



ELEMENTI DI RETI DI TELECOMUNICAZIONE. AA 21-22

PROGETTO “INTERNET VISTO DA BRESCIA”

PICCOLA SPERIMENTAZIONE PER MISURARE, TRAMITE IL COMANDO PING E ALCUNI SCRIPT PYTHON, LA DISTANZA IN TERMINI DI “ROUND TRIP TIME” (RTT) DI ALCUNI HOST RAPPRESENTATIVI IN ITALIA E NEL MONDO, COME VISTA DAGLI APPARATI DEGLI STUDENTI DEL CORSO DI “RETI DI TELECOMUNICAZIONE” ATTRAVERSO I LORO ISP (O LA RETE DELL'UNIVERSITÀ)

21 ottobre 2021

Istruzioni per eseguire correttamente le misurazioni di RTT, estenderle per ottenere un punteggio più alto, riportare correttamente i risultati su Moodle e installare il software necessario se non è ancora presente sul PC.

Indice

1 Istruzioni generali	1
1.1 Lista di indirizzi da “pingare”	2
1.2 Uso di autoping.py	4
1.2.1 Configurazione	4
1.2.2 Esecuzione	5
1.2.3 Elaborazione/Salvataggio risultati	6
1.3 Uso degli argomenti con autoping.py	6
1.4 Uso di rtt_plotter.py	7
2 Esecuzione degli esperimenti e consegna dei dati	8
2.1 Set minimo di misure, estensione e “punti guadagnati”	8
2.2 Pubblicazione dei dati	9
3 Guida all'installazione	10
3.1 Note utili per sistemi Linux	10
3.2 Note utili per sistemi Windows	10

1 Istruzioni generali

L'archivio tools.zip contiene i seguenti script, file di configurazione e dati utili:

- | | |
|----------------|---------------------|
| 1. IPlist.txt | 3. rtt_plotter.py |
| 2. autoping.py | 4. requirements.txt |

1.1 Lista di indirizzi da “pingare”

IPlist.txt è un file di testo che elenca, uno per ogni linea, il set di indirizzi IP che sono stati selezionati dai docenti per gli esperimenti. Questi IP sono stati selezionati nel tentativo di individuare host uniformemente sparsi nel mondo, ma con una densità maggiore in Italia e in Europa. Per individuarli si è partiti da alcuni Partially Qualified Domain Names (PQDNs)¹ noti, con presunta chiara collocazione geografica, quindi il loro IP è stato risolto e geolocalizzato tramite web-services pubblici quali ip-api.com. Per esempio, si ritiene che il sito web della nota compagnia nipponica TOSHIBA, raggiungibile via www.toshiba.co.jp, corrisponda ad un host che si trova fisicamente in Giappone. Continuando il nostro esempio, potete chiedere informazioni relative al PQDN toshiba.co.jp attraverso la query: <https://ip-api.com/#toshiba.co.jp>, il cui risultato è mostrato in Fig. 1.

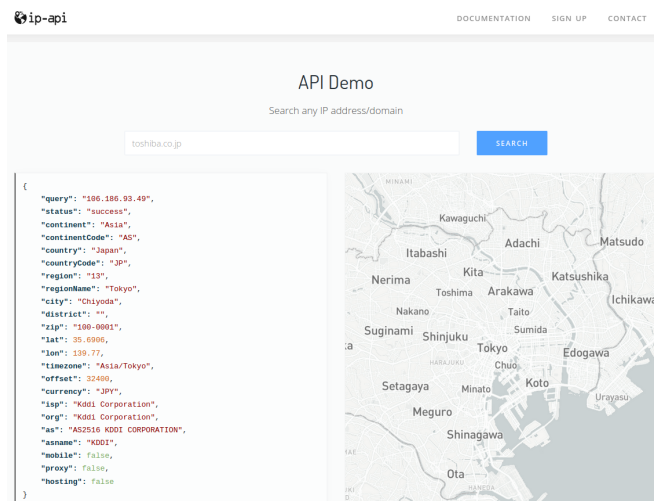


Figura 1: Esempio di query attraverso ip-api.com

Se tutti i test tramite ip-api.com sono positivi si può procedere ad inserire una nuova riga nel file IPlist.txt come mostrato in Snippet 1:

