



IP Addresses
subnetting
Reti di Accesso 2006/2007

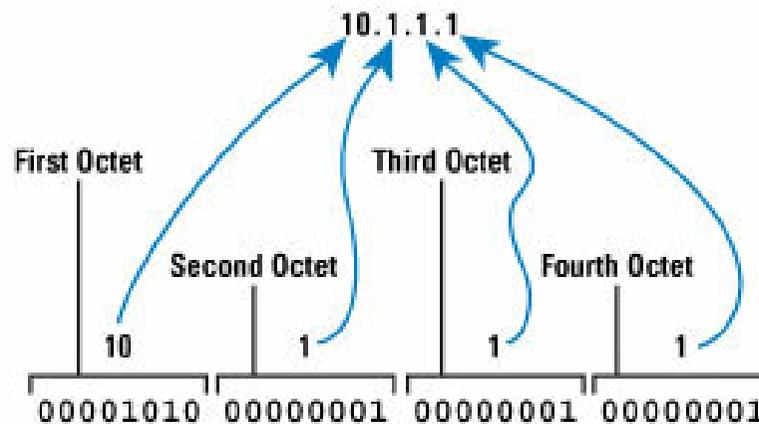


Metodo non convenzionale

- Presentiamo qui un modo non convenzionale per lavorare con indirizzi IPv4 (e IPv6).
- Spesso, infatti, ragionare (troppo) in termini di bit non è sufficientemente veloce per prendere decisioni “on the fly” (e per rispondere ai quesiti di esame!)

Review IPv4 addressing

- Gli indirizzi IPv4 sono essenzialmente stringhe numeriche a 32 bit (IPv6 a 128 bit).
- Per renderli più umanamente comprensibili li separiamo in gruppi da 8 bit, gli “octets” (router ed elaboratori non effettuano separazioni di sorta).

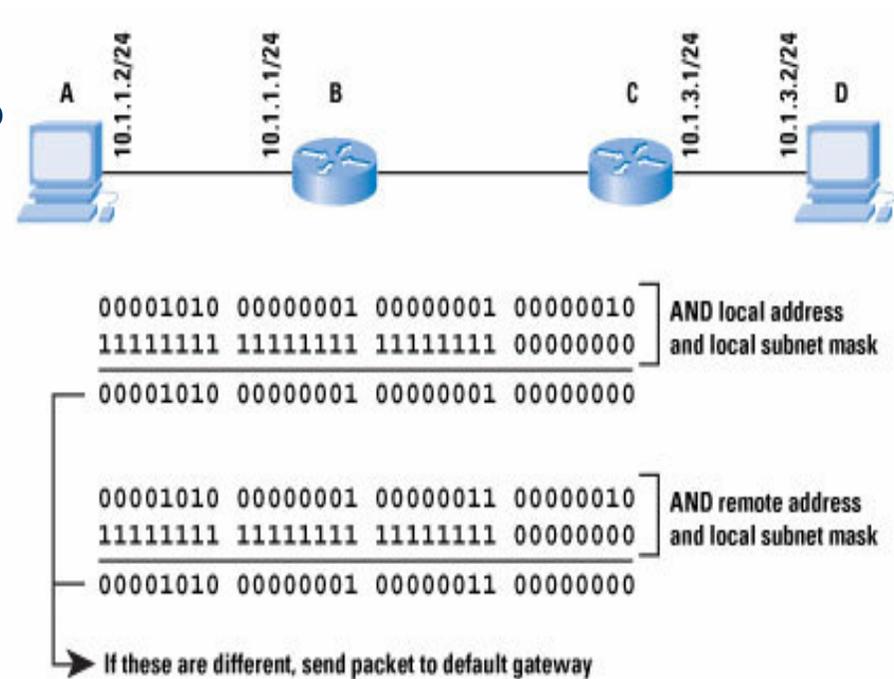


Review IPv4 addressing

- Naturalmente, ogni ottetto (byte) rappresenta un numero decimale fino a $2^8=256$ (0→255).
- Le cose possono complicarsi quando si ragiona in termini di “subnet” e si introducono i concetti di “subnet mask”, “supernetting” e “sub-subnet”.

Review IPv4 addressing -subnet mask

- Se l'host A, con indirizzo 10.1.1.2 vuole mandare un pacchetto all'host B, di indirizzo 10.1.3.2, come puo' capire se deve effettuare una richiesta ARP (stesso dominio di collisione ethernet) oppure inviare la richiesta al Gateway per quella sottorete (o al gateway di default)?
- A effettua un AND logico tra :
 1. *Indirizzo proprio e subnet mask*
 2. *Indirizzo destinazione e subnet mask*
 3. *Verifica se i due risultati sono uguali*Se lo sono, effettua una richiesta ARP per l'indirizzo di destinazione, altrimenti instrada il pacchetto verso il gateway per la sottorete di destinazione (facendo un ARP se occorre).



