

Fisica dei Sistemi Complessi

Relazione di progetto

3D Network Visualizator

Giacomo Benvenuti
Enrico D'Angelo
Fabrizio Nuzzo

Sommario

La visualizzazione dei dati è lo studio della rappresentazione visuale dei dati stessi, lo scopo principale è quello di comunicare le informazioni contenute nei dati in maniera chiara ed efficace. Per trasmettere le informazioni sia l'estetica che la funzionalità della visualizzazione devono andare di pari passo, in modo da fornire la comprensione delle informazioni contenute in un dataset altrimenti complesso da comprendere. Lo scopo fondamentale della visualizzazione dei dati è quindi la comunicazione degli aspetti chiave del dataset in una maniera più intuitiva.

Il progetto svolto ha lo scopo di visualizzare le interazioni che avvengono nei diversi istanti tra i nodi di una rete. 3D-Network-Visualizator visualizza queste interazioni in maniera grafica. I nodi che compongono la rete sono disposti in uno spazio tridimensionale secondo le loro coordinate. Le interazioni vengono visualizzate come degli archi che vanno da un nodo all'altro dando una diversa ampiezza all'arco in base alla frequenza dell'interazione, per meglio capire a colpo d'occhio l'importanza relativa delle diverse interazioni. Infine gli istanti scorrono su una linea temporale dando la possibilità di visualizzare come in un video i cambiamenti che avvengono nelle interazioni tra i diversi nodi.

1 Il Problema

L'uomo per sua natura ha una spiccata capacità d'analisi critica ma questa deve essere supportata da una chiara ed appropriata visualizzazione dei dati.

Osservando direttamente un dataset che rappresenta una rete e le interazioni che avvengono tra i suoi nodi, anche se di dimensioni modeste, non è immediato rispondere a domande quali: “quali sono le interazioni che avvengono?”, “quali nodi sono quelli con maggiori o minori interazioni?”, “esistono delle interazioni ricorrenti tra certi nodi?” Che sono solo alcune delle domande di interesse per chiunque si trovi ad analizzare un dataset di questo tipo.

1.1 Una possibile soluzione

Un modo semplice per rappresentare una rete consiste nell'utilizzare una tabella con l'elenco di tutto i nodi della rete riportato sia nelle righe che nelle colonne. In questo modo nella cella che incrocia due nodi viene riportato il valore che rappresenta l'interazione avvenuta tra quei due nodi. Una tabella può però rappresentare lo stato delle interazioni della rete in un solo istante, per poter osservare le interazioni nel tempo occorre quindi utilizzare una tabella per ogni istante.

1.2 Considerazioni

Nonostante questo approccio sia molto semplice, non è auspicabile perché inefficiente dal punto di vista computazionale e della leggibilità dei risultati; Se per esempio volessimo rappresentare le interazioni che avvengono in ogni istante di tempo che non coinvolgono tutte le coppie di nodi, le tabelle risulterebbero sparse, perciò viene utilizzata una gran quantità di memoria per memorizzare in realtà una piccola quantità di informazione. Questo problema viene ulteriormente peggiorato quando la rete contiene un gran numero di nodi.

2 Specifiche di funzionamento

3D-Network Visualizator come il nome lascia intendere, un visualizzatore grafico 3D delle interazioni che avvengono tra i nodi di una rete.

- Le informazioni di input, devono essere elaborate prima della visualizzazione grafica di quest'ultime.
- L'altezza dell'arco dipende dalla frequenza (cioè il valore dell'interazione).
- Deve essere possibile capire la velocità con cui i nodi comunicano

- Deve essere possibile cambiare l'angolazione della visualizzazione
- Deve essere possibile partizionare la rete per analizzare una parte di essa
- La visualizzazione deve essere personalizzabile tramite varie opzioni
- Deve essere possibile variare il tempo di visualizzazione di un istante

3 Tecnologie utilizzate

Per la realizzazione di questo progetto sono state utilizzate diverse tecnologie:

Processing: è una libreria di sviluppo e un linguaggio di programmazione open source, derivato da Java. Esso viene incontro a necessità di utilizzo più specifico dal punto di vista grafico. Le molteplici possibilità della libreria combaciano perfettamente con le nostre aspettative di rappresentazione grafica del progetto. Processing viene oggi utilizzato da studenti, artisti, designers, ricercatori e obbisti per produrre prototipi e prodotti finiti di livello professionale. Processing è stato inizialmente sviluppato da Ben Fry e Casey Reas nel 2001 mentre erano entrambi studenti sotto la supervisione di John Maeda al MIT Media Lab. Ulteriore sviluppo di Processing ha avuto luogo all'Interaction Design Institute Ivrea, Carnegie Mellon University, e alla UCLA, dove Reas è capo del Dipartimento di Design — Media Arts. Miami University, Oblong Industries, e la fondazione Rockefeller hanno contribuito finanziando il progetto.

XML: (eXtensible Markup Language). Per rappresentare l'input è necessario un formato che memorizzi solo le informazioni reali, cioè solo l'elenco delle interazioni avvenute in un dato istante. XML è un linguaggio di markup che permette di definire dati strutturati facilmente leggibili dall'uomo e facilmente analizzabili da un computer. L'utilizzo del formato XML per definire l'input del programma di visualizzazione della rete permette di memorizzare in modo efficiente solo le informazioni necessarie, rendendo, tra l'altro, la loro analisi più semplice da un punto di vista algoritmico.

Java linguaggio di programmazione orientato agli oggetti, indipendente dalla piattaforma.

4 Funzioni Implementate

4.1 creazione di una network history

3D network visualizator permettere di creare una simulazione personalizzata, questa funzione è disponibile nel pannello principale del programma, o tramite il menu File -> crea una nuova rete. i passi sono principalmente quattro:

1. Inserire nel campo “Nome della rete” il nome che identificherà il nostro file;
2. inserire nel campo “Nome del nodo” il nome del nodo che farà parte del nostro network nei campi “coordinate:” il valore delle coordinate X,Y,Z.
3. Inserire le varie interazioni tra i nodi tramite i campi:
 - istante (il momento in cui vogliamo che i nodi iniziano l’interazione).
 - campi “nodo sorgente:” e “nodo destinatario:” i nomi dei nodi interessati all’interazione (è possibile far partecipare più nodi a un’interazione).
 - campo “Tempo:” il tempo che i nodi saranno impegnati nell’interazione.
 - campo “Frequenza:” con quale frequenza i nodi si scambieranno messaggi.
4. tramite l’opzione “Usa sfera come struttura” è possibile utilizzare una sfera invece del classico piano.

4.2 importare una rete in formato xml

Tramite il menu File -> “Importa una nuova rete” è possibile caricare una file precedentemente creato tramite il pannello “crea network” o tramite il “generatore di reti random”.

Una volta importato il file, il programma visualizzerà il pannello di processing con la nostra struttura di base (piano o sfera) dove verranno posizionati i vari nodi durante una breve animazione introduttiva. Finita l’animazione è possibile far partire la simulazione tramite il tasto “play” posizionato nella barra dei comandi (Figura 6). In qualsiasi momento possibile fermare la simulazione tramite il tasto “pause” (Figura 6) per analizzare le interazioni di

