

Università degli Studi di Catania



Dipartimento di Matematica e Informatica

Corso di Laurea in Informatica Magistrale

PROGETTO P2P

**Simulazione di uno scenario di Disaster
Recovery su NS-3**

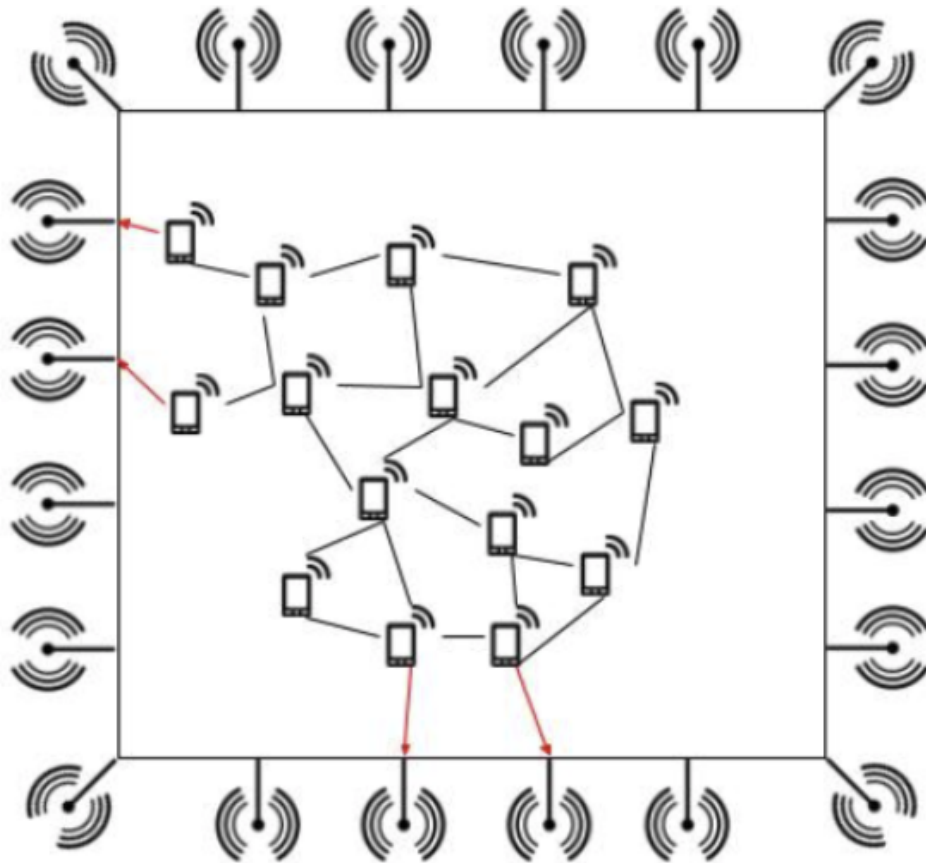
Studente: Mangano Fabio

Corso: Peer to Peer and Wireless Networks

Matricola: 1000023383

Data: 23 / 09 / 2021

Descrizione del progetto



Sia dato uno scenario di disastro naturale, nel quale l'infrastruttura fissa non è totalmente disponibile.

In particolare, nella zona coinvolta non vi è alcun nodo fisso, mentre attorno sono disponibili alcuni punti di accesso ad Internet, tramite AP. Gli Ap sono connessi tra loro tramite rete cablata.

Utilizzare il DSDV per trovare e tenere aggiornato un percorso in uscita verso uno dei nodi fissi a partire dai nodi interni (mobili).

- Lo scenario è di $5 \times 5 \text{ km}^2$
- I nodi nella zona del disastro hanno una scarsa mobilità (movimenti randomici in un raggio di 50 m).

