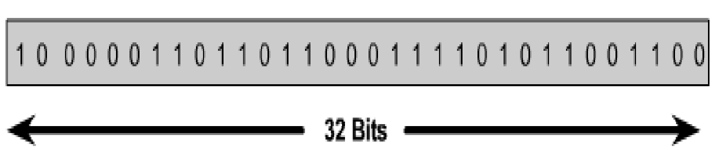
IP Addresses

IP Address

- To make the IP address easier to use, the address is usually written as four decimal numbers separated by periods.
- This way of writing the address is called the dotted decimal format.



Dotted-decimal notation

192.207.56.22

11000000 11001111 00111000 00010110

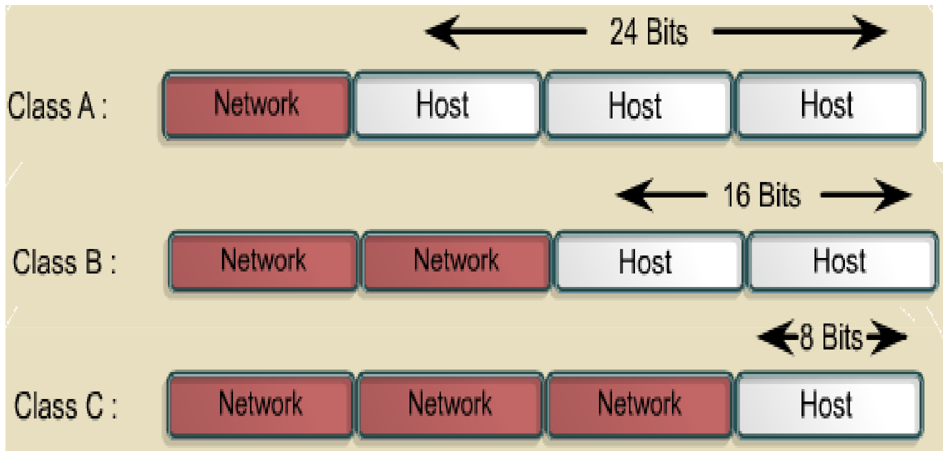
Network prefix and Host number

- The network prefix identifies a network and the host number identifies a specific host (actually, interface on the network).

network prefix	host number

CLASSFUL ADDRESSING

In classful addressing, the address space is divided into five classes: A, B, C, D, and E.



The Class D address class was created to enable multicasting.

IETF reserves Class E addresses for its own research.

Finding the class in binary notation

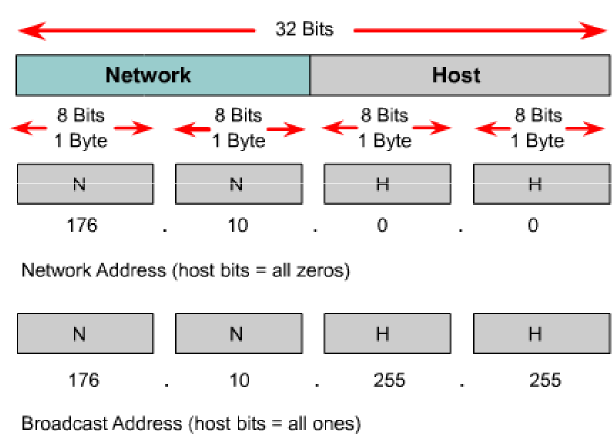
	First byte	Second byte	Third byte	Fourth byte
Class A	**0**			
Class B	**10**			
Class C	**110**			
Class D	**1110**			
Class E	**1111**			

Finding the class in decimal notation

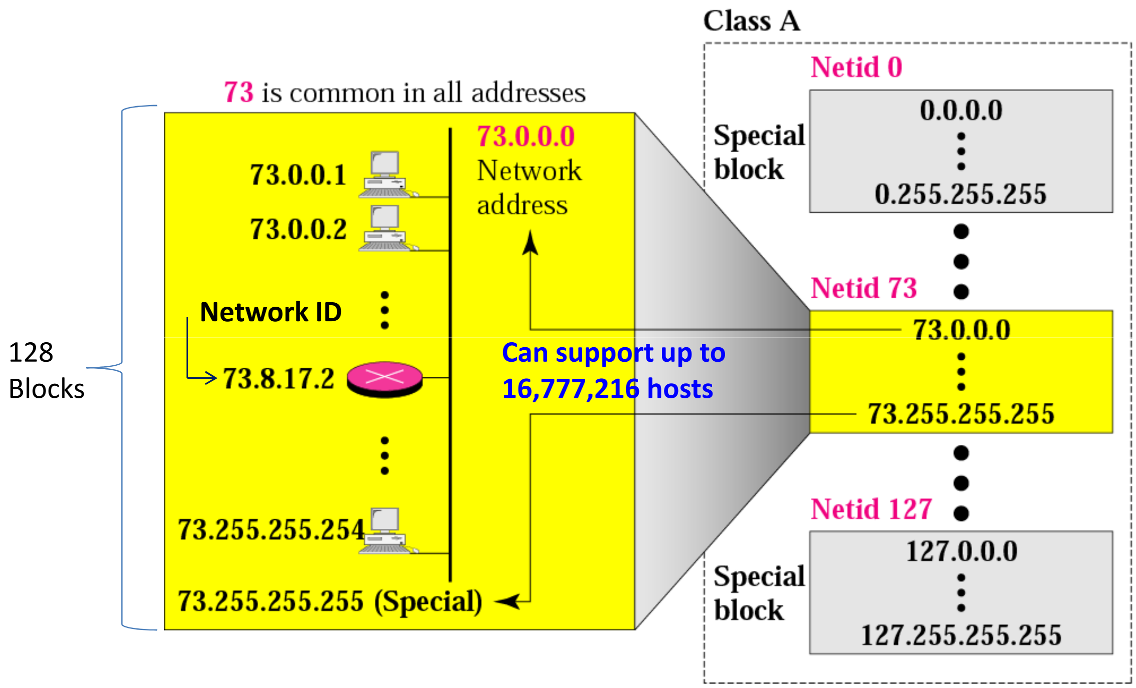
	First byte	Second byte	Third byte	Fourth byte
Class A	**0 to 127**			
Class B	**128 to 191**			
Class C	**192 to 223**			
Class D	**224 to 239**			
Class E	**240 to 255**			

Reserved IP Addresses

- Certain host addresses are reserved and cannot be assigned to devices on a network.
- An IP address that has binary 0s in all host bit positions is reserved for the **network address**.
- An IP address that has binary 1s in all host bit positions is reserved for the **broadcast address**.



