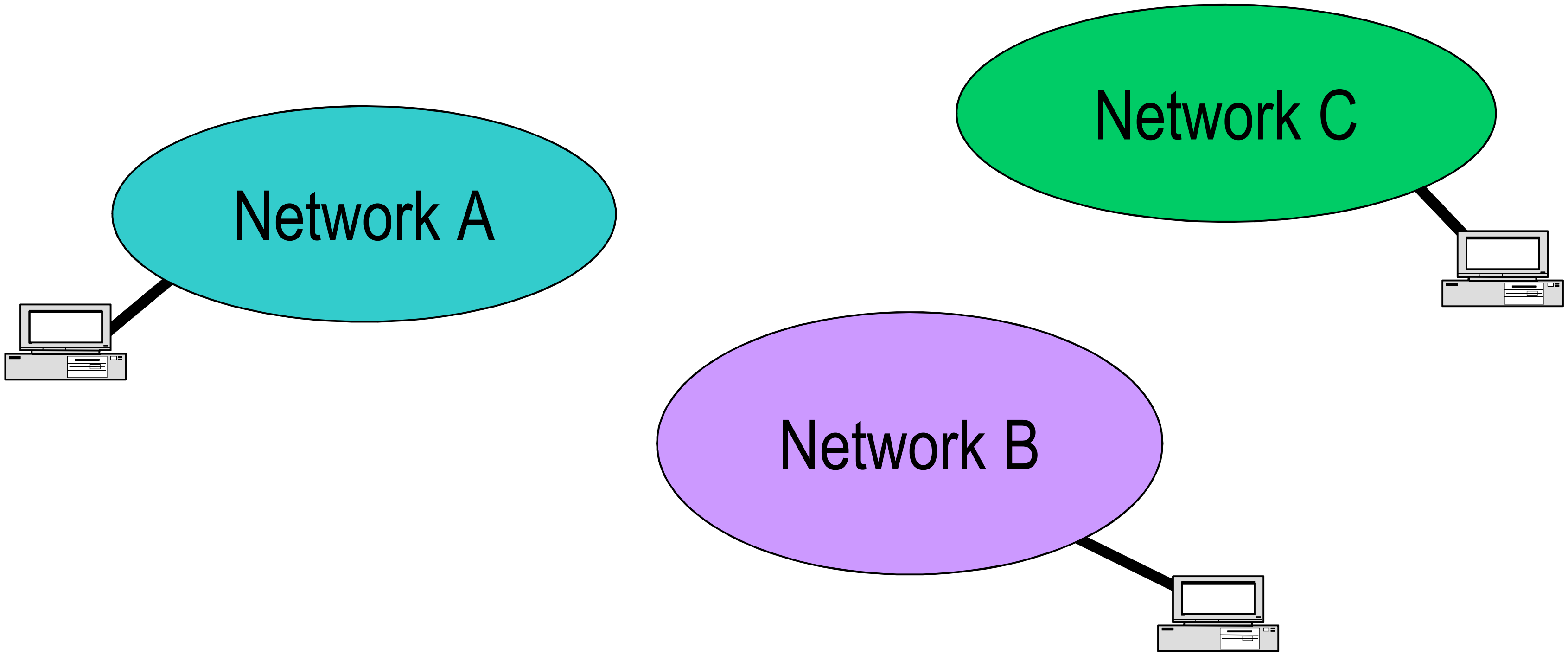
IP Addressing

- Internetworking (with TCP/IP)
- Classful addressing
- Subnetting and Supernetting
- Classless addressing

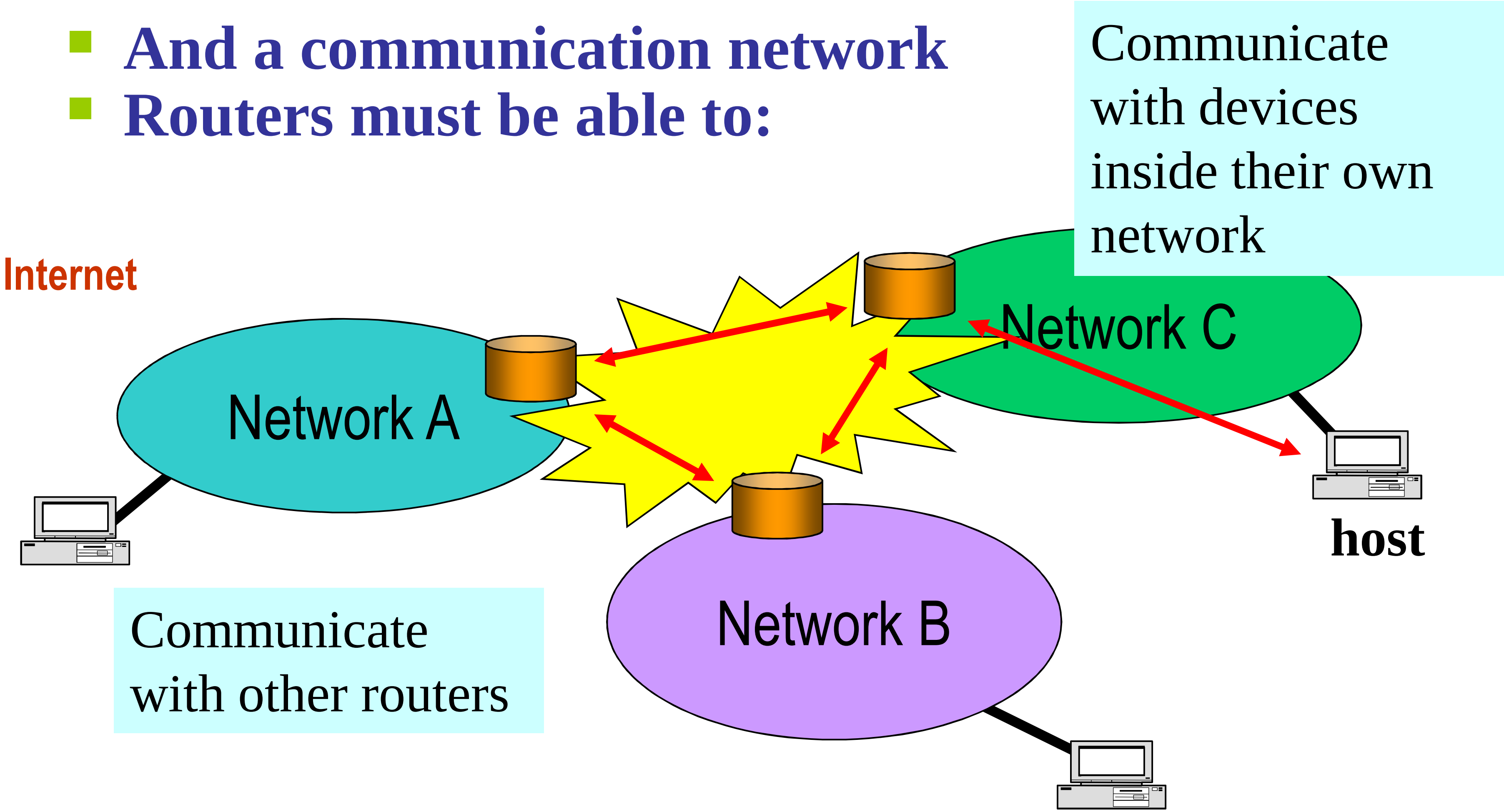
Internetworking

- **The concept of *internetworking:* we need to make different networks communicate with each other, even when they are physicall very different, in order to exchange information**



Internetworking

- To do this, we must add new devices called **gateways** or **routers**
- And a communication network
- Routers must be able to:



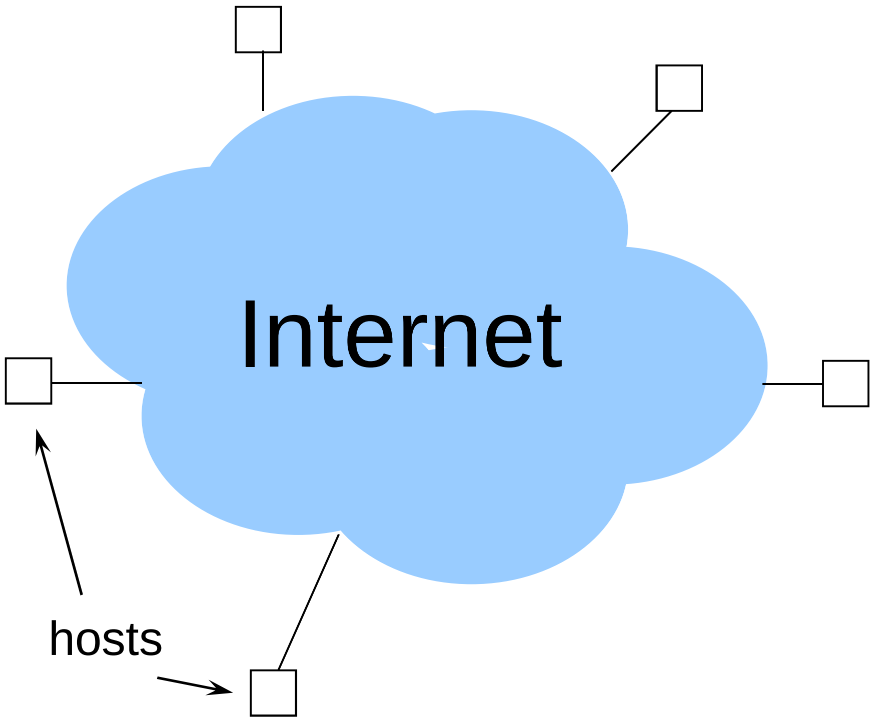
Important (1)

- **The Internet may be thought as a «big» network, similar to any other *physical* network.**
- **The big difference lies in the fact that the Internet is a *virtual (logical) network*, designed entirely «from scratch» and realized using exclusively software.**
- **For this reason, those who designed the Internet were perfectly free to choose packet length and format, delivery techniques, etc…**

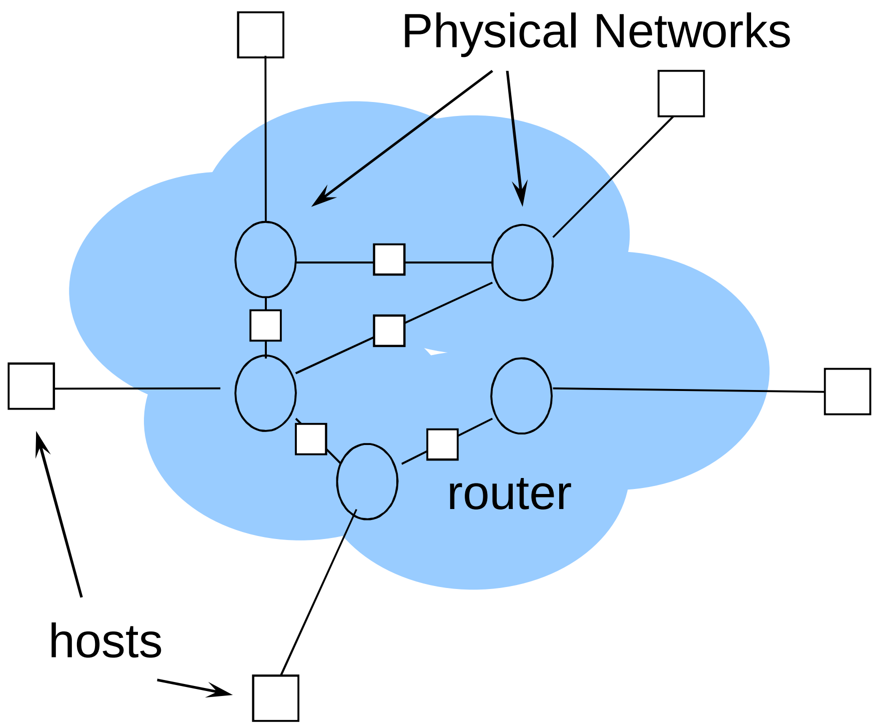
Important (2)

- **It is important to understand the Interent is <u>not</u> a new type of physical network.**
- **Internet is, instead, a way to interconnect the physical networks that already exist, and a set of «rules» (protocols) that permit to use such networks, in order to allow end users to interact and exchange information.**

Important (3)



The USER point of view: the user sees the TCP/IP network (the Internet); where each host (PC) is (or seems to be) connected to a single, big network.



This is the real structure of the Internet; with physical networks and routers that guarantee their interconnection

TCP/IP Networks

- Packet (or datagram) Networks
- Technology for interconnecting different networks and devices
- The core is the Internet Protocol (IP) (RFC 791, 1981)

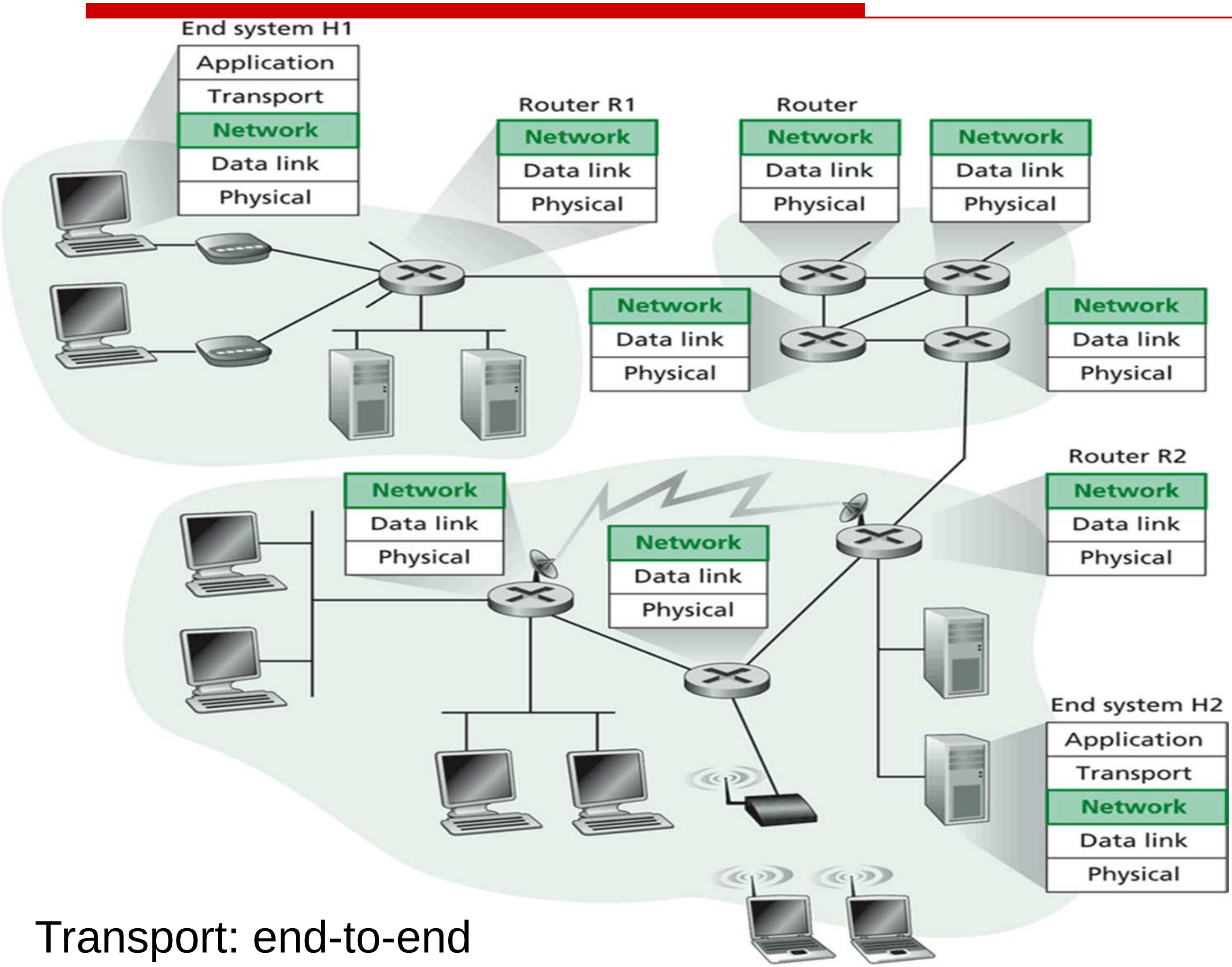
IP Communication Services

- Addressing: assign a universal and unique address
- Best-effort (hop-by-hop) Transfer: delivery and integrity are not ensured
- Fragmentation/De-fragmentation: if required by the local network

