Reti di Calcolatori

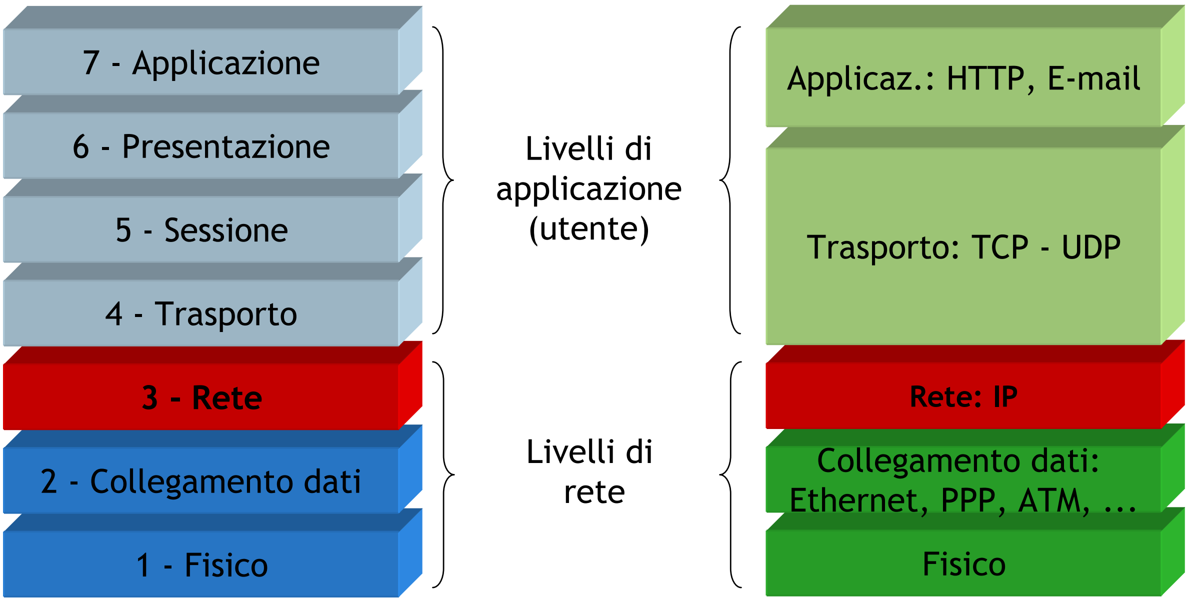
Il livello di rete

Universtità degli studi di Verona
Facoltà di Scienze MM.FF.NN.
A.A. 2009/2010
Laurea in Informatica
Docente: Damiano Carra

Acknowledgement

- ❑ *Credits*
  - *Part of the material is based on slides provided by the following authors*
    - *Jim Kurose, Keith Ross, "Computer Networking: A Top Down Approach," 4th edition, Addison-Wesley, July 2007*
    - *Douglas Comer, "Computer Networks and Internets," 5th edition, Prentice Hall*
    - *Behrouz A. Forouzan, Sophia Chung Fegan, "TCP/IP Protocol Suite," McGraw-Hill*, January 2005

Network level



7 - Applicazione

6 - Presentazione

5 - Sessione

4 - Trasporto

3 - Rete

2 - Collegamento dati

1 - Fisico

Livelli di applicazione (utente)

Livelli di rete

Applicaz.: HTTP, E-mail

Trasporto: TCP - UDP

Rete: IP

Collegamento dati: Ethernet, PPP, ATM, ...

Fisico

Introduction

Overall view

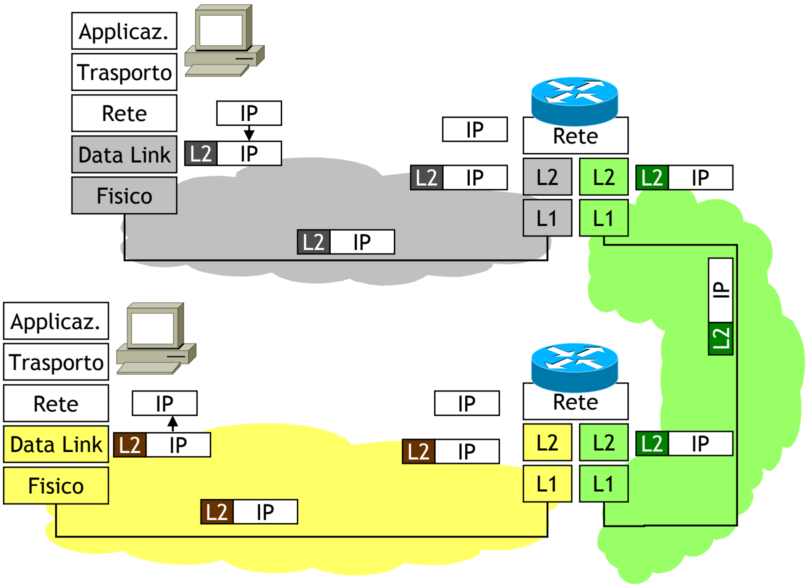
❑ Transport segment from sending to receiving host

❑ On sending side encapsulates segments into datagrams

❑ On rcving side, delivers segments to transport layer

❑ Network layer protocols in every host, router

❑ Router examines header fields in all IP datagrams passing through it



Two Key Network-Layer Functions

❑ Routing

- Determine route taken by packets from source to dest.

➔ Routing algorithms

  - Analogy (trip planning): process of planning trip from source to dest

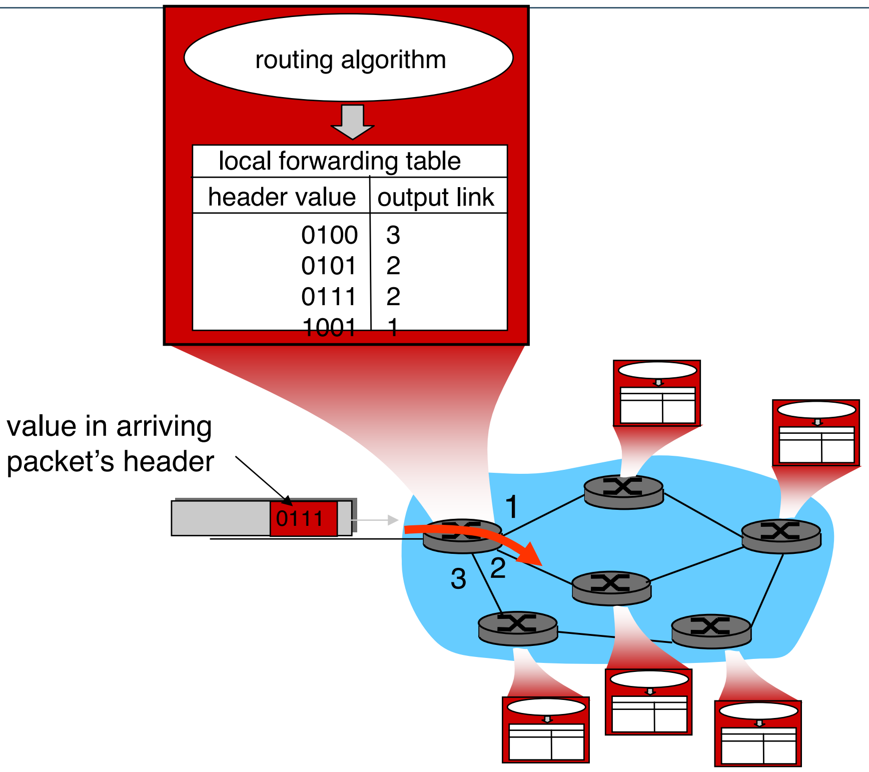
❑ Forwarding

- Move packets from router’s input to appropriate router output

  - Analogy (trip planning): process of getting through single interchange

❑ These function require a basic component: Addressing

Interplay between routing and forwarding



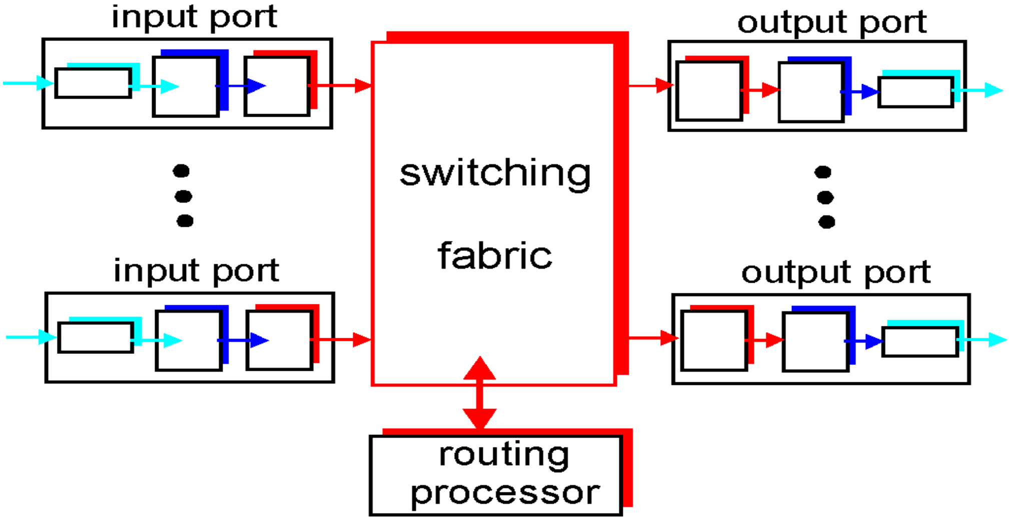
routing algorithm

local forwarding table	
header value	output link
0100	3
0101	2
0111	2
1001	1

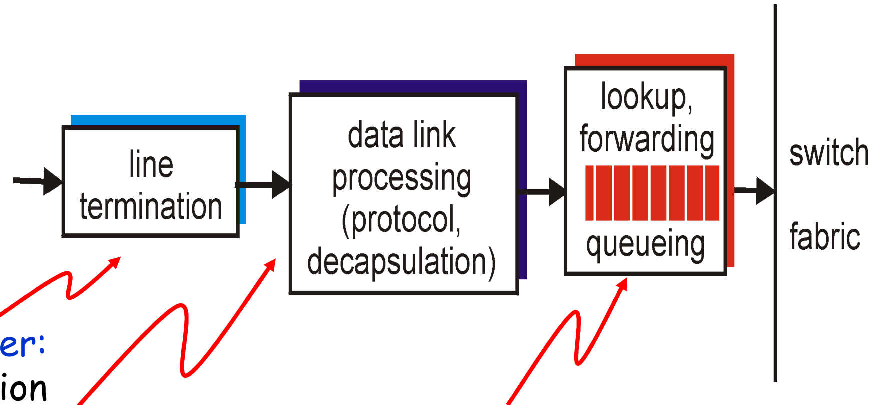
value in arriving packet's header

Router Architecture Overview

- ❑ Two key router functions:
  - run routing algorithms/protocol (RIP, OSPF, BGP)
  - forwarding datagrams from incoming to outgoing link