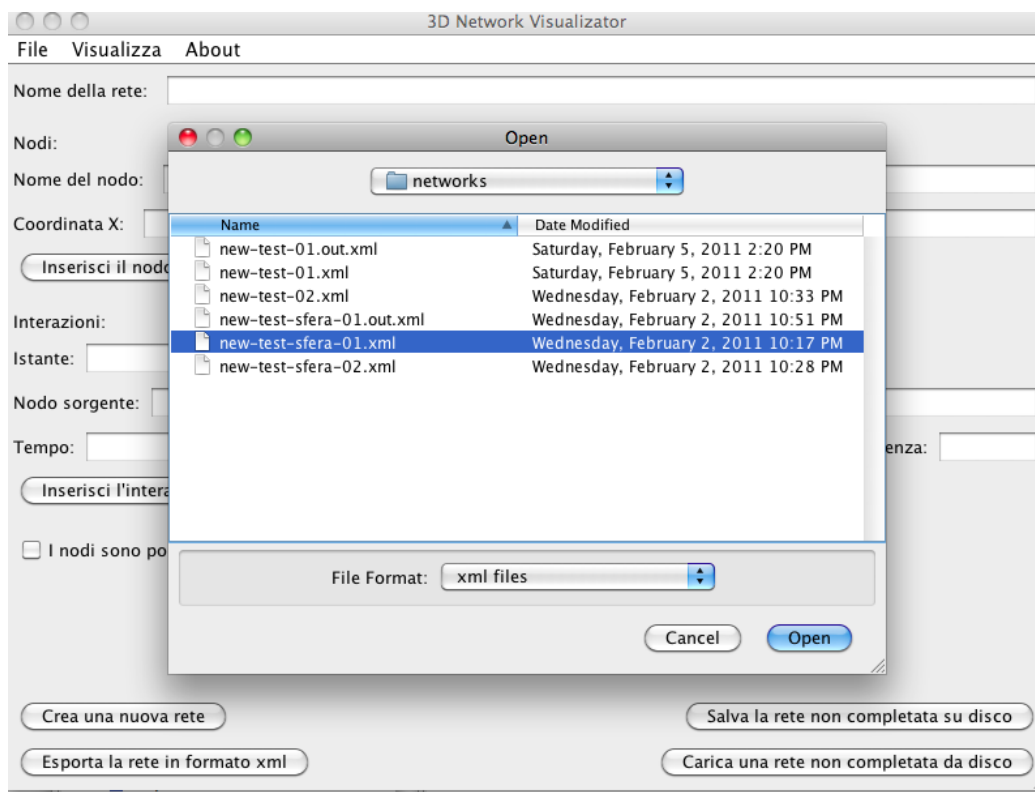


Figura 1: File dialog per importare una nuova rete.

quel momento. Lo slider posto sempre nella barra in basso (Figura 6) mostra l'avanzamento della simulazione in percentuale. Tramite il campo situato a fianco lo slider è possibile cambiare la durata di un istante, questa opzione è molto utile per soffermarsi sulle interazione che avvengono in ogni istante.

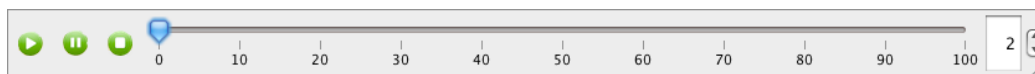


Figura 2: Slider per controllare l'animazione delle interazioni.

4.3 Opzioni di Visualizzazione

Tramite il menu Visualizza è possibile accedere a diverse opzioni di visualizzazioni tra cui: 1. Visualizza spazio 3d di riferimento (attiva di default) tramite questa opzione è possibile nascondere la griglia di riferimento dell'a-

nimazione dove sono posizionati i nodi. Questa opzione è utile per ridurre al minimo gli elementi presenti nel riquadro. 2. Visualizza solo archi entranti (disattivata di default) questa opzione è molto utile se vogliamo visualizzare solo le connessioni e il traffico in ingresso; questa opzione è utilizzabile in accoppiata alla funzione partizione in modo da visualizzare solo le connessioni e il traffico in ingresso di determinati nodi. 3. Visualizza gli archi di tutti i nodi (attiva di default) Questa opzione permette di nascondere in qualsiasi istante gli archi della rete, utile per visualizzare la posizione dei nodi sulla griglia; questa opzione è utilizzabile in accoppiata alla funzione partizione.

4.4 Interazione con una partizione della rete

Durante i primi test della nostra applicazione abbiamo notato che era difficile concentrare l'attenzione su una ristretta cerchia di nodi, appunto una partizione della rete. L'attività della rete cambia ogni istante e la possibilità che le connessioni interessino tutti i nodi non è remota. Uno scenario del genere è sicuramente utile per avere un quadro completo della situazione in un dato istante ma se la nostra analisi è rivolta soltanto a determinati nodi tutte le altre interazioni tra i nodi della rete risultano superflue.

Per ovviare a questo problema è stata implementata la funzione partizione; Questa funzione è disponibile in qualsiasi momento dell'animazione; per utilizzarla basta portarsi con il cursore sul nodo/i che ci interessano e selezionarli premendo il tasto "ctrl".

