



SAPIENZA  
UNIVERSITÀ DI ROMA

## Generazione e visualizzazione grafica di traffico di reti

Facoltà di Ingegneria dell'Informazione, Informatica e Statistica  
Laurea Triennale in Informatica

**Francesco Pannozzo**

Matricola 699427

Relatore

Prof. Daniele De Sensi

Anno Accademico 2023/2024

Tesi non ancora discussa

---

**Generazione e visualizzazione grafica di traffico di reti**

Relazione di tirocinio. Sapienza Università di Roma

© 2024 Francesco Pannozzo. Tutti i diritti riservati

Questa tesi è stata composta con L<sup>A</sup>T<sub>E</sub>X e la classe Sapthesis.

Email dell'autore: [francesco.pannozzo@libero.it](mailto:francesco.pannozzo@libero.it)

*Dedicato alla  
mia famiglia*



## Sommario

Questa relazione descrive il lavoro di tirocinio interno svolto presso l'università La Sapienza, concretizzato nella realizzazione di un progetto volto a realizzare un software per poter visualizzare in forma grafica l'andamento del traffico di una rete. Il progetto ha come obiettivo di mostrare il traffico di rete al variare del tempo e ciò viene raggiunto tramite grafiche e animazioni generate programmaticamente. L'idea dell'ambito di tirocinio nasce dalla volontà di sperimentare una realizzazione front-end tramite la libreria Manim, un motore di animazioni per video matematici esplicativi..



# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Ambito del tirocinio . . . . .	1
1.2	Motivazioni . . . . .	2
1.3	Stato dell'arte . . . . .	2
<b>2</b>	<b>Progettazione</b>	<b>5</b>
<b>3</b>	<b>Test</b>	<b>7</b>
3.1	Sotto capitolo test . . . . .	8





# Capitolo 1

## Introduzione

Nel mondo le reti informatiche sono oramai un concetto ben istanziato nella collettività, la loro presenza è soverchiante e si dirama nei più disparati settori. Basti pensare già alle reti PAN (Personal Area Network) le quali connettono dispositivi personali entro pochi metri e che ognuno di noi usa abitualmente nella propria casa, alle reti LAN (Local Area Network), anch'esse presenti nelle nostre case così come in uffici o edifici scolastici, le reti dei datacenter fino a giungere alla rete globale internet, la quale è creatrice a sua volta di paradigmi come può essere l'internet of things. Le reti informatiche sono impiegate nei più vari settori come l'istruzione, in cui le reti sono cruciali nelle scuole e nelle università per avere accesso a risorse educative o sfruttare l'e-learning, i servizi pubblici governativi e sanitari, nel settore ludico e multimediale come il gioco online e l'attuale streaming di contenuti multimediali: insomma, le reti informatiche sono di fatto una presenza piena e diffusissima ed è estremamente difficile riuscire a immaginare il mondo come lo vediamo oggi senza questa tecnologia. Con l'aumentare delle funzionalità legate alle reti, così come i dispositivi collegati a esse, capire cosa succede al loro interno, come si muovono i dati, è quindi di cruciale importanza, tramite l'analisi dei dati che vi fruiscono è possibile fare diagnostica, per quanto riguarda un discorso di monitoraggio, ma anche è possibile applicare le analisi in un ambito didattico e accademico. Capire cosa sta succedendo in una rete in modo immediato e visivo è lo scopo di questo progetto, il quale punta a mostrare, in modo grafico, l'andamento del traffico di una rete.

### 1.1 Ambito del tirocinio

Il progetto fa parte del percorso di tirocinio interno intrapreso presso l'Università La Sapienza di Roma. L'argomento su cui verte il progetto è la realizzazione di un visualizzatore grafico dell'andamento del traffico di una rete, basato su animazioni programmatiche. Il tool permette di visualizzare gli switch rappresentanti i vari endpoints e i link che li collegano i quali vengono colorati tramite animazioni nel tempo in base al traffico di rete precedentemente analizzato. Nel tool è presente anche una parte generativa di traffico di rete, una creazione di traffico fittizia di vitale importanza ai fini di testing.

## 1.2 Motivazioni

L'idea di sviluppare un visualizzatore grafico di traffico di rete è nata, in sede di proposta, dal Professore Daniele De Sensi, relatore del tirocinio, e dalla mia volontà di sviluppare un'applicazione avente il front-end come focus dell'esperienza. Nel mio personale corso di studi presso il Dipartimento di Informatica non ho avuto modo di studiare e approfondire un discorso legato al front-end, per cui la volontà di intraprendere questo percorso nasce in primis da un forte interesse verso questo aspetto dell'informatica e in secondo luogo per un completamento di formazione professionale personale.