Review IPv4 addressing

-subnet mask: dotted decimal and prefix length

- La subnet mask può essere espressa secondo diverse notazioni. La *prefix length* (a.b.c.d/x) è comunque da preferire e viene utilizzata negli apparati di rete di nuova generazione.
- La *dotted decimal* (es. 255.255.255.0) è utilizzata nella configurazione dei PC ed in alcuni protocolli di routing, che non hanno (ancora) lo spazio per il *prefix*, ma memorizzano i byte della subnet mask.
- Conviene imparare a ricordare velocemente gli ultimi numeri (decimali) della tabella seguente!

Review IPv4 addressing

-subnet mask: dotted decimal and prefix length

Binary Mask	Prefix Length	Subnet Mask
11111111 00000000 00000000 00000000	/8	255.0.0.0
11111111 10000000 00000000 00000000	/9	255.128.0.0
11111111 11000000 00000000 00000000	/10	255.192.0.0
11111111 11100000 00000000 00000000	/11	255.224.0.0
11111111 11110000 00000000 00000000	/12	255.240.0.0
11111111 11111000 00000000 00000000	/13	255.248.0.0
11111111 11111100 00000000 00000000	/14	255.252.0.0
11111111 11111110 00000000 00000000	/15	255.254.0.0
11111111 11111111 00000000 00000000	/16	255.255.0.0
11111111 11111111 10000000 00000000	/17	255.255.128.0
11111111 11111111 11000000 00000000	/18	255.255.192.0
11111111 11111111 11100000 00000000	/19	255.255.224.0
11111111 11111111 11110000 00000000	/20	255.255.240.0
11111111 11111111 11111000 00000000	/21	255.255.248.0
11111111 11111111 11111100 00000000	/22	255.255.252.0
11111111 11111111 11111110 00000000	/23	255.255.254.0
11111111 11111111 11111111 00000000	/24	255.255.255.0
11111111 11111111 11111111 10000000	/25	255.255.255.128
11111111 11111111 11111111 11000000	/26	255.255.255.192
11111111 11111111 11111111 11100000	/27	255.255.255.224
11111111 11111111 11111111 11110000	/28	255.255.255.240
11111111 11111111 11111111 11111000	/29	255.255.255.248
11111111 11111111 11111111 11111100	/30	255.255.255.252
11111111 11111111 11111111 11111110	/31	255.255.255.254
11111111 11111111 11111111 11111111	/32	255.255.255.255

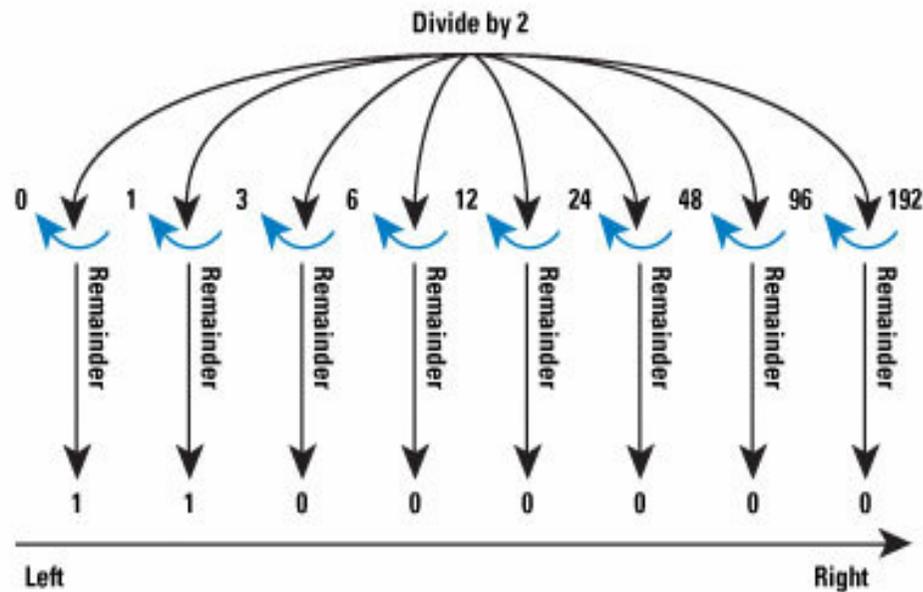
Working with IPv4: the hard way

- Per eseguire delle decisioni di routing, o per prevederle in una simulazione, dovremmo compiere i seguenti passi:
 1. Convertire l'indirizzo *sorgente* e *destinazione* in binario
 2. eseguire l'AND logico dei due risultati con la subnet mask
 3. verificare l'uguaglianza eventuale e prevedere l'utilizzo del gateway scorrendo la tabella di routing