- I nodi fissi sono tutti connessi tra loro e sono disposti ai bordi della zona interessata dal disastro, spazati tra loro di 1 km
- I nodi possono scoprire i propri vicini con appositi messaggi di “hello”
- I nodi mobili hanno cicli di On-Off

L'output richiesto è il grado di connettività del sistema, ossia si vuole valutare:

- La percentuale di nodi raggiungibili dalla rete di nodi mobili
- Gli intervalli di connessione / disconnessione di ciascun AP

DSDV

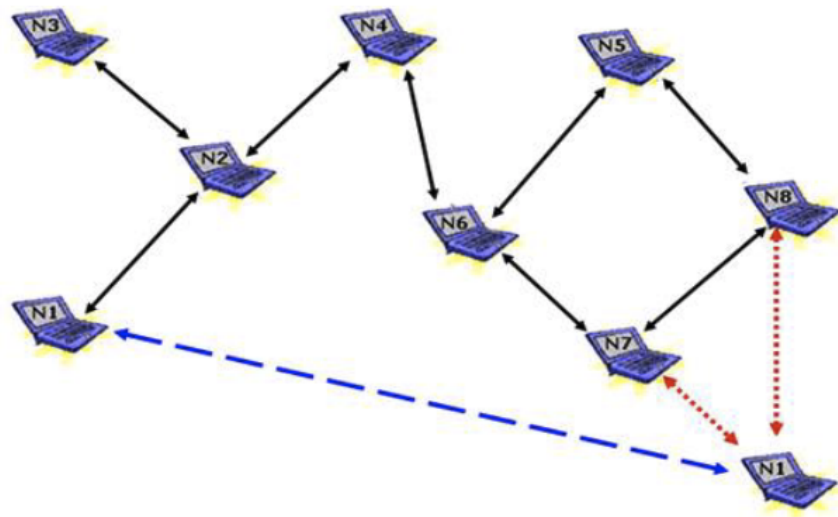


Figure 1. A network illustrating the movement of node N1

Descrizione del protocollo

DSDV (Destination Sequenced Distance Vector) è un protocollo di routing vettoriale hop-by-hop, basato sul algoritmo di routing di Bellman-Ford.

Ciascun nodo della rete mantiene una tabella di routing, contenente una entry per ciascuna delle destinazioni nella rete e il numero di hop necessari per raggiungere ciascuna di essa. Ad ogni voce è associato un numero di sequenza che aiuta a identificare le voci obsolete. Questo meccanismo consente al protocollo di evitare la formazione di loop di routing.

Ogni nodo invia periodicamente aggiornamenti in tutta la rete con un numero di sequenza pari che aumenta in modo monotono, per

rispecchiare i cambiamenti della rete.

Ciascun aggiornamento contiene l'indirizzo della destinazione, il numero di salti per raggiungere la destinazione, il numero di sequenza delle informazioni ricevute relative alla destinazione, nonché un nuovo numero di sequenza univoco per la trasmissione.

Viene sempre utilizzato il percorso etichettato con il numero di sequenza più recente.

Quando i vicini del nodo trasmittente ricevono questo aggiornamento, riconoscono di essere a un salto dal nodo sorgente e includono queste informazioni nei loro vettori di distanza.

Ogni nodo memorizza il "prossimo percorso di instradamento" per ogni destinazione raggiungibile nella propria tabella di instradamento. Il percorso utilizzato è quello con il numero di sequenza più alto, ovvero quello più recente.

Quando un vicino B di A scopre che A non è più raggiungibile, annuncia il percorso verso A con una metrica infinita e un numero di sequenza maggiore di uno rispetto all'ultimo numero di sequenza per il percorso costringendo tutti i nodi con B sul percorso verso A, a reimpostare le loro tabelle di routing.

Tipologie di aggiornamenti

Table 1

The Routing table at Node 4

Destination	Metric	Sequence number
N1	2	S406-N2
N2	1	S128-N2
N3	2	S564-N2
N4	0	S710-N6
N5	2	S392-N6
N6	1	S076-N6
N7	2	S128-N6
N8	3	S050-N6

Table 2

The Routing table updated for Node 4

Destination	Metric	Sequence number
N1	3	S516-N6
N2	1	S238-N2
N3	2	S674-N2
N4	0	S820-N4
N5	2	S502-N6
N6	1	S186-N6
N7	2	S238-N6
N8	3	S160-N6

Gli aggiornamenti delle tabella di routing in DSDV sono distribuiti tramite due diversi tipi di pacchetti di aggiornamento:

- *Dump completo*: questo tipo di aggiornamento contiene tutte le informazioni di routing disponibili su un nodo. Di conseguenza, può richiedere il trasferimento di più Network Protocol Data Unit se la tabella di routing è grande. I pacchetti di dump completi vengono trasmessi di rado se il nodo subisce solo movimenti occasionali.
- *Incrementale*: questo tipo di aggiornamento contiene solo le informazioni che sono cambiate dall'ultimo dump completo inviato dal nodo. Pertanto, i pacchetti incrementali consumano solo una frazione delle risorse di rete rispetto a un dump completo

NS-3

NS-3 è un simulatore open-source basato su eventi, progettato appositamente per la ricerca nelle reti di comunicazione informatica.

