**Corso di Reti per l’automazione industriale**

**Elaborato Omnet++**

**Andrea Calabretta 1000008923, Alessandro Mauro 1000009156**

**Relazione finale – Elaborato\_G**

1. **Introduzione**

L’obiettivo è quello di simulare una rete wireless ad hoc IEEE 802.11 in cui dei nodi mobili scambiano periodicamente dei messaggi.

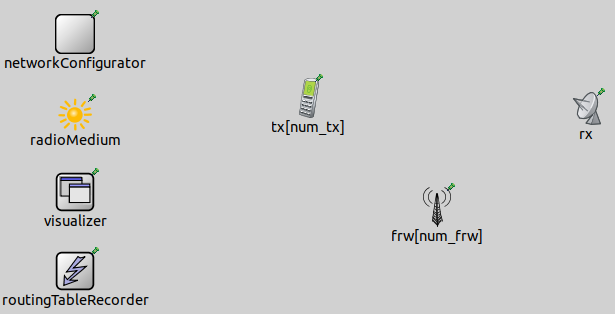


Figura 1 - Network.ned

In tutte le simulazioni sono presenti tre tipologie di nodi: **tx** è un array di [num\_tx] nodi che inviano messaggi, **frw** è un array di [num\_frw] nodi che si occupano di inoltrare i messaggi ricevuti mentre **rx** è un nodo che riceve i messaggi. Il submodule scelto per tutte e tre le tipologie è **adHocHost**.

Inoltre, sono stati utilizzati i seguenti submodules:

* **networkConfigurator** – InetworkConfigurator: si occupa di creare l’interfaccia di rete
* **radioMedium** – IEEE802.11ScalarRadioMedium:
* **visualizer** - IntegratedMultiVisualizer:
* **routingTableRecorder** – RoutingTableRecorder:

È stato valutato l’end to end delay del nodo ricevente in diverse configurazioni di rete; in una prima fase è stata implementata una morfologia di rete piuttosto semplice che presentava solo un nodo emettitore e un ricevitore. Successivamente, si è scelto di complicare lo scenario aumentando il numero di nodi e attivando la mobilità dei trasmettitori.

Metti anche le condizioni del canale

1. **Scenario**

Sono state implementate tre configurazioni di rete:

* Config. **Completa** -> 1 Nodo Ricevitore, 8 Forwarding Node, 11 End Node: ***Figura 2***
* Config. **InLineForwarder** -> 1 Nodo Ricevitore, 4 Forwarding Node, 1 End Node: ***Figura 3***
* Config. **Static Nodes** -> 1 Nodo Ricevitore, 2 Forwarding Node, 17 End Node: ***Figura 4***

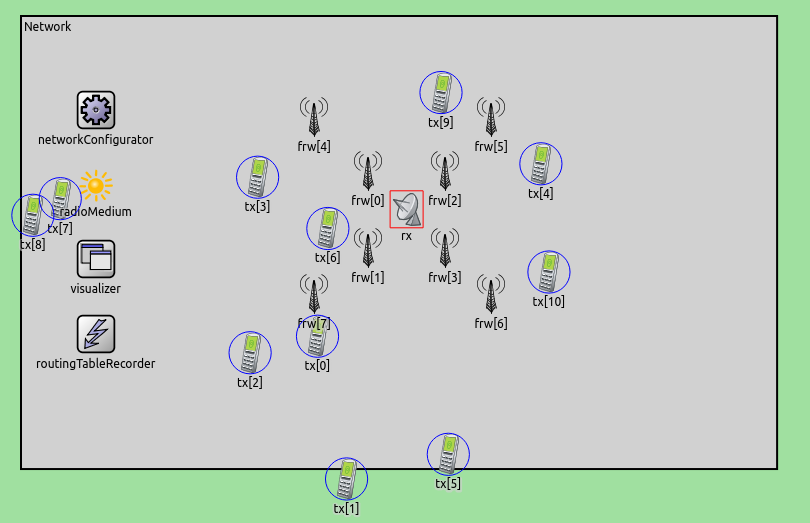


Figura 2 - Configurazione Completa

In questa configurazione di rete, la posizione degli End Node ***tx[\*]*** è generata in maniera casuale utilizzando la tipologia di mobilità *“MassMobility”*; si tratta tuttavia di una generazione pseudocasuale quindi, mandando più volte in esecuzione la simulazione, ciascun nodo occuperà la stessa posizione della simulazione precedente. (?)