In questo screenshot possiamo notare come la funzione partizione faciliti l'analisi di questa parte del network formata da 3 nodi nell'istante 6

4.5 Tipologie di struttura

4.5.1 Piano

4.5.2 Sfera

5 Il dataset

Il tipo di dataset che viene visualizzato contiene tutte le informazioni che riguardano i nodi che compongono la rete, e istante per istante, tutte le interazioni che avvengono tra i nodi. Gli attributi che riguardano i nodi sono:

l'identificativo È un identificativo numerico che viene utilizzato all'interno del dataset per indicare uno specifico nodo. Ogni nodo possiede un identificativo univoco.

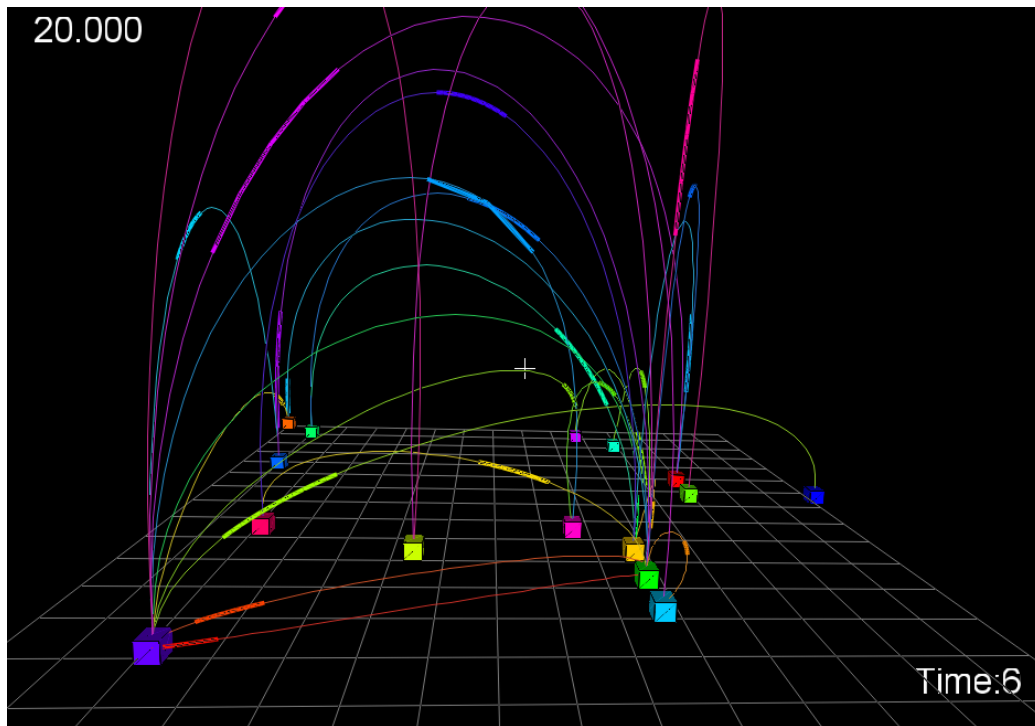


Figura 3: esempio di visualizzazione senza partizione.

l'etichetta L'etichetta svolge circa lo stesso ruolo dell'identificativo ma è più adatta per essere memorizzata dalle persone. gli identificativi servono al software per distinguere e identificare i diversi nodi, per questo motivo sono degli indici numerici. Per una persona è invece molto più facile memorizzare degli identificativi alfanumerici, e questo è lo scopo delle etichette. Inoltre quando i nodi rappresentano delle entità fisiche esistenti, come ad esempio delle città, è utile che le label rispecchino il nome di tale entità al fine di aiutare la comprensione del fenomeno. L'etichetta di un nodo non deve necessariamente essere univoca.

coordinate Sono le coordinate cartesiane dei nodi.