Lo sviluppo di NS-3 è iniziato come revisione di NS-2 guadagnando via via l'attenzione del mondo accademico e industriale a causa della fama ottenuta dal suo predecessore.

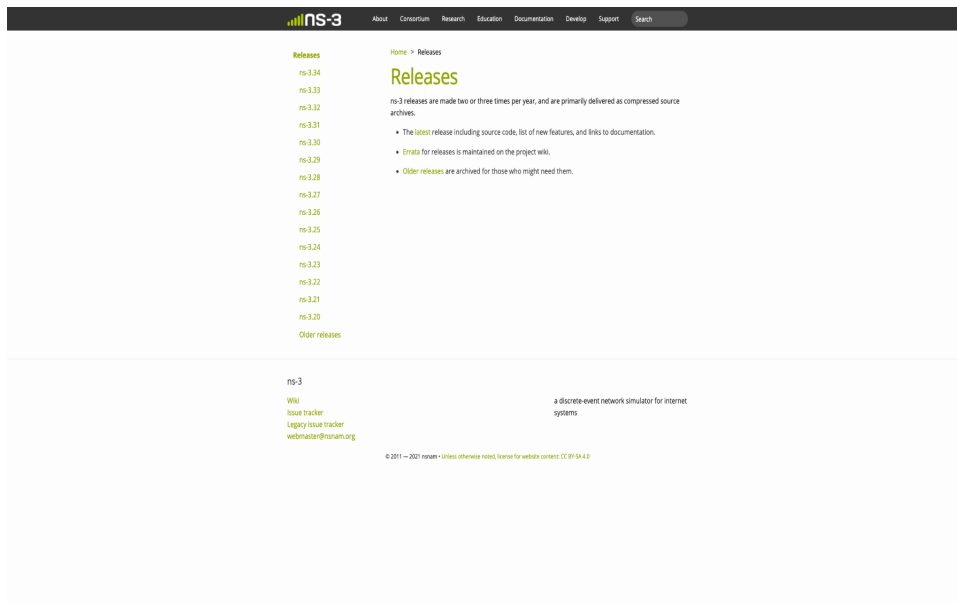
Dopo anni di sviluppo NS-3 contiene numerosi moduli per varie componenti di rete quali routing, protocolli layer-layer, applicazioni, ecc.

La grande novità che è stata introdotta con NS-3 è che i ricercatori possono utilizzare uno dei linguaggi più famosi come il C++ che prende il posto del vecchio linguaggio OTcl.

Senza dubbio NS-3 è diventato il simulatore di rete open source più utilizzato.

Setup ambiente ed esecuzione script

1. scaricare ns3 ed eseguire la procedura di installazione [link](#)



2. scaricare ed installare gnuplot al [link](#)

gnuplot download

Current gnuplot major version is 5.4

- [Primary download site on SourceForge](#)
- [Cygwin binaries](#)
- [Release Notes](#)

The most recent release was 5.4.2 (June 2021)

Downloads offered by others

Development version:

- Windows binaries built by Tamasz Matuszka ([cygnus](#)) and [Mik/W](#) and
- Crossbuild Macintosh and Linux (dsh) binaries may be available at [ns3.dynup.com](#)

Various Sources and Binaries (miscellaneous operating systems):

- [Cygwin](#) mirrors: [gnuplot.at](#)
- Linux rpm (devel) packages: see servers of your favourite distributor.
- 3rd party OS/2 binaries: [gnuplot.sourceforge.net](#)

