# Introduzione

Tutti i file **.pcap** ed i file **.xml**, sono stati generati utilizzando il parametro “RngRun”, dato dalla somma delle nostre matricole, che è pari a 7783477.

## Task 1

Q1) Tutti i frame ricevono l’acknowledgement? Spiegare perché.

A1) Analizzando la sequenza di pacchetti con il programma NetAnim è possibile verificare che tutti i pacchetti UDP ed ARP (tranne quelli broadcast) sono seguiti da un ACK di ricezione dal destinatario [**[immagine 1]**](#Immagine1). Abbiamo infatti usato una connessione CSMA/CA con lo standard IEEE 80211g, che prevede l’attesa dell’ACK da parte della stazione ricevente in un tempo SIFS (Short Inter Frame Space). Osservando lo scambio di pacchetti su Wireshark, notiamo che sia il nodo che ricevono i pacchetti scambiati tra , ed . Infatti, si trovano tutti nel range di copertura di ogni dispositivo [**[immagine 2]**](#Immagine2).

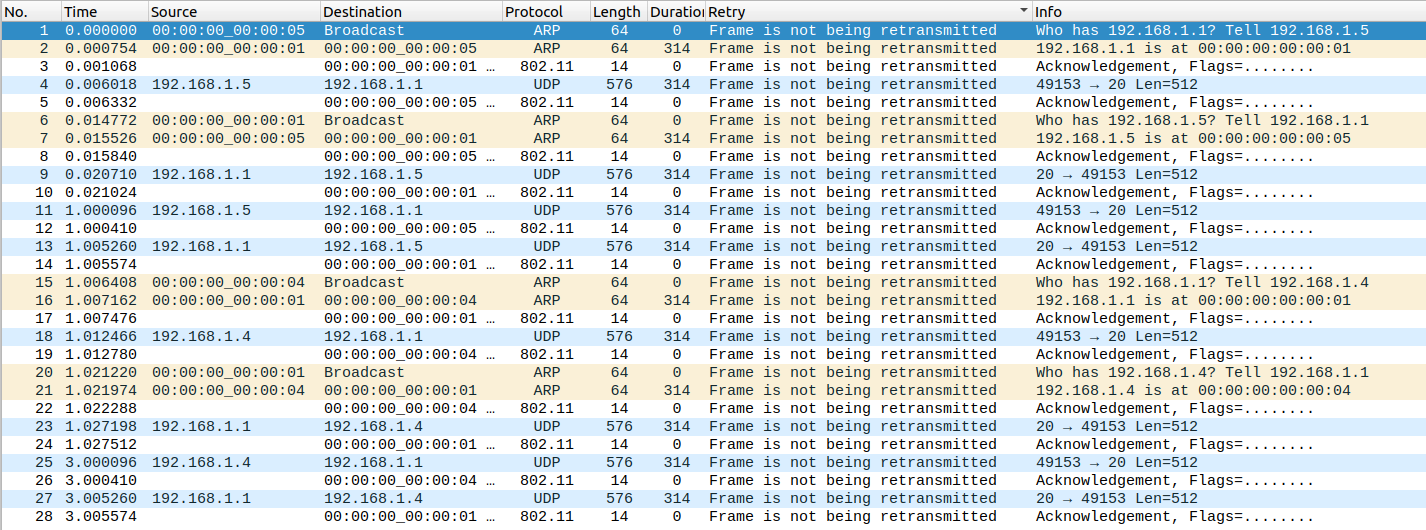


Immagine 1 (task1-n2-off.pcap)

