Computer Networks - *Xarxes de Computadors*

Outline

- Course Syllabus
- Unit 1: Introduction
- **Unit 2. IP Networks**
- Unit 3. LANs
- Unit 4. TCP
- Unit 5. Network applications

Based on: https://studies.ac.upc.edu/FIB/grau/XC/#slides

Unit 2: IP Networks

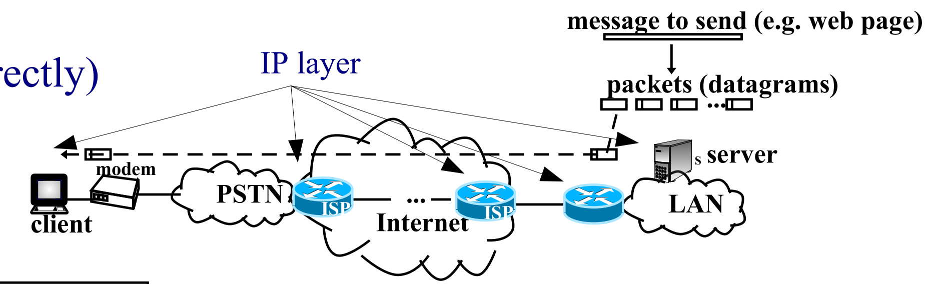
Outline

- **IP layer service**
- IP addresses
- Subnetting
- Routing tables
- ARP protocol
- IP header
- ICMP protocol
- DHCP protocol
- NAT
- Routing algorithms
- Security in IP

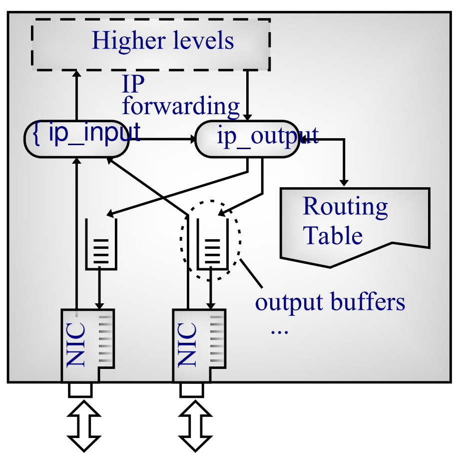
Unit 2: IP Networks

IP Layer Service

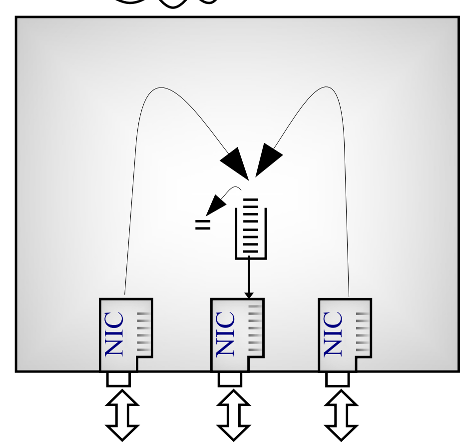
- Internet Protocol (IP)
  - Goal: routing packets from host to host
  - Design principle: interconnect hosts attached to LANs/WANs networks of different technologies
  - Characteristics
    - Connectionless (send/receive directly)
    - Stateless (no memory)
    - Best effort (no guarantee)



Commercial routers
(edge routers)



Basic router architecture

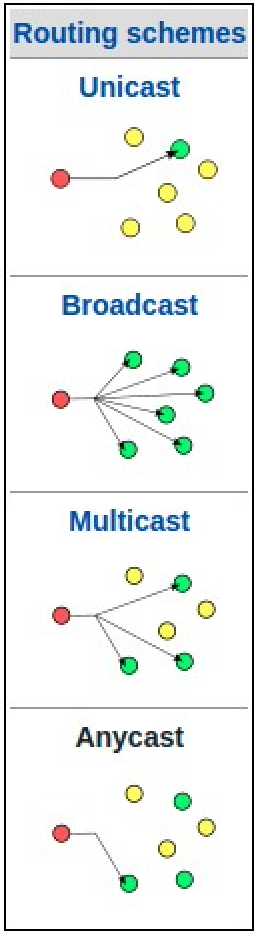


Looses may occur due to buffer overflow

Unit 2: IP Networks

IP Layer Service (cont.)

- Design implications
  - Packets can be delivered out-of-order
  - Each packet can take a different path to the destination
  - No error detection (=> no correction) in payload
    - Left to higher layers, if any
  - No congestion control (beyond "drop")
- Routing schemes: destinations
  - Unicast: one-to-one (multi-hop)
    - Ex. application layer: classic client-server
  - Broadcast: one-to-all (broadcast domain – data link layer)
    - Ex. link+IP layer: ARP request (upcoming slides)
  - Multicast: one-to-many
    - IP layer: Requires multicast routing: not supported by Internet
  - Anycast:
    - Application layer: DNS root nameservers clusters



https://en.wikipedia.org/wiki/Unicast

Unit 2: IP Networks

High Performance Routers

(core routers)

Source: Juniper

"There is a major upgrade going on in service providers upgrading their core networks," Chris Komatas, director of service provider marketing at Juniper, said.

"The next-generation core network is all about having the agility to support any service. T1600 is delivering No. 1 in scale, No. 1 in service control and No. 1 in efficiency. All the metrics that are important for a service provider."

The keys to the performance throughput on the Juniper T1600 are the 100Gbps-capable slots that can support all the major connectivity options that carriers may have. Among them is support for OC-768 (40 Gbps), OC-192 (10Gbps) and 10GbE (10 Gigabit Ethernet).

Juniper (www.juniper.net)

Figure 1. Cisco XR 12000 and 12000 series routing portfolio

Table 1. Product Specifications

Product Specification	Cisco XR 12000 and 12000 Series 16-Slot Chassis	Cisco XR 12000 and 12000 Series 10-Slot Chassis	Cisco XR 12000 and 12000 Series 6-Slot Chassis	Cisco XR 12000 and 12000 Series 4-Slot Chassis
Slot capacity	16 slots	10 slots	6 slots	4 slots
Aggregate switching capacity	Cisco 12016: 80 Gbps Cisco 12416: 320 Gbps Cisco 12816: 1280 Gbps	Cisco 12010: 50 Gbps Cisco 12410: 200 Gbps Cisco12810: 800 Gbps	Cisco 12006: 30 Gbps Cisco 12406: 120 Gbps	Cisco 12404: 80 Gbps
Full-duplex throughput per slot	Cisco 12016: 2.5 Gbps/slot Cisco 12416: 10 Gbps/slot Cisco 12816: 40 Gbps/slot	Cisco 12010: 2.5 Gbps/slot Cisco 12410: 10 Gbps/slot Cisco 12810: 40 Gbps/slot	Cisco 12006: 2.5 Gbps/slot Cisco 12406: 10 Gbps/slot	Cisco 12404: 10 Gbps/slot

cisco (www.cisco.com)

Unit 2: IP Networks

Outline

- IP layer service
- **IP addresses**
- Subnetting
- Routing tables
- ARP protocol
- IP header
- ICMP protocol
- DHCP protocol
- NAT
- Routing algorithms
- Security in IP

Unit 2: IP Networks

IPv4 Addresses (RFC 791)

```
          0               1               2               3      bytes
+---------------+---------------+---------------+---------------+
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1bits
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          netid                |            hostid             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- 32 bits (4 bytes)
- Dotted point notation: Four bytes in decimal, e.g. 147.83.24.28
- netid identifies the network
- hostid identifies the host within the network.
- An IP address identifies an *interface*: an attachment point to the network
- All IP addresses in Internet must be different. To achieve this goal, Internet Assigned Numbers Authority, IANA (http://www.iana.net) assign address blocs to Regional Internet Registries, RIR:
  - RIPE: Europe, http://www.ripe.net
  - ARIN: USA, http://www.arin.net
  - APNIC: ASIA http://www.apnic.net
  - LACNIC: Latin America, http://www.lacnic.net
  - AFRINIC: Afica, http://www.afrinic.net
- RIR assign addresses to ISPs, and ISPs to their customers

Unit 2: IP Networks

IP Addresses – Classes (obsolete)

- The highest bits identify the class
- The number of IP bits of netid/hostid varies in classes A/B/C
- D Class is for multicast addresses (e.g. 224.0.0.2: “all routers”)
- E Class are reserved addresses

Class	Netid [bytes]	Hostid [bytes]	Number of subnets	Netmask (slide 13)	Leading bits	Range
A	1	3	$2^7 = 128$	/8	0	0.0.0.0 – 127.255.255.255
B	2	2	$2^{14} = 16,384$	/16	10	128.0.0.0 – 191.255.255.255
C	3	1	$2^{21} = 2,097,152$	/24	110	192.0.0.0 – 223.255.255.255
D	-	-	-	-	1110	224.0.0.0 – 239.255.255.255
E	-	-	-	-	1111	240.0.0.0 – 255.255.255.255

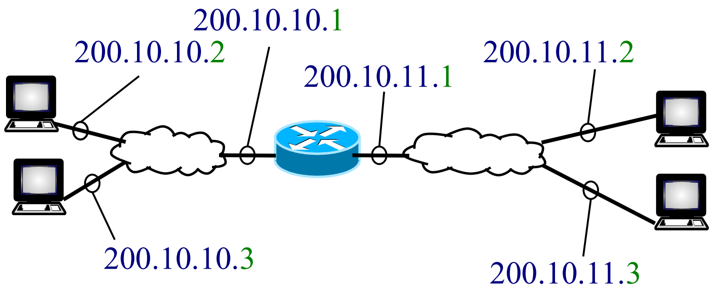
Unit 2: IP Networks

IP Addresses – Special Addresses

- Special addresses cannot be used for a physical interface
- Each network has two special addresses: network and broadcast addresses

netid	hostid	Meaning
xxx	all ‘0’	Identifies a network. It is used in routing tables.
xxx	all ‘1’	Broadcast in the net. xxx.
all ‘0’	all ’0’	Identifies “this host” in “this net.”. Used as source address in configuration protocols, e.g. DHCP.
all ‘1’	all ‘1’	broadcast in “this net.”. Used as destination address in configuration protocols, e.g. DHCP.
127	xxx	host loopback: interprocess communication with TCP/IP.

- Example:

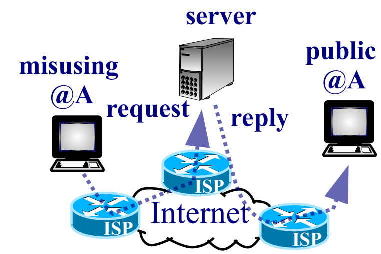


Unit 2: IP Networks

IP Addresses – Private Addresses (RFC 1918)

- Most commercial OSs include the TCP/IP stack.
- TCP/IP is used to network many kind of electronic devices:

- Addresses assigned to RIRs by IANA are called *public, global* or *registered*.
- What if we arbitrarily assign a registered address to a host?
  - It may be filtered by our ISP or cause trouble to the right host using that address.



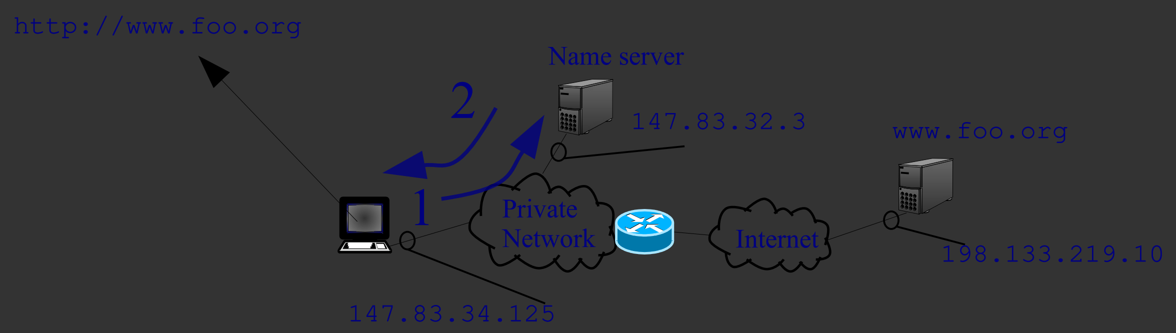
- **Private addresses** has been reserved for devices not using public addresses. These addresses are not assigned to any RIR (are not unique). There are addresses in each class:
  - 1 class A network: 10.0.0.0
  - 16 class B networks: 172.16.0.0 ~ 172.31.0.0
  - 256 class C networks: 192.168.0.0 ~ 192.168.255.0

Note: 169.254.0.0 – 169.254.255.255 is also a private range (used for IP autoconfiguration)

Unit 2: IP Networks

DNS – Protocol (EXPLAINED IN DETAIL IN UNIT 5)

- Client-server paradigm
- Short messages uses UDP.
- well-known port: 53



1 DNS Request

```
18:36:00.322370 IP (proto: UDP) 147.83.34.125.1333 >
    147.83.32.3.53:  53040+ A? www.foo.org. (31)
```

2 DNS Reply