Transport Layer Services

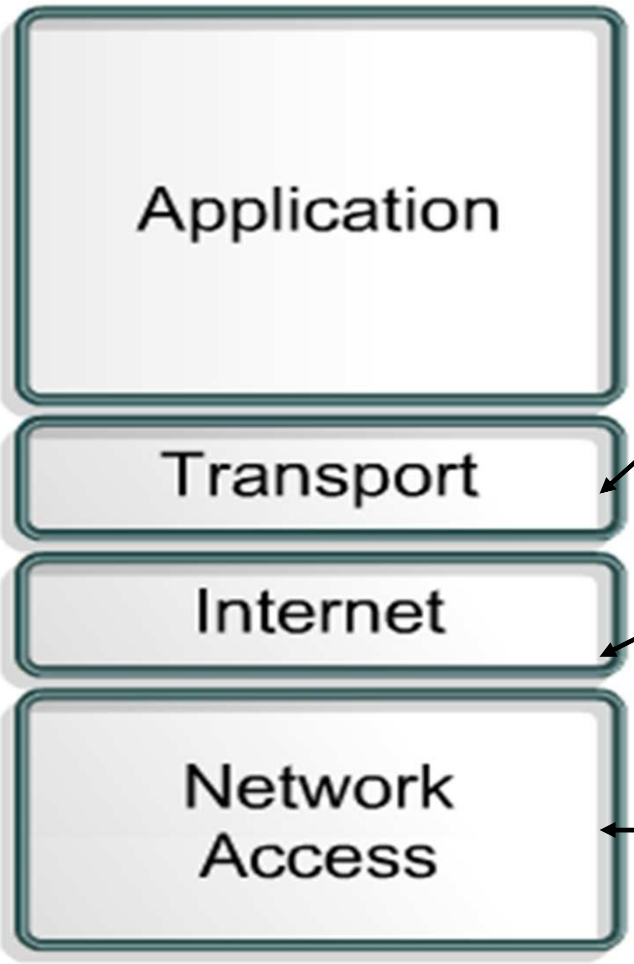
- *End To End Segments Delivery*: from process to process
  - TCP (*Transport Control Protocol*, RFC 793, '81)
    - Connection Oriented
    - *Reliable Delivery*
    - Flow Control
  - UDP (*User Datagram Protocol*, RFC 768 '81)
    - Connectionless
    - *Best Effort Delivery*

A Sample Network



Transport: end-to-end
Network: hop-by-hop

Addresses in TCP/IP Networks



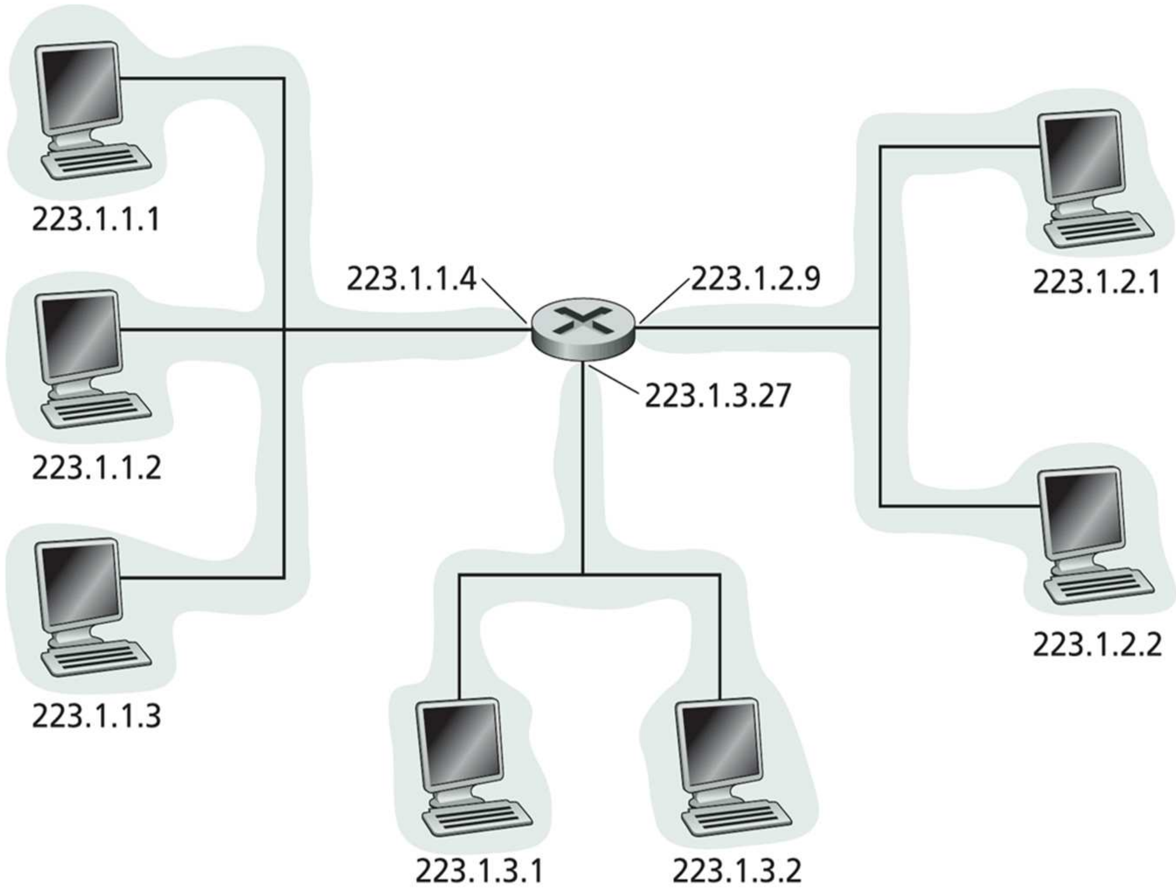
Port Address: destination process address (16 bit)

IP Address: internetworking address (32 bit)

Physical Address: depends on the underlying technology

IP Addressing

- Each *host* is characterized by one or more interfaces
- Need: uniqueness within the network
- Solution: one address for each interface



Source: Computer Networking, J. Kurose

IPv4 (RFC 791)

- 32 bits, 4 groups of 8 bits each (byte)

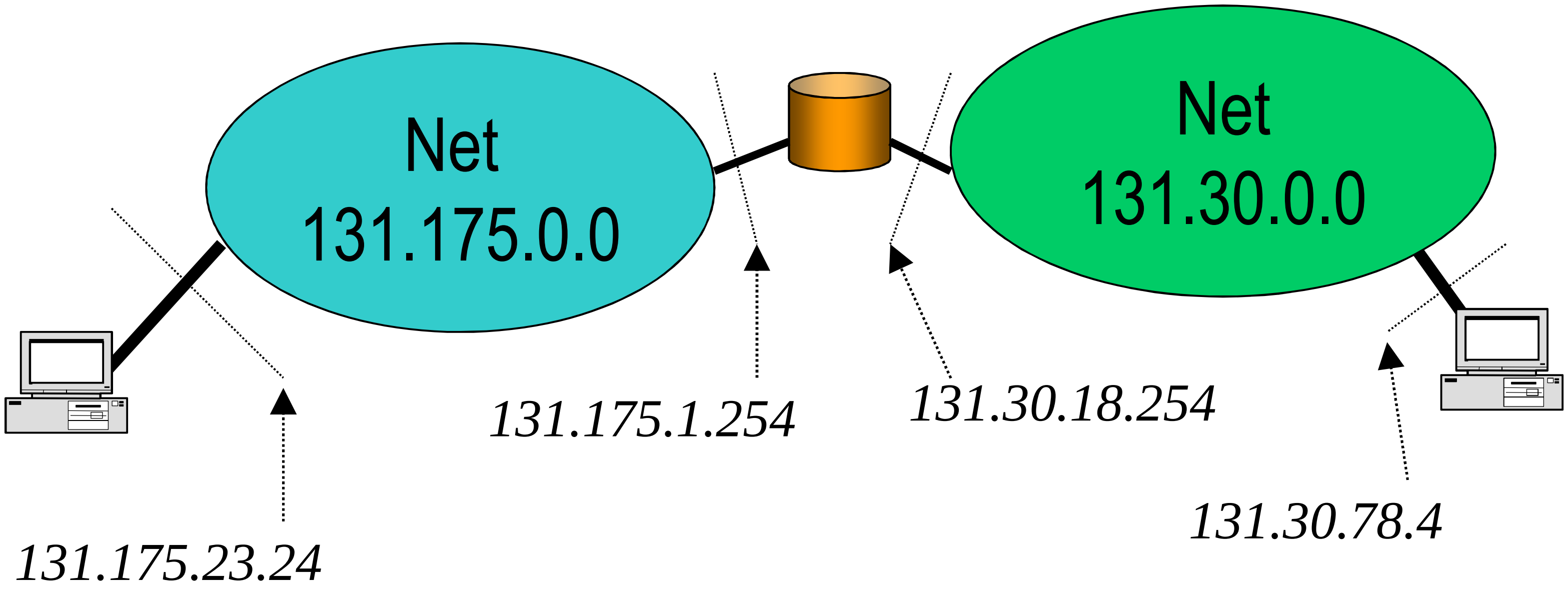
1 0 0 0 0 0 1 1	1 0 1 0 1 1 1 1	0 0 0 1 0 1 0 1	0 0 0 0 0 0 0 1

