

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

Calcolare il numero di indirizzi IP disponibili in una subnet è un compito fondamentale nella gestione di reti. In questo articolo, vedremo come implementare questa funzionalità in Java, sfruttando la conoscenza della maschera di sottorete (subnet mask).

Prima di immergerci nel codice, è essenziale capire alcuni concetti di base delle reti:

- **Indirizzo IP:** Un indirizzo univoco assegnato a ogni dispositivo su una rete.
- **Subnet Mask:** Usata per dividere l'indirizzo IP in una 'rete' e una 'parte host'.
- **CIDR (Classless Inter-Domain Routing):** Una notazione usata per specificare la subnet mask, es. `192.168.1.0/24` dove `/24` indica che i primi 24 bit dell'indirizzo sono la parte di 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.

Ecco un esempio di come potremmo implementare il calcolo degli indirizzi disponibili in una subnet usando Java:

```
public class SubnetCalculator {

    public static int calculateAvailableIPs(String
subnetMask) {
        String[] parts = subnetMask.split("\\.");
        int onesCount = 0;
        for (String part : parts) {
            int value = Integer.parseInt(part);
            onesCount += Integer.bitCount(value);
        }
        int hostBits = 32 - onesCount;
        return (int) Math.pow(2, hostBits) - 2;
    }

    public static void main(String[] args) {
        String subnetMask = "255.255.255.0"; //
Esempio: maschera di sottorete /24
        int availableIPs =
calculateAvailableIPs(subnetMask);
        System.out.println("Numero di indirizzi IP
disponibili: " + availableIPs);
    }
}
```

Ossia:

1. **Metodo calculateAvailableIPs:** Il metodo accetta una stringa che rappresenta la subnet mask e calcola il numero di bit impostati a 1.

Divide la subnet mask in quattro parti (i byte), converte ciascuna parte in un intero, e usa `Integer.bitCount` per contare quanti bit sono impostati a 1 in ogni byte.

2. **Calcolo degli indirizzi disponibili:** Sottraendo il numero totale di bit impostati a 1 da 32, otteniamo il numero di bit disponibili per gli indirizzi host. Utilizziamo `Math.pow` per calcolare $2^{(32-n)}$, e sottraiamo 2 per escludere l'indirizzo di rete e quello di broadcast.
3. **Main:** In questo metodo, definiamo un esempio di subnet mask e invochiamo il metodo `calculateAvailableIPs` per ottenere e stampare il numero di indirizzi IP disponibili.

In conclusione, questo esempio dimostra come si possa facilmente calcolare il numero di indirizzi IP disponibili in una subnet utilizzando Java. La comprensione e l'implementazione di questi concetti sono essenziali per la gestione efficace delle reti IP.