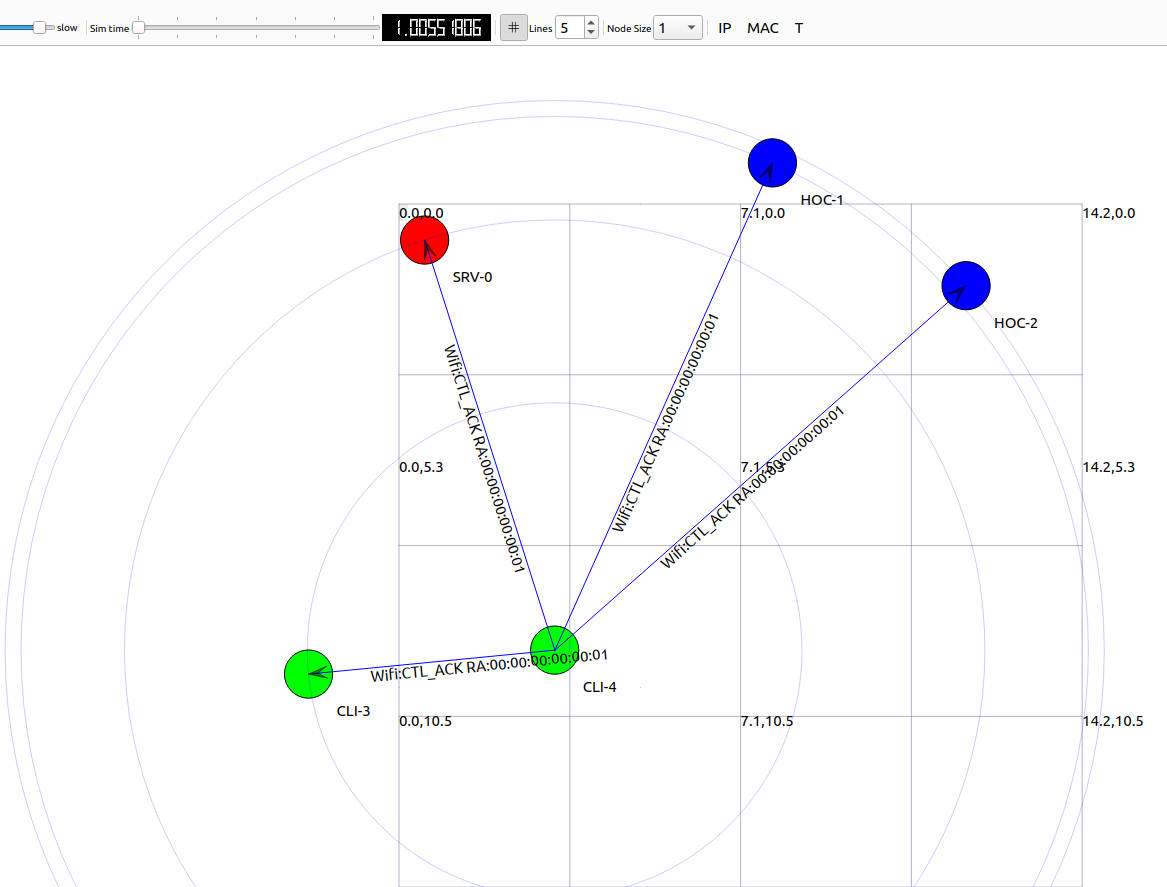


Immagine 2 (wireless-task1-rts-off.xml)

Q2) Vi sono delle collisioni nella rete? Spiegare perché. Come sei arrivato a questa conclusione?

A2) Non sono presenti collisioni in quanto non sono presenti ritrasmissioni di pacchetti. Questo si può verificare nella sezione “IEEE 802.11 Data, Flags > Frame control field > Flags > Retry” di Wireshark. Questo parametro è 0 per tutta la durata della simulazione, che implica “**Frame is not being retrasmitted**” [**[immagine 3]**](#Immagine3). Inoltre, l’unico momento possibile in cui potrebbe avvenire una collisione è al tempo , poiché sia sia desiderano inviare il proprio pacchetto. Grazie all’uso del “**Collision Avoidance**”, i due nodi sono in grado di rilevare il canale occupato e si mettono in attesa di un tempo casuale, nella speranza che i due tempi siano diversi per creare una sequenzialità [**[immagine 4]**](#Immagine4).

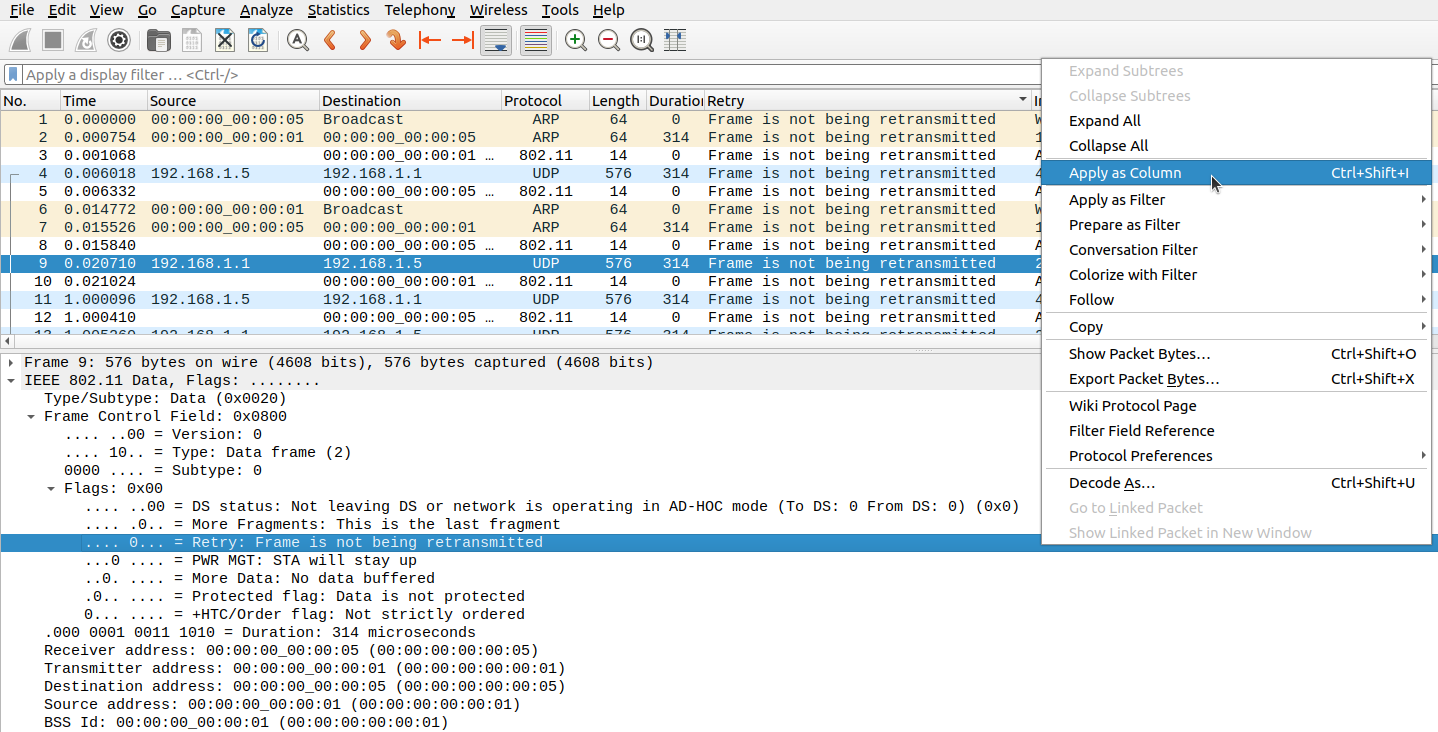


Immagine 3 (task1-n2-off.pcap)

