

# Manejo de IPs y máscaras en CIDR

Tomás Fernández Pena

José Carlos Cabaleiro Domínguez

Grado en Ingeniería Informática

Universidade de Santiago de Compostela

Redes, 2º Curso GrEI

[citius.usc.es](http://citius.usc.es)



Direcciones IP y máscaras



Centro Singular de Investigación  
en Tecnoloxías Intelixentes

Funciones de manejo de números de red

## Índice

**1** Direcciones IP y máscaras

**2** Funciones de manejo de números de red

# Índice

**1** Direcciones IP y máscaras

**2** Funciones de manejo de números de red

## Direccionamiento IPv4 tradicional

Clase	32 bits	Rango de direcciones
A	0 Red   Estación	1.0.0.1 hasta 126.255.255.254
B	10 Red   Estación	128.0.0.1 hasta 191.255.255.254
C	110 Red   Estación	192.0.0.1 hasta 223.255.255.254
D	1110 Dirección multicast	224.0.0.1 hasta 239.255.255.254
E	11110 Reservado para uso futuro	> 240.0.0.1

## Direccionamiento IPv4 con clases

- Restricción: un campo de red o de host no puede estar todo a 1s o todo a 0s
- Según esto tenemos:

**Clase A:** 126 ( $2^7 - 2$ ) redes con  $\approx$  16 millones de estaciones cada una  
 $(2^{24} - 2 = 16777214)$

**Clase B:** 16384 ( $2^{14}$ ) redes con 65534 ( $2^{16} - 2$ ) estaciones cada una

**Clase C:**  $\approx$  2 millones de redes ( $2^{21} = 2097152$ ) con 254 estaciones cada una  
 $(2^8 - 2)$

- En C existen funciones de manejo de las partes de red y de host

- ▷ `uint32_t inet_lnaof(), uint32_t inet_netof(), struct in_addr inet_makeaddr()`
- ▷ Solo funciona con IPv4 de clases A, B y C

## Direcciones especiales reservadas

- Identificación de redes: el nº de red y el resto a 0

- Ejemplos:

- ▷ Clase A → 10.0.0.0
- ▷ Clase B → 172.16.0.0
- ▷ Clase C → 193.144.84.0

- Dirección de broadcast: el nº de red y el resto a 1

- Ejemplos:

- ▷ Clase A → 10.255.255.255
- ▷ Clase B → 172.16.255.255
- ▷ Clase C → 193.144.84.255

## Subredes y máscaras

### Subredes

- Problema: el número de estaciones en una red puede ser demasiado grande ⇒ dificultades de administración
- Solución: dividir la red en subredes, que se gestionen de forma independiente pero que actúen como una sola de cara al exterior

### Máscara de subred

- Utilizamos parte del campo estación para indicar la subred
- Empleamos **máscaras** para delimitar la subred
- Formato de máscara: 32 bits de los que los  $n$  más significativos están a 1 y los  $32 - n$  restantes a 0
- Ejemplo: máscara de 27 bits, se denota como sufijo /27  
 $255.255.255.224 \equiv 11111111.11111111.11111111.11100000$

## Direccionamiento IPv4: ejemplo de máscara

- Dirección clase C 193.168.17.0/27 (o máscara 255.255.255.224)
  - ▷ Los 24 primeros bits indican la red (192.168.17)
  - ▷ Los 3 siguientes la subred
  - ▷ Los 5 últimos la posición de la estación en la subred
  - ▷ Tenemos  $2^3 = 8$  subredes, con  $2^5 - 2 = 30$  estaciones por subred
  - ▷ En total, podemos direccionar  $8 \times 30 = 240$  estaciones (254 en clase C sin máscara)

## Redes sin clase

- En 1993, se suprimen las clases
- Direcciones CIDR (**Classless Inter-Domain Routing**)
  - ▷ Sufijo  $/s \Rightarrow s$  bits para indicar la red y  $32 - s$  para indicar la estación ( $2^{(32-s)} - 2$  estaciones)
- Ejemplos: 193.168.64.0/18, 130.0.0.0/8
- También se conocen como **superredes**
- Ejemplo: 193.168.173.253/18
  - ▷ Nº de red: **11000001.10101000.10000000.00000000** = 193.168.128.0 (también se usa 193.168.128)
  - ▷ Broadcast: **11000001.10101000.10111111.11111111** = 193.168.191.255
  - ▷ Nº estación: **11000001.10101000.10101101.11111101** = estación nº 11773
  - ▷ Nº total de estaciones:  $2^{14} - 2 = 16382$