Snippet 1: Esempio di come riportare un indirizzo IP target

```
toshiba.co.jp,106.186.93.49
```

Le informazioni obbligatorie da inserire in ciascuna riga, separate da “,” sono:

1. Il PQDN (toshiba.co.jp)
2. L'indirizzo IP (106.186.93.49)

La lista completa di indirizzi IP assegnata ad ogni studente è quella contenuta nel file IPlist.txt per un totale di 33 indirizzi, ovvero:

```
// Lista degli indirizzi proposti dai docenti
//
// ITALIA: 11 indirizzi

// Cineca, consorzio per servizi web NIUR, Bologna
www.glb.cineca.it,130.186.28.141

www.assisiproloco.it,31.11.33.166

www.valleditria.it,89.40.175.190
```

```
www.produktorivini manduria.it,185.36.74.55

www.costadeifiori.it,185.21.117.35

// Università di Pisa, Pisa
wwwnew2.unipi.it,131.114.21.42

// Consorzio Nazionale Interuniversitario ↔
Telecomunicazioni, Parma
web8.cnit.it,217.9.64.17

// Università di Lecce
```

¹I termini Partially Qualified Domain Name (PQDN) e Fully Qualified Domain Name (FQDN) risulteranno più chiari quando a lezione affronteremo il Domain Name System (DNS). Si riferiscono a diverse caratteristiche di “mappatura” tra il nome a dominio e gli indirizzi IP, l'ideale per questo esperimento sarebbe usare FQDN che garantiscono una mappatura molto stabile con gli indirizzi IP, purtroppo questi nomi sono molto spesso poco significativi perché si riferiscono a una nomenclatura interna di una azienda per “riconoscere” un host o un cluster di host fisici o frutto di una virtualizzazione delle risorse.

```

www.unile.it,193.204.68.200

//Provincia autonoma di Bolzano
www.provincia.bz.it,62.101.0.185

//Provincia Sassari
www.provincia.sassari.it,151.11.50.191

//Biblioteca regionale di Palermo
www.bibliotecaregionalepalermo.it,88.33.205.19

// Europa: 9 indirizzi

// Lodz, Poland
mariacki.com,195.162.24.96

// KTH, Stoccolma
www.kth.se,130.237.28.40

// Nokia, London
nokia.com,162.13.40.196

// DomainFactory, Cologne, Germany
www.hausderwirtschaft.de,134.119.245.6

//Technische Universitat Muenchen
tum.de,129.187.255.151

//Universita' statale di Mosca
msu.ru,178.154.215.92

//Museo Louvre
www.louvre.fr,89.185.38.196

//Universita' di Valencia
www.uv.es,147.156.200.249

//Universita' Atene
uoa.gr,195.134.71.229

// Nord America, Costa Est: 3 indirizzi

// Computer Science Columbia University, New York
webcluster.cs.columbia.edu,128.59.11.206

// Anexio Corporation, North Carolina
anexio.com,69.10.48.106

// CenturyLink, Tampa
www.airboatusa.com,66.77.206.79

// Nord America, Costa Ovest: 3 indirizzi

// Isomedia, Seattle
www.isomedia.com,207.115.64.56

// Switch, Las Vegas

www.switch.com,216.115.77.36

// Unified Layer, Provo, Utah
www.downtowncasper.com,162.241.252.194

// Sud America : 4 indirizzi

// Prefeitura da Cidade do Rio de Janeiro
rio.rj.gov.br,187.111.98.123

// universidad central del Ecuador
www.uce.edu.ec,200.12.169.30

// NSS, Buenos Aires, Argentina
www.uthgracap.com.ar,201.216.200.117

// ANT, Montevideo
www.carlossicco.com.uy,179.27.153.55

// Far East (Cina esclusa): 5 indirizzi

// Toshiba Corporation, Giappone
toshiba.co.jp,106.186.93.49

// Sakura Internet, Osaka
www.hvf.jp,59.106.165.210

// WALLE, Hanoi
ctnhp.com.vn,103.101.163.94

// University of Christchurch, South Island, NZ
www.canterbury.ac.nz,132.181.106.9

// Young Australia Mines, Sydney, NSW, AU
yamines.com.au,122.201.88.33

// Cina: 3 indirizzi

// China Unicom, Shanghai
www.snhm.org.cn,210.22.96.81

// Chinanet, Chengdu
www.scmuseum.cn,118.122.91.2

// Beihang University, Pechino
proxy.buaa.edu.cn,106.39.41.46

// Africa: 3 indirizzi

// Algerie Telecom, Batna, Algeria
algeriatelecom.dz,197.112.0.130

// University of Lagos, Nigeria
unilag.edu.ng,196.45.48.45

// Telecom Malagasy, Antananarivo, Madagascar
www.ore.mg,41.188.7.72