Input Port Functions



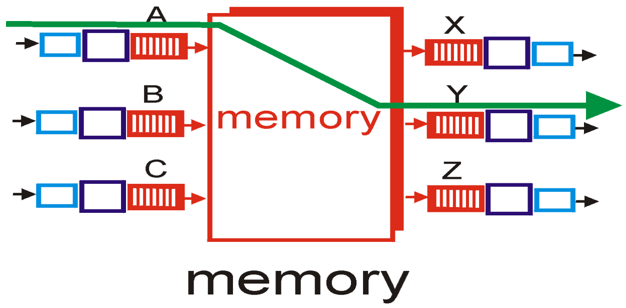
Physical layer:
bit-level reception

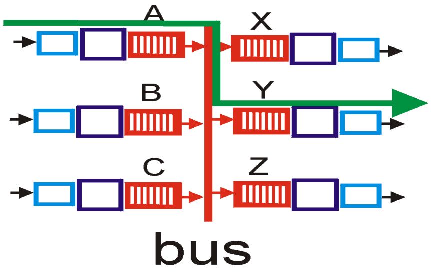
Data link layer:
e.g., Ethernet
see chapter 5

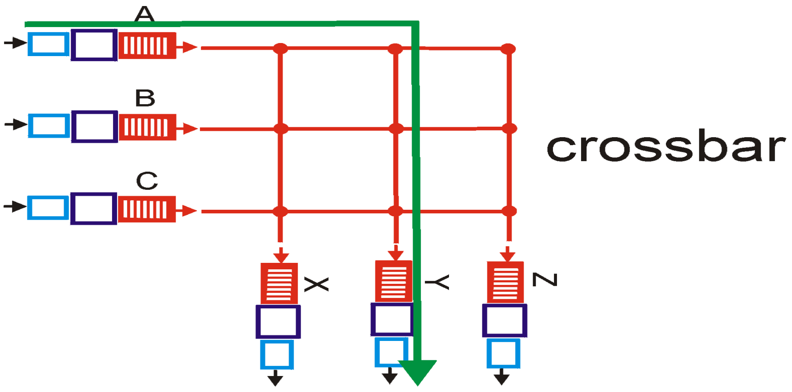
Decentralized switching:

- ❑ 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

Three types of switching fabrics

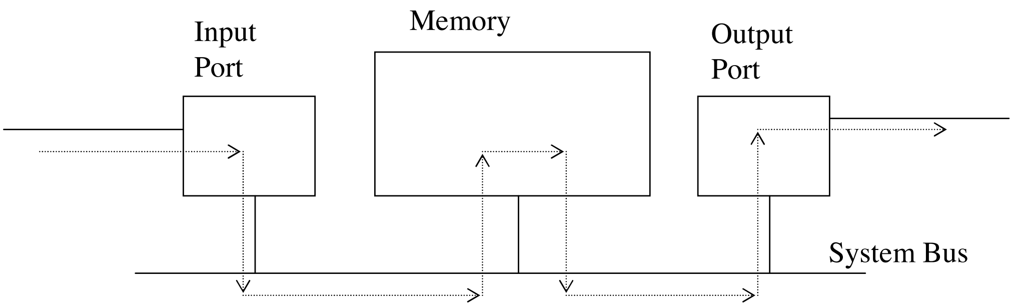






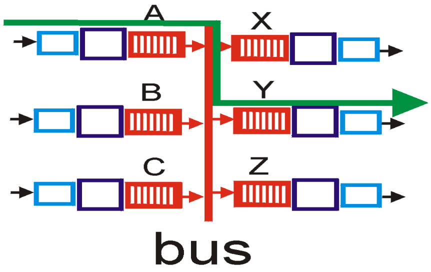
Switching Via Memory

- ❑ First generation routers:
  - traditional computers with switching under direct control of CPU
  - packet copied to system's memory
  - speed limited by memory bandwidth



Switching Via a Bus

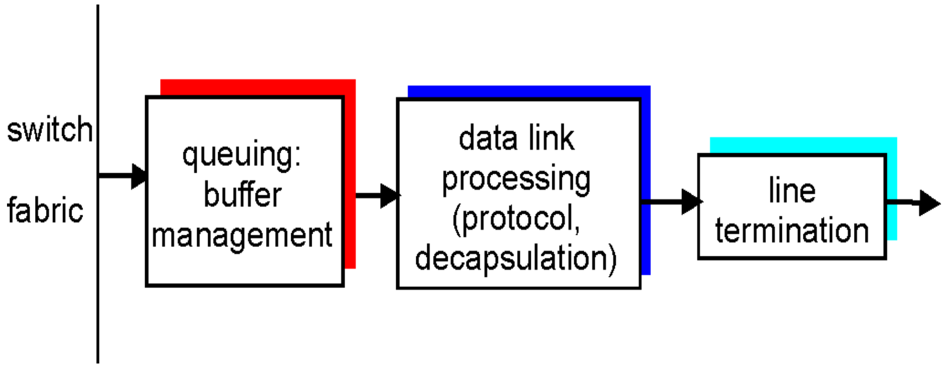
- ❑ Datagram from input port memory to output port memory via a shared bus
- ❑ Bus contention: switching speed limited by bus bandwidth
- ❑ 32 Gbps bus, Cisco 5600: sufficient speed for access and enterprise routers



Switching Via An Interconnection Network

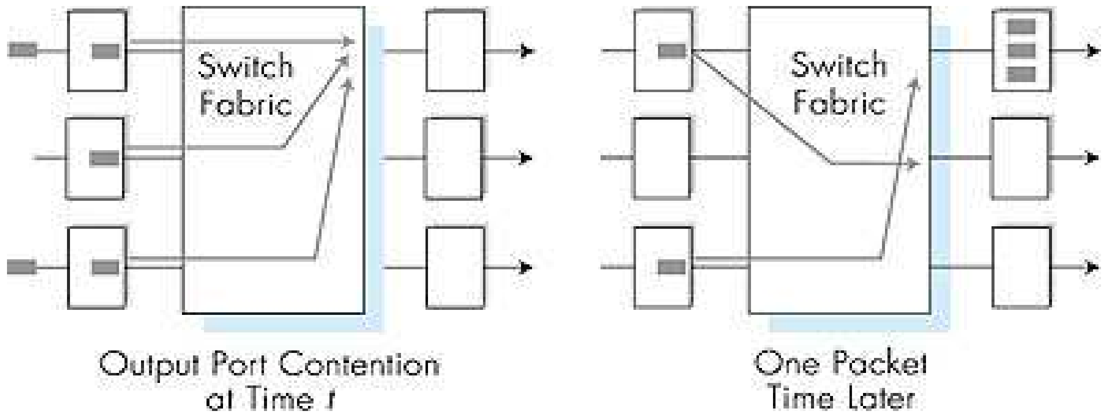
- ❑ Overcome bus bandwidth limitations
- ❑ Banyan networks, other interconnection nets initially developed to connect processors in multiprocessor
- ❑ Advanced design: fragmenting datagram into fixed length cells, switch cells through the fabric.
- ❑ Cisco 12000: switches 60 Gbps through the interconnection network

Output Ports



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

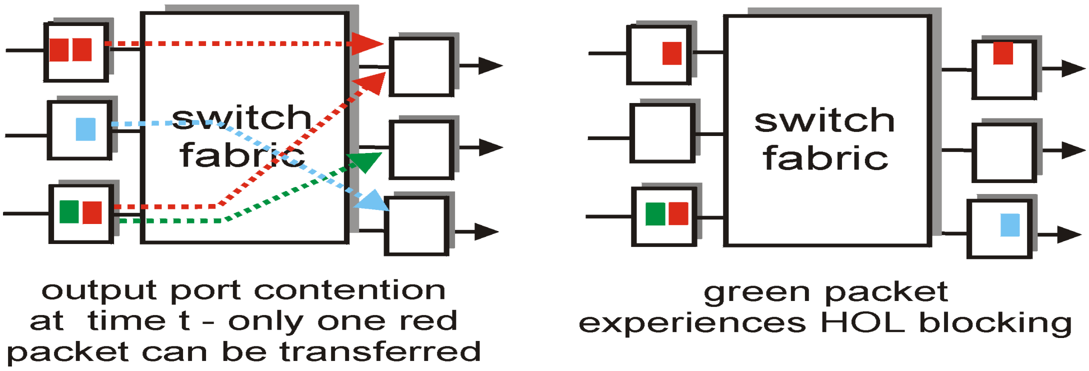
Output port queueing



- ❑ Buffering when arrival rate via switch exceeds output line speed
- ❑ Queueing (delay) and loss due to output port buffer overflow!