Working with IPv4: the hard way

-esempio con un indirizzo sorgente

- Ad esempio, 192.168.100.80/26, conversione in binario di 192:



Working with IPv4: the hard way

-esempio con un indirizzo sorgente

- La /26 si converte in 255.255.255.192
- L'indirizzo di rete (subnet) si calcola facendo un AND con la subnet mask
- L'indirizzo relativo di Host all'interno della sottorete si calcola facendo un NOR con la subnet mask
- Due host diversi della stessa subnet si parlano senza utilizzare un gateway.
- Nel nostro esempio, 192.168.100.80 può parlare direttamente con i 2^6-2 host della sottorete 192.168.100.64 (192.168.100.65 ► 192.168.100.127)

Network	11000000	10101000	01100100	01010000
	192	168	100	80
<hr/>				
	11111111	11111111	11111111	11000000
	8	+8	+8	+2 == 26
AND	11000000	10101000	01100100	01000000
	192	168	100	64
Host	11000000	10101000	01100100	01010000
	192	168	100	80
<hr/>				
	11111111	11111111	11111111	11000000
	8	+8	+8	+2 == 26
NOR	00000000	00000000	00000000	00010000
	0	0	0	16

Working with IPv4: the *easy way*

- *Very nice* finora, ma vi verrà chiesto di configurare e pensare delle reti in pochi secondi, lavorando con degli indirizzi a.b.c.d/x e con delle subnet mask espresse in *dotted decimal*.
- Passi necessari:
 1. Individuare l'ottetto di interesse (*working octect*) partendo da un indirizzo con *prefisso*
 2. Trovare il "*salto*" relativo individuando la sottorete a cui appartiene il nostro indirizzo

Working with IPv4: the *easy way* *-working octect*

- L'ottetto di lavoro è individuato dal *prefix*.
- Per prefissi da /0 a /7 l'ottetto è il primo, per prefissi da /8 a /15 è il secondo, da /16 a /23 è il terzo, da /24 a /31 è il quarto.
- Ovvero: $WorkingOctect = (prefix_length) \div (8) + 1$
- Ad esempio, se il prefisso è /26 o /16 viene individuato il *quarto* ottetto o il *terzo* ottetto di interesse, rispettivamente

Nota: $x \div (8)$ è la divisione senza resto $x/8$

Working with IPv4: the *easy way*

-individuare le sottoreti

- Per individuare la sottorete di interesse alla quale appartiene l'host considerato, espresso nella forma a.b.c.d/x, occorre individuare la prima sottorete della quale saranno multiple le successive (ovvero quella immediatamente seguente la .0)
 - Se il *prefix* è un multiplo di 8, una sottorete parte da .0
 - Se il (*prefix-1*) è un multiplo di 8, una prima sottorete (dopo la .0) parte da .128
 - Se il (*prefix-2*) è un multiplo di 8, una prima sottorete parte (dopo la .0) da .64
 - ...
 - Ovvero, $prefix_length \bmod(8)$ individua, nell'ordine, uno tra i numeri 128,64,32,16,8,4,2,0
 - Ad esempio, se il prefisso è /26 viene individuato il numero .64, indirizzo di partenza della prima sottorete dopo la "base" .0 .
 - Un volta individuato il numero di interesse, si stabilisce la rete alla quale appartiene il nostro IP scegliendo tra i multipli del risultato precedente.
 - Ad esempio, nel caso del nostro esempio, potremmo scegliere una tra le reti di indirizzo base 0,64,128,192 e se l'indirizzo fosse 10.1.1.129/26, la sottorete di interesse sarebbe la 10.1.1.128.