- 2^{32} = 4,294,967,296 available addresses
- Addresses are usually represented with decimal notation (each field in the range 0-255)

131.175.123.242

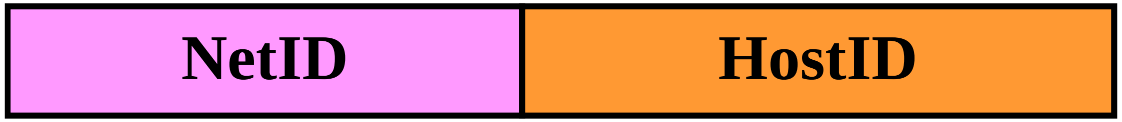
IP Addressing

- The IP address points to an interface between a host and the network
- Multi-interface devices must have multiple IP addresses

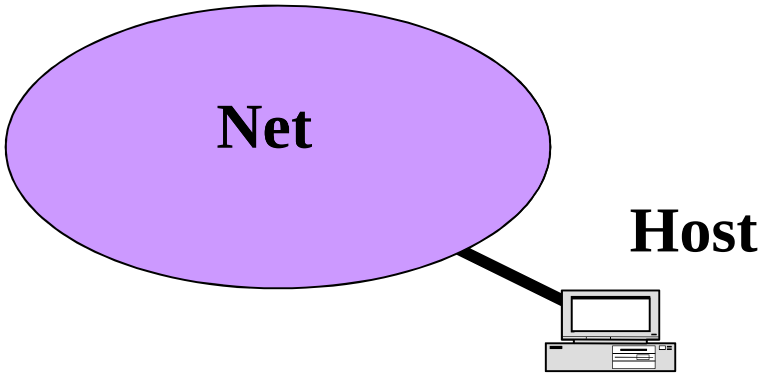


Addresses Management: Classful Way

- The address is divided into two fields (Hierarchical Structure)

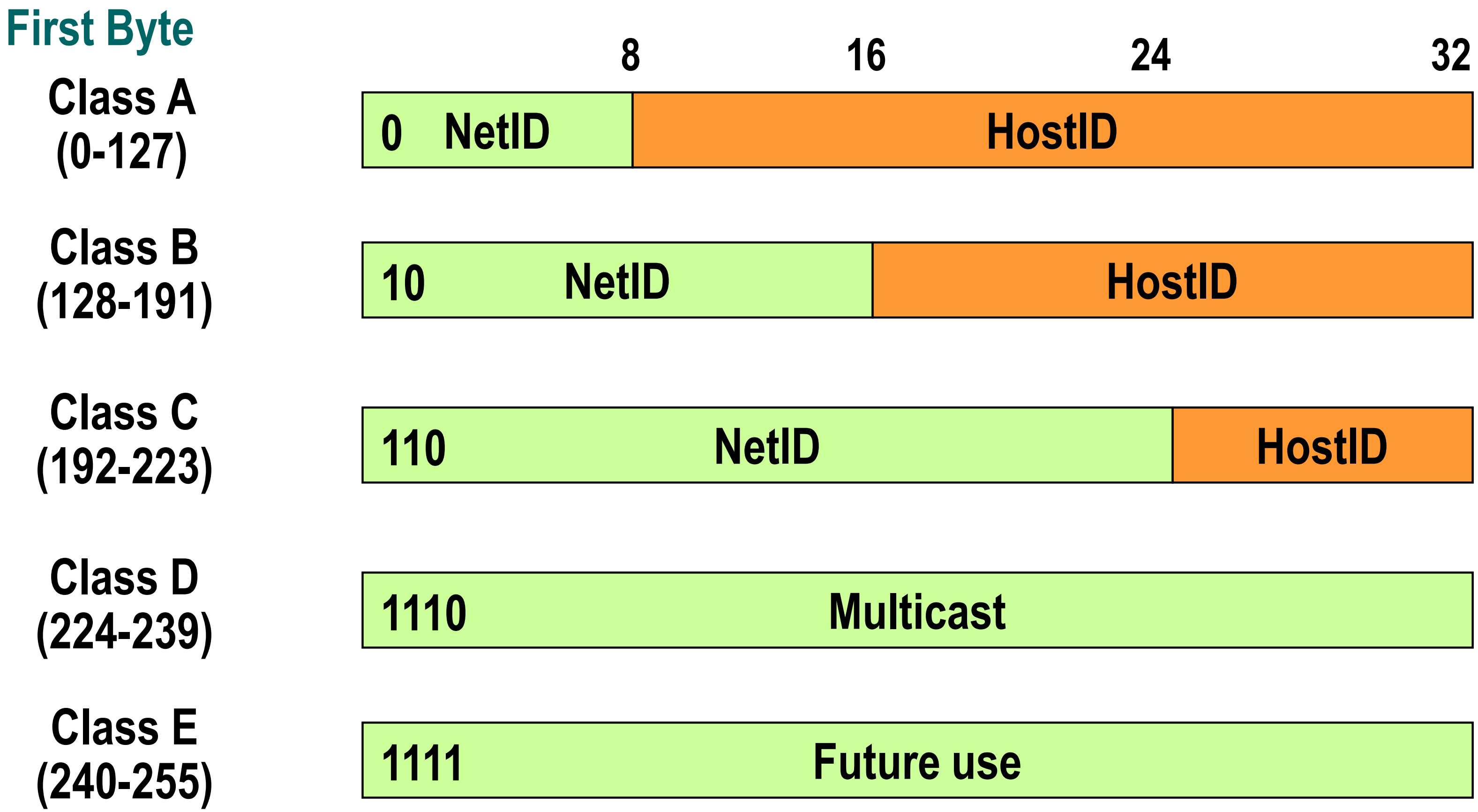


  - *NetID* (network ID) identifies the network
  - *HostID* (host ID) identifies the host within the network



All the *hosts within the same network do have* *__the same__* *network ID*

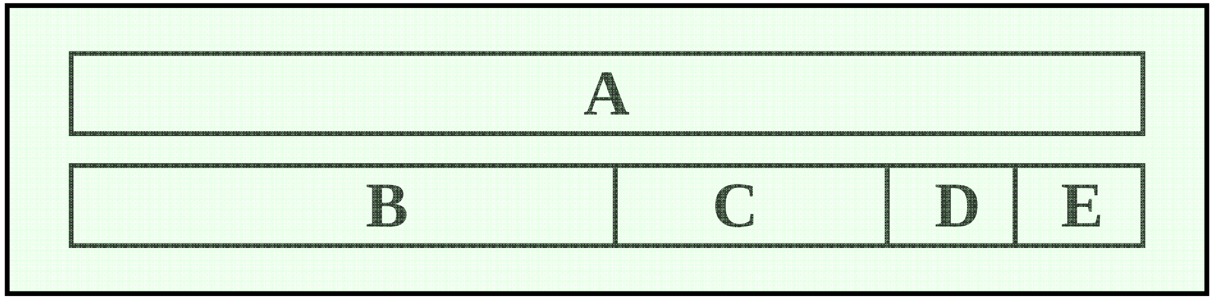
Classes



- Not all the addresses are available

Classful Addressing

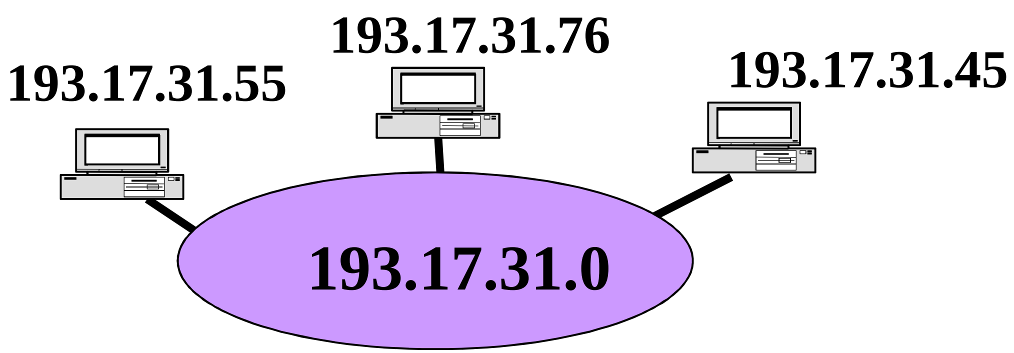
- The classes differentiate on the basis of the bits dedicated to the *NetID* and the *HostID*



