

# 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

These slides are based on the set of slides provided by Llorenç Cerdà for this course. They include some modifications and some new slides.

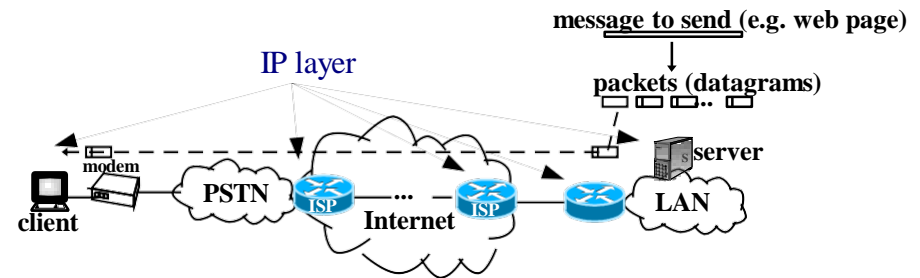
# Unit 2: IP Networks

## Outline

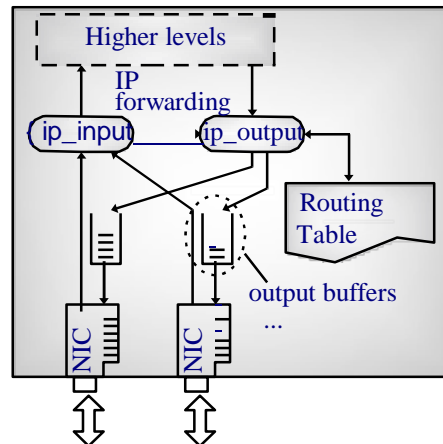
- **IP layer service**
- IP addresses
- Subnetting
- Routing tables
- ARP protocol
- IP header
- ICMP protocol
- DNS
- DHCP protocol
- NAT
- Routing algorithms
- Security in IP

# IP Layer Service

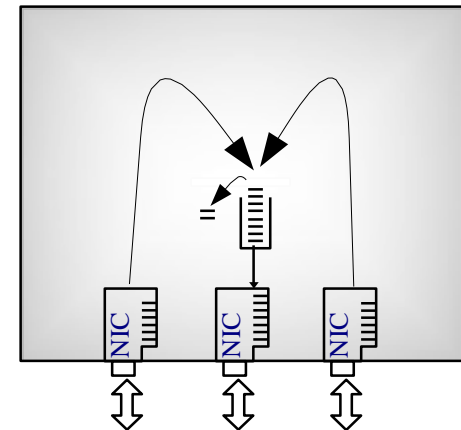
- Internet Protocol (IP) goal is **routing datagrams**.
- IP main design goal was interconnecting hosts attached to LANs/WANs **networks of different technologies**.
- IP characteristics:
  - **Connectionless**
  - **Stateless**
  - **Best effort**



Commercial routers



Basic router architecture



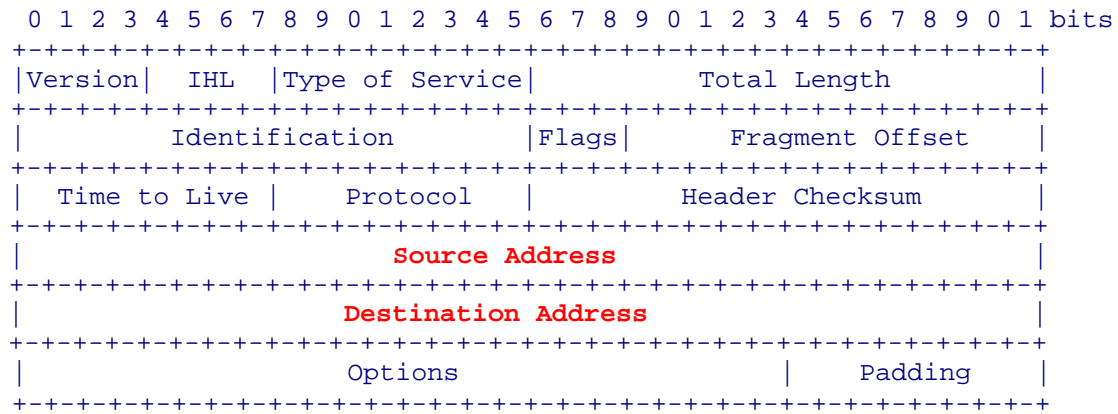
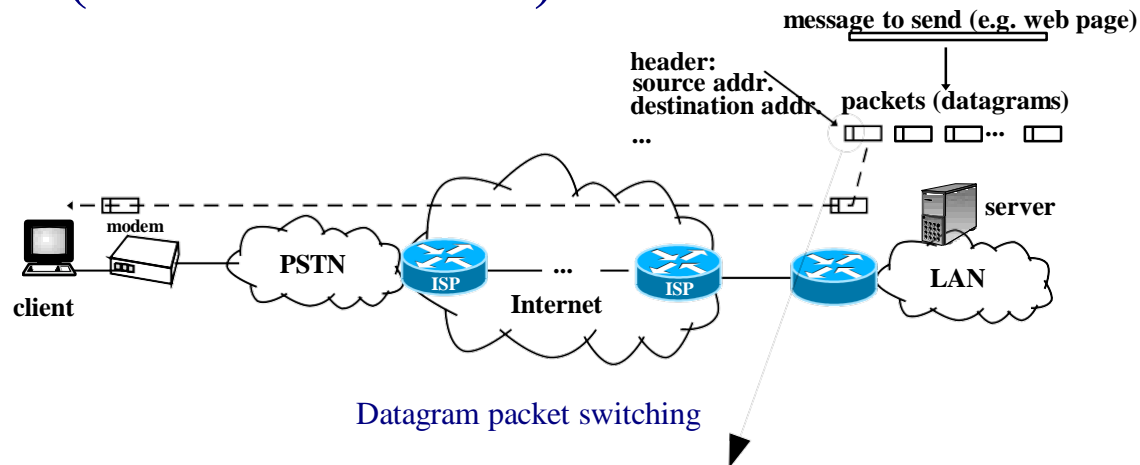
Looses may occur due to buffer overflow

# Unit 2: IP Networks

## Outline

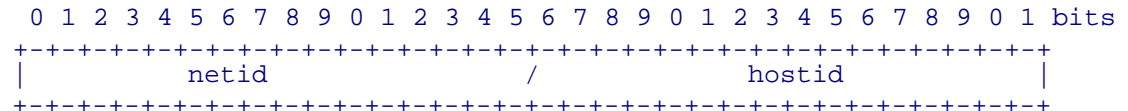
- IP layer service
- **IP addresses**
- Subnetting
- Routing tables
- ARP protocol
- IP header
- ICMP protocol
- DNS
- DHCP protocol
- NAT
- Routing algorithms
- Security in IP

# IP Addresses (RFC 791 Y1981)



IP datagram header

# IP Addresses



- 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: Africa, <http://www.afrinic.net>.

RIR assign addresses to ISPs, and ISPs to their customers.

# IP Addresses - Classes

- 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.

Classe	netid (bytes)	hostid (bytes)	Codification	range
A	1	3	0xxxx...x	0.0.0.0 ~ 127.255.255.255
B	2	2	10xxx...x	128.0.0.0 ~ 191.255.255.255
C	3	1	110xx...x	192.0.0.0 ~ 223.255.255.255
D	-	-	1110x...x	224.0.0.0 ~ 239.255.255.255
E	-	-	1111x...x	240.0.0.0 ~ 255.255.255.255

$2^7$  (128) class A networks with  $2^{24}$  addresses

$2^{14}$  (16.384) class B networks with  $2^{16}$  addresses

$2^{21}$  (2.097.152) class C networks with  $2^8$  addresses

$2^{28}$  class D (multicast) addresses

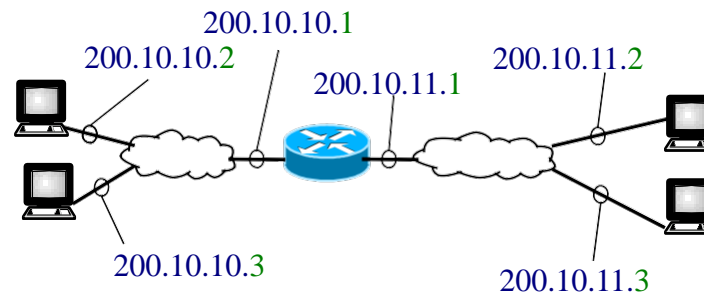
$2^{28}$  class E (reserved) addresses

# 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:**



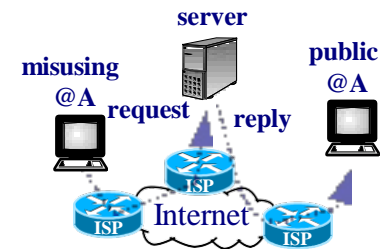


# IP Addresses – Private Addresses (RFC 1918 Y1996)

- 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



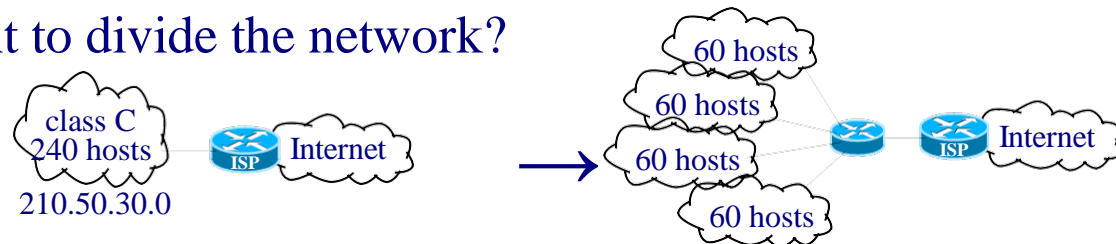
# Unit 2: IP Networks

## Outline

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

# Subnetting (RFC 950 Y1985)

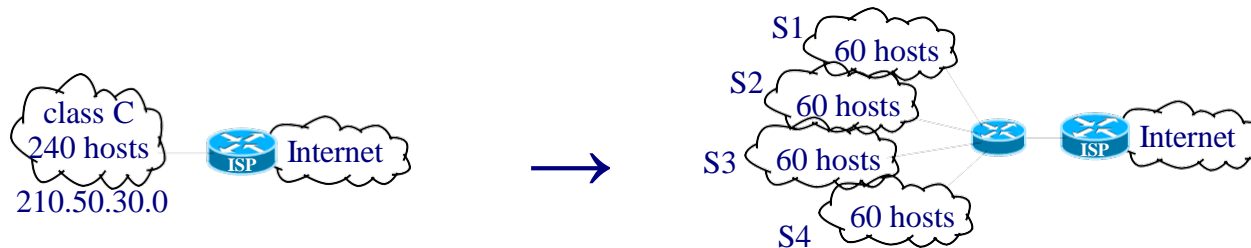
- Initially the netid was given by the address class: A with  $2^{24}$  addresses, B with  $2^{16}$  addresses and C with  $2^8$  addresses.
- What if we want to divide the network?



- 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** is used to identify the size of the netid+subnetid prefix.
- Mask **notations**:
  - dotted**, as 255.255.255.192
  - giving the **mask length** (number of bits) as 210.50.30.0/26

# IP Addresses – Subnetting Example

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



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	$2^6 - 2 = 62$

## IP Addresses – Variable Length Subnet Mask (VLSM)

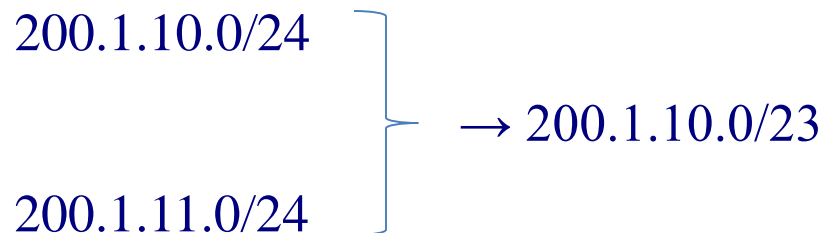
- Subnetworks of different sizes.
- Example, subnetting a class C address:
  - We have 1 byte for subnetid + hostid.
  - subnetid is green, chosen subnets addresses are underlined.

$$\begin{array}{l} \text{0000} \\ \hline \text{1000} \end{array} \rightarrow \begin{array}{l} \text{1000} \\ \hline \text{1100} \end{array} \rightarrow \begin{array}{l} \text{1100} \\ \hline \text{1101} \\ \hline \text{1110} \\ \hline \text{1111} \end{array}$$

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	$2^4 - 2 = 14$

## Classless Inter-Domain Routing, CIDR (RFC 1519 Y1993)

- Initially, Internet backbone routing tables did not use masks: netid was derived from the IP address class (default /8, /16, /24).
- 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 “**aggregate routes**”, and use masks instead of classes.
- Aggregation example:



- 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).