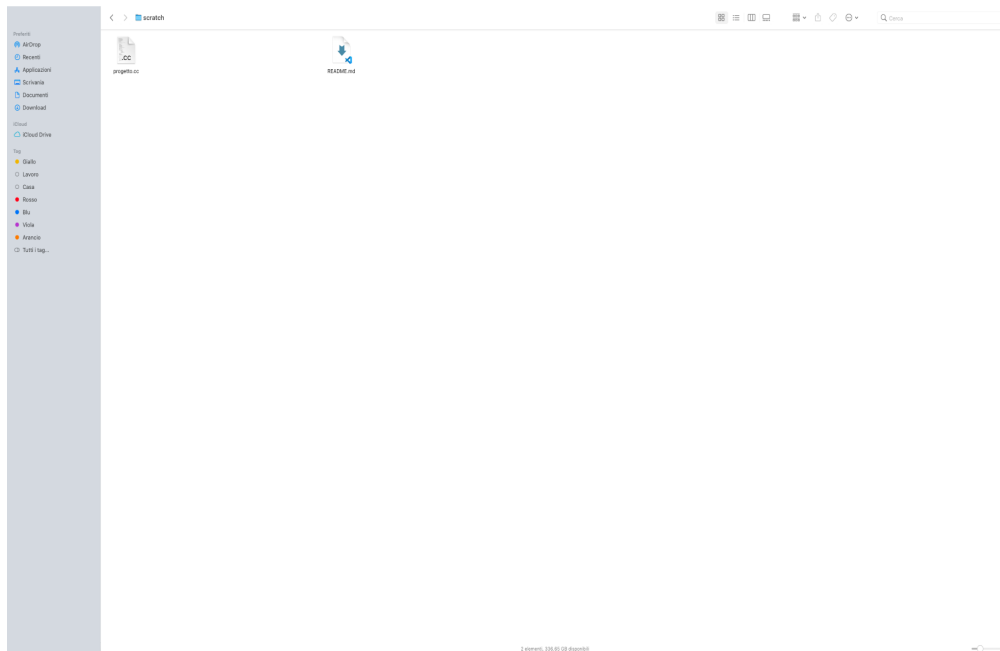
Version 5.0.5-2: [gnuplot.sourceforge.net](#)

Version 5.0.8: [gnuplot.sourceforge.net](#)

[Go back](#) to gnuplot homepage.

Version: June 2021

3. copiare il file “progetto.cc” in *ns3-allinone/ns3-dev/scratch*



4. eseguire lo script digitando i seguenti comandi

```
cd ./ns-3-dev
./waf --run scratch/"progetto.cc --nAPs=20 --nWifis=200
--totalTime=60"
```