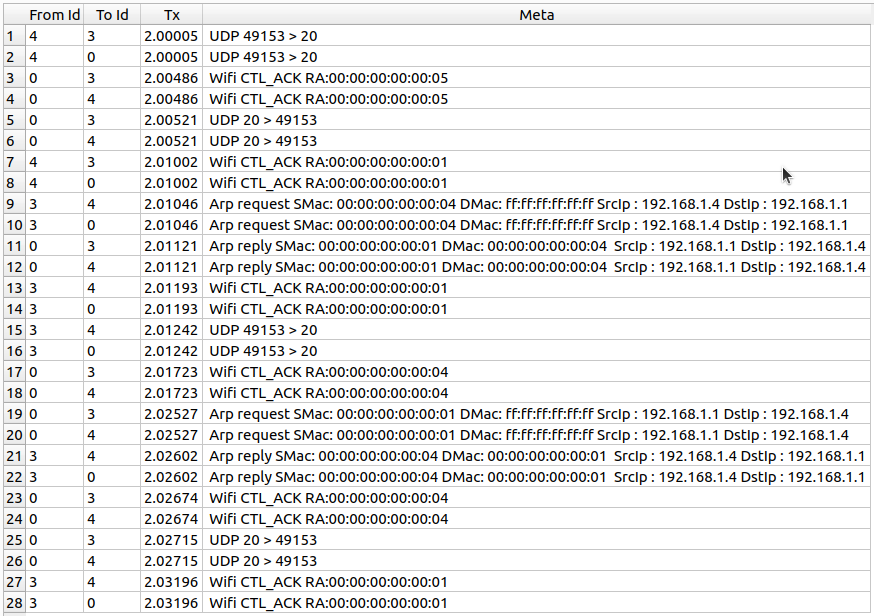
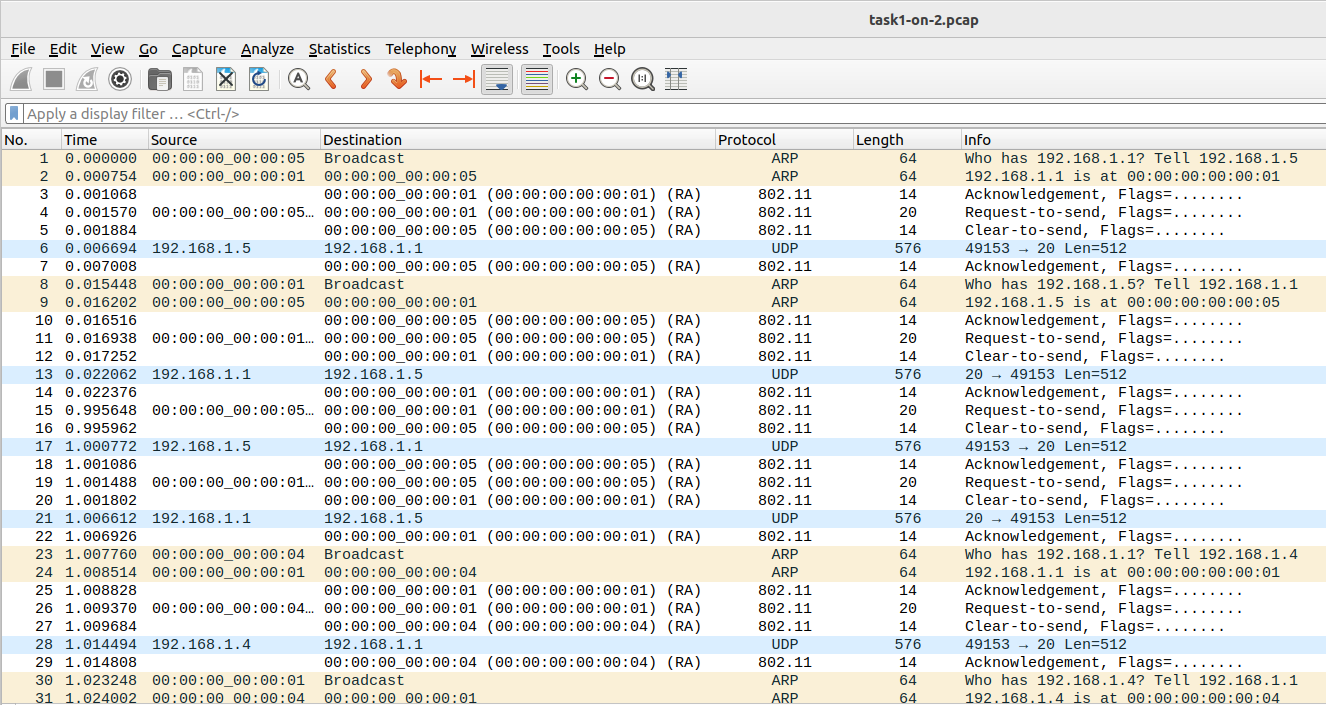
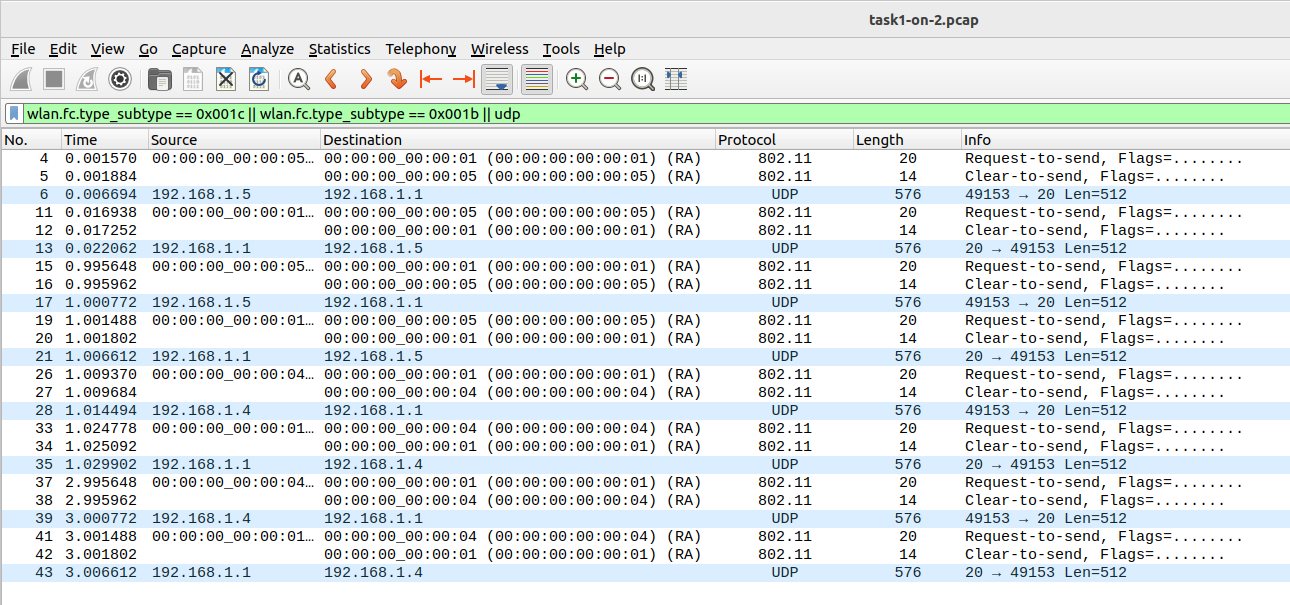