Working with IPv4: the *easy way*

-individuare le sottoreti: esempio 1

- Individuare la sottorete di 192.158.100.80/26
 - L'ottetto di interesse è il quarto (con maschera 11000000)
 - $26 \bmod(8) = 2$; individuiamo quindi la prima sottorete .64; le reti utili sono la 192.158.100.0, 192.158.100.64, 192.158.100.128, 192.158.100.192.
 - Sono evidentemente, in tutto, $256/64=4$, oppure 2 elevato al numero di bit pari ad uno della mask dell'ottetto di interesse.
 - La sottorete che ci riguarda è la 192.158.100.64, l'indirizzo di host relativo è $80-64=16$.
 - Un altro sistema semplice per individuare subito la grandezza della rete (il numero di host possibili) è considerare il numero di bit posti a 0 della parte di mask relativa all'ottetto di interesse ed ai suoi successivi : nel nostro caso sono 6, quindi $2^6=64$

Working with IPv4: the *easy way* *-individuare le sottoreti: esempio 2*

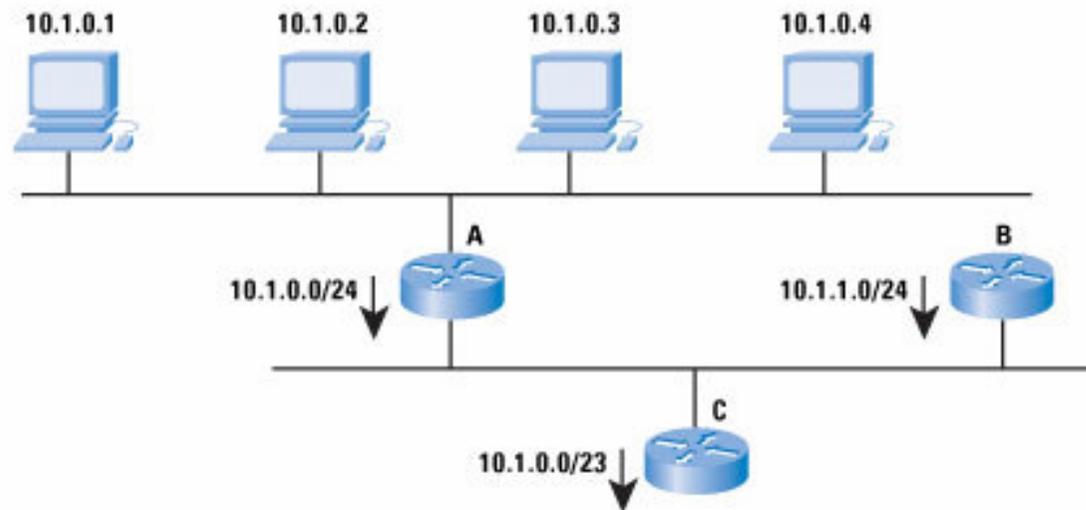
- Individuare la sottorete di 10.1.1.48/23
 - L'ottetto di interesse è il terzo (con maschera 11111110)
 - $23 \bmod(8) = 7$; individuiamo quindi la prima sottorete .2; le reti utili sono la 10.1.0.0, 10.1.2.0, 10.1.4.0 etc. notare che la prima sottorete contiene gli indirizzi da 10.1.0.0 a 10.1.1.255.
 - Sono evidentemente, in tutto, $256/2=128$, oppure 2 elevato al numero di bit pari ad uno della mask dell'ottetto di interesse.
 - La rete che ci riguarda è la 10.1.0.0, l'indirizzo di host relativo è “.1.48”, ovvero $256*1+48=304$.
 - Un altro sistema semplice per individuare subito la grandezza della rete (il numero di host possibili) è considerare il numero di bit posti a 0 della parte di mask relativa all'ottetto di interesse ed ai suoi successivi: nel nostro caso sono 9, quindi $2^9=512$.

Working with IPv4: the *easy way* *-individuare le sottoreti: esempio 3*

- Individuare la sottorete di 172.31.80.10/22
 - L'ottetto di interesse è il terzo (con maschera 11111100)
 - $22 \bmod(8) = 6$; individuiamo quindi la prima sottorete .4; le reti utili sono la 172.31.0.0, 172.31.4.0 etc. notare che la prima sottorete contiene gli indirizzi da 172.31.0.0 a 172.31.3.255.
 - Sono evidentemente, in tutto, $256/4=64$, oppure 2 elevato al numero di bit pari ad uno della mask dell'ottetto di interesse.
 - La rete che ci riguarda è la 172.31.80.0, l'indirizzo di host relativo è “.0.10”, ovvero $256*0+10=10$.
 - Un altro sistema semplice per individuare subito la grandezza della rete (il numero di host possibili) è considerare il numero di bit posti a 0 della parte di mask relativa all'ottetto di interesse ed ai suoi successivi: nel nostro caso sono 10, quindi $2^{10}=1024$.