5. Visualizzare l'output dello script in console:

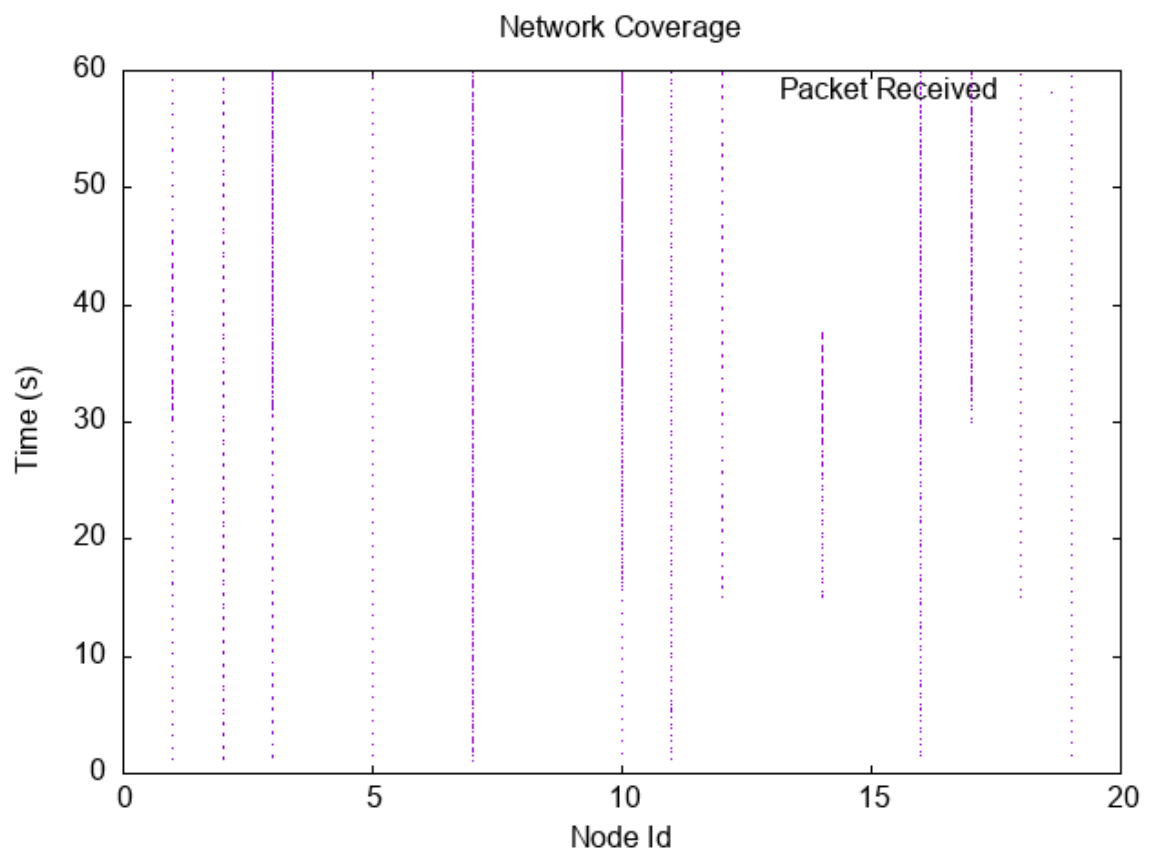
```
ns-3-dev git(master) # ./waf --run scratch/progetto.cc --nAPs=20 --nWifis=200 --totalTime=60
waf: Entering directory '/Users/fabianmango/Desktop/ns-3-allinone/ns-3-dev/build'
waf: Leaving directory '/Users/fabianmango/Desktop/ns-3-allinone/ns-3-dev/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.988s)
Creating 20 mobile wifi nodes...
Creating 20 APs...
Starting simulation for 60 s ...
At 1.443222s AP with id 5 has received a packet from the internal wifi network
At 1.443185s AP with id 9 has received a packet from the internal wifi network
At 1.626189s AP with id 5 has received a packet from the internal wifi network
At 1.624149s AP with id 5 has received a packet from the internal wifi network
At 1.883349s AP with id 9 has received a packet from the internal wifi network
At 2.433469s AP with id 5 has received a packet from the internal wifi network
At 2.595152s AP with id 9 has received a packet from the internal wifi network
At 2.603189s AP with id 5 has received a packet from the internal wifi network
At 2.818876s AP with id 5 has received a packet from the internal wifi network
At 2.879336s AP with id 9 has received a packet from the internal wifi network
At 3.433469s AP with id 5 has received a packet from the internal wifi network
At 3.595152s AP with id 9 has received a packet from the internal wifi network
At 3.623189s AP with id 5 has received a packet from the internal wifi network
At 3.818876s AP with id 5 has received a packet from the internal wifi network
At 3.879336s AP with id 9 has received a packet from the internal wifi network
At 4.433469s AP with id 5 has received a packet from the internal wifi network
At 4.595152s AP with id 9 has received a packet from the internal wifi network
At 4.626189s AP with id 5 has received a packet from the internal wifi network
At 4.818876s AP with id 5 has received a packet from the internal wifi network
At 4.879336s AP with id 9 has received a packet from the internal wifi network
At 5.261182s AP with id 18 has received a packet from the internal wifi network
At 5.338889s AP with id 18 has received a packet from the internal wifi network
At 5.389889s AP with id 18 has received a packet from the internal wifi network
```

6. Lanciare il seguente comando per generare il grafico 2-D:

```
ns-3-dev git:(master) x gnuplot disaster-recovery-simulation.plt
ns-3-dev git:(master) x
```

7. Visualizzare il file *disaster-recovery-simulation.png* generato dalla simulazione

8.



Requisiti simulazione

Area di simulazione

Area: 5x5 km

Access Points:

Numero: min=1, max=20

Distanza tra APs= 1km

Posizione: fissa, lungo il bordo dell'area

Comunicazione tra APs: point to point

Protocollo wifi: 802.11b

Wifi mobili:

Numero: min=1, max=inf.