```
18:36:00.323080 IP (proto: UDP) 147.83.32.3.53 > 147.83.34.125.1333:
    53040 1/2/2 www.foo.org. A 198.133.219.10 (115)
```

Unit 2: IP Networks

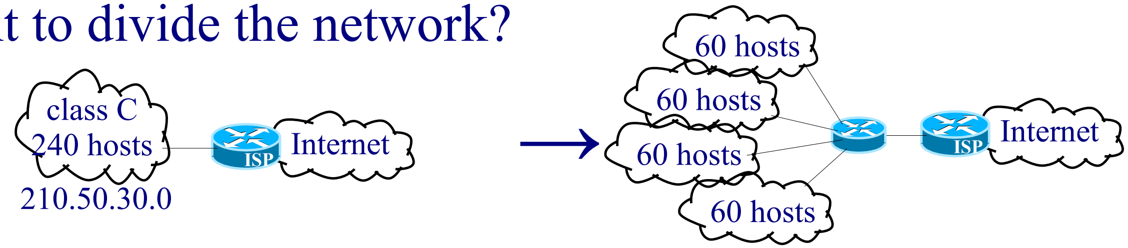
Outline

- IP layer service
- IP addresses
- **Subnetting**
- Routing tables
- ARP protocol
- IP header
- ICMP protocol
- DHCP protocol
- NAT
- Routing algorithms
- Security in IP

Unit 2: IP Networks

Subnetting (RFC 950)

- Initially the netid was given by the address class: A with 2^{24} addresses, B with 2^{16} addresses and C with 2^{8} addresses.
- What if we want to divide the network?



- Subnetting allows adding bits from the hostid to the netid (called subnetid bits).
- Example: For the ISP the network prefix is 24 bits. For the internal router the network prefix is 26 bits. The 2 extra bits allows 4 "subnetworks".
- A **mask** (*netmask*) is used to identify the size of the netid+subnetid prefix.
- Mask notations:
  - dot notation, as 255.255.255.192
  - slash notation, giving the mask length (number of bits) as 210.50.30.0/26 (=> mask length: 26 bits)

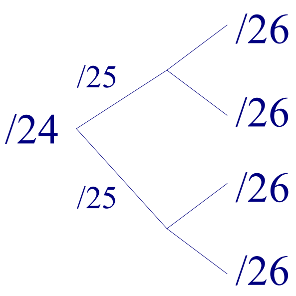
Unit 2: IP Networks

IP Addresses – Subnetting Example

- We want to subnet the address 210.50.30.0/24 in 4 subnets

Base address B = 210.50.30

subnet	subnetid	IP net. addr.	range	broadcast	available
S1	00	B.0/26	B.0 ~ B.63	B.63	$2^6 - 2 = 62$
S2	01	B.64/26	B.64 ~ B.127	B.127	$2^6 - 2 = 62$
S3	10	B.128/26	B.128 ~ B.191	B.191	$2^6 - 2 = 62$
S4	11	B.192/26	B.192 ~ B.255	B.255	$2^6 - 2 = 62$

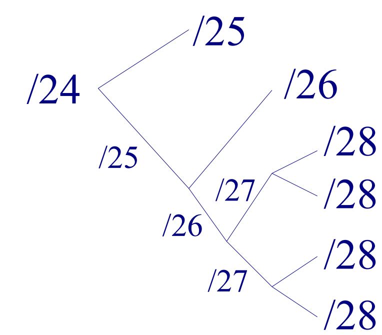


Unit 2: IP Networks

IP Addresses – Variable Length Subnet Mask (VLSM)

- Subnetworks of different sizes.
- Example, subnetting a class C address:
  - We have 1 byte for subnetid + hostid.
  - subnetid is green, chosen subnets addresses are underlined.

<u>0000</u>
1000 → <u>1000</u>
1100 → <u>1100</u>, <u>1101</u>, <u>1110</u>, <u>1111</u>



Base address B = 192.168.x, x ∈ {0, 255}

subnet	subnetid	IP net. addr.	range	broadcast	available
S1	0	B.0/25	B.0 ∼ B.127	B.127	$2^7 - 2 = 126$
S2	10	B.128/26	B.128 ∼ B.191	B.191	$2^6 - 2 = 62$
S3	1100	B.192/28	B.192 ∼ B.207	B.207	$2^4 - 2 = 14$
S4	1101	B.208/28	B.208 ∼ B.223	B.223	$2^4 - 2 = 14$
S5	1110	B.224/28	B.224 ∼ B.239	B.239	$2^4 - 2 = 14$
S6	1111	B.240/28	B.240 ∼ B.255	B.255	$2^4 - 2 = 14$

Unit 2: IP Networks

IP Addresses – Classless Inter-Domain Routing, CIDR (RFC 1519)

- Initially, Internet backbone routing tables did not use masks: netid was derived from the IP address class.
- When the number of networks in Internet started growing exponentially, routing tables size started exploding.
- In order to reduce routing tables size, CIDR proposed a “rational” geographical-based distribution of IP addresses to be able to “aggegate routes”, and use masks instead of classes.
- Aggregation example:

  200.1.10.0/24
  200.1.11.0/24 $\longrightarrow$ 200.1.10.0/23

- The term summarization is normally used when aggregation is done at a class boundary (e.g. a groups of subnets is summarized with their classful base address).
- NOTE: Aggregation cannot be done arbitrarily, otherwise the whole routing table could be aggregated in the default route 0.0.0.0/0. E.g. in BGP are specified which ranges can be aggregated; in RIP the term used is summarization (see Unit 2, Routing algorithms).

Unit 2: IP Networks

Example

- We have been assigned the 192.169.1.0/24 (public) address block
- We want to split it into two equal subnetworks
- Secondly, we want to split in the same manner the two subnetworks
- Challenge: for each of the three cases find the network address, the network mask, the total number of IPs available for hosts, the lowest and the highest host IP and the broadcast IP

1) Whole block (/24)

```
user@dac:~$ ipcalc 192.169.1.0/24
Address:   192.169.1.0          11000000.10101001.00000001. 00000000
Netmask:   255.255.255.0 = 24   11111111.11111111.11111111. 00000000
Wildcard:  0.0.0.255            00000000.00000000.00000000. 11111111
=>
Network:   192.169.1.0/24       11000000.10101001.00000001. 00000000
HostMin:   192.169.1.1          11000000.10101001.00000001. 00000001
HostMax:   192.169.1.254        11000000.10101001.00000001. 11111110
Broadcast: 192.169.1.255        11000000.10101001.00000001. 11111111
Hosts/Net: 254                   Class C
```

Unit 2: IP Networks

Example (cont.)

2) 2 subnets (two /25)

```
user@dac:~$ ipcalc 192.169.1.0/25
Address:   192.169.1.0          11000000.10101001.00000001.0 0000000
Netmask:   255.255.255.128 = 25 11111111.11111111.11111111.1 0000000
Wildcard:  0.0.0.127            00000000.00000000.00000000.0 1111111
=>
Network:   192.169.1.0/25       11000000.10101001.00000001.0 0000000
HostMin:   192.169.1.1          11000000.10101001.00000001.0 0000001
HostMax:   192.169.1.126        11000000.10101001.00000001.0 1111110
Broadcast: 192.169.1.127        11000000.10101001.00000001.0 1111111
Hosts/Net: 126                   Class C

user@dac:~$ ipcalc 192.169.1.128/25
Address:   192.169.1.128        11000000.10101001.00000001.1 0000000
Netmask:   255.255.255.128 = 25 11111111.11111111.11111111.1 0000000
Wildcard:  0.0.0.127            00000000.00000000.00000000.0 1111111
=>
Network:   192.169.1.128/25     11000000.10101001.00000001.1 0000000
HostMin:   192.169.1.129        11000000.10101001.00000001.1 0000001
HostMax:   192.169.1.254        11000000.10101001.00000001.1 1111110
Broadcast: 192.169.1.255        11000000.10101001.00000001.1 1111111
Hosts/Net: 126                   Class C
```

Example (cont.)

3) 4 subnets (four /26)

```
user@dac:~$ ipcalc 192.169.1.0/26
Address:   192.169.1.0          11000000.10101001.00000001.00 000000
Netmask:   255.255.255.192 = 26 11111111.11111111.11111111.11 000000
Wildcard:  0.0.0.63             00000000.00000000.00000000.00 111111
=>
Network:   192.169.1.0/25       11000000.10101001.00000001.00 000000
HostMin:   192.169.1.1          11000000.10101001.00000001.00 000001
HostMax:   192.169.1.62         11000000.10101001.00000001.00 111110
Broadcast: 192.169.1.63         11000000.10101001.00000001.00 111111
Hosts/Net: 62                    Class C

user@dac:~$ ipcalc 192.169.1.64/26
Address:   192.169.1.64         11000000.10101001.00000001.01 000000
Netmask:   255.255.255.192 = 26 11111111.11111111.11111111.11 000000
Wildcard:  0.0.0.63             00000000.00000000.00000000.00 111111
=>
Network:   192.169.1.64/25      11000000.10101001.00000001.01 000000
HostMin:   192.169.1.65         11000000.10101001.00000001.01 000001
HostMax:   192.169.1.126        11000000.10101001.00000001.01 111110
Broadcast: 192.169.1.127        11000000.10101001.00000001.01 111111
Hosts/Net: 62                    Class C

user@dac:~$ ipcalc 192.169.1.128/26
Address:   192.169.1.128        11000000.10101001.00000001.10 000000
Netmask:   255.255.255.192 = 26 11111111.11111111.11111111.11 000000
Wildcard:  0.0.0.63             00000000.00000000.00000000.10 111111
=>
Network:   192.169.1.128/25     11000000.10101001.00000001.10 000000
HostMin:   192.169.1.129        11000000.10101001.00000001.10 000001
HostMax:   192.169.1.190        11000000.10101001.00000001.10 111110
Broadcast: 192.169.1.191        11000000.10101001.00000001.10 111111
Hosts/Net: 62                    Class C

user@dac:~$ ipcalc 192.169.1.192/26
Address:   192.169.1.192        11000000.10101001.00000001.11 000000
Netmask:   255.255.255.192 = 26 11111111.11111111.11111111.11 000000
Wildcard:  0.0.0.63             00000000.00000000.00000000.00 111111
=>
Network:   192.169.192/25       11000000.10101001.00000001.11 000000
HostMin:   192.169.1.193        11000000.10101001.00000001.11 000001
HostMax:   192.169.1.254        11000000.10101001.00000001.11 111110
Broadcast: 192.169.1.255        11000000.10101001.00000001.11 111111
Hosts/Net: 62                    Class C
```

Unit 2: IP Networks

Networks aggregation

- Refer to several networks with a single larger network
- "The opposite" of subnetting; sometimes referred as *supernetting*
- Used in routing to reduce the number of networks to deal with
- Classless aggregation: according to CIDR
- Classful aggregation: according to class types A, B, etc. (e.g. 10.0.1.0/24 and 10.0.2.0/24 aggregates to 11.0.0.0/6)
- Note: called "summarization" in RIP
- To find the smallest classless aggregation of a given set of networks:
    1) Convert IPs in binary format
    2) The matching bits from left to right are the netid bits.

Example:

```
10.56.248.0/24        00001010.00111000.11111000.00000000
10.56.249.0/25        00001010.00111000.11111001.00000000
10.56.249.128/26      00001010.00111000.11111001.10000000
10.56.249.192/26      00001010.00111000.11111001.11000000
10.56.250.0/23        00001010.00111000.11111010.00000000
```

The CIDR network aggregation is 0001010.00111000.11111000.00000000 / 22 aka 10.56.248.0/22

Unit 2: IP Networks

Exercici resolt: 2021t-c1-sol.pdf

5. Marcar tots els blocs d'adreces següents que inclouen l'adreça 171.15.66.234

- ☐ 128.0.0.0/2
- ☐ 171.15.0.0/16
- ☐ 171.15.0.0/17
- ☐ 171.15.0.0/18
- ☐ 171.15.66.0/28
- ☐ 171.15.64.0/18
- ☐ 171.15.66.224/27
- ☐ 171.15.66.234/32

Bloc	Octet div.	Adreces octet	Host min	Host max	Inclòs?
128.0.0.0/2	1	/2 → 64	128.0.0.1	191.255.255.254	Sí
171.15.0.0/16	2|3	/0 → 256	171.15.0.1	171.15.255.254	Sí
171.15.0.0/17	3	/1 → 128	171.15.0.1	171.15.127.254	Sí
171.15.0.0/18	3	/2 → 64	171.15.0.1	171.15.63.254	No
171.15.66.0/28	4	/4 → 16	171.15.66.1	171.15.66.14	No
171.15.64.0/18	3	/2 → 64	171.15.64.1	171.15.127.254	Sí
171.15.66.224/27	4	/3 → 32	171.15.66.225	171.15.66.254	Sí
171.15.66.234**/32**	4	**/8 → 1**	171.15.66.234	171.15.66.234/32	Sí

Unit 2: IP Networks

Exercicis proposats

- Exàmens:
  - 2021t-c1
    - Problema 1, apartats a) i b)
- Col·lecció problemes:
  - Problema 1, apartats a) i b)
  - Problema 3, apartat a)
  - Problema 5, apartats a)
  - Problema 6, apartats a)
  - Problema 7, apartats a)
  - Problema 8, apartats a)

Unit 2: IP Networks

Outline

- IP layer service
- IP addresses
- Subnetting
- **Routing tables**
- ARP protocol
- IP header
- ICMP protocol
- DHCP protocol
- NAT
- Routing algorithms
- Security in IP

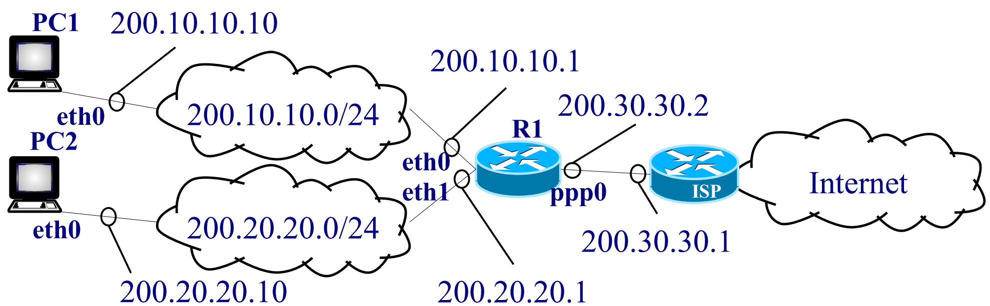
Unit 2: IP Networks

Routing Table

- ip_output() kernel function queries the routing table for each datagram.
- Routing can be:
  - Direct: The destination is directly connected to an interface.
  - Indirect: Otherwise. In this case, the datagram is sent to a router.
- Default route: Is an entry where to send all datagrams with a destination address to a network not present in the routing table. The default route address is 0.0.0.0/0.
- Hosts routing tables usually have two entries: The network where they are connected and a default route.

Unit 2: IP Networks

Routing Table – Unix Example



Genmask == Netmask

known destinations:
network & mask

how to reach the destinations:
(local) interface & gateway (next hop IP)

PC1 routing table:

Destination	Genmask	Gateway	Iface
200.10.10.0	255.255.255.0	0.0.0.0	eth0
0.0.0.0	0.0.0.0	200.10.10.1	eth0

PC2 routing table:

Destination	Genmask	Gateway	Iface
200.20.20.0	255.255.255.0	0.0.0.0	eth0
0.0.0.0	0.0.0.0	200.20.20.1	eth0

R1 routing table:

Destination	Genmask	Gateway	Iface
200.10.10.0	255.255.255.0	0.0.0.0	eth0
200.20.20.0	255.255.255.0	0.0.0.0	eth1
200.30.30.1	255.255.255.255	0.0.0.0	ppp0
0.0.0.0	0.0.0.0	200.30.30.1	ppp0

Unit 2: IP Networks

Routing Table – Tiscali ISP, CISCO 7200 Router

- Telnet to route-server.ip.tiscali.net (see http://www.bgp4.net server list)

