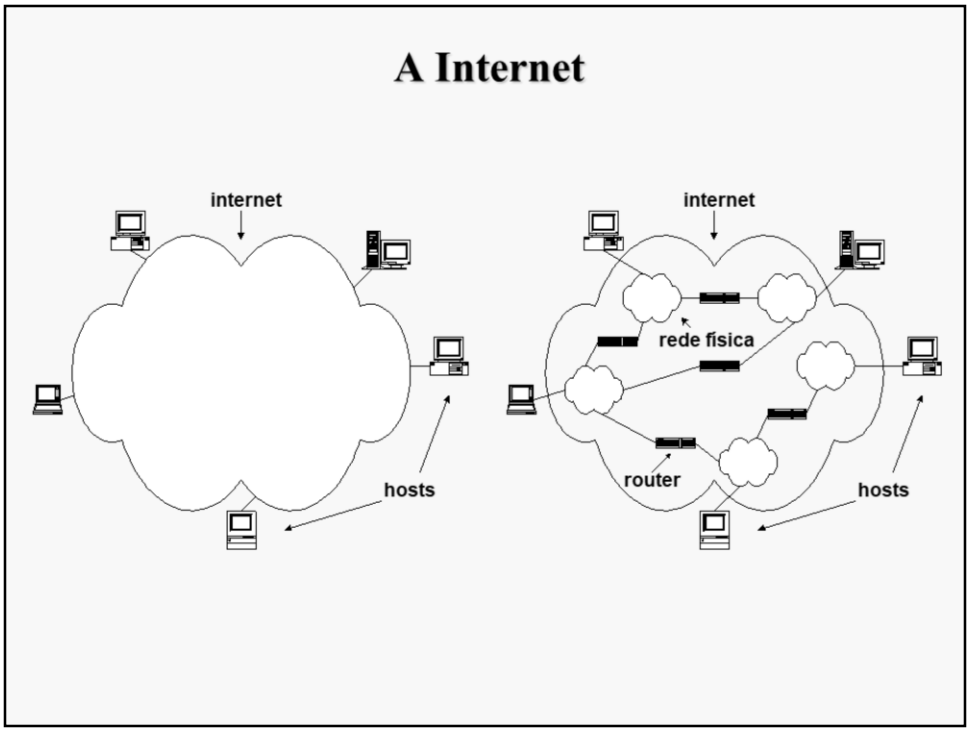

# Redes IP

## Fundamentos de Redes

**Mestrado Integrado em Engenharia de Computadores e Telemática**

**DETI-UA, 2018/2019**

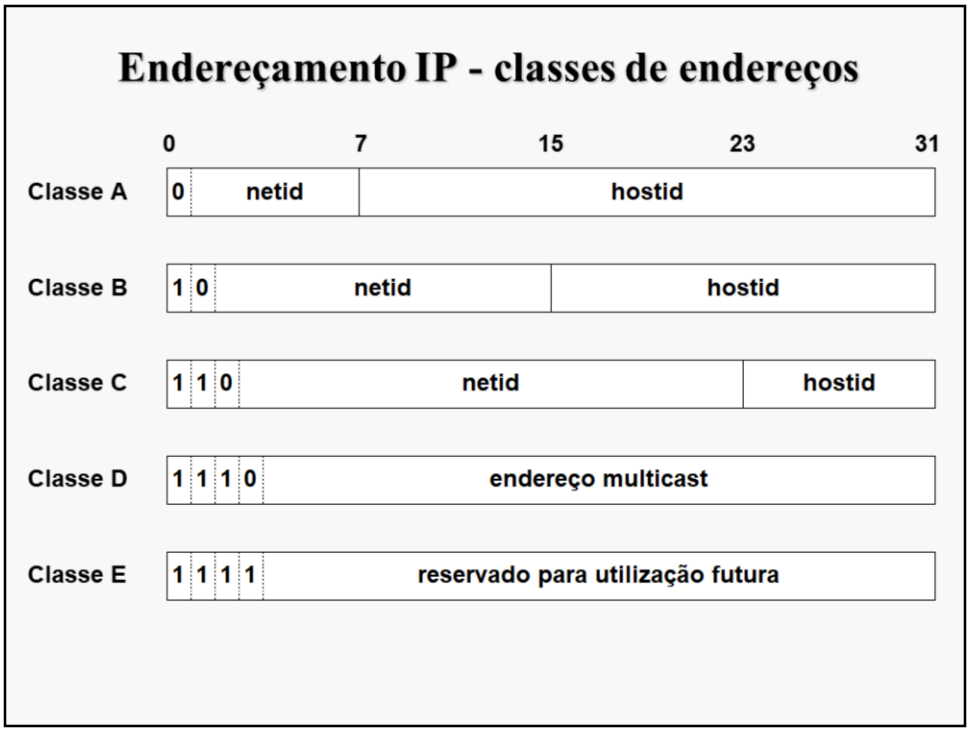



## The Internet

The Internet is a worldwide network system composed by millions of terminal hosts interconnected through many physical networks and routers (connecting the physical networks). It provides the communication means for all hosts to communicate between them.

The Internet Protocol (IP) is the basis of the Internet. This protocol works in such a way that the terminal hosts communicate between them always in the same way, whatever the physical networks and routers that are used to transmit the information from the origin hosts to the destination hosts.

Therefore, IP protocol provides an interface in such a way that terminal hosts “see” the Internet as a cloud without the need to know the details on how it is implemented.




## IP addressing

In order to communicate, each terminal host running the IP protocol must have an IP address. IP addresses (in the IP version 4, or IPv4 for short) are composed by 4 bytes and are classified in 5 different classes.

The addresses that can be assigned to terminal hosts are classified in classes A, B and C. These addresses are said to be unicast addresses since they are used for unicast communications (communications destined to a single host). Unicast addresses are structured in two parts: (i) a netid part, which identifies the IP subnet and (ii) a hostid part, which identifies the host within the IP subnet.

If the first bit (the most significant bit) is 0, the IP address belongs to class A and the netid part is defined by the first byte of the address. If the first two bits are 10, the IP address belongs to class B and the netid part is defined by the first two bytes of the address. If the first three bits are 110, the IP address belongs to class C and the netid part is defined by the first three bytes of the address.

Besides the unicast address classes, there are two additional classes. Class D addresses start with the first four bits 1110 and are used to multicast communications (communications destined to multiple hosts). Finally, class E addresses start with the first four bits 1111 and are reserved for future utilization.

## Divisão do espaço de endereçamento unicast

| Classe | # bits no prefixo | # máximo de redes | # bits no sufixo | # máximo de hosts por rede |
|---|---|---|---|---|
| A | 7 | 128 | 24 | 16,777,216 |
| B | 14 | 16,384 | 16 | 65,536 |
| C | 21 | 2,097,152 | 8 | 256 |

**NOTA: Nem todos os possíveis endereços podem ser usados!**

## Unicast addressing space division

The number of bits on each part of the unicast addresses define the total number of combinations for subnets and hosts on each subnet.
A class A address has 7 bits to define the netid (resulting in a total number of 128 combinations) and 24 bits to define the hostid (resulting in a total of 16777216 combinations).
A class B address has 14 bits to define the neti (resulting in a total number of 16,384 combinations) and 16 bits to define the hostid (resulting in a total of 65536 combinations).
A class C address has 21 bits to define the netid (resulting in a total number of 2097152 combinations) and 8 bits to define the hostid (resulting in a total of 256 combinations).

NOTE: Not all combinations are available to define the netid and the hostid since some combinations have special meanings (some of them shown in the next slide).

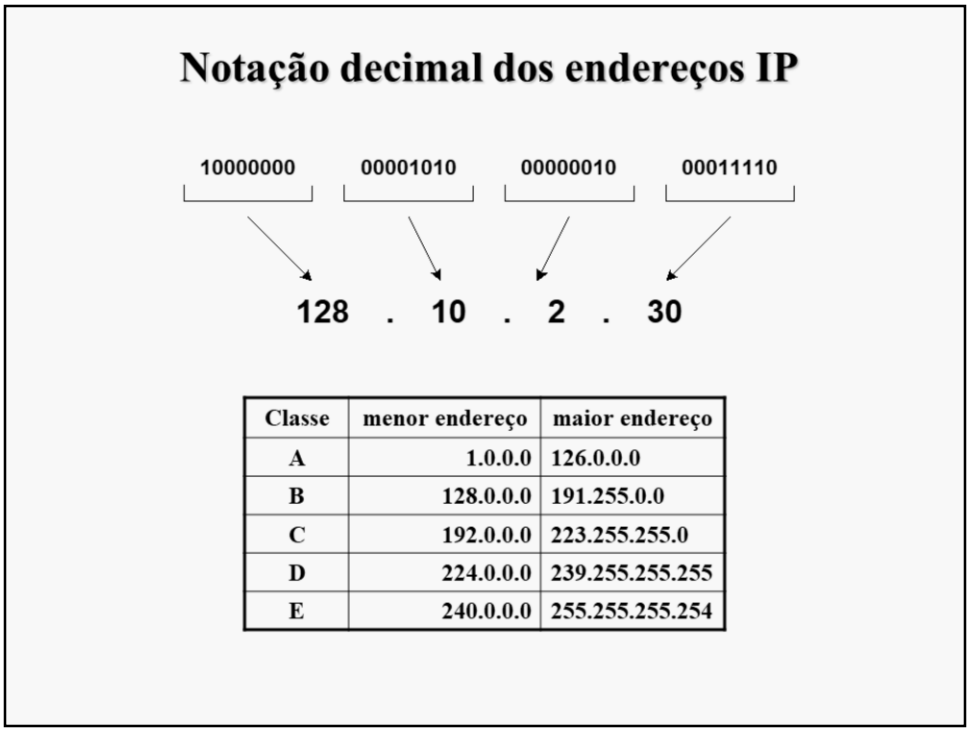
## Endereços IP especiais

| | | |
|---|---|---|
| tudo 0s | | ESTE HOST[1] |
| tudo 0s | host | host NESTA REDE[1] |
| tudo 1s | | BROADCAST LOCAL[2] |
| net | tudo 1s | BROADCAST DIRIGIDO PARA net[2] |
| 127 | qualquer (em geral 1) | LOOPBACK[3] |
| net | tudo 0s | ESTA net[4] |

[1] Permitido apenas na inicialização; nunca é endereço destino válido
[2] Nunca é endereço origem válido
[3] Nunca deve aparecer na rede
[4] Reservado para designar a rede

## Special IP addresses

- The address composed by all bits equal to 0 represents the host that is using it. It is allowed only on host initialization and cannot be used as a destination address.
- The address with the netid part composed by all bits equal to 0 represents the host (defined by the hostid part) on the local subnet. It cannot be used as a destination address.
- The address composed by all bits equal to 1 represents the local broadcast address. It is used as a destination address to send information to all hosts of the local subnet and cannot be used as a source address.
- The address with the hostid part composed by all bits equal to 1 represents the broadcast address of the subnet defined by the netid part. It is used as a destination address to send information to all hosts of a remote subnet and cannot be used as a source address.
- Any class A address starting by 01111111 (in decimal notation, 127) is a loopback address. It is used as a destination address by a host to send information to its own interface (i.e., for host internal communication) and cannot be used to send information to the network.
- The address with the hostid part composed by all bits equal to 0 represents the subnet defined by the netid part.




| Classe | menor endereço | maior endereço |
|---|---|---|
| A | 1.0.0.0 | 126.0.0.0 |
| B | 128.0.0.0 | 191.255.0.0 |
| C | 192.0.0.0 | 223.255.255.0 |
| D | 224.0.0.0 | 239.255.255.255 |
| E | 240.0.0.0 | 255.255.255.254 |

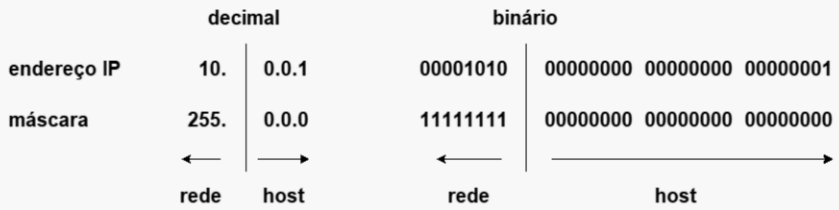
## IP address notation

An IP address is represented by four numbers, separated by dots. Each number is the decimal representation of the corresponding byte.