```

Ogni studente è libero di estendere questa lista. Le linee che cominciano con # o // verranno ignorate, così come le linee vuote.

NB: prima di aggiungere una nuova entry al file `IPlist.txt`, è necessario verificare che il comando ping, eseguito con il tal IP o PQDN come argomento, raggiunga un host capace di rispondere ai messaggi ICMP.

Aggiungere indirizzi su cui si fanno gli esperimenti è uno dei modi per aumentare il punteggio assegnato. Ovviamente aggiungere un singolo indirizzo non aggiunge molto al nostro progetto, ma raddoppiarli è sicuramente molto significativo e triplicarli ancora di più. Potete collaborare per trovare indirizzi significativi, anzi, vi invitiamo a “dialogare” in modo da avere più misure verso le stesse destinazioni da operatori diversi e anche a verificare tra di voi che gli indirizzi trovati siano significativi e stabili ...almeno quanto quelli che proponiamo noi. I nuovi indirizzi possono essere aggiunti al file `IPlist.txt`, oppure tenuti in file separati e copiati nel file `IPlist.txt` a seconda

degli esperimenti che volete fare. Potete ovviamente estendere `autoping.py` in modo da passare un filename diverso da `IPlist.txt` come parametro, ad esempio, implementando un nuovo argomento `--fileinput`.

1.2 Uso di `autoping.py`

`autoping.py` coordina 3 fasi principali degli esperimenti:

1. Configurazione
2. Esecuzione
3. Elaborazione/Salvataggio risultati

1.2.1 Configurazione

Eseguendo lo script `autoping.py` come mostrato in Snippet 2:

Snippet 2: Prima esecuzione di `autoping.py`

```
python autoping.py
```

una procedura guida lo studente nell'inserimento dei parametri di configurazione, come mostrato in Snippet 3:

Snippet 3: Eseguire il configuratore esperimenti

```
# CONFIGURING EXPERIMENTS-----
Please, insert the requested parameters to configure the experiments
Enter your first-name: renato
Enter your surname: locigno
Enter your CAP code: 25100
Who is your Internet Operator?: fastweb
Your connected to Internet from? ['HOME', 'MOBILE', 'UNIBS', 'OTHER']
home
What's your access technology? ['FTTC', 'FTTH', 'ADSL', 'FWA']
fttc
You connection is over WiFi or Ethernet? ['WIFI', 'ETHERNET']
wifi
Configuration concluded: this setup has been saved to /home/renato/Desktop/tools/↵
    confRENATO_LOCIGNO_25100_fastweb_HOME_FTTC_WIFI.json
You can reuse it later using the --config option
Configuration Summary:
{'cap': '25100',
 'localtech': 'WIFI',
 'name': 'RENATO',
 'operator': 'fastweb',
 'poa': 'HOME',
 'surname': 'LOCIGNO',
 'tech': 'FTTC'}
```

le impostazioni scelte vengono registrate in un file json, per poter essere riutilizzate in seguito. Disponendo di un file di configurazione valido è infatti possibile eseguire il `autoping.py` come mostrato in Snippet 4:

Snippet 4: Uso di `autoping` con un file di configurazione

```
$ python autoping.py -c confRENATO_LOCIGNO_25100_fastweb_HOME_FTTC_WIFI.json
# CONFIGURING EXPERIMENTS-----
Configuration Summary:
{'cap': '25100',
 'localtech': 'WIFI',
 'name': 'RENATO',
 'operator': 'fastweb',
 'poa': 'HOME',
 'surname': 'LOCIGNO',
 'tech': 'FTTC'}
```

L'uso dei file di configurazione è conveniente per ripetere esperimenti con la medesima configurazione, cosa che è richiesta per la corretta esecuzione degli esperimenti. Uno studente che volesse ripetere gli esperimenti ma in condizioni diverse (ad esempio dalla connessione cellulare anziché da casa) **deve eseguire nuovamente** `autoping.py` **per creare un secondo file di configurazione, che verrà memorizzato per successive esecuzioni, senza cancellare il precedente.**

1.2.2 Esecuzione

Se si dispone di un file `IPlist.txt` valido (nella stessa cartella dove viene eseguito `autoping.py`) e di una configurazione valida (nuova o caricata da file), `autoping.py` procede con l'esecuzione del comando ping verso tutte le entry valide elencate in `IPlist.txt`. Un esempio di questa fase è mostrato in Snippet 5:

Snippet 5: Esempio di esecuzione di `autoping.py`

```
[Omitted...]

# BUILDING LIST OF PING COMMANDS TO BE EXECUTED-----
Scanning IPlist.csv to validate IP addresses and QDNs...
Validating line 97: www.ore.mg,41...: 100%|#####| 33/33 [00:39<00:00, 1.19s/it]

Found #33 valid IP that will be pinged:

130.186.28.141 --> www.glb.cineca.it (IT)
162.13.40.196 --> nokia.com (GB)
134.119.245.6 --> www.hausderwirtschaft.de (DE)
128.59.11.206 --> webcluster.cs.columbia.edu (US)
[Omitted output]
106.186.93.49 --> toshiba.co.jp (JP)
103.101.163.94 --> ctnhp.com.vn (VN)
122.201.88.33 --> yamines.com.au (AU)
106.39.41.46 --> proxy.buaa.edu.cn (CN)
197.112.0.130 --> algerietelecom.dz (DZ)

Building output folder out/

PERFORMING PING COMMANDS...(in parallel subprocesses)
Finished to ping: 130.186.28.141 --> www.glb.cineca.it (IT), Result: OK
[Omitted output]
Finished to ping: 197.112.0.130 --> algerietelecom.dz (DZ), Result: OK

100% (33 of 33) |#####| Elapsed Time: 0:00:20 Time: 0:00:20
-----
```

NB: Prima di eseguire gli esperimenti con ping, `autoping.py` valida gli indirizzi IP (o PQDN) contenuti in `IPlist.txt` interrogando opportunamente l'API `ip-api.com`. Indirizzi non validi o PQDNs non combacianti con l'IP fornito in `IPlist.txt` vengono scartati in questa fase di validazione. Gli indirizzi non validi vengono mostrati nel terminale ed è opportuno rimuoverli da `IPlist.txt`.

Per gli indirizzi IP validi, `autoping.py` procede eseguendo il comando ping come mostrato in Snippet 5. L'output di ogni comando ping eseguito viene salvato nella cartella di output `out/`. Per quanti controlli e test si facciano è sempre possibile, purtroppo, che qualcuno dei comandi ping non vada a buon fine (timeout applicativi e simili incidenti) o che alcuni file di output risultino non interpretabili dal parser sviluppato per estrarre le statistiche raccolte dal comando ping. Per esempio, i file di output generati su macchine Windows con lingua installata diversa da Italiano o Inglese non sono supportati.

Il contenuto di un log, ottenuto eseguendo `autoping.py` su una macchina Linux (Ubuntu 20.04 LTS), viene mostrato a titolo di esempio in Snippet 6.