```
+---------------------------------------------------------------------+
|           TISCALI International Network - Route Monitor             |
|                              (AS3257)                               |
|   This system is solely for internet operational purposes. Any      |
|   misuse is strictly prohibited. All connections to this router     |
|   are logged.                                                       |
|   This server provides a view on the TISCALI routing table that     |
|   is used in Frankfurt/Germany. If you are interested in other      |
|   regions of the backbone check out http://www.ip.tiscali.net/lg    |
|   Please report problems to noc@tiscali.net                         |
+---------------------------------------------------------------------+
route-server.ip.tiscali.net> show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 213.200.64.93 to network 0.0.0.0
B       85.27.76.0/22 [20/10] via 213.200.64.93, 4w2d
B       85.196.154.0/24 [20/10] via 213.200.64.93, 1d09h
B       85.158.216.0/21 [20/10] via 213.200.64.93, 2w6d
B       85.193.136.0/22 [20/10] via 213.200.64.93, 3d08h
B       85.121.48.0/21 [20/0] via 213.200.64.93, 1w4d
B       85.187.201.0/24 [20/10] via 213.200.64.93, 4d19h
B       85.114.0.0/20 [20/10] via 213.200.64.93, 1w5d
B       85.119.16.0/24 [20/10] via 213.200.64.93, 4w0d
B       85.119.16.0/21 [20/10] via 213.200.64.93, 4w0d
B       85.105.0.0/17 [20/10] via 213.200.64.93, 4w2d
B       85.93.52.0/24 [20/10] via 213.200.64.93, 4w0d
...
```

Tiscali Network Map
http://www.tiscali.net

thousands of entries

Unit 2: IP Networks

Routing Table – Datagram Delivery Algorithm

- Given a datagram with destination IP address, *D*:

1. Check if the device itself is the destination:

```
if(D == address of any of the interfaces) {
     send the datagram to upper layers
}
```

2. Query the routing table:

```
for each routing table entry ordered from longest to shortest mask (Longest
  Prefix Match) {
     if(D in Destination table entry) {
          return (gateway, interface) ;
     }
```

Prefix == netid
(Suffix == hostid)

3. Forward the datagram:

(see next slide)

Unit 2: IP Networks

Routing Table – Datagram Delivery Algorithm (cont.)

3. Forward the datagram:

```
if(D in a prefix of a directly connected network address) {
     /* it is a direct routing */
     deliver datagram to D over that network link ;
} elseif (the routing table contains a route for D) {
     /* it is an indirect routing */
     send datagram to the next-hop address listed in the routing table ;
} elseif (a default route exists) {
     /* default routing */
     send the datagram to the default route ;
} else {
     send forwarding error message to the originator ;
}
```

Using ICMP. The error message sent to the originator notifies that the packet could not be delivered. The sending host should either stop transmitting or choose another address or route.

Unit 2: IP Networks

Routing Table – Another Unix Example

- Basic topology: Laptop connected to an access point (AP)

```
user@dac:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG    600    0        0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U     600    0        0 eht0
```

U – Route up
G – Gateway
H – Host

- Laptop – AP with 2 other connections

```
user@dac:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG    600    0        0 eth0
10.0.0.0        192.168.1.2     255.0.0.0       UG    0      0        0 eth0
10.0.0.2        192.168.1.3     255.255.255.255 UGH   0      0        0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U     600    0        0 eth0
```

Default route
Network route
Host route
Local network

Warning: In GNU/Linux `route` is deprecated; `ip route show` (from the iproute2 package) must be used instead

Warning: routes with a longer prefix always take priority so in this case lower routes have priority (e.g. 10.0.0.2/32 over 10.0.0.0/8); however, routes shown by `route` may no respect this criterion; `ip route show` respects it.

Unit 2: IP Networks

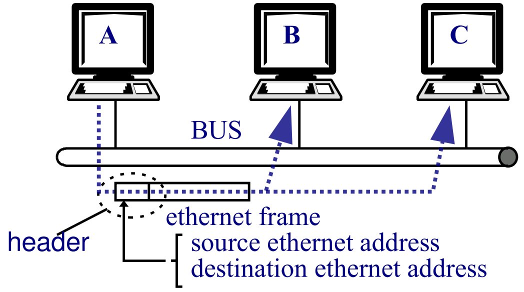
Outline

- IP layer service
- IP addresses
- Subnetting
- Routing tables
- **ARP protocol**
- IP header
- ICMP protocol
- DHCP protocol
- NAT
- Routing algorithms
- Security in IP

Unit 2: IP Networks

Address Resolution Protocol, ARP (RFC 826)

- To send the datagram, IP layer may have to pass a "physical address" to the NIC driver. Physical addresses are also called MAC or hardware addresses.
- ARP translate IP addresses to "physical addresses" (used by the physical network).
- If needed, IP calls ARP module to obtain the "physical addresses" before the NIC driver call.
- Ethernet example:



Ethernet bus deployments are not used any more.

WiFi is a more up-to-date example: in a WiFi network all nodes receive all packages because they share physical layer => the MAC addresses are very meaningful. .

Note: In IPv6, the protocol that links IP to MAC addresses is the Neighbor Discovery (ND) Protocol.

Unit 2: IP Networks

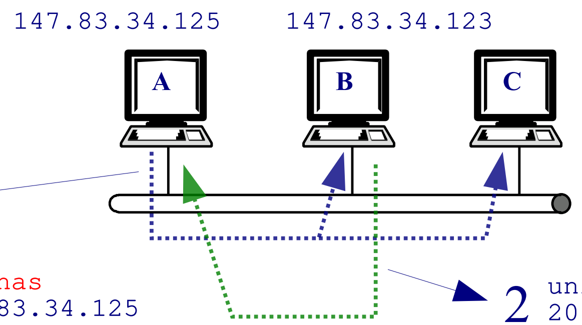
Address Resolution Protocol, messages

- When IP calls ARP:
  - If ARP table has the requested address, it is returned,
  - otherwise:
    - IP stores the datagram in a temporal buffer, and a resolution protocol is triggered.
    - IP initiates a timeout and starts forwarding the next datagram in the transmission queue.
    - If the timeout triggers before resolution, the datagram is removed.
    - If ARP returns the requested address, IP calls the driver with it.
- ARP resolution in an ethernet network (broadcast network):
  - A broadcast "ARP Request" message is sent indicating the IP address.
  - The station having the requested IP address sends a unicast "ARP Reply", and stores the requesting address in the ARP table.
  - Upon receiving the "ARP Reply", the requesting station return the IP call with it.
  - ARP entries have a timeout refreshed each time a match occurs.

Unit 2: IP Networks

Address Resolution Protocol, messages - Example

- ARP messages (tcpdump):



1 broadcast:
```
20:02:25.681331 arp who-has
 147.83.34.123 tell 147.83.34.125
```

2 unicast:
```
20:02:25.681490 arp reply 147.83.34.123
is-at 00:c0:49:d5:96:d8
```

- ARP tables:

```
A> /sbin/arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
147.83.34.123            ether   00:c0:49:d5:96:d8   C                     eth0