Il Ricevitore ***rx*** e i Forwarding Node ***frw[\*]*** sono statici e la tipologia di mobilità adottata è *“StationaryMobility”.*

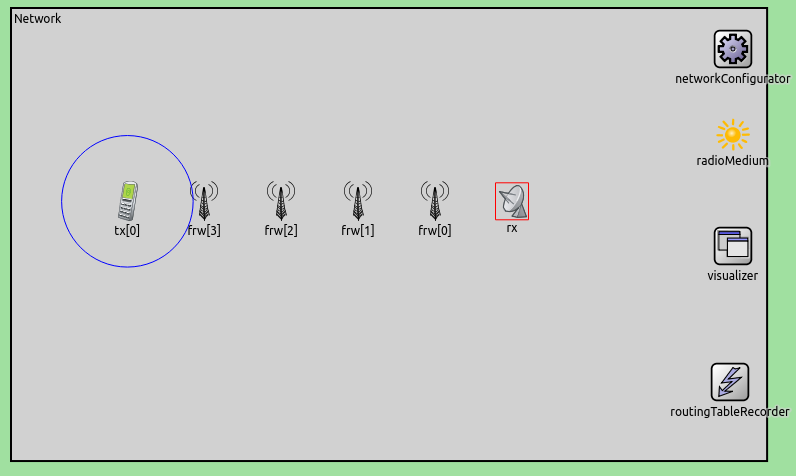


Figura 3 – InLineForwarder

Nella configurazione *“InLineForwarder”* la tipologia di mobilità scelta per il Ricevitore ***rx***e l’End Node ***tx*** è *“LinearMobility”*: i nodi infatti si muovono in maniera lineare lungo una retta parallela all’asse X allontanandosi dai Forwarding Node fino a raggiungere il confine della Network per poi riavvicinarsi.

I Forwarding Node sono invece statici (*“StationaryMobility”*).

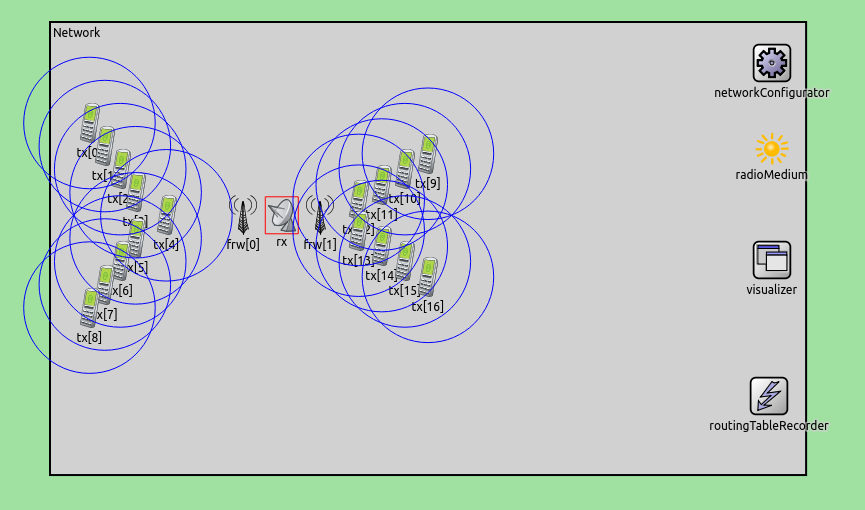


Figura 4 - Static Nodes

Infine, la configurazione *“StaticNodes”* prevede l’uso esclusivo di nodi statici (*“StationaryMobility”*).