Posizione: random

Cicli on/off: si, con durata random

Velocità nodo: 2 m/s

Raggio spostamento: 50m

Protocollo wifi: 802.11b

Protocollo di routing: DSDV

Input:

wifiModes: number, (default=50)

apNodes: number, (default=20)

simulationTime: seconds, (default=100s)

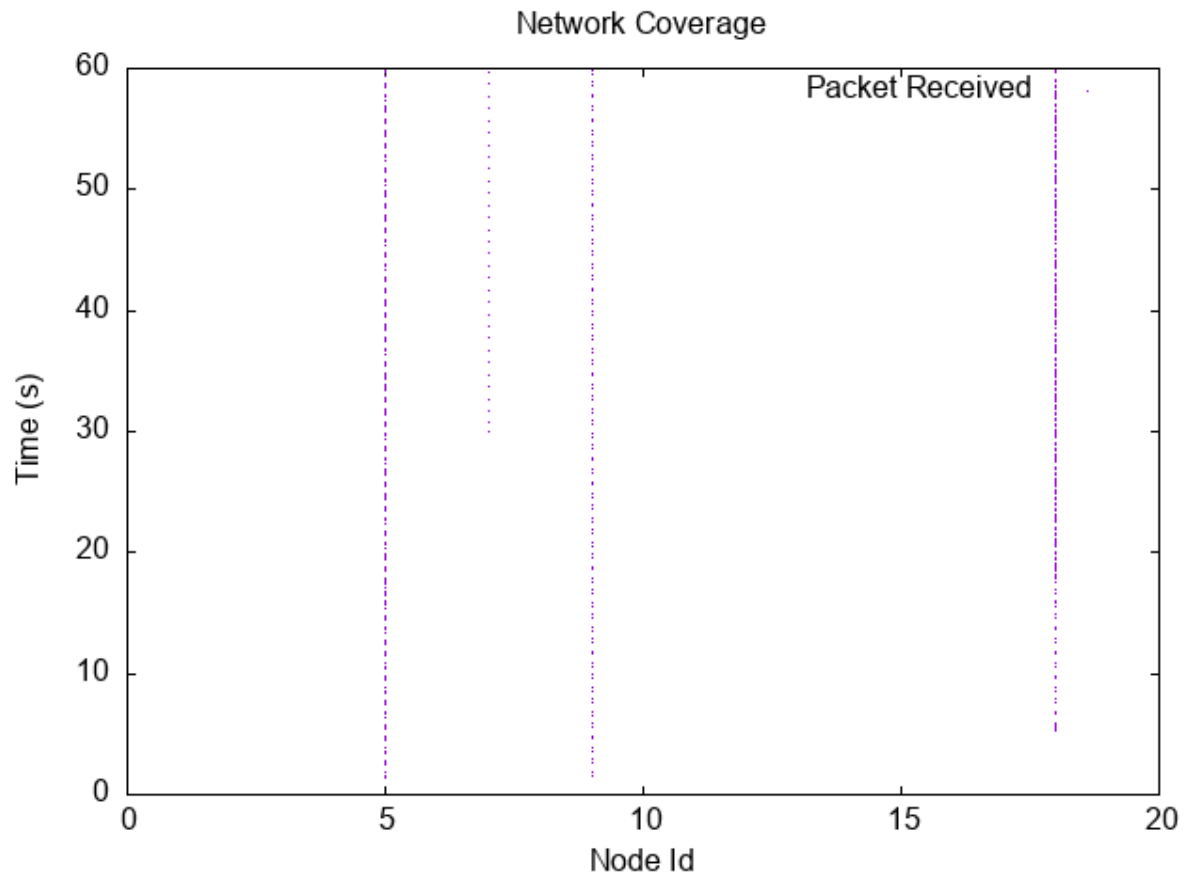
Output script:

numero di ap raggiungibili

copertura rete (rapporto tra ap raggiungibili e ap totali)

plot 2d

Simulazione Scenario (1)