Input Port Queuing

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


switch fabric
output port contention at time t - only one red packet can be transferred
switch fabric
green packet experiences HOL blocking

IP Protocol

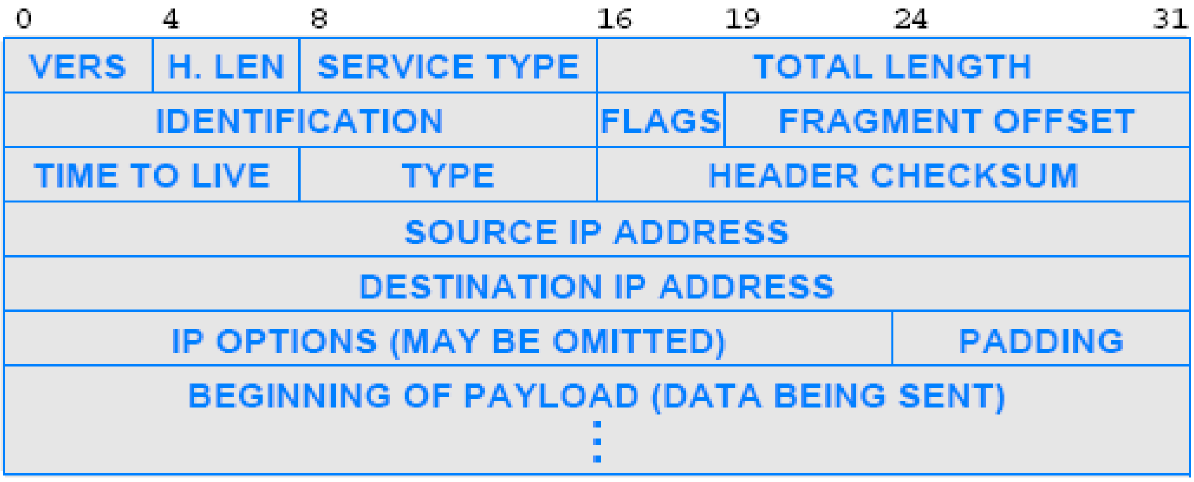
The IP Datagram

- TCP/IP uses the name IP datagram to refer to a packet
- Each datagram consists of a header
  - 20 to 60 bytes in length and contains information essential to routing and delivery
- followed by data area (payload)
  - The amount of data carried in a datagram is not fixed
  - The size of a datagram is determined by the application that sends data
  - A datagram can contain as little as a single octet of data or at most 64K

The IP Datagram Header Format

- ❑ What does a datagram header contain?
  - It contains information used to forward the datagram
- ❑ A datagram head contains information, such as:
  - the address of the source (the original sender)
  - the address of the destination (the ultimate recipient)
  - and a field that specifies the type of data being carried in the payload
- ❑ Each address in the header is an IP address
  - MAC addresses for the sender and recipient do not appear
- ❑ Each field in an IP datagram header has a fixed size
  - which makes header processing efficient

The IP Datagram Header Format



<table>
<tr><th>0</th><th>4</th><th>8</th><th>16</th><th>19</th><th>24</th><th>31</th></tr>
<tr><td>VERS</td><td>H. LEN</td><td>SERVICE TYPE</td><td colspan="4">TOTAL LENGTH</td></tr>
<tr><td colspan="3">IDENTIFICATION</td><td>FLAGS</td><td colspan="3">FRAGMENT OFFSET</td></tr>
<tr><td colspan="2">TIME TO LIVE</td><td>TYPE</td><td colspan="4">HEADER CHECKSUM</td></tr>
<tr><td colspan="7">SOURCE IP ADDRESS</td></tr>
<tr><td colspan="7">DESTINATION IP ADDRESS</td></tr>
<tr><td colspan="5">IP OPTIONS (MAY BE OMITTED)</td><td colspan="2">PADDING</td></tr>
<tr><td colspan="7">BEGINNING OF PAYLOAD (DATA BEING SENT)
⋮</td></tr>
</table>

The IP Datagram Header Format

- ❑ VERS
  - Each datagram begins with a 4-bit protocol version number
- ❑ H.LEN
  - 4-bit header specifies the number of 32-bit quantities in the header
  - If no options are present, the value is 5
- ❑ SERVICE TYPE
  - 8-bit field that carries a class of service for the datagram
    - seldom used in practice
- ❑ TOTAL LENGTH
  - 16-bit integer that specifies the total number of bytes in the datagram
    - including both the header and the data

The IP Datagram Header Format (cont’d)

- ❑ IDENTIFICATION
  - 16-bit number (usually sequential) assigned to the datagram
    - used to gather all fragments for reassembly to the datagram
- ❑ FLAGS
  - 3-bit field with individual bits specifying whether the datagram is a fragment
    - If so, then whether the fragment corresponds to the rightmost piece of the original datagram
- ❑ FRAGMENT OFFSET
  - 13-bit field that specifies where in the original datagram the data in this fragment belongs
  - the value of the field is multiplied by 8 to obtain an offset

The IP Datagram Header Format (cont’d)

❑ TIME TO LIVE

- 8-bit integer initialized by the original sender
- it is decremented by each router that processes the datagram
- if the value reaches zero (0)
    - the datagram is discarded and an error message is sent back to the source

❑ TYPE

- 8-bit field that specifies the type of the payload

❑ HEADER CHECKSUM

- 16-bit ones-complement checksum of header fields

❑ SOURCE IP ADDRESS

- 32-bit Internet address of the original sender

The IP Datagram Header Format (cont’d)

❑ DESTINATION IP ADDRESS

- The 32-bit Internet address of the ultimate destination

❑ IP OPTIONS

- Optional header fields used to control routing and datagram processing
- Most datagrams do not contain any options

❑ PADDING

- If options do not end on a 32-bit boundary
    - zero bits of padding are added to make the header a multiple of 32 bits

IP Fragmentation

MTU and Datagram Fragmentation

□ Each hardware technology specifies the maximum amount of data that a frame can carry

  – The limit is known as a Maximum Transmission Unit (MTU)

□ Network hardware is not designed to accept or transfer frames that carry more data than the MTU allows

  – A datagram must be smaller or equal to the network MTU

    - or it cannot be encapsulated for transmission

□ In an internet that contains heterogeneous networks, MTU restrictions create a problem

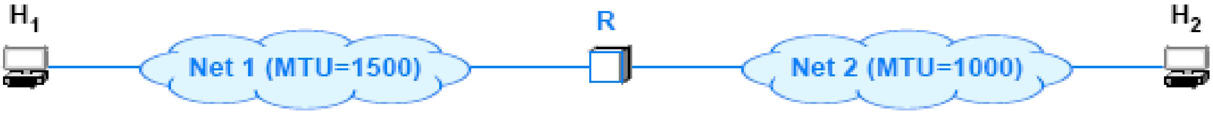
□ A router can connect networks with different MTU values

  – a datagram that a router receives over one network can be too large to send over another network

MTUs for some networks

Protocol	*MTU*
Hyperchannel	65,535
Token Ring (16 Mbps)	17,914
Token Ring (4 Mbps)	4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

MTU and Datagram Fragmentation



- ❑ Example: a router interconnects two networks with MTU values of 1500 and 1000
  - Host H_1 attaches to a network with an MTU of 1500
    - and can send a datagram that is up to 1500 octets
  - Host H_2 attaches to a network that has an MTU of 1000
    - which means that it cannot send/receive a datagram larger than 1000 octets
  - If host H_1 sends a 1500-octet datagram to host H_2
    - router R will not be able to encapsulate it for transmission across network 2

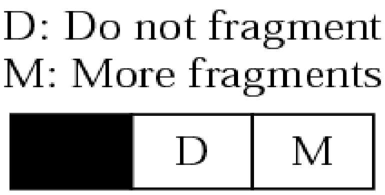
MTU and Datagram Fragmentation

- ❑ When a datagram is larger than the MTU of the network over which it must be sent
  - the router divides the datagram into smaller pieces called fragments
  - and sends each fragment independently
- ❑ A fragment has the same format as other datagrams
  - a bit in the FLAGS field of the header indicates whether a datagram is a fragment or a complete datagram
- ❑ Other fields in the header are assigned information for the ultimate destination to reassemble fragments
  - to reproduce the original datagram
- ❑ The FRAGMENT OFFSET specifies where in the original datagram the fragment belongs

MTU and Datagram Fragmentation

- ❑ A router uses the network MTU and the header size to calculate
  - the maximum amount of data that can be sent in each fragment
  - and the number of fragments that will be needed
- ❑ The router then creates the fragments
  - It uses fields from the original header to create a fragment header
    - For example, the router copies the IP SOURCE and IP DESTINATION fields from the datagram into the fragment header
  - It copies the appropriate data from the original datagram into the fragment
  - Then it transmits the result

