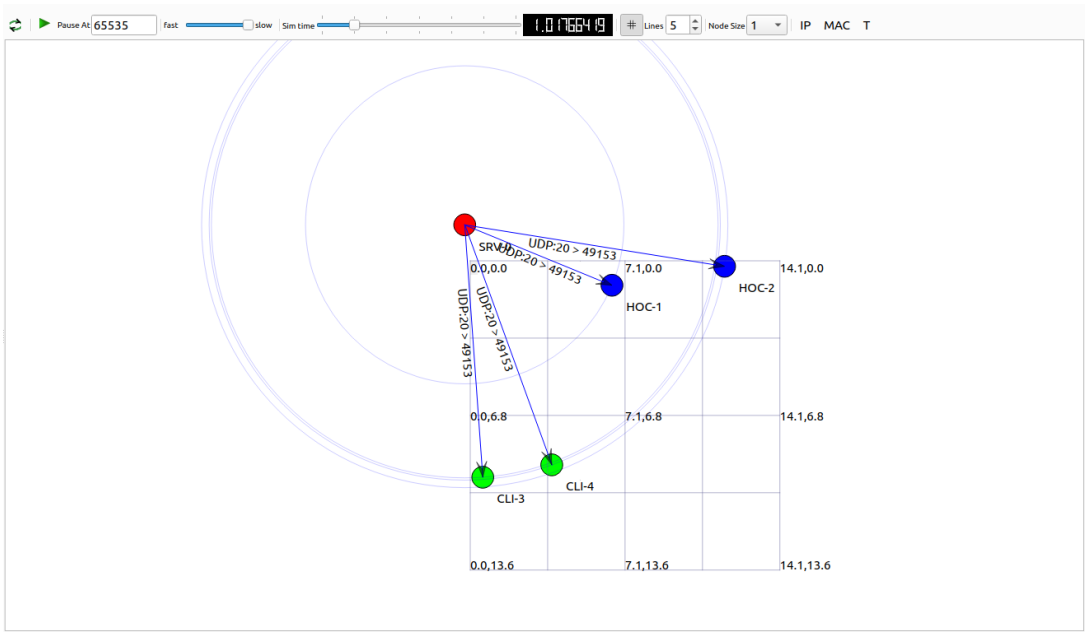
REPORT HOMEWORK

Task 1 : *Wireless Local Area Network – Ad-hoc Mode*

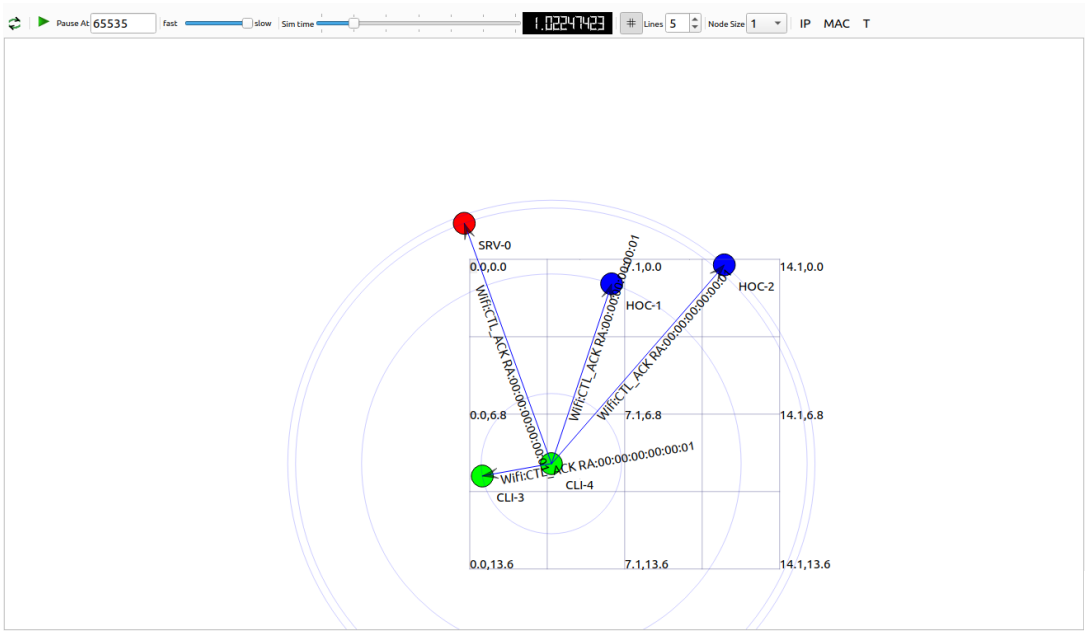
1) Tutti i frame ricevono l'acknoledgement? Spiegare perché.