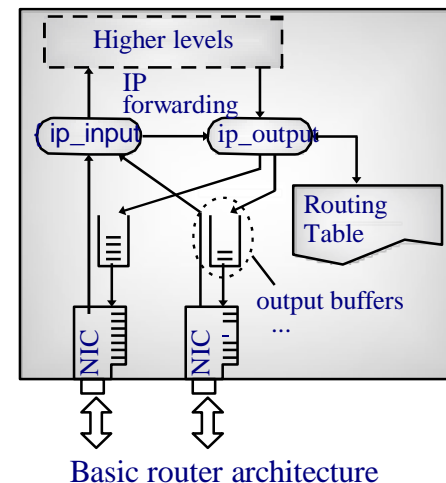
## Unit 2: IP Networks

### Outline

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

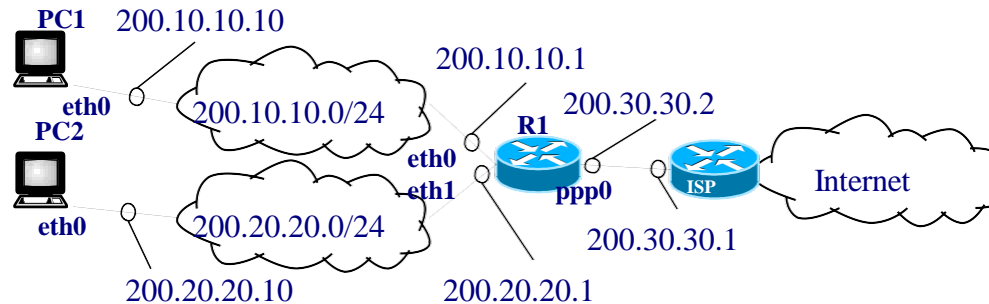
## Routing Table

- `ip_output()` kernel function consults 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 to and a default route.**





# Routing Table – Unix Example



known destinations

PC1 routing table:

Destination	Genmask
200.10.10.0	255.255.255.0
0.0.0.0	0.0.0.0

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
200.30.30.0	255.255.255.0	0.0.0.0	ppp0
0.0.0.0	0.0.0.0	200.30.30.1	ppp0

how to reach the destinations

# Routing Table – Datagram Delivery Algorithm

## 1. Check if the device itself is the destination:

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

## 2. Consult the routing table:

```
for each routing table entry ordered from longest to shortest mask  
(Longest Prefix Match) {  
    if((Datagram Destination IP address & mask) == Destination  
        table entry) {  
        return (gateway, interface) ;  
    }  
}
```

## 3. Forward the datagram

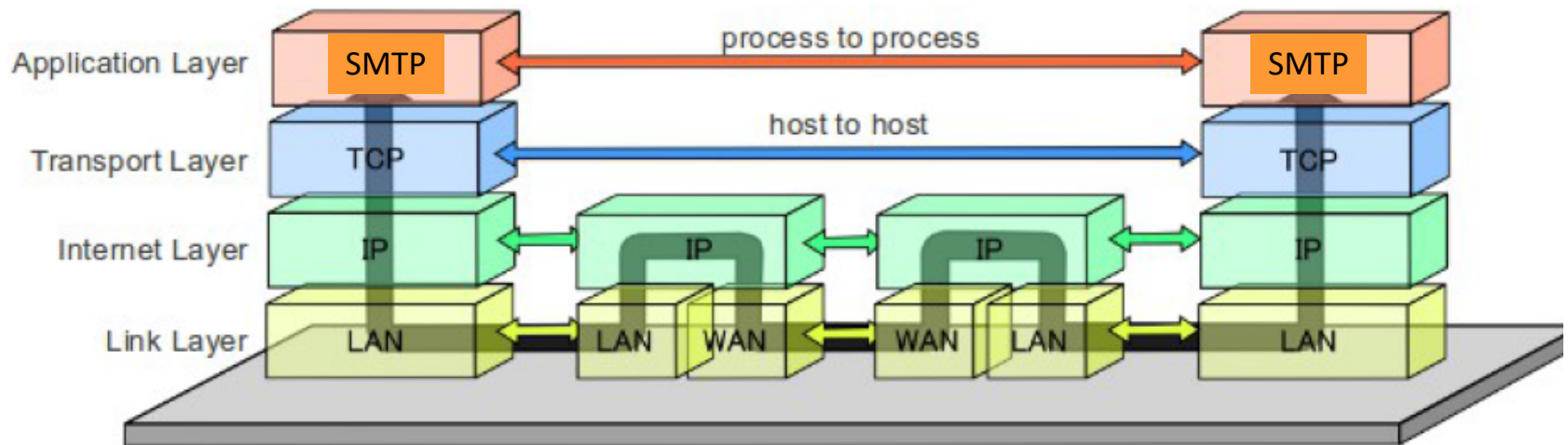
```
if(it is a direct routing) {  
    send the datagram to the Datagram Destination IP address  
} else { /* it is an indirect routing */  
    send the datagram to the gateway IP address  
}
```

# Unit 2: IP Networks

## Outline

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

## Data Flow of the Internet Protocol Suite



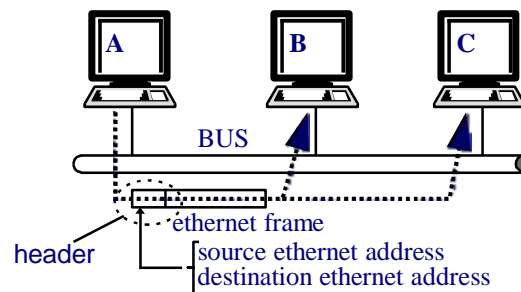
## Outgoing E-mail Frame

Destination MAC Address	Source MAC Address	Destination IP Address	Source IP Address	Destination TCP Port	Source TCP Port	
00:0C:78:52:F3:A5	0E:11:81:F2:C3:98	216.93.82.9	172.16.20.57	25	58631	Hi Mom 101101
MAC address of default gateway router's interface	Your NIC's MAC address	IP address of the SMTP server at your mom's ISP	IP address of your PC	Standard port number for SMTP	Randomly generated by your PC's TCP/IP stack	

# Address Resolution Protocol, ARP (RFC 826 Y1982)

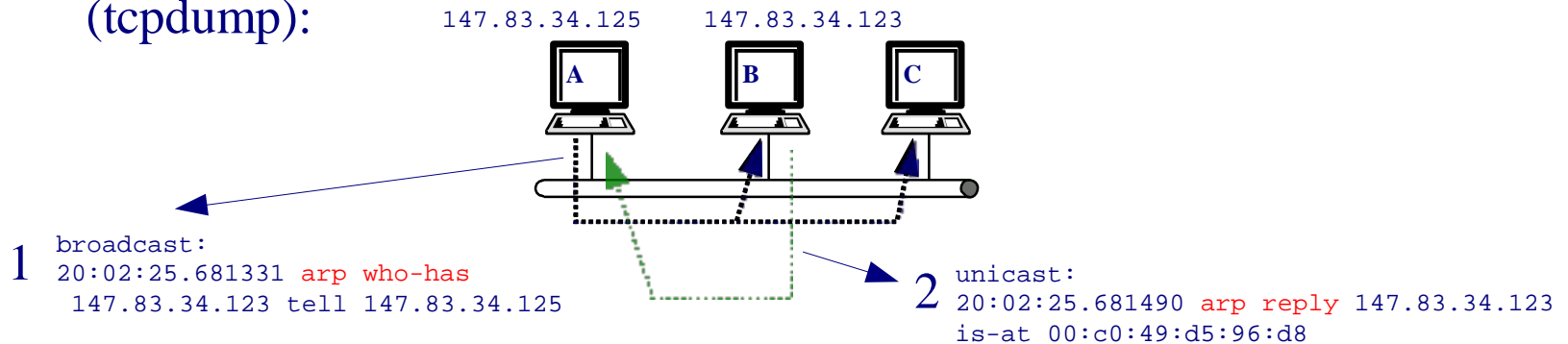
- To send the datagram, IP layer may have to pass a “**physical address**” to the NIC driver. Physical addresses are also called MAC or hardware addresses.
- **ARP** translate IP addresses to “**physical addresses**” (used by the physical network).
- If needed, **IP** calls **ARP** module to obtain the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

- Ethernet example:



# Address Resolution Protocol, messages - Example

- ARP messages (tcpdump):



- 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

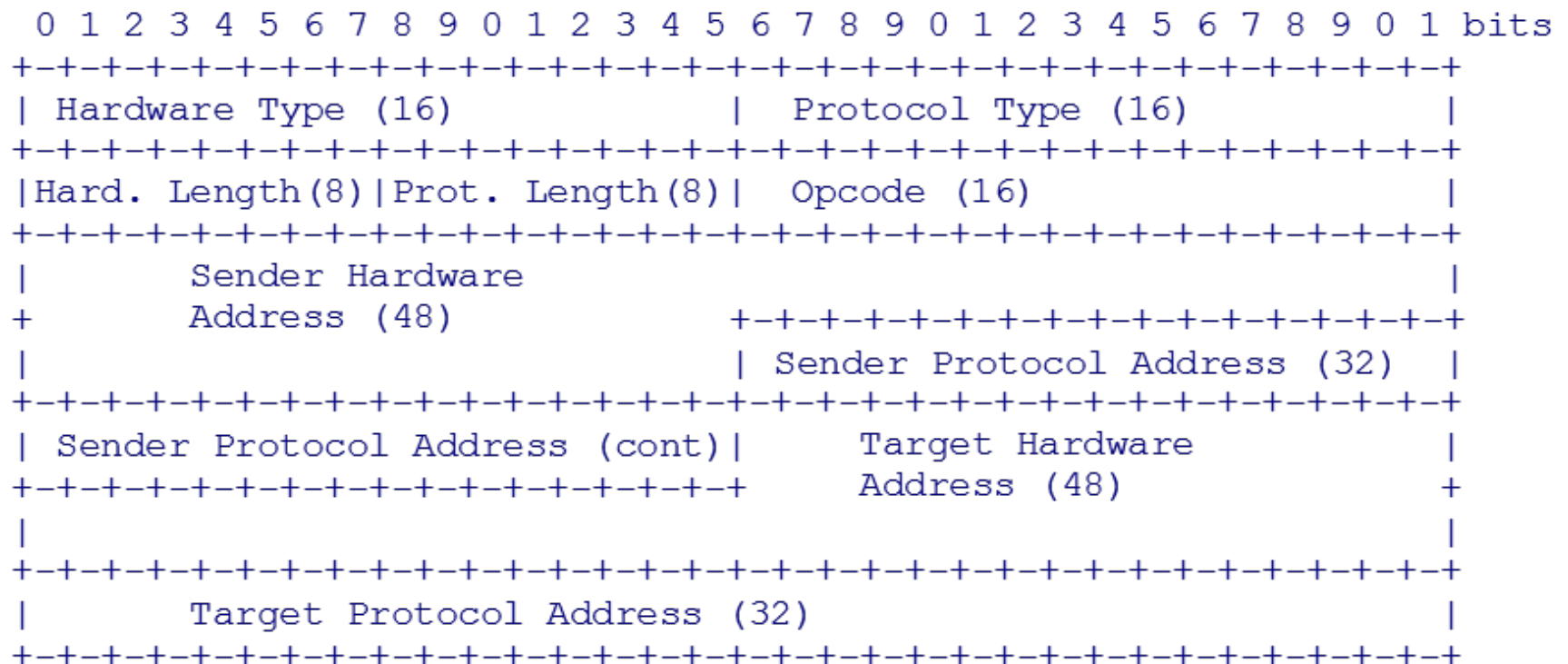
“Completed” flag

# Address Resolution Protocol

- **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.

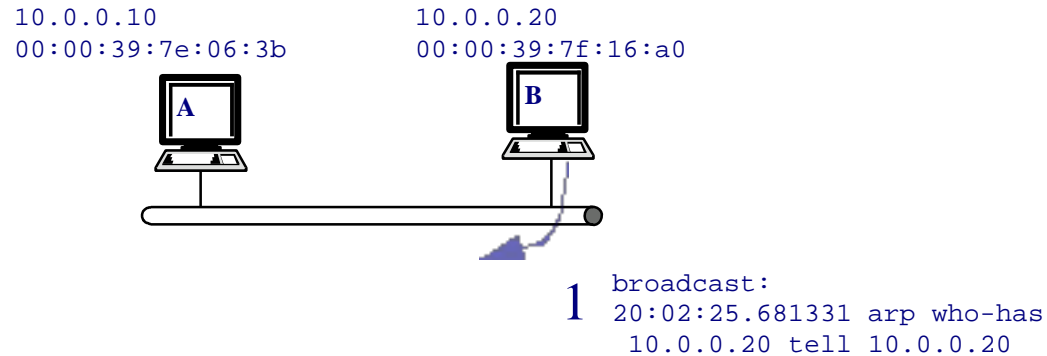
# Address Resolution Protocol – Message format (ethernet)

- ARP messages are encapsulated directly in a data-link frame.