Input

nWifis = 50

nAp = 20

time = 60s

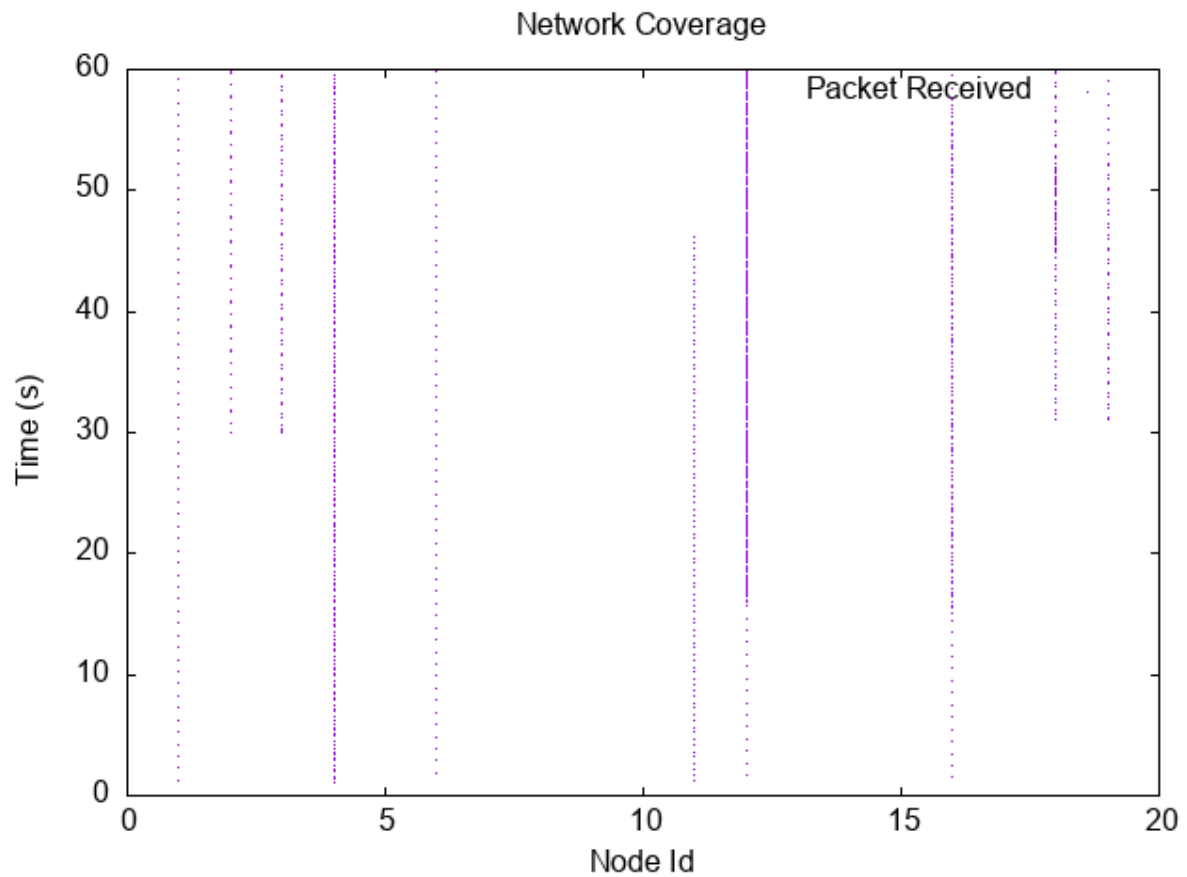
Output

Number of reachable APs: 4 of 20

Reachable AP ids: 5, 7, 9, 18, 19

Network coverage: 25%

Simulazione Scenario (2)