Taking into consideration the special IP addresses (presented in the last slide) and the decimal notation representation, the above table shows the lowest and the highest addresses of each IP address class.

## Máscaras

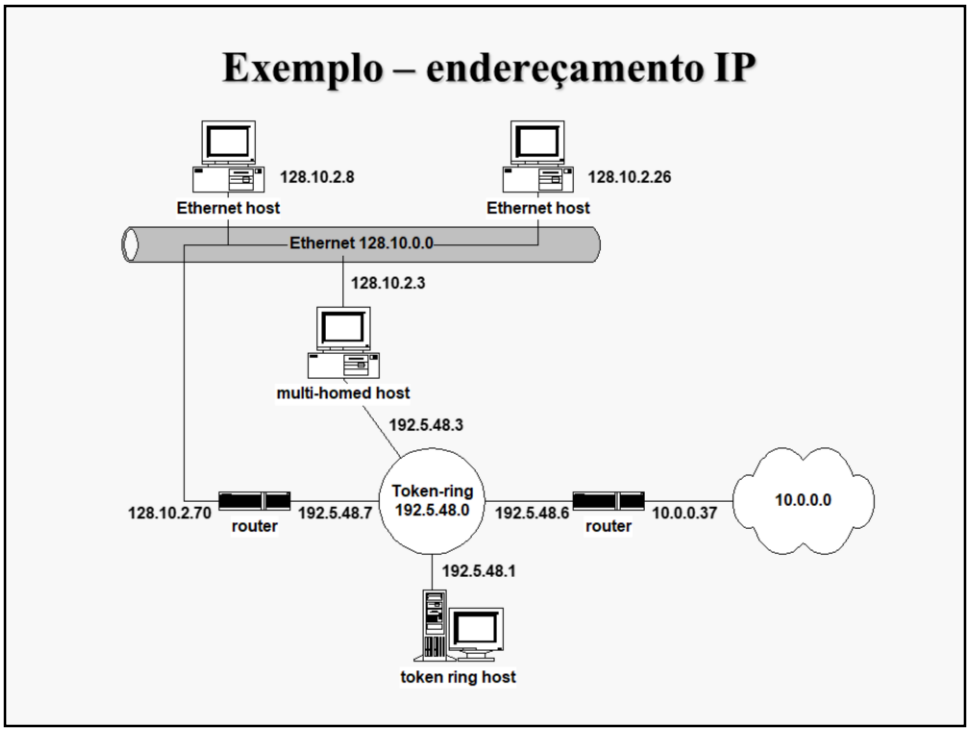
- **Inicialmente os endereços IP tinham fronteiras fixas, sendo a fronteira definida a partir dos primeiros bits do campo de endereço; é o caso dos endereços de classe A, B e C**
- **Depois passaram a ter fronteiras flexíveis, sendo estas definidas a partir de uma máscara**
- **A máscara é utilizada para separar a parte de rede da parte de host dos endereços**




## IP address masks (or netmasks)

Initially, the sizes of the netid and hostid parts of an unicast address were fixed and given by the definition of its class. Soon it was realized that the number of addresses of class A subnets were too large and the number of addresses of class C subnets were too small (at least for many situations).
Meanwhile, a more flexible way was adopted to define the netid and hostid parts of a unicast address, which is based on a mask (or netmask) composed also by four bytes and represented also in decimal notation. The netmask is always composed by a sequence of 1 bits and, then, a sequence of 0 bits. The 1 bits define the netid part of the address and the 0 bits define the hostid part of the address.

IMPORTANT: Besides enabling the choice of the appropriate size of netid and hostid parts, each host uses the netmask to determine the IP address of the subnet where it is attached to by making a bitwise 'and' operation between its IP address and the netmask.




## IP address assignment - example

Consider the above example of a network composed by physical networks (based on different technologies) and hosts connected to them. When assigning IP addresses to each network interface, the following rules must be obeyed:

- All network interfaces attached to the same physical network must have the same netid part and different hostid parts.
- All network interfaces attached to different physical networks must have different netid parts.

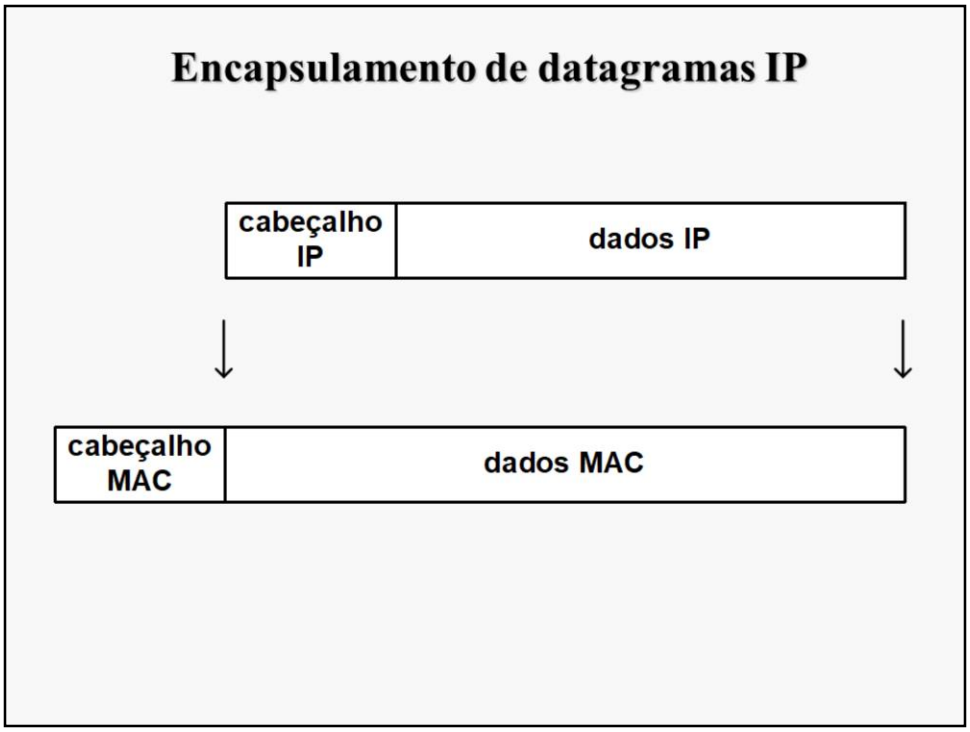
In this way, each host has a very simple way to determine if a destination host (defined by an IP address) is or is not in its own physical network: if the netid part of the IP destination address is equal to the netid part of its own IP address, the destination host is in the same physical network and the origin host can send the information directly to the destination host.

In the above example, we can distinguish three types of hosts:

Single-homed hosts – hosts with a single network interface.

Multi-homed hosts – hosts with more than one network interface; multi-homed hosts do not forward information received from one incoming network interface towards an outgoing network interface.

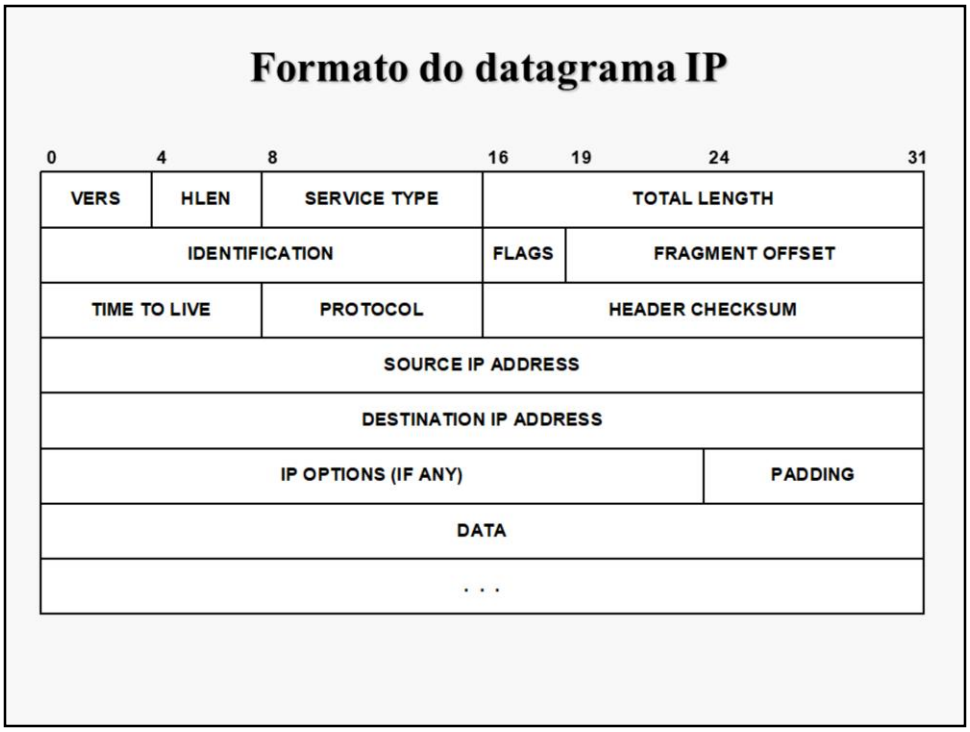
Routers – like multi-homed hosts, routers are hosts with more than one network interface; unlike multi-homed hosts, routers can forward information received from one incoming network interface towards an outgoing network interface.




## Encapsulation of IP datagrams

IP protocol sends information in the form of datagrams. For each block of bytes delivered by the above protocol, IP protocol adds an header forming in this way an IP datagram. Each IP datagram (composed by an IP header field and a data field) is delivered to the lower MAC layer to be sent to the network.
At each physical network, a MAC layer frame is composed by a MAC header field and a data field. At each physical network, each IP datagram is transmitted in the data field of a MAC layer frame (process known as encapsulation).

MAC (Medium Access Control) – is the protocol running on the physical network that manages the way how the transmission medium is used by each attached host to send MAC layer frames to other hosts.




## IP datagram format

An IP datagram is composed by an IP header field and a data field (where data is transported). The IP header has several mandatory fields with a total size of 20 bytes. The last two mandatory fields are the source and the destination IP addresses (obviously, each one with 4 bytes of size). The meaning of the other mandatory fields is explained in the next slides.