## Identificación de redes

- Para identificar las redes se especifica el número de red (con la parte de host a 0) y la máscara. Ejemplos:
  - ▷ 193.168.64.0/18
  - ▷ 130.0.0.0/8
  - ▷ Se pueden omitir los ceros de la parte de host. Por ejemplo:
    - 193.168.64/18
    - 130/8
- Lógicamente, en formato binario son equivalentes
- Los routers incluyen entradas de este tipo para saber hacia qué interfaz encaminar los paquetes
- En C hay funciones para manejar estas notaciones
  - ▷ `int inet_net_pton()` de textual a binario
  - ▷ `char *inet_net_ntop()` de binario a textual

# Índice

**1** Direcciones IP y máscaras

**2** Funciones de manejo de números de red

## Función `inet_ntop`: textual a binario

```
int inet_ntop(int af, const char *pres,
              void *netp, size_t nsize)
```

- Convierte a binario la IP especificada y devuelve el sufijo<sup>1</sup>
- Parámetros:
  - ▷ af entero que debe valer AF\_INET<sup>2</sup>
  - ▷ pres puntero a un string con la IP/sufijo a convertir
  - ▷ netp puntero al resultado, debe apuntar a una struct `in_addr`<sup>3</sup> y debe estar inicializada a 0
    - La IP se guarda en orden de red
  - ▷ nsize número de bytes en netp
- Valor devuelto: un entero indicando el sufijo especificado, -1 en caso de error
- Al compilar, usar la opción `-lresolv`

<sup>1</sup>Función no estándar pero ampliamente disponible

<sup>2</sup>Solo soporta IPv4

<sup>3</sup>Debe apuntar a un área reservada con el tamaño adecuado

## Función inet\_net\_pton: textual a binario

Ejemplo de uso:

```
struct in_addr mired;
mired.s_addr = 0;
if((sufijo = inet_net_pton(AF_INET, "193.20.102.40/14", (void *) &mired, sizeof(struct
    in_addr))) < 0) {
    fprintf(stderr, "Formato de direccion incorrecto");
    exit(EXIT_FAILURE);
}
printf(" %X %u\n", mired.s_addr, sufijo); // Imprime 286614C1 14

mired.s_addr = 0;
if((sufijo = inet_net_pton(AF_INET, "193.20.64/14", (void *) &mired, sizeof(struct in_addr)))
    < 0) {
    fprintf(stderr, "Formato de direccion incorrecto");
    exit(EXIT_FAILURE);
}
printf(" %X %u\n", mired.s_addr, sufijo); // Imprime 4014C1 14
```

## Función inet\_net\_ntop: binario a textual

```
char *inet_net_ntop(int af, const void *netp,
                    int bits, char *pres,
                    size_t psizes)
```

- Convierte el número red a formato presentación<sup>4</sup> (textual)
- Parámetros:
  - ▷ af igual que antes
  - ▷ netp puntero a una struct in\_addr
  - ▷ bits número de bits de la parte de red (el sufijo)
  - ▷ pres puntero a la cadena en la que se guardará el resultado<sup>5</sup>
  - ▷ psizes número de bytes disponibles en pres
- Si en netp se mete una IP completa, en pres guarda la parte de red/sufijo
- Valor devuelto: puntero a pres, NULL en caso de error
- Al compilar, usar la opción -lresolv

<sup>4</sup>Función no estándar pero ampliamente disponible

<sup>5</sup>Debe apuntar a un área reservada con el tamaño adecuado

## Función inet\_ntop: binario a textual

Ejemplo de uso:

```
struct in_addr mired, miip;
char red_text[INET_ADDRSTRLEN+3];

mired.s_addr = 0x4014C1; // 193.20.64.0
miip.s_addr = 0x286614C1; // 193.20.102.40
if(inet_ntop(AF_INET, (void *) &mired, 14, red_text, INET_ADDRSTRLEN+3) != NULL) {
    printf("%s\n", red_text); // Imprime 193.20/14
}

if(inet_ntop(AF_INET, (void *) &miip, 12, red_text, INET_ADDRSTRLEN+3) != NULL) {
    printf("%s\n", red_text); // Imprime 193.16/12
}
```