

# Indirizzamento IP

## Networking tools

**Liceo G.B. Brocchi – Bassano del Grappa (VI)**  
**Liceo Scientifico – opzione scienze applicate**  
Giovanni Mazzocchin

# Indirizzi IPv4

- Un indirizzo IPv4 è definito **indirizzo logico** perché non è rigidamente vincolato all'interfaccia di una macchina (come gli indirizzi MAC)
- È possibile infatti:
  - assegnare, in tempi diversi, uno stesso indirizzo IPv4 a due macchine diverse
  - assegnare alla stessa macchina indirizzi IPv4 diversi in tempi diversi

# Indirizzi IPv4

- Ad ogni interfaccia di rete (o host) di un dispositivo connesso va assegnato un indirizzo IPv4, con le seguenti caratteristiche:
  - permette di individuare la rete (o sottorete) IP a cui appartiene l'host
  - permette di individuare l'host stesso all'interno della rete
  - è costituito da 32 bit (4 ottetti)

<b>dotted decimal</b>	172	.	15	.	2	.	4
<b>binary</b>	10101100		00001111		00000010		00000100

# Indirizzi IPv4

- Un indirizzo IPv4 è composto da:
  - il **network prefix**, che permette di determinare l'indirizzo IPv4 della subnet (rete) a cui è connesso l'host
  - la **parte host**, che permette di identificare l'host all'interno della subnet
- Una rete IPv4 viene definita tramite l'assegnazione agli host di indirizzi IPv4 con lo stesso network prefix

# Metodo *classful*

- È stato il primo metodo (ideato nel 1981) per determinare i prefissi di rete degli indirizzi IPv4
- Indicazioni sulla parte host:
  - è lunga  $h$ :  $32 - (\text{network prefix length})$  bit
  - il numero massimo di host che possono appartenere ad una subnet IPv4 è quindi  $2^h$
  - la parte host con tutti i bit a 0 è riservata all'indirizzo di rete
  - la parte host con tutti i bit a 1 è riservata alle comunicazioni broadcast
    - ne consegue che **il numero massimo di host all'interno di una subnet IPv4 è uguale a  $2^h - 2$**

# Metodo *classful*

class name	MSBs* (fixed)	network prefix length		n. networks	host part length		n. hosts per subnet	
<b>A</b>	0	8 bits		$2^7$	24 bits		$2^{24} - 2$	
<b>B</b>	10	16 bits		$2^{14}$	16 bits		$2^{16} - 2$	
<b>C</b>	110	24 bits		$2^{21}$	8 bits		$2^8 - 2$	
<b>D**</b>								
<b>E**</b>								

\*: *most significant bits*

\*\* : le classi D ed E non saranno trattate

# Metodo *classful*

class	start address	end address	subnet mask
<b>A</b>	0.0.0.0	127.255.255.255	255.0.0.0 /8
<b>B</b>	128.0.0.0	191.255.255.255	255.255.0.0 /16
<b>C</b>	192.0.0.0	223.255.255.255	255.255.255.0 /24

# Metodo *classful* – subnet mask, wildcard mask

<b>dotted decimal</b>	10	.	2	.	56	.	21
<b>binary</b>	00001010		00000010		00111000		00010101
<b>subnet mask (dotted decimal)</b>	255	.	0	.	0	.	0
<b>subnet mask (binary)</b>	11111111		00000000		00000000		00000000

- la subnet mask permette di estrarre il network address da un indirizzo IPv4
- l'inverse mask (o wildcard mask) permette di ricavare l'indirizzo di broadcast della subnet



# Metodo *classful* – subnet mask, wildcard mask

0	0	0	0	1	0	1	0		0	0	0	0	0	0	1	0		0	0	1	1	1	0	0	0		0	0	0	1	0	1	0	1
1	1	1	1	1	1	1	1		0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0
0	0	0	0	1	0	1	0		0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0

**AND**

indirizzo IPv4 (prima riga): 10.2.56.21

subnet mask (seconda riga): 255.0.0.0

network address (terza riga): 10.0.0.0

# Metodo *classful* – subnet mask, wildcard mask

0	0	0	0	1	0	1	0		0	0	0	0	0	0	1	0		0	0	1	1	1	0	0	0		0	0	0	1	0	1	0	1
0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
0	0	0	0	1	0	1	0		1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1

**OR**

indirizzo IPv4 (prima riga): 10.2.56.21

wildcard mask (seconda riga): 0.255.255.255

broadcast address (terza riga): 10.255.255.255

# Metodo *classless*

- La suddivisione in classi determina uno spreco di indirizzi, in quanto il numero di host che costituiscono una rete IPv4 può essere inferiore al numero massimo di host previsto per una classe
- **CIDR** (**C**lassless **I**nter-**D**omain **R**outing, 1993) ha introdotto il metodo **classless**, che ha abolito la suddivisione in classi descritta sopra
- Con CIDR, il prefisso di rete può avere lunghezza qualsiasi, entro i 32 bit
- Notazione:
  - 10.10.20.20/24
  - 172.16.16.18/29
  - 129.168.179.172/30

# Metodo *classless*

- **Esempio 1:**

- determinare l'indirizzo della subnet a cui appartiene 10.10.20.20/24
- I 4 ottetti in binario sono:
  - 00001010    00001010    00010100    00010100
- I 4 ottetti della subnet mask e della wildcard mask in binario sono:
  - 11111111    11111111    11111111    00000000
  - 00000000    00000000    00000000    11111111
- In notazione decimale puntata, la subnet mask e la wildcard mask sono:
  - 255.255.255.0
  - 0.0.0.255