The IP header can have option fields. If there are option fields, the header size must be a multiple of 4 bytes (padding bytes are inserted, if required). Therefore, the IP header size can be 20, 24, 28, 32, and so on…

## Campos do Datagrama IPv4

- **Version** (4 bits) – versão do protocolo IP (atualmente a versão mais comum é a versão 4)
- **Header Length** (4 bits) – tamanho do cabeçalho em blocos de 4 octetos
  - quando não tem opções, o cabeçalho ocupa 5 blocos de 4 octetos e o primeiro octeto do cabeçalho IP assume o valor 0x45
- **Service Type** (1 byte) – tipo de serviço ao qual o pacote pertence
  - Identifica o tipo de serviço e o objetivo é diferenciar o tratamento dos pacotes pelos routers com base na qualidade do serviço pretendida (por defeito, este campo tem o valor 0x00)
- **Total Length** (2 bytes) – tamanho do datagrama IP em octetos, incluindo o cabeçalho.
  - o tamanho máximo do datagrama IP é 65 535 octetos
  - no entanto este tamanho está restringido pelo *Maximum Transmission Unit* (MTU) da rede (mecanismo de fragmentação e reagrupamento)

## IP version 4 header description

**Version** (4 bits) – version of the IP protocol (currently, version 4 is the most used version)
**Header Length** (4 bits) – size of IP header in multiple of 4 bytes (for example, if the header size is 20 bytes, the content of this field is 0x5).
**Service Type** (1 byte) – type of service of the IP datagram (used in quality of service architectures); the default value is 0x00
**Total Length** (2 bytes) – size of the IP datagram (header + data); the maximum size of an IP datagram is 65535 bytes; nevertheless, the physical networks have much lower MTU values; a fragmentation and reassembly mechanism is included in IP protocol to solve this issue.

MTU (Maximum Transmission Unit) of a physical network – the maximum size of the data field of its MAC layer frames (for example, the MTU of Ethernet is 1500 bytes).

## Campos do Datagrama IPv4 (continuação)

- **Identification** (2 bytes) – identificador atribuído pela estação que gerou o datagrama
  - este campo é mantido durante o processo de fragmentação permitindo o destinatário identificar os vários fragmentos de um mesmo pacote
- **Flags** (3 bits)
  - o primeiro bit está reservado para uso futuro (assume sempre o valor 0)
  - o segundo bit assume o valor 0 se o datagrama puder ser fragmentado e o valor 1 caso contrário
  - o terceiro bit assume o valor 0 se for o último fragmento e 1 se não for
- **Fragment Offset** (13 bits) – posição (em múltiplos de 8 bytes) do fragmento no datagrama original (o primeiro fragmento tem o valor 0x00)

## IP version 4 header description (continuation)

**Identification** (2 bytes) – a value assigned by the origin host to the IP datagram; this value is different for every new IP datagram; this value is copied to all IP fragment datagrams in the fragmentation of an original IP datagram ( in this way, the destination host can identify the IP fragment datagrams of each original IP datagram).
**Flags** (3 bits):
- first bit is reserved for future use (default value is 0)
- second bit is the "**do not fragment bit**": it is 1 if the source does not allow the IP datagram to be fragmented and 0 otherwise (if an IP datagram requires fragmentation to be transmitted over a physical network and this bit is 1, the IP datagram is discarded)
- the third bit is the "**last fragment bit**": it is 0 if the IP datagram is the last fragment of the original IP datagram or 1, otherwise
**Fragment Offset** (13 bits) – position (in multiples of 8 bytes) of this fragment on the original IP datagram; the Fragment Offset value indicates how many bytes are in all previous datagrams (first fragment has the value 0x00)

NOTE: a non fragmented IP datagram reaches the destination host with Fragment Offset = 0x00 and the "last fragment bit" = 0.

## Campos do Datagrama IPv4 (continuação)

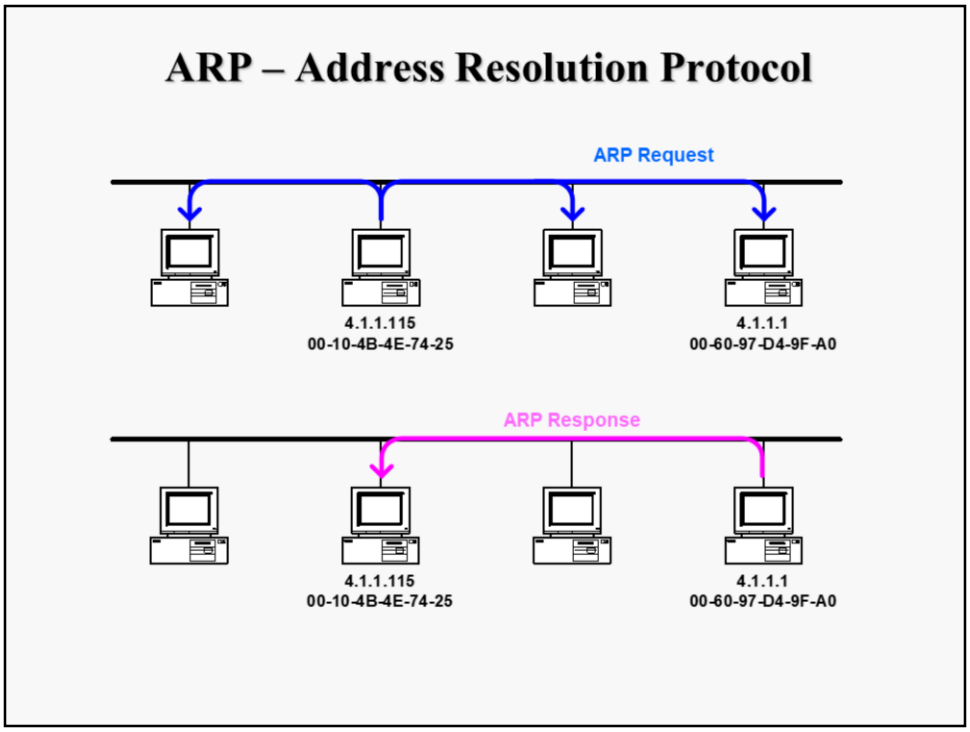
- **Time to Live** (1 byte) - o máximo tempo que o datagrama pode permanecer na rede
  - é alterado em cada router e quando atinge o valor 0 o datagrama é eliminado
  - cada router decrementa este campo em 1 unidade ou no número de segundos que demorou a processar o datagrama
- **Protocol** (1 byte) - especifica o protocolo de nível superior
  - exemplos: 1 - ICMP, 6 - TCP e 17 - UDP
- **Header Checksum** (2 bytes) - resultado da soma (em palavras de 16 bits) dos outros campos do cabeçalho
  - como o header é alterado em cada router, este valor é também recalculado
  - permite detetar erros de transmissão que alterem o cabeçalho do datagrama

## IP version 4 header description (continuation)

**Time to Live or TTL** (1 byte) – the maximum time that the IP datagram can be in transit before reaching the destination host; each router subtracts to this value the number of seconds that it takes to process it (this value is decremented at least by 1); if the value reaches 0, the router discards the IP datagram

**Protocol** (1 byte) – code specifying the higher layer protocol to which the data field belongs; examples: 1 - ICMP, 6 - TCP and 17 - UDP

**Header Checksum** (2 bytes) – result of the sum (in 16 bit words) of the other header fields; it enables each receiver (intermediate routers and destination host) to detect transmission errors in the IP header (if a transmission error is detected, the IP datagram is discarded); since the TTL field is changed by each router, the Header Checksum is also changed on each router




## ARP – Address Resolution Protocol

Each physical network technology has its own addresses. The technologies standardized by IEEE (for example, Ethernet, Token Ring, WiFi or WiMax), use the same addressing scheme: each address has a size of 6 bytes. These addresses are coded by manufacturers on NICs (Network Interface Cards) and are guaranteed to be unique (unlike IP addresses, physical addresses are represented in hexadecimal notation).

In the figure above, if the host 4.1.1.115 has an IP datagram to send to host 4.1.1.1, the IP datagram must be encapsulated on a MAC layer frame where the frame header must specify the origin and destination MAC addresses. Before doing that, host 4.1.1.115 must first know what is the MAC address of the host whose IP address is 4.1.1.1.

This is done through the Address Resolution Protocol (ARP). First, host 4.1.1.115 sends an ARP Request packet to all hosts requesting the MAC address of the host whose IP address is 4.1.1.1. If such an host is active, it sends an ARP reply packet, only to the requesting host, with the requested information.

## ARP Request

| No. | St. | Source Address | Dest Address | Layer | Summary | Len |
|---|---|---|---|---|---|---|
| 1 | Ok | This station | Broadcast | ARP | Op=ARP Request, | 46 |
| 2 | Ok | 006097D49FA0 | This station | ARP | Op=ARP Response | 64 |
| 3 | Ok | This station | Broadcast | ARP | Op=ARP Request, | 46 |

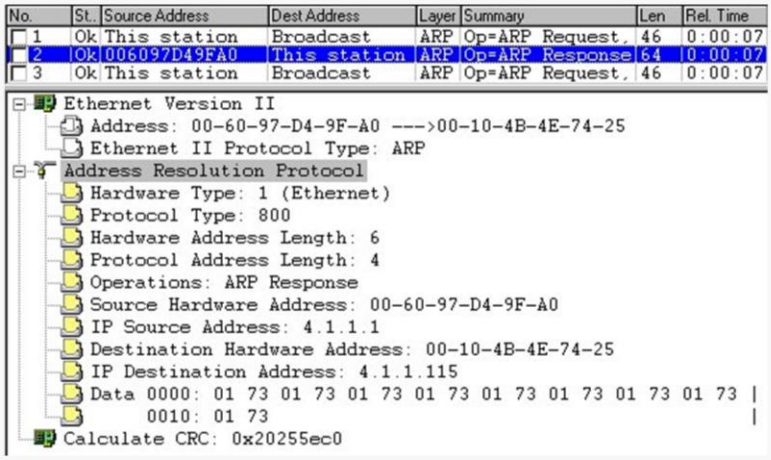
```
Ethernet Version II
    Address: 00-10-4B-4E-74-25 --->FF-FF-FF-FF-FF-FF
    Ethernet II Protocol Type: ARP
Address Resolution Protocol
    Hardware Type: 1 (Ethernet)
    Protocol Type: 800
    Hardware Address Length: 6
    Protocol Address Length: 4
    Operations: ARP Request
    Source Hardware Address: 00-10-4B-4E-74-25
    IP Source Address: 4.1.1.115
    Destination Hardware Address: 00-00-00-00-00-00
    IP Destination Address: 4.1.1.1
Calculate CRC: 0x27621e3b
```

**ARP Request enviado pela estação 4.1.1.115 para saber o endereço MAC da estação 4.1.1.1.**

## ARP Request

ARP packets are encapsulated in MAC layer frames. The above is the content of an ARP Request packet encapsulated on an Ethernet frame. In the Ethernet frame header, the origin address is the MAC address of host 4.1.1.115 and the destination address is the MAC broadcast address FF-FF-FF-FF-FF-FF (an address with all bits equal to 1). The ARP Request packet specifies the origin MAC and IP addresses, the destination IP address and an empty destination MAC address.