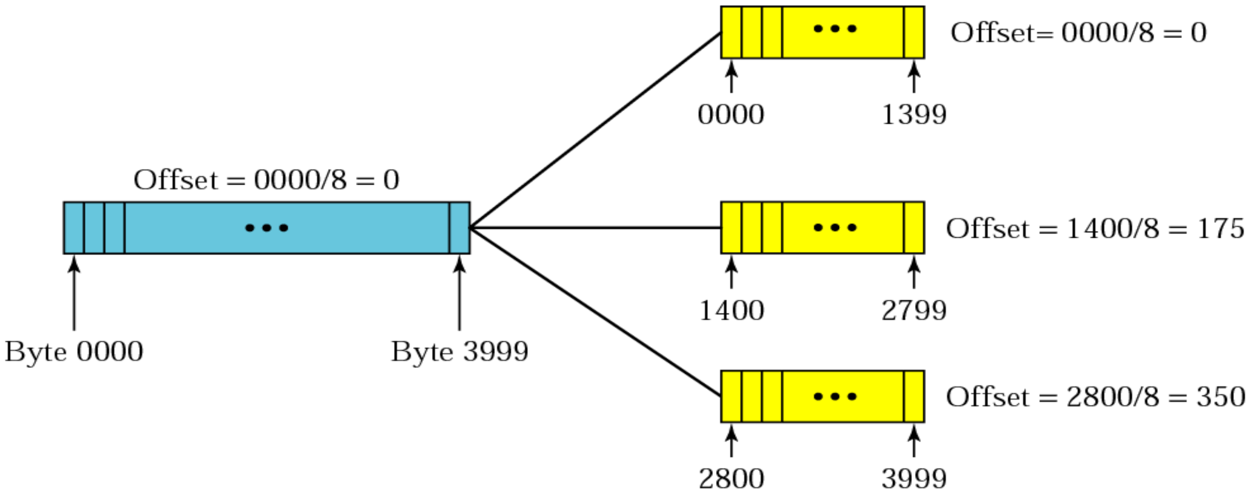
Flags field



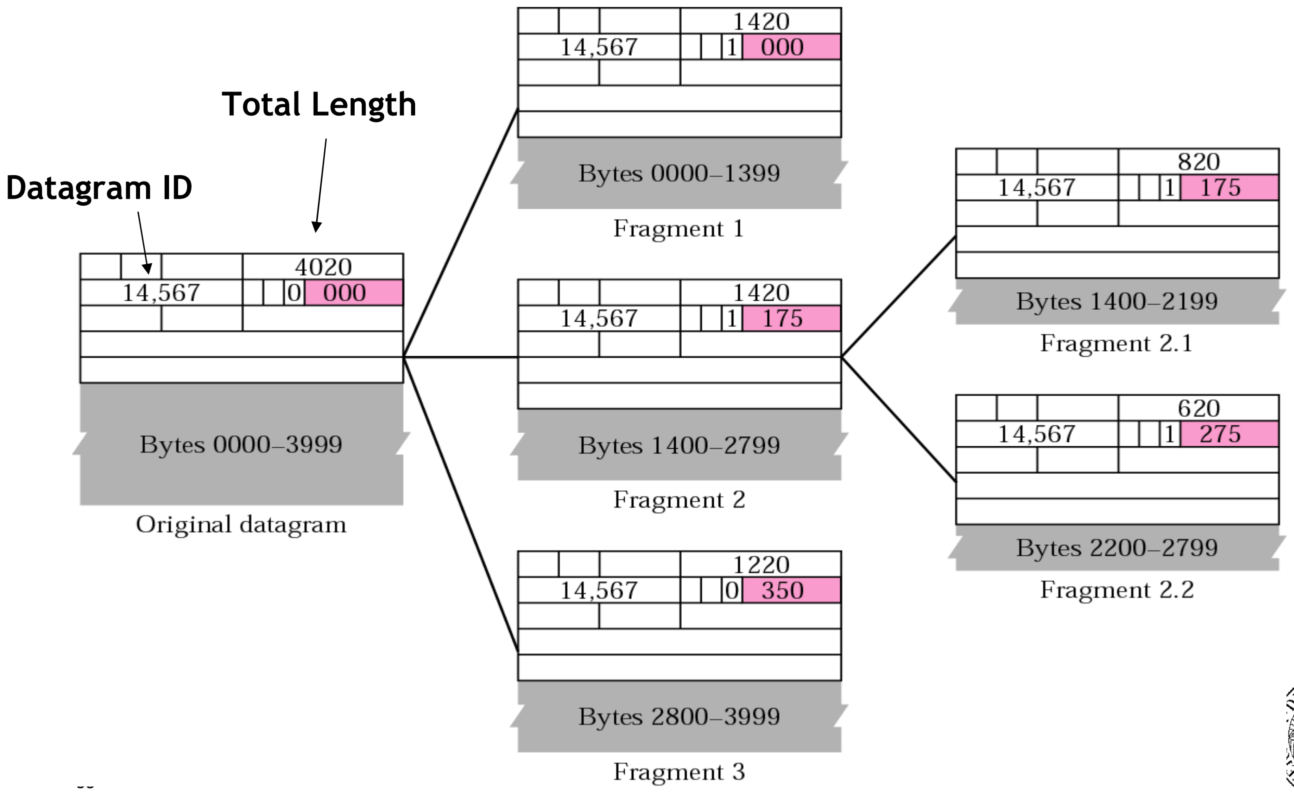
D: Do not fragment

M: More fragments

Fragmentation example



Fragmentation example



Questions

- A packet has arrived with an M bit value of 0. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?
  - If the M bit is 0, it means that there are no more fragments; the fragment is the last one. However, we cannot say if the original packet was fragmented or not. A nonfragmented packet is considered the last fragment

- A packet has arrived with an M bit value of 1. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?
  - If the M bit is 1, it means that there is at least one more fragment. This fragment can be the first one or a middle one, but not the last one. We don't know if it is the first one or a middle one; we need more information (the value of the fragmentation offset). See also the next example.

Questions

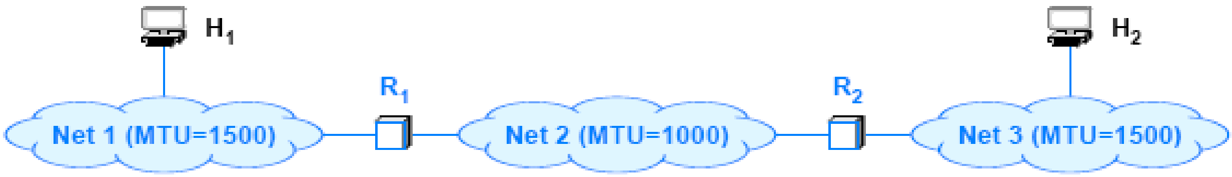
❑ A packet has arrived with an M bit value of 1 and a fragmentation offset value of zero. Is this the first fragment, the last fragment, or a middle fragment?.

- Because the M bit is 1, it is either the first fragment or a middle one. Because the offset value is 0, it is the first fragment

❑ A packet has arrived in which the offset value is 100. What is the number of the first byte? Do we know the number of the last byte?

- To find the number of the first byte, we multiply the offset value by 8. This means that the first byte number is 800. We cannot determine the number of the last byte unless we know the length of the data.

Reassembly of a Datagram from Fragments



❑ Example: packets sent from H1 to H2

- if host H1 sends a 1500-octet datagram to host H2, router R1 will divide the datagram into two fragments, which it will forward to R2
- Router R2 does not reassemble the fragments
  - Instead R uses the destination address in a fragment to forward the fragment as usual
- The ultimate destination host, H2, collects the fragments and reassembles them to produce the original datagram

Reassembly of a Datagram from Fragments

- Requiring the ultimate destination to reassemble fragments has two advantages:
  - It reduces the amount of state information in routers
    - When forwarding a datagram, a router does not need to know whether the datagram is a fragment
  - It allows routes to change dynamically
    - If an intermediate router were to reassemble fragments, all fragments would need to reach the router
- By postponing reassembly until the ultimate destination
  - IP is free to pass some fragments from a datagram along different routes than other fragments

The Consequence of Fragment Loss

- A datagram cannot be reassembled until all fragments arrive
- The receiver must save (buffer) the fragments
  - In case missing fragments are only delayed
  - A receiver cannot hold fragments an arbitrarily long time
- IP specifies a maximum time to hold fragments
- When the first fragment arrives from a given datagram
  - the receiver starts a reassembly timer
- If all fragments of a datagram arrive before the timer expires
  - the receiver cancels the timer and reassembles the datagram
- Otherwise the receiver discards the fragments

The Consequence of Fragment Loss

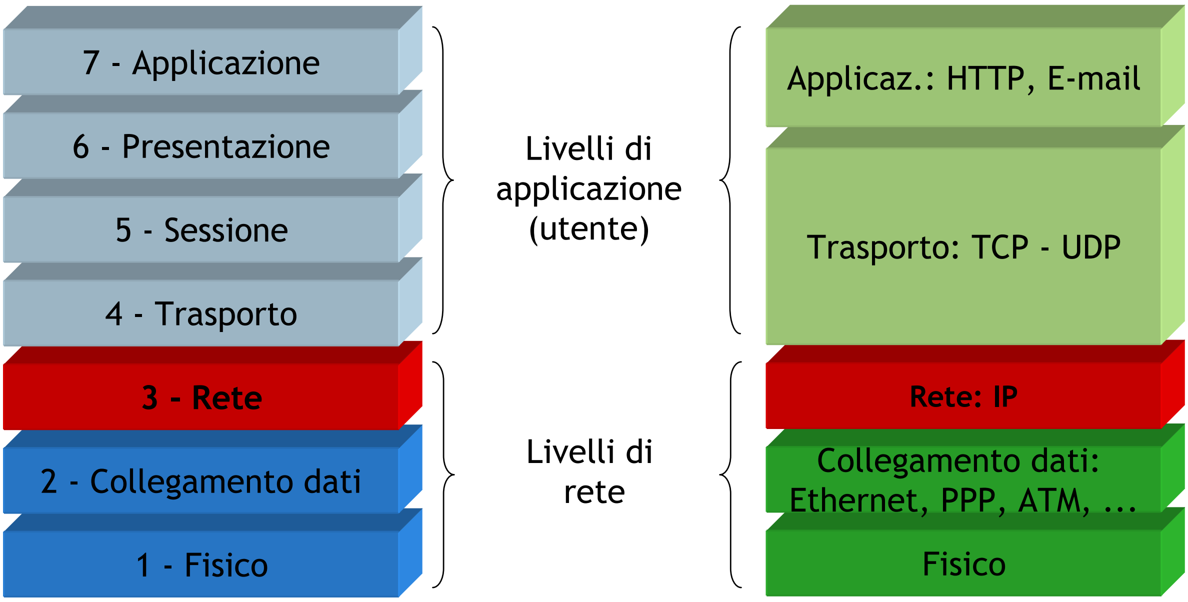
- ❑ The result of IP's reassembly timer is all-or-nothing:
  - either all fragments arrive and IP reassembles the datagram,
  - If not then IP discards the incomplete datagram
- ❑ There is no mechanism for a receiver to tell the sender which fragments have arrived
  - The sender does not know about fragmentation
- ❑ If a sender retransmits, the datagram routes may be different
  - a retransmission would not necessarily traverse the same routers
    - also, there is no guarantee that a retransmitted datagram would be fragmented in the same way as the original

Reti di Calcolatori

L'indirizzamento nel livello di rete

Universtità degli studi di Verona
Facoltà di Scienze MM.FF.NN.
A.A. 2009/2010
Laurea in Informatica
Docente: Damiano Carra

Network level



7 - Applicazione

6 - Presentazione

5 - Sessione

4 - Trasporto

3 - Rete

2 - Collegamento dati

1 - Fisico

Livelli di applicazione (utente)

Livelli di rete

Applicaz.: HTTP, E-mail

Trasporto: TCP - UDP

Rete: IP

Collegamento dati: Ethernet, PPP, ATM, ...

Fisico

Introduction

Overall view

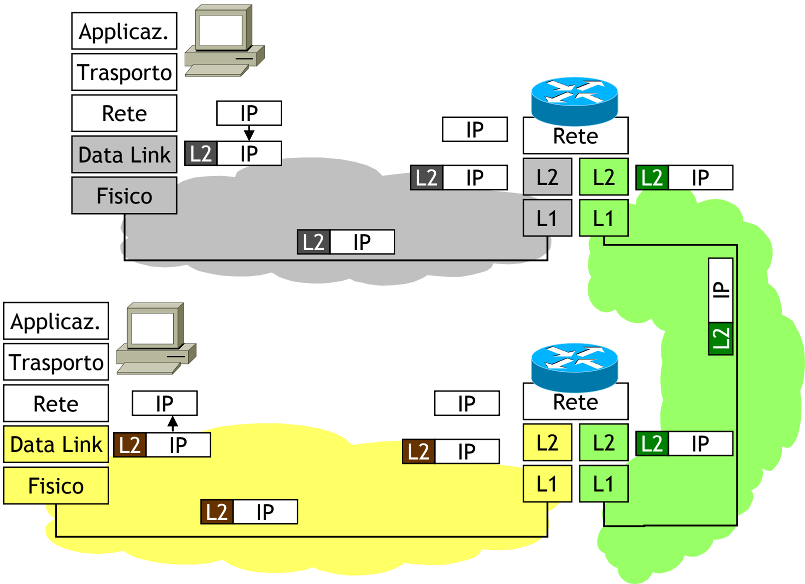
□ Transport segment from sending to receiving host

□ On sending side encapsulates segments into datagrams

□ On rcving side, delivers segments to transport layer

□ Network layer protocols in every host, router

□ Router examines header fields in all IP datagrams passing through it



Two Key Network-Layer Functions

□ Routing

- Determine route taken by packets from source to dest.