Example: Blocks in class A

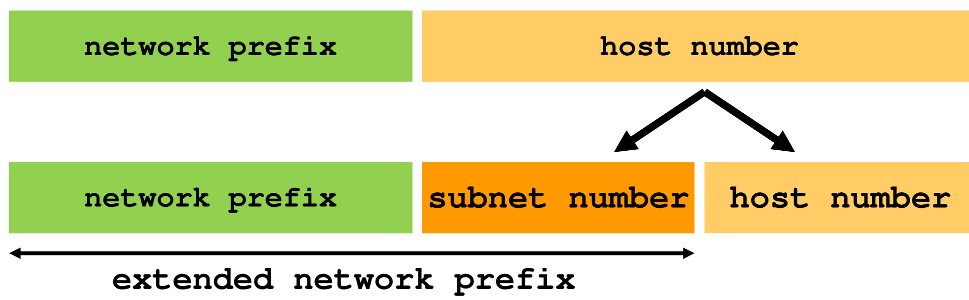


Problems with Classfull IP Addresses

- The classful address scheme had a number of problems
  - **Problem 1.** Too few network addresses for large networks
  - **Problem 2.** Two-layer hierarchy is not appropriate for large networks with Class A and Class B addresses.
  - **Problem 3.** Address Depletion
    - » Class A and Class B addresses are gone

Subnetting(3 level hierarchy)

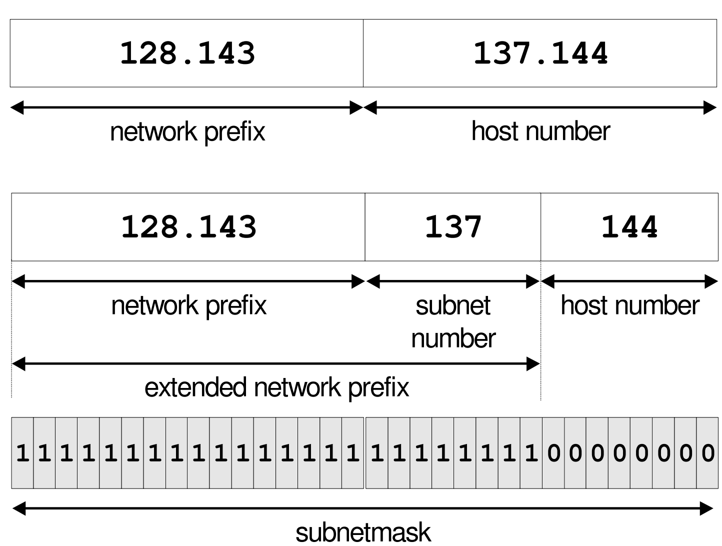
- Split the host number portion of an IP address into a `subnet number` and a (smaller) `host number`.
- Result is a 3-layer hierarchy



- The extended network prefix is also called subnetmask
- Then:
    - Subnets can be freely assigned within the organization
    - Internally, subnets are treated as separate networks
    - Subnet structure is not visible outside the organization

Subnetmask

- Routers and hosts use an **extended network prefix (subnetmask)** to identify the start of the host numbers



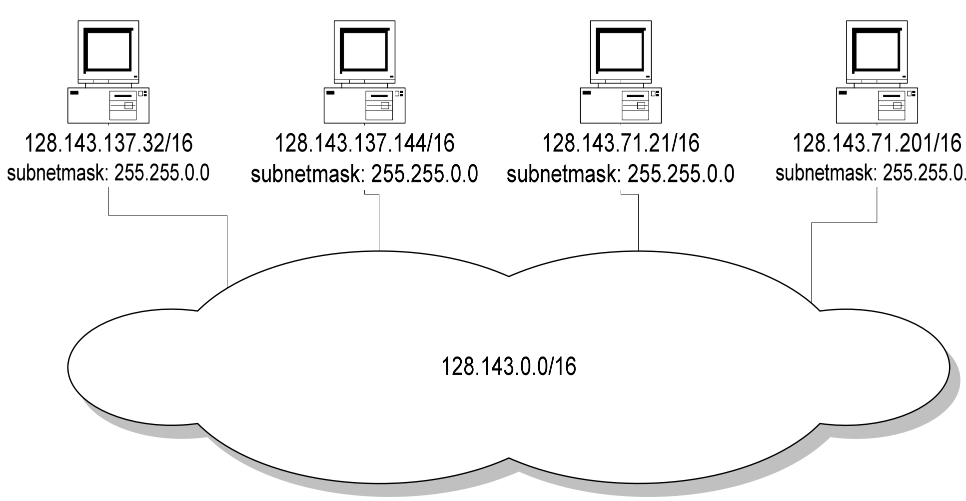
Example: Subnetmask

- 128.143.0.0/16 is the IP address of the network
- 128.143.137.0/24 is the IP address of the subnet

- When subnetting is used, one generally speaks of a “subnetmask” (instead of a netmask) and a “subnet” (instead of a network)
- Use of subnetting or length of the subnetmask is decided by the network administrator
- Consistency of subnetmasks is responsibility of administrator

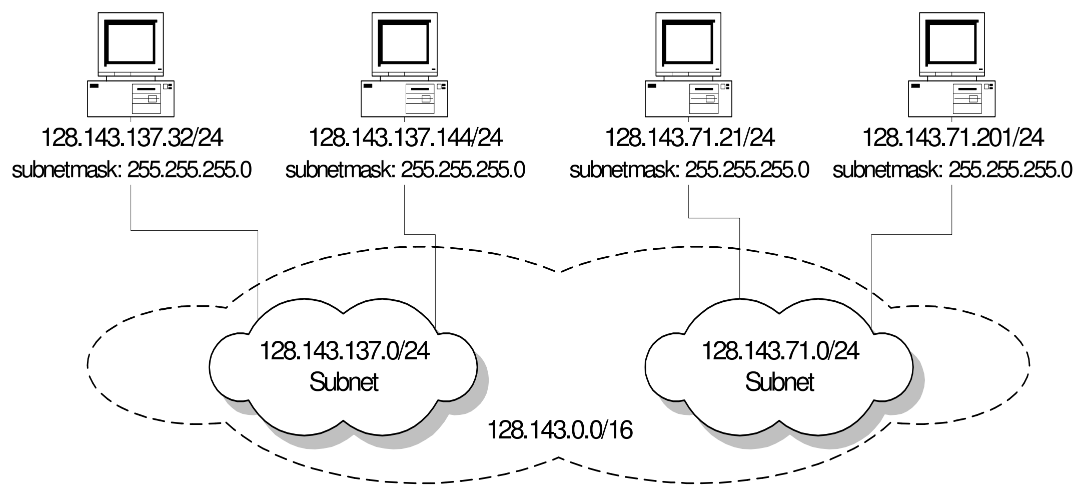
No Subnetting

- All hosts think that the other hosts are on the same network



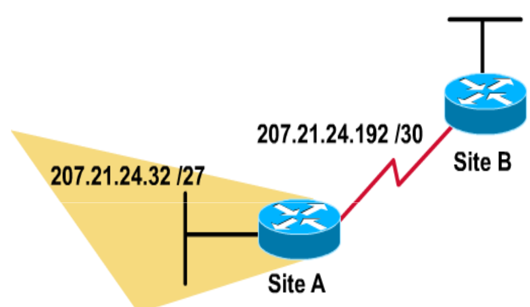
With Subnetting

- Hosts with same extended network prefix belong to the same network



Variable-Length Subnet Mask - VLSM

- VLSM allows you to use more than one subnet mask within the same network address space - subnetting a subnet