## ARP Reply



| No. | St. | Source Address | Dest Address | Layer | Summary | Len | Rel. Time |
|---|---|---|---|---|---|---|---|
| 1 | Ok | This station | Broadcast | ARP | Op=ARP Request, | 46 | 0:00:07 |
| 2 | Ok | 006097D49FA0 | This station | ARP | Op=ARP Response | 64 | 0:00:07 |
| 3 | Ok | This station | Broadcast | ARP | Op=ARP Request, | 46 | 0:00:07 |

```
Ethernet Version II
    Address: 00-60-97-D4-9F-A0 --->00-10-4B-4E-74-25
    Ethernet II Protocol Type: ARP
Address Resolution Protocol
    Hardware Type: 1 (Ethernet)
    Protocol Type: 800
    Hardware Address Length: 6
    Protocol Address Length: 4
    Operations: ARP Response
    Source Hardware Address: 00-60-97-D4-9F-A0
    IP Source Address: 4.1.1.1
    Destination Hardware Address: 00-10-4B-4E-74-25
    IP Destination Address: 4.1.1.115
    Data 0000: 01 73 01 73 01 73 01 73 01 73 01 73 01 73 01 73 |
         0010: 01 73                                           |
Calculate CRC: 0x20255ec0
```

**Resposta da estação 4.1.1.1 enviada através de ARP Response: o endereço MAC é 00-60-97-d4-9f-a0**

## ARP Response

The above is the content of an ARP Reply packet encapsulated on a Ethernet frame. In the Ethernet frame header, the origin address is the MAC address of host 4.1.1.1 and the destination address is the MAC address of host 4.1.1.115 (which is the requester). The ARP Reply specifies its MAC and IP addresses and the destination MAC and IP addresses.

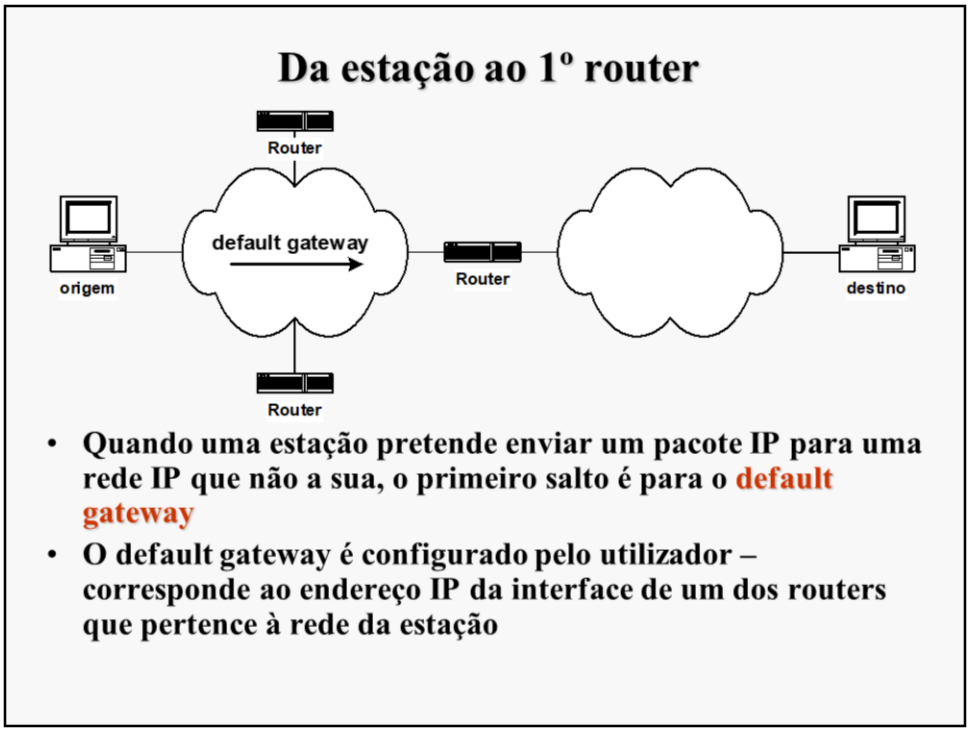
## O comando ARP

```
ARP -a [inet_addr] [-N if_addr]

  -a            Displays current ARP entries by interrogating the current
                protocol data.  If inet_addr is specified, the IP and Physical
                addresses for only the specified computer are displayed.  If
                more than one network interface uses ARP, entries for each ARP
                table are displayed.
  -g            Same as -a.
  inet_addr     Specifies an internet address.
  -N if_addr    Displays the ARP entries for the network interface specified
                by if_addr.
  -d            Deletes the host specified by inet_addr.
  -s            Adds the host and associates the Internet address inet_addr
                with the Physical address eth_addr.  The Physical address is
                given as 6 hexadecimal bytes separated by hyphens. The entry
                is permanent.
  eth_addr      Specifies a physical address.
  if_addr       If present, this specifies the Internet address of the
                interface whose address translation table should be modified.
                If not present, the first applicable interface will be used.
Example:
  > arp -s 157.55.85.212   00-aa-00-62-c6-09  .... Adds a static entry.
  > arp -a                                    .... Displays the arp table.
```

## ARP command

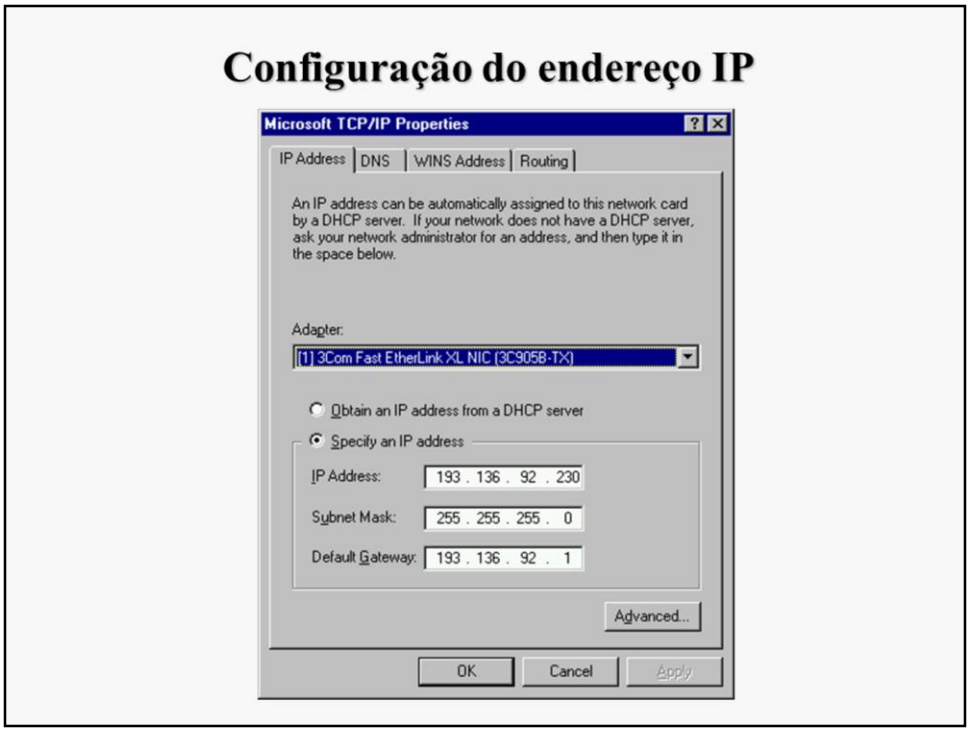
The mappings between MAC and IP addresses discovered by the ARP protocol are temporarily saved on an ARP table on each IP host. There discovered mappings are discarded after a time limit of no usage. They are not permanently saved since they can become obsolete (either because the IP addresses can be changed on hosts or because NICs can be replaced due to malfunctioning).

The ARP command (whose syntax is shown above for the Windows OS) can be used to manage the host ARP table (visualize the table, delete table entries, configure permanent entries, and so on…).



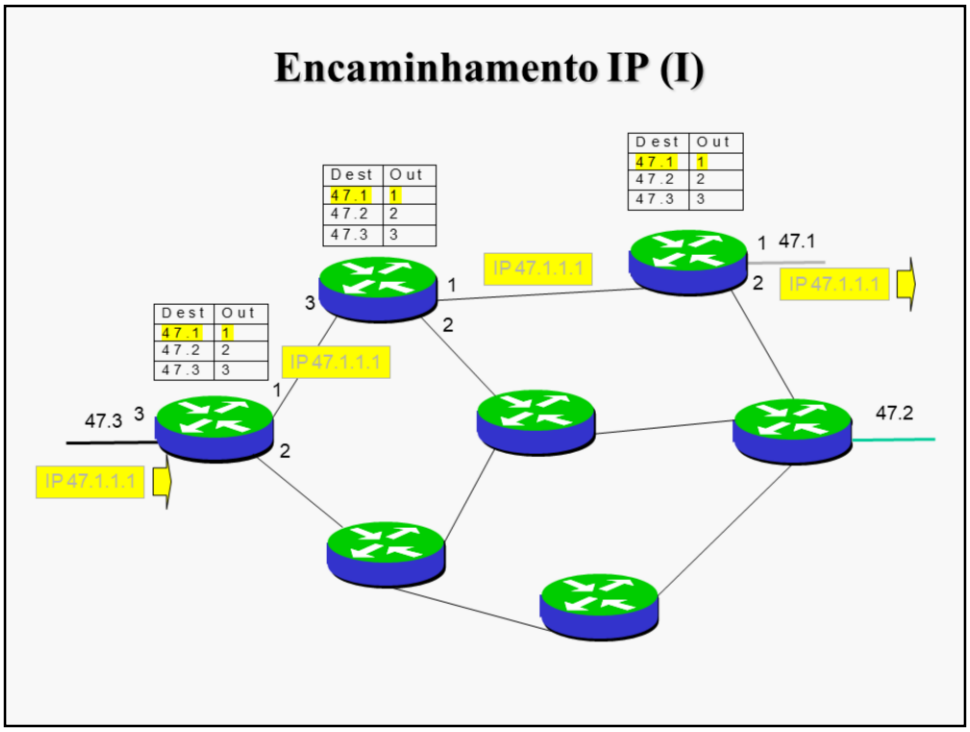

## From de origin IP host to the first router

When an IP host has an IP datagram for an IP destination address, the host compares the netid part of the IP destination address with the netid part of its own IP address. If they are not equal, it means that the destination host is not attached to its physical network. In this case, the host sends the packet to the Default Gateway.
In order to have global connectivity, an IP host must be configured with the IP address of its Default Gateway. This address must be an IP address assigned to a network interface of a router connected to its own physical network.