Gli attributi che riguardano le interazioni sono:

la sorgente È l'identificativo del nodo da cui ha origine l'interazione.

l'obiettivo È l'identificativo del nodo obiettivo dell'interazione.

quantità Rappresenta la quantità del flusso che costituisce l'interazione. Cosa sia effettivamente il flusso che costituisce l'interazione dipende dal

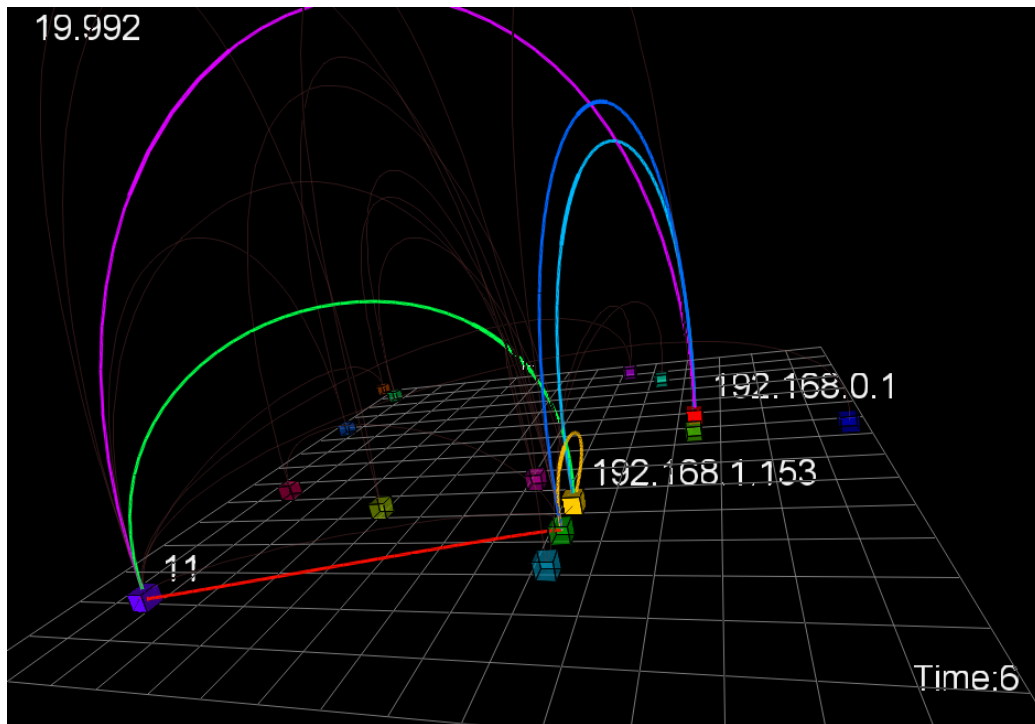


Figura 4: esempio di visualizzazione con partizione.

tipo di rete in esame. Può per esempio trattarsi di pacchetti scambiati da diversi elaboratori collegati via rete, o di automobili che si spostano da un luogo ad un altro.

frequenza È la frequenza che caratterizza l'interazione.

Le informazioni che riguardano le interazioni sono naturalmente suddivise per istanti. Anche gli istanti hanno degli attributi:

valore È l'identificativo dell'istante. Gli identificativi degli istanti devono essere univoci e consecutivi.

etichetta Anche in questo caso le etichette servono per meglio comprendere il fenomeno. Le etichette possono infatti rispecchiare il tempo fisico, l'ora, in cui sono state prese le misurazioni che costituiscono le informazioni riguardanti le interazioni.

