

Si consideri il seguente output del comando `ip route`:

```
$ip route
default via 172.20.0.1 dev eth0 proto kernel
172.20.0.0/20 dev eth0 proto kernel scope link src 172.20.4.226
172.30.0.0/20 dev eth1 proto kernel scope link src 172.30.1.5
172.50.0.0/20 via 172.30.0.1 dev eth1
172.50.8.0/24 via 172.20.0.10 dev eth0
```

Rispondere alle seguenti domande.

1. Che informazioni ci fornisce?
2. Qual è l'output del comando `ip route get` per gli indirizzi IP
 - a. 172.20.5.10
 - b. 172.20.26.10
 - c. 172.50.0.7
3. Nei casi a, b e c indicati sopra, quali sono gli indirizzi di origine e destinazione sia a livello di rete (IP) sia a livello di collegamento (MAC)?

Risposte

1.

Il comando `ip route` mostra la tabella di instradamento del nodo, ossia l'insieme delle rotte che definiscono come instradare il traffico verso differenti prefissi di rete. Quando un pacchetto deve essere inoltrato e il suo indirizzo di destinazione corrisponde a più prefissi, si sceglie la rotta con il prefisso più lungo (cioè la più specifica, che corrisponde alla sottorete con il minor numero di indirizzi). Questa regola è detta “corrispondenza al prefisso più lungo” (*longest prefix match*).

Le righe 2 e 3, etichettate con `scope link` (e prive dell'attributo `via`), forniscono rotte per reti direttamente connesse, raggiungibili cioè a livello di collegamento tramite l'interfaccia indicata (`dev`) senza passare per un router/gateway. L'attributo `src` indica l'indirizzo IP da usare come sorgente per i pacchetti la cui destinazione ricade in quella rete: di solito coincide con l'IP assegnato all'interfaccia stessa.

Queste rotte vengono gestite automaticamente dal kernel in seguito all'aggiunta o alla rimozione di interfacce e all'assegnazione o rimozione di indirizzi IP (cfr. `ip link`, `ip addr`).

Nello specifico:

- la rete 172.20.0.0/20 è direttamente connessa su `eth0` e si deve usare come indirizzo IP sorgente 172.20.4.226
- la rete 172.30.0.0/20 è direttamente connessa su `eth1` e si deve usare come indirizzo IP sorgente 172.30.1.5

Le righe 4 e 5 definiscono rotte per reti non direttamente connesse: per raggiungerle, i pacchetti devono essere inoltrati al router specificato con `via`, che a sua volta è raggiungibile tramite la sottorete connessa all'interfaccia indicata con `dev`. Il kernel verifica automaticamente che il gateway sia effettivamente raggiungibile su quell'interfaccia.

Nello specifico:

- la rete `172.50.0.0/20` è raggiungibile passando per il router `172.30.0.1`, che si trova nella sottorete direttamente connessa all'interfaccia `eth1`
- la rete `172.50.8.0/24` è raggiungibile passando per il router `172.20.0.1`, che si trova nella sottorete direttamente connessa all'interfaccia `eth0`

Si noti che il prefisso `172.50.8.0/24` è un sottoblocco del prefisso `172.50.0.0/20`; in fase di inoltro, la rotta a `/24` prevale su quella a `/20` grazie alla regola del *longest prefix match*.

Infine, la riga 1 corrisponde alla rotta di default (`0.0.0.0/0`): tutto il traffico verso destinazioni che non rientrano in rotte più specifiche viene inviato al gateway `172.20.0.1` tramite l'interfaccia `eth0`.

2.

Il comando `ip route get` calcola la rotta che verrà utilizzata per inoltrare un pacchetto destinato all'indirizzo IP fornito come argomento. Nell'output comparirà l'indirizzo IP di destinazione specificato, anziché un prefisso della tabella di instradamento. Differisce quindi dal comando `ip route show`, che elenca tutte le rotte esistenti (eventualmente filtrabili mediante criteri di selezione).

Il processo di inoltro sfrutta una struttura dati ottimizzata, chiamata Forwarding Information Base (FIB), costruita a partire da un sottoinsieme delle informazioni di instradamento presenti nella Routing Information Base (RIB). La FIB contiene solo le rotte effettivamente utilizzate per l'inoltro, ad esempio quelle a costo minimo.

