

REPORT TEAM 2 - HOMEWORK 2:


Camilla Santoro - Emanuele Mule - Simone Palumbo - Isidoro Tamassia

SOMMA MATRICOLE: 7753193

TASK 1

1. Tutti i frame ricevono l'acknowledgement? Spiegare perché

Osservando il pcap task1-off-2.pcap con Wireshark vediamo che tutti i frame ricevono l'acknowledgement CTL_ACK da parte del Wifi. In particolare, applicando il filtro “!(arp.dst.hw_mac==ff:ff:ff:ff:ff:ff)”, che nasconde i pacchetti di tipo Broadcast (che non ricevono mai ACK), osserviamo che ad ogni pacchetto inviato il sender riceve uno ed un solo ACK da parte del wifi (Immagine 1). Possiamo inoltre vedere che il numero di righe (12) che compaiono applicando il filtro “wlan.fc.type_subtype == 0x001d” (che ci evidenzia solamente gli ACK, Immagine 2) coincide con il numero di righe che compaiono applicando il filtro “!(arp.dst.hw_mac==ff:ff:ff:ff:ff:ff) && wlan.fc.type_subtype != 0x001d” (che invece evidenzia tutti i pacchetti che ci aspettiamo ricevano un ACK, Immagine 3). Dunque il numero di pacchetti inviati e di cui ci aspettiamo ACK coincide con il numero di ACK che effettivamente riceviamo.



No.	Time	Source	Destination	Protocol	Length	Info	Cumul
2	0.000754	00:00:00_00:00:01	00:00:00_00:00:05	ARP	64	192.168.1.1 is at 00:00:...	
3	0.001068		00:00:00_00:00:01 (...)	802.11	14	Acknowledgement, Flags=...	
4	0.006098	192.168.1.5	192.168.1.1	UDP	576	49153 → ftp-data(20) Len...	
5	0.006412		00:00:00_00:00:05 (...)	802.11	14	Acknowledgement, Flags=...	
7	0.016606	00:00:00_00:00:05	00:00:00_00:00:01	ARP	64	192.168.1.5 is at 00:00:...	
8	0.016920		00:00:00_00:00:05 (...)	802.11	14	Acknowledgement, Flags=...	
9	0.021790	192.168.1.1	192.168.1.5	UDP	576	ftp-data(20) → 49153 Len...	
10	0.022104		00:00:00_00:00:01 (...)	802.11	14	Acknowledgement, Flags=...	
11	1.003096	192.168.1.5	192.168.1.1	UDP	576	49153 → ftp-data(20) Len...	
12	1.003410		00:00:00_00:00:05 (...)	802.11	14	Acknowledgement, Flags=...	
13	1.008260	192.168.1.1	192.168.1.5	UDP	576	ftp-data(20) → 49153 Len...	
14	1.008574		00:00:00_00:00:01 (...)	802.11	14	Acknowledgement, Flags=...	
16	1.010282	00:00:00_00:00:01	00:00:00_00:00:04	ARP	64	192.168.1.1 is at 00:00:...	
17	1.010596		00:00:00_00:00:01 (...)	802.11	14	Acknowledgement, Flags=...	
18	1.015506	192.168.1.4	192.168.1.1	UDP	576	49153 → ftp-data(20) Len...	
19	1.015820		00:00:00_00:00:04 (...)	802.11	14	Acknowledgement, Flags=...	
21	1.022014	00:00:00_00:00:04	00:00:00_00:00:01	ARP	64	192.168.1.4 is at 00:00:...	
22	1.022328		00:00:00_00:00:04 (...)	802.11	14	Acknowledgement, Flags=...	
23	1.027218	192.168.1.1	192.168.1.4	UDP	576	ftp-data(20) → 49153 Len...	
24	1.027532		00:00:00_00:00:01 (...)	802.11	14	Acknowledgement, Flags=...	
25	3.003096	192.168.1.4	192.168.1.1	UDP	576	49153 → ftp-data(20) Len...	
26	3.003410		00:00:00_00:00:04 (...)	802.11	14	Acknowledgement, Flags=...	
27	3.008260	192.168.1.1	192.168.1.4	UDP	576	ftp-data(20) → 49153 Len...	
28	3.008574		00:00:00_00:00:01 (...)	802.11	14	Acknowledgement, Flags=...	