## 1.3 Stato dell'arte

L'esigenza di analisi di reti informatiche ha portato alla luce svariati tool che permettono appunto di analizzare cosa avviene in una rete, di studiarne i dati statistici e di visualizzare graficamente determinati scenari. L'università americana Johns Hopkins[5] ha stilato una lista di software per la visualizzazione e analisi di reti[6]:

- **Gephi[2]:** Gephi è il software leader di visualizzazione ed esplorazione per tutti i tipi di grafici e reti ed è open source. Le sue caratteristiche includono l'analisi esplorativa dei dati mediante manipolazioni di reti in tempo reale, analisi dei collegamenti per rivelare le strutture sottostanti delle associazioni tra oggetti, analisi dei social network per la creazione di connettori di dati sociali per mappare le organizzazioni della comunità e le reti di piccoli mondi, analisi della rete biologica per rappresentazione di modelli di dati biologici ed esportazione e creazione poster per promuovere e divulgare il lavoro scientifico con mappe stampabili di alta qualità.
- **Cytoscape[1]:** è una piattaforma software open source per visualizzare reti complesse e integrarle con qualsiasi tipo di dati. Consiste in una piattaforma per visualizzare reti di interazioni molecolari e percorsi biologici, potendo integrare queste reti con annotazioni, profili di espressione genica e altri dati. Originariamente progettato per la ricerca biologica, ora è una piattaforma generale per l'analisi e la visualizzazione di reti complesse.
- **GraphVis[3]:** è un software di visualizzazione di grafici open source. Caratteristiche, I programmi di layout Graphviz accettano descrizioni di grafici in un semplice linguaggio di testo e creano diagrammi in formati utili, come immagini e SVG per pagine web; PDF o Postscript per l'inclusione in altri documenti; o visualizzare in un browser grafico interattivo. Graphviz ha molte funzionalità utili per diagrammi concreti, come opzioni per colori, caratteri, layout di nodi tabulari, stili di linea, collegamenti ipertestuali e forme personalizzate.
- **igraph[4]:** è una collezione di librerie per creare, manipolare grafici e analizzare ponendo l'enfasi nell'efficienza, portabilità e facilità d'uso. Igraph è open source e gratuito e può essere programmato in R, Python, Mathematica e C/C++

- **UCINET6[9]:** è un pacchetto software per l'analisi dei dati dei social network. UCINET viene fornito con il tool di visualizzazione di rete NetDraw. Può leggere e scrivere una moltitudine di file di testo diversamente formattati, nonché file Excel. I metodi di analisi dei social network includono misure di centralità, identificazione di sottogruppi, analisi di ruolo, teoria dei grafi elementari e analisi statistica basata sulla permutazione. Inoltre, il pacchetto dispone di potenti routine di analisi delle matrici, come l'algebra delle matrici e la statistica multivariata.
- **SocNetV[8]:** è un'applicazione software gratuita multiplatforma per l'analisi e la visualizzazione dei social network. Tra le caratteristiche principali troviamo il poter disegnare i social network, caricare i campi da un file supportato (GraphML, GraphViz, Adjacency, EdgeList, GML, Pajek, UCINET, ecc.), personalizzare attori e collegamenti tramite sistema punta e clicca, analizzare le proprietà dei grafici e dei social network, produrre report HTML e incorporare layout di visualizzazione di rete
- **Pajek[7]:** è un software per la visualizzazione e l'analisi delle reti. La sua forza risiede nel poter analizzare reti complesse potendo arrivare fino a un miliardo di vertici. L'analisi e la visualizzazione vengono eseguite utilizzando sei tipi di dati: rete (grafico), partizione, vettore, cluster (sottoinsieme di vertici), permutazione (riordinamento dei vertici, proprietà ordinali); e gerarchia (struttura generale ad albero sui vertici).



## Capitolo 2

# Progettazione

...



## Capitolo 3

### Test

Per ottenere i lati di un rettangolo che abbia proporzioni  $16 : 9$  partendo da un quadrato di lato  $n$ , dobbiamo innanzitutto considerare che l'area del quadrato è data da  $A = n^2$ . Vogliamo che il rettangolo abbia la stessa area del quadrato ma rispetti le proporzioni  $16 : 9$ .