# Address Resolution Protocol – Gratuitous ARP



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

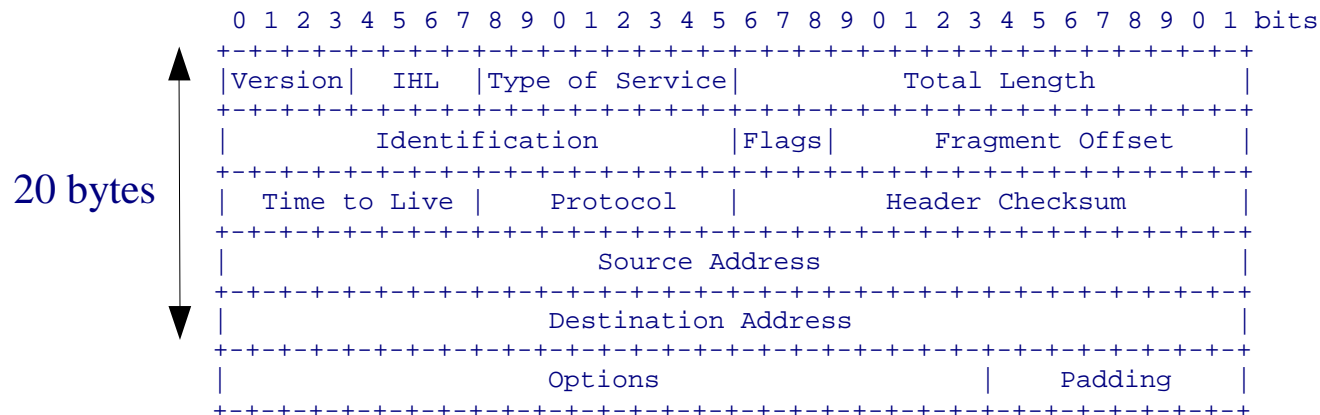
# Unit 2: IP Networks

## Outline

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

# IP Header (RFC 791 Y1981 updated by 1349 Y1992, 2474 Y1998, 6864 Y2013)

- **Version**: 4
- IP Header Length (**IHL**): Header size in 32 bit words.
- Type of Service: (**ToS**): *xxxdtrc0*.
- Total **Length**: Datagram size in bytes.
- **Identification/Flags/Fragment Offset**: used in fragmentation.
- Time to Live (**TTL**): if(--TTL==0) { discard ; }.
- **Protocol**: Encapsulated protocol (/etc/protocols in unix).
- Header **Checksum**: Header error detection.
- Source and Destination **Addresses**: End node addresses.
- **Options**: Record Route, Loose Source Routing, Strict Source Routing.



# IP Fragmentation

- Fragmentation may occur:
  - **Router**: Fragmentation may be needed when two networks with different *Maximum Transfer Unit (MTU)* are connected.
  - **Host**: Fragmentation may be needed using **UDP**. TCP segments are  $\leq$  MTU.
- 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 MTU path 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).

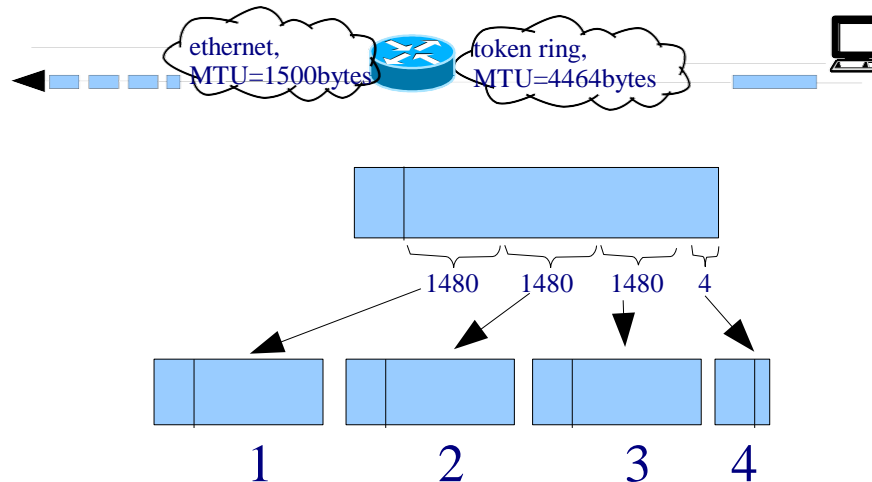


MTU examples:

- Ethernet: 1500
- WiFi: 2312
- Token Ring 4M: 4464
- FDDI: 4352
- PPP: 250
- X.25: 576

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



# Unit 2: IP Networks

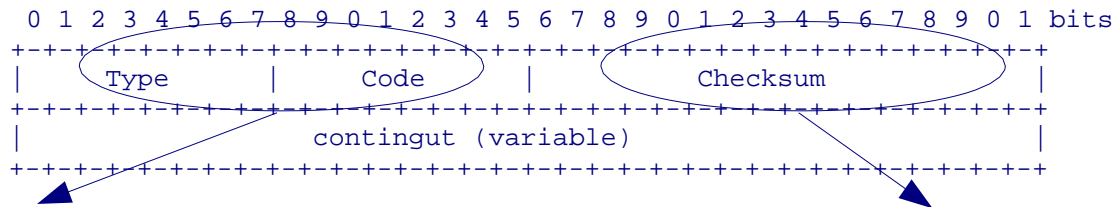
## Outline

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

# Internet Control Message Protocol, ICMP (RFC 792 Y1981)

- Used for attention and error messages.
- Can be **generated by** IP, TCP/UDP, and application layers. ICMP
- Messages are encapsulated into an IP datagram (protocol = 1)
- Messages can be: (i) **query**, (ii) **error**.
- An ICMP error message cannot generate another ICMP error message (to avoid loops).

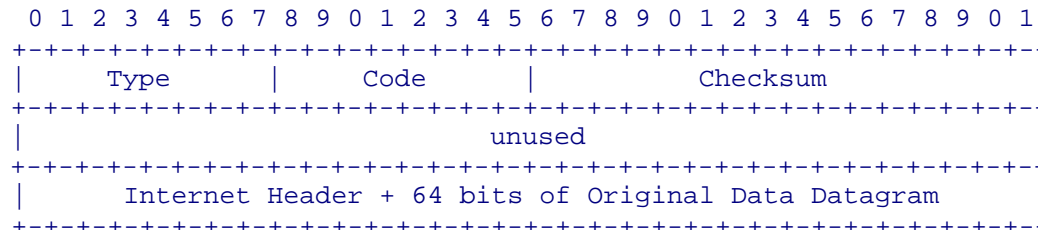
# ICMP general format message (RFC 792 Y1981)



Identifies the message

It 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**:



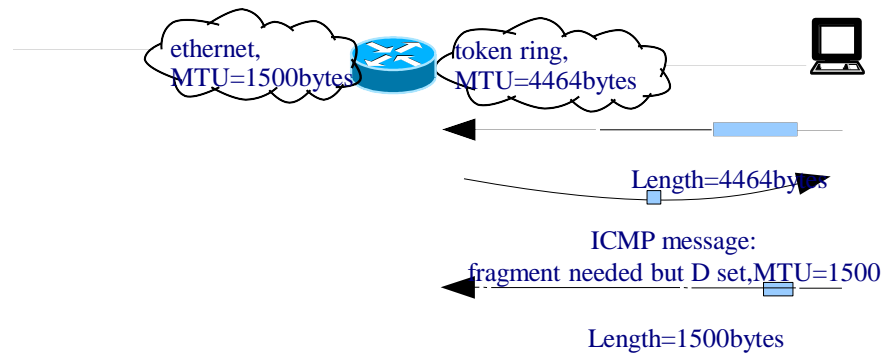


## Common ICMP messages

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 but 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 data-gram by the same interface it was received.
8	0	query	echo request	Request for reply
11	0	error	TTL=0 during transit	Sent by a router when --TTL=0

# MTU Path Discovery

- Used in modern **TCP** implementations.
- TCP by default chooses the maximum segment size, to avoid headers overhead (segment **efficiency** = TCP payload / (TCP payload +  $\Sigma$  TCP,IP,Data-link,Physical headers))
- 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

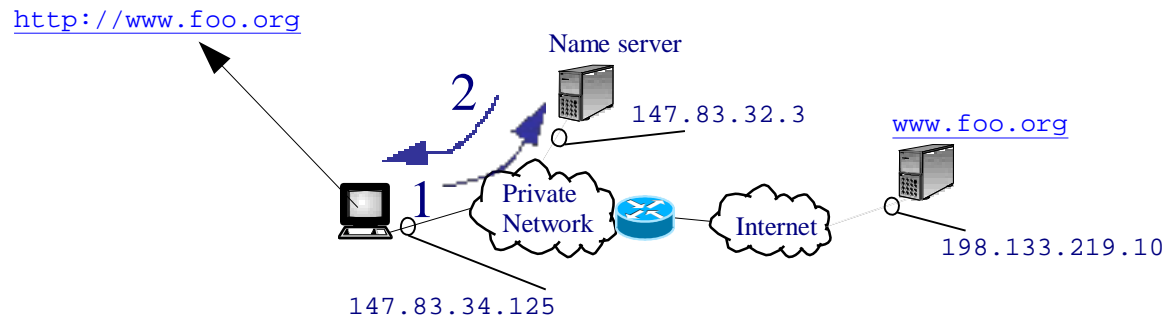
- IP layer service
- IP addresses
- Subnetting
- Routing tables
- ARP protocol
- IP header
- ICMP protocol
- DNS
- DHCP protocol
- NAT
- Routing algorithms
- Security in IP

# Domain Name System DNS (RFC 1034, 1035 Y1987)

- Allows users to use **names instead of IP addresses**: e.g. rogent.ac.upc.edu instead of 147.83.31.7, www.upc.edu instead of 147.83.194.21, etc.
- Names consists of a **node-name** and a **domain-name**:  
rogent.ac.upc.edu, www.upc.edu
- DNS consists of a **worldwide distributed data base**.
- DNS data base entries are referred to as *Resource Records (RR)*.
- The information associated with a name is composed of 1 or more RRs.
- Names are **case insensitive** (e.g. www.upc.edu and WWW.UPC.EDU are equivalent).

# DNS - Protocol

- Client-server paradigm
- UDP/TCP. For short messages it uses UDP.
- Well-known port: 53



```
1 18:36:00.322370 IP (proto: UDP) 147.83.34.125.1333 >
    147.83.32.3.53: 53040+ A? www.foo.org. (31)
2 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
- DNS
- **DHCP protocol**
- NAT
- Routing algorithms
- Security in IP

# Dynamic Host Configuration Protocol, DHCP (RFC 2131 Y1997)

- Improves and can interoperate with previous **BOOTP** protocol.
- Used for **automatic network configuration**:
  - Assign IP address and mask
  - Default route
  - Configure DNS servers
  - Hostname
  - DNS domain
- **IP address configuration** can be:
  - Dynamic: During a leasing time.
  - Automatic: Unlimited leasing time.
  - Manual: IP addresses are assigned to specific MAC addresses.

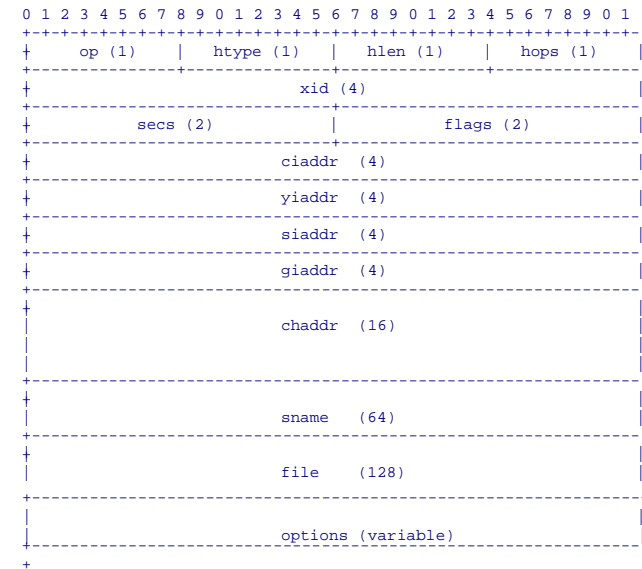
# DHCP – Protocol Messages (RFC 2131 Y1997)

- DHCPDISCOVER** - Client broadcast to locate available servers.
- DHCOFFER** - Server to client in response to DHCPDISCOVER with offer of configuration parameters.
- 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.
- DHCPACK** - Server to client with configuration parameters, including committed network address.
- 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
- DHCPDECLINE** - Client to server indicating network address is already in use.
- DHCPRELEASE** - Client to server relinquishing network address and cancelling remaining lease.
- DHCPINFORM** - Client to server, asking only for local configuration parameters; client already has externally configured network address.



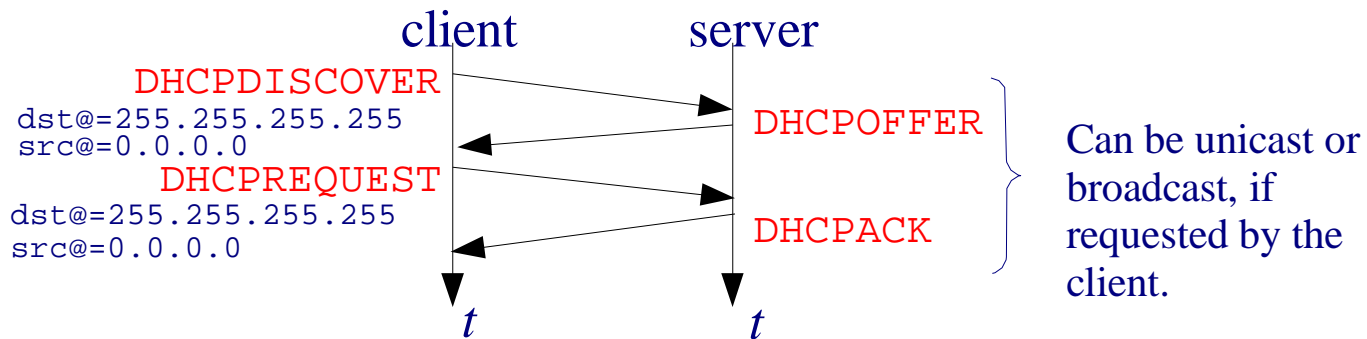
# DHCP – Message Fields (RFC 2131)

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.
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 DHCP OFFER message.
siaddr	4	IP address of next server to use in bootstrap; returned in DHCP OFFER, 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 DHCP OFFER.
options	var	Optional parameters field.

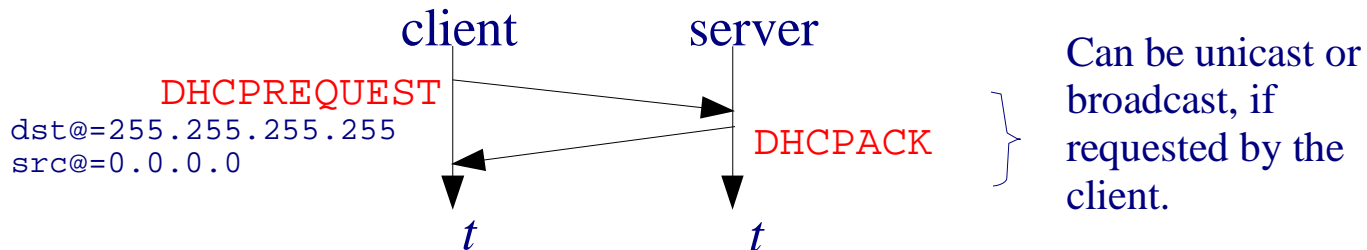


## DHCP – Client-server interaction (RFC 2131)

- 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.



# 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:78:00:00:00:00:00:00:00:00:00:00:00:00:00:00
OPTION: 53 ( 1) DHCP message type 3 (DHCPPREQUEST)
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:78:00:00:00:00:00:00:00:00:00:00:00:00:00:00
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
```