La rotta selezionata per raggiungere un dato indirizzo IP di destinazione è quella associata al prefisso più lungo, nel formato `a.b.c.d/n`, tale che i primi `n` bit dell'indirizzo IP coincidano con quelli del prefisso.

Un altro modo per esprimere il confronto è dire che un certo indirizzo IP `x` corrisponde a (ovvero rientra in) un prefisso `y/n` se:

$$x \& m = y$$

dove `&` rappresenta l'AND bit a bit e `m` (la *netmask*) è un numero a 32 bit in cui i primi `n` bit sono 1 e i rimanenti `32-n` bit sono 0 (nel caso di IPv6 occorre considerare invece numeri a 128 bit).

La corrispondenza dei bit più significativi di un indirizzo IP con quelli di un prefisso significa che l'indirizzo appartiene al blocco di indirizzi che hanno quel determinato prefisso. È possibile verificare questa corrispondenza controllando che l'indirizzo cada all'interno dell'intervallo associato al prefisso:

- il valore minimo del blocco ha la parte host a 0 (ossia coincide con l'indirizzo di rete);
- il valore massimo ha la parte host tutta a 1 (ossia coincide con l'indirizzo di broadcast).

Per esempio, consideriamo il prefisso 172.20.0.0/20. In binario si rappresenta così (32 bit totali, con i primi 20 che identificano la rete e gli ultimi 12 che identificano la parte host):

172.20.0.0/20:

10101100 . 00010100 . 0000|0000 . 00000000

portando tutti i bit relativi alla parte host a 1, si ottiene:

10101100 . 00010100 . 0000|1111 . 11111111

che corrisponde a 172.20.15.255.

In questo modo si verifica che qualsiasi indirizzo da 172.20.0.0 fino a 172.20.15.255 appartenga al blocco 172.20.0.0/20.

Ragionando in modo analogo, si ottengono gli intervalli corrispondenti ai diversi blocchi di indirizzi.

prefisso	intervallo di indirizzi
default (0/0)	0.0.0.0 - 255.255.255.255
172.20.0.0/20	172.20.0.0 - 172.20.15.255
172.30.0.0/20	172.30.0.0 - 172.30.15.255
172.50.0.0/20	172.50.0.0 - 172.50.15.255
172.50.8.0/24	172.50.8.0 - 172.50.8.255

nota: nei tre casi mostrerò i diversi modi di ragionare

a. 172.20.5.10

Come detto all'inizio, occorre confrontare l'indirizzo IP 172.20.5.10 con i prefissi contenuti nella tabella di instradamento (limitando il confronto ai primi n bit a partire da quelli più significativi, dove n è la lunghezza del prefisso) e scegliendo infine il prefisso più lungo tra quelli che corrispondenti.

- 0.0.0.0/0 → corrispondenza: i primi 0 bit dell'indirizzo IP coincidono con quelli del prefisso
- 172.20.0.0/20 → corrispondenza: i primi 20 bit dell'indirizzo IP coincidono con quelli del prefisso; infatti:
 - i primi due byte (172.20) sono uguali;

- nel terzo byte si considerano solo i primi 4 bit:
 byte dell'indirizzo: $(5)_{10} = (00000101)_2$
 byte del prefisso: $(0)_{10} = (00000000)_2$
 sono stati sottolineati i primi quattro bit (dal più significativo) nel byte dell'indirizzo e in quello del prefissi e chiaramente corrispondono, essendo 0000 .
- $172.30.0.0/20 \rightarrow$ non corrispondenza: i primi 20 bit dell'indirizzo differiscono da quelli del prefisso nel secondo byte ($20 \neq 30$)
- $172.50.0.0/20 \rightarrow$ non corrispondenza: i primi 20 bit dell'indirizzo differiscono da quelli del prefisso nel secondo byte ($20 \neq 50$)
- $172.50.8.0/24 \rightarrow$ non corrispondenza: i primi 20 bit dell'indirizzo differiscono da quelli del prefisso nel secondo byte ($20 \neq 50$).