➔ Routing algorithms

- Analogy (trip planning): process of planning trip from source to dest

□ Forwarding

- Move packets from router’s input to appropriate router output
  - Analogy (trip planning): process of getting through single interchange

□ These function require a basic component: Addressing

Addresses for the Internet

- ❑ Addressing is a critical component of the Internet
- ❑ All host computers must use a uniform addressing scheme
  - an arbitrary pair of application programs can communicate without knowing the type of network hardware being used
- ❑ Each address must be unique
- ❑ MAC addresses do not suffice because
  - the Internet can include multiple network technologies
  - and each technology defines its own MAC addresses
- ❑ IP addresses are supplied by protocol software
  - They are not part of the underlying network

The IP Addressing Scheme

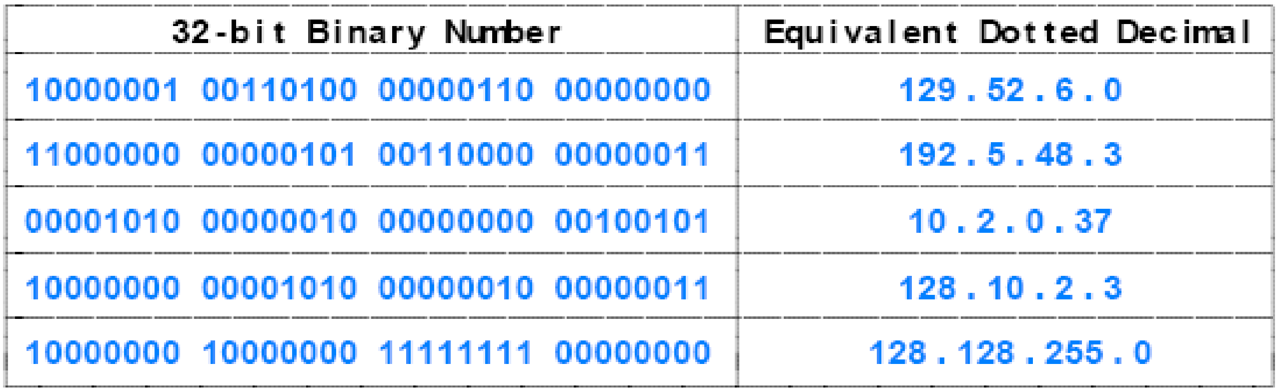
- ❑ Each host is assigned a unique 32-bit number
  - known as the host's IP address or Internet address
- ❑ When sending a packet across the Internet, sender's protocol software must specify
  - its own 32-bit IP address (the source address)
  - and the address of the intended recipient (the destination address)

Dotted Decimal Notation

- ❑ Instead of writing 32 bits, a notation more convenient for humans to understand is used
- ❑ Notation, known as dotted decimal notation, is
  - express each 8-bit section of a 32-bit number as a decimal value
  - use periods to separate the sections
- ❑ Dotted decimal treats each octet (byte) as an unsigned binary integer
  - the smallest value, 0
    - occurs when all bits of an octet are zero (0)
  - the largest value, 255
    - occurs when all bits of an octet are one (1)
  - dotted decimal addresses range

0.0.0.0 through 255.255.255.255

Dotted Decimal Notation: examples



32-bit Binary Number	Equivalent Dotted Decimal
10000001 00110100 00000110 00000000	129 . 52 . 6 . 0
11000000 00000101 00110000 00000011	192 . 5 . 48 . 3
00001010 00000010 00000000 00100101	10 . 2 . 0 . 37
10000000 00001010 00000010 00000011	128 . 10 . 2 . 3
10000000 10000000 11111111 00000000	128 . 128 . 255 . 0

The IP Address Hierarchy

- IP address is divided into two parts:
- A prefix ➔ identifies the physical network to which the host is attached (also known ad NetID)
  - Each network in the Internet is assigned a unique network number
- A suffix ➔ identifies a specific computer (host/node) on the network (also known ad HostID)
  - Each computer on a given network is assigned a unique suffix
- IP address scheme guarantees two properties:
  - Each computer is assigned a unique address
  - Network number (prefix) assignments must be coordinated globally
    - Suffixes are assigned locally without global coordination

Classful addressing

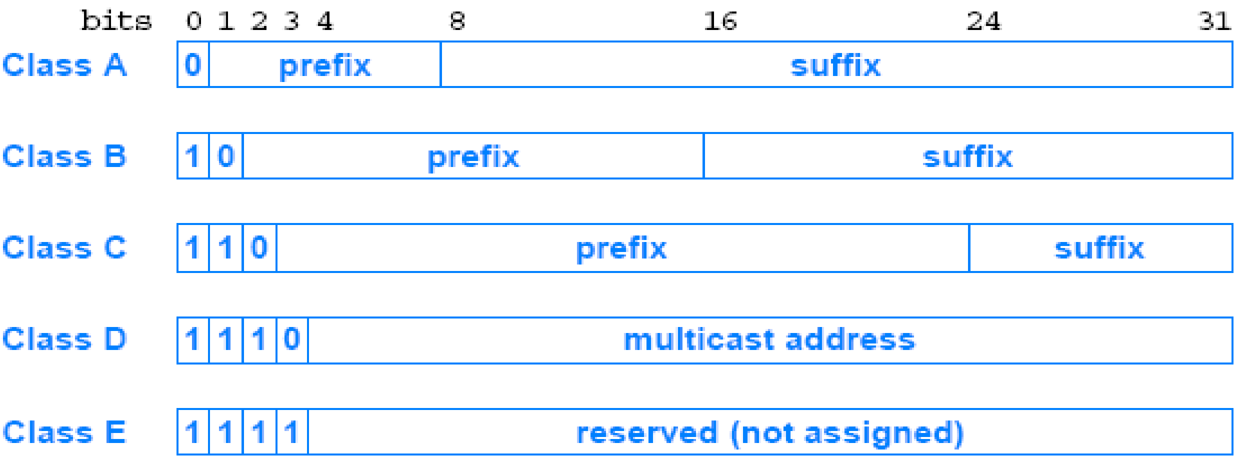
Original Classes of IP Addresses: bit tradeoff

- ❑ How many bits to place in each part of an IP address?
  - The prefix needs sufficient bits to allow a unique network number to be assigned to each physical network in the Internet
  - The suffix needs sufficient bits to permit each computer attached to a network to be assigned a unique suffix
- ❑ No simple choice was possible to allocate bits!
  - Choosing a large prefix accommodates many networks
    - but limits the size of each network
  - Choosing a large suffix means each physical network can contain many computers
    - but limits the total number of networks

Original Classes of IP Addresses

- ❑ Internet contains a few large physical networks and many small networks
  - the designers chose an addressing scheme to accommodate a combination of large and small networks
- ❑ The original classful IP addressing divided the IP address space into 3 primary classes
  - each class has a different size prefix and suffix
- ❑ The first four bits of an IP address determined the class to which the address belonged
  - It specifies how the remainder of the address was divided into prefix and suffix

Original Classes of IP Addresses



bits 0 1 2 3 4 8 16 24 31

Class A | 0 | prefix | suffix

Class B | 1 0 | prefix | suffix

Class C | 1 1 0 | prefix | suffix

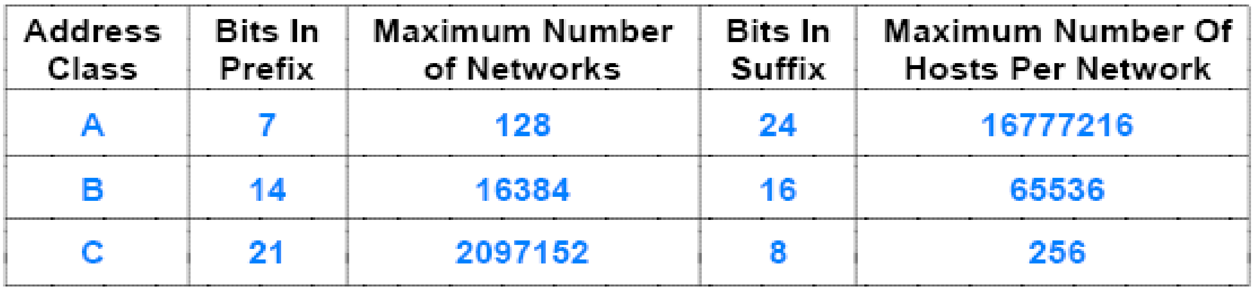
Class D | 1 1 1 0 | multicast address

Class E | 1 1 1 1 | reserved (not assigned)

Division of the Address Space

- ❑ The classful scheme divided the address space into unequal sizes
- ❑ The designers chose an unequal division to accommodate a variety of scenarios
  - For example, although it is limited to 128 networks, class A contains half of all addresses
    - The motivation was to allow major ISPs to each deploy a large network that connected millions of computers
  - Similarly, the motivation for class C was to allow an organization to have a few computers connected on a LAN

Division of the Address Space



Address Class	Bits In Prefix	Maximum Number of Networks	Bits In Suffix	Maximum Number Of Hosts Per Network
A	7	128	24	16777216
B	14	16384	16	65536
C	21	2097152	8	256

Authority for Addresses

- ❑ Internet Corporation for Assigned Names and Numbers (ICANN) authority has been established
  - to handle address assignment and adjudicate disputes
- ❑ ICANN does not assign individual prefixes
  - Instead, ICANN authorizes a set of registrars to do so
- ❑ Registrars make blocks of addresses available to ISPs
  - ISPs provide addresses to subscribers
- ❑ To obtain a prefix
  - a corporation usually contacts an ISP