---

### Configuración DHCP IPv4

Dirección IPv4 (LOCAL):

192.168.1.1

Máscara de subred

255.255.255.0

### DHCP

Estado

ACTIVADO

Dirección IPv4 inicio rango

192.168.1.33

Dirección IP fin de rango

192.168.1.199

### Configuración de servidores DNS (se recomienda no modificar)

Servidor DNS 1

80.58.61.250

Servidor DNS 2

80.58.61.254

### LAN IP Setup

IP Address :

Subnet Mask :

RIP Version :  Direction :

Multicast :

IGMP Snooping : ☐ Disabled ☒ Enabled

IGMP Quickleave : ☐ Disabled ☒ Enabled

### DHCP Server State

DHCP : ☐ Disable ☒ Enable ☐ DHCP Relay

### IP Addressing Values

IP Pool Starting Address :

Pool Size :

### DHCP Conditional Serving Pool

State : ☐ Disabled ☒ Enabled

Gateway :

Subnet Mask :

Pool Start :

Pool End :

DNS Server 1 :

DNS Server 2 :

VendorID :

VendorID Mode : ☐ Exact ☐ Prefix ☐ Suffix ☒ Substring

VendorID Exclude : ☒ Disabled ☐ Enabled

Option240 State: ☐ Disabled ☒ Enabled

Option240 Value:

### DHCP Server Lease Time

Lease Time :  seconds

### DNS Values

DNS Server 1 :

DNS Server 2 :

Detalles de la conexión de red:

Propiedad	Valor
Sufijo DNS específico para...	Home
Descripción	D-Link DWA-171 AC600 MU-MIMO Wi-Fi USE
Dirección física	C4-E9-0A-07-21-3A
Habilitado para DHCP	Sí
Dirección IPv4	192.168.1.69
Máscara de subred IPv4	255.255.255.0
Concesión obtenida	jueves, 04 de marzo de 2021 11:47:34
La concesión expira	viernes, 05 de marzo de 2021 23:38:33
Puerta de enlace predeter...	192.168.1.1
Servidor DHCP IPv4	192.168.1.1
Servidores DNS IPv4	80.58.61.250 80.58.61.254
Servidor WINS IPv4	192.168.1.1
Habilitado para NetBios a t...	Sí
Vínculo: dirección IPv6 local	fe80::a023:3477:5ab2:81b9%12
Puerta de enlace predeter...	
Servidor DNS IPv6	

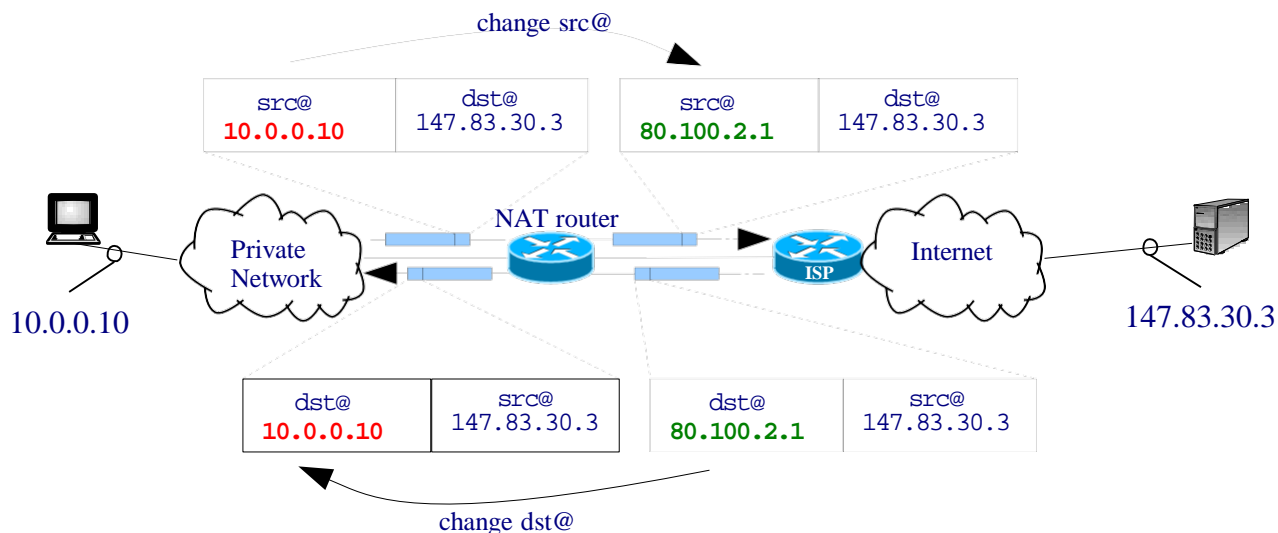
# Unit 2: IP Networks

## Outline

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

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

- Typical scenario: Private addresses (internal addresses) are translated to public addresses (external addresses).
- A NAT table is used for address mapping.
- **Advantages:**
  - Save public addresses.
  - Security.
  - Administration, e.g. changing ISP does not imply changing private network addressing.

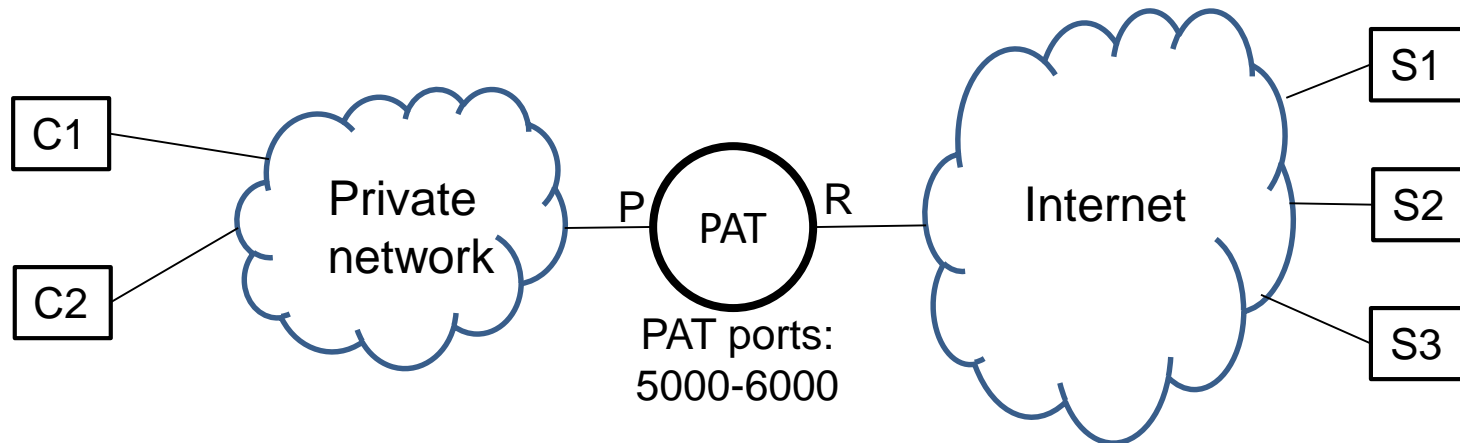




# NAT – Types of translations

- NOTE: NAT is a technique, not a protocol. Implementations and terminology may change from one manufacturer to another.
- **Basic NAT:**
  - 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.
- **Port and Address Translation, PAT (PNAT, NAPT):**
  - The same external address can be used for each internal address → a unique public IP address can be used for all hosts accessing Internet.
  - Each NAT table entry has the tuple: (int. address/port, ext. address/port)
  - Each connection requires one NAT table entry.
- The NAT table **entries** can be:
  - Static: Manually added.
  - Dynamic:
    - Entries are automatically added when an internal connection is initiated.
    - External addresses are chosen from a pool.
    - Table entries have a timeout.

## Port and Address Translation (PAT)



C1, C2, P are private IP addresses

R, S1, S2, S3 are public IP addresses

connection			
src	dst		
C1	P1	S1	80
S1	80	C1	P1
C1	P2	S1	53
S1	53	C1	P2
C2	P1	S1	80
S1	80	C2	P1
C2	P2	S2	80
S2	80	C2	P2

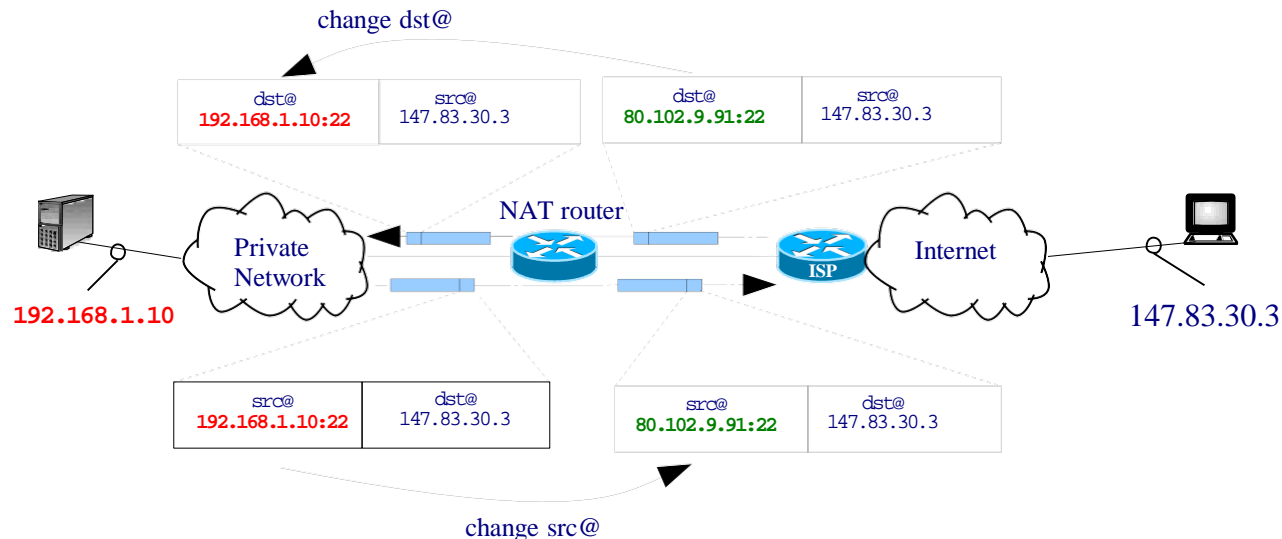
Inside		Outside		Foreign/remote	
C1	P1	R	5000	S1	80
C1	P2	R	5001	S1	53
C2	P1	R	5002	S1	80
C2	P2	R	5003	S2	80

connection			
src	dst		
R	5000	S1	80
S1	80	R	5000
R	5001	S1	53
S1	53	R	5001
R	5002	S1	80
S1	80	R	5002
R	5003	S2	80
S2	80	R	5003

# DNAT (Destination NAT)

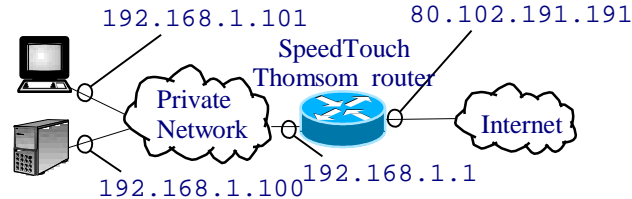
- What if we want external connections to internal servers? (DNAT in linux-iptables terminology).
- The address translation is exactly the same as NAT, but, the connection is initiated from an external client.
- Typically, some **static configuration** is needed to configure the server IP/port.