Immagine 1

task1-off-2.pcap

File Modifica Visualizza Vai Cattura Analizza Statistiche Telefonia Wireless Strumenti Aiuto

wlan.fc.type_subtype == 0x001d

No.	Time	Source	Destination	Protocol	Length	Info	Cumulative Bytes
3	0.001068	00:00:00_00:00:01	00:00:00_00:00:01	802.11	14	Acknowledgement, Flags=...	14
5	0.006412	00:00:00_00:00:05	00:00:00_00:00:05	802.11	14	Acknowledgement, Flags=...	28
8	0.016920	00:00:00_00:00:05	00:00:00_00:00:05	802.11	14	Acknowledgement, Flags=...	42
10	0.022104	00:00:00_00:00:01	00:00:00_00:00:01	802.11	14	Acknowledgement, Flags=...	56
12	1.003410	00:00:00_00:00:05	00:00:00_00:00:05	802.11	14	Acknowledgement, Flags=...	70
14	1.008574	00:00:00_00:00:01	00:00:00_00:00:01	802.11	14	Acknowledgement, Flags=...	84
17	1.015820	00:00:00_00:00:01	00:00:00_00:00:01	802.11	14	Acknowledgement, Flags=...	98
19	1.015820	00:00:00_00:00:04	00:00:00_00:00:04	802.11	14	Acknowledgement, Flags=...	112
22	1.022328	00:00:00_00:00:04	00:00:00_00:00:04	802.11	14	Acknowledgement, Flags=...	126
24	1.027532	00:00:00_00:00:01	00:00:00_00:00:01	802.11	14	Acknowledgement, Flags=...	140
26	3.003410	00:00:00_00:00:04	00:00:00_00:00:04	802.11	14	Acknowledgement, Flags=...	154
28	3.008574	00:00:00_00:00:01	00:00:00_00:00:01	802.11	14	Acknowledgement, Flags=...	168

task1-off-2.pcap Pacchetti: 28 - visualizzati: 12 (42.9%) Profilo: Default

Immagine 2

task1-off-2.pcap

File Modifica Visualizza Vai Cattura Analizza Statistiche Telefonia Wireless Strumenti Aiuto

(arp.dst.hw_mac == ff:ff:ff:ff:ff:ff) && wlan.fc.type_subtype != 0x001d

No.	Time	Source	Destination	Protocol	Length	Info	Cumulative Bytes
2	0.000754	00:00:00_00:00:01	00:00:00_00:00:05	ARP	64	192.168.1.1 is at 00:00:...	64
4	0.006098	192.168.1.5	192.168.1.1	UDP	576	49153 → ftp-data(20) Len...	640
7	0.016606	00:00:00_00:00:05	00:00:00_00:00:01	ARP	64	192.168.1.5 is at 00:00:...	704
9	0.021790	192.168.1.1	192.168.1.5	UDP	576	ftp-data(20) → 49153 Len...	1280
11	1.003096	192.168.1.5	192.168.1.1	UDP	576	49153 → ftp-data(20) Len...	1856
13	1.008260	192.168.1.1	192.168.1.5	UDP	576	ftp-data(20) → 49153 Len...	2432
16	1.010282	00:00:00_00:00:01	00:00:00_00:00:04	ARP	64	192.168.1.1 is at 00:00:...	2496
18	1.015506	192.168.1.4	192.168.1.1	UDP	576	49153 → ftp-data(20) Len...	3072
21	1.022014	00:00:00_00:00:04	00:00:00_00:00:01	ARP	64	192.168.1.4 is at 00:00:...	3136
23	1.027218	192.168.1.1	192.168.1.4	UDP	576	ftp-data(20) → 49153 Len...	3712
25	3.003096	192.168.1.4	192.168.1.1	UDP	576	49153 → ftp-data(20) Len...	4288
27	3.008260	192.168.1.1	192.168.1.4	UDP	576	ftp-data(20) → 49153 Len...	4864

task1-off-2.pcap Pacchetti: 28 - visualizzati: 12 (42.9%) Profilo: Default

Immagine 3

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

In base ai pcap generati sul nodo “task1-off-2.pcap” (Immagine 1) nel caso RTS off, non si verificano mai collisioni. Osservando i pcap generati su tutti i nodi notiamo infatti che tutti “sentono” le conversazioni rispettivamente tra il client nodo 4 ed il

server 0 e tra il client nodo 3 ed il server 0, e che sia i 4 pacchetti totali che i 4 acknowledgment arrivano correttamente.