Snippet 6: Contenuto di un file di log

```
PING 106.186.93.49 (106.186.93.49) 56(84) bytes of data.
64 bytes from 106.186.93.49: icmp_seq=1 ttl=239 time=380 ms
64 bytes from 106.186.93.49: icmp_seq=2 ttl=239 time=302 ms
64 bytes from 106.186.93.49: icmp_seq=3 ttl=239 time=325 ms
[Omitted...]
64 bytes from 106.186.93.49: icmp_seq=100 ttl=239 time=304 ms

--- 106.186.93.49 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 90004ms
rtt min/avg/max/mdev = 282.010/319.113/380.611/27.054 ms
```

1.2.3 Elaborazione/Salvataggio risultati

Terminata la fase di configurazione e quella di esecuzione, `autoping.py` procede con l'elaborazione dei file di log salvati in `out/`. Questa cartella, col tempo, contiene non soltanto i log creati dall'esecuzione corrente di `autoping.py`, ma anche log ottenuti precedentemente, così da non perdere preziosi risultati. Alcuni di questi log potrebbero risultare vuoti o non interpretabili da `autoping.py`: lo studente è invitato a rimuovere questi log dalla cartella `out/`. Un esempio della fase conclusiva di `autoping.py`, fase in cui tutti i log disponibili vengono analizzati e le relative statistiche estratte e salvate in un file di risultati, è mostrato in Snippet 7.

Snippet 7: Esempio di output della fase di elaborazione dei log.

```
Looking for logs inside out
Found these log-files:
--> ping_RENATO_LOCIGNO_25100_fastweb_HOME_FTTTC_WIFI_59-106-165-210_JP_14112020-12h31m48s.txt
--> ping_RENATO_LOCIGNO_25100_fastweb_HOME_FTTTC_WIFI_190-205-102-254_VE_14112020-12h31m35s.txt
[Omitted...]

Checking if logs have a valid filename...

CHECK COMPLETED. ValidLogs=38 NotValid=0
-----

Scanning valid logs to extract ping statistics...

SCAN COMPLETED
-----

surname    cap operator   poa accessTech ... minRTT   avgRTT   maxRTT   mdevRTT   TX  RX  lost
LOCIGNO    25100 fastweb    HOME      FTTC ... 280.916  333.987  390.216   32.710   81  80  1.23
LOCIGNO    25100 fastweb    HOME      FTTC ... 338.335  349.388  360.441   11.053    2   2  0.00
LOCIGNO    25100 fastweb    HOME      FTTC ... 237.454  278.169  316.035   30.830    4   4  0.00
[Omitted...]

[38 rows x 17 columns]
All your valid results have been saved to results_RENATO_LOCIGNO.csv

ALL RIGHT! All logs are well formatted and provided useful results, thank you!
```

Tutte le statistiche elaborate correttamente vengono registrate nel file `results_NOME_COGNOME.csv`. Aprendo tale file con un elaboratore di fogli di calcolo lo studente dovrebbe poter visualizzare tutte le statistiche come mostrato in Fig. 2. Attenzione a non modificare in alcun modo il formato del file `.csv`, altrimenti, potreste rendere tale file `csv` non processabile dallo script che elabora i grafici per illustrare i dati raccolti.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
name	surname	cap	operator	poa	accessTech	localTech	country	datetime	IP	minRTT	avgRTT	maxRTT	mdevRTT	TX	RX	lost
RENATO	LOCIGNO	25121	fastweb	HOME	FTTC	WIFI	JPN	2020-11-10 15:37:03	106.186.93.49	282.01000	319.11300	380.61100	27.05400	100	100	0.00000
RENATO	LOCIGNO	25121	fastweb	HOME	FTTC	WIFI	BRA	2020-11-10 15:36:54	198.49.133.253	224.79300	281.38600	333.47000	34.53600	100	100	0.00000

Figura 2: Alcuni risultati raccolti nel file `results.csv` mostrati in LibreOffice.