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



## NAT – ADSL commercial router example

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



```
linux # telnet 192.168.1.1
Trying 192.168.0.1...
Connected to 192.168.1.1.
=>nat
```

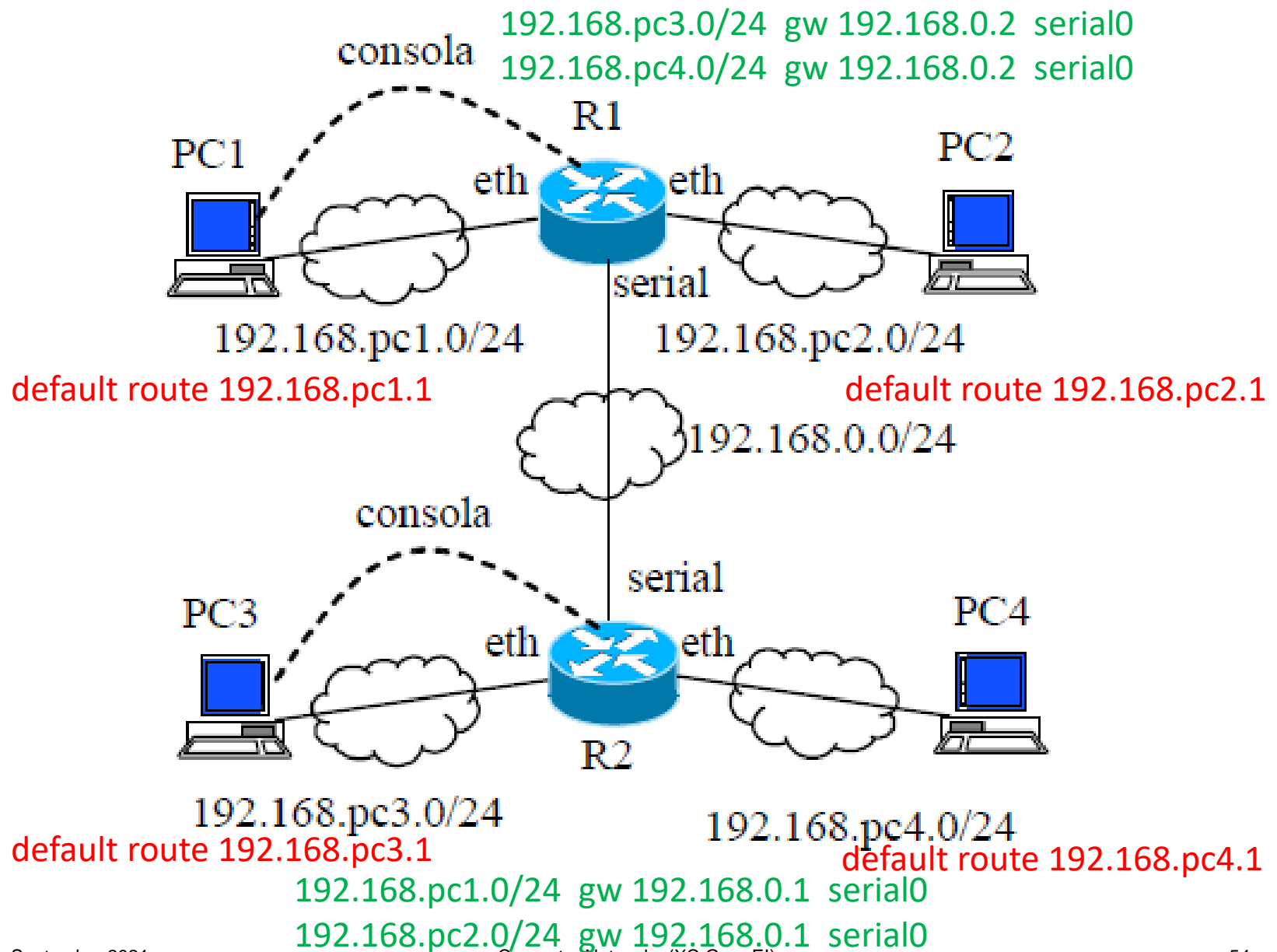
```
[nat]=>list
```

	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	
	...								

# Unit 2: IP Networks

## Outline

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

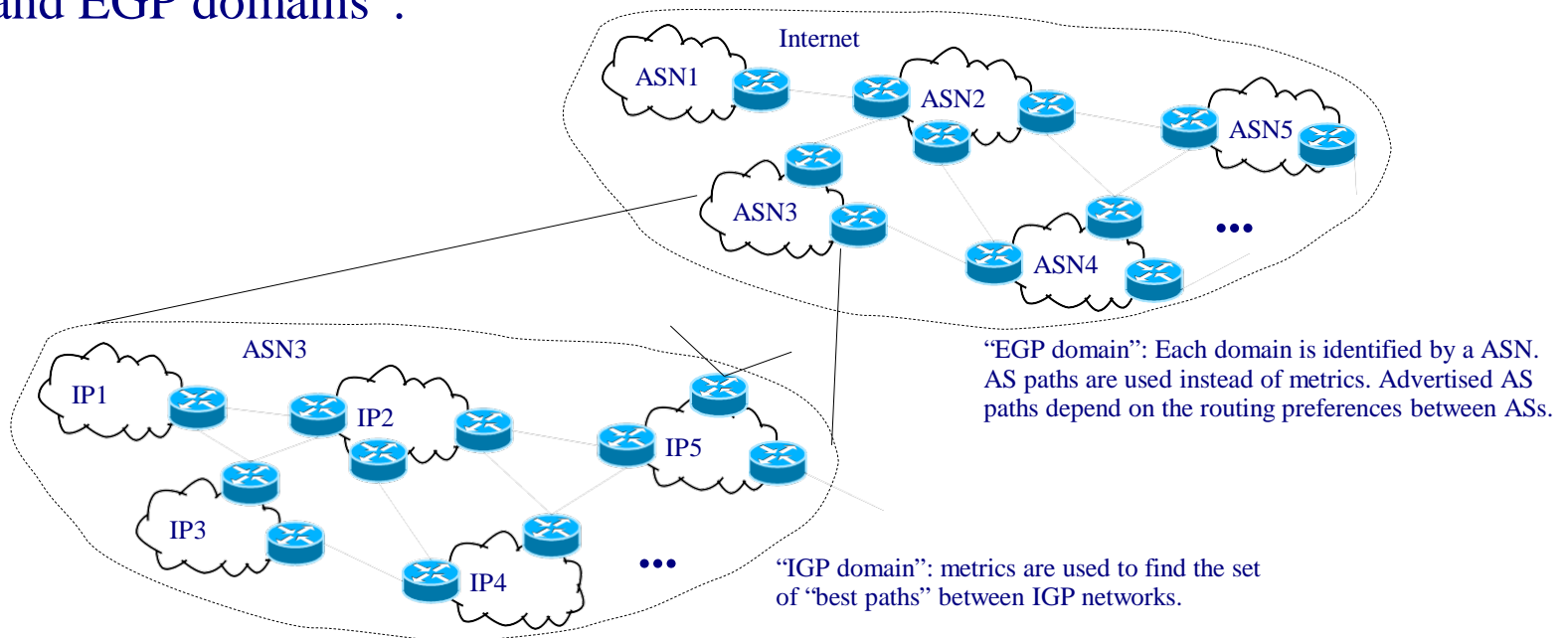


# Routing algorithms

- Objective: add entries to routing tables:
  - **Static**: Manual, scripts, DHCP.
  - **Dynamic**: Automatically update table entries, e.g. when a topology change occurs.
- 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.

# 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 bits** AS Number (ASN) assigned by IANA.
- ASs facilitate Internet routing by introducing a two-level hierarchy: “IGP and EGP domains”.



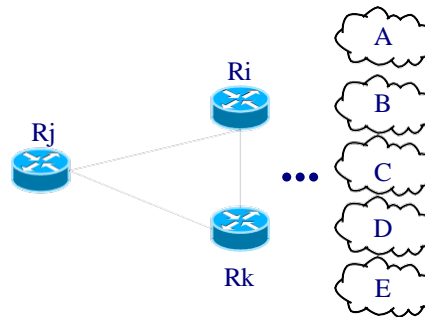


# Routing Information Protocol, RIP (RFC 2453 RIPv2 Y1998)

- 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 ...
- Routers send **RIP updates** every 30 seconds to the neighbors.
- RIP updates use **UDP**, **well-known port** = 520, to broadcast IP address.
- RIP updates include **destinations** and **metrics** tuples.
- A neighbor is considered down if no RIP messages are seen during **180 seconds**.
- **Infinite metric** is **16**.
- **RIP Version 2** allows variable masks and uses the multicast destination address 224.0.0.9 (all RIPv2 routers).
- The routing algorithm is known as “distance-vector” or “Bellman-Ford algorithm”.

# RIP – Routing Table (RT) Update Example

- Example: When **Ri** receives an update message from **Rj**:
  - Increase the message metrics.
  - Add new destinations.
  - Change entries with other routers with larger metrics.
  - Update metrics using Rj's gateway.



<u>D</u>	<u>G</u>	<u>M</u>
A	Rk	4
B	Rj	3
C	Rk	5
D	Rj	2

Ri's RT



<u>D</u>	<u>M</u>
A	1
B	4
C	5
D	1
E	3

Ri receives  
Rj's update  
message



<u>D</u>	<u>M</u>
A	2
B	5
C	6
D	2
E	4

Rj's metrics  
increased



<u>D</u>	<u>G</u>	<u>M</u>
A	Rj	2
B	Rj	5
C	Rk	5
D	Rj	2
E	Rj	4

Ri's RT  
updated

# RIP – Count to Infinity

- Depending on the route update message order, **convergence problems** may arise:



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

R1's RT

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

R2's RT

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

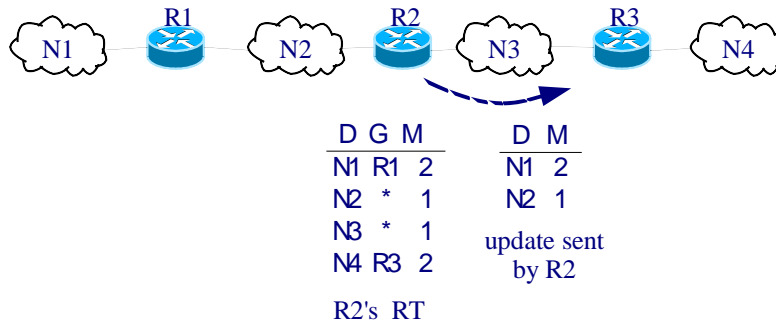
R3's RT

- Example: Evolution of **D=N4** entry when **R3 fails**:

	G	M	R3 fails	G	M	R1 upd	G	M	R2 upd	G	M	R1 upd	G	M	...	G	M
R1:	R2	3	→	R2	3	→	R2	3	→	R2	5	→	R2	5	...	R2	16
R2:	R3	2	→	R3	16	→	R1	4	→	R1	4	→	R1	6	...	R1	16

## 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:



- Split horizon with **Poisoned Reverse**: Consists of adding the entries having a gateway with metric  $M=16$ .
- **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 280 seconds. During this time, the entry is not updated.

# Open Shortest Path First, OSPF (RFC 2328, OSPF v2 Y1998)

- 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

## Outline

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

# 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, ...

## Security in IP – Attacks

- **Detection**: 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 replicates itself.

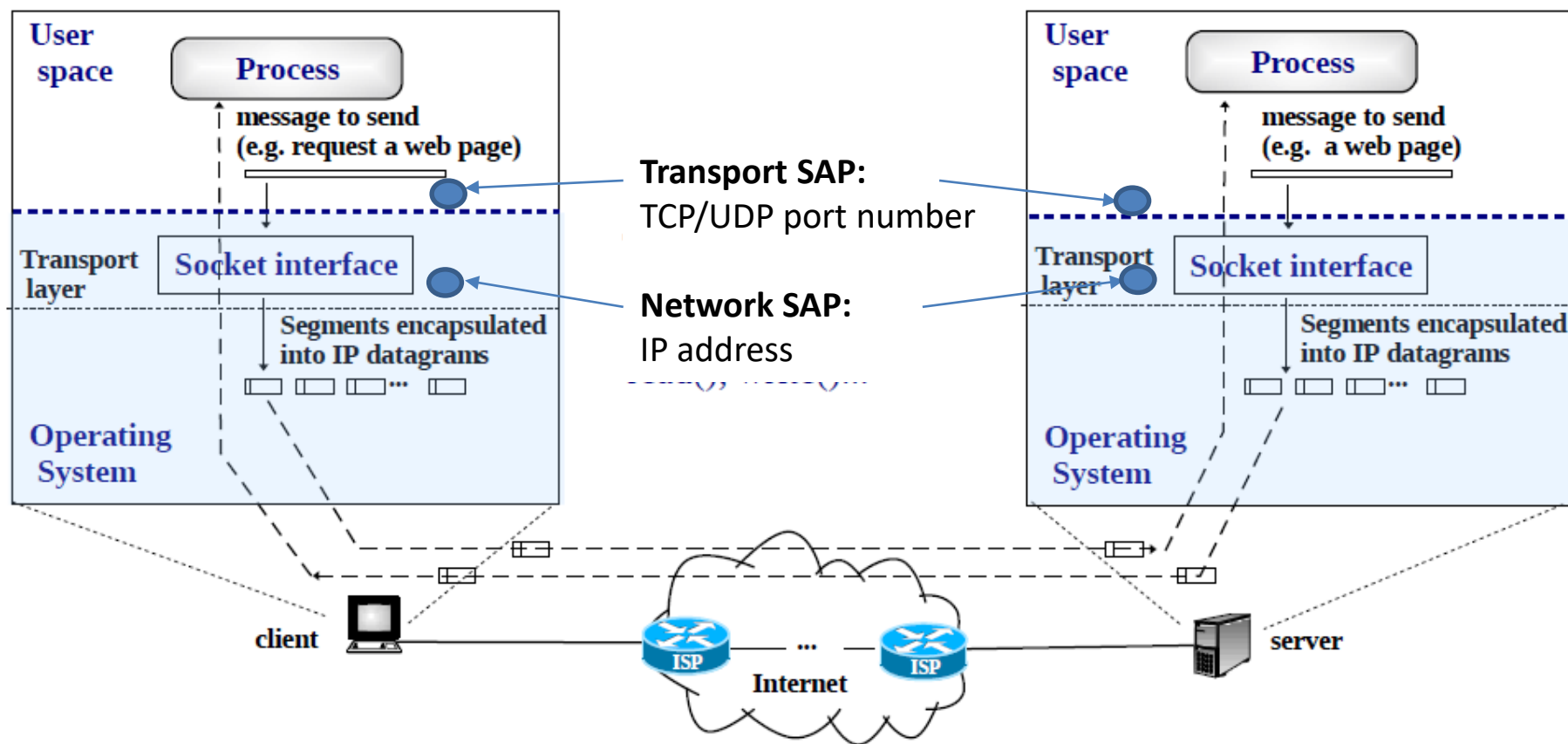
## Security in IP – Basic Solutions

- **Firewalls**.
- Virtual Private Networks (**VPN**).