B> /sbin/arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
147.83.34.125            ether   00:14:F1:CC:59:00   C                     eth0
```

"Completed" flag

Warning: In GNU/Linux `arp` is deprecated; `ip link show` (from the iproute2 package) must be used instead (use 'ip l -s' to get statistics)

- Another example:

Time	No.	Source	Destination	Protocol	Length	Info	Pk len	Hardware src	Hardware dst
0.000000000	1	02:42:ac:1c:00:03	ff:ff:ff:ff:ff:ff	ARP	42	Who has 172.28.0.2? Tell 172.28.0.3	42	02:42:ac:1c:00:03	ff:ff:ff:ff:ff:ff
0.000148889	2	02:42:ac:1c:00:02	02:42:ac:1c:00:03	ARP	42	172.28.0.2 is at 02:42:ac:1c:00:02	42	02:42:ac:1c:00:02	02:42:ac:1c:00:03
0.000203120	3	172.28.0.3	172.28.0.2	ICMP	98	Echo (ping) request id=0x0014, seq=1/256, ttl=64 (reply in 4)	98	02:42:ac:1c:00:03	02:42:ac:1c:00:02
0.000304023	4	172.28.0.2	172.28.0.3	ICMP	98	Echo (ping) reply id=0x0014, seq=1/256, ttl=64 (request in 3)	98	02:42:ac:1c:00:02	02:42:ac:1c:00:03
5.159069240	5	02:42:ac:1c:00:02	02:42:ac:1c:00:03	ARP	42	Who has 172.28.0.3? Tell 172.28.0.2	42	02:42:ac:1c:00:02	02:42:ac:1c:00:03
5.159182181	6	02:42:ac:1c:00:03	02:42:ac:1c:00:02	ARP	42	172.28.0.3 is at 02:42:ac:1c:00:03	42	02:42:ac:1c:00:03	02:42:ac:1c:00:02
2315.53197…	7	fe80::42:66ff:fe2a:…	ff02::fb	MDNS	210	Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR _p…	210	02:42:66:2a:02:ba	33:33:00:00:00:fb
2810.94390	8	fe80::42:66ff:fe2a:	ff02::fb	MDNS	95	Standard query 0x0000 PTR _ipp._tcp.local, "QM" question	95	02:42:66:2a:02:ba	33:33:00:00:00:fb

Unit 2: IP Networks

Address Resolution Protocol – Message format (ethernet)

- Is a link layer protocol
- ARP messages are encapsulated directly in a data-link frame

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 bits
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Hardware Type (16)            | Protocol Type (16)            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Hard. Length(8)|Prot. Length(8)| Opcode (16)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Sender Hardware                                          |
+      Address (48)             +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               | Sender Protocol Address (32)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Sender Protocol Address (cont)|    Target Hardware            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    Address (48)               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Target Protocol Address (32)                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Unit 2: IP Networks

Address Resolution Protocol – Message format (ethernet) (cont.)

- HTYPE (hardware type) – [2B] Specifies the network link protocol type
  - `0x0001` for Ethernet

- PTYPE (protocol type) – [2B] Specifies the internetwork protocol for which the ARP request is intended. Same numbering as EtherType (see Ethernet II header, Unit 3)
  - `0x0800` for IPv4

- HLEN (hardware length) - [1B] Specifies the hardware address length in octets
  - 6 for Ethernet (MAC address)

- PLEN (protocol length) - [1B] Specifies the internetwork address length in octets
  - 4 for IPv4

- OPER (operation) - [2B] Specifies the operation that the sender is performing
  - 1 for request
  - 2 for reply

- SHA (sender hardware address) [6B for Ethernet] Media address of the sender
  - In a request indicates the address of the host sending the request
  - In a reply indicates the address of the host that the request was looking for

- SPA (sender protocol address) [4B for IPv4] Internetwork address of the sender

- THA (Target hardware address) Media address of the intended receiver
  - In a request is ignored
  - In a reply indicates the address of the host that originated the ARP request

- TPA (Target protocol) [4B for IPv4] Internetwork address of the intended receiver

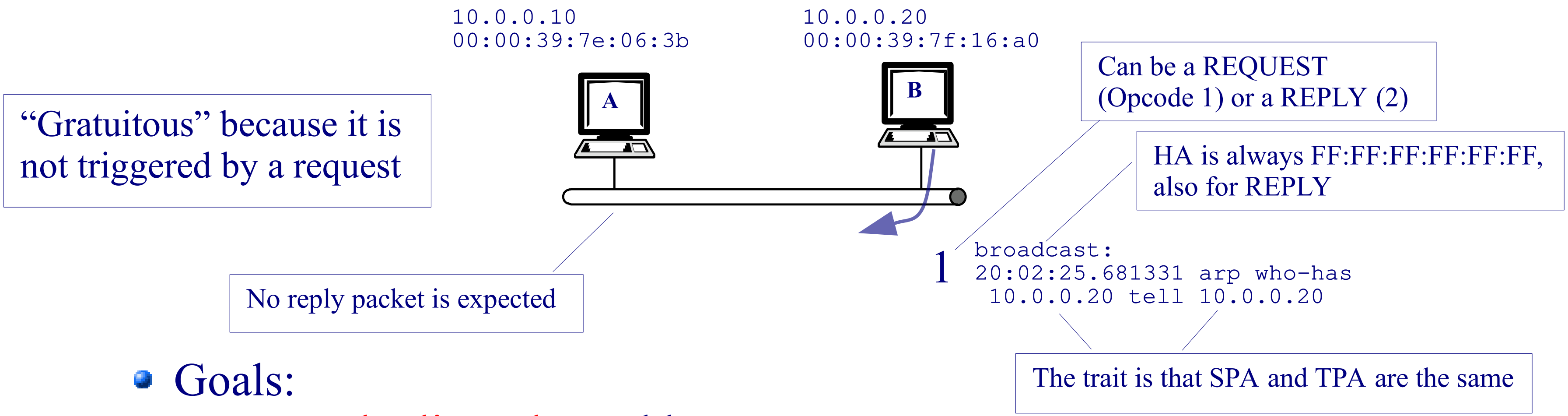
Internet Protocol (IPv4) over Ethernet ARP packet

Octet offset	0	1
0	Hardware type (HTYPE)	
2	Protocol type (PTYPE)	
4	Hardware address length (HLEN)	Protocol address length (PLEN)
6	Operation (OPER)	
8	Sender hardware address (SHA) (first 2 bytes)	
10	(next 2 bytes)	
12	(last 2 bytes)	
14	Sender protocol address (SPA) (first 2 bytes)	
16	(last 2 bytes)	
18	Target hardware address (THA) (first 2 bytes)	
20	(next 2 bytes)	
22	(last 2 bytes)	
24	Target protocol address (TPA) (first 2 bytes)	
26	(last 2 bytes)	

https://en.wikipedia.org/wiki/IPv4#Header

Unit 2: IP Networks

Address Resolution Protocol – Gratuitous ARP



- Goals:
  - Detect duplicated IP addresses.
  - Update MAC addresses in ARP tables after an IP or NIC change.

Recommendation: See https://wiki.wireshark.org/Gratuitous_ARP for further information and the example with ICMP in slide ~49

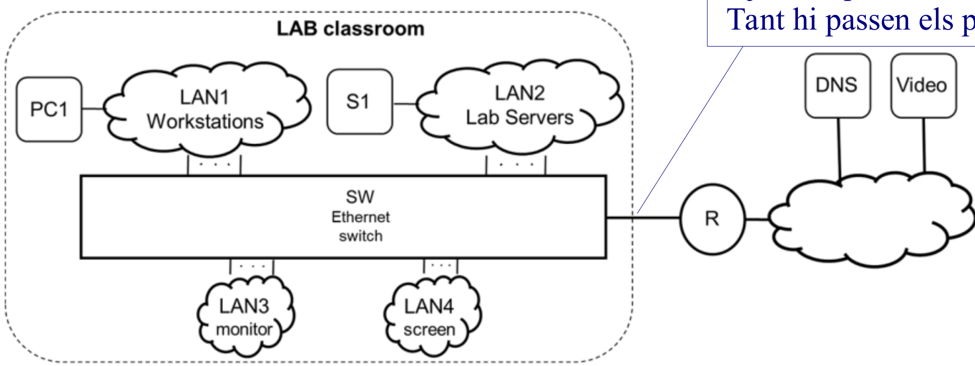
Exercici: 2021t-ef

Unit 2: IP Networks

Problema 1 (3,5 punts)

La figura mostra la configuració d'una aula de laboratori on hi ha llocs de treball (LAN1), servidors per donar suport als treballs dels laboratoris (LAN2), un PC de monitorització pel professor (LAN3) i una pantalla IP per a vídeo (LAN4). Cada laboratori disposa d'un commutador Ethernet (SW) on es configuren les 4 xarxes locals virtuals (VLAN) i l'adreçament proposat per a cada aula és 192.168.aula.0/24. El router R dona servei a més de 40 laboratoris amb la mateixa configuració.

Ajuda: aquest és l'enllaç que hem d'analitzar. Tant hi passen els paquets PC1-R com R-S1

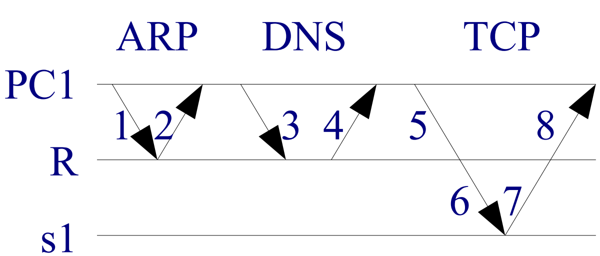


Ajuda: una connexió TCP comença amb el client enviat un SYN, al qual el servidor respon amb un SYN/ACK ...

c) La configuració que obté PC1 és: 192.168.1.2; router per defecte (gw): 192.168.1.1; DNS: 147.83.3.3. El PC1 inicia una connexió TCP amb el servidor s1-aula.fib.upc.edu. Completa la seqüència de trames i datagrames que passen per l'enllaç entre el commutador Ethernet i el router fins que PC1 rep el SYN/ACK. El router R ja té la informació a la taula ARP de tots els servidors.

Notació: majúscules per les adreces IP i minúscules per les adreces Ethernet (MAC). Exemple: PC1, pc1.

Ajuda: el cronograma d'intercanvi de paquets ens ajuda a visualitzar millor aquest intercanvi, I a comptar la quantitat de paquets



	Ethernet		ARP		IP			
	Origen	Destinació	Comanda	Missatge	Origen	Destinació	Protocol	Contingut
1								
2								
3								
4								
5								
6								
7								
8								

Advertència: no sempre cal omplir totes les cel·les de les taules dels anunciats

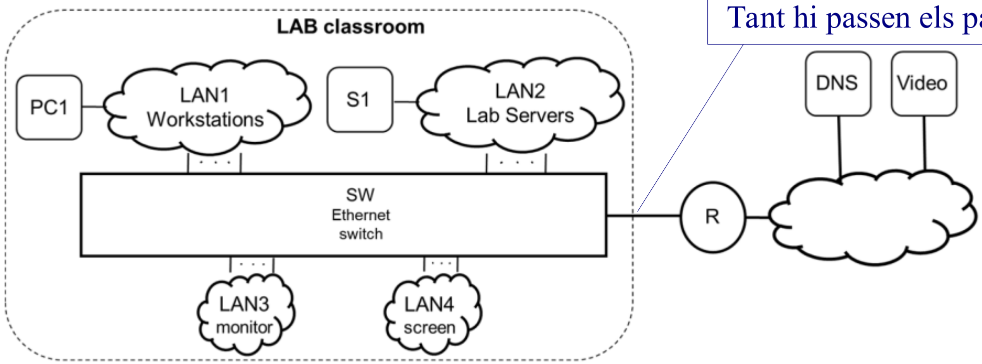
Exercici: 2021t-ef

Unit 2: IP Networks

Problema 1 (3,5 punts)

La figura mostra la configuració d'una aula de laboratori on hi ha llocs de treball (LAN1), servidors per donar suport als treballs dels laboratoris (LAN2), un PC de monitorització pel professor (LAN3) i una pantalla IP per a vídeo (LAN4). Cada laboratori disposa d'un commutador Ethernet (SW) on es configuren les 4 xarxes locals virtuals (VLAN) i l'adreçament proposat per a cada aula és 192.168.aula.0/24. El router R dona servei a més de 40 laboratoris amb la mateixa configuració.

Ajuda: aquest és l'enllaç que hem d'analitzar. Tant hi passen els paquets PC1-R com R-S1

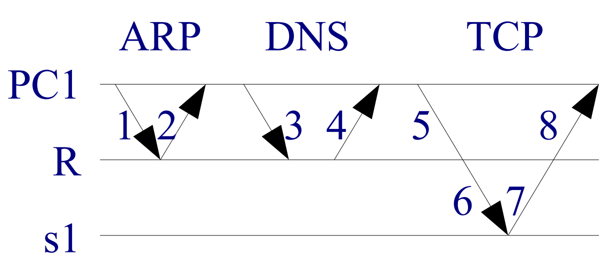


Ajuda: una connexió TCP comença amb el client enviat un SYN, al qual el servidor respon amb un SYN/ACK ...

c) (0,5 punts) La configuració que obté PC1 és: 192.168.1.2; router per defecte (gw): 192.168.1.1; DNS: 147.83.3.3. El PC1 inicia una connexió TCP amb el servidor s1-aula.fib.upc.edu. Completa la seqüència de trames i datagrames que passen per l'enllaç entre el commutador Ethernet i el router fins que PC1 rep el SYN/ACK. <u>El router R ja té la informació a la taula ARP de tots els servidors</u>.

Notació: majúscules per les adreces IP i minúscules per les adreces Ethernet (MAC). Exemple: PC1, pc1.

Ajuda: el cronograma d'intercanvi de paquets ens ajuda a visualitzar millor aquest intercanvi, I a comptar la quantitat de paquets



	Ethernet		ARP		IP			
	Origen	Destinació	Comanda	Missatge	Origen	Destinació	Protocol	Contingut
1	pc1	FF...FF	REQ	R?				
2	r	pc1	RESP	R -> r				
3	pc1	r			PC1	DNS	UDP	s1-aula.fib.upc.edu A?
4	r	pc1			DNS	PC1	UDP	DNS A=S1
5	pc1	r			PC1	S1	TCP	SYN
6	r	s1			PC1	S1	TCP	SYN
7	s1	r			S1	PC1	TCP	SYN/ACK
8	r	pc1			S1	PC1	TCP	SYN/ACK

Advertència: no sempre cal omplir totes les cel·les de les taules dels anunciats

Unit 2: IP Networks

Outline

- IP layer service
- IP addresses
- Subnetting
- Routing tables
- ARP protocol
- **IP header**
- ICMP protocol
- DHCP protocol
- NAT
- Routing algorithms
- Security in IP

Unit 2: IP Networks

IPv4 Header (RFC 791 + updates)

<table>
<tr><th>Offsets</th><th>Octet</th><th colspan="8">0</th><th colspan="8">1</th><th colspan="8">2</th><th colspan="8">3</th></tr>
<tr><th>Octet</th><th>Bit</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th></tr>
<tr><td>0</td><td>0</td><td colspan="4">Version</td><td colspan="4">IHL</td><td colspan="6">DSCP</td><td colspan="2">ECN</td><td colspan="16">Total Length</td></tr>
<tr><td>4</td><td>32</td><td colspan="16">Identification</td><td colspan="3">Flags</td><td colspan="13">Fragment Offset</td></tr>
<tr><td>8</td><td>64</td><td colspan="8">Time To Live</td><td colspan="8">Protocol</td><td colspan="16">Header Checksum</td></tr>
<tr><td>12</td><td>96</td><td colspan="32">Source IP Address</td></tr>
<tr><td>16</td><td>128</td><td colspan="32">Destination IP Address</td></tr>
<tr><td>20</td><td>160</td><td colspan="32" rowspan="3">Options (if IHL > 5)</td></tr>
<tr><td>⋮</td><td>⋮</td></tr>
<tr><td>56</td><td>448</td></tr>
</table>

https://en.wikipedia.org/wiki/IPv4#Header

```
        0               1               2               3       bytes
+---------------+---------------+---------------+---------------+
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 bits
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |    DSCP   |ENC|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    [Padding]  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Unit 2: IP Networks

IPv4 Header (cont.)

- 14 fields, 13 required
- `Version` – [4b] Version of the header: IPv4 vs. IPv6
  - `0100` for IPv4
  - `0110` for IPv6
- `IHL` (Internet Header Length) – [4b] Number of 32-bit words in the header
  - Min 5 (4*5 = 20 bytes); max 15 (4*15 = 60 bytes)
  - Length = [20, 24, 28, …, 60] bytes
- `DSCP` & `ECN` (previously `ToS`) – [6b] & [2b]
  - Differentiated Services Code Point – related to quality of service (QoS) management (e.g. voice over IP, VoIP)
  - Explicit Congestion Notification – related to end-to-end notification of network congestion

Unit 2: IP Networks

IPv4 Header (cont.)

- `Total Length` – [16b] Packet size in bytes (header + data)
  - Min 20 bytes (header without data)
  - Max 65,535 (2^{16})
- `Identification` – [16b] Used for identifying the group of fragments of a single (fragmented) IP datagram
- `Flags` – [3b] Used to control fragmentation or identify fragments
  - bit 0: Reserved; always zero
  - bit 1: Don't Fragment (DF)
  - bit 2: More Fragments (MF)
- `Fragment offset` – [13b] Offset of a particular fragment relative to the beginning of the original unfragmented IP datagram in units of eight-byte blocks
- **`Time to Live`** – [8b] Each hop is decreased by one. IF zero, the router discards the packet and typically sends an ICMP time exceeded message to the sender.

Unit 2: IP Networks

IPv4 Header (cont.)

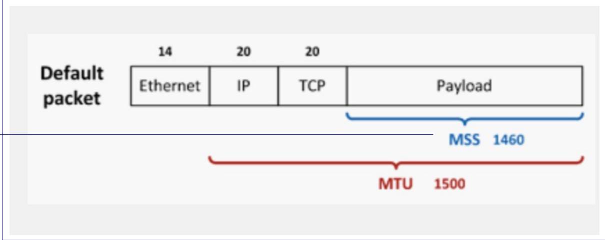
- **`Protocol`** – [8b]
  - `0x01` (1) for ICMP
  - `0x04` (4) for IP-in-IP
  - `0x06` (6) for TCP
  - `0x11` (17) for UDP
- `Header checksum` – [16b] Used for error-checking of the header. Each router calculates the checksum of the header and compares it to the checksum field. If the values do not match, the router discards the packet.
- **`Source IP address`** – [32b]
- **`Destination IP address`** – [32b]
- Options – [max 40B] Options in 32-byte words

Unit 2: IP Networks

IP Fragmentation

- Fragmentation may occur:
  - Router: Fragmentation may be needed when two networks with different *Maximum Transfer Unit* (MTU) are connected.

TCP: current common value MSS = 1448 see ch4 for details



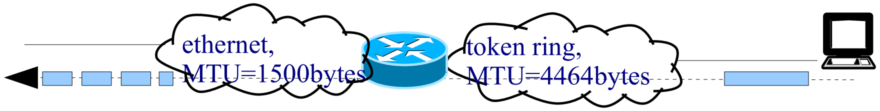
Only in UDP!
Given that the UDP header is 8-byte long and that the Ethernet frame's payload is 1500-byte, fragmentation will occur when the payload of an UDP datagram is larger than 1472-byte (1500-20-8 = ETH-IP-UDP)

TCP: See MTU Path Discovery

  - Host: Fragmentation may be needed using UDP. TCP segments are always ≤ MTU.

Always only one UDP header, regardless fragmentation, because fragmentation occurs one layer below

- Datagrams are reconstructed at the destination.
- Fields:
  - Identification (16 bits): identify fragments from the same datagram.
  - Flags (3 bits):
    - D, don't fragment. Used in path MTU discovery
    - M, More fragments: Set to 0 only in the last fragment
  - Offset (13 bits): Position of the fragment first byte in the original datagram in 8 byte words (indexed at 0).