## IP host configuration

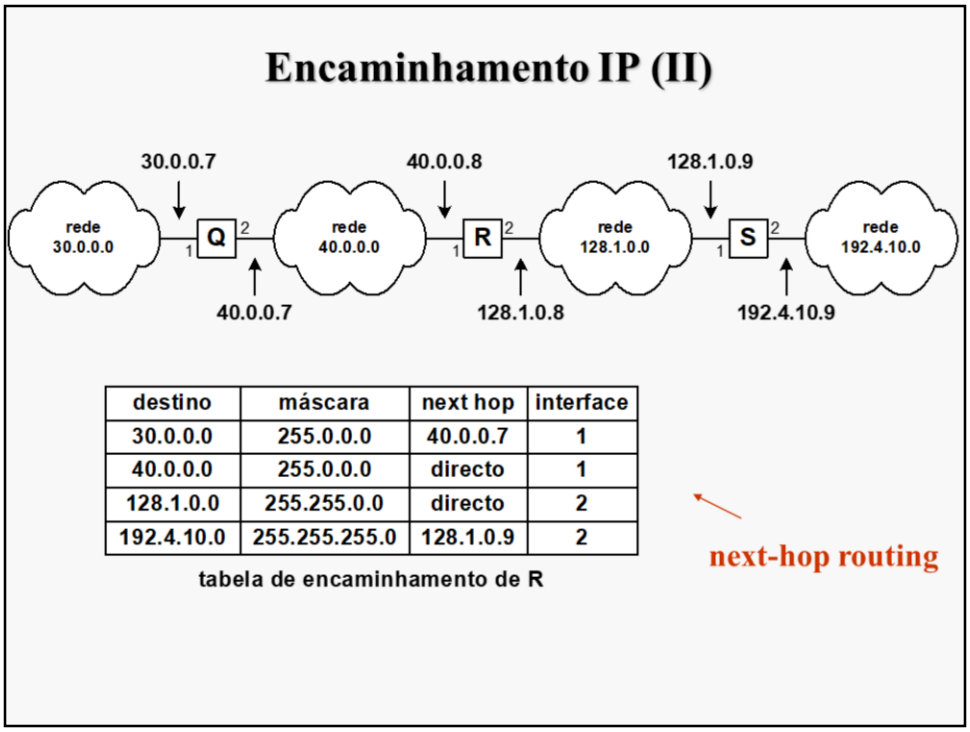
The figure above shows an example of an IP host configuration window (in Windows OS) where the basic information is requested: the host IP address, the netmask and the IP address of its Default Gateway.




## IP routing (I)

Routers are the network elements responsible to forward each IP datagram towards its destination host. In order to reach this task, each router has a routing table which defines the output port to be used towards each possible destination.

In the figure above, this task is illustrated in a very simplified way. The first router receives on port 3 an IP datagram from the origin host to the IP destination host 47.1.1.1. The router checks its routing table and finds an entry stating that datagrams for IP addresses starting by 47.1 should be transmitted through output port 1 and, therefore, it forwards this datagram through this port. The process is repeated on the second and the third routers.



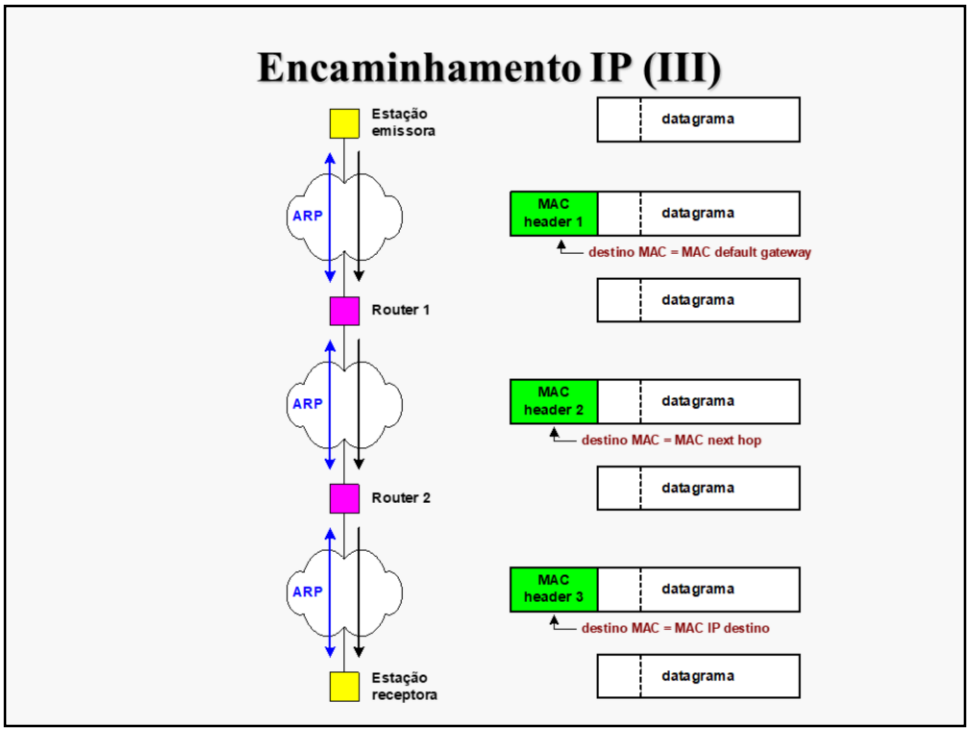

## IP routing (II)

In the above example, some more details on routing tables are shown. It shows the routing table of router R in the network shown. Each routing table entry identifies:

- a destination network (specified by its IP network address and netmask),
- the output port to forward the datagrams towards the destination network,
- the IP address of the next router network interface in the path towards the destination network.

Routing on IP networks is sometimes also referred to as “next-hop routing” since each router forwards each datagram based on the identification of the next hop router in the path towards the destination.

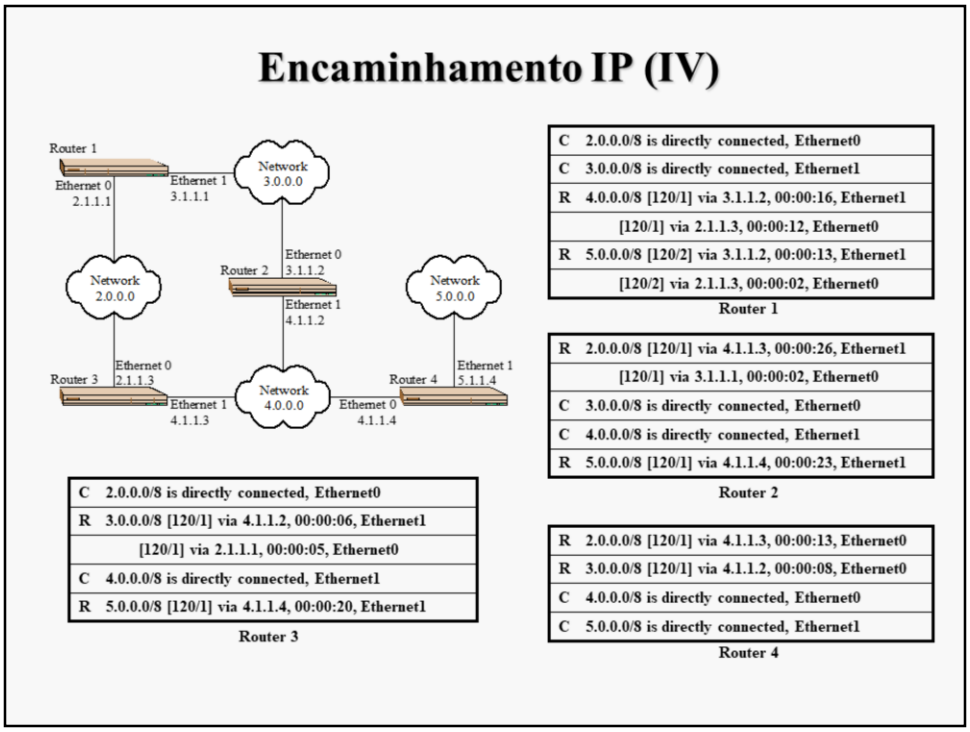
Note that if the destination network is directly attached to the router, the IP address of the next hop router is absent (in the routing table) since, in this case, the router must send the datagram directly to the destination host.




## IP routing (III)

In general, the transmission of an IP datagram from an origin host to a destination host involves the following steps:

1. If necessary, the origin host discovers (through ARP) the MAC address of its Default Gateway.
2. The IP datagram is encapsulated on a MAC layer frame (with its MAC address as source address and the MAC address of the Default Gateway as destination address) and is sent to the network.
3. On each router before the last in the path towards the destination host:
    - 3.1. The router decapsulates the IP datagram from the incoming MAC layer frame.
    - 3.2. From its routing table, gets the outgoing port to be used and the next hop IP address for the destination network.
    - 3.3. If necessary, discovers (through ARP) the MAC address of the next hop IP address.
    - 3.4. The IP datagram is encapsulated on a MAC layer frame (with the MAC address of the outgoing port as source address and the MAC address of the next hop router as destination address) and is sent through the outgoing port.
4. On the last router in the path towards the destination host:
    - 4.1. The router decapsulates the IP datagram from the incoming MAC layer frame.
    - 4.2. From its routing table, gets the outgoing port to be used and the information that the destination host is in the directly attached network.
    - 4.3. If necessary, discovers (through ARP) the MAC address of the destination IP address.
    - 4.4. The IP datagram is encapsulated on a MAC layer frame (with the MAC address of the outgoing port as source address and the MAC address of the destination host as destination address) and is sent through the outgoing port.

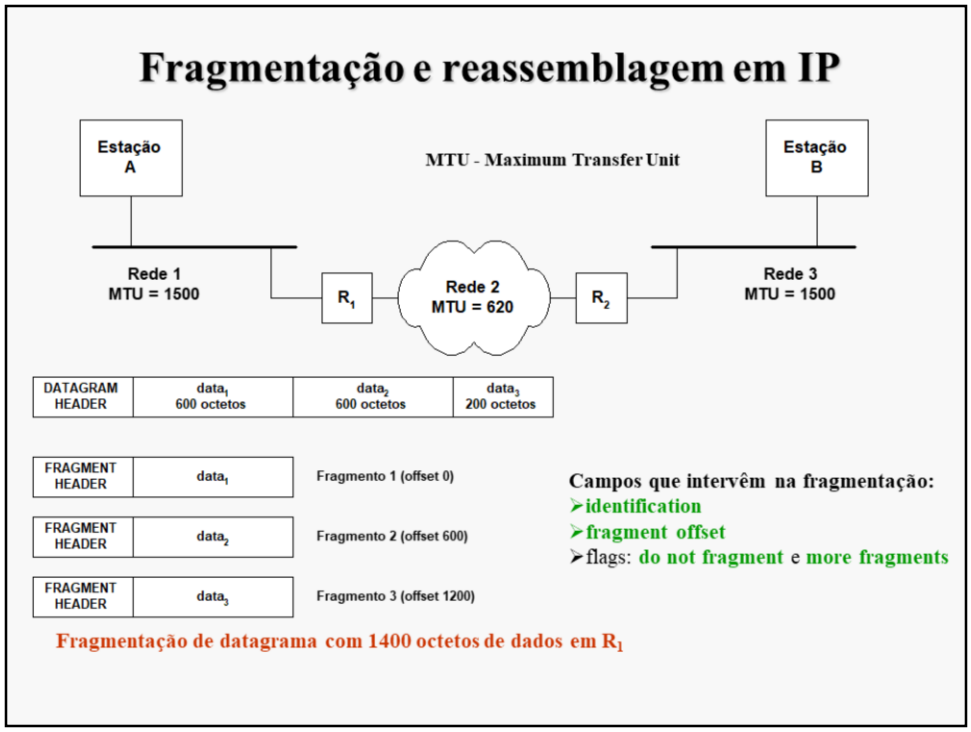



## IP routing (IV)

For the network shown above, composed by four routers interconnecting four Ethernet networks, the routing tables observed in the routers are presented.

In this case, the RIP routing protocol is active on all routers to compute the routing tables. This routing protocol composes the routing tables with the next hop routers that provide the minimum number of hops towards the destination network (entries starting with the letter 'R').

Note that a router might have more than one entry for each destination (if there are multiple routing paths with the same minimum number of hops). When this happens, routers implement load balancing: they use all entries in a way to equally balance the use of all routing paths.