Summarization e Subnets

Una *supernet* è una rete che contiene interamente un'altra rete. Per esempio, 10.1.0.0/24 e 10.1.1.0/24 sono sottoreti di 10.1.0.0/23, mentre 10.1.0.0/23 è una *supernet* di 10.1.0.0/24 e 10.1.1.0/24



Summarization e Subnets

- esempio in binario

- Facendo riferimento alla figura precedente:
- Esprimendo 10.1.0.0/24 e 10.1.1.0/24 in binario, si nota che cambia soltanto il bit numero 24. Se noi diminuiamo il *prefix* portandolo a 23, notiamo che abbiamo “mascherato” questo singolo bit, facendo in modo che 10.1.0.0/23 copra lo stesso range di indirizzi di 10.1.0.0/24 e 10.1.1.0/24 combinati assieme.

00001010	00000001	00000000	00000000	10.1.0.0/24
11111111	11111111	11111111	00000000	
00001010	00000001	00000001	00000000	10.1.1.0/24
11111111	11111111	11111111	00000000	
00001010	00000001	00000000	00000000	10.1.0.0/23
11111111	11111111	11111110	00000000	

Changing Bit

Un problema tipico di subnetting

- *Abbiamo 5 sottoreti con i seguenti numeri di host: 58, 14, 29, 49, e 3, ed abbiamo lo spazio di indirizzi 160.80.1.0/24. Determinare la suddivisione in sottoreti distinte.*
- **Passi necessari:**
 1. Ordinare le reti dalla più grande alla più piccola.
 2. Trovare la minima sottorete che soddisfa i requisiti del numero di host necessari + K dove K è, a seconda dei casi, 0, 2, 3...
 3. Per i link point-to-point è possibile utilizzare anche una /31, che ha due indirizzi di host disponibili e quindi non c'è l'indirizzo "di rete" e l'indirizzo di broadcast.
 4. Continuare ad allocare sottoreti fino ad esaurire il numero di sottoreti possibili.

Un problema tipico di subnetting -considerazioni preliminari

- Considerazioni preliminari:
 - K può avere valore:
 - 0 nel caso di link punto punto, come specificato (nel caso /31). Nel caso di point-to-point non sono necessari indirizzi di broadcast e di rete.
 - 2 nel caso di subnet generiche, in cui un indirizzo è riservato per la rete stessa e l'altro per l'indirizzo di broadcast
 - 3,4,.. quando nella sottorete assegnata è necessario configurare indirizzi di firewall, switch attivi necessari al funzionamento della subnet.
 - E' necessario ordinare le sottoreti dalla più grande alla più piccola in quanto se facessimo il contrario dovremmo spezzare la rete più grande in più sottoreti di dimensione troppo piccola e tutte uguali, mentre partendo dalla più grande è possibile fare delle subnet di subnet (sub-subnet che saranno evidenti poi con l'esempio).
 - E' necessario, nella pratica creare delle sottoreti *layer3* perché:
 - il traffico locale a un segmento può essere mantenuto locale, anche per la *security*
 - Esistono tecnologie di rete incompatibili tra loro (esempio ADSL/ATM, ethernet)
 - Nel caso di grosse reti Layer2, anche con bridge/switch, il traffico broadcast e multicast viene inoltrato su tutti i segmenti ma può essere fermato dai router
 - Si decide di implementare delle *vlan*

Un problema tipico di subnetting

-svolgimento dell'esempio

- Riordinare i numeri 58, 14, 29, 49, 3 in 58, 49, 29, 14, 3.
- Si parte con 58.
 - Il numero più piccolo maggiore di $(58+2)$ è 64; 64 rappresenta 2^6 host e 2^2 sottoreti.
 - Aggiungendo i due bit di mask alla rete data otteniamo un primo prefisso di /26: la prima rete è 160.80.1.0/26.
 - La seconda rete è: $160.80.1.0 + 64$, che abbiamo a disposizione a partire da 160.80.1.65.
- Il blocco successivo è composto da 49 host:
 - Il numero più piccolo maggiore di $(49+2)$ è 64; 64 rappresenta 2^6 host e 2^2 sottoreti.
 - Aggiungendo i due bit di mask alla rete data otteniamo un secondo prefisso /26: la seconda rete è 160.80.1.64/26.
 - La rete successiva è $160.80.1.64 + 64$, che abbiamo a disposizione a partire da 160.80.1.129
- Il blocco successivo è composto da 29 host:
 - Il numero più piccolo maggiore di $(29 + 2)$ è 32; 32 rappresenta 2^5 host e, quindi, 2^3 sottoreti.
 - Aggiungendo i tre bit di mask alla rete data otteniamo un terzo prefisso /27: la terza rete è 160.80.1.128/27 .
 - La prossima rete disponibile è a questo punto $160.80.1.128 + 32$, che abbiamo a disposizione a partire da 160.80.1.161.