Classless addressing

Subnet and Classless Addressing

- ❑ As the Internet grew the original classful addressing scheme became a limitation
- ❑ Everyone demanded a class A or class B address
  - So they would have enough addresses for future growth
    - but many addresses in class A and B were unused
- ❑ Two mechanisms, closely related, were designed to overcome the limitation
  - Subnet addressing
  - Classless addressing
- ❑ Instead of having three distinct address classes, allow the division between prefix/suffix on an arbitrary bit boundary

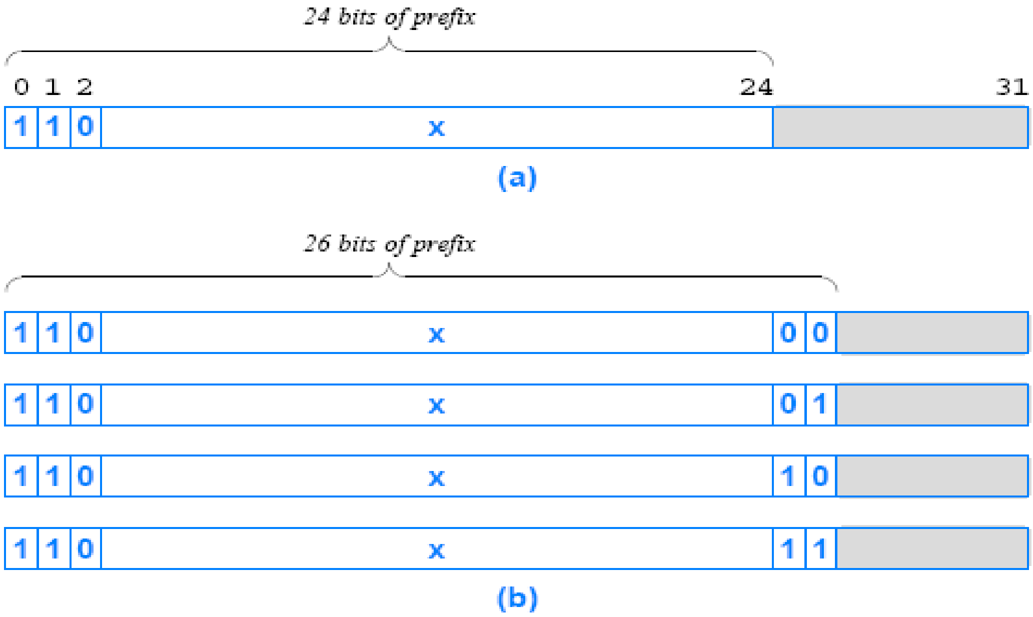
Subnet and Classless Addressing: Motivation

- ❑ Consider an ISP that hands out prefixes. Suppose a customer of the ISP requests a prefix for a network that contains 55 hosts
  - classful addressing requires a complete class C prefix
  - only 6 bits of suffix are needed to represent all possible host values
    - means 190 of the 254 possible suffixes would never be assigned
  - most of the class C address space is wasted
- ❑ For the above example
  - classless addressing allows the ISP to assign
    - a prefix that is 26 bits long
    - a suffix that is 6 bits long

Subnet and Classless Addressing: Example

- ❑ Assume an ISP owns a class C prefix
  - Classful addressing assigns the entire prefix to one organization
- ❑ With classless addressing
  - the ISP can divide the prefix into several longer prefixes
- ❑ For instance, the ISP can divide a class C prefix into 4 longer prefixes
  - each one can accommodate a network of up to 62 hosts
    - all 0s and all 1s are reserved
- ❑ The original class C address has 8 bits of suffix
  - and each of the classless addresses has 6 bits of suffix
- ❑ Thus, instead of wasting addresses
  - ISP can assign each of the 4 classless prefixes to a subscriber

Subnet and Classless Addressing: Example



Address Masks

- ❑ How can an IP address be divided at an arbitrary boundary?
- ❑ The classless and subnet addressing schemes require hosts and routers to store an additional piece of information:
  - a value that specifies the exact boundary between prefix and suffix
- ❑ To mark the boundary, IP uses a 32-bit value
  - known as an address mask, also called a subnet mask
- ❑ Why store the boundary size as a bit mask?
  - A mask makes processing efficient
- ❑ Hosts and routers need to compare the network prefix portion of the address to a value in their forwarding tables
  - The bit-mask representation makes the comparison efficient

Address Masks

- ❑ Suppose a router is given
  - a destination address, D
  - a network prefix represented as a 32-bit value, N
  - a 32-bit address mask, M
- ❑ Assume the top bits of N contain a network prefix, and the remaining bits have been set to zero
- ❑ To test whether the destination lies on the specified network, the router tests the condition:

N == (D & M)

- ❑ The router
  - uses the mask with a "logical and (&)" operation to set the host bits of address D to zero (0)
  - and then compares the result with the network prefix N

Address Masks: Example

❑ Consider the following 32-bit network prefix:

10000000 00001010 00000000 00000000 → 128.10.0.0

❑ Consider a 32-bit mask:

11111111 11111111 00000000 00000000 → 255.255.0.0

❑ Consider a 32-bit destination address, which has a

10000000 00001010 00000010 00000011 → 128.10.2.3

❑ A logical and between the destination address and the address mask extracts the high-order 16-bits

10000000 00001010 00000000 00000000 → 128.10.0.0

CIDR Notation

- Classless Inter-Domain Routing (CIDR)
- Consider the mask needed for the example in Slide 23
  - It has 26 bits of 1s followed by 6 bits of 0s
  - In dotted decimal, the mask is: 255.255.255.192
- The general form of CIDR notation is: ddd.ddd.ddd.ddd/m
  - ddd is the decimal value for an octet of the address
  - m is the number of one bits in the mask
- Thus, one might write the following: 192.5.48.69/26
  - which specifies a mask of 26 bits

A CIDR Example

- Assume an ISP has the following block 128.211.0.0/16
- Suppose the ISP has 2 customers
  - one customer needs 12 IP addresses and the other needs 9
- The ISP can assign
  - customer1 CIDR: 128.211.0.16/28
  - customer2 CIDR: 128.211.0.32/28
  - both customers have the same mask size (28 bits), the prefixes differ

A CIDR Example (cont'd)

- The binary value assigned to customer1 is:
  - 10000000 11010011 00000000 00010000
- The binary value assigned to customer2 is:
  - 10000000 11010011 00000000 00100000
- There is no ambiguity
  - Each customer has a unique prefix
  - More important, the ISP retains most of the original address block
    - it can then allocate to other customers

CIDR Host Addresses

- Once an ISP assigns a customer a CIDR prefix
  - the customer can assign host addresses for its network users
- Suppose an organization is assigned 128.211.0.16/28
  - the organization will have 4-bits to use as a host address field
- Disadvantage of classless addressing
  - Because the host suffix can start on an arbitrary boundary, values are not easy to read in dotted decimal

CIDR Host Addresses

Network Prefix 128.211.0.16 / 28

```
0                                                          28      31
1 0 0 0 0 0 0 0 1 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
```

Address Mask 255.255.255.240

```
0                                                          28      31
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 1 1 0 0 0 0
```

Lowest Host Address 128.211.0.17

```
0                                                          28      31
1 0 0 0 0 0 0 0 1 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
```

Highest Host Address 128.211.0.30

```
0                                                          28      31
1 0 0 0 0 0 0 0 1 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0
```

Special IP addresses

Special IP Addresses

- IP defines a set of special address forms that are reserved
  - That is, special addresses are never assigned to hosts
- Examples:
  - Network Address
  - Directed Broadcast Address
  - Limited Broadcast Address
  - This Computer Address
  - Loopback Address

Network Address

- It is convenient to have an address that can be used to denote the prefix assigned to a given network
- IP reserves host address zero
  - and uses it to denote a network
- Thus, the address 128.211.0.16/28 denotes a network
  - because the bits beyond the 28 are zero
  - 10000000 11010011 00000000 00010000
- A network address should never appear as the destination address in a packet

Directed Broadcast Address

- ❑ To simplify broadcasting (send to all)
  - IP defines a directed broadcast address for each physical network
- ❑ When a packet is sent to a network's directed broadcast
  - a single copy of the packet travels across the Internet
    - until it reaches the specified network
  - the packet is then delivered to all hosts on the network
- ❑ The directed broadcast address for a network is formed by adding a suffix that consists of all 1 bits to the network prefix
  - 10000000 11010011 00000000 00011111

Limited Broadcast Address

- ❑ Limited broadcast refers to a broadcast on a directly-connected network:
  - informally, we say that the broadcast is limited to a “single wire”
- ❑ Limited broadcast is used during system startup
  - by a computer that does not yet know the network number
- ❑ IP reserves the address consisting of 32-bits of 1s
  - 11111111 11111111 11111111 11111111
- ❑ Thus, IP will broadcast any packet sent to the all-1s address across the local network

This Computer Address

- ❑ A computer needs to know its IP address
  - before it can send or receive Internet packets
- ❑ TCP/IP contains protocols a computer can use to obtain its IP address automatically when the computer boots
  - … but the startup protocols also use an IP to communicate
- ❑ When using such startup protocols
  - a computer cannot supply a correct IP source address
  - To handle such cases IP reserves the address that consists of all 0s to mean this computer
  - 00000000 00000000 00000000 00000000

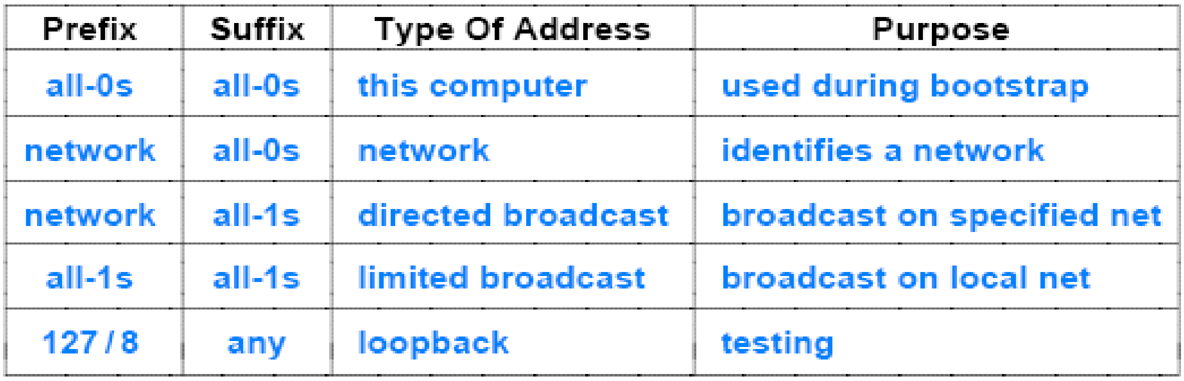
Loopback Address

- ❑ Loopback address used to test network applications
  - e.g., for preliminary debugging after a network application has been created
- ❑ A programmer must have two application programs that are intended to communicate across a network
- ❑ Instead of executing each program on a separate computer
  - the programmer runs both programs on a single computer
  - and instructs them to use a loopback address when communicating
- ❑ When one application sends data to another
  - data travels down the protocol stack to the IP software
  - then forwards it back up through the protocol stack to the second program

