**Task n.1**

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

Si in quanto lo standard utilizza il seguente meccanismo:

* La stazione trasmittente testa il mezzo, e se è occupato la trasmissione viene rimandata. Se il mezzo è libero per un certo tempo nello standard (DIFS), la stazione effettua la trasmissione.
* La stazione ricevente controlla il campo CRC del pacchetto ricevuto e invia un pacchetto di acknowledgement (ACK) che indica alla stazione trasmittente che non si è verificata nessuna situazione di collisione.

Se la stazione che ha iniziato la trasmissione non riceve l’ACK allora ritrasmetterà il pacchetto fintanto che non lo riceve.

Dall'analisi del documento Wireshark risulta che tutti i frame ricevono l'ACK.

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

Nella rete non sono presenti collisioni in quanto ad ogni frame è associato un unico ACK di avvenuta ricezione, e i frame non vengono rispediti più di una volta; quindi, non sono presenti collisioni nella rete.

Si noti infatti che sono stati mandati 12 messaggi ACK per 8 pacchetti UDP e 4 protocolli ARP non *Broadcast*:

Immagine che contiene tavolo

Descrizione generata automaticamente



**3) Come si può forzare i nodi ad utilizzare la procedura di handshake RTS/CTS vista in classe?**

**Quale è il ragionamento dietro questa procedura?**

È presente un parametro chiamato *RTS Threshold* che indica la grandezza della frame oltre il quale l'RTS/CTS handshake deve avvenire.

Se la grandezza del pacchetto del nodo è maggiore dell'*RTS Threshold*, l'RTS/CTS handshake ha inizio, sennò il frame viene mandato immediatamente.

Nel codice è possibile cambiare questo valore, infatti dato che la lunghezza dei frame mandati è di massimo 576 bytes, impostando l'*RTS Threshold* ad esempio a 575 è possibile forzare l'uso dell'handshake:



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

**4.1) Ci sono delle collisioni adesso?**

Attivando l'RTS/CTS su ogni frame, anche questa volta non sono presenti collisioni in quanto il numero di frame mandati è uguale al numero di ACK ricevuti.

**4.2) Quali sono i benefici di RTS/CTS?**

Il beneficio dell'RTS/CTS consiste nel fatto che riduce la probabilità di collisione su un'area di ricezione che è nascosta all'interno dell'intervallo di tempo necessario alla trasmissione dell'RTS, poiché la stazione sente il CTS e definisce il mezzo come occupato fino alla fine della trasmissione.

L’informazione relativa al tempo di trasmissione protegge inoltre l’area del trasmettitore dalle collisioni durante l’ACK da parte di quelle stazioni che sono fuori dall’area di visibilità della stazione che deve fornire l’acknowledge.

Bisogna inoltre osservare che a causa delle ridotte dimensioni dei pacchetti RTS e CTS, il meccanismo riduce anche l’overhead dovuto alla collisione, poiché non è necessaria la ritrasmissione dell’intero pacchetto dati.

**4.3) Dove si può trovare ed analizzare le informazioni relative al Network Allocation Vector?**

Nell'header dell'indirizzo MAC è presente un campo chiamato *Duration/ID* all'interno del quale è presente il valore del NAV in microsecondi.

Immagine che contiene testo

Descrizione generata automaticamente

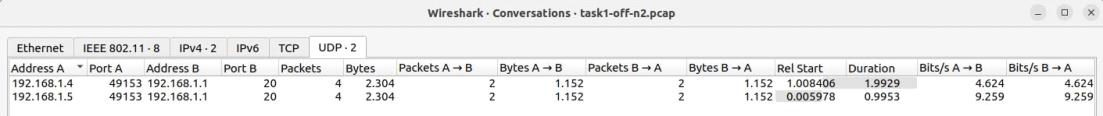
Si noti che il campo *Duration* è stato pure applicato come colonna nell'interfaccia Wireshark.