Class	Addresses #	Percentage
A	2^{31}=2,147483648	50%
B	2^{30}=1,073,741,824	25%
C	2^{29}=536,870,912	12.5%
D	2^{28}=268,435,456	6.25%
E	2^{28}=268,435,456	6.25%

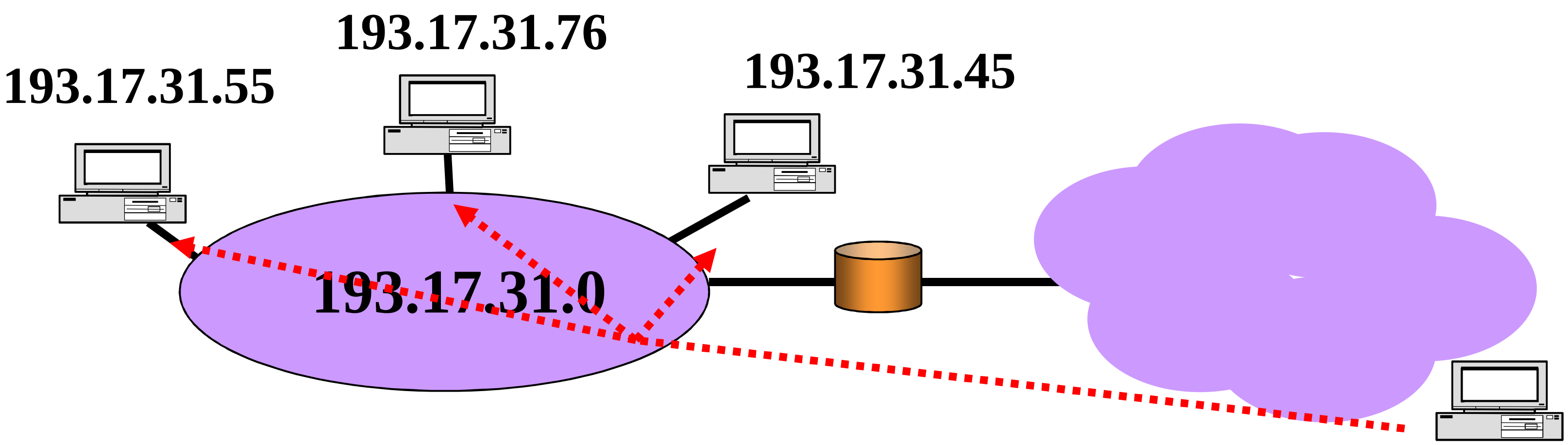
Special Addresses

- Same Network ID
  - *HostID* field set to 0
  - Identifies the network in the NetID
  - Used in the *routing tables*
- Examples:
  - class B : 131.175.0.0
  - class C: 193.17.31.0



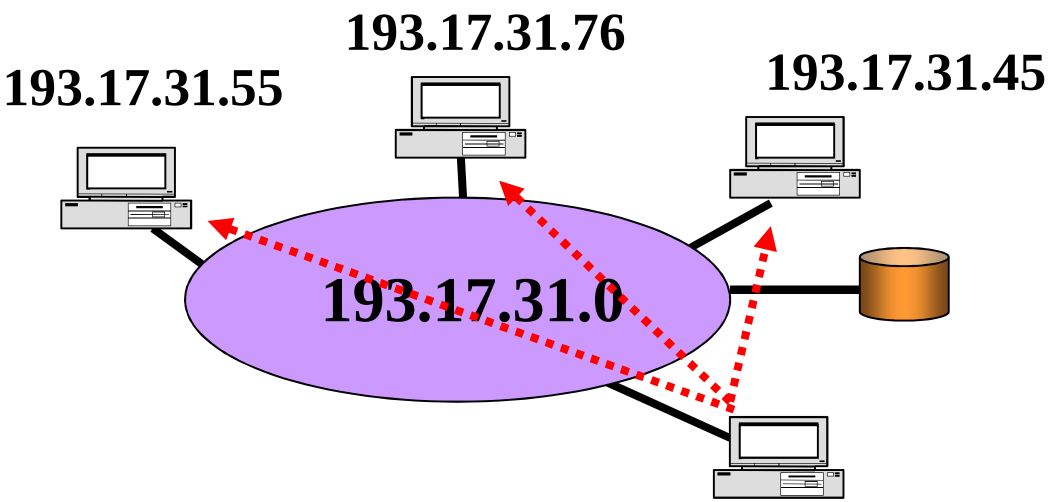
Special Addresses

- *Direct Broadcast*:
  - *HostID* field set to 1
  - Broadcast within the network specified by the *NetID*
  - used as destination field of an IP packet
  - example: 193.17.31.255



Special Addresses

- Limited *broadcast*:
  - An only 1s address (255.255.255.255)
  - *Broadcast* within the same network of the sender
  - The packet is filtered by the router
  - Used as destination field of an IP packet



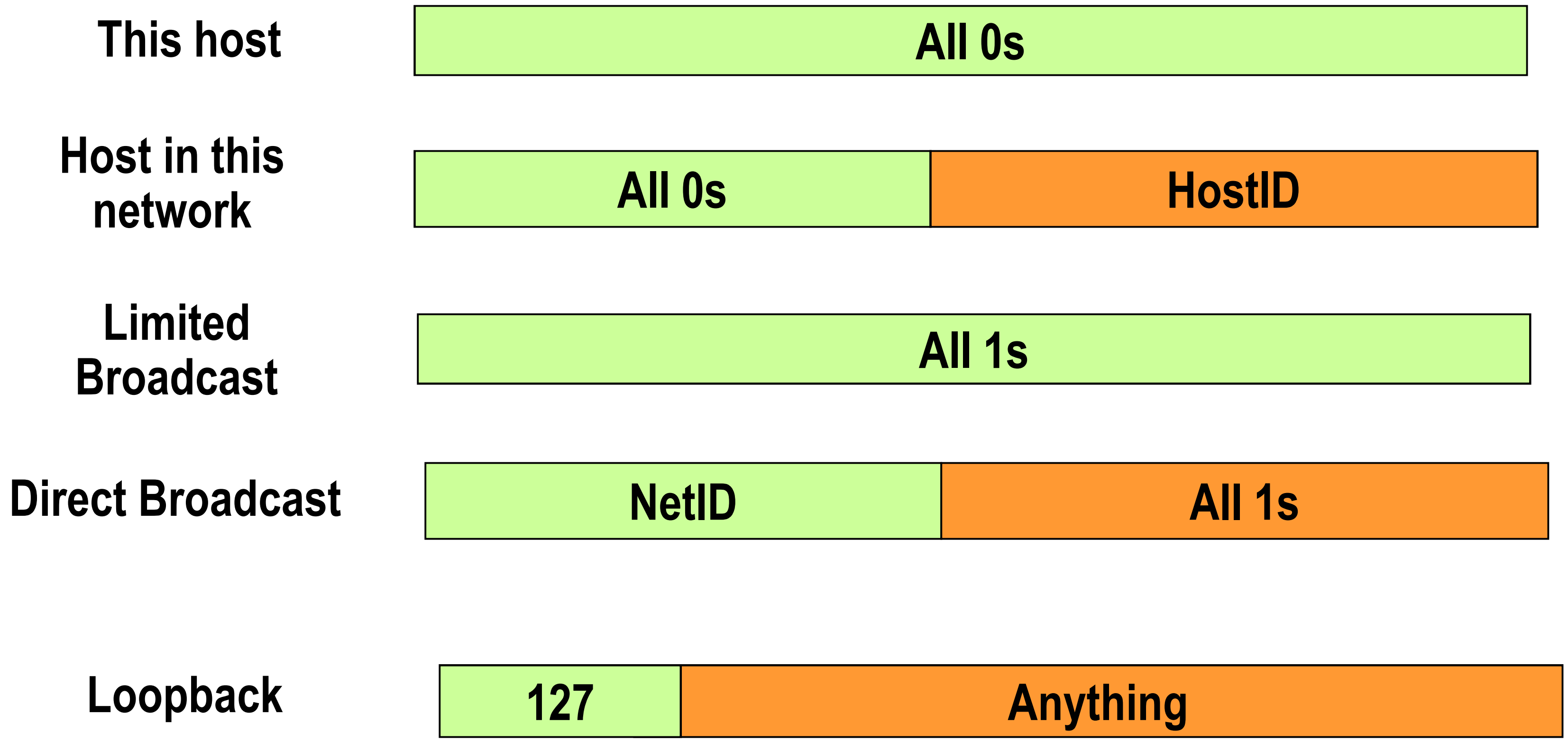
Special Addresses

- *NetID* field set to "all 0s", points to the host within the same network of the sender whose address fills the *host field*
  - Used as destination field, filtered by routers
  - example: 0.0.21.173 (class B network)