Tutti i frame che non sono frame di controllo del protocollo 802.11, cioè Wi-Fi, ricevono l'Acknowledgment. In particolare, le ARP replies e la trasmissione dei pacchetti UDP ricevono Acknowledgment come risposta, mentre le ARP requests e le frames di controllo no. Per frames di controllo si intendono gli Acknowledgment stessi e, qualora venissero attivati con l'apposita variabile nel codice, i pacchetti RTS (Request-to-Send) e CTS (Clear-to-Send).



[Fig.1]

[cattura della risposta dell'Echo Server situato sul nodo 0 al primo pacchetto inviato dall'Echo Client, posto sul nodo 4, al tempo t=1s]



[Fig.2]

[cattura dell'Acknoledgement alla frame sopracitata]

Gruppo di lavoro: #34
Componenti: Alessio Maiola (GL), Alessandro Maone, Andrea De Carlo, Davide Caronna

2) Vi sono delle collisioni nella rete? Spiegare perché. Come sei arrivato a questa conclusione?

In questa rete non vi sono collisioni, come abbiamo denotato andando a leggere la flag "Retry" all'interno del Frame Control Field dell'header del protocollo 802.11. L'unico momento in cui rischia di esserci una collisione è a t=2s all'interno della simulazione, tuttavia, poiché uno dei client è alla prima trasmissione, effettua prima un'ARP request, con una frame di dimensione inferiore, riducendo il rischio di collisioni. Abbiamo anche testato che, aumentando il numero di pacchetti trasmessi a uno dei due client, in modo che entrambi trasmettessero a t=4s, si sarebbe verificata la collisione. Tuttavia, il test non è all'interno delle catture fornite, in quanto non esplicitamente richiesto.

3) Come si può forzare i nodi ad utilizzare la procedura di handshake RTS/CTS vista in classe? Qual è il ragionamento dietro questa procedura?

È presente un parametro chiamato RtsCtsThreshold (classe WifiRemoteStationManager), che costituisce la lunghezza minima in bytes, espressa come unsigned int, che se ecceduta dalla dimensione del pacchetto forza l'utilizzo del sistema di prenotazione per la trasmissione, attraverso l'invio di una frame per richiedere la trasmissione (RTS) e la ricezione di una che consenta di cominciare a trasmettere (CTS). Noi abbiamo settato tale parametro a 100 bytes, in quanto nella simulazione i pacchetti UDP, con massima probabilità di collidere, sono lunghi 512 bytes. Settandolo a 50 bytes, avremmo protetto anche ai segmenti ARP, tuttavia la loro probabilità di collidere è molto inferiore.

4) Forzare l'uso di RTS/CTS nella rete utilizzando il parametro useRtsCts:

- Ci sono delle collisioni adesso?
- Quali sono i benefici di RTS/CTS?
- Dove si può trovare ed analizzare le informazioni relative al Network Allocation Vector?

Continuano a non verificarsi collisioni, anche in virtù del sistema di prenotazione della trasmissione. Il sistema di prenotazione evita le collisioni, eliminando il problema dell'hidden terminal: un terminale in trasmissione, che l'altro pronto a trasmettere non rileva. Gli unici pacchetti che possono collidere sono proprio le richieste di trasmissione, tuttavia è poco probabile che ciò avvenga in quanto hanno dimensione esigua (20 B). Inoltre l'overhead aggiunto è marginale (2 SIFS + tempo di trasmissione di RTS e CTS, lungo 14 B).

Nell'header di RTS e CTS per il protocollo 802.11, all'interno del campo "Duration", subito dopo Frame Control Field, è contenuta la durata del Network Allocation Vector.

5) Calcolare il throughput medio complessivo delle applicazioni

Per il calcolo del Throughput, consideriamo la quantità di dati trasmessa dai client, cioè due frames per ogni client lunghi 512 B e le risposte del server, con pacchetti della stessa dimensione. Essendoci due client, vengono trasmessi in tutto 512 B/frame*2 frames/client*2 clients*2 = 4096 B.

Senza RTS e CTS, la trasmissione del primo pacchetto dal nodo 4 avviene a t = 1.005992 s, mentre l'ultimo pacchetto, trasmesso dal server al nodo 3, arriva a t = 4.010014 s, quindi il throughput medio complessivo delle applicazioni è pari a 4096B/3,004022s = 1363.5 Bps = 10.91 kbps.