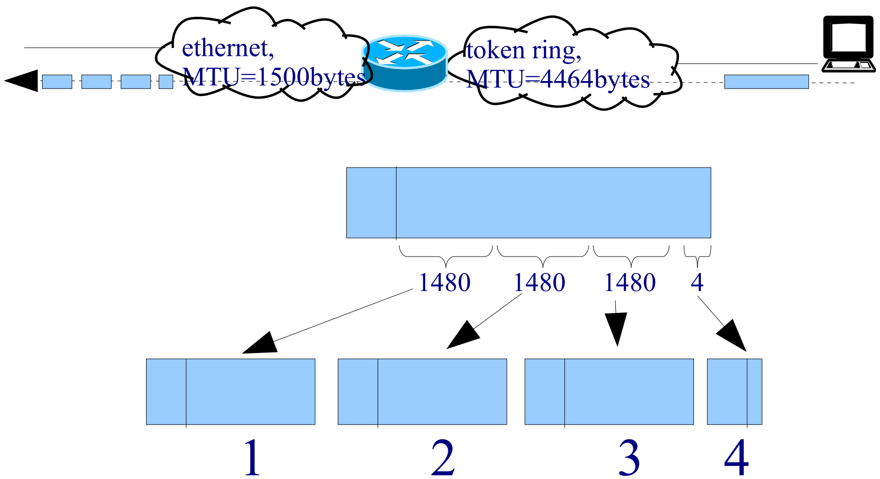
Unit 2: IP Networks

IP Fragmentation - Example

- Original datagram = 4464 bytes (4Mbps Token Ring): 20 header + 4444 payload.
- Fragment size = $\left\lfloor \frac{1500-20}{8} \right\rfloor = 185$ 8-byte-words (1480 bytes)

  Reminder: fragments sizes in bytes are multiple of 8

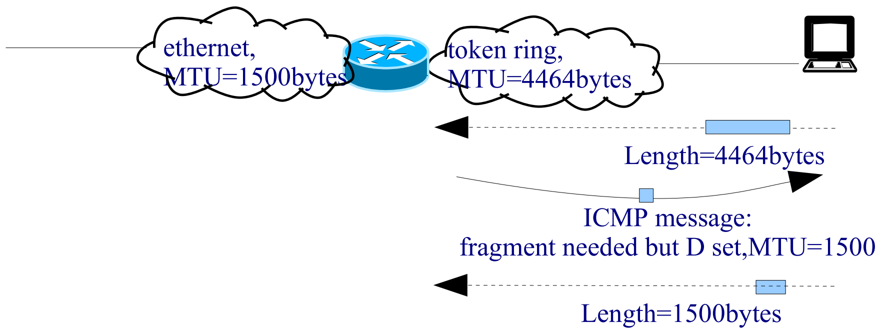
  - 1st fragment: offset = 0 , M = 1. 0~1479 payload bytes.
  - 2nd fragment: offset = 185, M = 1. 1480~2959 payload bytes.
  - 3rd fragment: offset = 370, M = 1 . 2960~4439 payload bytes.
  - 4th fragment: offset = 555, M = 0 . 4440~4443 payload bytes.



Unit 2: IP Networks

MTU Path Discovery (RFC 1191)

- Used in modern TCP implementations.
- TCP by default chooses the Maximum Segment Size (MSS), to avoid headers overhead (segment efficiency = TCP payload / (TCP payload + Σ TCP,IP,Data-link,Physical headers)
  - Default MSS = MTU – (TCP header + IP header) (see Unit 4)
- Goal: avoid fragmentation: The DF flag is set to one, segment size is reduced upon receiving ICMP error message "fragmentation needed but DF flag set"



Unit 2: IP Networks

Outline

- IP layer service
- IP addresses
- Subnetting
- Routing tables
- ARP protocol
- IP header
- **ICMP protocol**
- DHCP protocol
- NAT
- Routing algorithms
- Security in IP

Unit 2: IP Networks

Internet Control Message Protocol, ICMP (RFC 792)

- Used for attention and error messages.
- Can be generated by
  - IP (e.g. TTL expiration)
  - ARP (e.g. resolution not possible)
  - Applications (e.g. ping)
- Are encapsulated into an IP datagram
  - (no UDP/TCP!)
  - It is an internet layer protocol
- Can be
  - i) query
  - ii) error
- Error messages are sent to the source IP of the datagram that generated the error condition
- An ICMP error message cannot generate another ICMP error message (to avoid loops)

Error messages only! Ping is a query message => it can trigger an ICMP error message (e.g. "network unreachable", "time exceed")

Unit 2: IP Networks

ICMP general format message (RFC 792)

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 bits
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  content (variable)                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Identifies the message

Is computed using all the message

- Query type messages have an identifier field, for request-reply correspondence.
- Error messages have a field where the first 8 bytes of the datagram payload causing the error are copied. These bytes capture the TCP/UDP ports. E.g. Destination Unreachable Message:

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             unused                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Internet Header + 64 bits of Original Data Datagram      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Unit 2: IP Networks

Common ICMP messages

Type	Code	query/error	Name	Description
0	0	query	echo reply	Reply an echo request
3	0	error	network unreachable	Network not in the RT.
	1	error	host unreachable	ARP cannot solve the address.
	2	error	protocol unreachable	IP cannot deliver the payload
	3	error	port unreachable	TCP/UDP cannot deliver the payload
	4	error	fragmentation needed and DF set	MTU path discovery
4	0	error	source quench	Sent by a congested router.
5	0	error	redirect for network	When the router send a datagram by the same interface it was received.
8	0	query	echo request	Request for reply
11	0	error	time exceeded, also known as TTL=0 during transit	Sent by a router when `--TTL=0`

Example – Gratuitous ARP & ICMP

xc-grau-2-ip__ARP_Gratuitous_wireshark.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Source	Destination	Protocol	Length	Info
1	38:00:25:8a:1b:50	ff:ff:ff:ff:ff:ff	ARP	42	Gratuitous ARP for 10.192.8.128 (Request)
2	38:00:25:8a:1b:50	ff:ff:ff:ff:ff:ff	ARP	42	Gratuitous ARP for 10.192.8.128 (Reply)
3	38:00:25:8a:1b:50	ff:ff:ff:ff:ff:ff	ARP	42	Who has 10.192.1.1? Tell 10.192.8.128
4	00:00:0c:07:ac:b2	38:00:25:8a:1b:50	ARP	60	10.192.1.1 is at 00:00:0c:07:ac:b2
5	10.192.8.128	10.192.1.1	ICMP	98	Echo (ping) request id=0x000d, seq=1/256, ttl=64 (reply in 6)
6	10.192.1.1	10.192.8.128	ICMP	98	Echo (ping) reply id=0x000d, seq=1/256, ttl=254 (request in 5)

```
Frame 2: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface wlp0s20f3, id 0
Ethernet II, Src: IntelCor_8a:1b:50 (38:00:25:8a:1b:50), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    Address: Broadcast (ff:ff:ff:ff:ff:ff)
    .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the factory default)
    .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
  Source: IntelCor_8a:1b:50 (38:00:25:8a:1b:50)
    Address: IntelCor_8a:1b:50 (38:00:25:8a:1b:50)
    .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  Type: ARP (0x0806)
Address Resolution Protocol (reply/gratuitous ARP)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  [Is gratuitous: True]
  Sender MAC address: IntelCor_8a:1b:50 (38:00:25:8a:1b:50)
  Sender IP address: 10.192.8.128
  Target MAC address: IntelCor_8a:1b:50 (38:00:25:8a:1b:50)
  Target IP address: 10.192.8.128
```

```
0000  ff ff ff ff ff ff 38 00  25 8a 1b 50 08 06 00 01   ······8· %··P····
0010  08 00 06 04 00 02 38 00  25 8a 1b 50 0a c0 08 80   ······8· %··P····
0020  38 00 25 8a 1b 50 0a c0  08 80                     8·%··P·· ··
```

```
#### frames: 1
## unsolicited ARP REQUEST (-U)
roger@c3:~$ arping -c 1 -U -I wlp0s20f3 10.192.8.128
ARPING 10.192.8.128 from 10.192.8.128 wlp0s20f3
Sent 1 probes (1 broadcast(s))
Received 0 response(s)

#### frames: 2
## unsolicited ARP REPLY (-A)
roger@c3:~$ arping -c 1 -A -I wlp0s20f3 10.192.8.128
ARPING 10.192.8.128 from 10.192.8.128 wlp0s20f3
Sent 1 probes (1 broadcast(s))
Received 0 response(s)

#### frames: 3-6
## standard ping -> triggers ARP REQUSET - ARP REPLY
roger@c3:~$ ping -c1 10.192.1.1
PING 10.192.1.1 (10.192.1.1) 56(84) bytes of data.
64 bytes from 10.192.1.1: icmp_seq=1 ttl=254 time=3.25 ms
[...]
```

Source Hardware Address (eth.src), 6 bytes | Packets: 6 · Displayed: 6 (100.0%) · Dropped: 0 (0.0%) | Profile: Default

Unit 2: IP Networks

Outline

- IP layer service
- IP addresses
- Subnetting
- Routing tables
- ARP protocol
- IP header
- ICMP protocol
- **DHCP protocol**
- NAT
- Routing algorithms
- Security in IP

Unit 2: IP Networks

Dynamic Host Configuration Protocol, DHCP (RFC 2131)

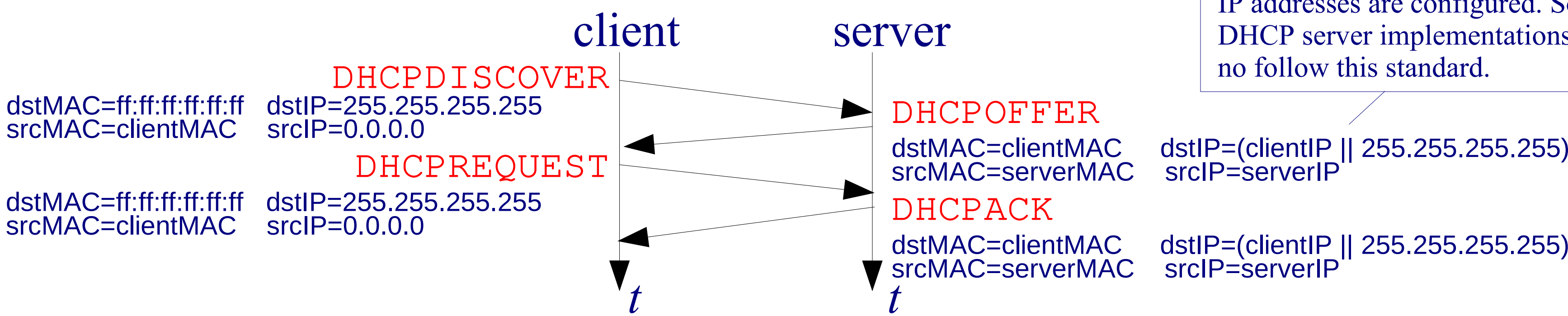
- Improves and can interoperate with previous BOOTP protocol.
- Used for automatic network configuration for assigning:
  - IP address and mask,
  - Default route,
  - Hostname,
  - DNS domain,
  - DNS servers,
  - etc.
- IP address configuration can be:
  - Dynamic: During a leasing time.
  - Automatic: Unlimited leasing time.
  - Manual: IP addresses are assigned to specific MAC addresses.
- It is an application layer protocol (server-client paradigm)
  - Uses UDP as transport protocol

Unit 2: IP Networks

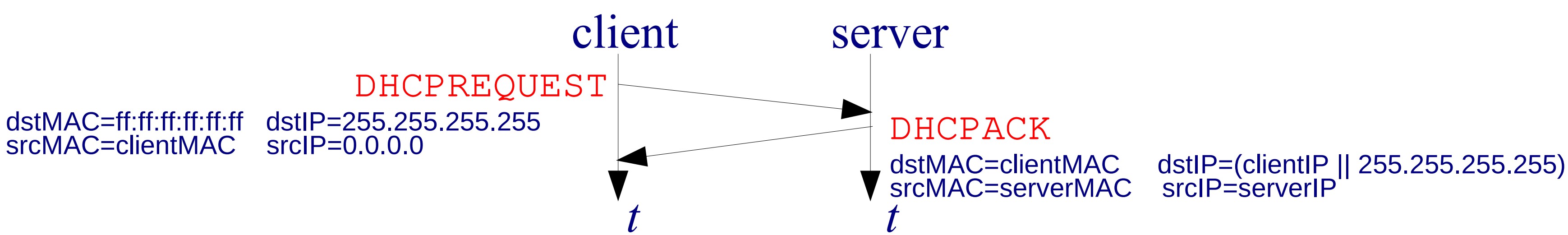
DHCP – Client-server interaction (RFC 2131)

The client specifies whether in the reply (BOOTREPLY) the destination IP should be unicast or broadcast (i.e. clinetIP or 255.255.255.255). Unicast is the standard. Broadcast must be used only by those (old) hosts which cannot accept unicast packets before IP addresses are configured. Some DHCP server implementations do no follow this standard.

- UDP, server port = 67, client port = 68.



- The client can directly send `DHCPREQUEST`:
  - After rebooting if it remembers and wishes to reuse a previously allocated network address.
  - Extending the lease on a particular network address.



Unit 2: IP Networks

DHCP – Message Fields (RFC 2131)

(informative slide, don't learn the message fields by heart!)