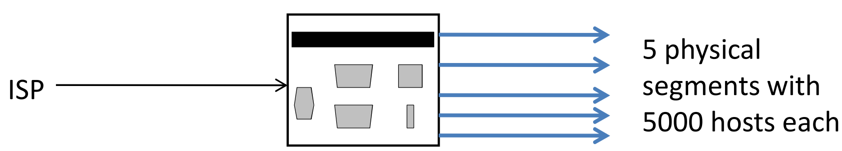
S	Subnet Add
0	207.21.24.0/27
1	207.21.24.32/27
2	207.21.24.64/27
3	207.21.24.96/27
4	207.21.24.128/27
5	207.21.24.160/27
6	207.21.24.192/27
7	207.21.24.224/27

Sub-sub	Sub-Subnet Add
Sub 0	207.21.24.192/30
Sub 1	207.21.24.196/30
……..	
Sub 5	207.21.24.212/30
Sub 6	207.21.24.216/30
Sub 7	207.21.24.220/30

An Example Problem in Subnetting

- Problem:

  Create an IP Addressing Plan for a Company that:
  - Has 5 Physical segments that each have a maximum of 5000 host on each segment
  - And is assigned a class B Address 152.77.0.0



Subnet IDs

- Portions of the Assigned Network ID are Defined by Subnet IDs
  - **152.77.0.0 (Network IP Address)**
  - **255.255.0.0 (Default Subnet Mask)**
- Network . Network . Host . Host (Default SNM)
- Network . Network . SN-ID . Host (Custom SNM)
  - All Device/Hosts Share the Assigned Network ID (All Physical Segments)
  - Each Physical Segment of the Network has a Unique Subnet-ID and the Subnet ID is Common to All Hosts on a Physical Segment
  - Each Host on the Network has a Host ID Unique to its Subnet ID

Subnet ID/Host Chart for Class B Networks

- 152.77.0.0 Network Address
- 255.255.0.0 Default SNM
- 11111111.11111111.0000 0000.0000 0000 SNM (Binary)

SNM (Last Two Octets)	SNM	#of SN-IDs*	#of Hosts Per SN-ID
1000 0000 0000 0000	128	2-2=0	32768-2=32766
1100 0000 0000 0000	192	4-2=2	16384-2=16382
1110 0000 0000 0000	**224**	**8-2=6**	**8192-2=8190**
1111 0000 0000 0000	240	16-2=14	4096-2=4094
1111 1000 0000 0000	248	32-2=30	2048-2=2046
1111 1100 0000 0000	252	64-2=62	1024-2=1022
1111 1110 0000 0000	254	128-2=126	512-2=510
1111 1111 0000 0000	255	256-2=254	256-2=254

CIDR - Classless Interdomain Routing

- IP backbone routers have one routing table entry for each network address:
  - With subnetting, a backbone router only needs to know one entry for each Class A, B, or C networks
  - This is acceptable for Class A and Class B networks
    - 2^7 = 128 Class A networks
    - 2^{14} = 16,384 Class B networks
  - But this is not acceptable for Class C networks
    - 2^{21} = 2,097,152 Class C networks
- In 1993, the size of the routing tables started to outgrow the capacity of routers
- Consequence: The Class-based assignment of IP addresses had to be abandoned

CIDR - Classless Interdomain Routing

- **Goals:**
  - New interpretation of the IP address space
  - Restructure IP address assignments to increase efficiency
  - Hierarchical routing aggregation to minimize route table entries

- CIDR (Classless Interdomain routing)
  - abandons the notion of classes
  - **Key Concept:** The length of the network prefix in the IP addresses is kept arbitrary (VLSM)
  - **Consequence:** Size of the network prefix must be provided with an IP address

CIDR Notation

- CIDR notation of an IP address:

  192.0.2.0/18

  - "18" is the prefix length. It states that the first 18 bits are the network prefix of the address (and 14 bits are available for specific host addresses)

- CIDR notation allows to drop trailing zeros of network addresses:

  192.0.2.0/18 can be written as **192.0.2/18**

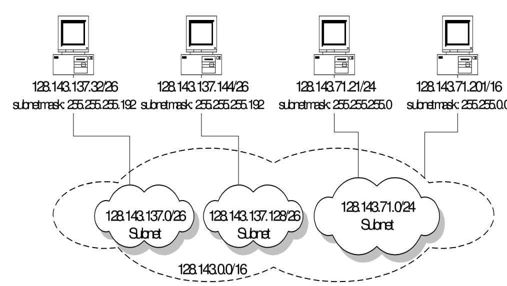
CIDR address blocks

- CIDR notation can nicely express blocks of addresses
- Blocks are used when allocating IP addresses for a company and for routing tables (route aggregation)

CIDR Block Prefix	**# of Host Addresses**
/27	**32**
/26	**64**
/25	**128**
/24	**256**
/23	**512**
/22	**1,024**
/21	**2,048**
/20	**4,096**
/19	**8,192**
/18	**16,384**
/17	**32,768**
/16	**65,536**
/15	**131,072**
/14	**262,144**
/13	**524,288**

Subnetting and Supernetting

- CIDR is compatible with **subnetting**:
  - Subnets are created by extending the prefix
- CIDR can do more:
  - CIDR can refer to multiple networks with a single prefix:
    - 128.143.0.0/16 and 128.173.0.0/16 can be summarized as 128.128.0.0/10
  - This is called **supernetting** (In fact, CIDR and supernetting are often used as the same thing)
  - If neighboring networks have similar address blocks, supernetting reduces the size of routing tables

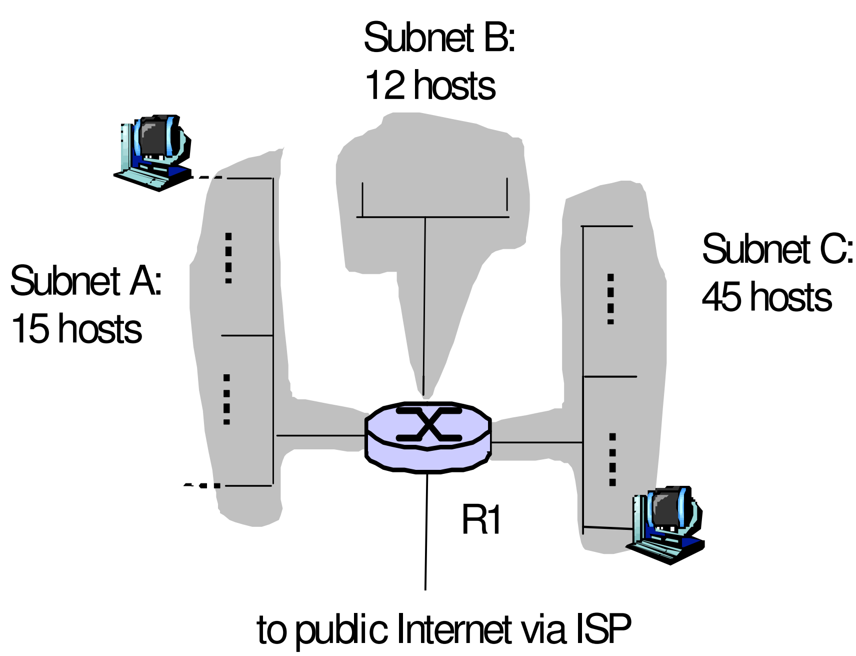


CIDR and Address assignments

- Exploiting supernetting to reduce size of routing tables:
  - Backbone ISPs obtain blocks of IP addresses and allocate portions of their address blocks to their customers.
  - Customers can allocate a portion of their address block to their customers.