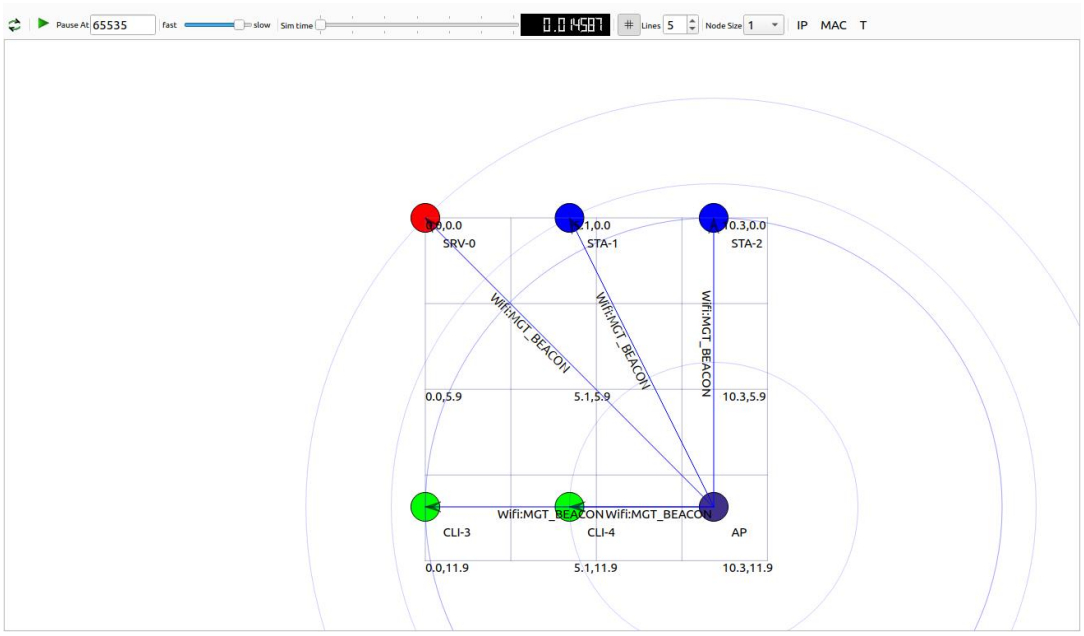
Utilizzando RTS e CTS il primo pacchetto parte a t = 1.006668 s, l'ultimo pacchetto arriva a t = 4.011366 s, perciò il throughput diventa 1363 Bps.

Gruppo di lavoro: #34
Componenti: Alessio Maiola (GL), Alessandro Maone, Andrea De Carlo, Davide Caronna

Task 2 : *Wireless Local Area Network – Infrastructure Mode*

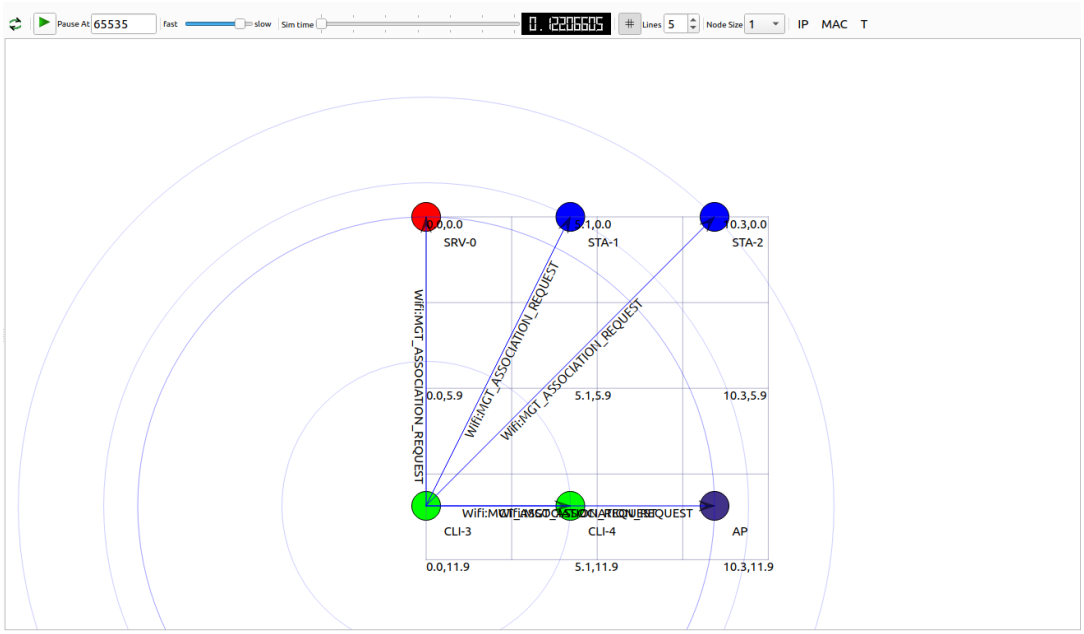
1) Spiegare il comportamento dell'AP. Cosa succede fin dal primo momento dell'inizio della simulazione?

