

Python: calcolare il numero di indirizzi IP disponibili in una subnet

Calcolare il numero di indirizzi IP disponibili in una subnet è una competenza fondamentale per chi si occupa di networking. In questo articolo, illustreremo come utilizzare Python per eseguire questi calcoli, sfruttando le potenzialità del linguaggio e di alcune librerie utili.

Prima di procedere con il codice Python, è importante capire alcuni concetti di base sulla subnetting e sugli indirizzi IP. Un indirizzo IP (Internet Protocol) è un identificativo unico assegnato a ciascun dispositivo su una rete. Gli indirizzi IP possono essere di due tipi: IPv4 e IPv6. Qui ci concentreremo sugli indirizzi IPv4, che sono composti da 32 bit e di solito vengono espressi in notazione decimale puntata (esempio: 192.168.1.1).

Una subnet è una divisione di una rete IP in blocchi più piccoli, il che permette una gestione più efficiente del traffico e una maggiore sicurezza. La capacità di una subnet è determinata dalla sua maschera di rete, che specifica quanti bit dell'indirizzo IP sono riservati per l'identificazione della rete e quanti per gli host all'interno di quella rete.

La formula per calcolare il numero di indirizzi disponibili in una subnet è:

Numero di indirizzi = $2^{(32 - \text{numero di bit della maschera di rete})}$

Per una subnet con una maschera di rete di 24 bit (comunemente espressa come /24 o 255.255.255.0), il calcolo sarebbe:

$$2^{(32-24)} = 256$$

Di questi 256 indirizzi, uno è riservato per l'indirizzo di rete e uno per il broadcast, lasciando quindi 254 indirizzi utilizzabili per gli host.

Python offre diverse librerie che possono facilitare il calcolo degli indirizzi IP disponibili. Una delle più comuni è `ipaddress`. Ecco un esempio di codice che utilizza questa libreria per calcolare il numero di indirizzi IP disponibili in una subnet:

```
import ipaddress

def calc_available_addresses(subnet):
    # Creazione di un oggetto di rete IP utilizzando
    # la libreria ipaddress
    net_obj = ipaddress.ip_network(subnet)

    # Calcolo del numero totale di indirizzi nella
    # subnet
    total_addresses = net_obj.num_addresses

    # Sottrazione degli indirizzi di rete e broadcast
    available_addresses = total_addresses - 2

    return available_addresses

# Esempio di utilizzo
subnet = "192.168.1.0/24"
available_addresses =
calc_available_addresses(subnet)
print(f"Indirizzi IP utilizzabili:
{available_addresses}")
```

In conclusione, calcolare il numero di indirizzi IP in una subnet è essenziale per la pianificazione e la gestione delle reti. Utilizzando Python e le sue librerie, si può semplificare notevolmente questo compito, rendendo i calcoli rapidi e precisi. Con la conoscenza adeguata e gli strumenti giusti, gestire le subnet diventa un'attività molto più gestibile.