- An "all 0s" address points the sender of the packet
  - Used as source field when in the IP address acquisition phase
  - example: 0.0.0.0

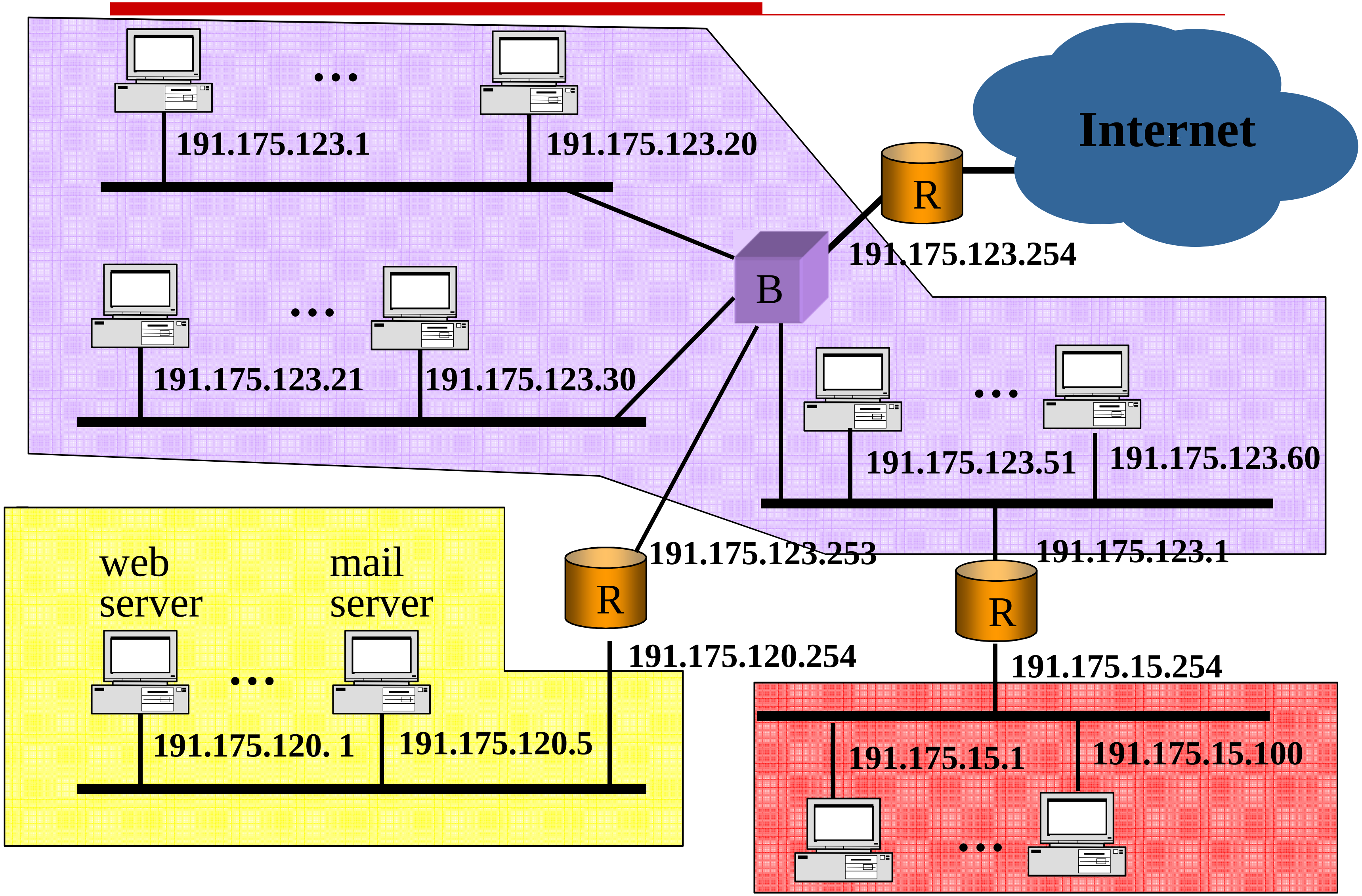
- First byte set to 127 is the *loopback* address on the very same host
  - Used in the OS to test the operation of the network stack
  - example: 127.0.0.0

Special Addresses: summary



- The first two addresses can be used only during startup phases, and they are NEVER a valid destination address
- The 3rd e 4th addresses are never a valid source address
- The 5th address should never be seen in a network

An Addressing Plan



Classful Addressing: Shortcomings

- Class A:
  - 125 NetID
  - 16,777,216 HostID for each network (!!Too Many!!)
- Class B:
  - 16,368 NetID
  - 65,536 HostID for each network (!!Still Too Many!!)
- Class C:
  - 2,096,902 NetID
  - 256 HostID (!!Too Few!!)

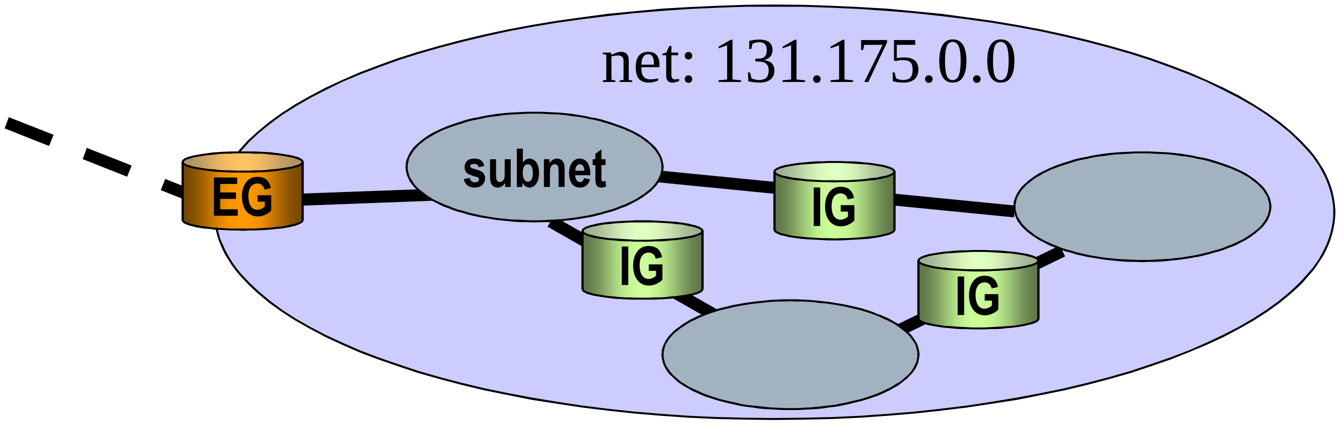
Classful Addressing Limitation

- Low Flexibility (intrinsic)
- Limited Number of Addresses (intrinsic)
- Growing Demand for addresses (external)

- Possible Solutions:
  - *Subnetting/Supernetting*
  - *Classless* addressing
  - Private addressing (*Intranet* and NAT)

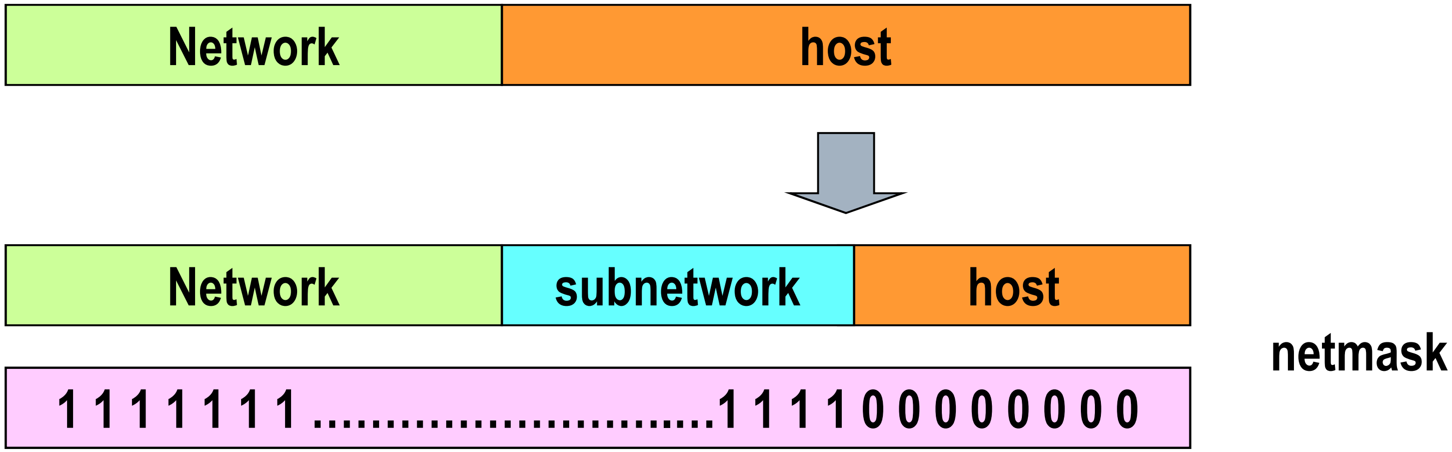
Subnetting (RFC 950)