L'access point sin dall'inizio della sua attività invia continuamente e regolarmente beacon frames. Queste frames permettono agli hosts di connettersi all'AP, come si può vedere dai primi pacchetti catturati da wireshark (un beacon frame, varie association requests con annesse responses e inframezzate da relativi Ack). Durante tutta la comunicazione, l'AP svolge un ruolo da "mediatore": le frames vengono trasmesse dagli hosts in broadcast con flag "To DS" pari a 1 e indirizzate all'AP. Questo, dopo la ricezione, risponde con Ack, per poi inviarle in broadcast a tutti i terminali, indirizzandole alla reale destinazione, con flag "From DS" pari a 1. Infine, l'AP riceve un Ack dal terminale di destinazione.



[Fig.3]

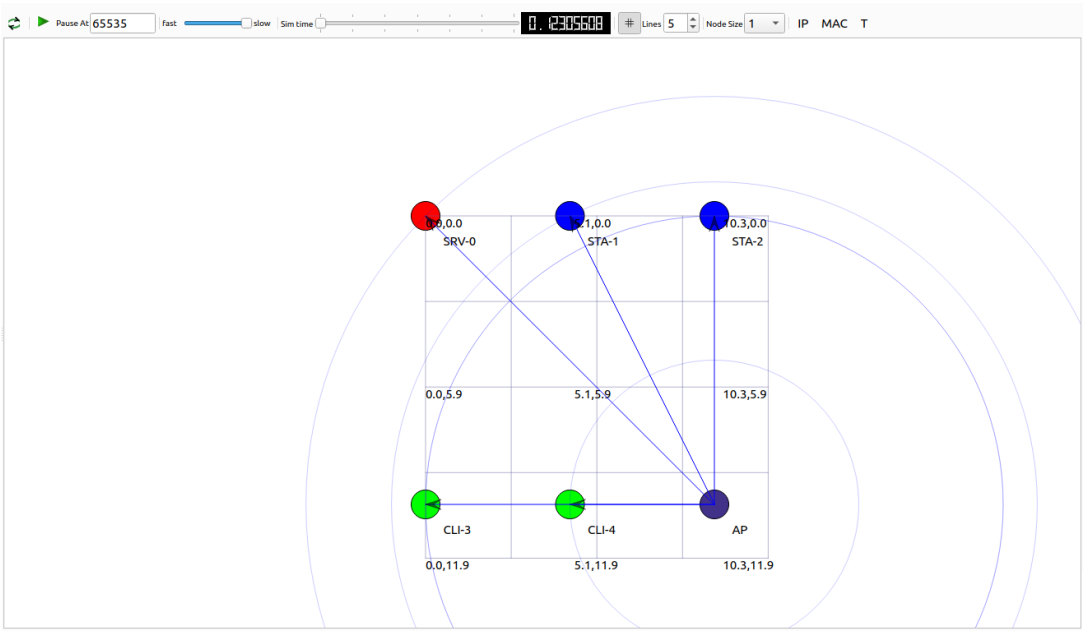
[cattura primo beacon frame dell'Access Point]



[Fig.4]

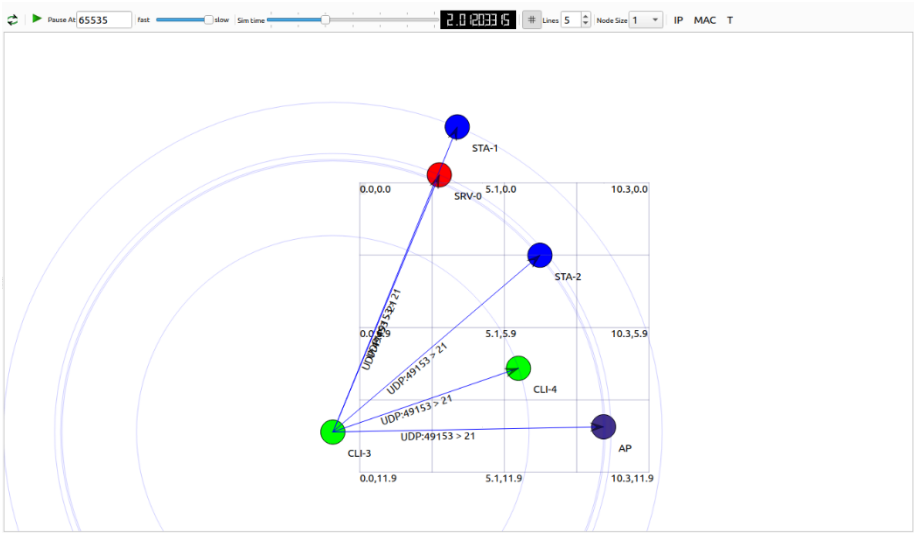
[cattura association request del nodo 3]

