

Relazione  
Monitor di stato degli host tramite ping  
Progetto di laboratorio programmazione di reti

Pulga Luca - matricola 0001091461

08 luglio 2024

# Indice

<b>1</b>	<b>Traccia: Monitoraggio di Rete</b>	<b>2</b>
1.1	Punteggio . . . . .	2
1.2	Requisiti di sistema . . . . .	2
<b>2</b>	<b>Funzionamento dello script</b>	<b>3</b>
<b>3</b>	<b>Come utilizzare lo script</b>	<b>4</b>
3.0.1	Script . . . . .	4
3.0.2	Aggiunta di host . . . . .	4
3.0.3	Visualizzare lo stato degli host aggiunti . . . . .	4
3.0.4	Uscita dal Programma . . . . .	4
<b>4</b>	<b>Gestione degli errori</b>	<b>5</b>
4.0.1	subprocess.CalledProcessError: . . . . .	5
4.0.2	Exception (generica): . . . . .	5
4.0.3	Errore di input da parte dell'utente: . . . . .	5
<b>5</b>	<b>Considerazioni aggiuntive</b>	<b>8</b>

# Capitolo 1

## Traccia: Monitoraggio di Rete

Realizzare uno script Python per monitorare lo stato di una rete, controllando la disponibilità di uno o più host tramite il protocollo ICMP (ping). Lo script deve consentire all'utente di specificare gli indirizzi IP degli host da monitorare e deve visualizzare lo stato (online/offline) di ciascun host.

### 1.1 Punteggio

Regole per l'assegnazione dei punti:

- 1 punto: Implementazione funzionante dello script che può inviare pacchetti ICMP per controllare la disponibilità degli host specificati.
- 1 punto: Capacità dello script di gestire più host e visualizzare correttamente il loro stato.
- 1 punto: Gestione adeguata degli errori e delle eccezioni (ad esempio, timeout delle richieste ICMP).
- 1 punto: Documentazione chiara e ben strutturata in un PDF che spieghi il funzionamento dello script, come utilizzarlo e considerazioni aggiuntive.

### 1.2 Requisiti di sistema

Sistema operativi compatibili: Windows, Linux, MacOS  
Python 3.x installato

## Capitolo 2

# Funzionamento dello script

Lo script realizzato fornisce una semplice e chiara interfaccia grafica per monitorare lo stato di connessione di più indirizzi IP, in tempo reale, utilizzando il comando **ping** e presentando i risultati in modo più intuitivo possibile all'utente. Quando lo script viene eseguito, si aprirà la finestra principale di Tkinter e l'utente potrà inserire gli indirizzi IP da monitorare. Premendo il pulsante per avviare il monitoraggio, verranno creati i thread per monitorare ciascun indirizzo IP, aggiornando continuamente lo stato degli host nella finestra. Di seguito sono riportate ulteriori informazioni relativamente allo script:

- **Interfaccia Utente:** l'interfaccia utente ha tre pannelli principali: uno per inserire gli indirizzi IP (in alto), uno per visualizzare i log relativi allo stato dei singoli IP inseriti e un altro pannello per visualizzare lo stato degli host in una Listbox (a destra). Gli indirizzi IP possono essere inseriti in un campo di testo separato da virgole e il monitoraggio inizia premendo un pulsante "**Start Monitoring hosts**";
- **Monitoraggio degli host:** gli indirizzi IP inseriti sono validati e monitorati utilizzando thread separati per evitare blocchi nell'interfaccia utente. Ogni thread esegue un ping continuo sull'host assegnato e aggiorna l'interfaccia con lo stato (online/offline);
- **Ping degli host:** il metodo `ping_host` utilizza il comando *ping* appropriato in base al sistema operativo per determinare se l'host è raggiungibile o meno. I risultati del *ping* sono visualizzati sia in una TextArea sia in una ListBox con colori diversi per gli stati (verde per online, rosso per offline);
- **Gestione degli errori:** viene effettuata la validazione degli indirizzi IP inseriti. Se non validi, viene mostrato un messaggio di errore.