- *Example:* private organizations with a class B address assigned (2^{16} -2 = 65534 host addresses) may develop *Intranets* with few hundreds (tens) hosts subnets



Subnetting

- Basic Idea: split up the IP address introducing a further hierarchical level
- The *host* field is split into a *subnet* field and a new shorter *host* field



- The splitting is performed through a *netmask* composed of a sequence of 1s (Net + Subnet) and a sequence of 0s (host)

Netmask

- A way to get rid of classes
- Represented in decimal notation

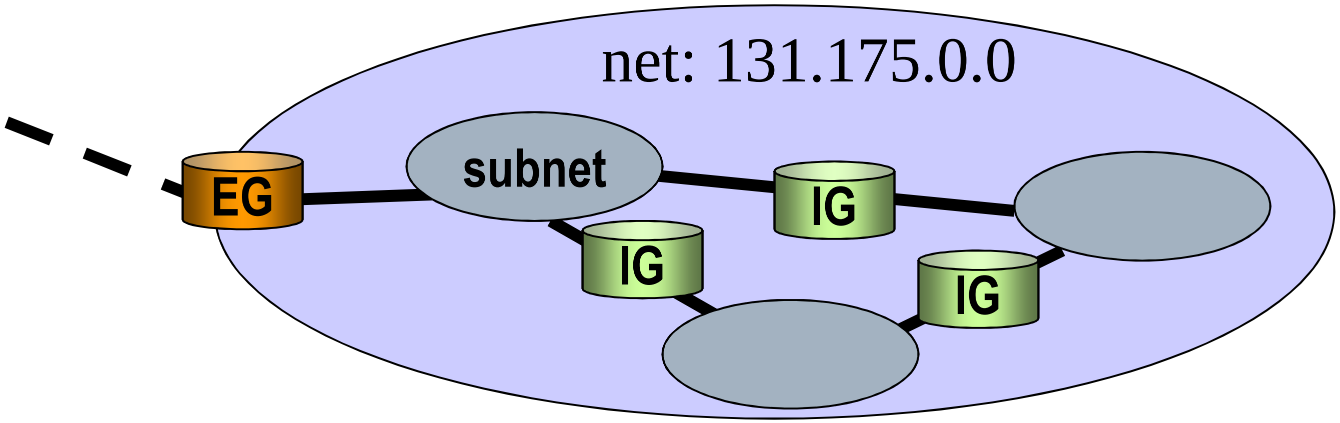
1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 | 0 0 0 0 0 0 0 0

netmask: 255.255.255.0

network: 131.175.0.0

subnets: 131.175.0.0, 131.175.1.0, …, 131.175.254.0, 131.75.255.0

Subnetting



- Routers <u>outside</u> the *Intranet* know only the network 131.175.0.0
- <u>Internal</u> *routers (*Interior Gateways, IG*)* must handle *subnets* and *netmasks*

Netmask

- The decimal representation of continuous *netmasks* :

255	1	1	1	1	1	1	1	1
254	1	1	1	1	1	1	1	0
252	1	1	1	1	1	1	0	0
248	1	1	1	1	1	0	0	0
240	1	1	1	1	0	0	0	0
224	1	1	1	0	0	0	0	0
192	1	1	0	0	0	0	0	0
128	1	0	0	0	0	0	0	0

- Alternatively the *netmask* can be represented as the number of consecutive ones (prefix):
  - 131.175.21.0/24

Ordinary Phone Network Analogy

- The hierarchical subdivision of the address is leveraged from the ordinary phone networks.

IP Realm					Telephone Nets			
141 . 14	.	192	.	192	(310)	824	-	2296
Site		Subnet		host	Area Code	Exchange		Connection

Subnetting Example (1)

- **We are given the network address: 132.78.0.0**
- **We must create networks with *at least* 500 hosts each**

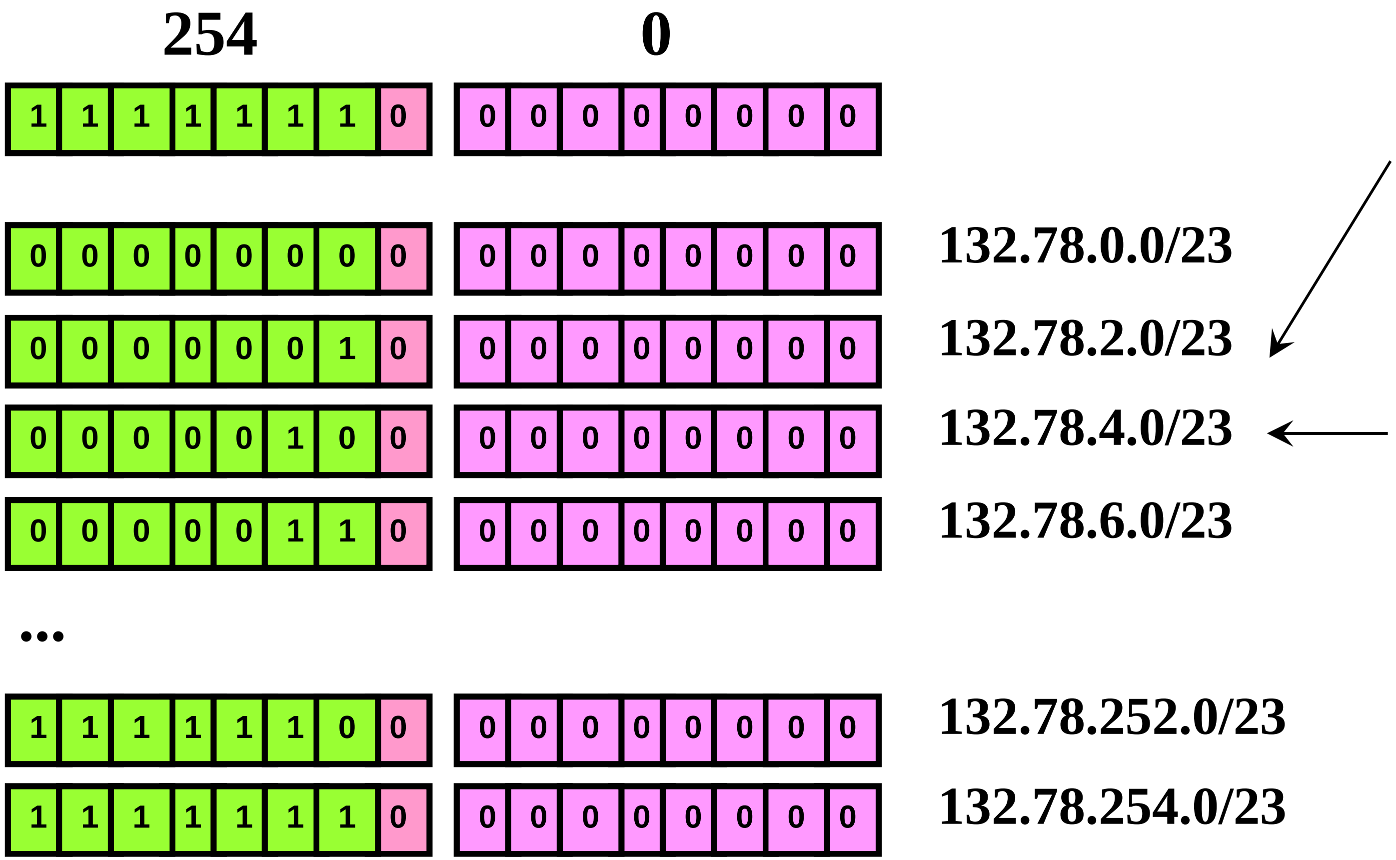
- **The NetID is 16 bits long**
- **2^9=512, so we need 9 bits for the hostID field**
- **16-9=7 bits are available for SubnetID field**
- **The netmask is composed of 16+7=23 bits**

1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 0	0 0 0 0 0 0 0 0

255.255.254.0

Subnetting Example (1)

- **Number of available subnets: 2^7=128, each of them containing up to 2^9-2=510 hosts**



What does the address 132.78.3.0 represent ?