## An end-to-end connection is univocally defined by:

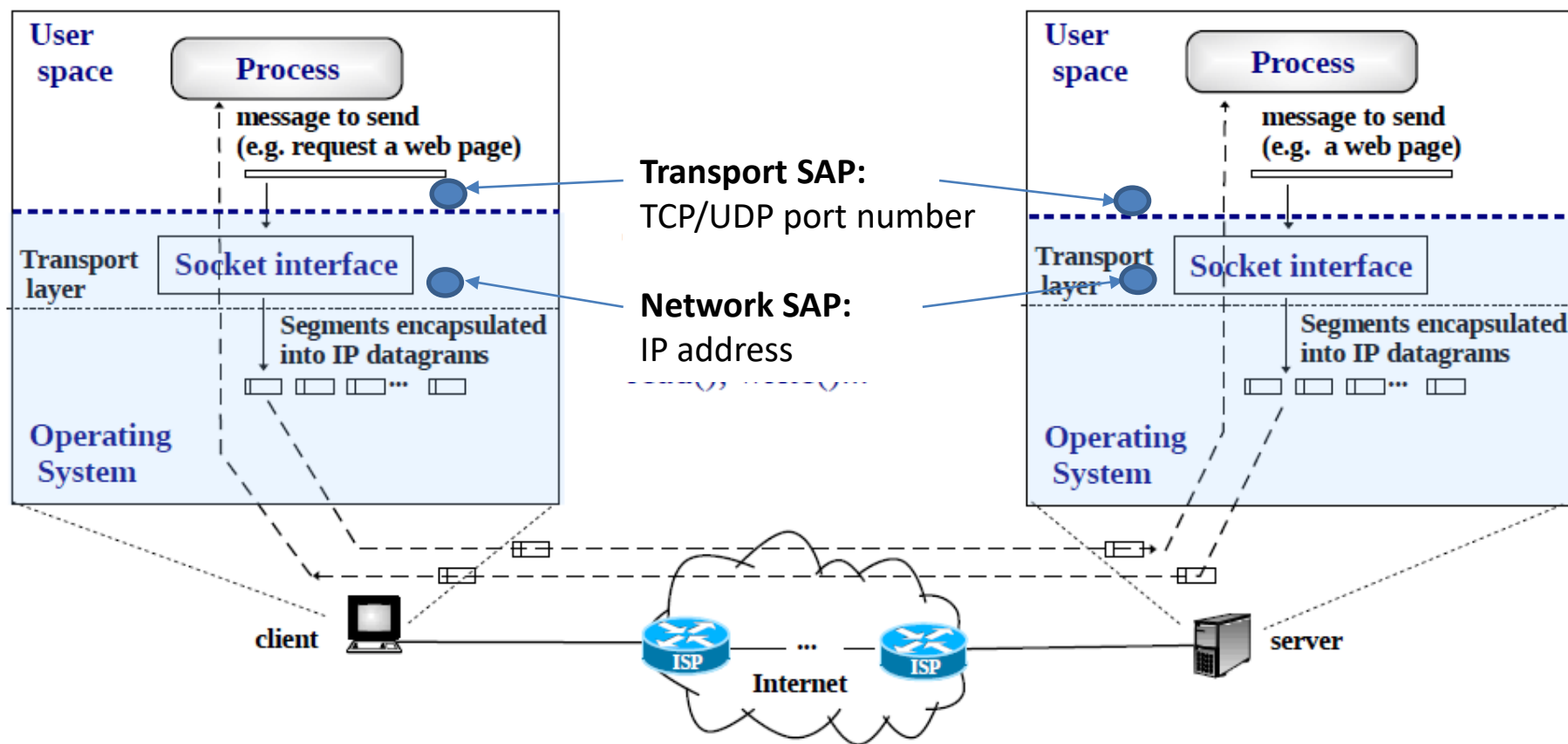
source IP address (source device),  
destination IP address (destination device),  
source port number (application at source device),  
destination port number (application at destination device) and  
the protocol used in the communication



## An end-to-end connection is univocally defined by:

source IP address (source device),  
destination IP address (destination device),  
source port number (application at source device),  
destination port number (application at destination device) and  
the protocol used in the communication

Header  
IP  
IP  
TCP/UDP  
TCP/UDP  
IP



# Example

## **Connection between an email client and an email server:**

IP of the client device

Transport Protocol: TCP

Port number assigned to the email client by the client device

IP of the email server

Standard port number for email (SMTP): 25

**The set of five fields identify a connection (flow)**

# Example

## Client to Server

### Connection between an email client and an email server:

IP of the client device

Source IP header

Transport Protocol: TCP

Source IP header

Port number assigned to the email client by the email client

Source TCP header

IP of the email server

Destination IP header

Standard port number for email (SMTP): 25

Destination TCP  
header

**The set of five fields identify a connection (flow)**

# Example

## Server to Client

### Connection between an email client and an email server:

IP of the client device  
Transport Protocol: TCP  
Port number assigned by the email client

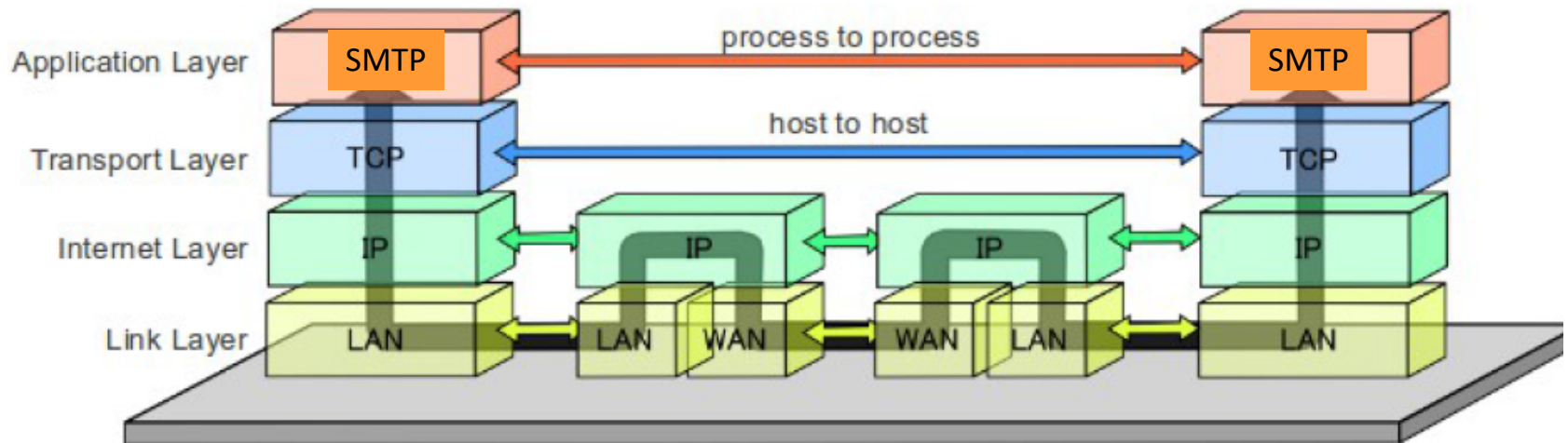
Destination IP header  
Destination IP header  
Destination TCP header

IP of the email server  
Standard port number for email (SMTP): 25

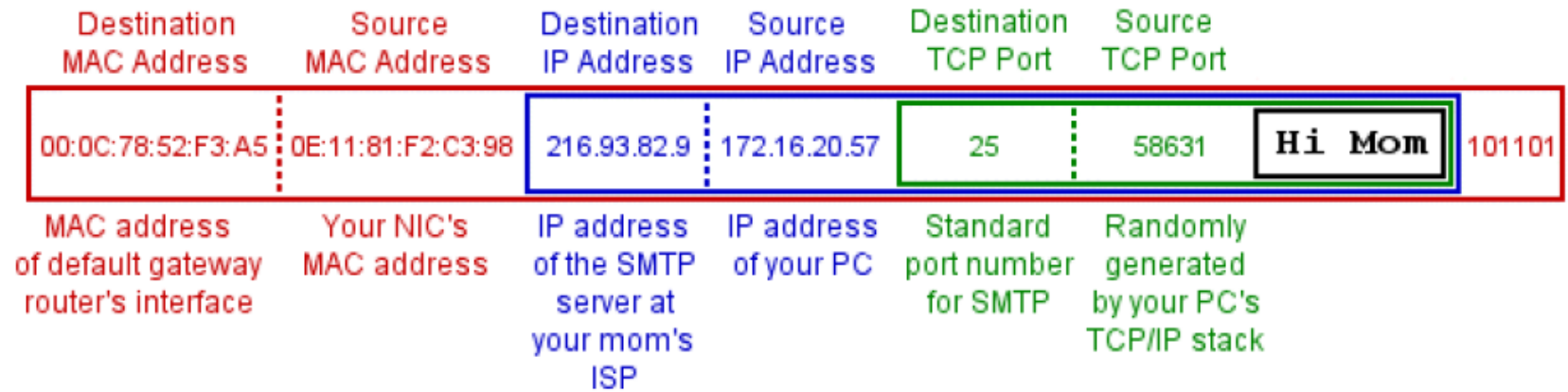
Source IP header  
Source TCP header

**The set of five fields identify a connection (flow)**

## Data Flow of the Internet Protocol Suite

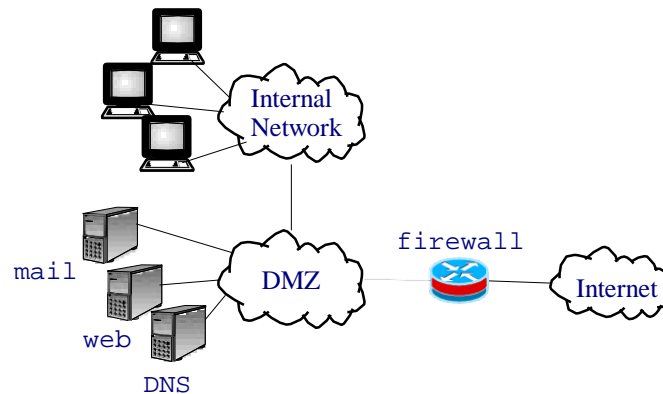


## Outgoing E-mail Frame



# Security in IP – Firewalls

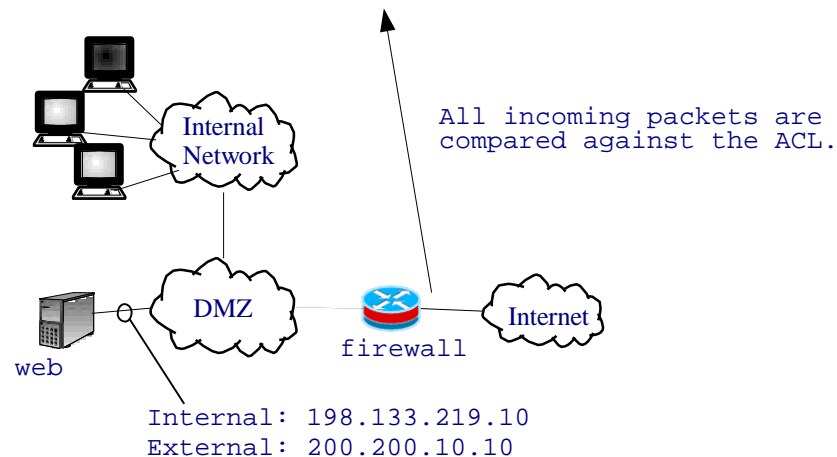
- **Firewall**: System or group of systems that enforces an access control policy to a network.
- There are many **firewall types**: From simple packet filtering based on IP/TCP/UDP header rules, to state-full connection tracking and application-based filtering, defense against network attacks, ...