Immagine 4 (wireless-task1-rts-off.xml)

Q3) Come si può forzare i nodi ad utilizzare la procedura di handshake RTS/CTS vista in classe? Qual è il ragionamento dietro questa procedura?

A3) È possibile forzare l’uso di RTS/CTS impostando un “**threshold limit**”, in modo che il protocollo si applichi solamente ai pacchetti scambiati con una dimensione maggiore del suddetto limite. Avendolo impostato a 100, sono solo i pacchetti UDP, con dimensione 512bytes, ad essere colpiti dal cambiamento [**[immagine 5]**](#Immagine5). L’RTS/CTS è una tecnica di trasmissione che prevede l’invio ulteriore di due pacchetti prima di mandare un messaggio. Inizialmente viene inviato il pacchetto RTS che indica la durata della trasmissione, in risposta viene inviato un pacchetto CTS che impedisce agli altri nodi di inviare messaggi. A questo scambio è seguito il messaggio UDP [**[immagine 6]**](#Immagine6).

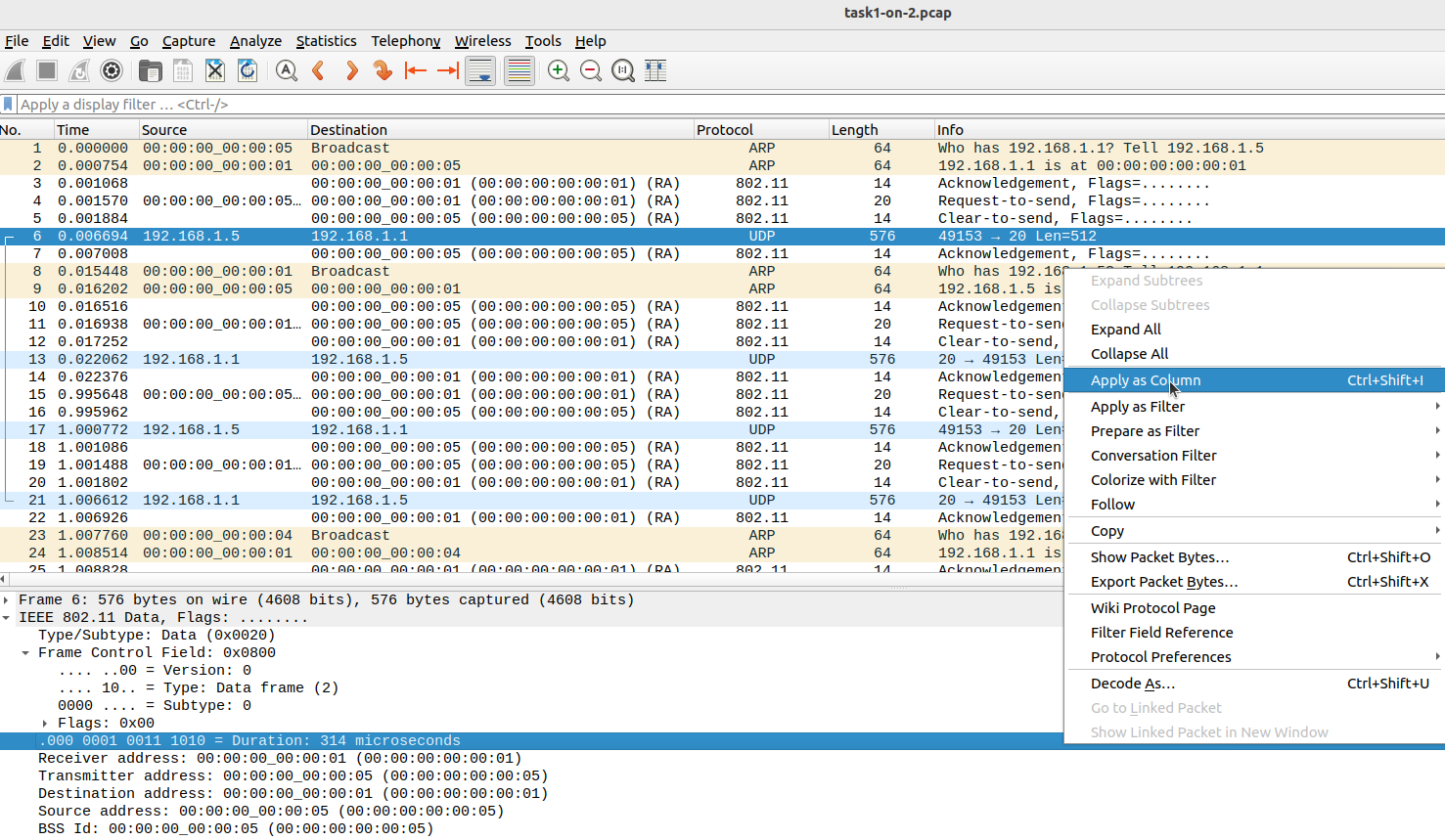
 Immagine 5 (task1-n2-on.pcap)