1.3 Uso degli argomenti con `autoping.py`

Per ottenere un messaggio di aiuto circa gli argomenti utili supportati da `autoping.py`, è sufficiente digitare:

```
python autoping.py -h
```

oppure

```
python autoping.py --help
```

In entrambi i casi l'utente potrà visualizzare il messaggio di aiuto mostrato in Snippet 8:

Snippet 8: Messaggio di aiuto per capire il funzionamento di autoping.py

```
usage: Run this script without arguments or, if you have a configuration-file, pass it via the -c (--config)↵
option.
-h gives you a help message
-c (or --config)      OPTIONAL argument to provide a configuration file
-n (or --numping)     OPTIONAL argument to indicate the number of icmp_echo_request to be sent
                     with each ping command. The default value is 100
-p (or --postprocess) OPTIONAL argument to provide the folder that contains the log that must be ↵
                     postprocessed.
                     If --postprocess is set, the script will just postprocess logs, without performing↵
                     new experiments
-j (or --numcores)    OPTIONAL argument to indicate how many processes shall be used to launch many ping↵
                     commands in parallel

Try with this:
python3 autoping.py
```

Si ribadisce quindi il ruolo e l'uso possibile dei seguenti argomenti:

- -c o --config per caricare un file di configurazione. Ad es:

```
python autoping.py -c confRENATO_LOCIGNO_25100_fastweb_HOME_FTTC_WIFI.json
```

- -n o --numping per indicare il numero di icmp_echo_request da inviare col comando ping.
NB: il valore di default è 100, e può essere utile ridurlo (anche di molto) per una fase di “test” prima di eseguire le misure vere e proprie.

```
python autoping.py -n 8
```

- -p o --postprocess per indicare la cartella che contiene i log da elaborare. Se questo argomento viene fornito, allora autoping.py procede solo con l'elaborazione dei log, senza eseguire nuove configurazioni o nuovi esperimenti.

```
python autoping.py -p out/
```

- -j o --numcores per indicare il numero di processi da utilizzare per lanciare in parallelo diversi comandi ping. L'uso di questo parametro è limitato ad un numero massimo di 32 processi paralleli: si consiglia l'abilitazione di un discreto numero di processi paralleli (5–10) per accelerare l'esecuzione di tutti i comandi ping e quindi rendere più veloce la raccolta dati, tuttavia bisogna evitare di “sovraccaricare” il sistema rischiando di falsare i risultati e rischiando che il nostro provider ritenga eccessivo il numero di messaggi ICMP che arrivano dalla nostra linea e pensi a un attacco di denial of service o simili.

1.4 Uso di rtt_plotter.py

rtt_plotter.py parte da uno o più file .csv prodotti da autoping.py per generare alcuni grafici che consentono di visualizzare alcune caratteristiche dei dati raccolti. rtt_plotter.py richiede un solo argomento obbligatorio:

- -f o --finput per caricare un file .csv oppure una cartella che contiene file .csv prodotti da autoping.py

L'output del post processing grafico è prodotto in una cartella .\plot\ che contiene vari plot in formato .pdf organizzati in tre sottocartelle:

```
plot/
|-- LOSSES/
|      |-- hist_losses_bw=1.pdf
|      |-- hist_losses_bw=2.pdf
|      |-- hist_losses_bw=3.pdf
|      |-- hist_losses_bw=5.pdf
|-- RTT/
|      |-- hist_meanRTT_bw=10.pdf
```

```
|      |-- hist_meanRTT_bw=20.pdf
|      |-- hist_meanRTT_bw=5.pdf
|-- SD/
|      |-- hist_sdRTT_bw=1.pdf
|      |-- hist_sdRTT_bw=2.pdf
|      |-- hist_sdRTT_bw=5.pdf
|-- meanRTT-distribCOMPARISON-by-CONTINENT.pdf
|-- meanRTT-distribCOMPARISON-by-WORLDCATEGORIES.pdf
|-- meanRTT-distribCOMPARISON.pdf
|-- meanRTTmeanCOMPARISON.pdf
|-- OPERATORcomparisonHOME.pdf
|-- OPERATORcomparisonMOBILE.pdf
|-- OPERATOR-violin-HOMEvsMOBILE-byContinent.pdf
```

`bw` è l'ampiezza del bin dell'istogramma del grafico, i file nella cartella `.plot\` sono dei boxplot con valor medio, mediana, 25° (Q_1) e 75° (Q_3) percentile, con i “baffi” che si estendono a coprire fino all'ultimo punto compreso nell'intervallo $Q_1 - 1.5(Q_3 - Q_1)$ e $Q_3 + 1.5(Q_3 - Q_1)$. Ulteriori informazioni per “giocare” con i bloxplot si trovano su https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.boxplot.html.