1. **Scelte implementative**
2. **Statistiche**

Il dato raccolto è l’**end-to-end** **delay** ovvero il tempo impiegato da un pacchetto per essere trasmesso attraverso la rete dal nodo sorgente alla destinazione.

Per ciascuna configurazione di rete è stato raccolto l’end-to-end delay evidenziando i valori massimo, minimo e medio, valutato in due differenti condizioni di canale: una prevede un fattore di attenuazione α=3 e una varianza σ=7 mentre l’altra, fornita dal docente, prevede α=4.03 e σ=4.98.   
Questa seconda condizione si è rivelata la peggiore poiché quasi la totalità dei pacchetti inviati dagli End Node non veniva mai ricevuta a causa proprio delle pessime condizioni del canale di trasmissione.

1. **Risultati**

Configurazione Completa

La configurazione Completa eseguita per 1 secondo ha generato il seguente grafico relativo all’end-to-end delay:

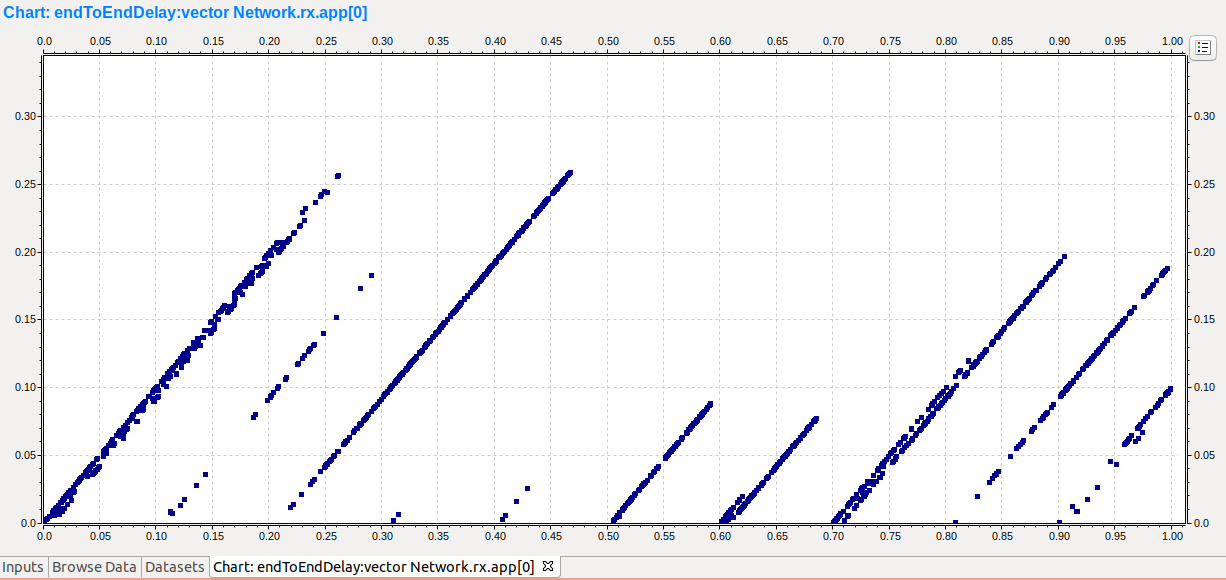


Figura 5 - End-to-end delay Configurazione Completa

Dal grafico si evince che i primi pacchetti che finiscono nelle code dei Forwarding Nodes hanno bassi valori di end-to-end delay. Via via che queste code iniziano a popolarsi di pacchetti, l’end-to-end delay aumenta linearmente.

In particolare, il grafico presenta sull’asse X lo scorrere del tempo di simulazione; ogni 10 centesimi di secondo gli End Node generano e inviano i pacchetti.

Nella figura successiva sono riportati i valori di end-to-end delay tra i quali quelli di nostro interesse: **massimo, minimo e medio**.