**5) Calcolare il throughput medio complessivo delle applicazioni**

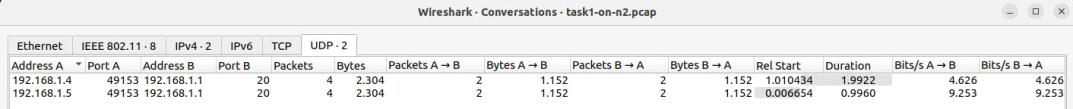
Su Wireshark andando su *Statistics -> Conversations -> UDP•2*, è possibile vedere i Throughput delle varie applicazioni.

Ad esempio, se si considerano i due pacchetti mandati dal nodo 3 al nodo 0 (in totale (576\*2)\*8 bit = 9216 bit), essi vengono trasmessi in 1.9929s (campo *Duration*), e si ha dunque un Throughput di 9216/1.9929=4624 bits/s (campo Bits/s A -> B)).

I Throughput delle altre applicazioni sono riportati di seguito:



Nel caso in cui si abiliti l'RTS/CTS, il throughput varia:



Si noti che sul nodo 3 il throughput aumenta leggermente, mentre sul nodo 4 il throughput diminuisce, in quanto al secondo 2 il nodo 3 sta già occupando il mezzo; quindi, non viene trovato libero dal nodo 4.

**Task n.2**

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

Al secondo 0.0 l’AP invia un beacon frame in broadcast, così che i nodi nelle vicinanze sappiano dell’esistenza della rete.

Successivamente i vari nodi che hanno necessità di unirsi alla rete inviano un “*association request*” all’AP e aspettano di ricevere sempre dall’AP un “*association response*”, che sta ad indicare l’avvenuta partecipazione del nodo nel network relativo all’AP.

L’AP continua costantemente ad inviare beacon frame in broadcast, per avvisare eventuali altri nodi della presenza della rete.

Inoltre, tutti i pacchetti UDP inviati, vengono ricevuti dall’AP che ha lo scopo di rinviarli nella destinazione corretta. Anche nel RtsCts, è l’AP a ricevere eventuali RTS ed è sempre l’AP a rispondere con CTS.

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

È possibile isolare i beacon frame utilizzando i seguenti filtri:

*wlan[0] == 0x80*

*wlan.fc.type\_subtype==8*

Su Wireshark i dati più rilevanti da analizzare sono:

***Timestamp*** = tempo in riferimento utilizzato dai dispositivi collegati nel BSS per sincronizzarsi con la WLAN

***Beacon Interval*** = intervallo tra due trasmissioni beacon consecutive nel frame

***Capabilities information*** = valore in esadecimale che contiene informazioni sulla rete; nel nostro caso ESS=1 (rete infrastrutturante) e IBSS=0 (rete wireless ad-hoc), quindi il tipo di rete è infrastrutturante. Inoltre fornisce informazioni sull'uso dello short slot time, supportato dalla versione 802.11g

***SSID*** = indica il nome della rete (58220035)

**3) Come per il Task 1, forzare l’uso di RTS/CTS nella rete utilizzando il parametro “useRtsCts”:**

**o Ci sono delle collisioni adesso? Spiegare il perché.**

* RTS/CTS off

Andando sul campo *Analyze -> Expert Information*, tra le anomalie abbiamo notato il campo "*Retransmission (retry)*", indica quali pacchetti/frame sono stati ritrasmessi:

Immagine che contiene testo

Descrizione generata automaticamente

Considerando ad esempio il pacchetto UDP n° 113, ciò è messo anche in evidenza dal fatto che tra le *Flags* è presente un bit pari a 1 di *Retry*:

Immagine che contiene testo

Descrizione generata automaticamente

Questo indica una collisione del pacchetto UDP, con conseguente ritrasmissione

* RTS/CTS on

In questo caso non è più presente la ritrasmissione del pacchetto UDP che utilizza l'RTS/CTS per evitare le collisioni.