Fa eccezione il file `meanRTTmeanCOMPARISON` che è un semplice istogramma delle medie.

Attenzione: `rtt_plotter.py` riscrive i file ad ogni nuova esecuzione quindi, se volete salvare dei plot ottenuti processando dataset di misure diverse, dovete rinominare la cartella `plot` o i file `file` al suo interno.

2 Esecuzione degli esperimenti e consegna dei dati

Ciascuna esecuzione del comando

```
python autoping.py -c <conffile.json>
```

produce un file di output con nome univoco nella cartella `out/`, senza cancellare i file generati durante esperimenti precedenti. Alla fine dell'esperimento lo script processa automaticamente tutti i file all'interno di `out/` e scrive il riassunto dei risultati nel file `results_NOME_COGNOME.csv`. Quando avrete terminato la vostra raccolta dati vi verrà chiesto di caricare su moodle il vostro `results_NOME_COGNOME.csv` finale.

2.1 Set minimo di misure, estensione e “punti guadagnati”

Il requisito minimo per partecipare al progetto è **eseguire 10 volte a distanza di alcune ore l'una dall'altra** lo script `autoping.py` **con -n 100** (valore di default) e poi caricare i risultati su moodle, nel relativo assignment che troverete nella vostra area (seguiranno istruzioni più dettagliate).

I risultati comprendono il file `results_NOME_COGNOME.csv` e uno zip di tutti i log, quindi un file `nome-cognome_log.zip` con tutto il contenuto della cartella `out/`. **N.B.** il file deve essere uno zip, non rar, targz o altro!!

I file potranno essere caricati più volte, fino a domenica 5 Dicembre. La consegna di nuovi file sovrascrive quelli vecchi, noi conteremo solamente i risultati finali (attenzione a non consegnare file vuoti all'ultimo momento!), ma vi invitiamo a caricare risultati, anche non definitivi, appena ne avete perché questo consente di iniziare l'elaborazione grafica e anche di mettervi a disposizione “strumenti” ulteriori per guadagnare punti.

Come descritto sopra il requisito minimo (quindi 1 punto in più all'esame) è l'esecuzione dell'esperimento in modo corretto 10 volte con gli indirizzi IP che vi abbiamo suggerito noi. Il modo più semplice (e anche un po' noioso a essere sinceri) per guadagnare ulteriori punti è aggiungere indirizzi come descritto nella sezione 1.1: raddoppiare gli indirizzi (2 punti), triplicarli o più (3 punti), oppure fare test con operatori diversi: 2 operatori diversi vale due punti ecc. ... Ovviamente gli indirizzi devono funzionare ed essere opportunamente testati in modo che se proviamo anche noi non ci siano sorprese.

Tuttavia potete aumentare i punti guadagnati anche in altri modi ... tanti altri modi, la cosa che apprezziamo di più è che venga da voi una proposta valida, quelli che seguono sono solo alcuni esempi.

- Sviluppando in autonomia uno script python che automatizza la ripetizione delle misure in istanti di tempo diversi e selezionati (orario di punta, notte, ...).

- Estendendo il generatore di plot per produrre visualizzazioni più sofisticate ed avanzate.
-
- Qualunque idea per fare una “misura” su e di Internet da casa vostra, che non sia usare i siti per misurare la velocità dell’ADSL!

Quindi è possibile guadagnare punti semplicemente (!?) con idee che funzionano per migliorare la raccolta dei dati o la loro qualità, ma è anche possibile guadagnare punti con idee varie per il post-processing dei dati, oppure ancora per disegnare le rotte calcolate con traceroute su una mappa (i router spesso sono geolocalizzati anche se in modo approssimativo), oppure ...