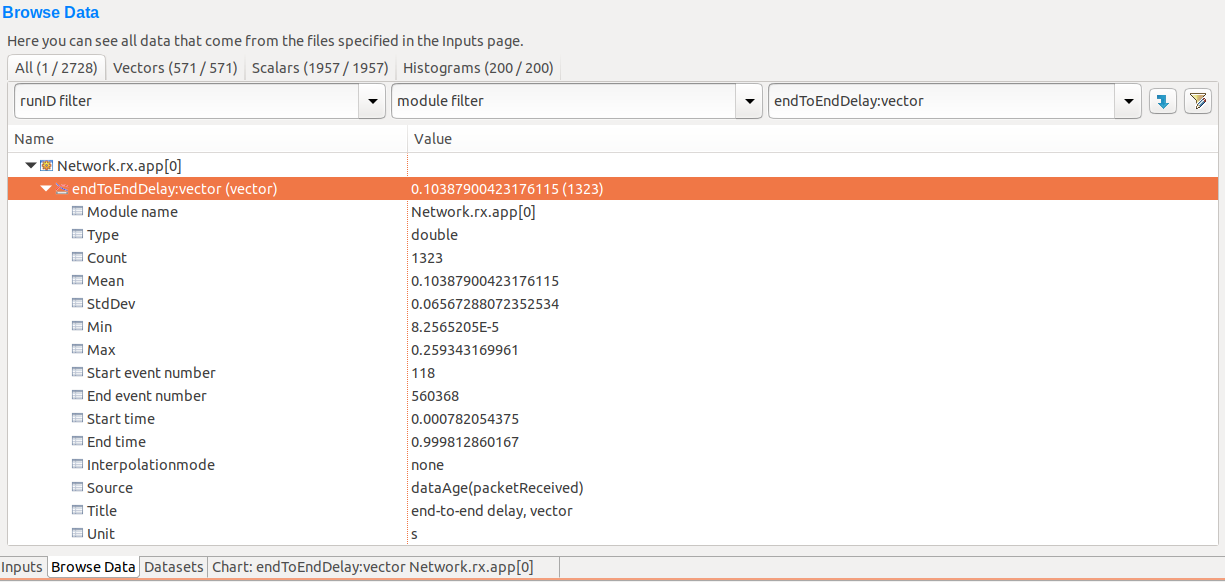


Figura 6 – Statistiche

*Quello che ci aspettavamo era che il grafico di questo end to end delay fosse esponenziale in base a quanto studiato riguardo la teoria delle code. Il motivo per cui il grafico è lineare piuttosto che esponenziale potrebbe essere riconducibile al fatto che le code dei forwarder non sono abbastanza lunghe da determinare un aumento del ritardo di tipo esponenziale.*

Cambiando le condizioni di canale, impostando un fattore di attenuazione **α** = 4.03 e una varianza **σ** = 4.98, il numero di pacchetti ricevuti da ***rx*** è pari a 1 in 10 secondi di simulazione.

Poiché quasi la totalità dei pacchetti è perduta a causa delle pessime condizioni di canale, il grafico dell’end-to-end delay in questa nuova configurazione, mostrato in ***Figura 7***, risulta privo di informazioni.