# Metodo *classless*

- Si esegue l'AND bitwise tra l'indirizzo e la subnet mask:

00001010	00001010	00010100	00010100	&
11111111	11111111	11111111	00000000	=
<b>00001010</b>	<b>00001010</b>	<b>00010100</b>	<b>00000000</b>	

- Si converte il risultato in decimale:
  - 10.10.20.0
- Quindi:
  - l'indirizzo della subnet è 10.10.20.0/24

# Metodo *classless*

- Si esegue l'OR bitwise tra l'indirizzo e la wildcard mask:

00001010	00001010	00010100	00010100	
00000000	11111111	11111111	11111111	=
<b>00001010</b>	<b>11111111</b>	<b>11111111</b>	<b>11111111</b>	

- Si converte il risultato in decimale:
  - 10.255.255.255
- Quindi:
  - l'indirizzo di broadcast è 10.255.255.255/24

# Metodo *classless*

- **Esempio 2:**

- determinare l'indirizzo della subnet a cui appartiene 172.16.16.18/29
- I 4 ottetti in binario sono:
  - 10101100 00010000 00010000 00010010
- I 4 ottetti della subnet mask in binario sono:
  - 11111111 11111111 11111111 11111000
- In notazione decimale puntata, la subnet mask è quindi:
  - 255.255.255.248

# Metodo *classless*

- Si esegue l'AND bitwise tra l'indirizzo e la subnet mask:

10101100	00010000	00010000	00010010	&
11111111	11111111	11111111	11111000	=
<b>10101100</b>	<b>00010000</b>	<b>00010000</b>	<b>00010000</b>	

- Si converte il risultato in decimale:
  - 172.16.16.16
- Quindi:
  - l'indirizzo della subnet è 172.16.16.16/29



# Metodo *classless*

- Si esegue l'OR bitwise tra l'indirizzo e la wildcard mask:

10101100	00010000	00010000	00010010	
00000000	00000000	00000000	00000111	=
<b>10101100</b>	<b>00010000</b>	<b>00010000</b>	<b>00010111</b>	

- Si converte il risultato in decimale:
  - 172.16.16.23
- Quindi:
  - l'indirizzo di broadcast è 172.16.16.23/29

# Indirizzi IPv4 riservati

- `0.0.0.0`: utilizzato come indirizzo sorgente da un host a cui non è ancora stato assegnato un indirizzo IPv4
- `127.0.0.1`: indirizzo di loopback, utilizzato per verificare se sul proprio computer funziona la suite TCP/IP, oppure per accedere ad un servizio installato sul proprio computer (vi ricordate `localhost`?)

# Indirizzi IPv4 pubblici

- Sono gli indirizzi che permettono la comunicazione tra host attraverso Internet
- Sono unici a livello mondiale e assegnati da una gerarchia di autorità ([IANA](#), i vari [RIR](#))

# Indirizzi IPv4 privati

- Gli indirizzi IPv4 privati sono utilizzabili per host appartenenti a reti private
- Un host non può accedere ad Internet tramite il proprio indirizzo privato
- Gli indirizzi privati devono essere univoci all'interno di una rete privata, ma possono esistere due host con lo stesso indirizzo privato appartenenti a due reti private diverse interconnesse
- Capiremo meglio il significato di questi indirizzi vedendo la funzione **NAT** (***N**etwork **A**ddress **T**ranslation*)
- Sono costituiti da 3 blocchi di indirizzi:
  - 10.0.0.0/8
  - 172.16.0.0/12
  - 192.168.0.0/16

# IPv6

- **IPv6** - **IPng** (***IP** next **g**eneration*) è l'evoluzione di IPv4. Esso supera alcuni limiti di IPv4, dei quali il più evidente è lo spazio di indirizzi limitato (CIDR e NAT mitigano il problema, ma non lo risolvono)
- IPv6 prevede indirizzi a 128 bit
- Le novità introdotte da IPv4 sono molte e non riguardano soltanto la lunghezza degli indirizzi, ma non le vedremo

# IPv6

- Gli indirizzi IPv6 sono indirizzi a 128 bit (16 byte), espressi in esadecimale
- Rappresentazione: 8 campi da 2 byte, separati da :
- Esempio:
  - fe80:0000:0000:0000:9a5e:b268:d3fb:2932

# Il comando ping

- ping è un tool che permette di verificare se un host è raggiungibile su una rete IP (si basa su **ICMP** - *Internet Control Message Protocol*)
- Provare:

- ping `www.google.com`

- output:

```
PING www.google.com (142.251.209.36) 56(84) bytes of data.  
64 bytes from mil04s51-in-f4.1e100.net (142.251.209.36):  
icmp_seq=1 ttl=112 time=37.5 ms  
64 bytes from mil04s51-in-f4.1e100.net (142.251.209.36):  
icmp_seq=2 ttl=112 time=32.7 ms  
64 bytes from mil04s51-in-f4.1e100.net (142.251.209.36):  
icmp_seq=3 ttl=112 time=43.2 ms
```

...

# Il comando traceroute

- traceroute è un tool che permette di vedere la route di un pacchetto su una rete IP
- Provare:
  - `traceroute english.pku.edu.cn`



# Da vedere/leggere/visitare a casa

- <https://geolocation-db.com/>
- [\*\*How Traceroute Works \(Building a Movie Scene 'Trace' Map\) - Computerphile\*\*](#)