Gruppo di lavoro: #34
Componenti: Alessio Maiola (GL), Alessandro Maone, Andrea De Carlo, Davide Caronna



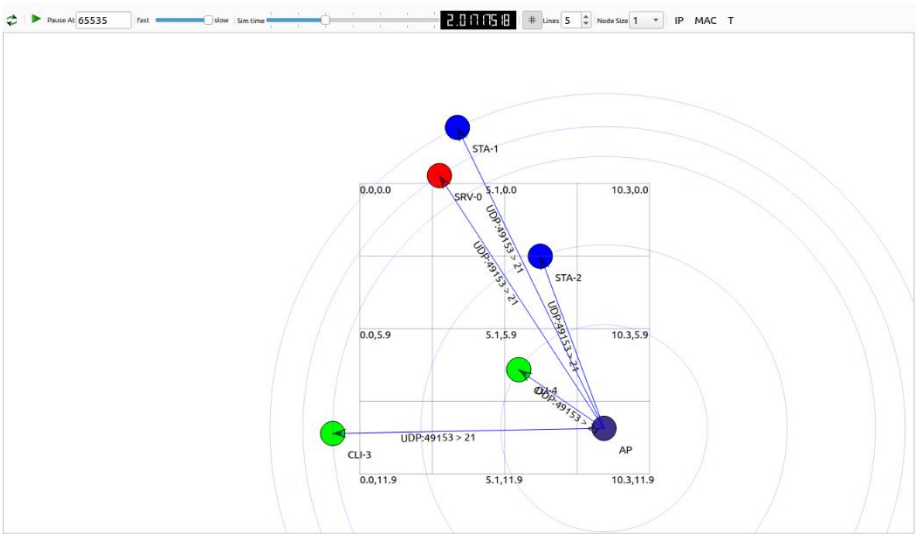
[Fig.5]

[cattura association response al nodo 3 (non etichettata da NetAnim ma verificata dal confronto con il file di pcap)]



[Fig.6]

[cattura del pacchetto udp (source: client-3, receiver: ap, destination: server-0)]



[Fig.7]

[rinvio da parte dell'ap del pacchetto sopracitato (source: ap, destination: server-0)]

Gruppo di lavoro: #34
Componenti: Alessio Maiola (GL), Alessandro Maone, Andrea De Carlo, Davide Caronna

2) Analizzare il beacon frame. Quali sono le sue parti più rilevanti? Specificare il filtro Wireshark ed il file utilizzati per l'analisi.

Le parti più rilevanti di un beacon frame sono il nome della rete (SSID) (nel nostro caso uguale alla somma delle matricole = 7782857), l'indirizzo MAC dell'access point (AP) e il beacon interval, uguale a 100 TU = 102.4 ms, ovvero il tempo tra un beacon frame e il successivo. Queste informazioni sono importanti perché permettono ai dispositivi client di identificare la rete, ricevendone un avviso della relativa presenza, stabilire una connessione con l'AP e sincronizzarsi una volta connessi. Inoltre, queste informazioni possono essere utilizzate dall'amministratore di rete per monitorare e gestire la rete.

File considerato: task2-off-5.pcap
Filtro wireshark: wlan[0]==0x80

3) Come per il Task 1, forzare l'uso di RTS/CTS nella rete utilizzando il parametro "useRtsCts":

- Ci sono delle collisioni adesso? Spiegare il perché.

In questa task si verifica effettivamente una collisione a tempo t=4s, quando i nodi 3 e 4 provano contemporaneamente a trasmettere un pacchetto UDP. In particolare collidono la risposta del server, trasmessa dall'AP al nodo 4, e il pacchetto inviato dal nodo 3 all'AP. L'utilizzo del sistema di prenotazione della trasmissione, come è visibile anche dalle catture su Wireshark, elimina la collisione. Essa si evidenzia sempre dalla flag "Retry" del Frame Control Field dell'header del protocollo 802.11, pari a 1 per i pacchetti UDP sopracitati senza RTS/CTS. Forzando l'uso del sistema di prenotazione, solo i pacchetti RTS possono collidere, tuttavia, in virtù della loro esigua dimensione, ciò non si verifica.