## Fragmentation and reassembly process

When an IP datagram is larger than the MTU of the physical network, the sending host must fragment it in multiple smaller IP datagrams whose size is not larger than the MTU. The fragmentation operation can be done by either the origin host or any router.

IMPORTANT NOTE: All IP fragment datagrams are forwarded individually towards the destination host. The reassembly operation (i.e., the operation of joining all fragments to recover the original IP datagram) is conducted only by destination host.

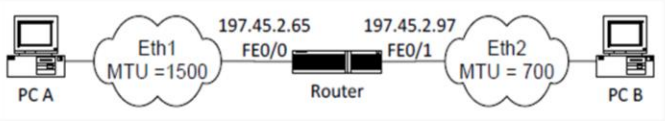
The IP fragmentation process is as follows:

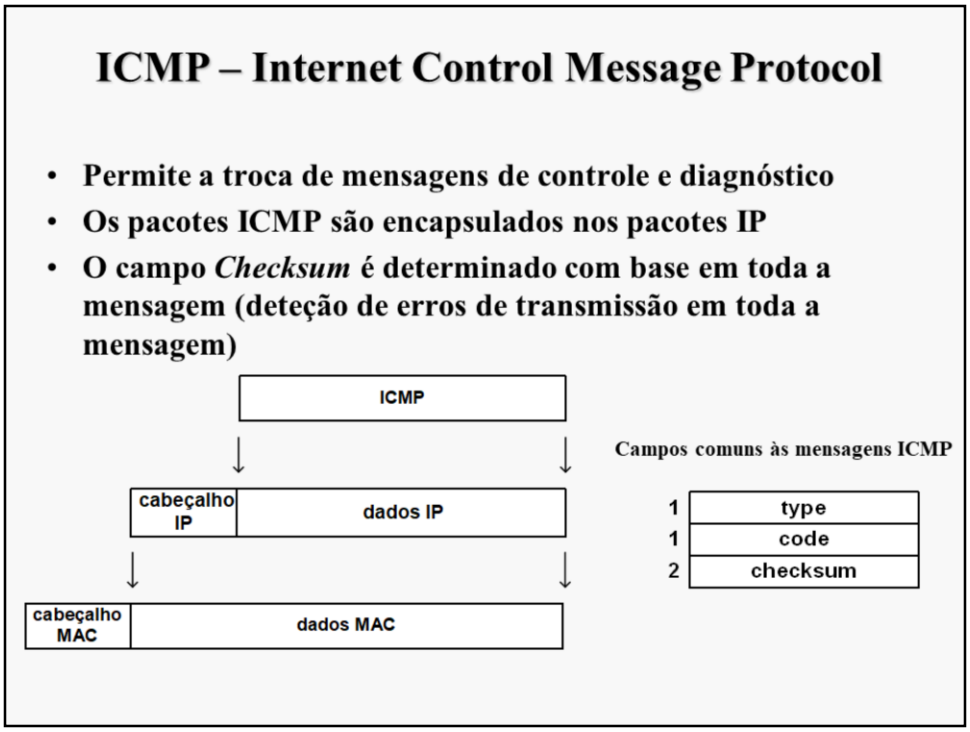
1. The data field is segmented in an ordered set of blocks such that each block plus the header is not larger than the MTU. Each block with its header forms an IP fragmented datagram.
2. The Identification field of all fragments is set with the Identification field of the original IP datagram (in this way, the destination host can identify all fragments of an IP datagram).
3. The Fragment Offset field of fragment *n* is set with the total number of data bytes send by all previous fragments from 1 to *n* – 1 (in this way, the destination host can identify missing fragments and can order the fragments if they are received out of order).
4. The 'more fragments' flag is set with 0 in the last fragment and 1 in all previous fragments (in this way, the destination host can know what is the last fragment and, therefore, check if all fragments were received).

In the above figure, router $R_1$ has an IP packet with 1400 bytes of data for host B. The packet is fragmented in three IP fragmented packets. Since the MTU of the forwarding network is 620 bytes, the first two fragments have data blocks of 600 bytes and the third fragment has the remaining 200 bytes. The Fragment Offset is 600 in the second fragment (the data of the first fragment) and is 1200 in the third fragment (the total data of the first and second fragments).

# Exemplo

- **Num ping do PC A para o PC B, o PC A envia uma mensagem ICMP de 900 bytes. Os pacotes IP que transportam esta mensagem têm o campo IDENTIFICATION com o valor 385. Indique justificadamente quantos fragmentos IP são recebidos pelo PC B, quem gera os fragmentos IP e qual o tamanho (em Bytes) de cada fragmento IP (incluindo o cabeçalho).**






## Internet Control Message Protocol (ICMP)

Internet Control Message Protocol (ICMP) is part of the Internet Protocol Suite. ICMP messages are generated either in response to errors in IP datagrams or for diagnostic or routing purposes. In the case of response to errors, ICMP messages are always sent to the IP address of the origin host of the IP datagram.

ICMP messages are encapsulated on IP datagrams. The three first fields of an ICMP message are common to all ICMP messages: **type** (1 byte), **code** (1 byte) and **checksum** (2 bytes).

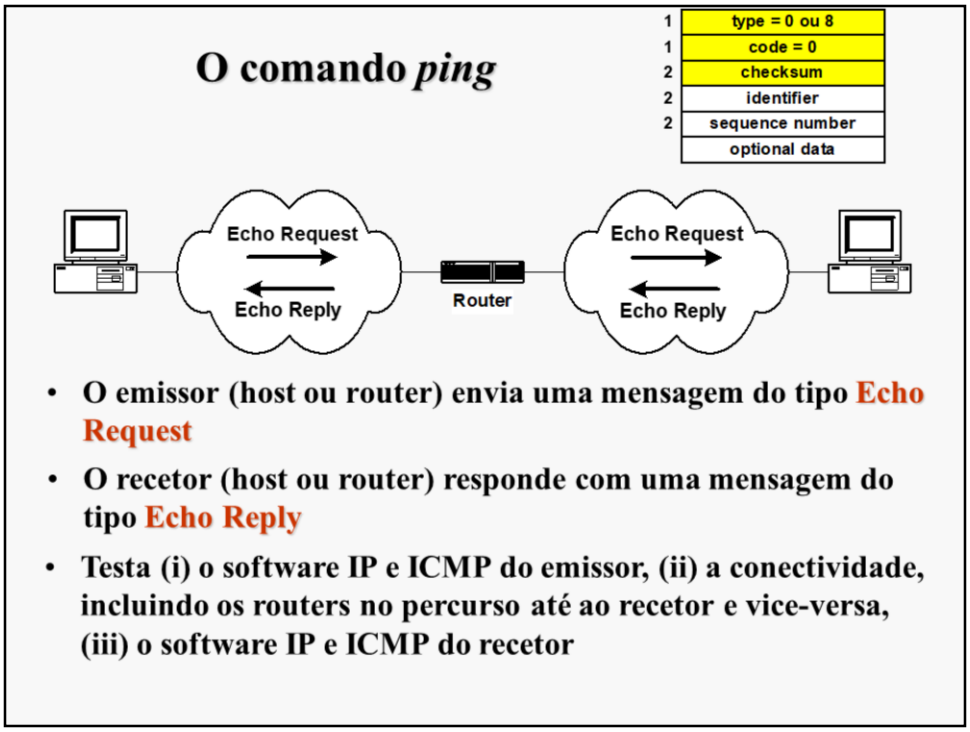
The checksum field is computed based on the content of the complete message and enables the destination host to check for transmission errors on the complete message.

## Tipos de mensagens ICMP

| Campo type | Significado |
|---|---|
| 0 | Echo Reply |
| 3 | Destination unreachable |
| 4 | Source quench |
| 5 | Redirect |
| 8 | Echo request |
| 11 | Time exceeded |
| 12 | Parameter problem |
| 13 | Timestamp request |
| 14 | Timestamp reply |

## ICMP message types

The type field (the first field of an ICMP message) defines the type of ICMP message. In the above, some ICMP message types are identified together with their assigned type values. In the following slides, some of these message types are further addressed.




## *ping* command

The *ping* command uses the ICMP Echo Request and ICMP Echo Reply messages. When a ping command is run on an origin host to a remote IP address, some ICMP Echo Request messages are sent by the origin host for the remote IP address. When a remote host receives an ICMP Request message from an origin IP address, it sends back an ICMP Echo Reply message.

The type field is either 8 (ICMP Echo Request) or 0 (ICMP Echo Reply) and the code field is always zero. Both messages have two additional fields: the **identifier** field (2 bytes) and the **sequence number** field (2 bytes). The content of these two fields on the ICMP Echo Request messages are copied to the ICMP Echo Reply messages sent back to the origin host.

At the end of the message, optional data can be inserted to generate ICMP messages of different sizes (for example, to test fragmentation and reassembly malfunctioning). The ICMP Echo Reply messages are defined with the same optional data size as the one of the received ICMP Echo Request messages.

A successful run of ping command is when an Echo Reply is received for each sent Echo Request message. This command tests the correct operation of the TCP/IP protocol stack on the origin host, the network connectivity between origin and destination hosts and the TCP/IP protocol stack on the destination host.

## Opções *ping*

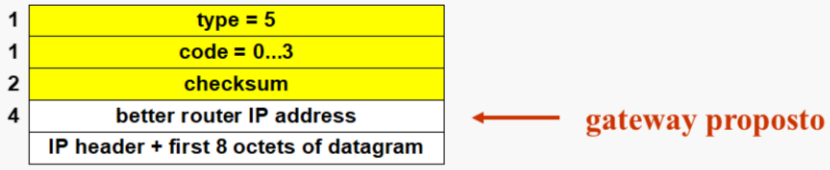
```
C:\>ping

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
            [-r count] [-s count] [[-j host-list] | [-k host-list]]
            [-w timeout] destination-list

Options:
    -t             Ping the specifed host until stopped.
                   To see statistics and continue - type Control-Break;
                   To stop - type Control-C.
    -a             Resolve addresses to hostnames.
    -n count       Number of echo requests to send.
    -l size        Send buffer size.
    -f             Set Don't Fragment flag in packet.
    -i TTL         Time To Live.
    -v TOS         Type Of Service.
    -r count       Record route for count hops.
    -s count       Timestamp for count hops.
    -j host-list   Loose source route along host-list.
    -k host-list   Strict source route along host-list.
    -w timeout     Timeout in milliseconds to wait for each reply.
```

## *ping* options

The *ping* command (whose syntax is shown above for the Windows OS)) can be used to set the content of some fields of the ICMP Echo Request messages.
For example, the `-l` option enables the user to define the size of the optional data, the `-i` option enables the user to define the TTL field of the IP header of the messages, the `-n` option enables the user to define how many ICMP Echo Request messages are sent, and so on…

## ICMP Redirect

- **Quando um router deteta que uma estação está a usar uma rota que não é a melhor envia-lhe um mensagem ICMP Redirect para que ele mude de rota**
- **O router inicial, para além do ICMP Redirect, envia também o datagrama original para o destino**
- **Não possibilita mudanças de rotas entre routers; apenas entre um host e um router ligados à mesma rede**

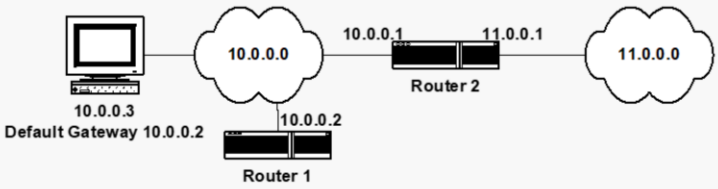



## ICMP Redirect message

The ICMP Redirect message is used when a router receives an IP datagram from an host and detects that it is not the appropriate Default Gateway to be used for that IP datagram, i.e. the router checks its routing table and sees that another router exists on the same physical network with a more direct route.
In this situation, the router: (i) forwards the IP datagram to the destination and (ii) sends to the origin host an ICMP Redirect message with the IP address of the other router.
The ICMP Redirect message does not allow to change routes between routers.