Sceglierò quindi il prefisso nella seconda corrispondenza perché è il più lungo.

```
$ ip route get 172.20.5.10
172.20.5.10 dev eth0 src 172.20.4.226 uid 1000
cache
```

Questo output è stato ottenuto a partire dalla rotta numero 2

Nota:

- uid 1000 rappresenta l'ID dell'utente che ha richiesto la rotta (può essere diverso)
- cache indica che il risultato è memorizzato nella cache del kernel per risposte più rapide in futuro

b. $172.20.26.10$

Utilizzo la netmask per verificare se un indirizzo IP rientra in un determinato prefisso. La prima colonna indica se c'è una corrispondenza.

	prefisso	
✓	$0.0.0.0/0$	netmask = $0.0.0.0$ $172.20.26.10 \& 0.0.0.0 = 0.0.0.0$
	$172.20.0.0/20$	netmask = $255.255.240.0$ $172.20.26.10 \& 255.255.240.0 = 172.20.16.0 \neq 172.20.0.0$ dato che $(255)_{10} = (1111\ 1111)_2$ e $(26)_{10} \& (240)_{10} = (0001\ 1010)_2 \& (1111\ 0000)_2 = (0001\ 0000)_2 = (16)_{10}$
	$172.30.0.0/20$	$172.20.26.10 \& 255.255.240.0 = 172.20.16.0 \neq 172.30.0.0$

	172.50.0.0/20	172.20.26.10 & 255.255.240.0 = 172.20.16.0 ≠ 172.50.0.0
	172.50.8.0/24	netmask = 255.255.255.0 172.20.26.10 & 255.255.255.0 = 172.20.26.0 ≠ 172.50.8.0

```
$ ip route get 172.20.26.10
```

```
172.20.26.10 via 172.20.0.1 dev eth0 src 172.20.4.226 uid 1000
cache
```

Questo output è stato calcolato a partire dalla rotta di default (riga 1), in quanto è l'unica che include l'indirizzo specificato. L'attributo `src` è stato scelto avendo dedotto che il router appartiene alla rete 172.20.0.0/20 direttamente connessa all'interfaccia di uscita `eth0`, alla quale è stato assegnato l'indirizzo 172.20.4.226 (riga 2).

c. 172.50.0.7

Considero gli intervalli calcolati in precedenza per ciascun prefisso e verifico a quale di essi appartenga l'indirizzo 172.50.0.7.

L'appartenenza a un intervallo richiede che l'indirizzo sia maggiore o uguale all'estremo inferiore e minore o uguale all'estremo superiore; anziché convertire gli indirizzi IP nei numeri a 32 bit corrispondenti, è possibile confrontarli in maniera lessicografica a partire dal byte più significativo (https://it.wikipedia.org/wiki/Ordine_lessicografico).

La prima colonna indica se c'è una corrispondenza.

	prefisso	intervallo di indirizzi	
✓	default (0/0)	0.0.0.0 – 255.255.255.255	L'indirizzo 172.50.0.7 ricade ovviamente nell'intervallo che include tutti gli indirizzi
	172.20.0.0/20	172.20.0.0 – 172.20.15.255	L'indirizzo 172.50.0.7 è fuori dall'intervallo perché è maggiore dell'estremo superiore, ovvero: 172.50.0.7 > 172.20.15.255, in quanto 172 = 172, ma 50 > 20
	172.30.0.0/20	172.30.0.0 – 172.30.15.255	L'indirizzo 172.50.0.7 è fuori dall'intervallo perché è maggiore dell'estremo superiore, ovvero: 172.50.0.7 > 172.30.15.255, in quanto 172 = 172, ma 30 > 20
✓	172.50.0.0/20	172.50.0.0 – 172.50.15.255	L'indirizzo 172.50.0.7 ricade nell'intervallo perché: 1. è maggiore o uguale all'estremo inferiore, ovvero: 172.50.0.7 ≥ 172.50.0.0 in

