

# Go: calcolare il numero di indirizzi IP disponibili in una subnet

Calcolare il numero di indirizzi IP disponibili in una subnet è un'operazione cruciale nella gestione delle reti, specialmente quando si prevede la dimensione di una rete o si configura il networking per applicazioni e servizi. Usando il linguaggio di programmazione Go, possiamo sviluppare un'utilità che ci permette di eseguire questo calcolo in modo efficiente. In questo articolo, vedremo come implementare una funzione in Go per calcolare il numero di indirizzi IP disponibili in una data subnet.

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 come potresti implementare una funzione in Go che calcola il numero di indirizzi IP utilizzabili in una subnet:

```
package main

import (
    "fmt"
    "math"
    "net"
    "os"
)

// calculateIPs calcola il numero di indirizzi IP
// utilizzabili in una subnet CIDR.
func calculateIPs(cidr string) (int, error) {
    _, ipv4Net, err := net.ParseCIDR(cidr)
    if err != nil {
        return 0, err
    }
    maskSize, _ := ipv4Net.Mask.Size()

    totalIPs := int(math.Pow(2, float64(32-
maskSize))) - 2
    if totalIPs < 0 {
        totalIPs = 0 // Gestisce il caso delle subnet
con maschera /31 e /32
    }
    return totalIPs, nil
}
```

```
func main() {
    if len(os.Args) < 2 {
        fmt.Println("Uso: go run main.go <CIDR>")
        os.Exit(1)
    }
    cidr := os.Args[1]

    count, err := calculateIPs(cidr)
    if err != nil {
        fmt.Println("Errore durante il calcolo degli
indirizzi IP:", err)
        os.Exit(1)
    }
    fmt.Printf("Numero di indirizzi IP utilizzabili
per %s: %d\n", cidr, count)
}
```

In conclusione, questo strumento semplice ma efficace in Go ti permette di calcolare rapidamente il numero di indirizzi IP utilizzabili in qualsiasi subnet CIDR. È utile per gli amministratori di sistema e per chiunque lavori con la configurazione di reti. Con la comprensione di questo strumento, puoi facilmente espandere o modificare il codice per includere ulteriori funzionalità di networking.