Only these 2 exist (not to be confused with `options`).
BOOTREQUEST: client to server
BOOTREPLY: server to client

```
FIELD      OCTETS       DESCRIPTION
op            1  Message op code / message type. 1 = BOOTREQUEST, 2 = BOOTREPLY.
htype         1  Hardware address type.
hlen          1  Hardware address length.
hops          1  Client sets to zero, optionally used by relay agents
                 when booting via a relay agent.
xid           4  Transaction ID, a random number chosen by the
                 client, used by the client and server to associate
                 messages and responses between a client and a
                 server.
secs          2  Filled in by client, seconds elapsed since client
                 began address acquisition or renewal process.
flags         2  Flags (0x0000 unicast, 0x8000 broadcast).
ciaddr        4  Client IP address; only filled in if client is in
                 BOUND, RENEW or REBINDING state and can respond
                 to ARP requests.
yiaddr        4  'your' (client) IP address. Set by the server in
                 a DHCPOFFER message.
siaddr        4  IP address of next server to use in bootstrap;
                 returned in DHCPOFFER, DHCPACK by server.
giaddr        4  Relay agent IP address, used in booting via a
                 relay agent.
chaddr       16  Client hardware address.
sname        64  Optional server host name, null terminated string.
file        128  Boot file name, null terminated string; "generic"
                 name or null in DHCPDISCOVER, fully qualified
                 directory-path name in DHCPOFFER.
options     var  Optional parameters field.
```

DHCP messages (`DHCPDISCOVER, DHCPOFFER, etc.`) go here. See next slide

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     op (1)    |   htype (1)   |   hlen (1)    |   hops (1)    |
+---------------+---------------+---------------+---------------+
|                            xid (4)                            |
+-------------------------------+-------------------------------+
|           secs (2)            |           flags (2)           |
+-------------------------------+-------------------------------+
|                          ciaddr  (4)                          |
+---------------------------------------------------------------+
|                          yiaddr  (4)                          |
+---------------------------------------------------------------+
|                          siaddr  (4)                          |
+---------------------------------------------------------------+
|                          giaddr  (4)                          |
+---------------------------------------------------------------+
|                                                               |
|                          chaddr  (16)                         |
|                                                               |
|                                                               |
+---------------------------------------------------------------+
|                                                               |
|                          sname   (64)                         |
+---------------------------------------------------------------+
|                                                               |
|                          file    (128)                        |
+---------------------------------------------------------------+
|                                                               |
|                          options (variable)                   |
+---------------------------------------------------------------+
```

Unit 2: IP Networks

DHCP – Protocol Messages (RFC 2131)

Additional messages in later RFCs (RFC3203, RFC4388, RFC6926, RFC7724)

Code	Message	Use
1	DHCPDISCOVER	Client broadcast to locate available servers.
2	DHCPOFFER	Server to client in response to DHCPDISCOVER with offer of configuration parameters.
3	DHCPREQUEST	Client message to servers either (a) requesting offered parameters from one server and implicitly declining offers from all others, (b) confirming correctness of previously allocated address after, e.g., system reboot, or (c) extending the lease on a particular network address.
5	DHCPACK	Server to client with configuration parameters, including committed network address.
4	DHCPDECLINE	Client to server indicating network address is already in use.
6	DHCPNAK	Server to client indicating client's notion of network address is incorrect (e.g., client has moved to new subnet) or client's lease as expired
7	DHCPRELEASE	Client to server relinquishing network address and cancelling remaining lease.
8	DHCPINFORM	Client to server, asking only for local configuration parameters; client already has externally configured network address.

Unit 2: IP Networks

DHCP – Example: tcpdump/dhcpdump capture

```
linux # tcpdump -lenx -s 1500 -i eth0 port bootps or port bootpc | dhcpdump
   TIME: 17:09:24.616312
     IP: 0.0.0.0.68 (00:30:1b:b4:6d:78) > 255.255.255.255.67 (ff:ff:ff:ff:ff:ff)
     OP: 1 (BOOTPREQUEST)
  HTYPE: 1 (Ethernet)
    XID: 181f0139
  FLAGS: 0
 CIADDR: 0.0.0.0
 YIADDR: 0.0.0.0
 SIADDR: 0.0.0.0
 GIADDR: 0.0.0.0
 CHADDR: 00:30:1b:b4:6d:78:00:00:00:00:00:00:00:00:00:00
 OPTION:  53 (  1) DHCP message type         3 (DHCPREQUEST)
 OPTION:  57 (  2) Maximum DHCP message size 576
 OPTION:  50 (  4) Request IP address        192.168.1.100
 OPTION:  51 (  4) IP address leasetime      -1 ()
 OPTION:  55 ( 21) Parameter Request List      1 (Subnet mask)
                                               3 (Routers)
                                               6 (DNS server)
                                              12 (Host name)
                                              15 (Domainname)
                                              23 (Default IP TTL)
                                              28 (Broadcast address)
                                              29 (Perform mask discovery)
                                              42 (NTP servers)
                                               9 (LPR server)
                                             119 (Domain Search)
                                             ...
---------------------------------------------------------------------------
   TIME: 17:09:24.619312
     IP: 192.168.1.1.67 (00:18:39:5d:74:9d) > 192.168.1.100.68 (00:30:1b:b4:6d:78)
     OP: 2 (BOOTPREPLY)
  HTYPE: 1 (Ethernet)
    XID: 181f0139
  FLAGS: 0
 CIADDR: 0.0.0.0
 YIADDR: 192.168.1.100
 SIADDR: 192.168.1.1
 GIADDR: 0.0.0.0
 CHADDR: 00:30:1b:b4:6d:78:00:00:00:00:00:00:00:00:00:00
 OPTION:  53 (  1) DHCP message type         5 (DHCPACK)
 OPTION:  54 (  4) Server identifier         192.168.1.1
 OPTION:  51 (  4) IP address leasetime      86400 (24h)
 OPTION:   1 (  4) Subnet mask               255.255.255.0
 OPTION:   3 (  4) Routers                   192.168.1.1
 OPTION:   6 (  4) DNS server                192.168.0.1
 OPTION:  15 (  3) Domainname                lan
```

Protocol numbers (IP header): 67 server, 68 client

Unicast in the BOOTREPY because the BOOTREQUEST had the Broadcast flag set to 0

Unit 2: IP Networks

DHCP – Example: Wireshark captures

Linux commands
```
## dhcp release (packet 1)
sudo dhclient -r -v eth0
## dhcp discover (packet 2-5)
sudo dhclient v eth0
```

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

dhcp

Time	No.	Source	Destination	Protocol	Length	Info	Pk len	Hardware src	Hardware dst
0.000000000	1	172.28.0.100	172.28.0.2	DHCP	342	DHCP Release - Transaction ID 0xfcb2e55e	342	aa:04:6c:fe:c4:cd	02:42:ac:1c:00:02
8.091772102	15	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xa10d5020	342	02:42:66:2a:02:ba	ff:ff:ff:ff:ff:ff
8.092447314	16	172.28.0.2	172.28.1.10	DHCP	342	DHCP Offer - Transaction ID 0xa10d5020	342	02:42:ac:1c:00:02	02:42:66:2a:02:ba
8.093157703	17	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0xa10d5020	342	02:42:66:2a:02:ba	ff:ff:ff:ff:ff:ff
8.111699728	18	172.28.0.2	172.28.1.10	DHCP	342	DHCP ACK - Transaction ID 0xa10d5020	342	02:42:ac:1c:00:02	02:42:66:2a:02:ba

Broadcast bit obeyed

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

dhcp

Time	No.	Source	Destination	Protocol	Length	Info	Pk len	Hardware src	Hardware dst
11.773075151	3	192.168.1.171	192.168.1.1	DHCP	342	DHCP Release - Transaction ID 0x9491e234	342	38:00:25:8a:1b:50	c0:fd:84:9e:e7:6f
62.410651348	10	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x3655b616	342	38:00:25:8a:1b:50	ff:ff:ff:ff:ff:ff
64.715543005	12	192.168.1.1	255.255.255.255	DHCP	324	DHCP Offer - Transaction ID 0x3655b616	324	c0:fd:84:9e:e7:6f	38:00:25:8a:1b:50
64.716397170	13	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x3655b616	342	38:00:25:8a:1b:50	ff:ff:ff:ff:ff:ff
64.813518666	14	192.168.1.1	255.255.255.255	DHCP	324	DHCP ACK - Transaction ID 0x3655b616	324	c0:fd:84:9e:e7:6f	38:00:25:8a:1b:50

Broadcast bit not obeyed

Unit 2: IP Networks

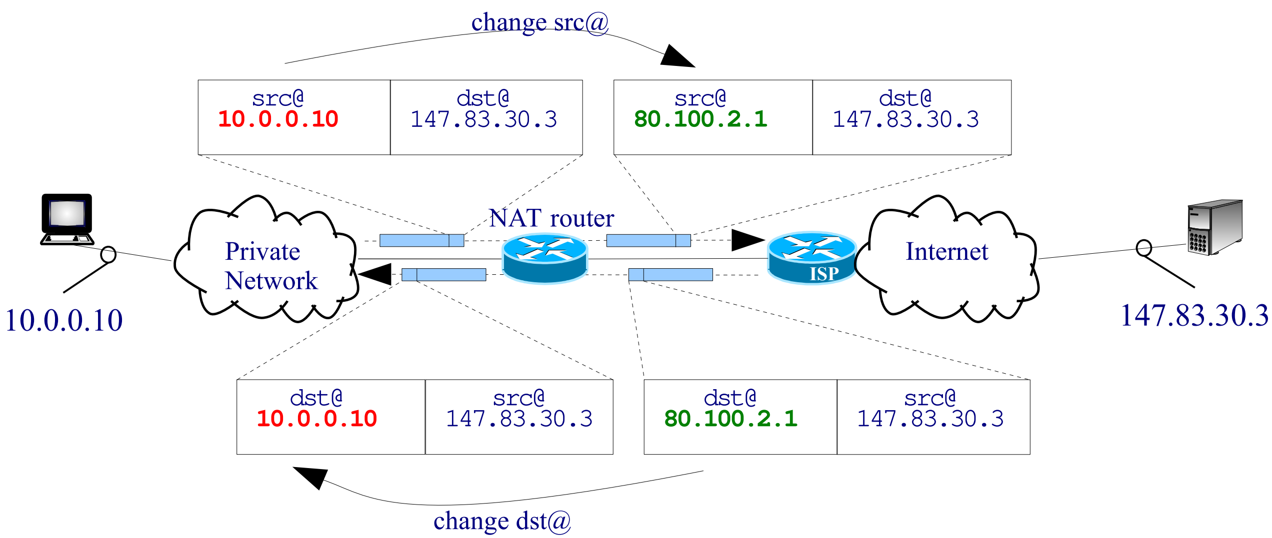
Outline

- IP layer service
- IP addresses
- Subnetting
- Routing tables
- ARP protocol
- IP header
- ICMP protocol
- DHCP protocol
- **NAT**
- Routing algorithms
- Security in IP

Unit 2: IP Networks

Network Address Translation, NAT (RFCs 1631, 2663 3022)

- NOTE: NAT is a technique, not a protocol. Implementations and terminology may change from one manufacturer to another.
- A NAT table is used for address mapping.
- Advantages:
  - Administration, e.g. changing ISP does not imply changing private network addressing.
  - Save public addresses. Example: Private addresses (internal addresses) are translated to public addresses (external addresses).



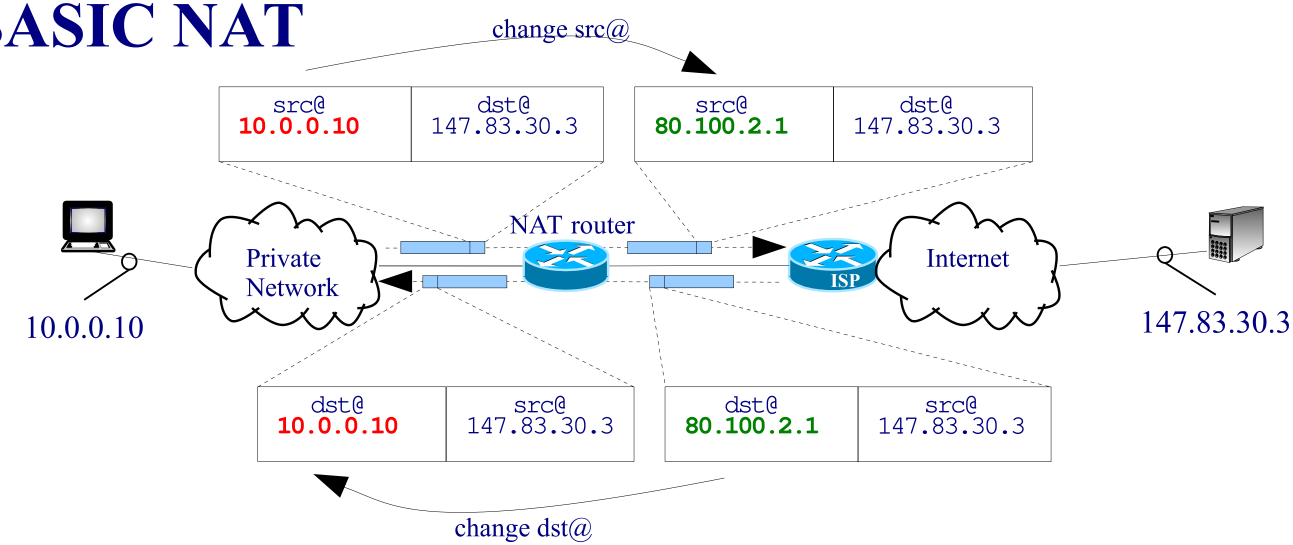
Unit 2: IP Networks

NAT – Types of translations

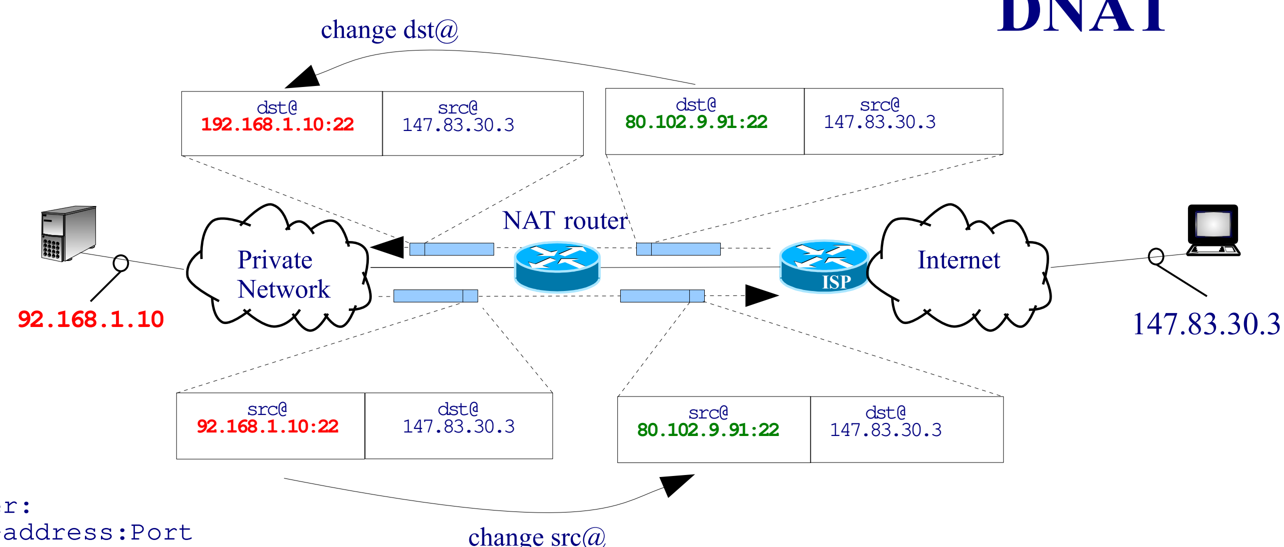
- Basic NAT (one-to-one):
  - A different external address is used for each internal address:
    - A different public IP address is needed for each hosts accessing Internet.
  - Each NAT table entry has the tuple: (internal address, external address).
  - Each host requires one NAT table entry.
  - Manually configured
- Port and Address Translation, PAT (one-to-many):
  - The same external address can be used for each internal address.
    - A unique public IP address can be used for all (internal) hosts accessing Internet.
  - Each NAT table entry has the tuple: (int. addr., int. port, ext. addr., ext. port)
  - Each connection requires one NAT table entry.
  - Entries are automatically added when an internal connection is initiated.
- Destination NAT, DNAT can be:
  - Enables external connections to internal servers.
  - The address translation is exactly the same as NAT, but, the connection is initiated from an external client.
  - Typically, some static configuration is needed to configure the server IP/port.

Unit 2: IP Networks

BASIC NAT



DNAT

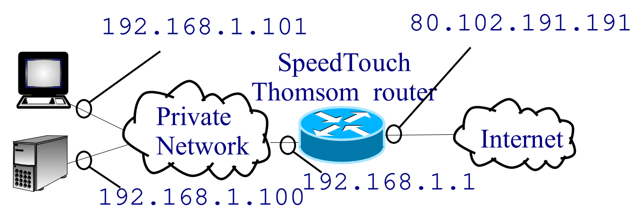