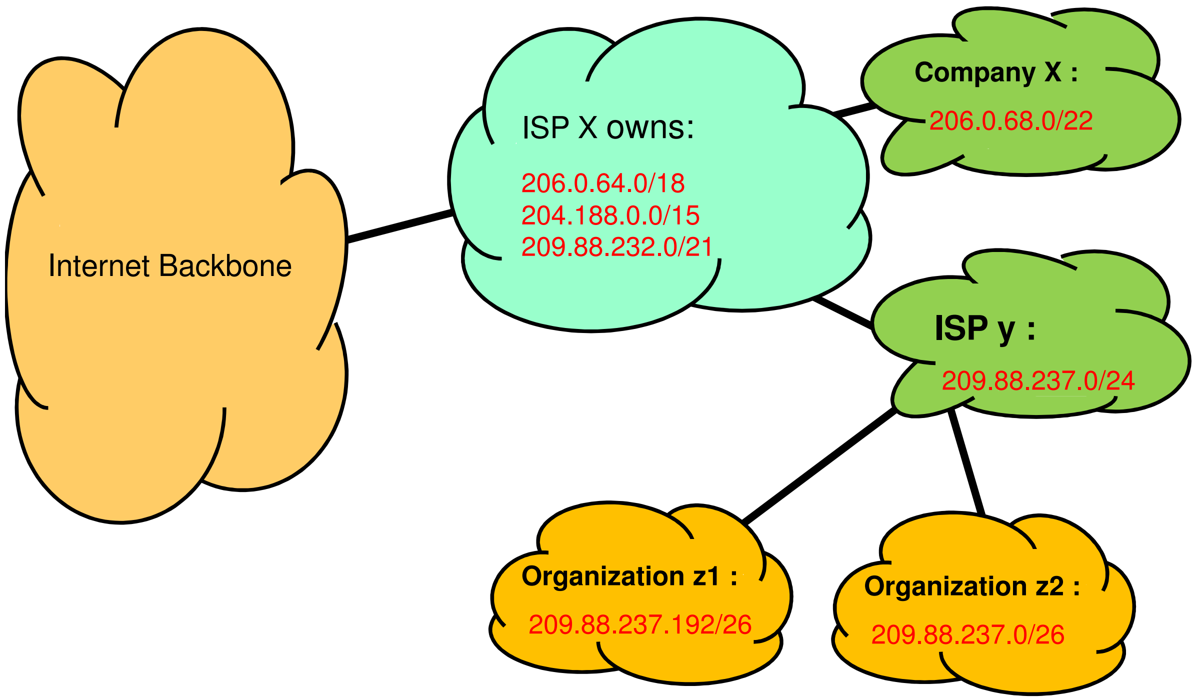
Example:

- Assume that an ISP owns the address block 206.0.64.0/18, which represents 16,384 (2^{14}) IP addresses
- Suppose a client requires 800 host addresses
- With classful addresses: need to assign a class B address (and waste ~64,700 addresses) or four individual Class Cs (and introducing 4 new routes into the global Internet routing tables)
- With CIDR: Assign a /22 block, e.g., 206.0.68.0/22, and allocated a block of 1,024 (2^{10}) IP addresses.



Using the public address of router R1 **128.119.40.0/23**

CIDR and Routing Information



CIDR and Routing

- **Aggregation** of routing table entries:
  - 128.143.0.0/16 and 128.144.0.0/16 are represented as 128.142.0.0/15
- **Longest prefix match**: Routing table lookup finds the routing entry that matches the longest prefix

What is the outgoing interface for 128.143.137.0/24 ?

Prefix	Interface
128.0.0.0/4	interface #5
128.128.0.0/9	interface #2
128.143.128.0/17	interface #1

Routing table

Route aggregation can be exploited when IP address blocks are assigned in an hierarchical fashion

CIDR and Routing

Longest prefix match: Routing table lookup finds the routing entry that matches the longest prefix

What is the outgoing interface for
128.143.137.0/24 ?

Apply /17 on 128.143.137.0 to get network Address:
10000000. 10001111. 10001001.00000000
Now take 17 bits from above and set the rest To 0.
10000000. 10001111.10000000.00000000 =
128.143.128.0
Now look for this in the routing table, we find the first entry as exact match. So
Forward this packet through interface #1

Prefix	Interface
128.143.128.0/17	interface #1
128.128.0.0/9	interface #2
128.0.0.0/4	interface #5

Routing table

Solve it:

206.0.68.5/22

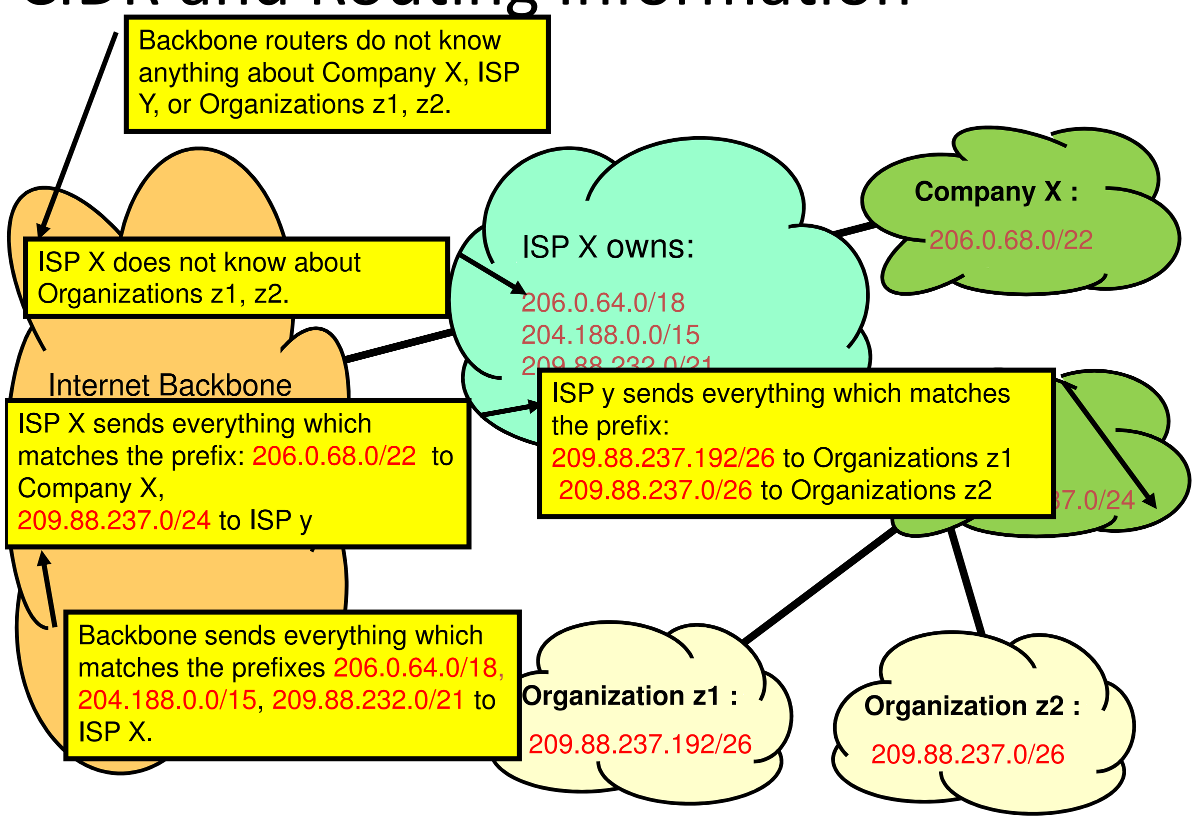
Destination Address	Next Hop
206.0.64.0/18	R1
204.188.0.0/15	R2
209.88.232.0/21	R3
Default	R4