Input

nWifis = 100

nAp = 20

time = 60s

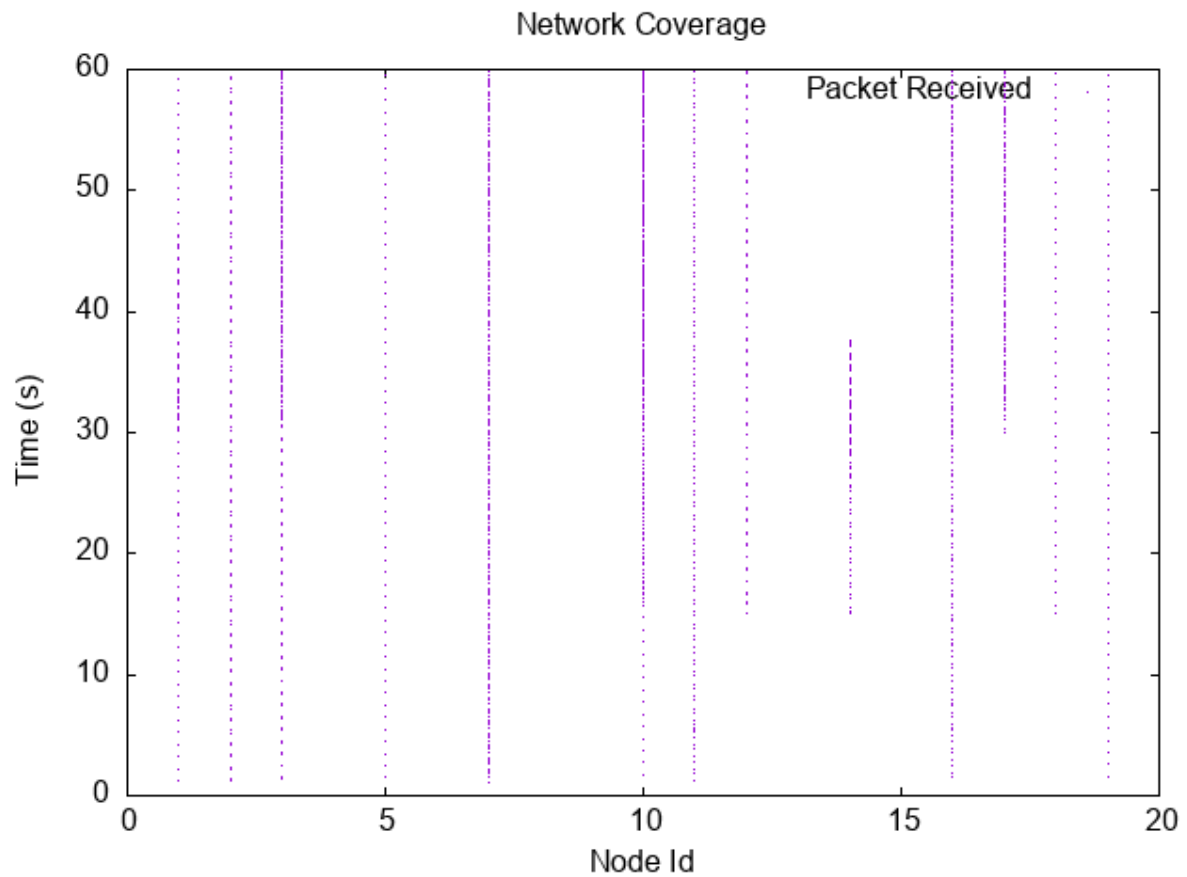
Output

Number of reachable APs: 11 of 20

Reachable AP ids: 0, 1, 2, 3, 4, 6, 11, 12, 16, 18, 19

Network coverage: 55%

Simulazione Scenario (3)



Input

nWifis = 200

nAp = 20

time = 60s

Output

Number of reachable APs: 14 of 20

Reachable AP ids: 0, 1, 2, 3, 5, 7, 10, 11, 12, 14, 16, 17, 18, 19

Network coverage: 70%

Risultati simulazione

Per la simulazione dello scenario richiesto, si è preferito garantire la massima copertura periferica possibile, installando il numero massimo di APs consentito dai vincoli dello stesso.

Per quanto riguarda il numero di nodi wifi mobili, si è considerato un fattore moltiplicativo 2x (50, 100, 200), per rilevare con maggiore facilità la variazione in termine di network coverage.

In particolare:

- la simulazione del *primo scenario* (50 nodi wifi), ha evidenziato un numero di nodi Ap raggiungibili pari a 5 su 20, per una network coverage pari al 25%. Si è evidenziato inoltre per i nodi con id 7 e 19 un periodo di disconnessione (0 pacchetti ricevuti dalla rete), pari o maggiore del 50% al tempo totale della simulazione.

- la simulazione del *primo scenario* (100 nodi wifi), ha evidenziato un numero di nodi Ap raggiungibili pari a 11 su 20, per una network coverage pari al 55%. Si è evidenziato inoltre per i nodi con id 0, 2, 3, 7, 8, 9, 10, 13, 14, 15 un periodo di disconnessione (0 pacchetti ricevuti dalla rete), pari o maggiore al 50% del tempo totale della simulazione.

- la simulazione del *primo scenario* (200 nodi wifi), ha evidenziato un numero di nodi Ap raggiungibili pari a 14 su 20, per una network coverage pari al 70%. Si è evidenziato inoltre per i nodi con id 4, 6, 8, 9, 13, 15, 19 un periodo di disconnessione (0 pacchetti ricevuti dalla rete), pari o maggiore al 50% del tempo totale della simulazione.