Telecomunicazioni, a.a. 2022/2023

Report Homework N° 2

Gruppo N° 59

Parametro utilizzato per RngRun: *7801081* (somma dei numeri di matricola)

Risposte Task 1 (*AD-HOC*):

1)

Nella presente simulazione, esaminando attentamente le tracce .pcap generate, sia forzando il parametro RTS / CTS che non, a condizione che non ci siano collisioni, si riscontra la presenza della totalità (100%) degli acknoledgement che ci si aspetterebbe, in accordo con la teoria del protocollo impiegato, ovvero 802.11 (nello specifico 802.11g, come da requisiti di progetto).

Osservazione: nella simulazione si impiega UDP allo strato di trasporto dello stack protocollare ISO/OSI. Nonostante esso di per sé non supporti i riscontri sulle trasmissioni (ACK) (che lo renderebbero un protocollo "Reliable"), facendo uso della rete wireless con protocollo della famiglia 802.11, i livelli sottostanti si prendono carico di fornire un canale affidabile alle trasmissioni dati.

2)

La risposta alla presente domanda deve essere distinta, come meglio riportato nella risposta 4) della presente sezione, sia nel caso in cui si forzi l'uso del parametro RTS / CTS e sia nel caso in cui non se ne faccia uso.

Si rimanda al caso di utilizzo del suddetto parametro alla risposta 4).

Per quanto concerne la simulazione senza impiego di RTS / CTS, è possibile riscontrare una collisione al tempo di simulazione t = 2s.

Nella fattispecie del caso in esame, tale collisione trova spiegazione nel fatto che mancando un meccanismo di “prenotazione” e/o “sincronizzazione” del canale wireless, che per natura è intrinsecamente un canale broadcast, nel momento in cui due client si trovano a comunicare un dato nello stesso identico istante, è evidente che il rischio di collisione risulti essere molto elevato.

Un riscontro sperimentale della avvenuta collisione è ottenuto dalla traccia task1-off-2.pcap impostando il filtro “wlan.fc.retry == 1”, il quale mostra i pacchetti scambiati tramite canale wireless in cui la trama di livello trasporto, che impiega il protocollo 802.11g, ha il flag “RETRY” settato ad 1.

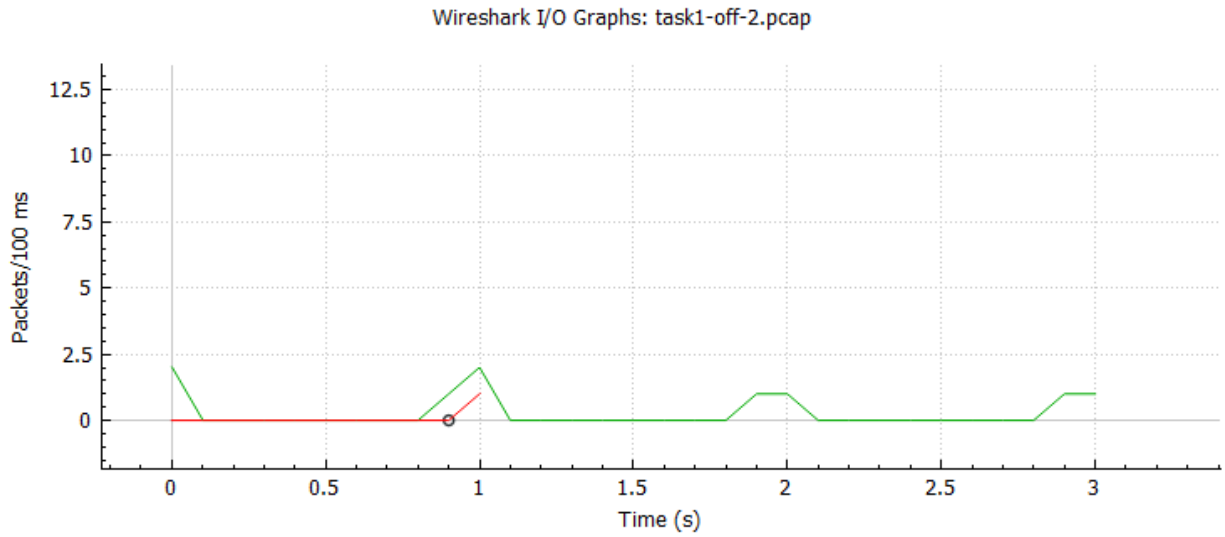

Figura 1: *retrasmission* ■

3)