Solve it:

206.0.68.5/22

Destination Address	Next Hop
209.88.232.0/21	R3
206.0.64.0/18	R1
204.188.0.0/15	R2
Default	R4

CIDR and Routing Information



Routing table lookup: Longest Prefix Match

With CIDR, there can be multiple matches for a destination address in the routing table

Longest Prefix Match: Search for the routing table entry that has the longest match with the prefix of the destination IP address (=Most Specific Router):

1. *Search for a match on all 32 bits*
2. *Search for a match for 31 bits*

   .....

32. *Search for a mach on 0 bits*

Needed: Data structures that support a fast longest prefix match lookup!

Problems with IPv4 IP Addresses

Problem 5. In CIDR, the IP addresses in a corporate network are obtained from the service provider. Changing the service provider requires changing all IP addresses in the network

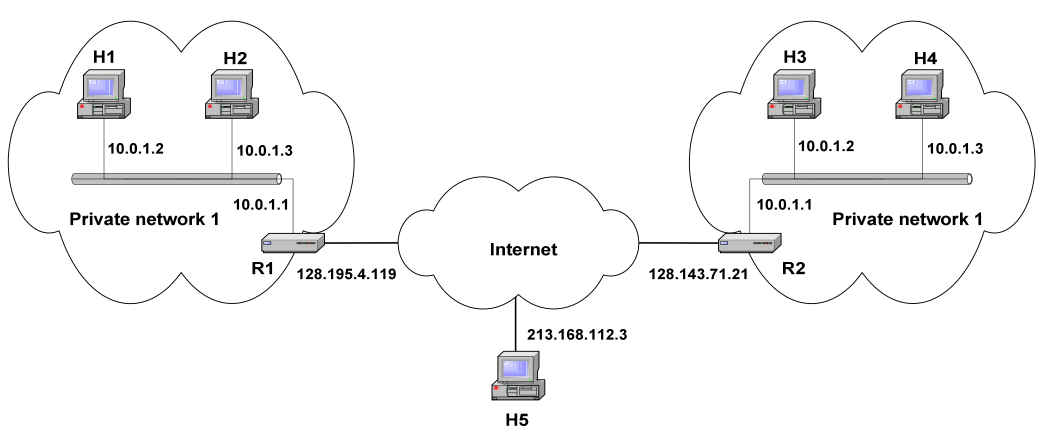
- **Sol :** **private addresses:**
  - Assign private addresses to the hosts of the corporate network
  - NAT device has static address translation entries which bind the private address of a host to the public address.
  - Migration to a new network service provider merely requires an update of the NAT device. The migration is not noticeable to the hosts on the network.

Private Network

- A *Private IP* network is an IP network that is not directly connected to the Internet
- IP addresses in a private network can be assigned arbitrarily.
  - Not registered and not guaranteed to be globally unique
- Generally, private networks use addresses from the following experimental address ranges (*non-routable addresses*):

Range	Total
10.0.0.0 to 10.255.255.255	2^{24}
172.16.0.0 to 172.31.255.255	2^{20}
192.168.0.0 to 192.168.255.255	2^{16}

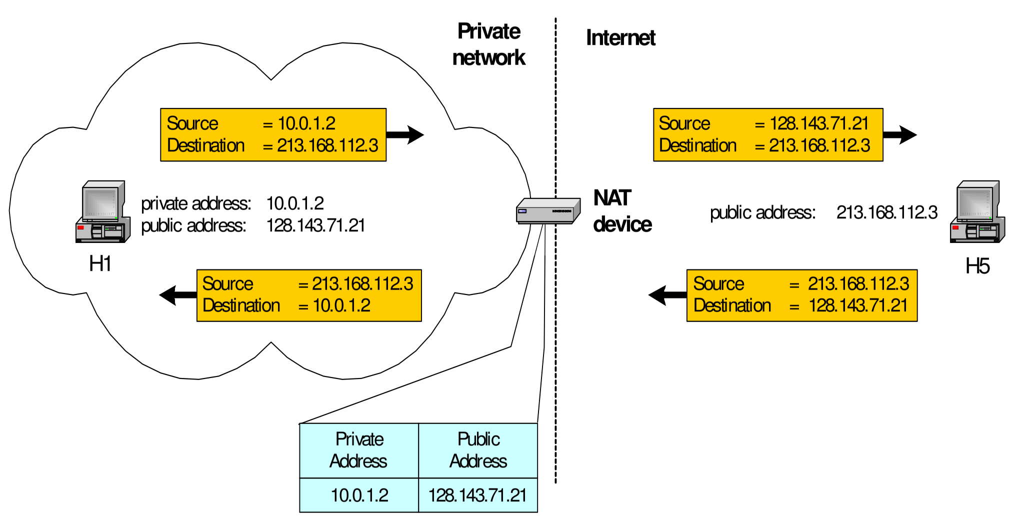
Private Addresses



Network Address Translation (NAT)

- NAT is a router function where IP addresses (and possibly port numbers) of IP datagrams are replaced at the boundary of a private network
- NAT is a method that enables hosts on private networks to communicate with hosts on the Internet
- NAT is run on routers that connect private networks to the public Internet, to replace the IP address-port pair of an IP packet with another IP address-port pair.