Un problema tipico di subnetting

-svolgimento dell'esempio

- Il blocco successivo è composto da 14 host:
 - Il numero più piccolo maggiore di $(14 + 2)$ è 16; 16 rappresenta 2^4 host e 2^4 sottoreti.
 - Aggiungendo i 4 bit di mask alla rete data otteniamo un quarto prefisso /28: la quarta rete è 160.80.1.160/28.
 - La rete successiva è $160.80.1.160 + 16$, che abbiamo disponibile a partire dall'indirizzo 160.80.1.177.

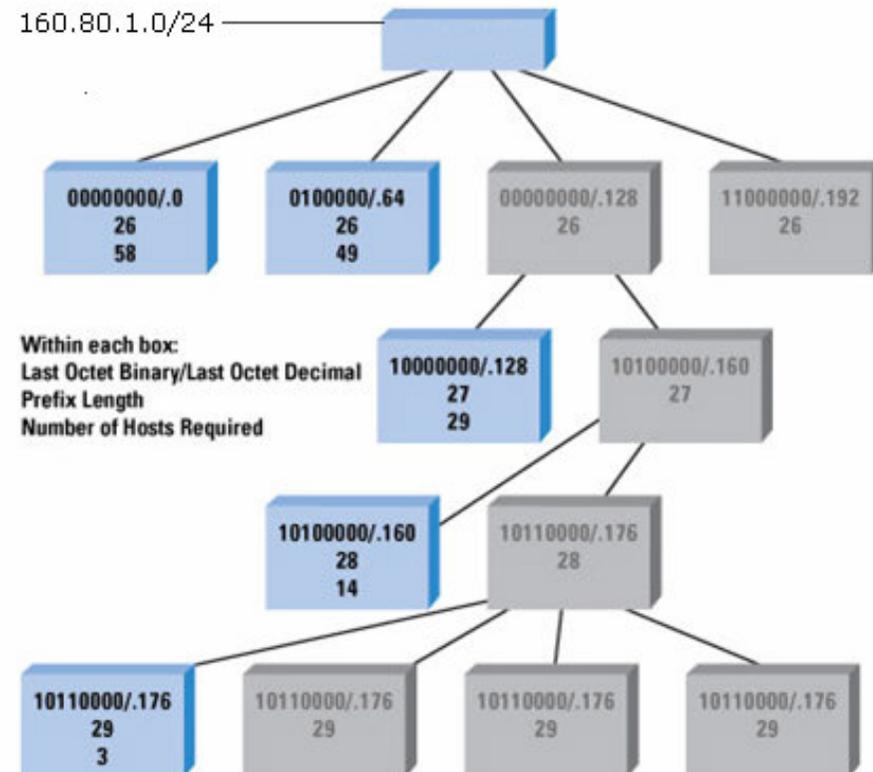
 - L'ultimo blocco è composto da 4 host:
 - Il numero più piccolo maggiore di $(3 + 2)$ è 8: 8 rappresenta 2^3 host e quindi 2^5 sottoreti.
 - Aggiungendo 5 bit di mask alla rete data otteniamo un quinto prefisso /29: la quinta rete è 160.80.1.176/29
 - Avendo terminato il numero di host da allocare, il processo è concluso.

Un problema tipico di subnetting -rappresentazione grafica

In grigio sono evidenziate le reti ancora da allocare

Notare che, partendo dalla sottorete più piccola, non avremmo potuto ottimizzare lo spazio utilizzato e fare aggregazione delle tabelle di routing.

In pratica è molto difficile pre-allocare le sottoreti nel tempo ed è difficile "ordinarle" come noi abbiamo fatto: per questo rimangono inutilizzati molti indirizzi ip, considerando anche un margine di crescita del parco macchine non immediatamente preventivabile.



Indirizzi IPv6 e subnetting

- Gli indirizzi IPv6 sono composti da 128 bit, espressi graficamente non come ottetti, ma come quadruple di cifre esadecimali, rappresentanti ognuna 4 bit. Gli ottetti possono essere estratti da coppie di cifre esadecimali.
- Ad esempio, *2002:FF10:9876:DD0A:9090:4896:AC56:0E01* è un indirizzo IPv6.
- Potrebbe anche essere indicato da 16 ottetti separati da punti, ma la grafia in uso, più pratica anche per i calcoli, è quella indicata.
- Con IPv6 sparisce sostanzialmente il problema del subnetting e del calcolo, dal momento che alloca *sempre* una /64 per gli indirizzi *link-local* appartenenti allo stesso dominio di collisione (o allo stesso dominio layer2, più precisamente).