# Security in IP – Basic Firewall Configuration

- Access Control List, **ACL**

Protocol	IP-src	IP-dst	Port-src	Port-dst	Action
TCP	<i>any</i>	200.200.10.10/32	<i>any</i>	80	<i>accept</i>
TCP	<i>any</i>	<i>any</i>	< 1024	$\geq$ 1024	<i>accept</i>
ICMP	<i>any</i>	<i>any</i>	–	–	<i>accept</i>
IP	<i>any</i>	<i>any</i>	–	–	<i>deny</i>

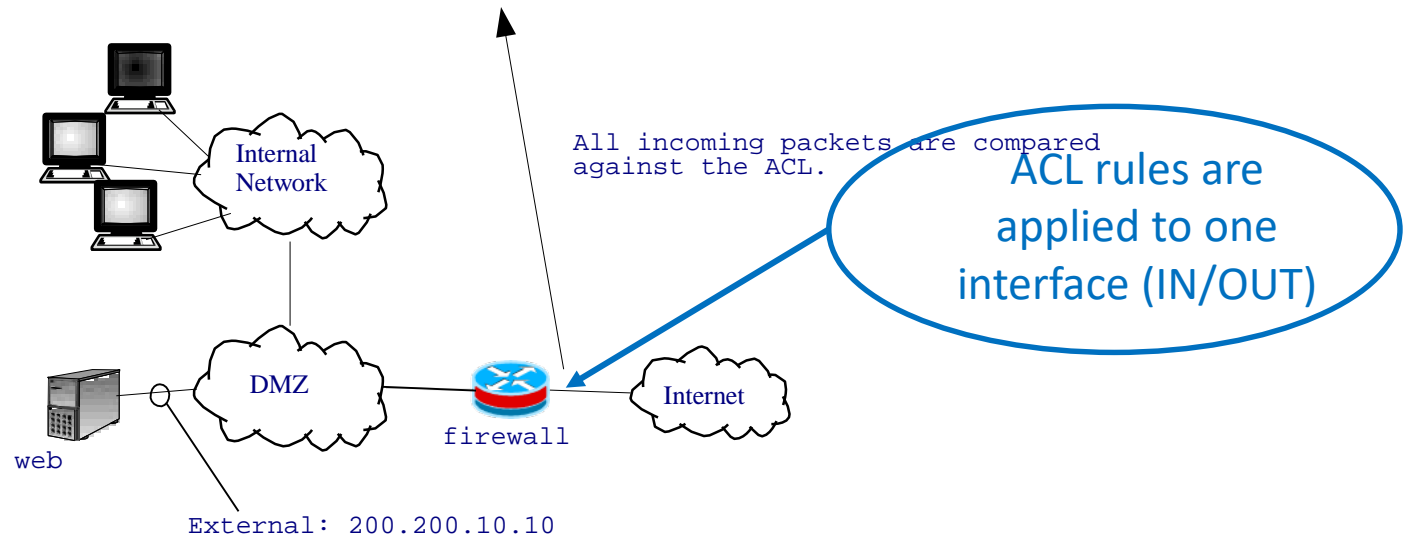




# Security in IP/TCP/UDP – Basic Firewall Configuration

- Access Control List, **ACL**

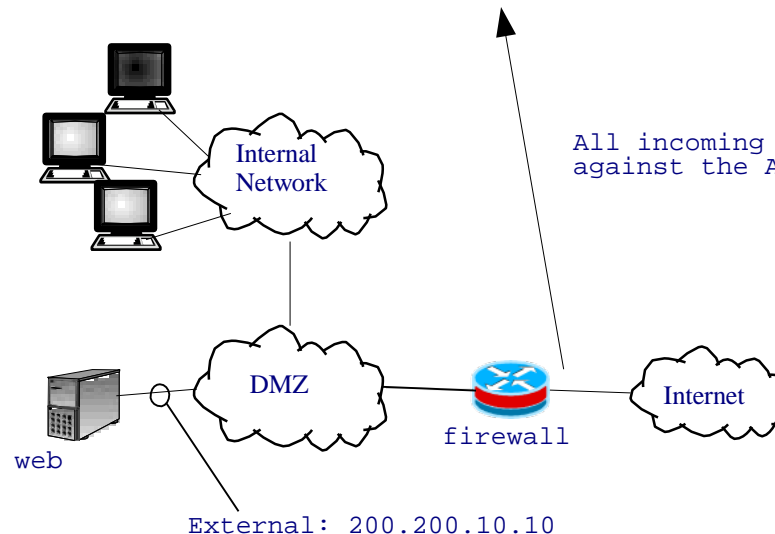
Protocol	IP-src	IP-dst	Port-src	Port-dst	Action
TCP	<i>any</i>	200.200.10.10/32	<i>any</i>	80	<i>accept</i>
TCP	<i>any</i>	<i>any</i>	< 1024	$\geq 1024$	<i>accept</i>
ICMP	<i>any</i>	<i>any</i>	–	–	<i>accept</i>
IP	<i>any</i>	<i>any</i>	–	–	<i>deny</i>



# Security in IP/TCP/UDP – Basic Firewall Configuration

- Access Control List, **ACL**

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

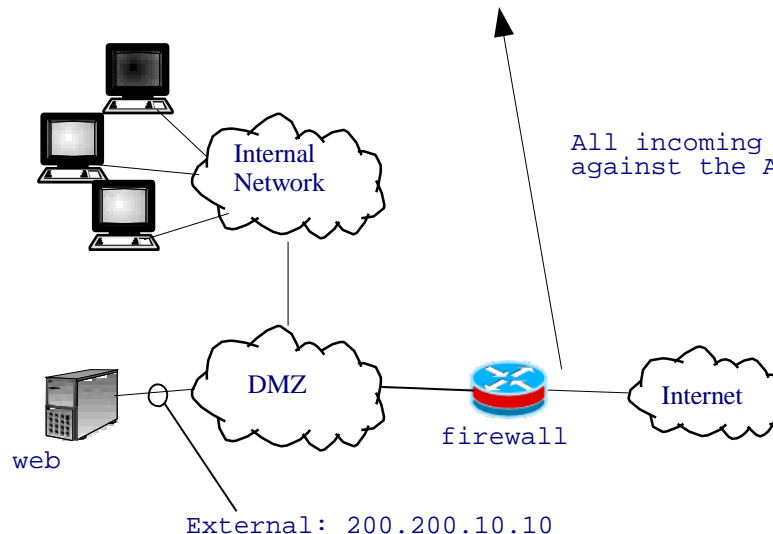


**ONLY the Web server  
accepts connections from  
an external web client**

# Security in IP/TCP/UDP – Basic Firewall Configuration

- NAT
- Access Control List, **ACL**

Protocol	IP-src	IP-dst	Port-src	Port-dst	Action
TCP	<i>any</i>	200.200.10.10/32	<i>any</i>	80	<i>accept</i>
<b>TCP</b>	<b><i>any</i></b>	<b><i>any</i></b>	<b>&lt; 1024</b>	<b>≥ 1024</b>	<i>accept</i>
ICMP	<i>any</i>	<i>any</i>	–	–	<i>accept</i>
IP	<i>any</i>	<i>any</i>	–	–	<i>deny</i>



incoming

**FROM**

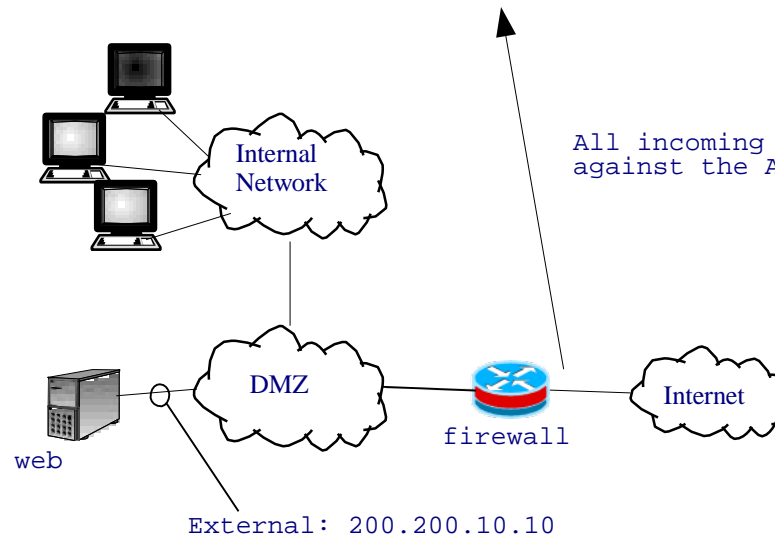
**TO**

**ANY internal client accepts connections from any external server**

# Security in IP/TCP/UDP – Basic Firewall Configuration

- NAT
- Access Control List, **ACL**

Protocol	IP-src	IP-dst	Port-src	Port-dst	Action
TCP	<i>any</i>	200.200.10.10/32	<i>any</i>	80	<i>accept</i>
TCP	<i>any</i>	<i>any</i>	< 1024	> 1024	<i>accept</i>
<b>ICMP</b>	<b><i>any</i></b>	<i>any</i>	–	–	<i>accept</i>
IP	<i>any</i>	<i>any</i>	–	–	<i>deny</i>



All incoming packets are compared against the ACL.

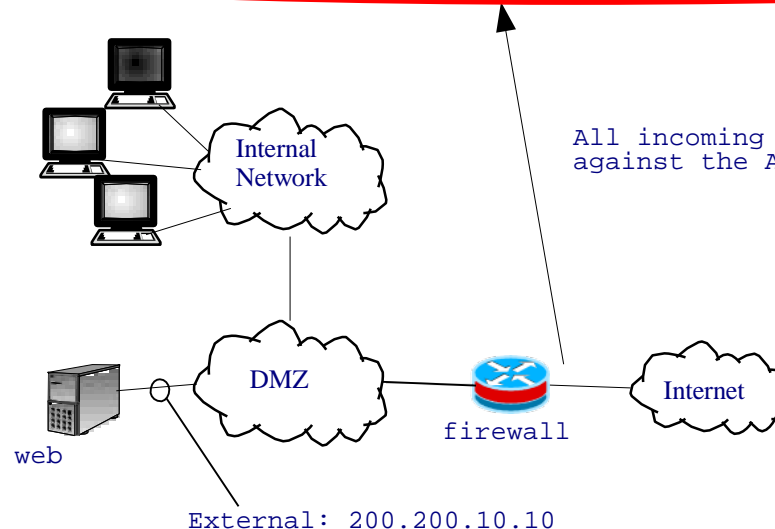


***ANY incoming ICMP datagram going to ANY internal device is allowed***

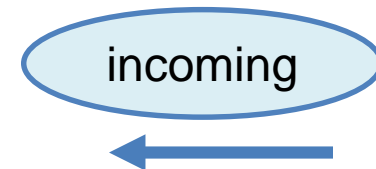
# Security in IP/TCP/UDP – Basic Firewall Configuration

- NAT
- Access Control List, **ACL**

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



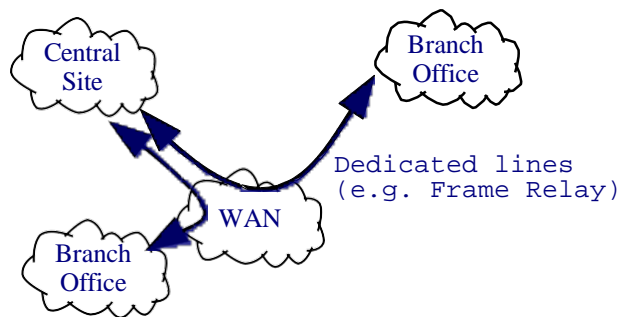
All incoming packets are compared against the ACL.



**ANY other flows than the previous ones are NOT allowed to enter**

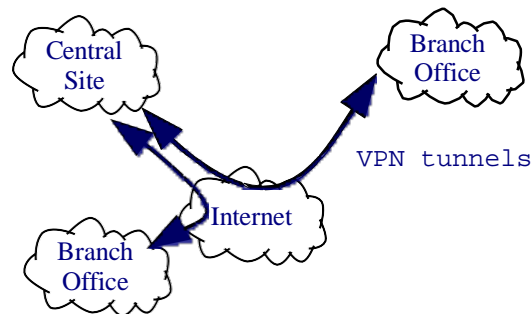
# Security in IP – Virtual Private Network, VPN

- 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.