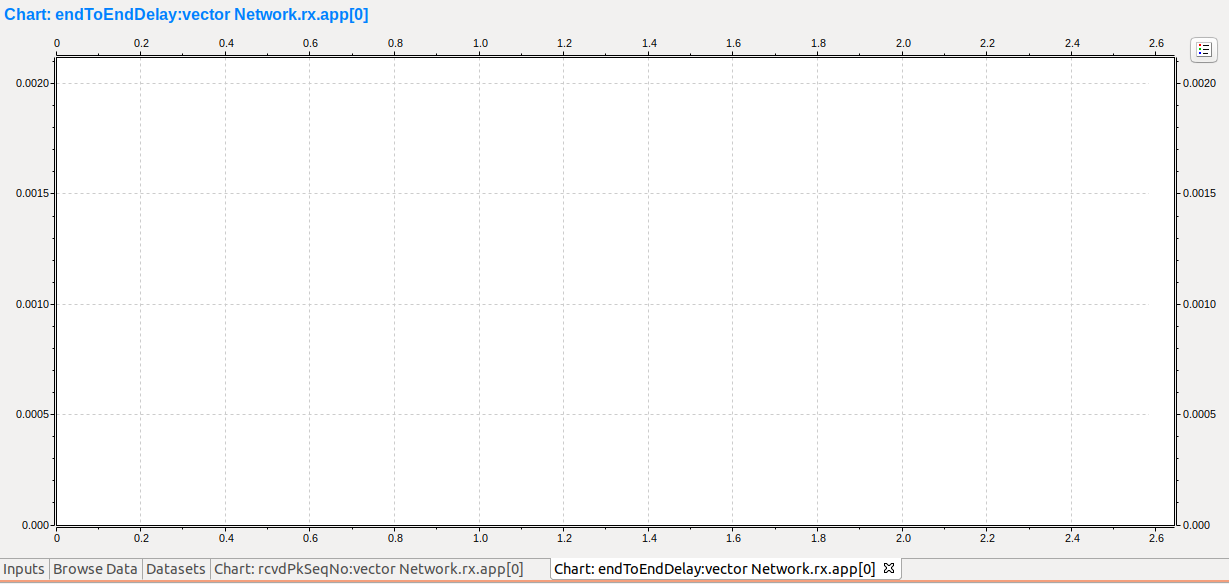


Figura 7 - End-to-end delay Config. Completa α= 4.03, σ= 4.98

Configurazione InLineForwarder

In questa configurazione, con lo scorrere del tempo, il mittente e il ricevitore si allontanano dai Forwarding Node.   
Dopo essersi allontanati dal range di copertura dei Forwarding Node, il mittente e il ricevitore ritornano al suo interno e la ricezione dei pacchetti avviene soltanto dopo 11 secondi di simulazione.

Alla fine della simulazione della durata di 15 secondi, il ricevitore ha ricevuto 14 pacchetti e il grafico relativo all’end- to-end delay è il seguente:

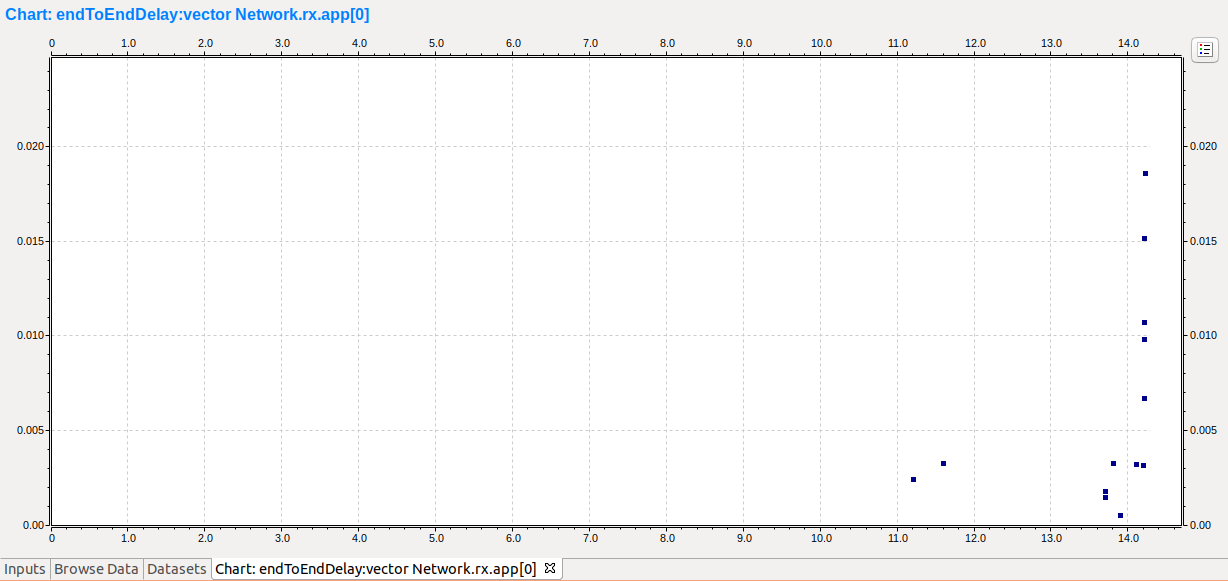


Figura 8 - End-to-end delay InLineForwarder

Inizialmente, a causa della mancanza di copertura tra i nodi, il grafico mostra l’assenza di end-to-end delay fino a 11 secondi poiché fino a quel momento i pacchetti vengono perduti; successivamente, quando mittente e ricevitore si avvicinano ai Forwarding Node, il ritardo di propagazione misurato aumenta esponenzialmente in corrispondenza del burst di pacchetti trasmessi (***Figura 9***).