The type field of an ICMP Redirect message is 5.
The code field is used to give more information on which IP datagrams should "be redirected":
Code 0 - Redirect datagrams for the network
Code 1 - Redirect datagrams for the host
Code 2 - Redirect datagrams for the Type of Service and the network
Code 3 - Redirect datagrams for the Type of Service and the host

## Exemplo – ICMP Redirect (I)




Active Routes:

| Network Destination | Netmask | Gateway | Interface | Metric |
|---|---|---|---|---|
| 0.0.0.0 | 0.0.0.0 | 10.0.0.2 | 10.0.0.3 | 1 |
| 10.0.0.0 | 255.0.0.0 | 10.0.0.3 | 10.0.0.3 | 1 |
| 10.0.0.3 | 255.255.255.255 | 127.0.0.1 | 127.0.0.1 | 1 |
| 10.255.255.255 | 255.255.255.255 | 10.0.0.3 | 10.0.0.3 | 1 |
| 127.0.0.0 | 255.0.0.0 | 127.0.0.1 | 127.0.0.1 | 1 |
| 224.0.0.0 | 224.0.0.0 | 10.0.0.3 | 10.0.0.3 | 1 |
| 224.0.0.0 | 224.0.0.0 | 110.0.0.3 | 110.0.0.3 | 1 |
| 255.255.255.255 | 255.255.255.255 | 10.0.0.3 | 10.0.0.3 | 1 |

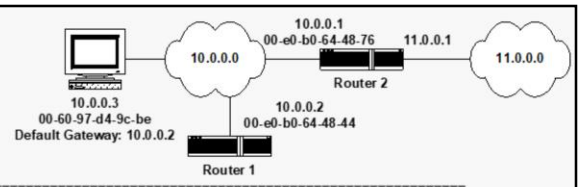
Tabela de encaminhamento inicial da estação 10.0.0.3

## ICMP Redirect example (I)

Consider the above example. Executing the 'route print' command on a DOS window (in the Windows OS) of the host 10.0.0.3, we can check the host routing table. The above routing table is a possible one for the network shown.

Note that if no other line matches a destination IP address, the first line states that the Default Gateway is the host with address 10.0.0.2 (Router 1) that can be reached through output interface 10.0.0.3 (its own network interface).

## Exemplo (II)




Active Routes:

| Network Destination | Netmask | Gateway | Interface | Metric |
|---|---|---|---|---|
| 0.0.0.0 | 0.0.0.0 | 10.0.0.2 | 10.0.0.3 | 1 |
| 10.0.0.0 | 255.0.0.0 | 10.0.0.3 | 10.0.0.3 | 1 |
| 10.0.0.3 | 255.255.255.255 | 127.0.0.1 | 127.0.0.1 | 1 |
| 10.255.255.255 | 255.255.255.255 | 10.0.0.3 | 10.0.0.3 | 1 |
| 11.0.0.1 | 255.255.255.255 | 10.0.0.1 | 10.0.0.3 | 1 |
| 127.0.0.0 | 255.0.0.0 | 127.0.0.1 | 127.0.0.1 | 1 |
| 224.0.0.0 | 224.0.0.0 | 10.0.0.3 | 10.0.0.3 | 1 |
| 224.0.0.0 | 224.0.0.0 | 110.0.0.3 | 110.0.0.3 | 1 |
| 255.255.255.255 | 255.255.255.255 | 10.0.0.3 | 10.0.0.3 | 1 |

**Tabela de encaminhamento final da estação 10.0.0.3**

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | Ok | 10.0.0.3 | 11.0.0.1 | ICMP | Type=Echo Request, ID=256, Seq No=1280 | 78 |
| 2 | Ok | 10.0.0.2 | 10.0.0.3 | ICMP | Type=Redirect,Code=for the Host,Gateway IP=10.0. | 74 |
| 3 | Ok | 10.0.0.2 | 10.0.0.3 | ICMP | Type=Redirect,Code=for the Net,Gateway IP=10.0.0 | 74 |
| 4 | Ok | 10.0.0.3 | 11.0.0.1 | ICMP | Type=Echo Request, ID=256, Seq No=1280 | 78 |
| 5 | Ok | 11.0.0.1 | 10.0.0.3 | ICMP | Type=Echo Reply, ID=256, Seq No=1280 | 78 |
| 6 | Ok | 006097D49CBE | Broadcast | ARP | Op=ARP Request, Src IP=10.0.0.3, Dest IP=10.0.0. | 46 |
| 7 | Ok | 00E0B0644876 | 006097D49CBE | ARP | Op=ARP Response, Src IP=10.0.0.1, Dest IP=10.0.0 | 64 |
| 8 | Ok | 10.0.0.3 | 11.0.0.1 | ICMP | Type=Echo Request, ID=256, Seq No=1536 | 78 |
| 9 | Ok | 11.0.0.1 | 10.0.0.3 | ICMP | Type=Echo Reply, ID=256, Seq No=1536 | 78 |
| 10 | Ok | 10.0.0.3 | 11.0.0.1 | ICMP | Type=Echo Request, ID=256, Seq No=1792 | 78 |
| 11 | Ok | 11.0.0.1 | 10.0.0.3 | ICMP | Type=Echo Reply, ID=256, Seq No=1792 | 78 |
| 12 | Ok | 10.0.0.3 | 11.0.0.1 | ICMP | Type=Echo Request, ID=256, Seq No=2048 | 78 |
| 13 | Ok | 11.0.0.1 | 10.0.0.3 | ICMP | Type=Echo Reply, ID=256, Seq No=2048 | 78 |

**Pacotes capturados na rede 10.0.0.0**

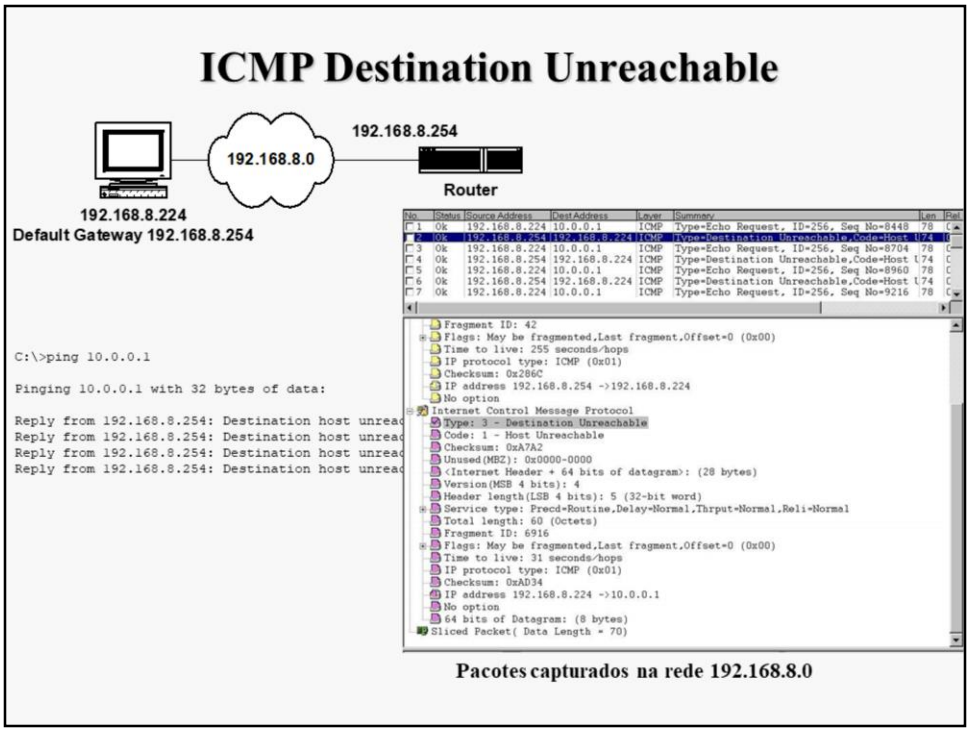
## ICMP Redirect example (II)

After running a ping command on host 10.0.0.3 for the remote address 11.0.0.1, the routing table of host 10.0.0.3 has now an additional line stating that the gateway to be used for the destination address 11.0.0.1 is 10.0.0.1.

Analysing the packets captured on network 10.0.0.0, we see that the first ICMP Echo Request is sent to Router 1, this router sends an ICMP Redirect message to the host 10.0.0.3 and forwards the ICMP Echo Request to Router 2. In the next ICMP Echo Requests, they are now sent directly to Router 2.

Router 1 has detected that the IP address 10.0.0.1 is a better route for the destination address 11.0.0.1 because: (i) the outgoing interface to forward the IP datagram is its incoming interface and (ii) its routing table indicates 10.0.0.1 as the next hop router address towards the destination network.

Note that if no ICMP Redirect was issued by Router 1, all IP datagrams sent from host 10.0.0.3 to other networks would be transmitted twice on the network 10.0.0.0.




## ICMP Destination Unreachable message

The ICMP Destination Unreachable message (type field is 3) is used when the destination of an IP datagram cannot be reached.

There are 6 possible values for the code field:

Code 0 - Net Unreachable - sent by a router if it does not know a route to the requested network.

Code 1 - Host Unreachable - sent by a router if it knows a route to the requested network but cannot reach the destination host.

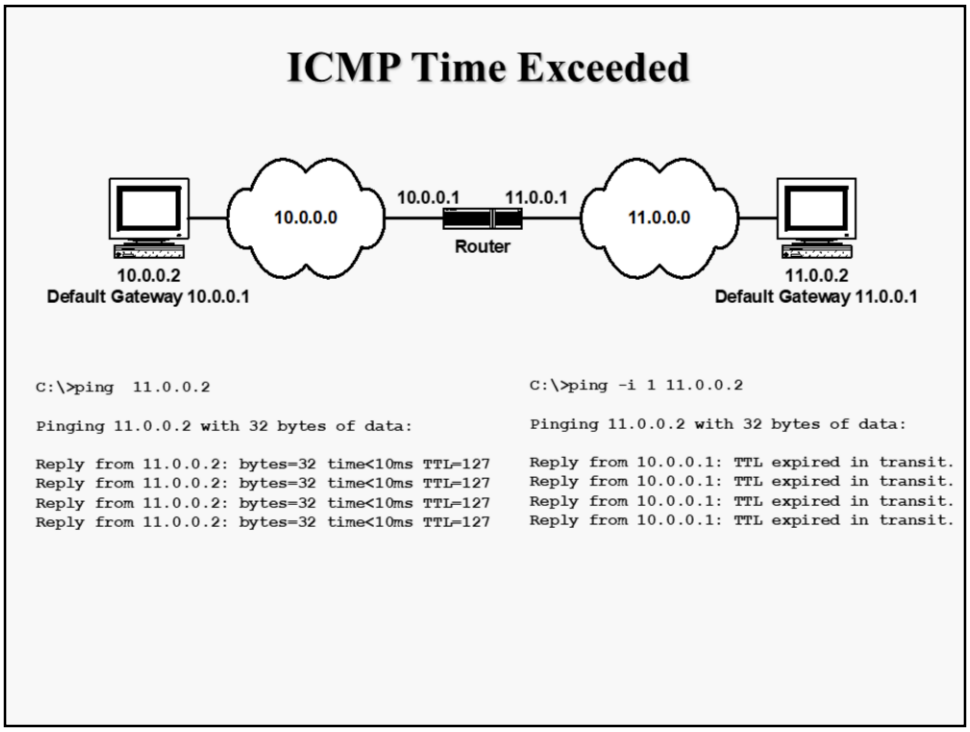
Code 2 - Protocol Unreachable – sent by the destination host if the destination protocol is not running.