Immagine 6 (task1-n2-off.pcap)

Q4) 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?

A4) In questa configurazione non ci sono collisioni dato l’uso del protocollo RTS/CTS. Il beneficio principale è infatti quello di diminuire la probabilità di collisioni attraverso una prenotazione del canale per una trasmissione sicura del pacchetto. Così facendo, viene risparmiato il tempo di una eventuale ritrasmissione; questo metodo viene spesso usato per pacchetti di grandi dimensioni. In Wireshark è possibile trovare il parametro **NAV** all’interno dell’header MAC, guardando la sezione “IEEE 802.11 Data > Frame Control Field > Duration” [**[immagine 7]**](#Immagine7). Questo valore indica la durata effettiva della comunicazione, includendo l’invio del messaggio e la ricezione del pacchetto ACK.

Immagine 7 (task1-n2-off.pcap)

Q5) Calcolare il throughput medio complessivo delle applicazioni

A5) Il Throughput a livello di applicazione riguarda esclusivamente i messaggi UDP, che sono composti da di dati e di header. Il throughput è la dimensione totale dei pacchetti inviati sul tempo totale della trasmissione, ovvero il tempo dall'invio del primo messaggio UDP fino alla ricezione dell'ultimo ACK. Consideriamo i due casi con e senza procedura RTS/CTS, analizzando solo i pacchetti UDP [**[immagine 8]**](#Immagine8) senza tener conto dell’header dato che consideriamo il throughput delle applicazioni.

Come previsto il throughput con RTS/CTS è leggermente minore, visto che viene speso del tempo per garantire una trasmissione sicura.

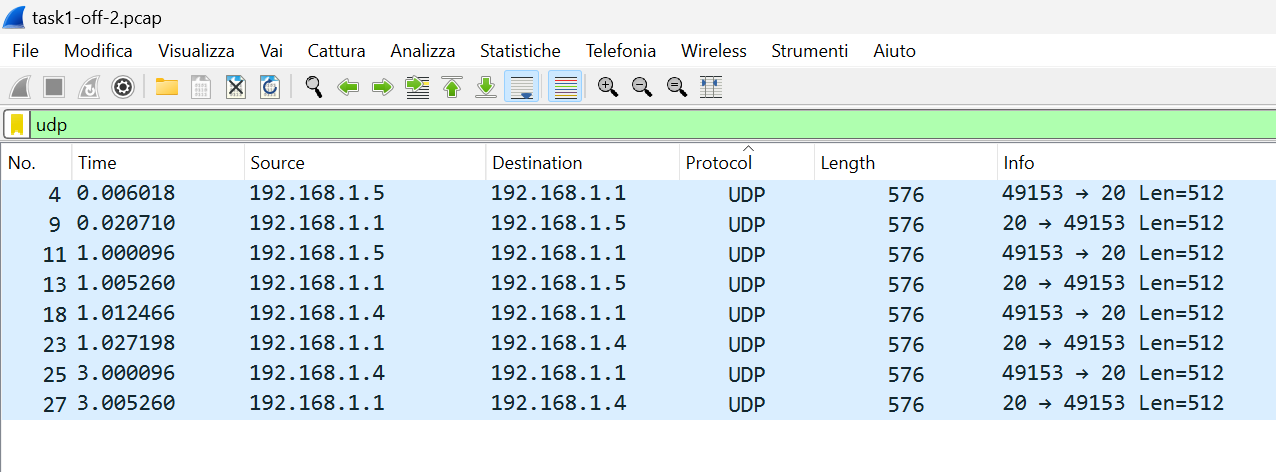


Immagine 8 (task1-off-2.pcap)

## Task2

Q1) Spiegare il comportamento dell’AP. Cosa succede fin dal primo momento dell’inizio della simulazione?

A1) L’Access Point ha una posizione fissa nello spazio e gestisce lo scambio di dati Client/Server e le richieste ARP, aggiornando la rispettiva tabella [**[immagine 9]**](#Immagine9). Come previsto dallo standard IEEE 802.11g, l’AP trasmette frame beacon in broadcast ad intervalli regolari di , come si può verificare con i programmi Wireshark e NetAnim. Il frame beacon serve a segnalare la sua presenza agli altri nodi e per sincronizzare le operazioni dei nodi connessi.  Seguono i messaggi di associazione tra i vari nodi con l’AP: all’invio del frame di “Association Request” da parte di un nodo, l’Access Point risponde con un frame di “Association Response”, necessario per stabilire delle connessioni logiche.