Basic operation of NAT

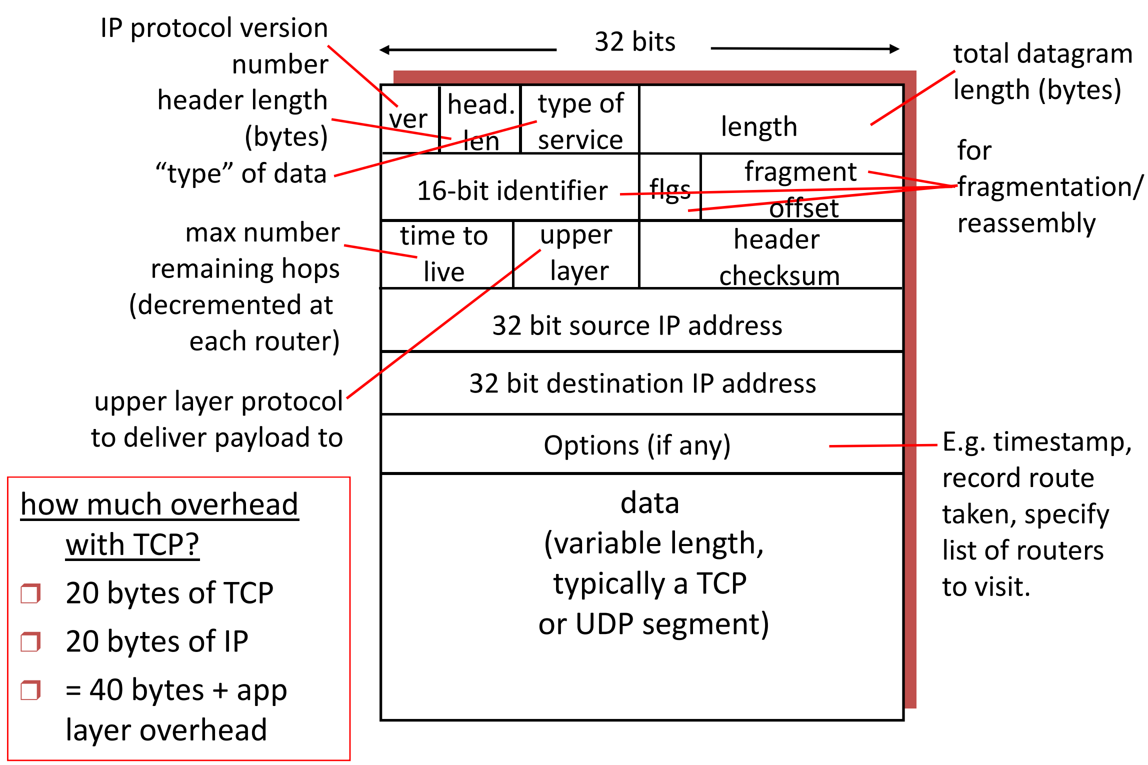


- NAT device has address translation table

Concerns about NAT

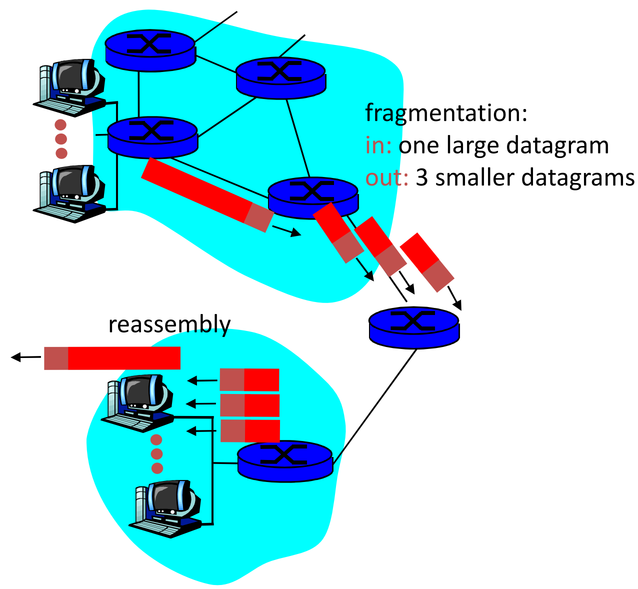
- **IP address in application data:**
  - Applications that carry IP addresses in the payload of the application data generally do not work across a private-public network boundary.
  - Some NAT devices inspect the payload of widely used application layer protocols and, if an IP address is detected in the application-layer header or the application payload, translate the address according to the address translation table.

IP datagram format



IP Fragmentation & Reassembly

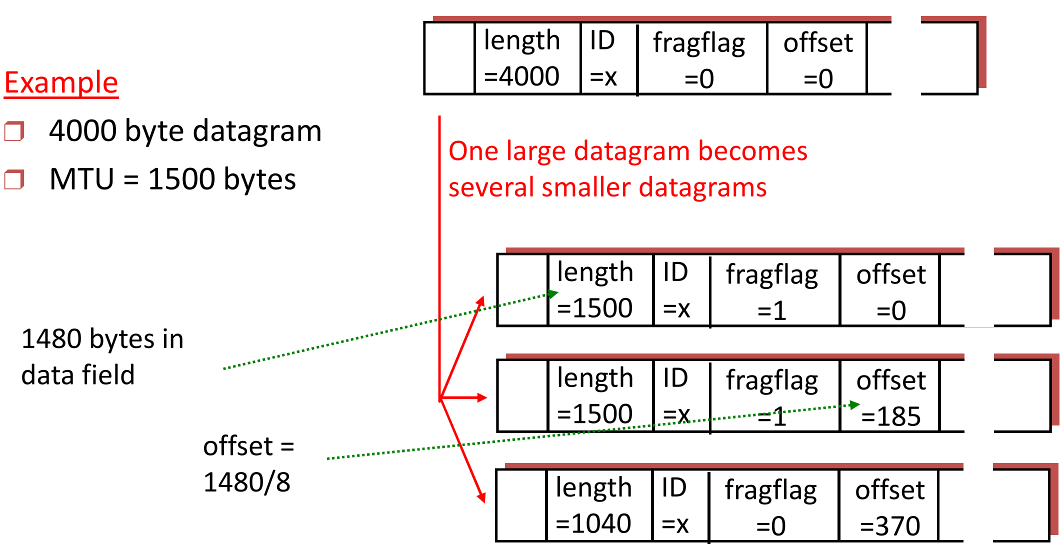
- network links have MTU (max.transfer size) - largest possible link-level frame.
  - different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify,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