Indirizzi IPv6 e subnetting

- E' possibile tuttavia fare regolarmente subnetting, con la “easy way” indicata, come nei precedenti esempi, per esigenze particolari:
- Ad esempio, `2002:FF10:9876:DD0A:9090:4896:AC56:0E01/63`:
 - $63/8 = 7$ con resto di 7; l'ottetto da considerare è l'ottavo (0A), la prima sotto-rete da considerare è quella al settimo posto della sequenza decimale 128,64,32,16,8,4,**2**,0
 - La rete sarà quindi `2002:FF10:9876:DD0A::/63`, ovvero una delle 00, **02**,04,06,08,0A,0C,0E
 - Nel nostro caso, ad esempio, l'indirizzo `2002:FF10:9876:DD0A:9090:4896:AC56:0E01` e l'indirizzo `2002:FF10:9876:DD0B:9090:4896:AC56:0E01` parleranno senza passare per un router, appartenendo alla stessa sottorete identificata dall'ottavo “ottetto” 0A.
 - Come dicevamo, in ogni caso, gli indirizzi che parlano sullo stesso link locale sono tutti identificati ed hanno la stessa mask /64 ed abbiamo “forzato” il concetto di ottetto, che vuole essere ben lontano dalle specifiche di IPv6, che esprime piuttosto la stringa decimale come sequenza alfanumerica.

Da puntualizzare:

- Dimenticate il concetto di classe A (/8), classe B (/16), classe C (/24): dai primi anni '90 (RFC 1517,1518,1519,1520 del 1993) su Internet si utilizza indirizzamento CIDR (Classless Interdomain Routing). Se ciò non fosse avvenuto, Internet non sarebbe in funzione a causa del fatto che:
 - L'address space per le classi B è prossimo all'esaurimento: una classe C è troppo piccola per molte organizzazioni, mentre una classe B è troppo grande e molti indirizzi vengono non utilizzati (vedi ad esempio le università italiane)
 - La routing table di internet è molto grande e cresce continuamente. Servono grossi router per permettere ad un provider di affacciarsi con il proprio Autonomous System ed avere il "full routing" (cosa quasi indispensabile per il *multihoming*)
 - L'indirizzamento per classi, sebbene avesse "self-encoded" la prefix-length, non permette facilmente il supernetting: è improbabile che una organizzazione abbia 255 classi C o B contigue da aggregare in una classe superiore: questo avrebbe portato ad una crescita ancora più veloce delle tabelle di routing globali.

Da puntualizzare:

- La versione “interior”, ovvero interna ad un AS del CIDR viene comunemente chiamata VLSM (Variable Length Subnet Mask) ed è quella che abbiamo visto nel nostro esempio di subnetting a partire da 160.80.1.0/24.
- I prerequisiti per l'applicazione del “variable prefix” sono comunque:
 - Il protocollo di routing deve trasportare l'informazione sul prefisso in ogni *routing advertisement* (cosa non vera nelle prime versioni di BGP o in RIP-1)
 - Tutti i router devono implementare il concetto di *longest match* di una regola di routing (vedremo nella prossima slide)
 - Si deve favorire la “*routing aggregation*” facendo in modo che la topologia fisica e logica della rete possano andare a coincidere, in maniera tale da aggregare più rotte in un'unica rotta di prefisso inferiore.

