

Progetto di Programmazione di Reti

Lisa Vandi, matricola 0001071265, lisa.vandi2@studio.unibo.it

Ingegneria e Scienze Informatiche, a.a. 2023-2024

Documentazione della traccia 3: Monitoraggio di Rete

Introduzione

La documentazione seguente fornisce una descrizione dettagliata dello script Python “**monitoring.py**”, insieme alle istruzioni per l'installazione e l'utilizzo.

L'elaborato svolto consiste in uno strumento di monitoraggio della rete scritto in Python. Utilizza il modulo “pythonping” per inviare richieste ICMP (ping) agli host specificati dall'utente, verificandone la raggiungibilità. Inoltre, include anche funzionalità per la validazione degli indirizzi IP e per la gestione delle eccezioni.

Descrizione del protocollo ICMP

Il protocollo ICMP (Internet Control Message Protocol) è un protocollo di servizio per reti a pacchetto che si occupa di trasmettere informazioni riguardanti malfunzionamenti, controllo o messaggi tra i vari componenti di una rete di calcolatori. Esso è utilizzato da molti applicativi di rete, tra cui Ping, che verifica la raggiungibilità di un qualsiasi host.

In particolare, Ping è un'utility di rete utilizzata per testare la raggiungibilità di un host su una rete IP e per misurare il tempo necessario per i pacchetti di andata e ritorno verso l'host di destinazione. Utilizza il protocollo ICMP per inviare pacchetti di prova di tipo echo-request all'host di destinazione e ricevere pacchetti di tipo echo-reply. Il tempo impiegato per ricevere la risposta è misurato, fornendo un'indicazione del round-trip time (RTT). Attraverso il Ping, si ottengono le seguenti informazioni:

- Stato di raggiungibilità: se l'host risponde, allora è considerato raggiungibile
- Tempo di risposta
- Perdita di pacchetti e tasso di errore: se un determinato numero di pacchetti echo-request non riceve risposta, allora si considera che ci sia una perdita di pacchetti.

Requisiti

Al fine di eseguire correttamente il codice fornito, innanzitutto è necessario installare il pacchetto ‘pythonping’. Tale installazione è possibile mediante il comando ‘pip’:

```
pip install pythonping
```

La libreria ‘pythonping’ fornisce un'interfaccia semplice per eseguire ping in Python e analizzare le risposte.

Struttura e funzionamento del codice

- Funzione **ping_host(host)**: essa prende in input un parametro host, che è l'indirizzo IP dell'host da monitorare. Quindi, esegue il comando ping sull'host e restituisce True se l'host è raggiungibile, False altrimenti.
La funzione ping() di pythonping restituisce un oggetto di tipo ResponseList, cioè una lista di oggetti Response, ciascuno contenente il pacchetto ricevuto e alcune meta-informazioni, come il tempo impiegato a ricevere la risposta e messaggi di errore. In più, la libreria pythonping gestisce i timeout internamente e permette di verificare se una risposta è stata ricevuta utilizzando la proprietà success delle risposte Response. Questo approccio consente di gestire i timeout senza dover definire eccezioni personalizzate.

Oltre a ciò, la funzione `ping_host(host)` gestisce anche eventuali eccezioni per mezzo dell'utilizzo del costrutto `try/except`.

- Funzione **`valid_ip(ip)`**: verifica se un indirizzo IP è valido. Essa prende in input una stringa contenente un indirizzo IP. Se la stringa fornita come parametro è un indirizzo IP valido, la funzione `ip_address` della libreria `ipaddress` restituirà un oggetto `IPv4Address` o `IPv6Address` e la funzione `valid_ip` restituirà `True`.
Se la stringa fornita non è un indirizzo IP valido, la funzione `ip_address` solleverà un'eccezione `ValueError`. Questa eccezione viene catturata nel blocco `except` e viene stampato un messaggio di errore. Dunque, la funzione `valid_ip` restituirà `False`.
- Funzione **`main()`**: è la funzione principale dello script "monitoring.py".
È presente un blocco `try/except` per gestire eventuali errori di inserimento dati da parte dell'utente, dal momento che, all'interno di tale blocco, gli viene richiesto di inserire il numero di host che desidera monitorare (minimo 1). Se tale inserimento risulta corretto, allora l'utente potrà proseguire con l'inserimento da tastiera degli indirizzi IP degli host di cui controllare lo stato, dopo aver verificato la correttezza di tali indirizzi attraverso la funzione `valid_ip(ip)`. Successivamente, viene utilizzata la funzione `ping_host(host)` per determinare se ogni host è online o offline.

Esecuzione dello script

Dopo aver aperto un terminale ed essersi spostati nella directory in cui è salvato lo script, si lancia lo script con il seguente comando:

```
python monitoring.py
```

Successivamente, si seguono le indicazioni fornite dallo script che appaiono sul terminale:

```
Inserire il numero di host da monitorare: █
```

```
Inserire l'indirizzo IP dell'host 1: █
```

Gestione delle eccezioni

La funzione "`ping_host(host)`" utilizza un blocco `try/except` per gestire le eccezioni che possono verificarsi durante l'esecuzione del ping verso un host: ogni eccezione viene catturata dal blocco generico "`except Exception as e`" che stampa il messaggio di errore e restituisce `False`. Per quel che riguarda le eccezioni relative alla scadenza di un Timeout nelle richieste ICMP, esse sono gestite internamente dalla libreria `Pyhtonping`, come specificato sopra.

La funzione "`valid_ip(ip)`" si comporta in maniera analoga: se l'indirizzo IP fornito non è valido, viene sollevata l'eccezione "`ValueError`". Quindi, viene stampato un messaggio di errore con il dettaglio dell'eccezione e la funzione restituisce `False`.

Infine, la funzione "`main()`" gestisce le eccezioni relative all'input dell'utente quando si specifica il numero di host da monitorare; infatti, tale numero deve essere almeno pari a 1.