Inoltre applicando il filtro “(wlan.fc.retry == 1)” notiamo che non vengono evidenziate ritrasmissioni (Immagine 4).

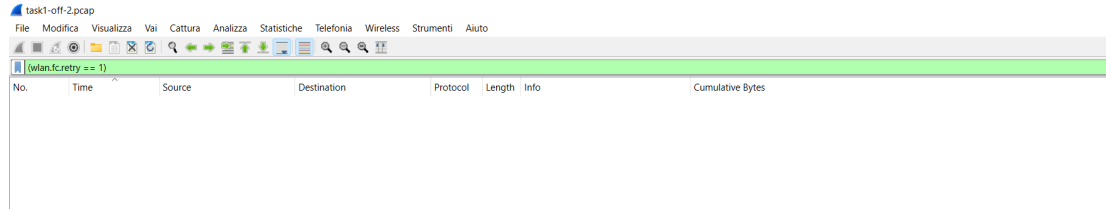


Immagine 4

Un’ipotesi che potrebbe spiegare l’assenza di collisioni è la distanza temporale con la quale i nodi inviano i propri pacchetti. Questi, infatti, eseguono carrier sensing sul canale e prima di inviare aspettano un minimo periodo di tempo denominato DIFS per i nodi che devono inviare un pacchetto e SIFS per i nodi che devono inviare un ACK.

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?

Per forzare i nodi ad utilizzare la procedura di handshake RTS/CTS su un’infrastruttura ad hoc ci sono due possibilità:

- Si aggiunge un master che coordina i nodi
- Quando un nodo vuole instaurare una comunicazione con un altro nodo invia un RTS al nodo destinatario, e comincerà a trasmettere solamente quando avrà ricevuto da esso un CTS.

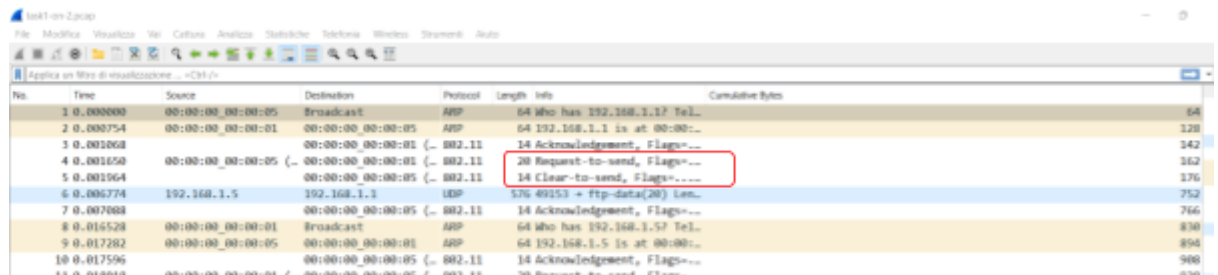
Il ragionamento dietro questa procedura prevede che un nodo, prima di inviare un pacchetto, invii una Request to Send sul canale per accertarsi che tutti siano avvisati dell’imminente trasmissione e aspetti di ricevere una Clear to Send prima di effettuare l’invio. Questa procedura rende possibile un coordinamento tra i nodi presenti su uno stesso canale. Infatti per abilitare o meno l’utilizzo dell’ handshake abbiamo abbassato (o alzato) opportunamente la soglia di dimensione massima del pacchetto per usufruire del servizio RTS/CTS. Tale servizio si attiva (on) quando la dimensione dei pacchetti che inviamo (nel nostro caso 512 byte escluso header) supera la soglia da noi impostata, ovvero 100 byte. Se non vogliamo che si attivi, è sufficiente alzare la soglia ad un valore opportunamente grande.

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

- Ci sono delle collisioni adesso? A maggior ragione, usando RTS/CTS non ci sono collisioni.