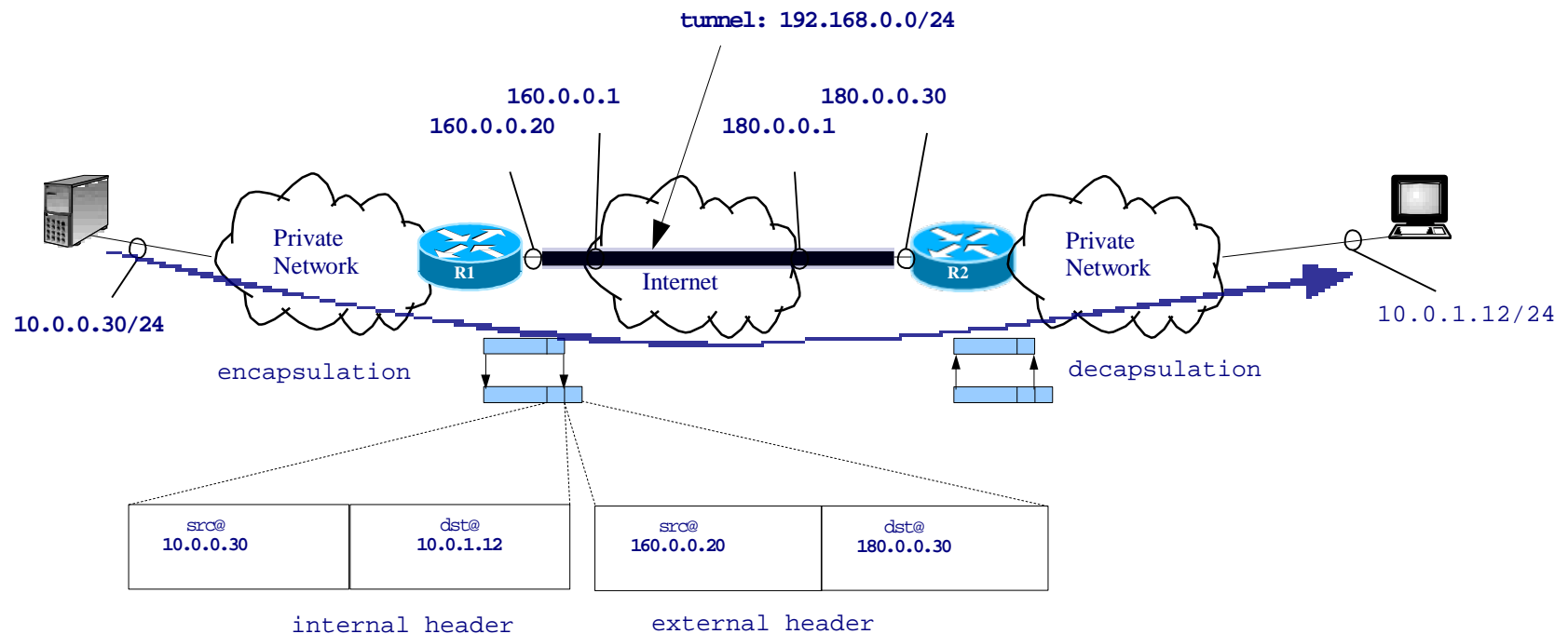


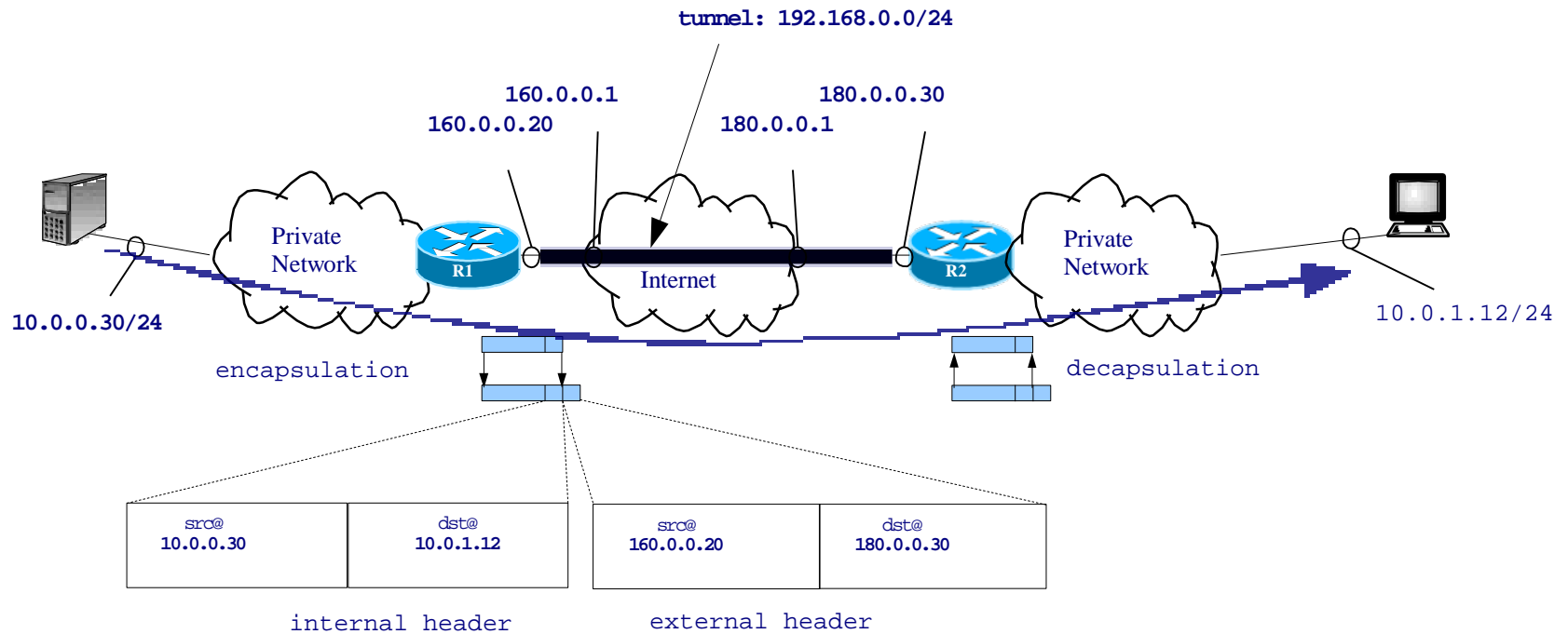
## Virtual Private Network (VPN)

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

# Security in IP – VPN Security

- Authentication
- Cryptography
- **Tunneling**





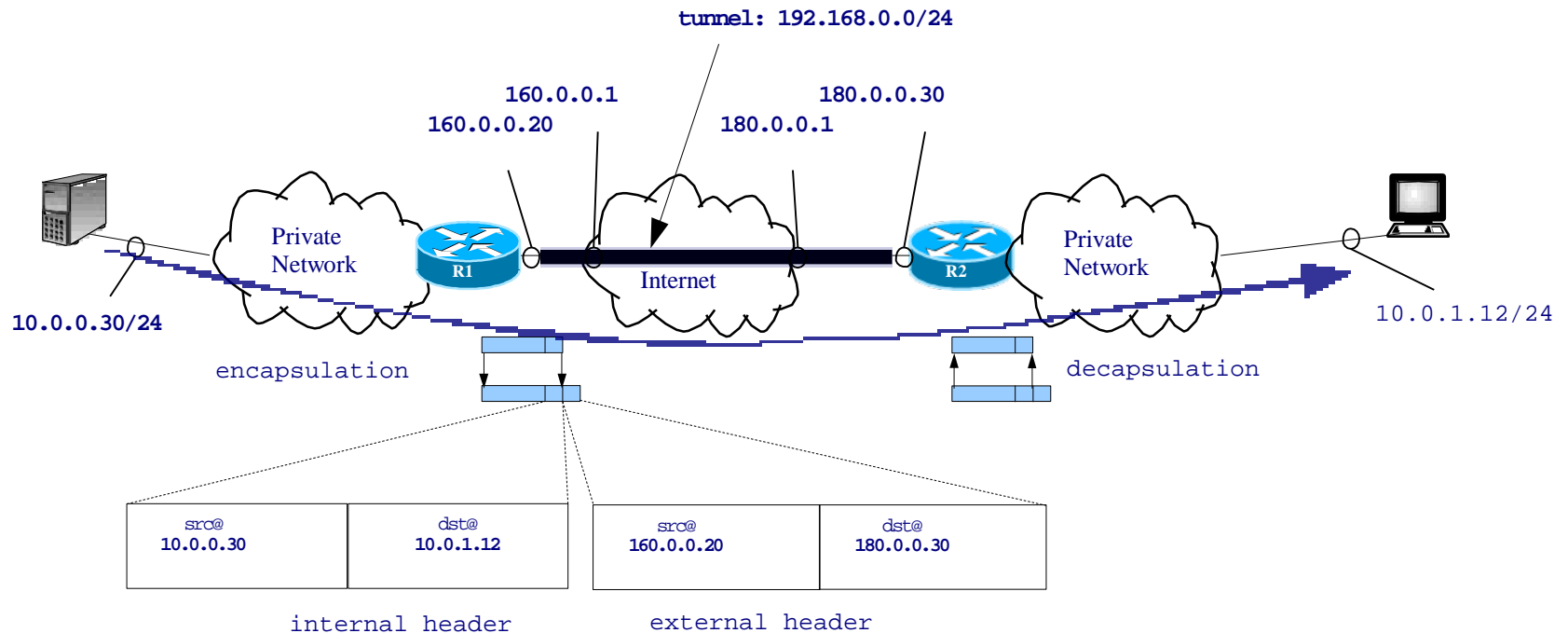
## Routing tables before the tunnel configuration

Destination	Gateway	Genmask	Iface	Destination	Gateway	Genmask	Iface
10.0.0.0	0.0.0.0	255.255.255.0	eth0	10.0.1.0	0.0.0.0	255.255.255.0	eth0
160.0.0.1	0.0.0.0	255.255.255.255	ppp0	180.0.0.1	0.0.0.0	255.255.255.255	ppp0
0.0.0.0	160.0.0.1	0.0.0.0	ppp0	0.0.0.0	180.0.0.1	0.0.0.0	ppp0

R1 Routing Table

R2 Routing Table





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`

## Tunnel configuration

Destination	Gateway	Genmask	Iface	Destination	Gateway	Genmask	Iface
10.0.0.0	0.0.0.0	255.255.255.0	eth0	10.0.1.0	0.0.0.0	255.255.255.0	eth0
160.0.0.1	0.0.0.0	255.255.255.255	ppp0	180.0.0.1	0.0.0.0	255.255.255.255	ppp0
0.0.0.0	160.0.0.1	0.0.0.0	ppp0	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	tunl0	192.168.0.0	0.0.0.0	255.255.255.0	tunl0
10.0.1.0	192.168.0.2	255.255.255.0	tunl0	10.0.0.0	192.168.0.1	255.255.255.0	tunl0

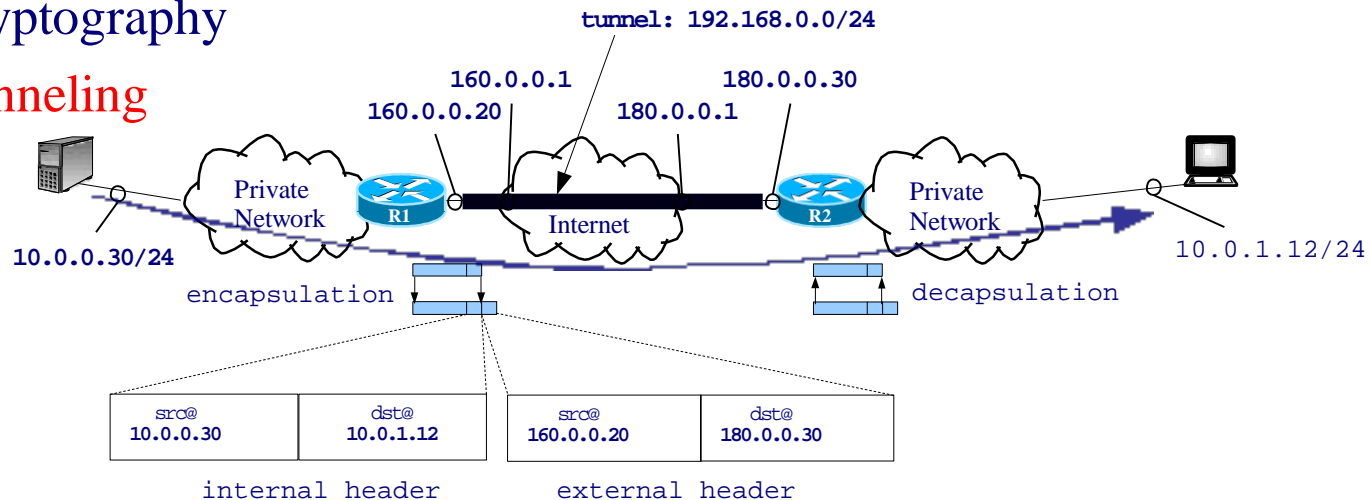
R1 Routing Table

Virtual interface for the tunnel  
 Static route to remote network

R2 Routing Table

# Security in IP – VPN Security

- Authentication
- Cryptography
- Tunneling



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
```

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	tunl0
10.0.1.0	192.168.0.2	255.255.255.0	tunl0

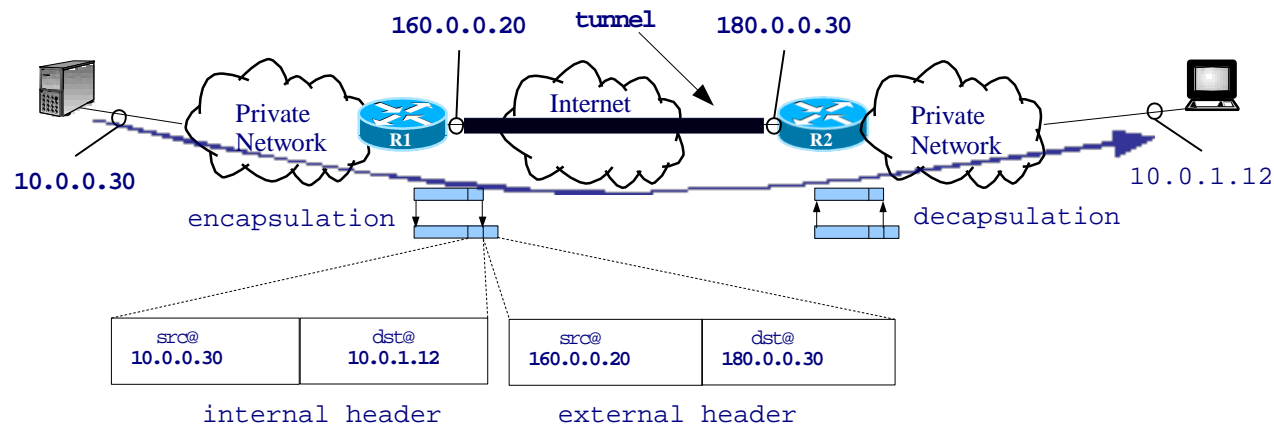
R1 Routing Table

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	tunl0
10.0.0.0	192.168.0.1	255.255.255.0	tunl0

R2 Routing Table

# Security in IP – VPN Tunneling Issues

- **Fragmentation** inside the tunnel will use the external header, thus, the exit router of the tunnel should reassemble fragmented datagrams.
- **ICMP** messages sent inside the tunnel are addressed to the tunnel entry.
- **MTU path discovery** may fail (may not detect small MTU in the tunnel).
- **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 fragments the datagrams before encapsulation, if needed, to avoid the exit router having to reassemble fragmented datagrams.



# Security in IP – VPN Tunneling

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

# IP Networks

## Outline

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