Loopback Address (cont’d)

- ❑ IP reserves the network prefix 127/8 for use with loopback
- ❑ The host address used with 127 is irrelevant
  - all host addresses are treated the same
  - programmers often use host number 1
  - so it makes 127.0.0.1 the most popular loopback address
- ❑ During loopback testing no packets ever leave a computer
  - the IP software forwards packets from one application to another
- ❑ The loopback address never appears in a packet traveling across a network

Summary of Special IP Addresses

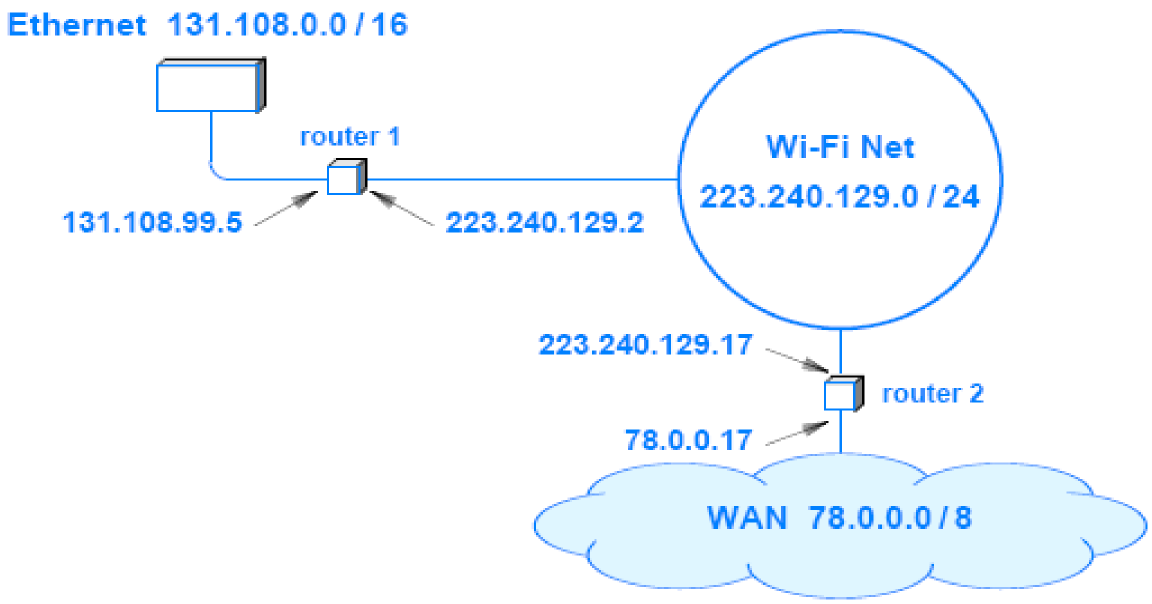


Prefix	Suffix	Type Of Address	Purpose
all-0s	all-0s	this computer	used during bootstrap
network	all-0s	network	identifies a network
network	all-1s	directed broadcast	broadcast on specified net
all-1s	all-1s	limited broadcast	broadcast on local net
127 / 8	any	loopback	testing

Routers and the IP Addressing Principle

- ❑ Each router is assigned two or more IP addresses
  - one address for each network to which the router attaches
- ❑ To understand why, recall two facts:
  - A router has connections to multiple physical networks
  - Each IP address contains a prefix that specifies a physical network
- ❑ A single IP address does not suffice for a router
  - because each router connects to multiple networks
  - and each network has a unique prefix
- ❑ The IP scheme can be explained by a principle:
  - An IP address does not identify a specific computer
  - each address identifies a connection between a computer and a network
  - A computer with multiple network connections (e.g., a router) must be assigned one IP address for each connection

Routers and the IP Addressing Principle



Verifica contenuti

Exercise 1

Change the following IP addresses from binary notation to dotted-decimal notation.

a. ***10000001 00001011 00001011 11101111***
b. ***11000001 10000011 00011011 11111111***
c. ***11100111 11011011 10001011 01101111***
d. ***11111001 10011011 11111011 00001111***

Solution
We replace each group of 8 bits with its equivalent decimal number (see Appendix B) and add dots for separation:

a. ***129.11.11.239*** *b.* ***193.131.27.255***
c. ***231.219.139.111*** *d.* ***249.155.251.15***

Exercise 2

Change the following IP addresses from dotted-decimal notation to binary notation.

a. 111.56.45.78 *b. 221.34.7.82*
c. 241.8.56.12 *d. 75.45.34.78*

Solution

We replace each decimal number with its binary equivalent:

a. 01101111 00111000 00101101 01001110
b. 11011101 00100010 00000111 01010010
c. 11110001 00001000 00111000 00001100
d. 01001011 00101101 00100010 01001110

Exercise 3

Find the error, if any, in the following IP addresses:

a. 111.56.045.78 *b. 221.34.7.8.20*

c. 75.45.301.14 *d. 11100010.23.14.67*

Solution

a. There are no leading zeroes in dotted-decimal notation (045).

b. We may not have more than four numbers in an IP address.

c. In dotted-decimal notation, each number is less than or equal to 255; 301 is outside this range.

d. A mixture of binary notation and dotted-decimal notation is not allowed.

Reti di Calcolatori

Esercizi su indirizzamento IP

Universtità degli studi di Verona
Facoltà di Scienze MM.FF.NN.
A.A. 2009/2010
Laurea in Informatica
Docente: Damiano Carra

Exercise 1

What is the first address in the block if one of the addresses is 167.199.170.82/27?

Solution

The prefix length is 27, which means that we must keep the first 27 bits as is and change the remaining bits (5) to 0s. The following shows the process:

Address in binary: 10100111 11000111 10101010 01010010
Keep the left 27 bits: 10100111 11000111 10101010 01000000
Result in CIDR notation: 167.199.170.64/27

Exercise 2

Find the number of addresses in the block if one of the addresses is 140.120.84.24/20.

Solution

The prefix length is 20. The number of addresses in the block is 2^{32-20} or 2^{12} or 4096. Note that this is a large block with 4096 addresses.

What is the first address in the block if one of the addresses is 140.120.84.24/20?

Solution

The first address is 140.120.80.0/20.

Exercise 3

Find the block if one of the addresses is 190.87.140.202/29.

Solution

The number of addresses is 2^{32-29} = 8. The first address is 190.87.140.200/29, the last address is 190.87.140.207/20

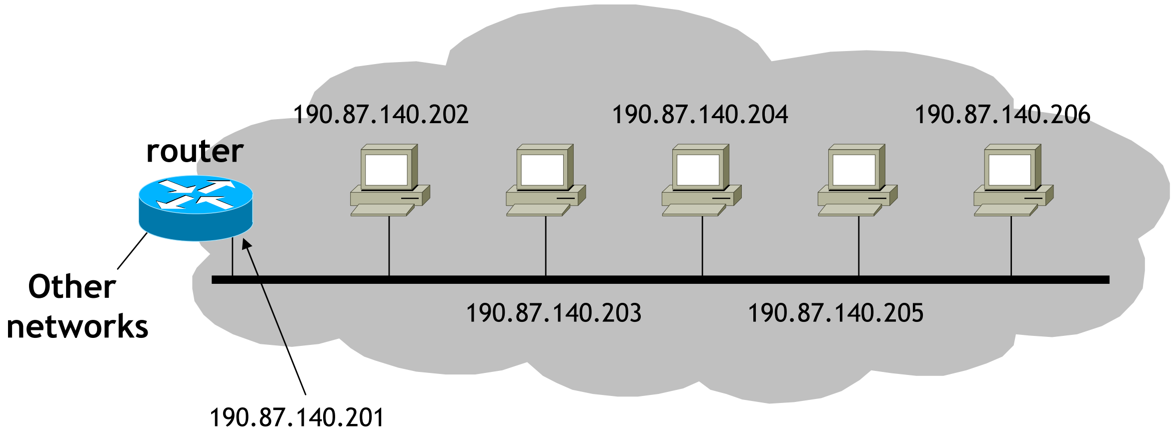
Show a network configuration for the block for this example.