			<p>quanto $172 = 172,50 = 50,0 = 0$, e $7 \geq 0$</p> <p>2. è minore o uguale all'estremo superiore, ovvero: $172.50.0.7 \leq 172.50.15.255$ in quanto $172 = 172,50 = 50,0 < 15$</p>
	172.50.8.0/24	172.50.8.0 - 172.50.8.255	<p>L'indirizzo $172.50.0.7$ è fuori dall'intervallo perché è minore dell'estremo inferiore, ovvero: $172.50.0.7 < 172.50.8.0$, in quanto $172 = 172,50 = 50$, ma $0 < 8$</p>

Si considera quindi la rotta per il prefisso più lungo per cui c'è una corrispondenza:
 $172.50.0.0/20$

```
$ ip route get 172.50.0.7
172.50.0.7 via 172.30.0.1 dev eth1 src 172.30.1.5 uid 1000
Cache
```

Questo output è stato calcolato a partire dalla rotta più specifica che include l'indirizzo fornito (riga 4), utilizzando l'indirizzo sorgente configurato sulla rete direttamente connessa all'interfaccia eth 1 (riga 3), nella quale si trova il router next-hop.

3.

Quando un host deve inviare un datagramma, incapsulandolo in un frame, utilizza i seguenti indirizzi:

- Livello di rete:
 - sorgente: indirizzo IP scelto come sorgente in base all'attributo `src` della rotta restituita da `ip route get`. Solitamente, l'indirizzo IP assegnato all'interfaccia di uscita
 - destinazione: indirizzo IP del destinatario finale del pacchetto, indipendentemente dal fatto che il pacchetto debba attraversare router intermedi
- Livello di collegamento:
 - sorgente: indirizzo MAC assegnato all'interfaccia di uscita
 - destinazione: indirizzo MAC assegnato all'interfaccia del nodo adiacente che riceve il frame: appartiene al destinatario finale (se situato nella stessa sottorete dell'host sorgente) oppure al primo router lungo il percorso verso destinazioni al di fuori della sottorete.

L'indirizzo MAC sorgente può essere ricavato dall'output del comando `ip link`.

L'indirizzo MAC di destinazione può essere ottenuto risolvendo l'indirizzo IP dell'host di destinazione o del successivo router (a seconda del tipo di instradamento): innanzitutto si controlla la tabella ARP associata all'interfaccia di uscita e, se non si trova alcuna corrispondenza, si utilizza il protocollo ARP per cercare il MAC richiesto.

Vale la pena notare che eventuali router intermedi non modificano gli indirizzi IP sorgente e destinazione, ma cambiano gli indirizzi MAC a livello di collegamento: il MAC sorgente diventa quello dell'interfaccia di uscita del router, mentre il MAC destinazione è quello dell'interfaccia del nodo adiacente successivo (un altro router oppure l'host di destinazione).

a. 172.20.5.10

Instradamento diretto:

- Livello di collegamento:
 - sorgente: indirizzo MAC assegnato a eth0 (vedi ip link)
 - destinazione: indirizzo MAC dell'interfaccia cui è assegnato l'indirizzo IP 172.20.5.10: cercato nella tabella ARP di eth0 e se assente determinato risolvendo l'indirizzo IP col protocollo ARP
- Livello di rete:
 - sorgente: 172.20.4.226
 - destinazione: 172.20.5.10

b. 172.20.26.10

Instradamento indiretto attraverso 172.20.0.1 su eth0:

- Livello di collegamento:
 - sorgente: indirizzo MAC assegnato a eth0 (vedi ip link)
 - destinazione: indirizzo MAC dell'interfaccia cui è assegnato l'indirizzo IP 172.20.0.1 (il router next-hop): cercato nella tabella ARP di eth0 e se assente determinato risolvendo l'indirizzo IP col protocollo ARP
- Livello di rete:
 - sorgente: 172.20.4.226
 - destinazione: 172.20.5.10

c. 172.50.0.7

Instradamento indiretto attraverso 172.30.0.1 su eth1:

- Livello di collegamento:
 - sorgente: indirizzo MAC assegnato a eth1 (vedi ip link)
 - destinazione: indirizzo MAC dell'interfaccia cui è assegnato l'indirizzo IP 172.30.0.1 (il router): cercato nella tabella ARP di eth1 e se assente determinato risolvendo l'indirizzo IP col protocollo ARP
- Livello di rete:
 - sorgente: 172.30.1.5

- destinazione: 172.50.0.7