Denotiamo con  $l$  la lunghezza e con  $h$  l'altezza del rettangolo. La condizione di proporzione si può esprimere come

$$\frac{l}{h} = \frac{16}{9}.$$

Dato che l'area del rettangolo deve essere uguale a quella del quadrato, abbiamo che

$$l \cdot h = n^2.$$

Utilizzando la proporzione, possiamo esprimere  $l$  in termini di  $h$  come

$$l = \frac{16}{9}h.$$

Sostituendo questa espressione nell'equazione dell'area, otteniamo

$$\frac{16}{9}h \cdot h = n^2,$$

che si semplifica in

$$\frac{16}{9}h^2 = n^2.$$

Da qui, isoliamo  $h$  ottenendo

$$h^2 = \frac{9}{16}n^2 \implies h = n \cdot \frac{3}{4}.$$

Risostituendo il valore di  $h$  nell'espressione di  $l$ , abbiamo

$$l = \frac{16}{9} \cdot n \cdot \frac{3}{4} = n \cdot \frac{4}{3}.$$

Quindi, per un quadrato di lato  $n$ , per ottenere i lati di un rettangolo che mantenga la stessa area ( $n^2$ ) con proporzioni  $16 : 9$ , l'altezza  $h$  del rettangolo sarà  $n \cdot \frac{3}{4}$  e la lunghezza  $l$  sarà  $n \cdot \frac{4}{3}$ .

Quindi il tutto funziona poichè è sempre vero quanto segue:

$$(l + 1)(m + 1) > ml \tag{3.1}$$

### 3.1 Sotto capitolo test

Ecco un esempio di codice YAML:

```
- coordinates:
  - [1, 2]
  - [3, 0]
```

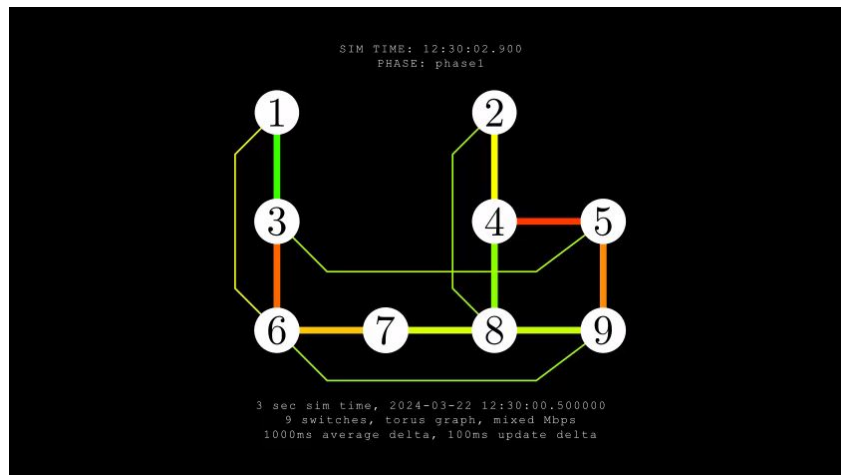
E ora un esempio di codice Python:

```
links = {}
# Extracting links data
for content in networkData[CONST.NETWORK["LINKS"]]:
    links[frozenset({content["endpoints"][CONST.EP_A],
                    content["endpoints"][CONST.EP_B]})] = {
        #"linkID": link,
        "capacity": content["capacity"],
        "trafficDT":0,
        "trafficUDT":0,
        "updateDeltaTraffic": [],
        "traffic": []
    }
```

La complessità temporale dell'algoritmo è  $O(m + n)$ .

$$O(m + n) \quad (3.2)$$

.. .. .



**Figura 3.1.** a nice plot

As you can see in the figure 3.1, the function grows near 0. Also, in the page 8 is the same example.



# Bibliografia

- [1] Cytoscape Consortium. *Cytoscape: An Open Source Platform for Complex Network Analysis and Visualization*. <https://cytoscape.org>. Accessed: 2024-04-12. 2024.
- [2] Gephi. *Gephi: The Open Graph Viz Platform*. <https://gephi.org>. Accessed: 2024-04-12. 2024.
- [3] Graphviz. *Graphviz - Graph Visualization Software*. <https://graphviz.org>. Accessed: 2024-04-12. 2024.
- [4] igraph. *igraph - The Network Analysis Package*. <https://igraph.org>. Accessed: 2024-04-12. 2024.
- [5] Johns Hopkins University. *Homepage of Johns Hopkins University*. <https://www.jhu.edu/>. Accessed: 2024-04-12. 2024.
- [6] Johns Hopkins University Libraries. *Network Data Visualization Guide*. <https://guides.library.jhu.edu/datavisualization/network>. Accessed: 2024-04-12. 2024.
- [7] Pajek. *Pajek - Program for Large Network Analysis*. <http://mrvar.fdv.uni-lj.si/pajek/>. Accessed: 2024-04-12. 2024.
- [8] SocNetV. *Social Network Visualizer (SocNetV)*. <https://socnetv.org>. Accessed: 2024-04-12. 2024.
- [9] UCINET. *UCINET Software*. <https://sites.google.com/site/ucinetsoftware/>. Accessed: 2024-04-12. 2024.

