57

IP addresses: how to get one?

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

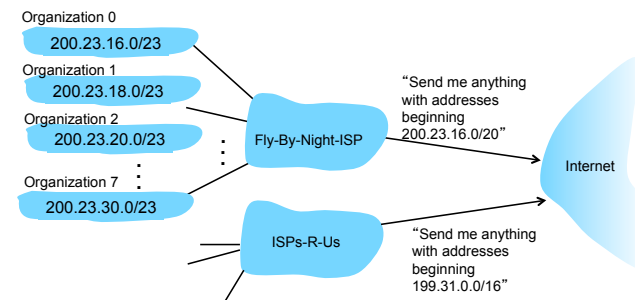
A: gets allocated portion of its provider ISP's address space

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 4-58

Hierarchical addressing: route aggregation

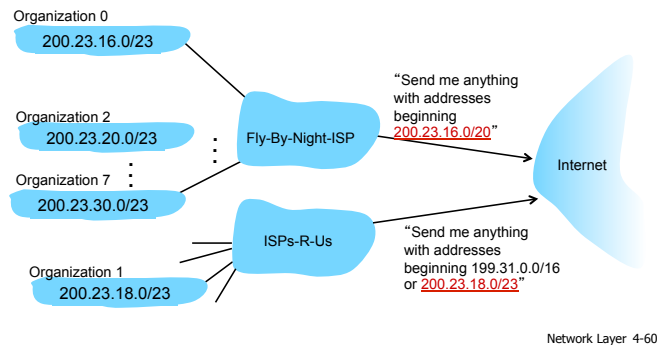
hierarchical addressing allows efficient advertisement of routing information:



Network Layer 4-59

Hierarchical addressing: more specific routes

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



IP addressing: the last word...

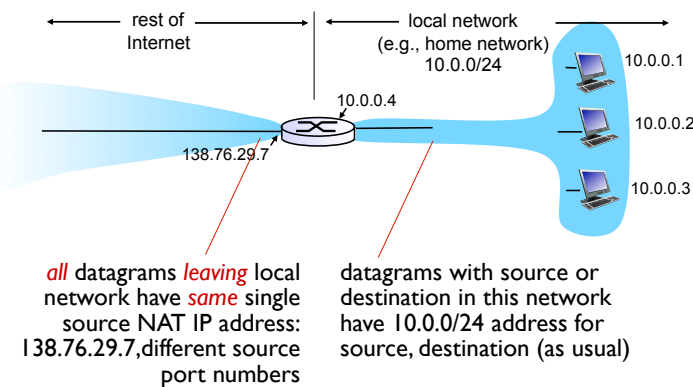
Q: how does an ISP get block of addresses?

A: ICANN: Internet Corporation for Assigned Names and Numbers <http://www.icann.org/>

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

Network Layer 4-61

NAT: network address translation



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 4-63

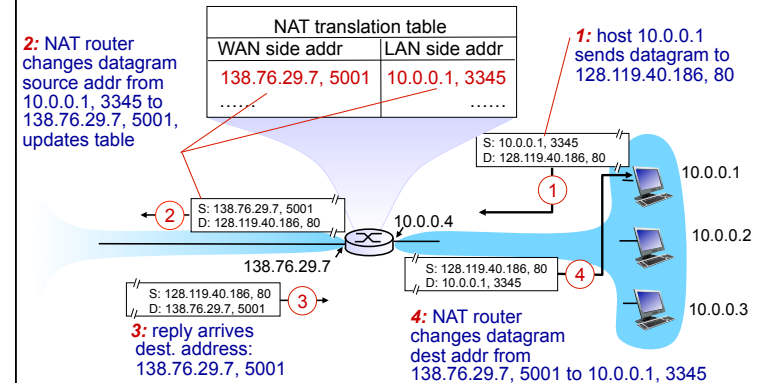
NAT: network address translation

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 4-64

NAT: network address translation



Network Layer 4-65

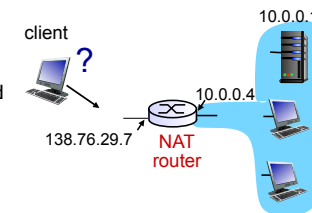
NAT: network address translation

- ❖ 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, e.g., P2P applications
 - address shortage should instead be solved by IPv6

Network Layer 4-66

NAT traversal problem

- ❖ client wants to connect to server with address 10.0.0.1
 - server address 10.0.0.1 local to LAN (client can't use it as destination addr)
 - only one externally visible NATed address: 138.76.29.7
- ❖ **solution 1:** statically configure NAT to forward incoming connection requests at given port to server
 - e.g., (138.76.29.7, port 2500) always forwarded to 10.0.0.1 port 25000



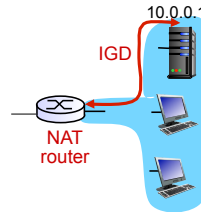
Network Layer 4-67

NAT traversal problem

❖ **solution 2:** Universal Plug and Play (UPnP) Internet Gateway Device (IGD) Protocol. Allows NATed host to:

- ❖ learn public IP address (138.76.29.7)
- ❖ add/remove port mappings (with lease times)

i.e., automate static NAT port map configuration

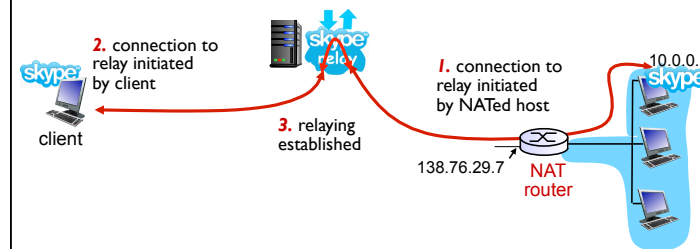


Network Layer 4-68

NAT traversal problem

❖ **solution 3:** relaying (used in Skype)

- NATed client establishes connection to relay
- external client connects to relay
- relay bridges packets between to connections



Network Layer 4-69

Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- datagram format
- IPv4 addressing
- ICMP
- IPv6

4.5 routing algorithms

- link state
- distance vector
- hierarchical routing

4.6 routing in the Internet

- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

Network Layer 4-70

ICMP: internet control message protocol

❖ Why ICMP?

- Lack of error control, e.g. error-reporting and error correcting, and network assistance mechanisms
- What if a router must discard a datagram because the datagram's TTL=0?
- What if the final host must discard a number of fragments because it has not received all fragments by a certain time?
- What is a host needs to determine if another host/router is alive?

❖ ICMP – “companion” to IP, intend to compensate for IP deficiencies

- IP header's “protocol field” set to 1 if the packet carries an ICMP message

Network Layer 4-71

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

Network Layer 4-72

Traceroute and ICMP

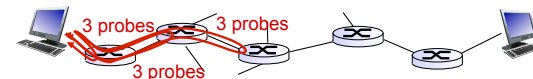
- ❖ source sends series of UDP segments to dest
 - first set has TTL = 1
 - second set has TTL=2, etc.
 - unlikely port number

- ❖ when *n*th set of datagrams arrives to *n*th router:
 - router discards datagrams
 - and sends source ICMP messages (type 11, code 0)
 - ICMP messages includes name of router & IP address

- ❖ when ICMP messages arrives, source records RTTs

stopping criteria:

- ❖ UDP segment eventually arrives at destination host
- ❖ destination returns ICMP “port unreachable” message (type 3, code 3)
- ❖ source stops



Network Layer 4-73

IPv6: motivation

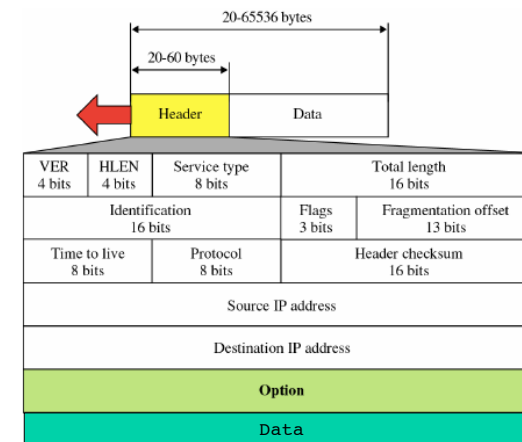
- ❖ *initial motivation:* 32-bit address space soon to be completely allocated.
- ❖ additional motivation:
 - header format helps speed processing/forwarding
 - header changes to facilitate QoS

IPv6 datagram format:

- expanded addressing capability: 128-bit address
- fixed-length 40 byte header
- no fragmentation allowed

Network Layer 4-74

IPv4 datagram format

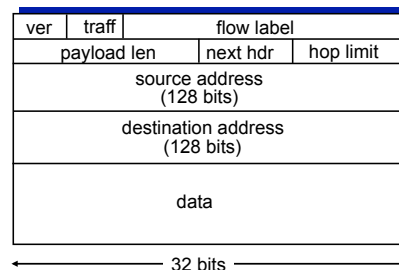


Network Layer 4-75

IPv6 datagram format

A streamlined 40-byte header:

- **traffic class:** identify priority among datagrams in flow
- **flow Label:** identify datagrams in same “flow.”
- **next header:** identify upper layer protocol for data



Network Layer 4-76

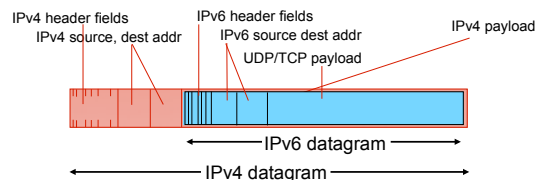
Other changes from IPv4

- ❖ **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 4-77

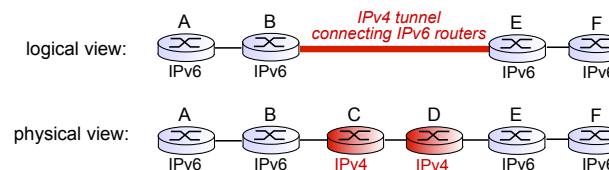
Transition from IPv4 to IPv6

- ❖ not all routers can be upgraded simultaneously
 - no “flag days”
 - how will network operate with mixed IPv4 and IPv6 routers?
- ❖ **tunneling:** IPv6 datagram carried as *payload* in IPv4 datagram among IPv4 routers



Network Layer 4-78

Tunneling



Network Layer 4-79

Tunneling