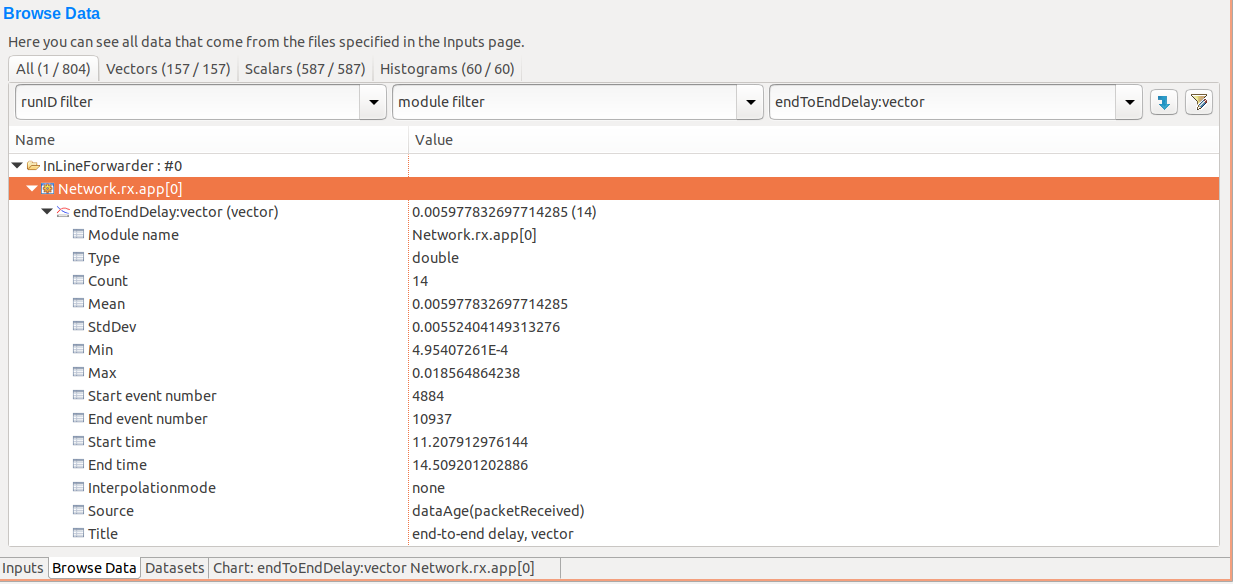


Figura 9 - Statistiche InLineForwarder

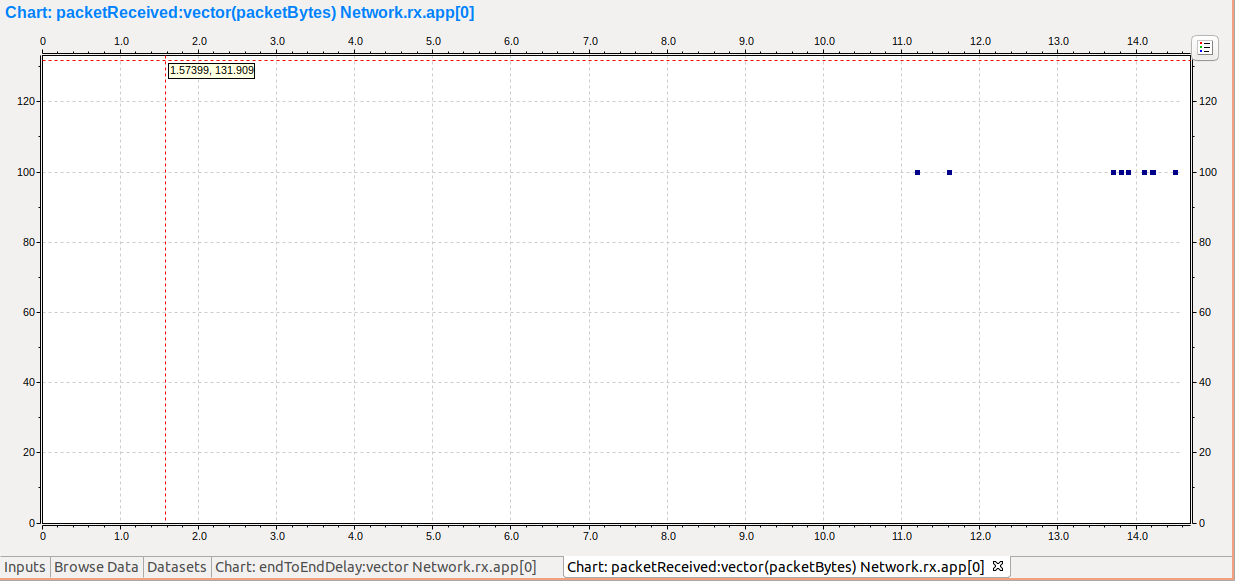


Figura 10 - Istanti di ricezione dei pacchetti

Impostando le condizioni di canale ai valori α=4.03 e σ=4.98 e in base alle posizioni statiche dei Forwarding Node, il risultato è analogo a quello ottenuto con la configurazione “*Completa*” poiché il numero di pacchetti ricevuti è nullo e quindi non ha senso parlare di end-to-end delay.

Configurazione StaticNodes

Infine, il grafico relativo all’end-to-end delay di questa configurazione presenta un andamento lineare, come mostrato in ***Figura 11***.

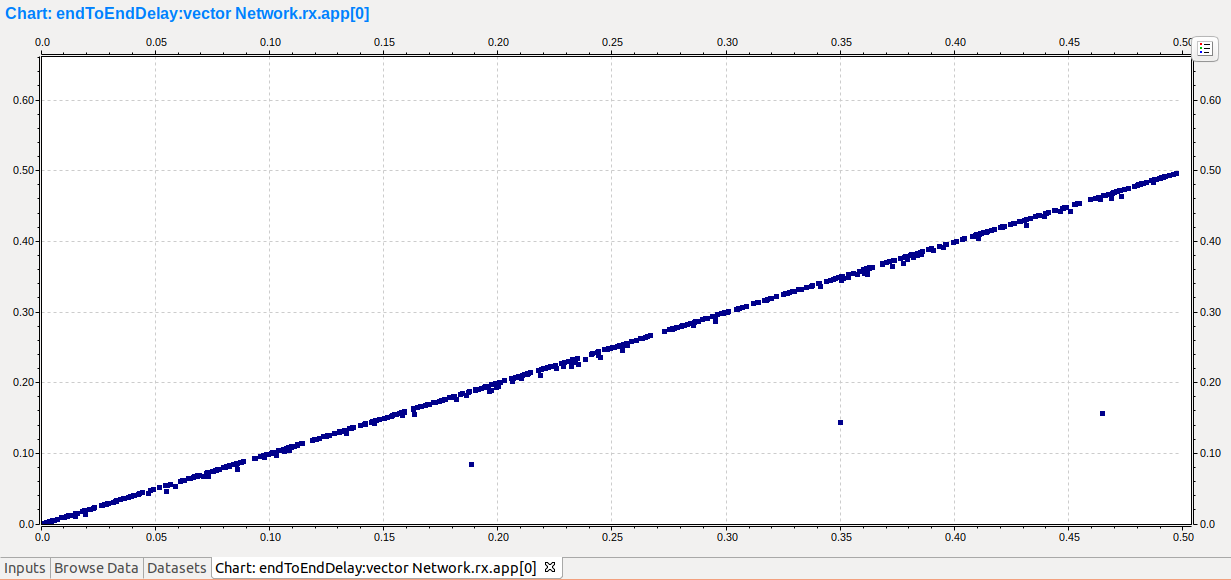


Figura 11 - End-to-end delay StaticNodes

Dal grafico si evince che col passare del tempo, in una simulazione della durata di 500 millisecondi, aumenta il valore di end-to-end delay misurato.