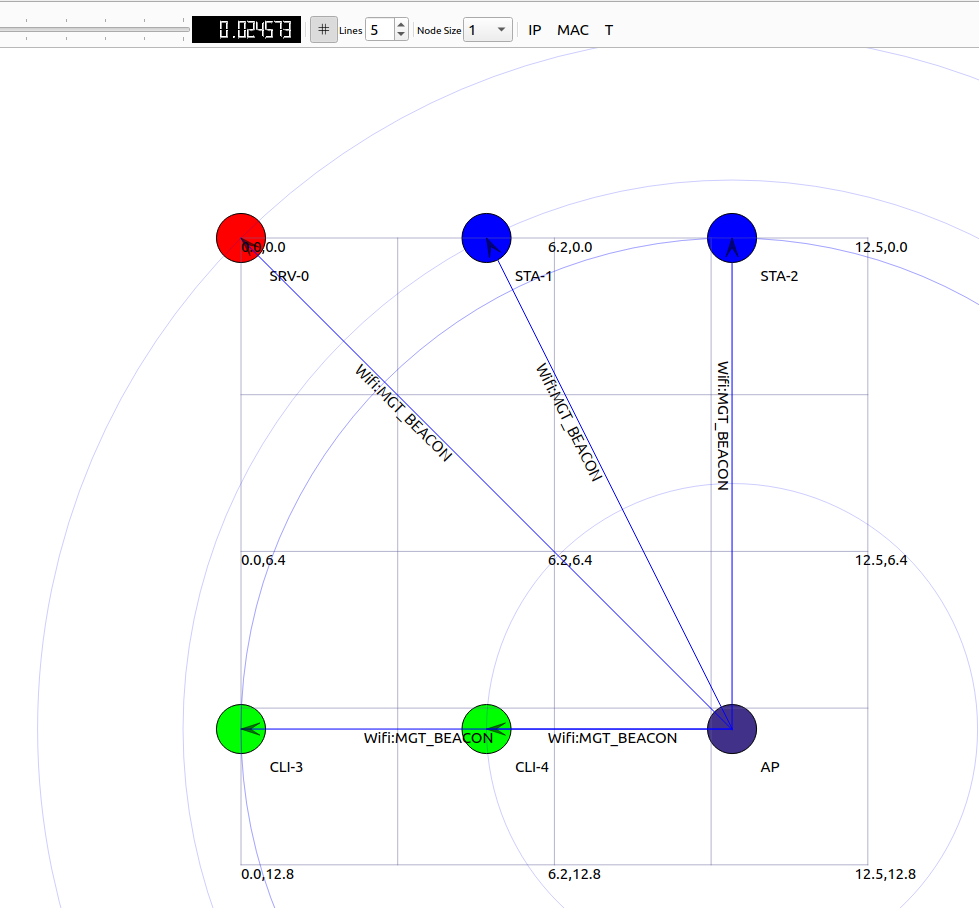


Immagine 9 (wireless-task2-rts-off.xml)

Q2) Analizzare il beacon frame. Quali sono le sue parti più rilevanti? Specificare il filtro Wireshark ed il file utilizzati per l’analisi.

A3) Il file analizzato è “**task2-off-4.pcap**”, mentre il filtro utilizzato su Wireshark è “**wlan.fc.type\_subtype == 0x08**” [**[immagine 10]**](#Immagine10). I campi significativi nel corpo di un frame beacon sono i seguenti:

1. **Timestamp**: tempo (in microsecondi) in cui l'AP è attivo. Questo risulta utile per la sincronizzazione tra tutte le stazioni connesse allo stesso AP;
2. **Beacon Interval**: intervallo di tempo tra due trasmissioni di frame beacon consecutive;
3. **Capability Information**: informazioni sulla capacità della rete o del dispositivo. Verifica se il tipo di rete in uso è una rete ad hoc o una rete infrastrutturale;
4. **Service Set Identifiers**: codice identificativo di una LAN wireless;
5. **Supported rates**: indica i bit rate che sono supportati.