Da puntualizzare: -*longest match*

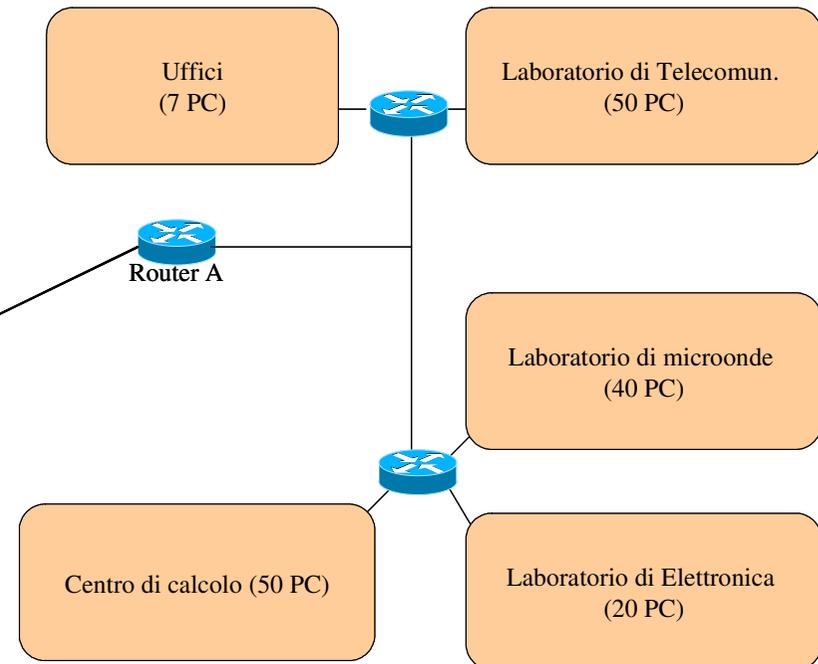
- Se la destination di un pacchetto IP fosse: 11.1.2.5 e ci fossero tre prefissi nella tabella di routing, ad esempio 11.1.2.0/24, 11.1.0.0/16, e 11.0.0.0/8 il router deve selezionare la rotte più specifica (la 11.1.2.0/24), visto il più alto numero di bit corrispondenti nell'indirizzo di destinazione.
- Non sempre i sistemi operativi (soprattutto quelli di apparati *embedded*) rispettano questa regola, che è dipendente dalle implementazioni dello *stack* tcp-ip. Quindi spesso non resta che verificare "a mano" utilizzando tool quali *traceroute* l'effettivo percorso layer3.

```
Destination 11.1.2.5 = 00001011.00000001.00000010.00000101
★ Route #1 11.1.2.0/24 = 00001011.00000001.00000010.00000000
Route #2 11.1.0.0/16 = 00001011.00000001.00000000.00000000
Route #3 11.0.0.0/8 = 00001011.00000000.00000000.00000000
```

Esercizio di esame (anni passati)

Con riferimento alla rete illustrata in figura, nell'assunzione che sia stato acquistato uno spazio di indirizzamento 131.175.21.0/24:

- i) Si assegnino gli indirizzi alle varie sottoreti, nel modo più economico possibile (se avanza uno spazio di indirizzi lo si metta in evidenza);
- ii) Si assegnino indirizzi IP alle interfacce dei router
- iii) si mostri la tabella di instradamento del router A, cercando ove possibile di aggregare al massimo gli instradamenti.



Esercizio di esame (anni passati)

- Ricordiamoci che per ogni subnet ci serve:
 - 1 indirizzo IP che identifica la rete
 - 1 indirizzo IP per il broadcast
 - 1 Indirizzo IP per ogni host
- NB! Anche il router è un host!

- Quindi ci servono ALMENO:
 - Laboratorio di Telecomunicazioni: 53 IP
 - Centro di calcolo : 53 IP
 - Laboratorio di microonde: 43 IP
 - Laboratorio di elettronica 23 IP
 - Uffici 10 IP
- NB Servono anche almeno 5 IP nella subnet di scambio tra i routers.

- TOTALE servono almeno 187 IP
- Possiamo fare subnets da 4 8 16 32 64 128 ... indirizzi IP.

Esercizio di esame (anni passati)

- Quindi:
 - Laboratorio di Telecomunicazioni: 53 IP
 - Usiamo 64 Indirizzi, maschera 255.255.255.192 (/26)
 - Centro di calcolo : 53 IP
 - Usiamo 64 Indirizzi, maschera 255.255.255.192 (/26)
 - Laboratorio di microonde: 43 IP
 - Usiamo 64 Indirizzi, maschera 255.255.255.192 (/26)
 - Laboratorio di elettronica 23 IP
 - Usiamo 32 Indirizzi, maschera 255.255.255.224 (/27)
 - Uffici 10 IP
 - Usiamo 16 Indirizzi, maschera 255.255.255.240 (/28)
 - Tra i routers 5 IP
 - Usiamo 8 indirizzi, maschera 255.255.255.248 (/29)
- Abbiamo risparmiato la
- Subnet: 131.175.21.248/29 con 8 indirizzi IP che possiamo usare in futuro.

Esercizio di esame (anni passati)

- Router A IP: 131.175.21.241
- Router alto IP: 131.175.21.242
- Router basso IP: 131.175.21.243

Destination	Mask	GW
131.175.21.240	/29	0.0.0.0
131.175.21.224	/28	131.175.21.242
131.175.21.192	/27	131.175.21.243
131.175.21.0	/26	131.175.21.242
131.175.21.64	/26	131.175.21.243
131.175.21.128	/26	131.175.21.243
0.0.0.0	0.0.0.0	eth1

Esercizio di esame (anni passati)

Destination	Mask	GW
131.175.21.240	/29	0.0.0.0
131.175.21.224	/28	131.175.21.242
131.175.21.192	/27	131.175.21.243
131.175.21.0	/26	131.175.21.242
131.175.21.64	/25	131.175.21.243
0.0.0.0	0.0.0.0	eth1

Tabella di routing ottimizzata ?