

# Progetto SGR

Massimiliano Bertolucci

Gennaio 2014

## 1 Introduzione

Una rete di computer trova una naturale rappresentazione grafica con la struttura matematica *grafo* in cui ogni host della rete viene rappresentato come nodo del grafo e i link di collegamento tra gli host vengono rappresentati come archi tra i nodi del grafo. Tale rappresentazione però può risultare di difficile lettura se la rete che si vuole rappresentare è fortemente connessa e di grandi dimensioni. Una rappresentazione migliore in queste circostanze può essere la matrice di adiacenza. Dato un grafo con  $N$  nodi la sua matrice di adiacenza è una matrice quadrata  $N \times N$  dove sia gli indici di riga che gli indici di colonna sono i nomi dei nodi e il valore della matrice  $(i,j)$  è a 1 se solo se esiste un arco orientato tra il nodo  $i$  e il nodo  $j$ .

Esempio di grafo e la relativa matrice di adiacenza :

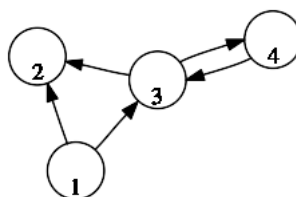


Figura 1:

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

L'obiettivo di questo progetto è quello di rappresentare il traffico di rete monitorata con il tool ntopng usando una matrice di adiacenza per mettere in evidenza e confrontare i diversi host in termini di flusso e tipologia.

## 2 Implementazione

Il tool di monitoraggio ntopng ha al suo interno un motore di scripting LuaJIT. Grazie a questo motore è possibile creare estensioni web semplicemente creando degli script in linguaggio Lua. Ho sviluppato due scripts per il mio progetto. Il primo script in linguaggio Lua permette di prelevare i dati di interesse della rete monitorata producendo in output un json.

Esempio di file json generato.

```
{
  "data": {
    "links": [
      {
        "source": 4, "source_ip": "192.168.1.105",
        "target": 10, "target_ip": "224.0.0.252",
        "value": 3, "styleColumn": "client"
      }
    ],
    "nodes": [
      {
        "name": "192.168.1.1", "count": 1,
        "count_perc": 7, "group": "local",
        "label": "192.168.1.1",
        "sent": 1, "rcvd": 0, "tot": 1
      }
    ]
  }
}
```

Il secondo script scritto in Javascript permette di elaborare i dati json e produrre il grafico della matrice di adiacenza avvalendosi della libreria d3.js. Lo script

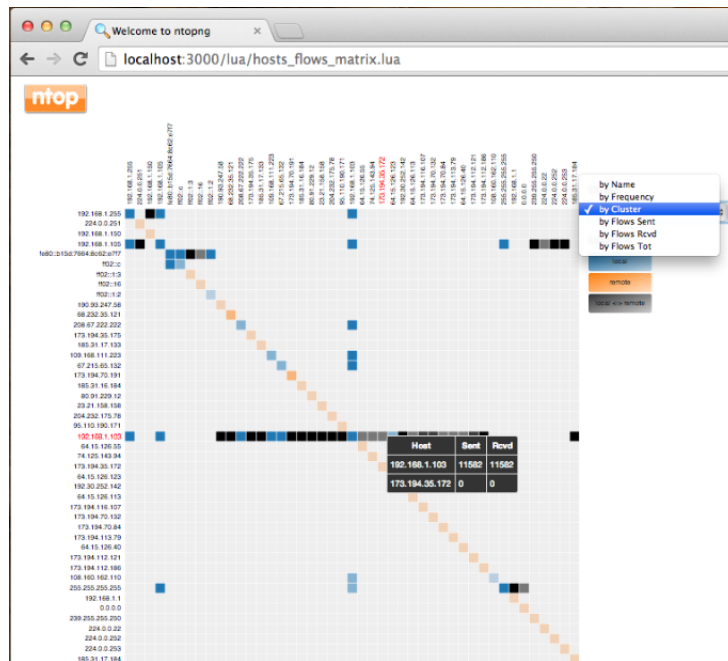


Figura 2:

Esempio grafico di matrice di adiacenza

permette di ordinare gli elementi della matrice secondo un certo criterio. E possibile ordinare la matrice secondo l'indirizzo IP oppure secondo la tipologia se locale o remota o secondo la quantita di flusso generata dalla macchina.