La procedura di “prenotazione” del canale wireless, in considerazione del fatto che si forza l’utilizzo del meccanismo RTS / CTS, è implementabile in NS-3 a livello di script modificando il valore di default del parametro RTS / CTS THRESHOLD come esposto in Figura 2.

```
//config rts/cts
UintegerValue ctsThreshold = (useRtsCts? UintegerValue(100) : UintegerValue(2346));
Config::SetDefault("ns3::WifiRemoteStationManager::RtsCtsThreshold", ctsThreshold);
```

Figura 2: *Rts / Cts Threshold setting*

Il ragionamento che sta alla base dell'utilizzo di questo meccanismo, che poi è il modo in cui si risolve un famoso problema in ambito di comunicazioni wireless (ovvero quello dell' "hidden terminal problem"), è di prenotare il mezzo di trasmissione condiviso affinchè nessuno degli altri dispositivi possa inviare dati sul medesimo mezzo fintantoché il dispositivo autorizzato non termini il proprio processo di trasmissione.

Questo meccanismo si implementa in più fasi:

- dapprima il nodo che intente trasmettere dati invia un RTS (Request To Send);
- quando è possibile servirlo, lo stesso nodo riceve un CTS (Clear To Send);
- solo a questo punto tale nodo è abilitato ad impiegare il canale per trasmettere i propri dati.

Il processo sovraesposto è in grado di essere impiegato nei canali wireless (o più generalmente nei mezzi broadcast) in quanto, durante lo scambio di tali messaggi di controllo, tutte le altre stazioni (nodi) sono in grado di comprendere che un altro nodo è stato abilitato o meno alla trasmissione, determinando, così in loro, l'attesa di un tempo atto ad evitare ogni forma di collisione.

Inoltre, il nodo che riceve il CTS è abilitato ad impiegare il canale wireless solo per un certo slot temporale (indicato nel campo "duration" della trama del protocollo 802.11) al termine del quale gli altri dispositivi saranno in grado, a loro volta, di usufruire dello stesso meccanismo di "handshake" (o prenotazione).

4)

Conseguentemente all'impiego del parametro RTS / CTS della presente simulazione, si evince che il meccanismo sovraesposto ha consentito di prevenire ogni collisione nella trasmissione/ricezione.

L'impiego di tale parametro, come è facilmente intuibile, comporta un enorme beneficio in termini di riduzioni del numero di collisioni che avvengono nel canale wireless.

Questo beneficio ha tuttavia un costo che si paga in termini di ritardi di trasmissione e maggiore complessità nella gestione dei messaggi di RTS / CTS stessi. Pertanto è generalmente consigliato il suo utilizzo in reti wireless caratterizzate da un numero considerevole di nodi, potenzialmente in grado di generare traffico che comporta un elevato numero di collisioni.

Come accennato nella risposta 3) il protocollo della famiglia 802.11 utilizzato, presenta molteplici campi di controllo negli header delle sue trame.

Come mostrato in Figura 3, tra i vari campi che popolano il suddetto header, si trova il campo denominato "Duration ID". In tale campo, a patto che non si tratti di trame con flag di "associacion" settato a true, si può ricavare il riferimento temporale del NAV (Network Allocation Vector).

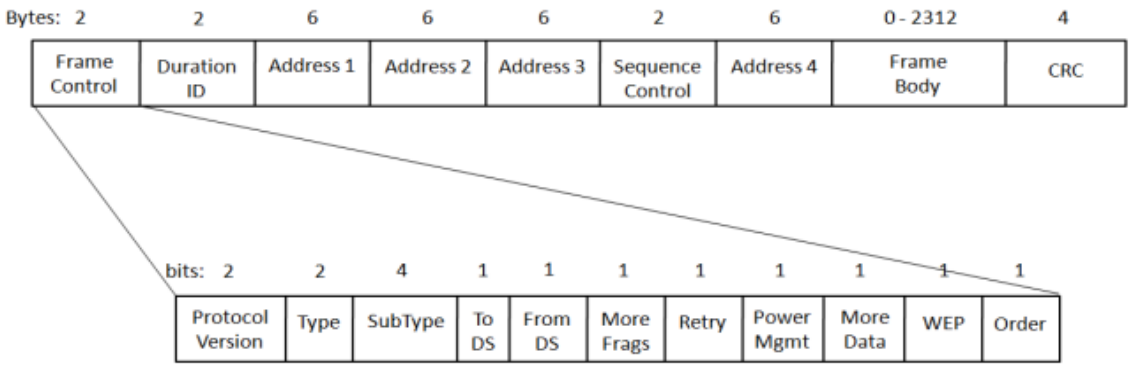

Figura 3: *802.11 frame fields*

5)

Il throughput medio complessivo degli applicativi UDP Client / Server del contesto in oggetto allo studio, è stimabile con una procedura di calcolo a partire dalle tracce .pcap generate.