- Quali sono i benefici di RTS/CTS? Osservando i pcap generati su tutti i nodi (Immagine 5) notiamo che adesso, quando un nodo vuole instaurare una comunicazione, invia una Request to Send al destinatario e aspetta che questo gli

risponda con una Clear to Send. In questo modo tutti i nodi all'interno della rete sono a conoscenza della trasmissione in corso e quindi si evitano collisioni. Il beneficio principale che notiamo è il coordinamento tra i nodi. Naturalmente, il tutto è effettuato a costo di un tempo di esecuzione della comunicazione maggiore.

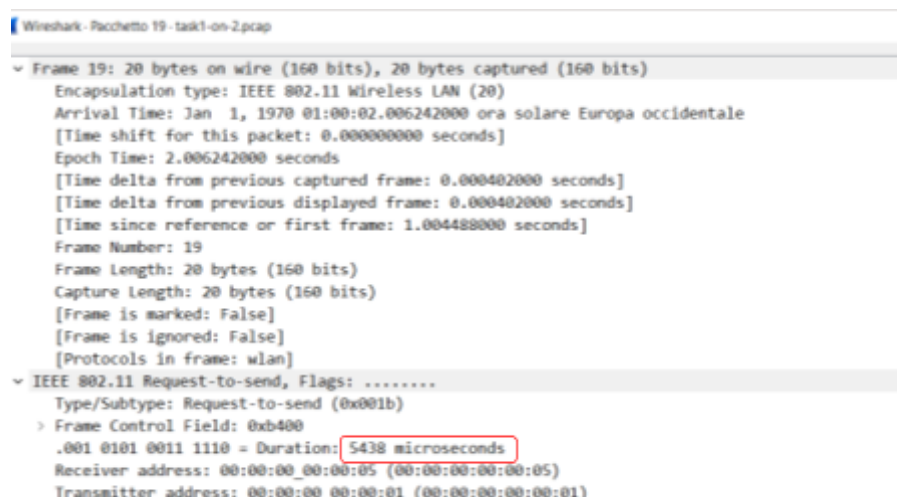


No.	Time	Source	Destination	Protocol	Length	Info	Cumulative Bytes
1	0.000000	00:00:00_00:00:05	Broadcast	ARP	64	64 Who has 192.168.1.1? Tel...	64
2	0.000754	00:00:00_00:00:01	00:00:00_00:00:05	ARP	64	64 192.168.1.1 is at 00:00:...	128
3	0.001068	00:00:00_00:00:01	00:00:00_00:00:05	802.11	14	14 Acknowledgment, Flags=...	142
4	0.001650	00:00:00_00:00:05	00:00:00_00:00:01	802.11	20	20 Request-to-send, Flags=...	162
5	0.001964	00:00:00_00:00:05	00:00:00_00:00:01	802.11	14	14 Clear-to-send, Flags=...	176
6	0.006774	192.168.1.5	192.168.1.1	UDP	576	576 49153 → 5555 [RST] Seq=...	752
7	0.007008	00:00:00_00:00:05	00:00:00_00:00:01	802.11	14	14 Acknowledgment, Flags=...	766
8	0.016528	00:00:00_00:00:01	Broadcast	ARP	64	64 Who has 192.168.1.5? Tel...	830
9	0.017282	00:00:00_00:00:05	00:00:00_00:00:01	ARP	64	64 192.168.1.5 is at 00:00:...	894
10	0.017596	00:00:00_00:00:05	00:00:00_00:00:01	802.11	14	14 Acknowledgment, Flags=...	908

Immagine 5

- Dove si può trovare ed analizzare le informazioni relative al Network Allocation Vector?

Il NAV si ricava dal parametro Duration dei frame MAC a livello 2. Possiamo vedere ad esempio che la Duration presente nella Request to Send effettuata dal nodo 0 e destinata al nodo 4 è pari a 5438 microsecondi (Immagine 6). Tutti i nodi che ricevono questa RTS e non sono i destinatari della comunicazione imposteranno il proprio NAV=Duration.



```

Wireshark - Pacchetto 19 - task1-on-2.pcap
▼ Frame 19: 20 bytes on wire (160 bits), 20 bytes captured (160 bits) on interface 0
  Encapsulation type: IEEE 802.11 Wireless LAN (20)
  Arrival Time: Jan 1, 1970 01:00:02.006242000 ora solare Europa occidentale
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 2.006242000 seconds
  [Time delta from previous captured frame: 0.000402000 seconds]
  [Time delta from previous displayed frame: 0.000402000 seconds]
  [Time since reference or first frame: 1.004488000 seconds]
  Frame Number: 19
  Frame Length: 20 bytes (160 bits)
  Capture Length: 20 bytes (160 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: wlan]
  ▼ IEEE 802.11 Request-to-send, Flags: .....
    Type/Subtype: Request-to-send (0x001b)
    > Frame Control Field: 0xb400
      .001 0101 0011 1110 = Duration: 5438 microseconds
    Receiver address: 00:00:00_00:00:05 (00:00:00:00:00:05)
    Transmitter address: 00:00:00_00:00:01 (00:00:00:00:00:01)
  