Code 3 - Port Unreachable – sent by the destination host if no application is active on the destination port number.

Code 4 - Cannot Fragment - sent by a router if it needs to fragment an IP datagram but the 'do not fragment' bit is 1 in the IP header.

Code 5 - Source Route Failed - IP Source Routing is one of the IP Header Options.

In the above example, when running the ping command in the host for the IP address 192.168.8.254, the ICMP Echo Request messages reach the router that does not know how to reach the destination host. The ICMP Echo Requests are discarded and the Router sends to the origin host an ICMP Destination Unreachable message with code Host Unreachable. The outcome of the ping command indicates the IP address of the router reporting the situation.



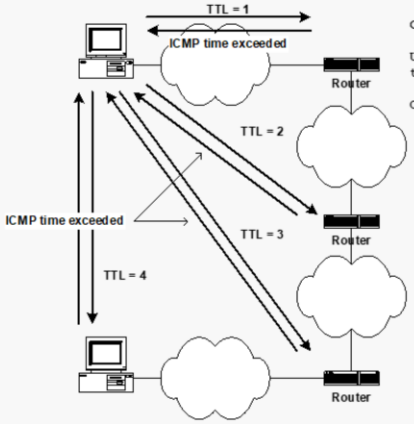

## ICMP Time Exceeded message

The ICMP Time Exceeded message (type field is 11) is send by a router to the origin host of an incoming IP datagram when it is discarded due to the fact that its TTL value reaches zero.

In the above example, if the ping is set with an origin TTL equal to 1, the first router discards the message and replies with an ICMP Time Exceeded message (indicated in the output of the ping command).

## Comando *tracert*

- **Permite descobrir o percurso utilizado no encaminhamento dos pacotes**
- **Recorre ao campo TTL e a mensagens ICMP Time Exceeded**




```
C:\>tracert

Usage: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout]
target_name

Options:
    -d                  Do not resolve addresses to hostnames.
    -h maximum_hops     Maximum number of hops to search for target
    -j host-list        Loose source route along host-list.
    -w timeout          Wait timeout milliseconds for each reply.
```

## *tracert* command

*tracert* command is a diagnosis tool for displaying routing paths and measuring transit delays of IP datagrams across the IP network.

When running the *tracert* command on an origin host to a destination IP address, the origin host starts sending 3 ICMP Echo Request messages with TTL = 1. For each of these messages, the first router replies with one ICMP Time Exceeded message (the three messages give three measures of the round-trip-time to the first router). Then, the origin host repeats the process with growing values of TTL until it receives ICMP Echo Replies from the destination host. For each value of TTL, a new router of the routing path is discovered and three measures of the round-trip-time are obtained for that router.

## Exemplo – *tracert*

```
C:\>tracert -d 193.136.173.30

Tracing route to 193.136.173.30 over a maximum of 30 hops

  1    <10 ms    <10 ms    <10 ms  193.136.92.1
  2    <10 ms    <10 ms    <10 ms  193.137.172.254
  3    <10 ms    <10 ms    <10 ms  193.136.173.30

Trace complete.
```

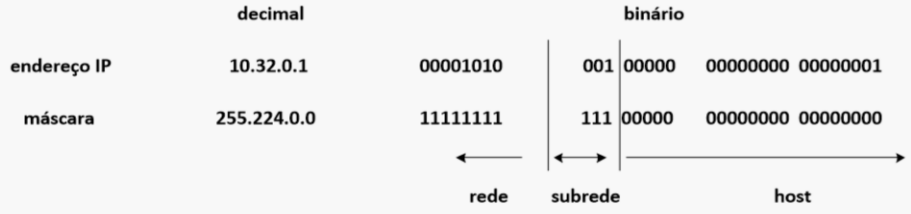
| No. | Source Address | Dest Address | Summary |
|---|---|---|---|
| 1 | [193.136.92.243] | [193.136.173.30] | ICMP: Echo |
| 2 | [193.136.92.1] | [193.136.92.243] | ICMP: Time exceeded |
| 3 | [193.136.92.243] | [193.136.173.30] | ICMP: Echo |
| 4 | [193.136.92.1] | [193.136.92.243] | ICMP: Time exceeded |
| 5 | [193.136.92.243] | [193.136.173.30] | ICMP: Echo |
| 6 | [193.136.92.1] | [193.136.92.243] | ICMP: Time exceeded |
| 7 | [193.136.92.243] | [193.136.173.30] | ICMP: Echo |
| 8 | [193.137.172.254] | [193.136.92.243] | ICMP: Time exceeded |
| 9 | [193.136.92.243] | [193.136.173.30] | ICMP: Echo |
| 10 | [193.137.172.254] | [193.136.92.243] | ICMP: Time exceeded |
| 11 | [193.136.92.243] | [193.136.173.30] | ICMP: Echo |
| 12 | [193.137.172.254] | [193.136.92.243] | ICMP: Time exceeded |
| 13 | [193.136.92.243] | [193.136.173.30] | ICMP: Echo |
| 14 | [193.136.173.30] | [193.136.92.243] | ICMP: Echo reply |
| 15 | [193.136.92.243] | [193.136.173.30] | ICMP: Echo |
| 16 | [193.136.173.30] | [193.136.92.243] | ICMP: Echo reply |
| 17 | [193.136.92.243] | [193.136.173.30] | ICMP: Echo |
| 18 | [193.136.173.30] | [193.136.92.243] | ICMP: Echo reply |

## ***tracert* example**

In the above capture, we can check that the display of the *tracert* command is in accordance with the IP addresses of the routers that have replied with ICMP Time Exceeded messages.

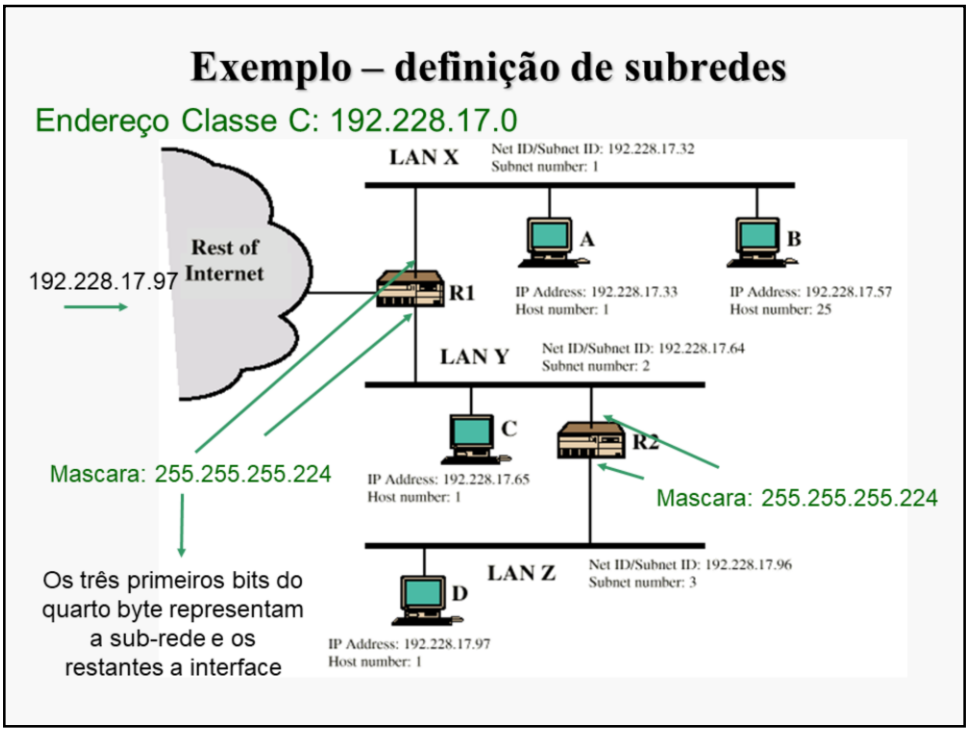
## Subredes

- **Uma subrede (subnet) é um subconjunto de uma rede de classe A, B ou C**
- **A utilização de máscaras, permite que uma rede seja dividida em subredes estendendo a parte de rede à parte de host do endereço IP; esta técnica aumenta o número de redes e reduz o número de hosts**




## IP subnets

Netmasks give more flexibility to network managers on the usage of the addressing space. A subnet of an original network address is defined when some bits of the original hostid part are used to define the netid part of the address.
In the above example, the IP address 10.32.0.1, defined with netmask 255.224.0.0, uses the first eleven bits to define the netid part and 21 bits to define the hostid part. It is an address defined on a subnet of class A network address 10.0.0.0 since it uses the three most significant bits (assigned with 001) of its hostid part to define the subnet.
In this way, a single class A network address can be organized in smaller subnets to be assigned to different physical networks.




## IP subnets example

In the above example, the class C network address 192.228.17.0 (netmask 255.255.255.0) was assigned to a client by its Internet Service Provider (ISP). The client has to assign addresses to hosts but its network is composed by three Local Area Networks (LAN X, LAN Y and LAN Z) separated by 2 routers (R1 and R2). Since it has to assign different netid parts to different LANs, it must resort to the segmentation of the assigned network address into multiple subnets.

By using netmask 255.255.255.224, three additional bits are available for netid definition (in a total of 27 bits). In the above example, the used network addresses are 192.228.17.32 (in LAN X), 192.228.17.64 (in LAN Y) and 192.228.17.96 (in LAN Z). Each host is assigned with an IP address whose 27 first bits (its netid part) are equal to the 27 first bits of its network address. Since there are 5 bits to identify hosts, each host can be identified by a number between 1 and 30 (remember why 0 and 31 cannot be used). The IP address of each host is obtained by adding the assigned number to the IP address of the network (for example, host B has the IP address 192.228.17.57 which results from adding the assigned number 25 to the network address 192.229.17.32).

## Questões sobre Máscaras de rede e sub-rede

- **Qual o endereço de broadcast da rede 175.0.115.128/25? E da rede 175.0.200.0 máscara 255.255.248.0?**
- **Qual a primeira máquina numa rede que tem uma máquina com o endereço 175.0.92.191/23?**
- **Qual a última máquina da rede 175.0.32.0 máscara 255.255.248.0?**
- **Qual a rede da máquina 175.0.22.79/25? E 175.0.117.215/23?**
- **Quantas redes e com quantas máquinas se obtêm na rede particionada como 175.0.4.0 máscara 255.255.255.252? E de 175.0.114.0 255.255.255.240?**

39