

DVR Simulator

Come avviare il progetto

Aprire un terminale nella directory del progetto e seguire le istruzioni in base al sistema operativo:

Windows

Eseguire il seguente comando:

```
.\start.bat
```

Mac / Linux

Eseguire il seguente comando:

```
chmod 770 ./start.sh && ./start.sh
```

il comando `chmod` è utilizzato per dare i permessi allo script di avvio di essere eseguito, altrimenti lo script non viene riconosciuto come tale dal SO.

Comportamento del progetto all'avvio

Lo script "start" crea ed esegue un virtual enviroment python in cui verranno installate le dipendenze del progetto, e poi successivamente il progetto verrà eseguito proprio in questo enviroment; questa scelta è molto utile dato che di fatto il virtual enviroment è una cartella, che se eliminata può essere ricreata a piacimento dallo script o anche a mano se lo si desidera.

Una volta installate le dipendenze, che si trovano nel file "requirements.txt", il progetto si avvierà nel terminale corrente mostrando la seguente schermata

```

Lista dei file presenti nella cartella NetworksDefaultTypes:
0 - FullyConnected.py
1 - HomeToGoogle.py
2 - SimpleGraph.py
3 - StarCenter.py
Quale rete si vuole creare? █

```

Nella quale l'utente può scegliere quale tipo di network già implementate creare e poi visualizzare.

Questo comporta che il progetto può essere ampliato con qualsiasi rete custom l'utente vuole creare, basterebbe utilizzare i metodi già implementati della classe Net.py e Node.py e poi inserire il file custom nella cartella NetworksDefaultTypes. Una volta scelta il tipo di network da creare l'utente dovrà inserire nel terminale il numero presente affianco al file, ad esempio, se si vuole creare una rete fully connected si deve digitare zero e premere invio.

Scelta la rete ed inserito il numero di riferimento apparirà a schermo una immagine creata sul momento della rete, utile per verificare che le routing tables siano giuste, ed inoltre la stessa immagine verrà salvata nella cartella NetImages presente nella route del progetto (se non è presente verrà creata).

Le routing tables verranno rese disponibili per la lettura nel terminale sotto forma di tabella come quella sulla destra.

Inoltre ogni rete avrà un file .log personale in cui vengono salvate le routing tables (questo file compare una volta eseguito il file).

Tabella di routing per il nodo X:

```

+-----+-----+-----+
----+
| Destinazione | Costo | Passando
da |
+-----+-----+-----+
-----+
|      Z      |  N   |    Y
|
+-----+-----+-----+
----+

```

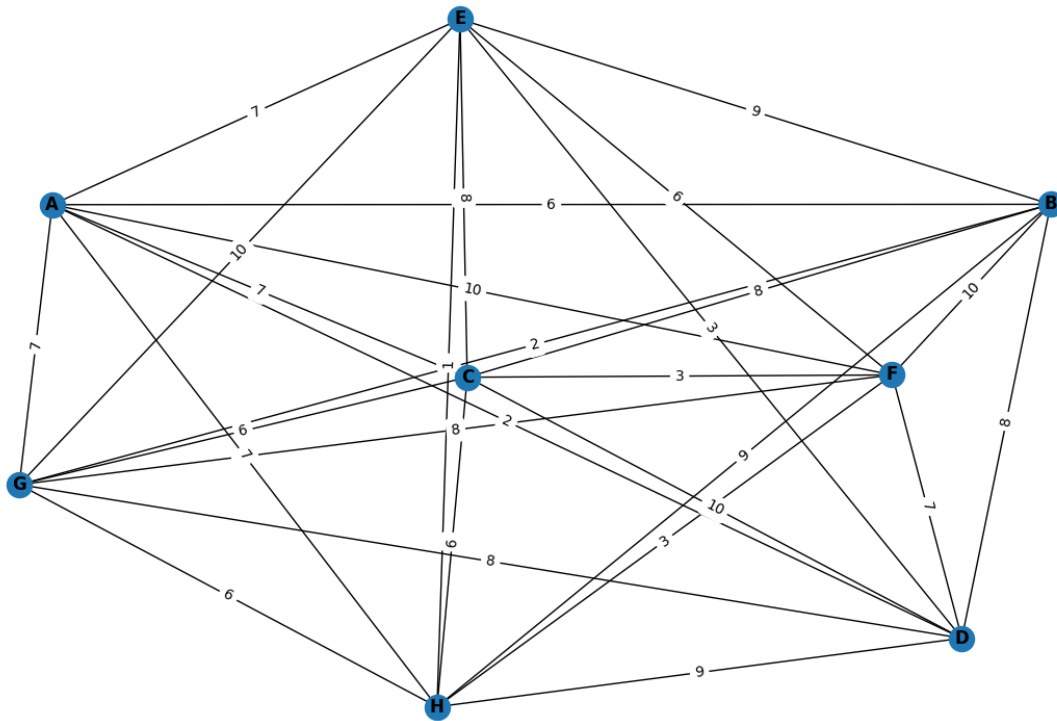
Una volta chiusa l'immagine il programma ripartirà da capo richiedendo quale network si vuole visualizzare.