In modo particolare si ottiene valutando il numero complessivo dei Bytes trasmessi dagli applicativi in rapporto al tempo intercorso tra l'istante in cui il primo segmento viene trasmesso dal client verso il server e l'istante in cui l'ultimo segmento (echo) viene trasmesso dal server verso il client.

Con tale procedimento, si ottiene il risultato di: circa ***1536 Bytes / s***.

$$\frac{8\text{pkts} \times 576\ \text{B}}{3\text{s}}$$

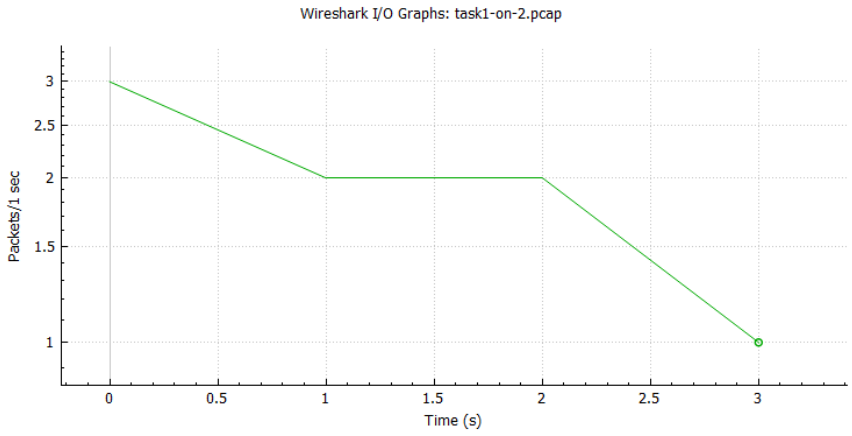

Figura 4: *Wireshark statistic tool pkts/s*

Nota:

Non essendo diversamente specificato, nel calcolo del risultato sovraesposto si è tenuto conto della totalità dei pacchetti scambiati sia dagli applicativi client che da quelli server, senza fare distinzione tra i flussi di dati.

Inoltre, si è ritenuto rilevante includere la dimensione degli header dei pacchetti nello stesso calcolo, dal momento che si è chiesto il Throughput complessivo e non il "Goodput" dei dati.

Risposte Task 2 (*INFRASTRUCTURE*):

1)

Nella configurazione proposta nel presente Task, l'AP (Access Point) svolge il ruolo di coordinatore ed intermediario tra i messaggi scambiati per mezzo del canale wireless tra le varie Stations. Per fare ciò, questo particolare nodo della rete, sin dall'inizio della simulazione, invia in broadcast (mac address: "ff : ff : ff : ff : ff : ff") dei **beacon frame** che forniscono alle station i parametri da impostare per potersi associare.

Il passo successivo sarà, come appena suggerito, quello dell'invio da parte delle station interessate alla comunicazione, di trame di **Association Request** le quali verranno seguite da un **Ack** e da una **Association Response** da parte dello stesso AP.

Osservazione: il processo di associazione è scenario del maggior numero di collisioni nel contesto della presente simulazione, in quanto è una fase in cui non si hanno ancora parametri per implementare efficacemente la funzione di collision avoidance con tecniche quali ad esempio RTS/CTS.

2)

I beacon frame citati nella risposta 1) della sezione relativa al Task in esame, estraibili dal contesto generale della traccia .pcap dell'AP mediante il filtro Wireshark "wlan.fc.subtype == 0x008", sono composti da particolari campi che dettano quindi i parametri con cui le station, come sopra citato, possono associarsi per scambiare dati utilizzando l'infrastruttura wireless centralizzata.

Sicuramente, è di particolare rilievo il campo **SSID**; i parametri dell'**AARF**, quindi i **bit rate supportati**; il **beacon interval**; il **Channel Number** ed i parametri **ESS** e **BSS ID**, che forniscono informazioni utili sulla topologia della rete; nonché, sicuramente, l'**indirizzo** con il quale si raggiunge l'**AP** a livello **MAC**.