Metteremo a disposizione di **tutti** tutti i dati raccolti ogni 3-4 giorni. A quel punto potrete giocare anche voi con dati e script per ricavare elaborando non solo i vostri dati ma quelli raccolti da tutti i partecipanti: chiunque fornisca elaborazioni avanzate dei dati raccolti offrendo nuove prospettive (grafici o altro) per la loro interpretazione ottiene anche un punteggio più alto.

N.B. I dati vanno raccolti e consegnati entro il 5 dicembre come indicato su Moodle, tuttavia il progetto resta aperto per chi vuole collaborare e contribuire all’interpretazione dei dati, che può portare anche ad aumentare i punti assegnati (non oltre 3), oltre che a essere molto utile per capire come manipolare, capire e presentare dati di misura. Chiuderemo la fase di consegna dei contributi all’elaborazione dei dati il 23 dicembre (la registrazione dei voti, anche per chi fa il preappello sarà insieme a primo appello di gennaio), durante le vacanze di Natale assegneremo tutti i punti guadagnati e a inizio gennaio procediamo con la pubblicazione del rapporto finale.

2.2 Pubblicazione dei dati

Scriveremo un rapporto su queste misure simile a quello dello scorso anno, eventualmente con “confronti evolutivi”, da pubblicare sul nostro sito web, ma anche da proporre a qualche giornale locale se vogliono scrivere un articolo... o magari intervistare qualcuno di voi sull’esperienza. Nell’assignment di Moodle vi chiederemo anche se volete comparire con nome e cognome su questo report (non influenza la votazione, è una libera scelta personale) e se volete potrete caricare una vostra foto da usare per fare un “mashup” del gruppo di lavoro da includere nel report.

3 Guida all'installazione

Questa piccola guida è scritta per python3.7. Sostituire a seconda della propria preferenza/disponibilità di python nel sistema "3.7" con altre versioni di python, purchè ≥ 3.6 . Questa guida è stata testata su una macchina Ubuntu 20.04 LTS.

Installazione con pip automatica

```
python3 -m pip install -r requirements.txt
```

Installazione con pip manuale

```
python3 -m pip install --upgrade numpy
python3 -m pip install --upgrade matplotlib
python3 -m pip install --upgrade seaborn
python3 -m pip install --upgrade pandas
python3 -m pip install --upgrade pandas_schema
python3 -m pip install --upgrade pycountry_convert
python3 -m pip install --upgrade progressbar2
```

3.1 Note utili per sistemi Linux

Per installare una versione di python (ad es., la 3.7):

```
sudo apt install python3.7
```

Per impostare la versione di sistema di python3:

```
sudo update-alternatives --config python3
```

3.2 Note utili per sistemi Windows

Per installare Python con Microsoft Store:

1. Passare al menu Start (icona di Windows in basso a sinistra), digitare "Microsoft Store" e selezionare il collegamento per aprire lo Store.
2. Dopo l'apertura dello Store, selezionare Ricerca nel menu in alto a destra e immettere "Python". Aprire "Python 3.8" nei risultati visualizzati in App. Selezionare Ottieni.
3. Al termine del processo di download e installazione di Python, aprire Windows PowerShell usando il menu Start (icona di Windows in basso a sinistra). Dopo l'apertura di PowerShell, immettere `python -V` per verificare che Python3 sia stato installato nel computer.
4. L'installazione di Python da Microsoft Store include pip, l'utilità standard di gestione pacchetti. Pip consente di installare e gestire altri pacchetti che non fanno parte della libreria standard di Python. Per verificare se è disponibile anche pip per l'installazione e la gestione dei pacchetti, immettere `pip --version`.

Link Utili

- Introduzione all'uso di Python in Windows per principianti
<https://docs.microsoft.com/it-it/windows/python/beginners>
- Link a python3.7 nello store <https://www.microsoft.com/en-us/p/python-37/9nj46sx7x90p>