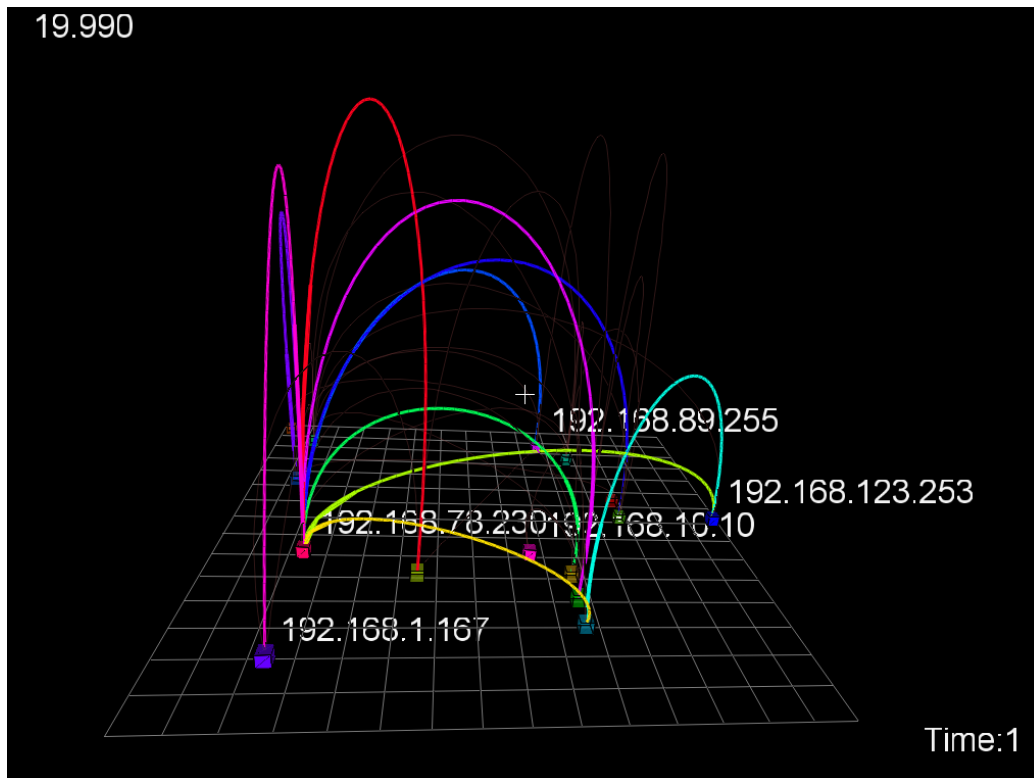


Figura 5: esempio di visualizzazione con partizione.

5.1 Il file network XML

L'input del nostro visualizzatore è un file che contiene la descrizione della topologia di una rete e le interazioni che avvengono tra i suoi nodi. Come formato del file di input abbiamo scelto l'XML per diversi motivi. In primo luogo per la facilità con cui può essere processato da un calcolatore. In secondo luogo per la sua portabilità su diverse piattaforme e linguaggi di programmazione, in questo modo lo stesso file di input può essere utilizzato facilmente anche da altri software scritti in altri linguaggi di programmazione per altre piattaforme software. Infine per la sua caratteristica di essere facilmente leggibile anche dall'uomo. Il principale punto a sfavore di questo formato è la scarsa compattezza, nel senso che oltre alle informazioni relative alla rete il file contiene anche molte meta-informazioni che servono appunto per facilitare l'elaborazione del file e facilitarne la sua leggibilità. Questo significa che per file di input che rappresentano reti di grandi dimensioni, ad esempio con migliaia di nodi e centinaia di interazioni, molto spazio per la memorizzazione del file su disco sarà occupato dalle meta-informazioni. In