3)

L'utilizzo del meccanismo RTS / CTS, nel contesto del presente scenario infrastrutturale, mette in evidenza come l'AP coordini il traffico assegnando le risorse del canale wireless alle varie station che ne necessitano per poter inviare dati, su base richiesta e concessione dello stesso canale.

Come facilmente intuibile, infatti, senza l'impiego di tale meccanismo, da un'attenta analisi dei file .pcap generati dall'applicativo mediante NS-3, si riscontra la presenza di collisioni al tempo t = 4s dei segmenti UDP inviati contemporaneamente, nonché di parte delle trame di Association Request.

Discorso completamente opposto, che trova ampio riscontro nella teoria, è ciò che concerne lo stesso scenario con l'uso del meccanismo RTS / CTS. Infatti, facendo uso di tale meccanismo, come già anticipato nell'osservazione fatta alla fine della risposta 1) della presente sezione, le collisioni avvengono solo su parte delle trame di Association Request.

La trattazione del comportamento dei dispositivi che impiegano il meccanismo di "prenotazione" del canale wireless, già ampiamente discussa nelle precedenti risposte, è facilmente adattabile al contesto attuale, con la differenza che in questo caso, il coordinatore che detiene la responsabilità di gestire il meccanismo di RTS / CTS è l'AP e non le singole stazioni trasmittenti come nel caso di una rete AD – HOC. In tal senso le RTS (Request To Sent) verranno inviate con destinazione l'AP, il quale concederà il CTS (Clear To Send).

Una rappresentazione grafica, estrapolata dal tool di simulazione visiva NetAnim, è illustrata in Figura 5.

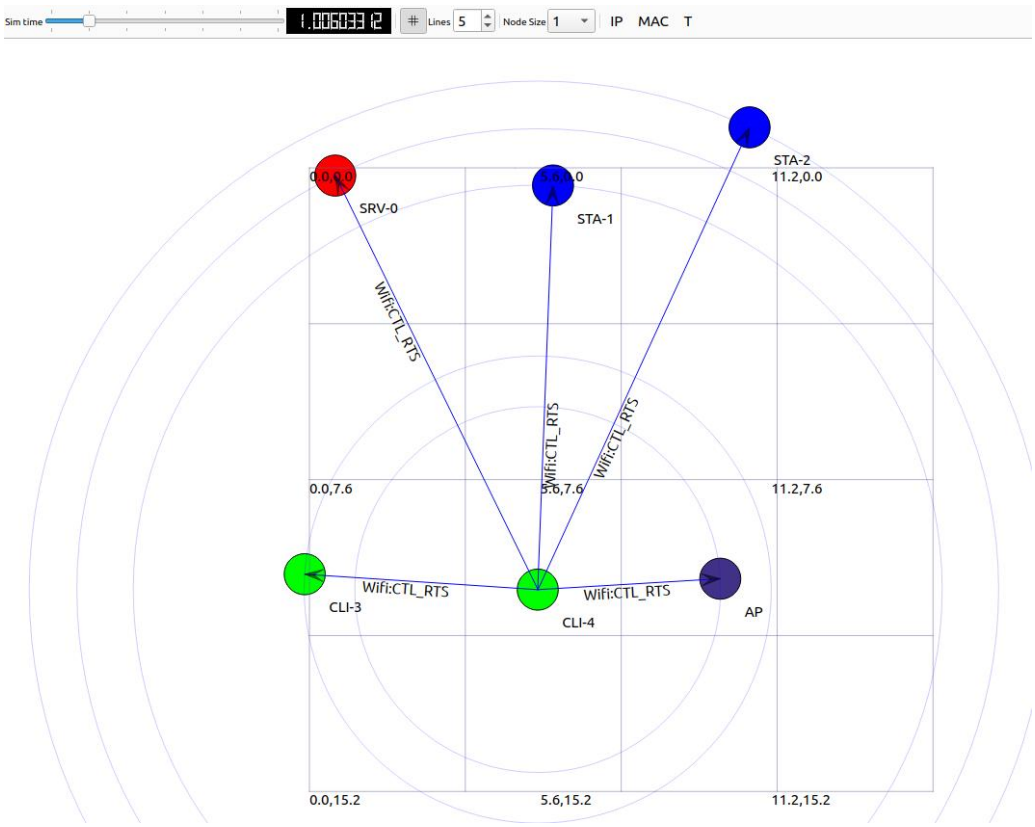

Figura 5: *invio di RTS da client-4 ad AP*