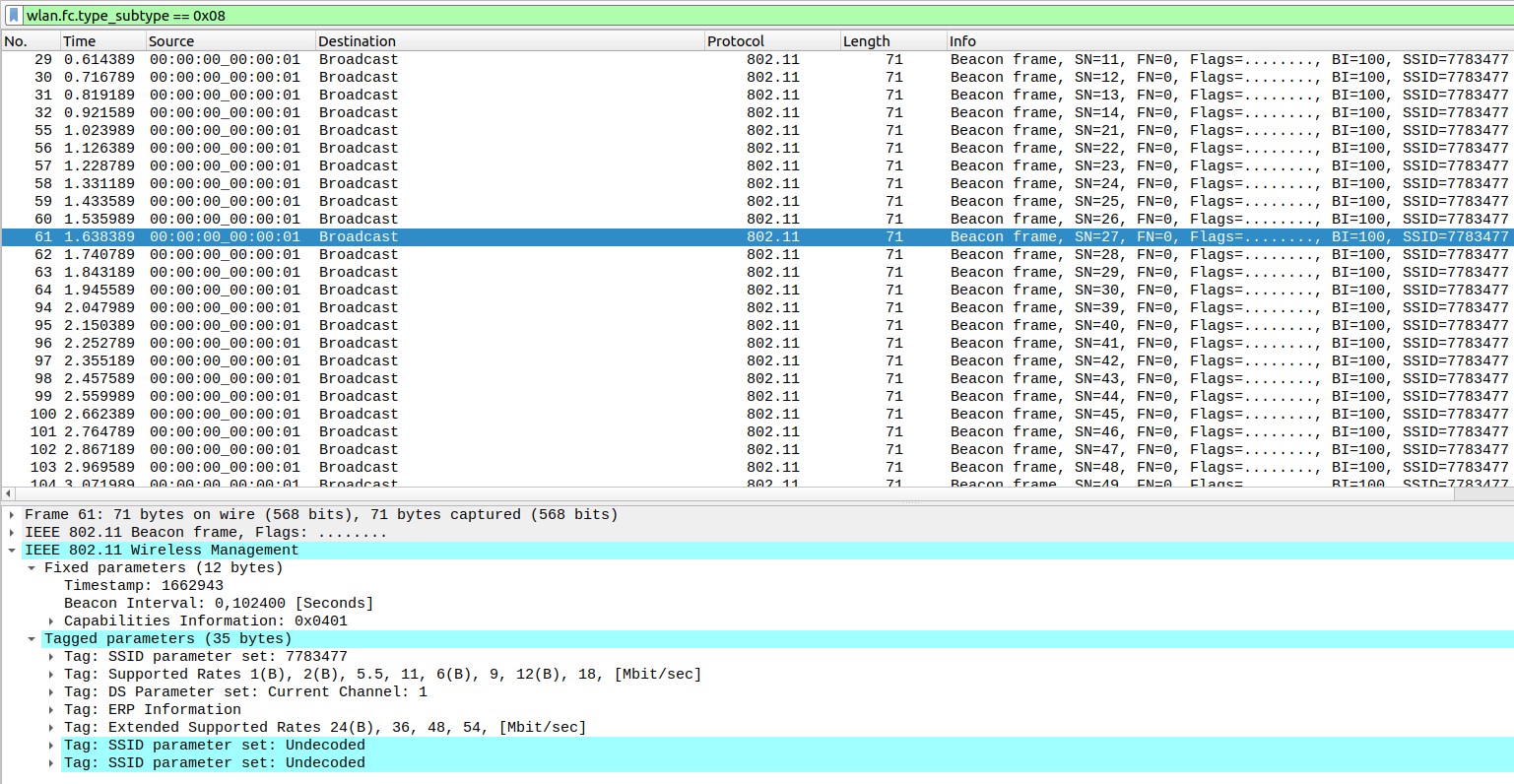


Immagine 10 (task2-off-5.pcap)

Q3) 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.

A3) Senza l’uso di RTS/CTS sono presenti potenziali collisioni durante la fase di associazione dei dispositivi con l’AP [**[immagine 11]**](#Immagine11). Durante l’invio del pacchetto UDP al secondo , notiamo che il suddetto viene ritrasmesso: il flag “**Frame is being retransmitted**” ne è una conferma [**[immagine 12]**](#Immagine12). Applicando RTS/CTS la sequenza di messaggi è più ordinata ed i pacchetti vengono ritrasmessi data l’assenza di collisioni. Questo avviene solo per i pacchetti UDP! Dato che gli altri hanno dimensione inferiore del “threshold limit”, non traggono beneficio dal RTS/CTS [**[immagine 13]**](#Immagine13). Grazie a questo protocollo è facile osservare una grande differenza in termini di efficacia, data l’assenza di collisioni per pacchetti di grandi dimensioni.