IP Fragmentation and Reassembly

Example

- 4000 byte datagram
- MTU = 1500 bytes



Problems with IPv4 IP Addresses

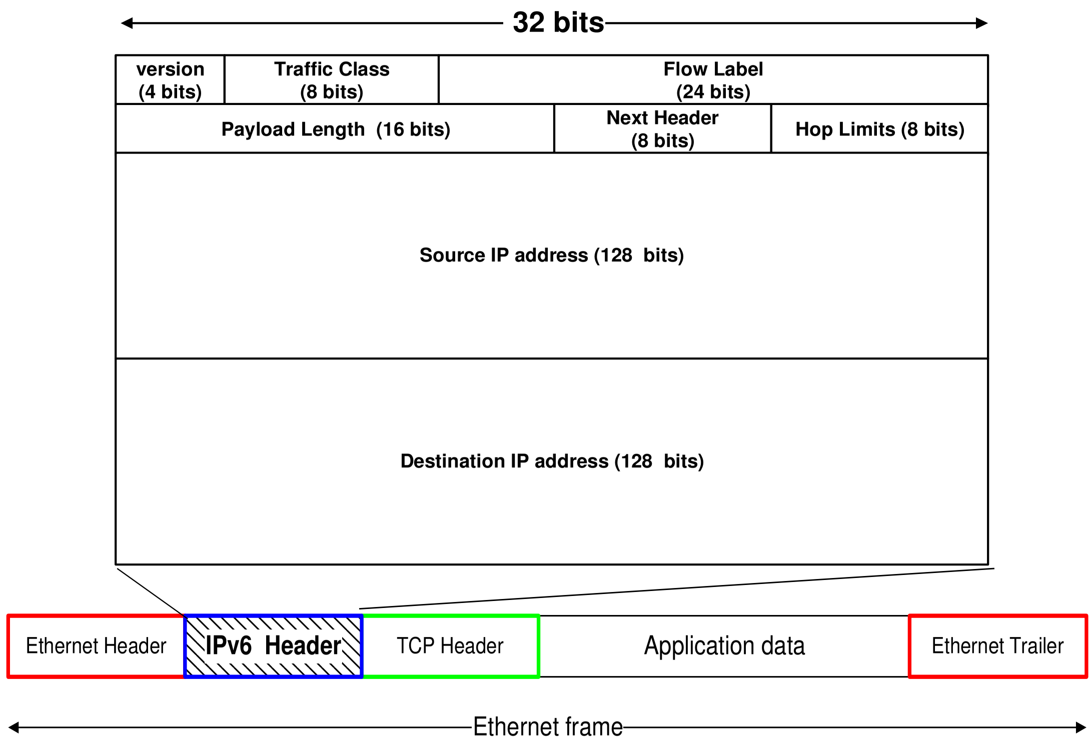
Problem 6. The Internet is going to outgrow the 32-bit addresses ($2^{32} \approx$ 4G)

- **Sol : IP Version 6**

IPv6 - IP Version 6

- **IP Version 6**
  - Is the successor to the currently used IPv4
  - Specification completed in 1994
  - Makes improvements to IPv4 (no revolutionary changes)
- One (not the only !) feature of IPv6 is a significant increase in of the IP address to **128 bits (16 bytes)**
  - IPv6 will solve – for the foreseeable future – the problems with IP addressing

IPv6 Header



IPv6 vs. IPv4: Address Comparison

- **IPv4** has a maximum of

  $2^{32} \approx 4$ billion addresses

- **IPv6** has a maximum of

  $\mathbf{2^{128} = (2^{32})^4 \approx 4}$ **billion x 4 billion x 4 billion x 4 billion addresses**

Notation of IPv6 addresses

- **Convention**: The 128-bit IPv6 address is written as **eight 16-bit integers** (using hexadecimal digits for each integer)

  CEDF:BP76:3245:4464:FACE:2E50:3025:DF12

- **Short notation:**
- Abbreviations of leading zeroes:
  CEDF:BP76:0000:0000:009E:0000:3025:DF12 → **CEDF:BP76:0:0:9E :0:3025:DF12**
- ":0000:0000:0000" can be written as "::"
  CEDF:BP76:0:0:FACE:0:3025:DF12 → **CEDF:BP76::FACE:0:3025:DF12**
- IPv6 addresses derived from IPv4 addresses have 96 leading zero bits. Convention allows to use IPv4 notation for the last 32 bits.
  ::80:8F:89:90 → **::128.143.137.144**

IPv6 Provider-Based Addresses

- The first IPv6 addresses will be allocated to a provider-based plan

010	Registry ID	Provider ID	Subscriber ID	Subnetwork ID	Interface ID

- Type: Set to "010" for provider-based addresses
- Registry: identifies the agency that registered the address (type+**Registry ID = 8 bits)**

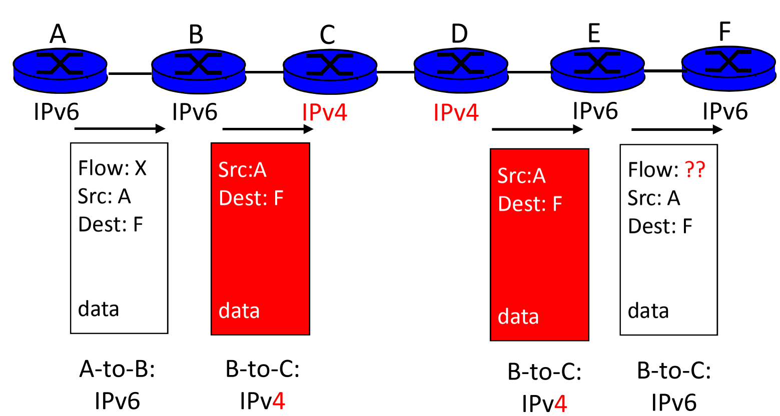
The following fields have a variable length (recommeded length in "()")

- Provider: Id of Internet access provider *(16 bits)*
- Subscriber: Id of the organization at provider *(24 bits)*
- Subnetwork: Id of subnet within organization *(32 bits)*
- Interface: identifies an interface at a node *(48 bits)*

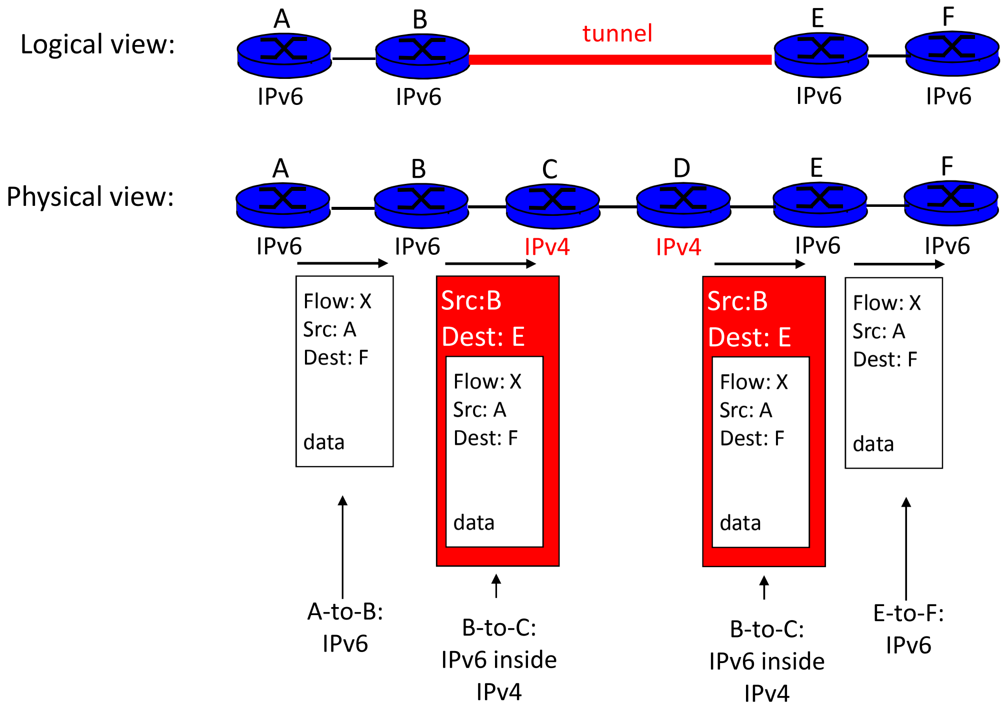
Transition From IPv4 To IPv6

- Not all routers can be upgraded simultaneous
  - no “flag days”
  - How will the network operate with mixed IPv4 and IPv6 routers?
- Two proposed approaches:
  - *Dual Stack*: some routers with dual stack (v6, v4) can “translate” between formats
  - *Tunneling:* IPv6 carried as payload in IPv4 datagram among IPv4 routers