Di seguito sono elencati tra gli altri, i valori di massimo, minimo e medio end-to-end delay.

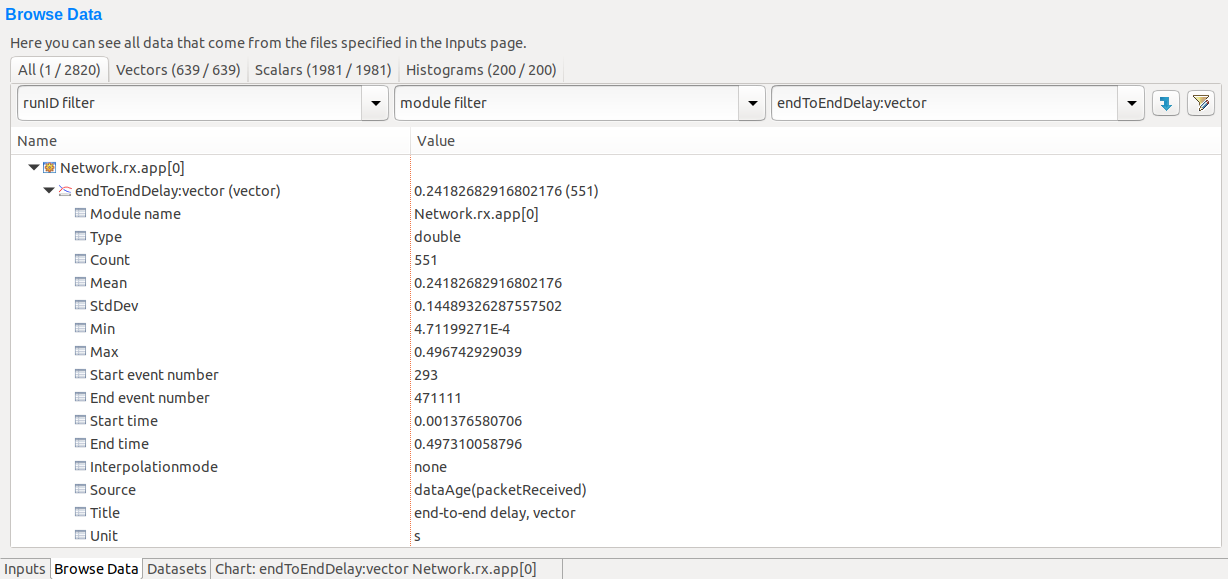


Figura 12 – Statistiche

Peggiorando le condizioni di canale, anche in questa configurazione, il Nodo Ricevitore non riceve alcun pacchetto quindi non ha senso parlare di end-to-end delay.

[Figura 1 - Network.ned 1](#_Toc41233647)

[Figura 2 - Configurazione Completa 2](#_Toc41233648)

[Figura 3 – InLineForwarder 3](#_Toc41233649)

[Figura 4 - Static Nodes 3](#_Toc41233650)

[Figura 5 - End-to-end delay Configurazione Completa 4](#_Toc41233651)

[Figura 6 – Statistiche 5](#_Toc41233652)

[Figura 7 - End-to-end delay Config. Completa α= 4.03, σ= 4.98 6](#_Toc41233653)

[Figura 8 - End-to-end delay InLineForwarder 6](#_Toc41233654)

[Figura 9 - Statistiche InLineForwarder 7](#_Toc41233655)

[Figura 10 - Istanti di ricezione dei pacchetti 7](#_Toc41233656)

[Figura 11 - End-to-end delay StaticNodes 8](#_Toc41233657)

Andrea Calabretta

Alessandro Mauro