Solution

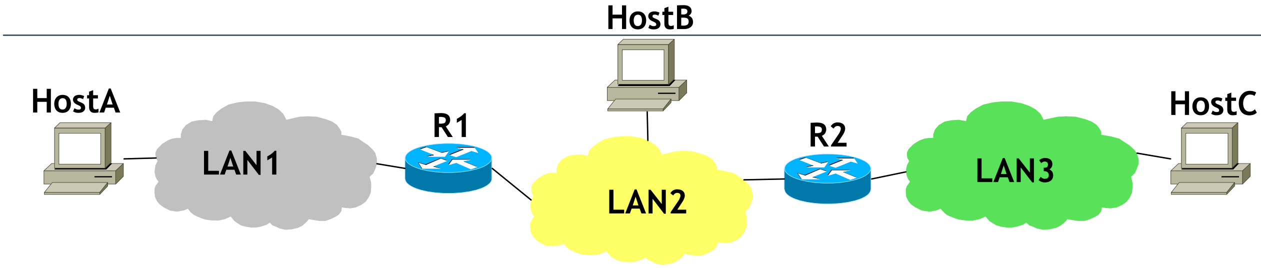
The first address needs to be used as the network address and the last address is kept as a special address (limited broadcast address). See next slide

Exercise 3 (cont'd)

Special addresses
Network address → 190.87.140.200/20
Broadcast address → 190.87.140.207

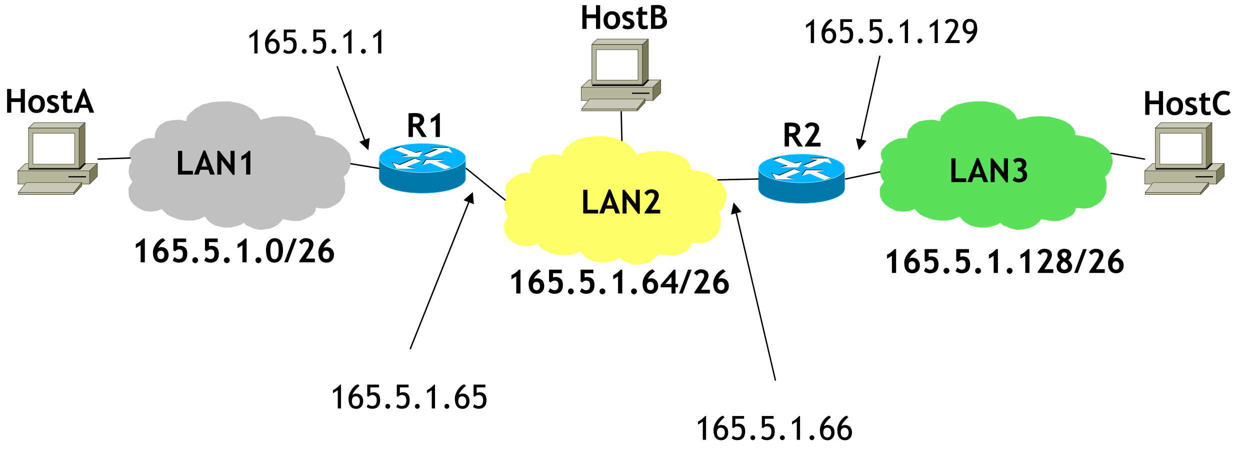


Exercise 4

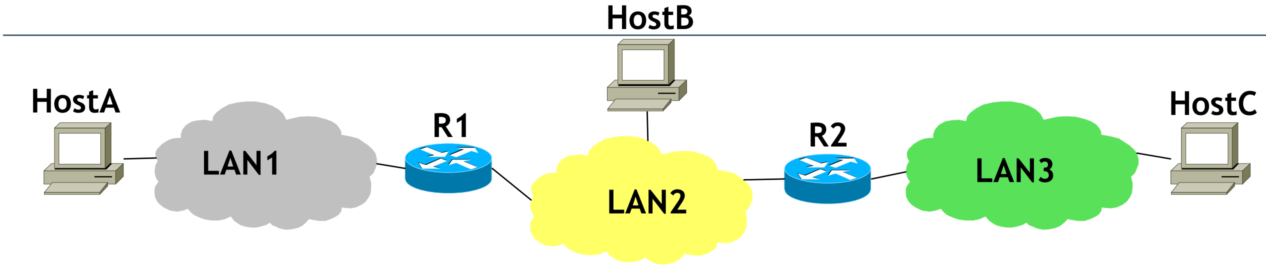


- ❑ Al sito rappresentato e' stato assegnato il blocco CIDR 16.5.1.0/24
  - si assegnino gli indirizzi di rete alle tre LAN partizionando il blocco in modo che ciascuna LAN possa contenere lo stesso numero di stazioni (massimizzando tale numero)
  - per ogni LAN si specifichi l'indirizzo di broadcast spiegando come si ottiene
  - si assegni il primo (o i primi) indirizzo(i) disponibile(i) per ogni LAN all'interfaccia del router ad essa collegato

Exercise 4 - Solution

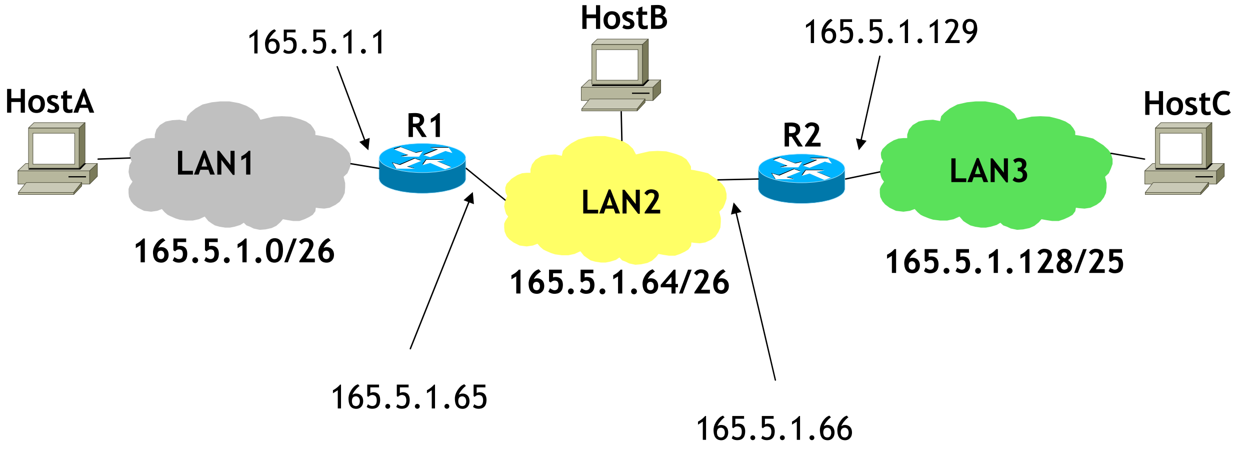


Exercise 5

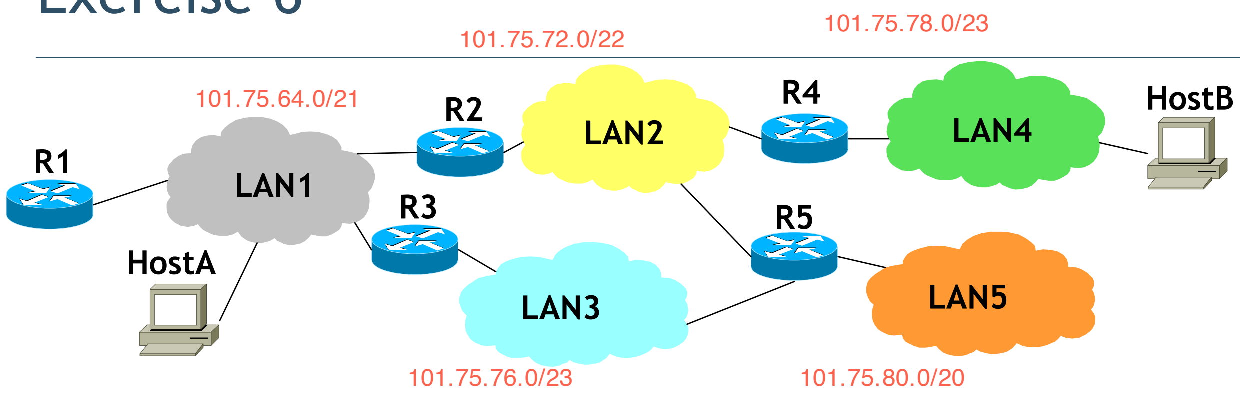


- ❑ Al sito rappresentato e' stato assegnato il blocco CIDR 16.5.1.0/24
  - si assegnino gli indirizzi di rete alle tre LAN partizionando il blocco in modo che ciascuna LAN1 e LAN2 possano contenere almeno 32 stazioni ciascuna, mentre LAN3 ne possa contenere almeno 64
  - per ogni LAN si specifichi l'indirizzo di broadcast spiegando come si ottiene
  - si assegni il primo (o i primi) indirizzo(i) disponibile(i) per ogni LAN all'interfaccia del router ad essa collegato

Exercise 5 - Solution



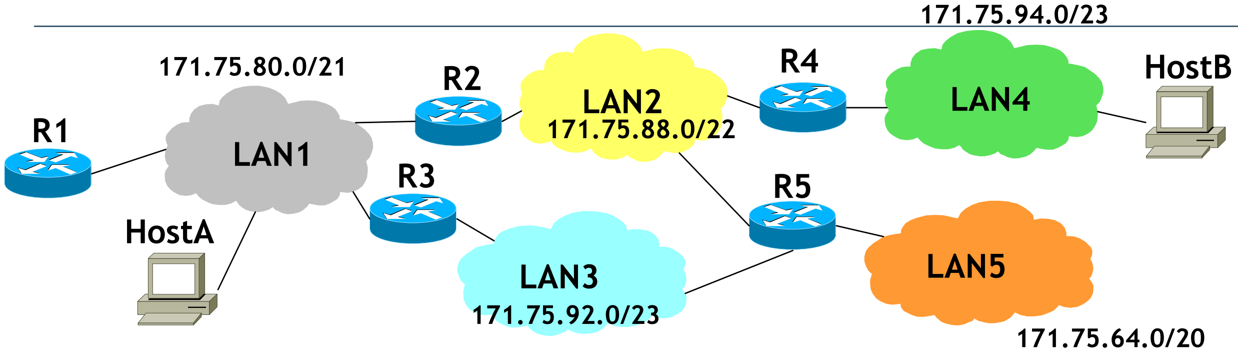
Exercise 6



□ Con riferimento alla figura:

- Si scriva il blocco CIDR a dimensione minima contenente gli indirizzi 101.75.79.255 e 101.75.80.0.
- Si utilizzi il blocco calcolato al punto precedente per assegnare il piano di indirizzamento alle reti LAN1/2/3/4/5 rispettando i seguenti vincoli:
  - LAN 1 ha netmask /21,
  - LAN 2 deve ospitare 1000 host, netmask= /22
  - LAN 3 ha netmask /23,
  - LAN 4 deve ospitare 400 host, netmask= /23
  - LAN 5 ha a disposizione metà dell'intero blocco di indirizzi. netmask= /20

Exercise 6 - Solution



❑ 101.75.79.255 ➔	01100101	01001011	01001111	11111111
❑ 101.75.80.0 ➔	01100101	01001011	01010000	00000000
❑ Parte comune ➔	01100101	01001011	01000000	00000000

- Che corrisponde a 101.75.64.0/19

❑ LAN1 → /21

❑ LAN2 → 1000 host, servono almeno 10 bit per l'HostID → /22

❑ LAN3 → /23

❑ LAN4 → 400 host, servono almeno 9 bit per l'HostID → /23

❑ LAN5 → /20