What is the direct broadcast address for this network ?

132.78.5.255

Subnetting Example (2)

- **We are given the address: 128.234.0.0**
- **We must create <u>at least</u> 1000 «small» subnetworks**

- **The NetID is 16 bits long**
- **2^{10}=1024, so we need 10 bits for the subnetID field**
- **We still have 16-10= 6 bits fot the HostID field**
- **The netmask is therefore 16+10=26 bits long**

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0

255.255.255.192

Subnetting Example (2)

- The maximum number of hosts in each of the 1024 subnetworks is equal to 2^6-2=62 hosts

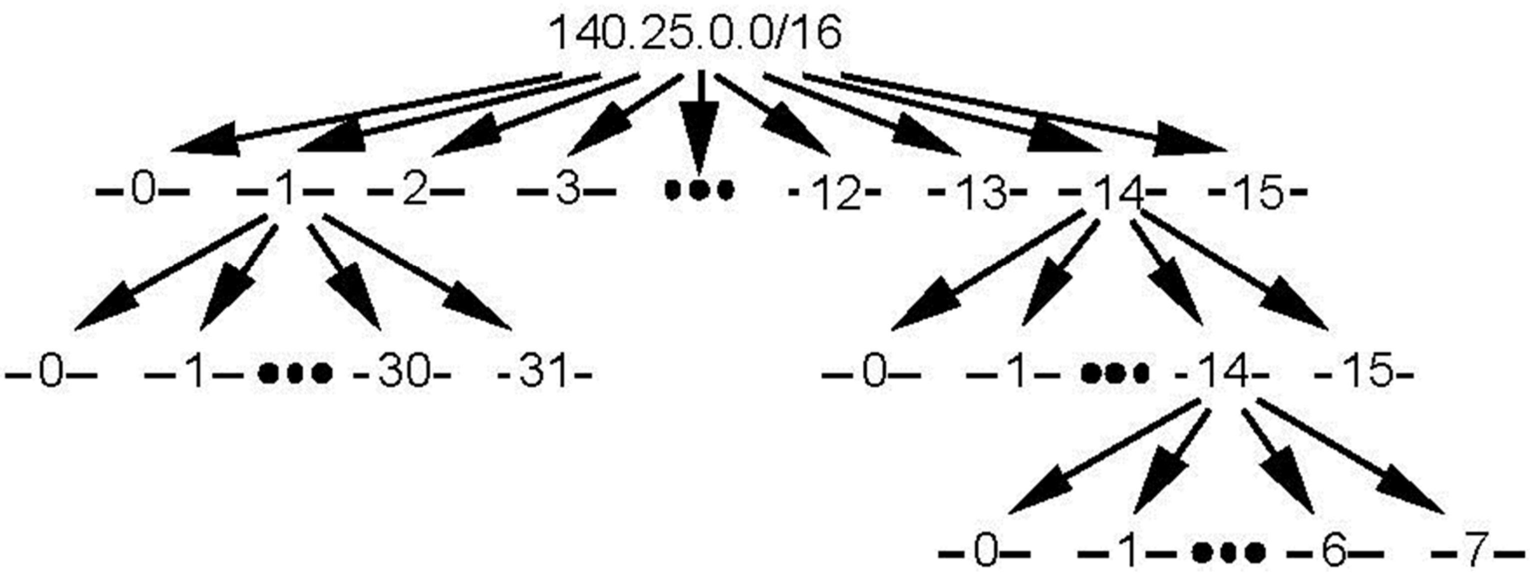
255								192								
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	128.234.0.0/26
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	128.234.0.64/26
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	128.234.0.128/26
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	128.234.0.192/26
…																
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	128.234.255.128/26
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	128.234.255.192/26

When The Subnetting Fails..

- An organization is assigned a class C address and needs to set up 3 subnets with 60 hosts and 2 subnets with 30 hosts.
- Classical Subnetting fails:
  - Subnet with 2 bits: 4 nets with 64 hosts each
  - Subnet with 3 bits: 8 nets with 32 hosts each
- Way Out ??

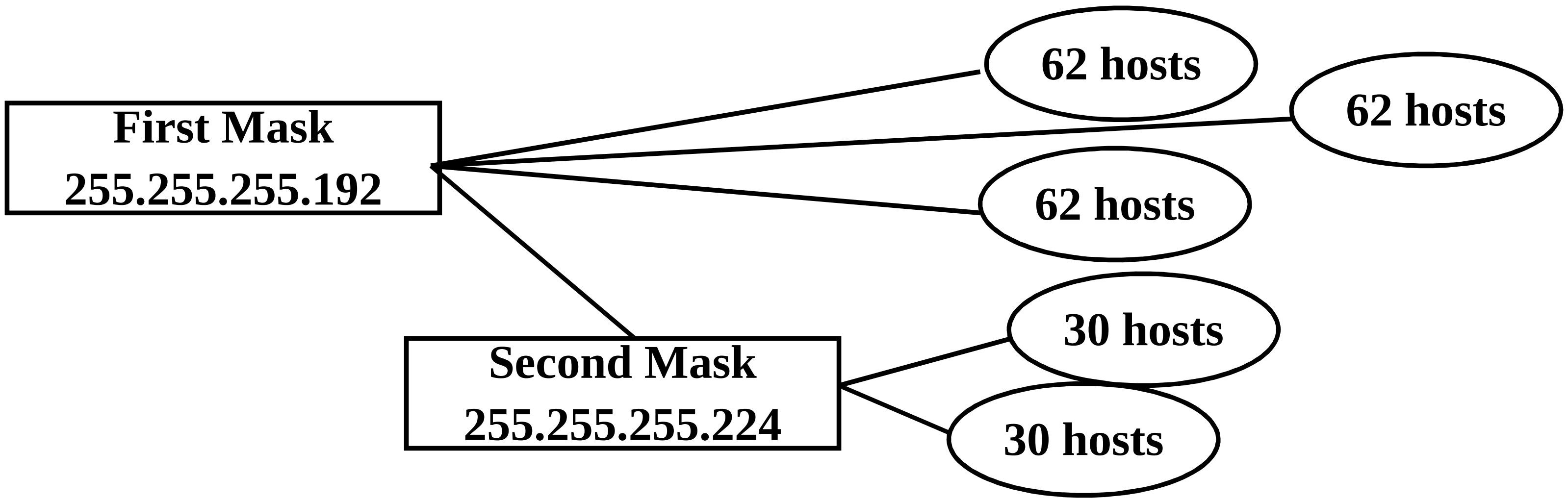
Variable Length Subnet Masks (VLSM)

- It is possible to apply subnetting in a «recursive» (or «hierarchical») way
- Example:



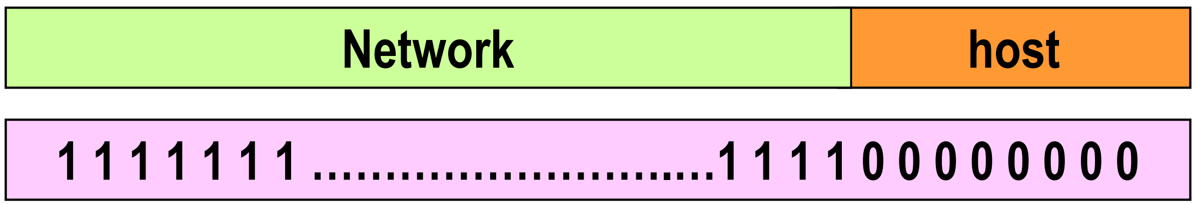
Variable Netmask

- Apply two netmasks serially:
  - The first 255.255.255.192 with 26 1s defines 4 subnets with 64 host addresses
  - Take one of the subnets above and apply to it a netmask 255.255.255.224 dividing it into two subnets with 30 hosts each



Supernetting

- Opposite as subnetting
- *Problem:* class A and B addresses exhaustion, class C addresses can serve few hosts
- *Solution:* group several class Cs to define a bigger network (Supernetting)
- *Realization:* use of *netmask*



Supernetting

- 4 contiguous class C addresses can be grouped to form a network with 1024 hosts:

193.23.136.0
193.23.137.0
193.23.138.0
193.23.139.0

→ **193.23.136.0/22**

1 1 0 0 0 0 0 1	0 0 0 1 0 1 1 1	1 0 0 0 1 0 0 0	0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 0 0	0 0 0 0 0 0 0 0

Classless Inter Domain Routing (CIDR)

- *Netmask* extension
- No more classes
- Flexible address assignment (power of two)
- Impact on routing

IP Addresses and Domain Names

- IP addresses are globally assigned
- The Internet uses symbolic names also, still globally assigned