Reti già presenti

FullyConnected

Realizza una network composta da 8 nodi che sono tutti quanti connessi fra loro creando una maglia completa, quindi ogni nodo è collegato a tutti gli altri nodi.

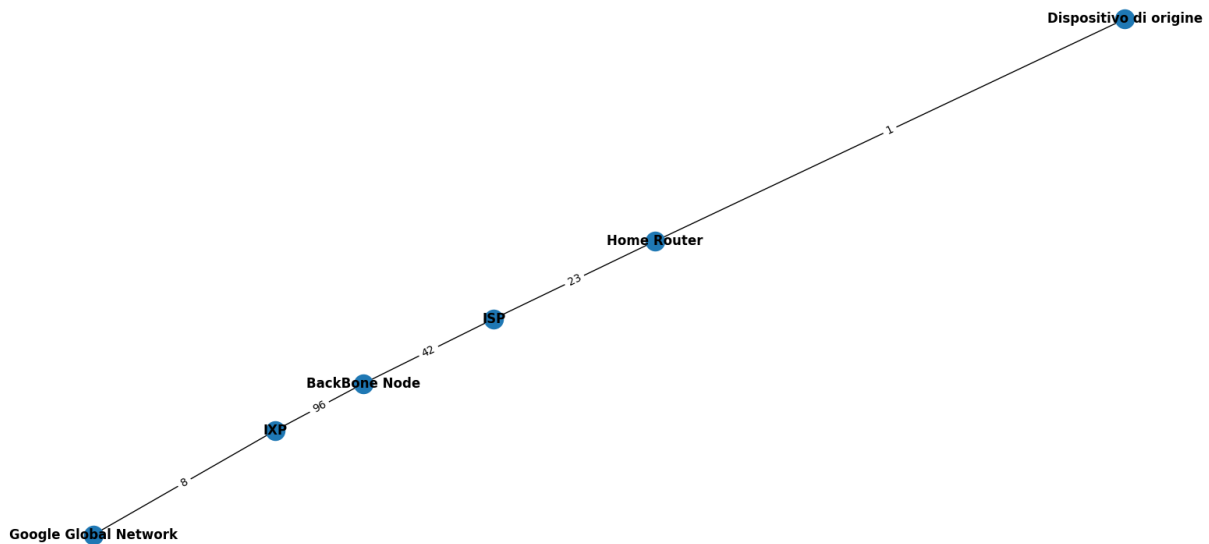
La foto realizzata sarà simile a la seguente:



HomeToGoogle

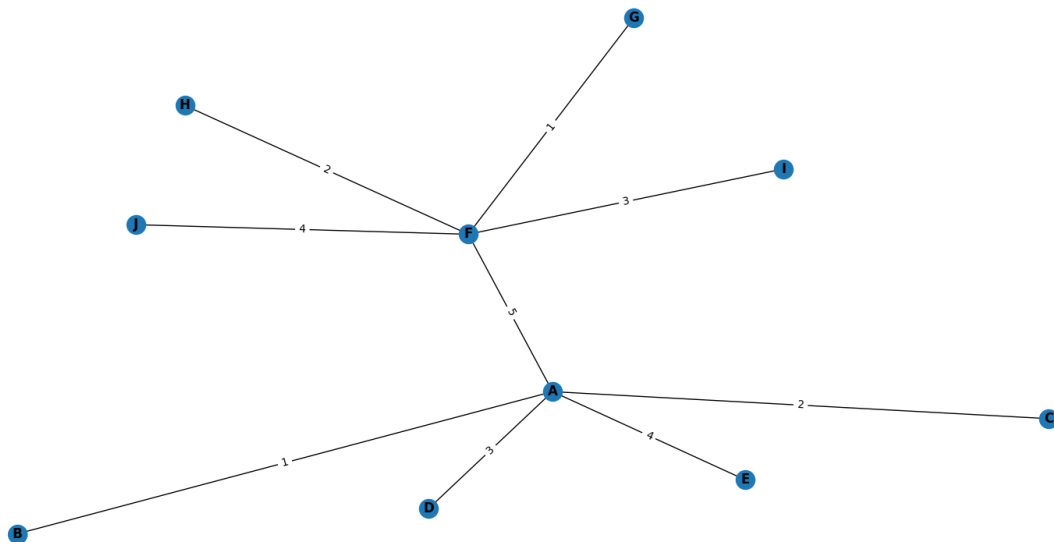
In questo tipo di network ho voluto rappresentare il tragitto che un pacchetto mandato da un computer domestico verso il server centrale di google percorrerebbe, ovviamente è molto semplificata data la moltitudine di nodi presenti in questo tragitto nella realtà.

Il tragitto avrebbe un aspetto di questo tipo:



SimpleGraph

Questo file python realizza una network in cui sono presenti due centri stella, ovvero due nodi ai quali sono collegati molteplici altri nodi, ed in più, questi due nodi sono collegati fra loro, realizzando, di fatto, una rete di questo tipo:



StarCenter

Crea un nodo centro stella, quindi tutti gli altri nodi sono solo collegati ad esso.