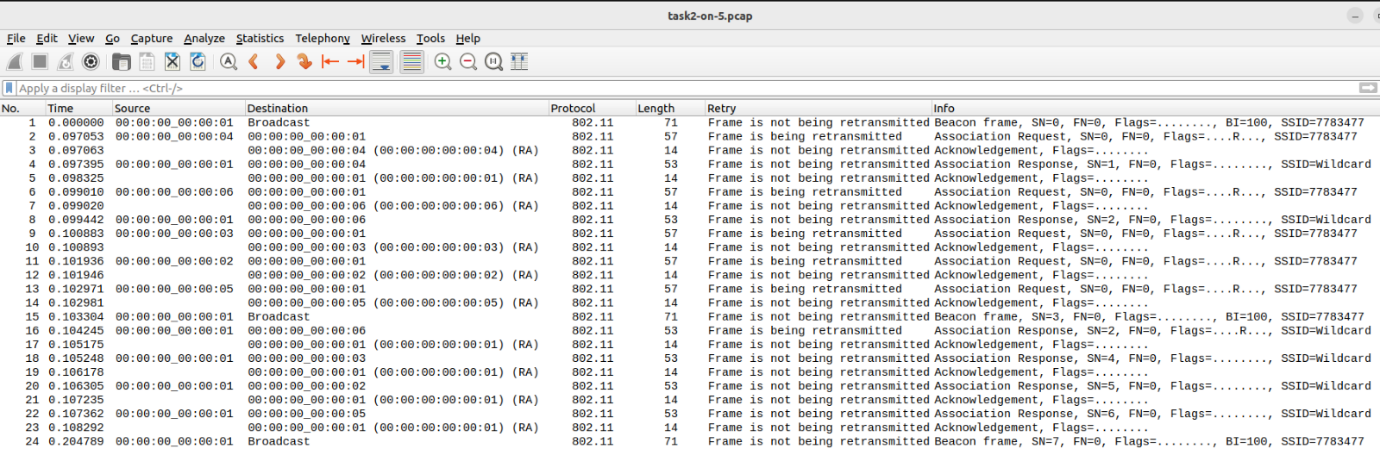
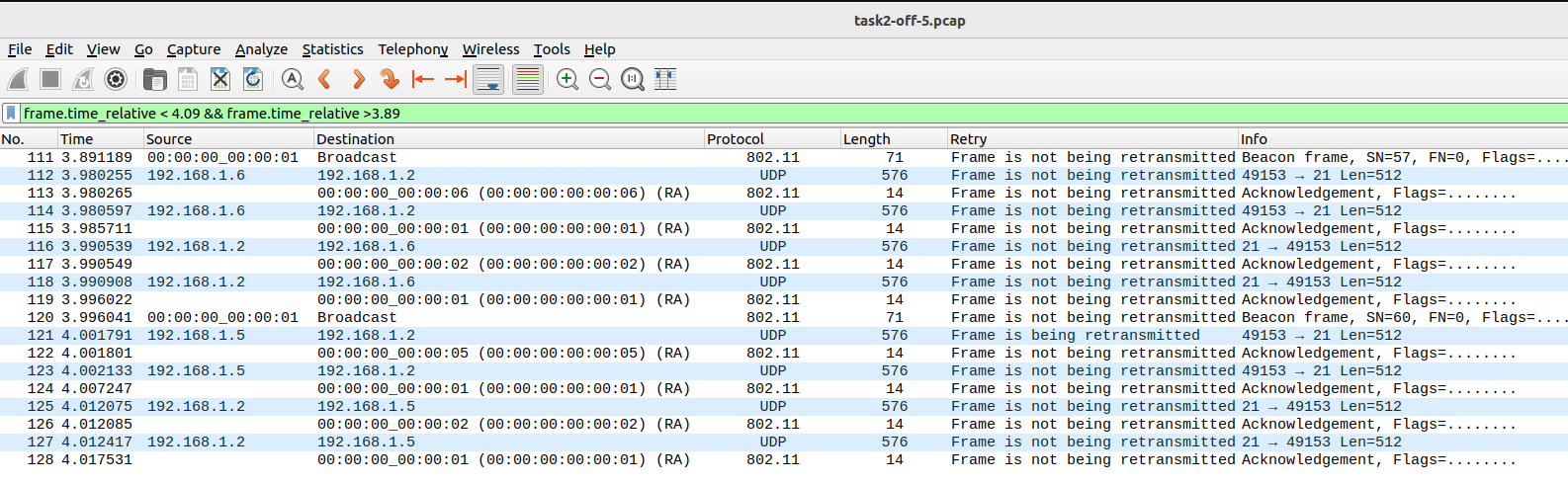
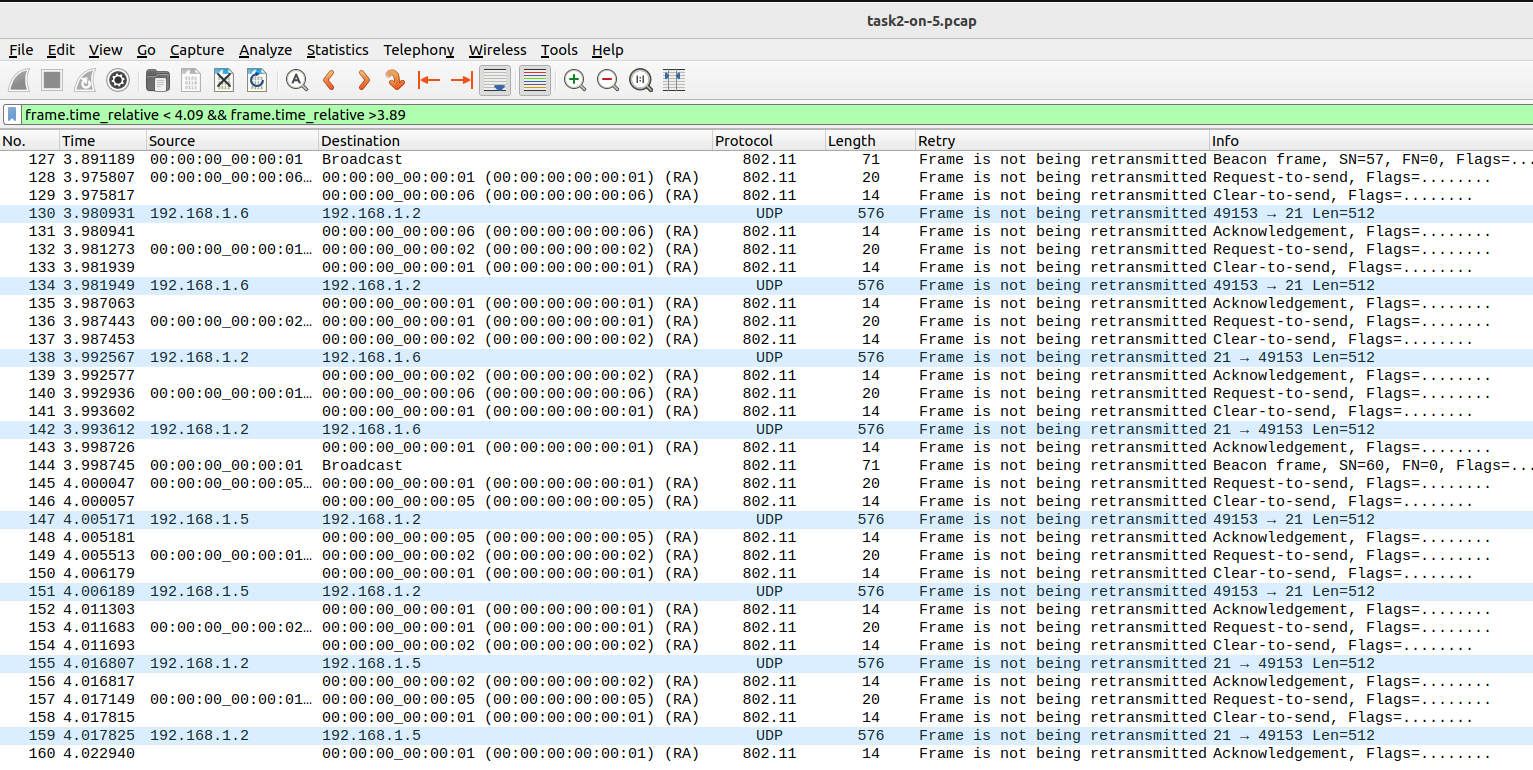


Immagine 11 (task2-on-5.pcap)

 Immagine 12 (task2-off-5.pcap)

 Immagine 13 (task2-on-5.pcap)

### Team 25

Lavoro compiuto dal Team 25, composto da:

* Simone Federico Laganà – 1946083
* Filippo Guerra – 1931976
* Giulio Di Gregorio – 1943235
* Marika Fuccio – 1962183