# Capitolo 3

## Come utilizzare lo script

### 3.0.1 Script

Assicurarsi che il file dello script **network\_monitoring.py** sia presente nel sistema. E' possibile eseguire lo script nei seguenti modi:

- Doppio click sullo script;
- Aprire un terminale nella directory contenente lo script ed eseguire lo script Python con il comando **python network\_monitoring.py**.

### 3.0.2 Aggiunta di host

Inserire l'hostname o l'indirizzo IP dell'host che si desidera monitorare. E' possibile inserire più indirizzi IP separati dalla *virgola*. Una volta inseriti, è possibile cliccare il pulsante "**Start Monitoring hosts**".

### 3.0.3 Visualizzare lo stato degli host aggiunti

E' possibile visualizzare lo stato degli host sia dai log nel pannello centrale, sia nel pannello laterale destra, per una visualizzazione più intuitiva.

### 3.0.4 Uscita dal Programma

Per uscire dal programma, sarà necessario solamente chiudere l'interfaccia.

# Capitolo 4

## Gestione degli errori

### 4.0.1 subprocess.CalledProcessError:

**Condizione:** Errore durante l'esecuzione del comando ping (host offline).

**Azione:** Restituisce 'offline'.

---

```
except CalledProcessError:  
    return 'offline'
```

---

### 4.0.2 Exception (generica):

**Condizione:** Qualsiasi altra eccezione durante l'esecuzione del comando ping.

**Azione:** Stampa il messaggio di errore e restituisce 'error'.

---

```
except Exception as e:  
    print(f"Error pinging {host}: {e}")  
    return 'error'
```

---

### 4.0.3 Errore di input da parte dell'utente:

**Indirizzi IP mancanti:**

**Condizione:** Nessun indirizzo IP inserito.

**Azione:** Mostra un messagebox con il messaggio "Enter valid IP addresses."

---

```
if not hosts:  
    messagebox.showerror("Error", "Enter valid IP addresses.")
```

---

```
return
```

---

### Tutti gli indirizzi IP non validi:

**Condizione:** Nessun indirizzo IP è valido.

**Azione:** Mostra un messagebox con il messaggio "Enter valid IP addresses."

---

```
if not valid_hosts:
    messagebox.showerror("Error", "Enter valid IP addresses.")
    return
```

---

### Indirizzi IP parzialmente validi/non validi:

**Condizione:** Almeno un indirizzo IP non valido. **Azione:** Mostra un messagebox con il messaggio "Invalid IP addresses: ..." seguito dall'elenco degli IP non validi.

---

```
if invalid_hosts:
    messagebox.showerror("Error", f"Invalid IP addresses: {'',
        '.join(invalid_hosts)}")
    return
```

---

### Errore di Formato IP

**Descrizione:** Un indirizzo IP non rispetta il formato numerico xxx.xxx.xxx.xxx (dove xxx è un numero tra 0 e 255).

**Gestione:** La funzione `validate_ip` ritorna `False`. Viene mostrato un messagebox di errore con l'elenco degli IP non validi.

---

```
if not ip_pattern.match(ip):
    return False
```

---

### Errore di Valori IP

**Descrizione:** Un indirizzo IP contiene valori numerici fuori dall'intervallo 0-255.

**Gestione:** La funzione `validate_ip` ritorna `False`. Viene mostrato un messagebox di errore con l'elenco degli IP non validi.

---

```
parts = ip.split('.')  
if not all(0 <= int(part) <= 255 for part in parts):  
    return False
```

---



# Capitolo 5

## Considerazioni aggiuntive

Lo script utilizza il comando **ping** del **sistema operativo**, dunque è necessario assicurarsi che questo funzioni correttamente e abbia i permessi necessari.