```
Static entry in the NAT router:
Inside-address:Port  Outside-address:Port
192.168.1.10:22      80.102.9.91:22
```

Unit 2: IP Networks

NAT – ADSL commercial router example

- NAT outgoing packets to 80.102.191.191
- DNAT incoming packets, port 22 (ssh) to 192.168.1.100



In PAT, source ports of packets to internet may be changed (e.g. to distinguish 2 connections of 2 different internal devices with same source port and same dstIP and destination port, or because the desired source port is already in use) but destination ports are kept as set by the internal devices, as same as the IP (otherwise the packets would not reach the intended destination)

```
linux # telnet 192.168.1.1
Trying 192.168.0.1...
Connected to 192.168.1.1.
=>nat
[nat]=>list
     Indx Prot Inside-address:Port  Outside-address:Port  Foreign-address:Port Flgs  Expir  State  Control
DNAT  2    6    192.168.1.100:22     80.102.191.191:22            0.0.0.0:0      instance
      6    6    192.168.1.101:1420   80.102.191.191:10079   83.60.122.22:45730   1    14m48       1
SNAT 11    6    192.168.1.101:1337   80.102.191.191:10060  85.56.136.231:16000   1    14m30       1
     12    6    192.168.1.101:1402   80.102.191.191:10064   82.159.8.187:1755    1      14s       5
     ...
```

Source NAT (SNAT) is a common synonym of PAT. The name is a counterpart of destination NAT (DNAT).

In Linux kernel NAT is implemented as part of the Netfilter framework. iptables is the user-space utility program that allows a system administrator to configure the IP packet filter rules. Example:

```
root@OpenWrt:~# iptables -L -t nat
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
prerouting_rule  all  --  anywhere             anywhere            /* !fw3: Custom prerouting rule chain */
zone_lan_prerouting  all  --  anywhere             anywhere            /* !fw3 */
zone_wan_prerouting  all  --  anywhere             anywhere            /* !fw3 */
...
```

Unit 2: IP Networks

Outline

- IP layer service
- IP addresses
- Subnetting
- Routing tables
- ARP protocol
- IP header
- ICMP protocol
- DHCP protocol
- NAT
- **Routing algorithms**
- Security in IP

Unit 2: IP Networks

Routing algorithms

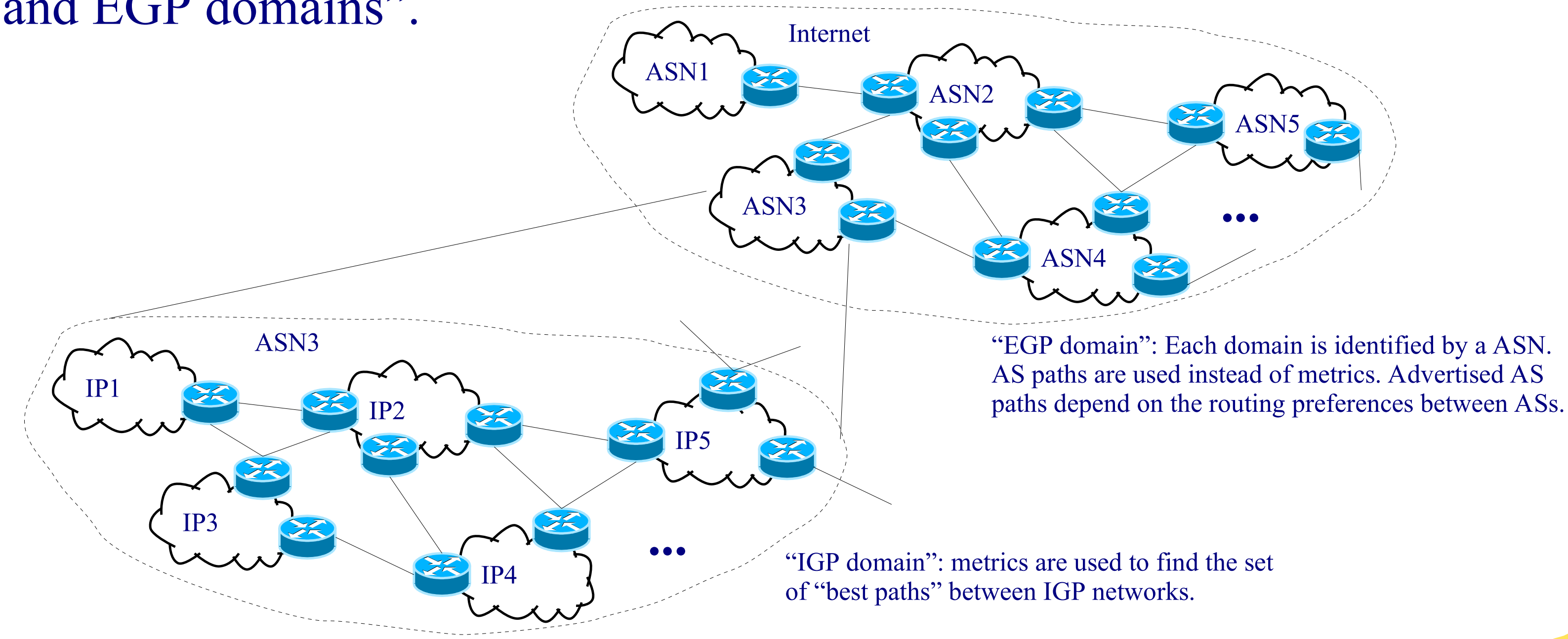
- Objective: add/update/remove entries to routing tables. Can be:
  - Static: Manual, scripts, DHCP.
  - Dynamic: Automatically update table entries, e.g. when a topology change occurs. This is done by a routing algorithm (also routing protocol).

- Internet is organized in Autonomous Systems (AS). In terms of ASs, routing algorithms are classified as:
  - Interior Gateway Protocols (IGPs): Inside the same AS. Examples:
    - RFC standards: RIP, OSPF.
    - Proprietary: CISCO IGRP.
  - Exterior Gateway Protocols (EGPs): Between different Ass.
    - Currently BGPv4.

Unit 2: IP Networks

Routing algorithms - Autonomous Systems (AS)

- AS definition (RFC 1930): "An AS is a connected group of one or more IP prefixes run by one or more network operators which has a SINGLE and CLEARLY DEFINED routing policy".
- Each AS is identified by a 16 or 32 bits AS Number (ASN) assigned by IANA.
- ASs facilitate Internet routing by introducing a two-level hierarchy: "IGP and EGP domains".



Unit 2: IP Networks

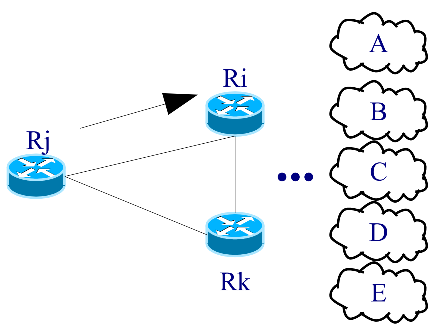
Routing Information Protocol, RIP (RFC 2453)

- The metric (distance) to a destination is the number of hops (i.e. transmissions) to reach the destination: 1 if the destination is attached to a directly connected network, 2 if 1 additional router is needed, etc.
- Infinite metric is 16.
- Routers send RIP updates every 30 seconds to the neighbors.
- RIP updates use UDP, with the same source and destination port: 520, broadcast dst. IP addr. (Version 1).
- RIP updates include destinations and metrics tuples.
- A neighbor is considered down if no RIP messages are seen during 180 seconds.
- Two versions of RIP:
  - Version 2 i) allows variable masks (=> masks are added to the messages) and ii) uses the multicast dst. address 224.0.0.9 (all RIPv2 routers).
  - RIP Version 1 is classful aggregation protocol.
- The routing algorithms where it is not known the whole topology but just the distance to each destination, are known as "distance-vector". Many of the use a distributed variant of the "Bellman-Ford" algorithm for selecting the shortest path.

Unit 2: IP Networks

RIP – Routing Table (RT) Update Example

- Upon receiving an update
  1) Increase the message metrics (+1 to all metrics received)
  2) change the routing table if:
     – There is a better path (lower metric) towards a destination
     – The gateway being used changes the metric
     – There is a new route
- Example: When Ri receives an update message from Rj:



D	G	M
A	Rk	4
B	Rj	3
C	Rk	5
D	Rj	2

Ri's RT

➡

D	M
A	1
B	4
C	5
D	1
E	3

Ri receives Rj's update message

➡

D	M
A	2
B	5
C	6
D	2
E	4

Rj's metrics increased

➡

D	G	M
A	Rj	2
B	Rj	5
C	Rk	5
D	Rj	2
E	Rj	4

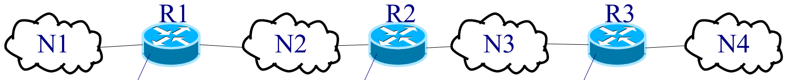
Ri's RT updated

Note: updates do not include GWs

Unit 2: IP Networks

RIP – Count to Infinity: Problem

- When there are changes on the topology (e.g. a router fails), depending on the route update message order, convergence problems may arise
- Example: Evolution of destination N4 entry when R3 fails
  - Starting point: stable topology & stable routing tables:



R1's RT

D	G	M
N1	*	1
N2	*	1
N3	R2	2
N4	R2	3

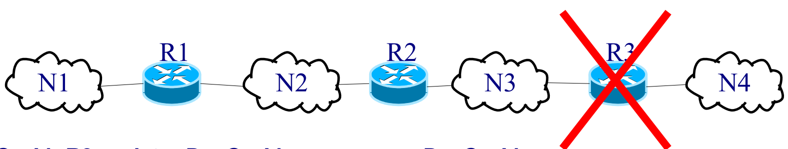
R2's RT

D	G	M
N1	R1	2
N2	*	1
N3	*	1
N4	R3	2

R3's RT

D	G	M
N1	R2	3
N2	R2	2
N3	*	1
N4	*	1

  - Now R3 fails => N4 becomes unreachable:



	D G M	R3 fails	D G M	R1 update	D G M	R2 update	D G M	R2 update	D G M		D G M
R1:	N4 R2 3	→	N4 R2 3	→	N4 R2 3	→	N4 R2 **5**	→	N4 R2 5	...	N4 R2 **16**
R2:	N4 R3 2	→	N4 R3 **16**	→	N4 R1 **4**	→	N4 R1 4	→	N4 R1 **6**	...	N4 R1 **16**

The change is only noticed by R3 => only R3 changes its RT

This is where the sequence problem (loop) starts: the first update is sent by R1. If it had been sent by R2 → no loop

R2 changes N4's route (GW and metric) because it been offered a better path towards that destination

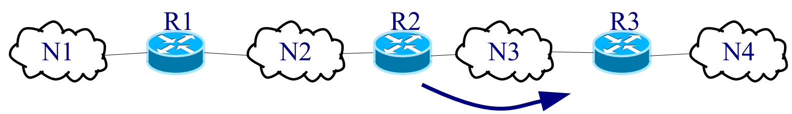
R1 updates N4's route despite being worse because it is comes from its GW (R2) towards that destination

R2 updates N4's route despite being worse because it is comes from its GW (R1) towards that destination

Unit 2: IP Networks

RIP – Count to Infinity: Solutions

- Split horizon: When the router sends the update, removes the entries having a gateway in the interface where the update is sent:



D	G	M
N1	R1	2
N2	*	1
N3	*	1
N4	R3	2

R2's RT

D	M
N1	2
N2	1

update sent by R2

- Split horizon with Poisoned Reverse: the same as split horizon except that the entries with M=16 are not removed. The poison reverse rule overwrites split horizon rule: *poisoned routes* are also sent ("back") to the neighbor from which were learnt.
- Triggered updates: Consists of sending the update before the 30 seconds timer expires when a metric change in the routing table.
- Hold down timer (CISCO): When a route becomes unreachable (metric = 16), the entry is placed in *holddown* during 180 seconds. During this time, the entry is not updated.

To allocate time for poisoned routes to propagate.

Unit 2: IP Networks

Open Shortest Path First, OSPF (RFC 2328)

- IETF standard for high performance IGP routing protocol.
- *Link-state* protocol: Routers monitor neighbor routers and networks and send this information to all OSPF routers (*Link State Advertisements*, LSA).
- LSA are encapsulated into IP datagrams with multicast destination address 224.0.0.5, and routed using *flooding*.
- LSA are only sent when changes in the neighborhood occur, or when a LSA Request is received.
- Neighbor routers are monitored using a *hello protocol*.
- OSPF routers maintain a LS database with the information received with LSA. The Shortest Path First algorithm (Dijkstra algorithm) is used to optimal build routing table entries.
- The metric is computed taking into account link bitrates, delays etc.
- The infinite metric is the maximum metric value.
- There is no convergence (count to infinity) problems.

Unit 2: IP Networks

Distance-vector vs Link-state (I)

- Distance-vector routing protocols
  - Distance→"metric"; vector→"direction"
  - Each node (just) build and maintain information about its best next hop towards every destination (local knowledge)
    - => less memory requirements because nodes do not info of the full topology
    - Strategy know as "routing by rumor"
  - The nodes send all the entire routing table (but just) to their neighbors
    - Less bandwidth (protocol overhead)
  - Algorithm to find the shortest path (i.e. the best route) towards a destination:
    - Distributed variant of Bellman-Ford
  - Persistent loops (count to inf.) if no measures are taken. Some measures:
    - Split-horizon with poisoned reverse (e.g. RIP)
    - Introduction of sequence numbers (each router will always prefer routes with the most recent sequence number) (e.g. DSDV)
  - Examples: RIP, BGP, DSDV, BMX, Batman-adv, Babel

Unit 2: IP Networks

Distance-vector vs Link-state (II)

- Link-state routing protocols
  - Each node builds a map (a tree) of the full network topology (global knowledge)
    - Higher memory requirements
  - The nodes link-state advertisements that are flooded through all the network
    - Less protocol overhead (less protocol traffic)
  - Algorithm to find the shortest path (i.e. the best route) towards a destination:
    - Dijkstra
      - More computing intensive
  - The metric is computed taking into account link bitrates, delays etc.
    - Many of the modern distance-vector RP too
  - No persistent loops (only transient)
  - Examples: OSPF, OSLR, ISIS

Unit 2: IP Networks

Outline

- IP layer service
- IP addresses
- Subnetting
- Routing tables
- ARP protocol
- IP header
- ICMP protocol
- DHCP protocol
- NAT
- Routing algorithms
- **Security in IP**

Unit 2: IP Networks

Security in IP

- Goals:
  - Confidentiality: Who can access.
  - Integrity: Who can modify the data.
  - Availability: Access guarantee.
- Vulnerabilities:
  - Technological: Protocols (e.g. ftp and telnet send messages in “clear text”) and networking devices (routers...)
  - Configuration: Servers, passwords, ...
  - Missing security policies: Secure servers, encryption, firewalls, ...

Unit 2: IP Networks

Security in IP – Attacks

- Reconnaissance: Previous to an attack.
  - Available IP addresses.
  - Available servers and ports.
  - Types of OSs, versions, devices...
  - Eavesdropping
- Access: Unauthorized access to an account or service.
- Denial of Service: Disables or corrupts networks, systems, or services.
- Viruses, worms , trojan horses...: Malicious software that replicate itself.

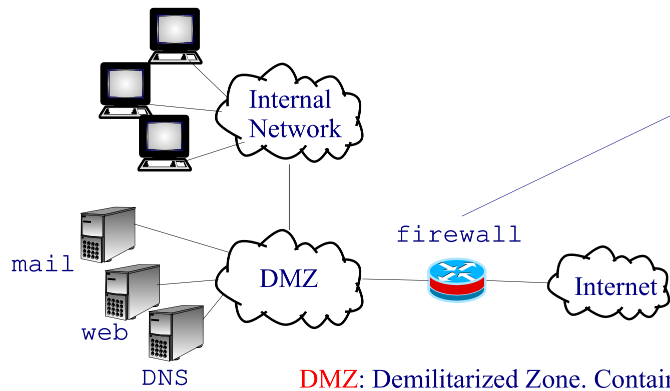
Security in IP – Basic Solutions

- Firewalls.
- Virtual Private Networks (VPN) with encrypted payload.

Unit 2: IP Networks

Security in IP – Firewalls

- Firewall: System or group of systems that enforces an access control policy to a network.
- There are many firewall types:
  - From simple packet filtering based on IP/TCP/UDP header rules,
  - to state-full connection tracking
  - and application-based filtering (packet inspection)



Middlebox is a generic term to refer devices installed between routers (frequently without knowledge of one of both communicating parties). They include: firewalls, traffic shapers, proxies, etc.

DMZ: Demilitarized Zone. Contains servers exposed to the Internet.

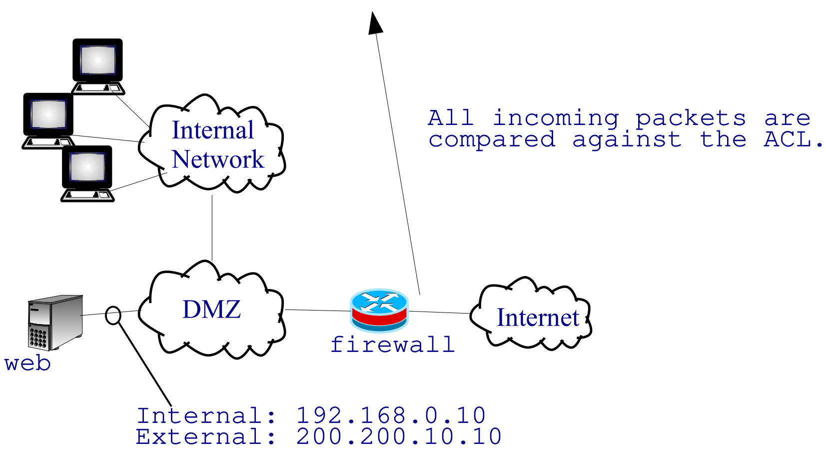
Unit 2: IP Networks

Security in IP – Basic Firewall Configuration

- NAT
- Access Control List, ACL

The order in which the entries appear in the list is crucial: once a packet matches a rule the corresponding action action is taken and no further entries are processed

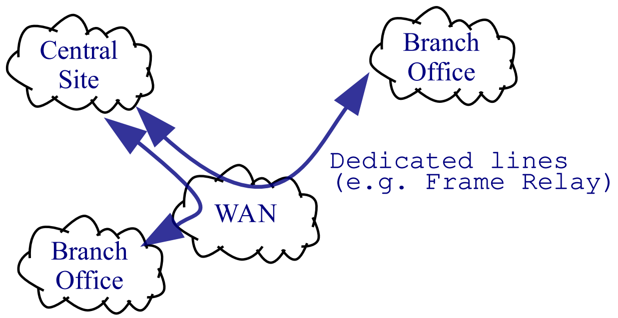
Protocol	IP-src	IP-dst	Port-src	Port-dst	Action
TCP	*any*	200.200.10.10/32	*any*	80	*accept*
TCP	*any*	*any*	< 1024	≥ 1024	*accept*
ICMP	*any*	*any*	–	–	*accept*
IP	*any*	*any*	–	–	*deny*

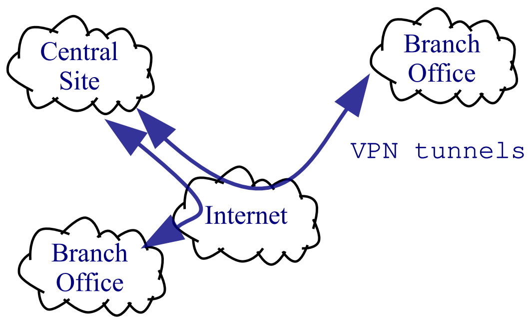


Unit 2: IP Networks

Security in IP – Virtual Private Network, VPN

- Provides connectivity for remote users over a public infrastructure, as they would have over a private network.





Conventional Private Network

- More cost.
- Less flexible.
- WAN management.

VPN

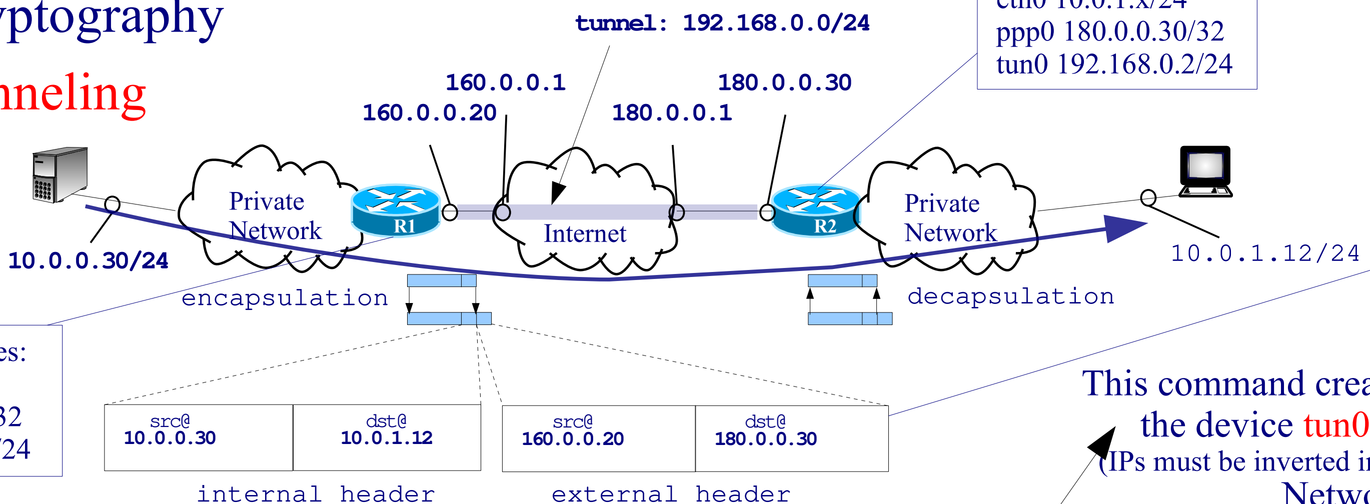
- Less cost.
- More flexible.
- Simple management.
- Internet availability.

Unit 2: IP Networks

Security in IP – VPN Security

- Authentication
- Cryptography
- Tunneling

R2 has 3 interfaces:
eth0 10.0.1.x/24
ppp0 180.0.0.30/32
tun0 192.168.0.2/24



Note: a second IP header is added

R1 has 3 interfaces:
eth0 10.0.0.x/24
ppp0 160.0.0.20/32
tun0 192.168.0.1/24

tun0 is a virtual interface

This command creates the device tun0
(IPs must be inverted in R2)

Network reachable through the tunnel
(192.168.0.1 in R2)

Network of the tunnel