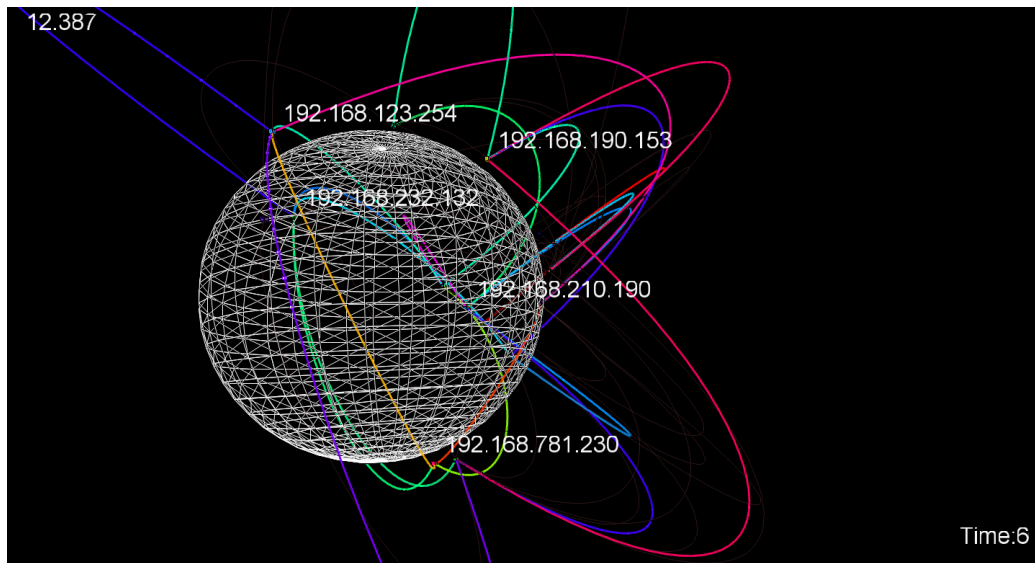


Figura 6: esempio di visualizzazione con partizione.

questi casi un formato binario sarebbe stato sicuramente più compatto, ma avrebbe perso tutte le caratteristiche positive elencate sopra.

Qui di seguito viene riportato un file di input di esempio:

```
<?xml version="1.0" encoding="UTF-8"?>
<network>
  <static-data>
    <network-name>new-test-02</network-name>
    <nodes-list>
      <node id="0" label="192.168.0.1" x="2" y="0" z="6" />
      <node id="1" label="192.168.0.2" x="3" y="0" z="8" />
    </nodes-list>
    <flat>true</flat>
  </static-data>
  <dynamic-data>
    <instant value="0" label="0">
      <interaction source="1" target="0" quantity="1" frequency="0.6080425" />
    </instant>
    <instant value="1" label="1">
      <interaction source="1" target="0" quantity="8" frequency="1.31866568" />
    </instant>
  </dynamic-data>
</network>
```

La struttura del file di input è divisa in due parti principali:

static-data: Questa parte contiene le informazioni statiche della rete, cioè la sua topologia, è composta dai seguenti elementi:

- network-name: il nome della rete
- nodes-list: l'elenco dei nodi presenti nella rete
- flat: indica se la rete deve essere visualizzata in uno spazio piano o sferico

Ogni nodo in nodes-list è descritto con i seguenti attributi:

- id: è l'identificativo univoco del nodo
- label: è l'etichetta del nodo
- x, y e z: sono le coordinate del nodo.

dynamic-data: Questa parte contiene tutte le informazioni dinamiche della rete, cioè le interazioni che avvengono tra i suoi nodi. È strutturata come una lista di istanti. Ogni istante è strutturato come una lista di interazioni. Ogni interazione è descritta con i seguenti attributi:

- source: è l'identificativo del nodo dal quale parte l'interazione
- target: è l'identificativo del nodo obiettivo dell'interazione
- quantity: rappresenta la quantità di dati scambiati durante l'interazione
- frequency: è la frequenza dell'interazione

6 Conclusioni

Lo studio e l'implementazione del programma 3D Network Visualizator ci ha dato la possibilità di scoprire i vantaggi derivati da una efficace ed adatta visualizzazione dei dati. Spesso per effettuare analisi non basta avere statistiche dettagliate e indici ricavati da grandi quantità di dati per il semplice motivo che viene a mancare una visione d'insieme che potrebbe portare un risultato d'analisi inaspettato. Inizialmente il nostro dominio applicativo erano le reti di calcolatori dislocate geograficamente che scambiano pacchetti di dati. Il dominio in questione per quanto può essere banalmente descrivibile per esempio tramite un grafico bidimensionale, esclude alcune informazioni che potrebbero essere importanti in caso di analisi approfondite. Una fra tutte la frequenza con cui i nostri ipotetici calcolatori si scambiano pacchetti;

come rendere visibile questa informazione? L'introduzione della terza dimensione è stata essenziale per poter aggiungere elementi. Prendendo in esempio la frequenza con cui i nodi si scambiano i pacchetti, la nostra soluzione è stata di curvare il link in modo da creare un arco in relazione all'aumento della frequenza di comunicazione. Questo tipo di approccio è risultato di sicuro impatto per avere un'idea dei nodi che hanno un scambio maggiore dei pacchetti nella rete in un dato istante.