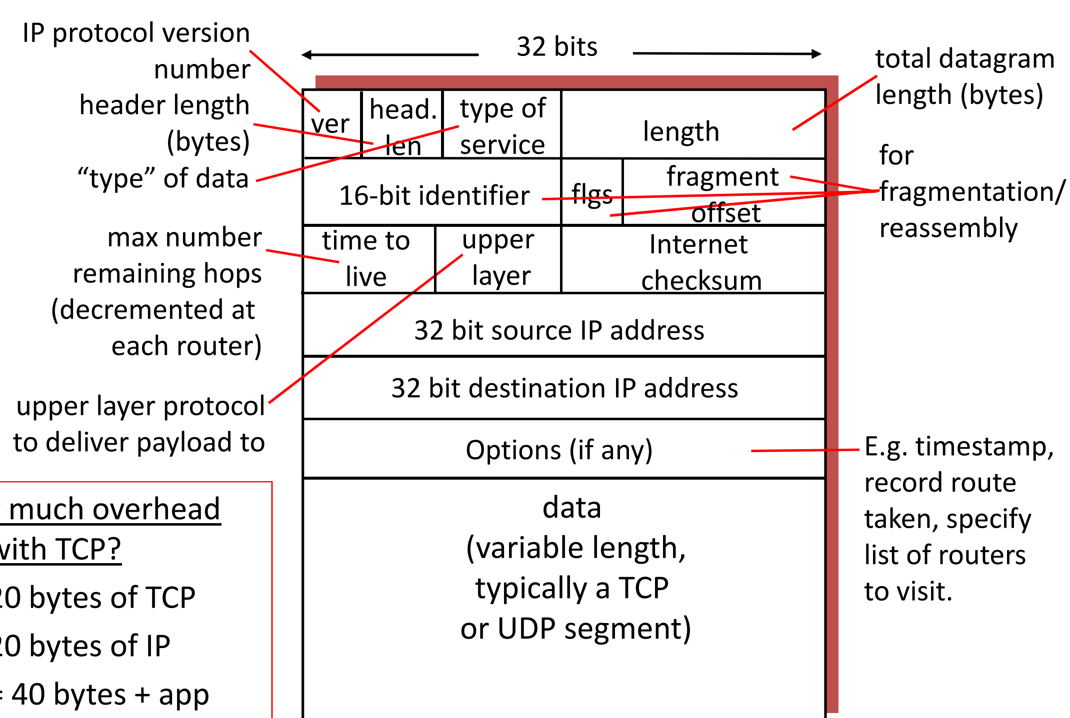
Dual Stack Approach



Tunneling



IPv4 datagram format

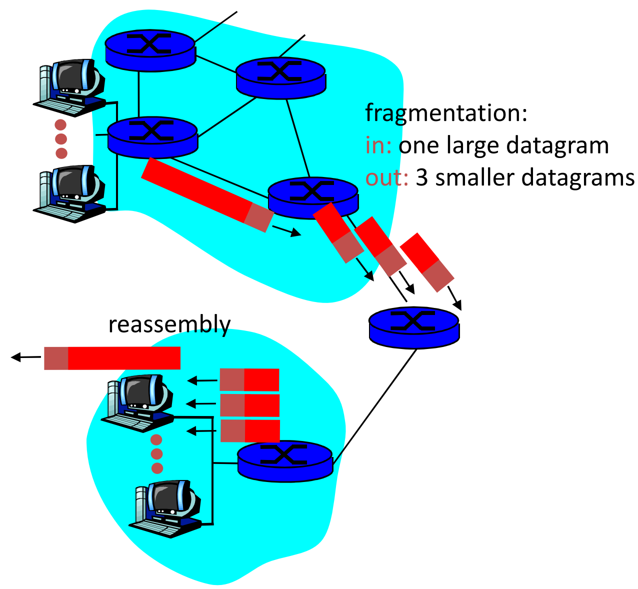


how much overhead with TCP?

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

IP Fragmentation & Reassembly

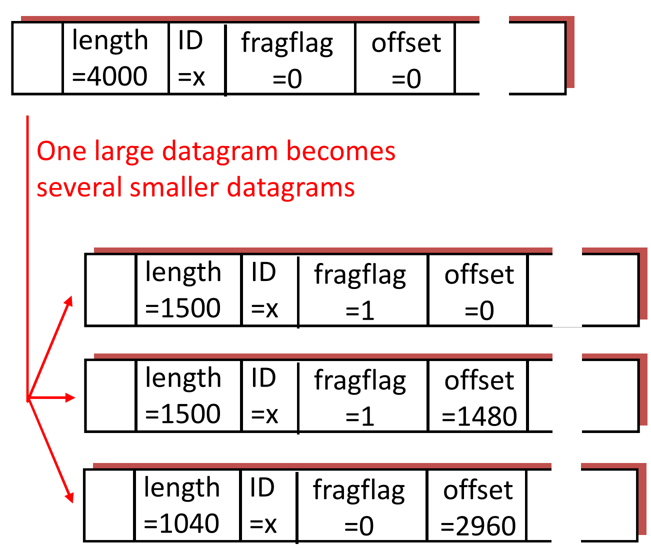
- network links have MTU (max.transfer size) - largest possible link-level frame.
  - different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify,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



IP Fragmentation and Reassembly

Example

- 4000 byte datagram
- MTU = 1500 bytes



ICMP: Internet Control Message Protocol

- used by hosts, routers, gateways to communication 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

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

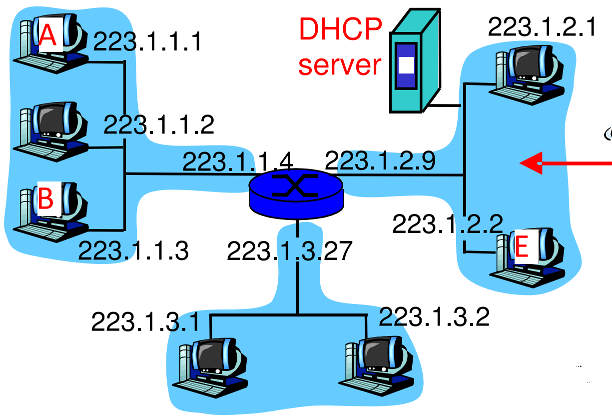
Allows reuse of addresses (only hold address while connected an "on"

Support for mobile users who want to join network (more shortly)

DHCP overview:

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

DHCP client-server scenario



arriving DHCP client needs address in this network

DHCP client-server scenario