```
Example: creating a tunnel in linux:
R1# ip tunnel add tun0 mode gre remote  180.0.0.30 local 160.0.0.20 ttl 255
R1# ifconfig tun0 192.168.0.1 netmask 255.255.255.0
R1# route add -net 10.0.1.0 netmask 255.255.255.0 gw 192.168.0.2
```

R1 Routing Table

Destination	Gateway	Genmask	Iface
10.0.0.0	0.0.0.0	255.255.255.0	eth0
160.0.0.1	0.0.0.0	255.255.255.255	ppp0
0.0.0.0	160.0.0.1	0.0.0.0	ppp0
192.168.0.0	0.0.0.0	255.255.255.0	tun0
10.0.1.0	192.168.0.2	255.255.255.0	tun0

R2 Routing Table

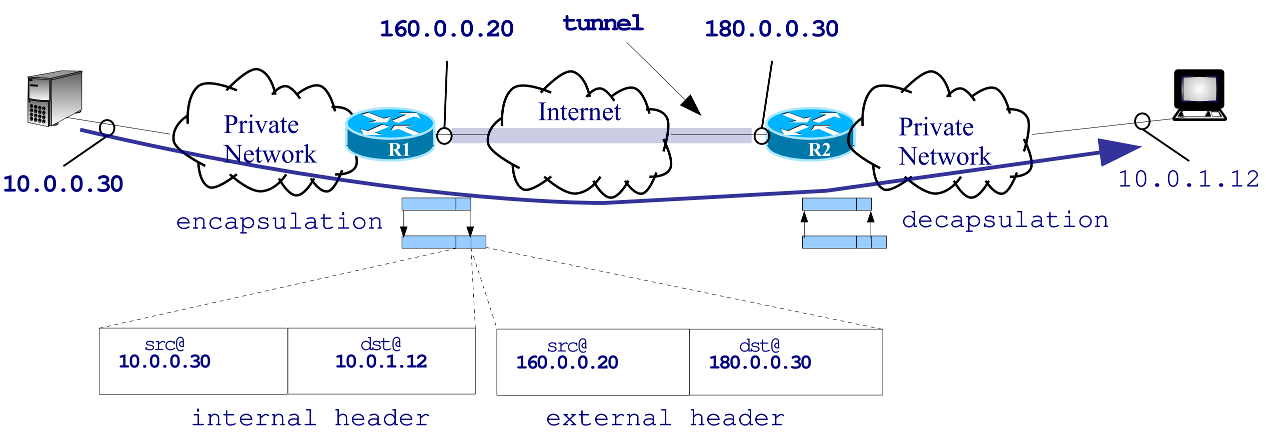
Destination	Gateway	Genmask	Iface
10.0.1.0	0.0.0.0	255.255.255.0	eth0
180.0.0.1	0.0.0.0	255.255.255.255	ppp0
0.0.0.0	180.0.0.1	0.0.0.0	ppp0
192.168.0.0	0.0.0.0	255.255.255.0	tun0
10.0.0.0	192.168.0.1	255.255.255.0	tun0

Note: In these routing tables the routes are not sorted according to the length of the netmask criterion.

Unit 2: IP Networks

Security in IP – VPN Tunneling Problems

- Fragmentation inside the tunnel will use the external header, thus, the exit router of the tunnel may reassemble fragmented datagrams.
- ICMP messages sent inside the tunnel are addressed to the tunnel entry.
- MTU path discovery may fail.
- Solution: the router entry maintains a "tunnel state", e.g. the tunnel MTU, and generate ICMP messages that would be generated inside the tunnel. Furthermore, the tunnel entry router typically fragment the datagrams, if needed, before encapsulation, to avoid the exit router having to reassemble fragmented datagrams.



Unit 2: IP Networks

Security in IP – VPN Tunneling

Types of tunnels:

- IP over IP (RFC 2003): Basic encapsulation.
- Generic Routing Encapsulation, GRE (RFC 1701): There is an additional GRE header: allows encapsulating other protocols (not only IP).
- Point-to-Point Tunneling Protocol, PPTP (RFC 2637): Add the ppp functionalities.
- IPsec (RFC 2401): Standards to introduce authentication and encryption and tunneling to IP layer.

Unit 2: IP Networks

Exercicis proposats

- Exàmens:
  - 2020t-c1: tot
  - 2021t-ef: Problema 1 fins a Apartat i) inclòs
  - 2021t-c1: tot
  - 2021p-c1: tot
  - 2021p-ef: Problema 1 tot
  - 2022t-c1: tot
- Col·lecció problemes:
  - Problema 1
  - Problema 3
  - Problema 5
  - Problema 6
  - Problema 7
  - Problema 8