```

Immagine 6

5. Calcolare il throughput medio complessivo delle applicazioni

Ci sono diversi modi per calcolare il throughput.

Nel caso RTS off, se ad esempio consideriamo solamente il “goodput” in un intervallo di tempo che va dal tempo dell’invio del primo pacchetto (1,02002 sec, campionato sul nodo che effettua l’invio) al tempo di ricezione dell’ultimo (4,008260 sec,

campionato sul nodo che lo riceve) otteniamo $4096/3,006258 = 1.362,49$ byte per secondo medi.

Applichiamo la formula: $\frac{L}{\Delta t}$ e non consideriamo gli ack.

In alternativa, se non consideriamo il goodput ma tutti pacchetti che sono stati trasmessi sul canale durante la simulazione eseguiamo il calcolo: 5288 byte totali trasmessi / 3.00827 secondi = 1757 byte per secondo medi.

Applichiamo la formula: $\frac{H+L}{\Delta t}$ e consideriamo gli ack e messaggi broadcast.

Le stesse considerazioni possono essere effettuate anche nel caso di RTS on con la differenza che nel caso in cui consideriamo nel calcolo del TH tutti i pacchetti trasmessi sul canale, il risultato sarà maggiore a causa del maggior numero di pacchetti che attraversano il canale a parità di tempo.

TASK 2

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

L'AP presente nella rete si occupa di gestire le Association Request e di inviare periodicamente dei beacon frames a tutti i nodi presenti nella rete. Fin dal primo momento vediamo che l'AP invia dei beacon frames in broadcast per annunciare a tutti i nodi che sono all'interno di una rete wireless. Ogni comunicazione che avviene sulla rete ha come destinatario MAC l'access point, che poi si occupa di ritrasmettere il pacchetto all'IP destinatario. Questo giustifica il fatto che, guardando il pcap (Immagine 7), ogni pacchetto UDP sembrerebbe essere stato inviato due volte: in realtà la prima trasmissione è tra la sorgente e l'AP, mentre la seconda è la ritrasmissione che l'AP fa verso il destinatario. Questo si può verificare osservando le informazioni presenti in questi pacchetti (Immagine 8) che evidenziano il fatto che il receiver del pacchetto è l'indirizzo MAC dell'AP, mentre il destinatario effettivo è l'indirizzo MAC del nodo 0.

No.	Time	Source	Destination	Protocol	Length	Info	Cumulative bytes
28	0.716789	00:00:00_00:00:06	Broadcast	802.11	71	Beacon frame, Sh=12, Fh=...	
29	0.819189	00:00:00_00:00:06	Broadcast	802.11	71	Beacon frame, Sh=13, Fh=...	
30	0.905122	00:00:00_00:00:05	Broadcast	ARP	64	who has 192.168.1.1? Tel...	
31	0.905436	00:00:00_00:00:05	00:00:00_00:00:05 (... 802.11	802.11	14	Acknowledgement, Flags=...	
32	0.906168	00:00:00_00:00:05	Broadcast	ARP	64	who has 192.168.1.1? Tel...	
33	0.906500	00:00:00_00:00:01	00:00:00_00:00:05	ARP	64	192.168.1.1 is at 00:00:...	
34	0.907214	00:00:00_00:00:01	00:00:00_00:00:01 (... 802.11	802.11	14	Acknowledgement, Flags=...	
35	0.907582	00:00:00_00:00:01	00:00:00_00:00:05	ARP	64	192.168.1.1 is at 00:00:...	
36	0.908298	00:00:00_00:00:06	00:00:00_00:00:06 (... 802.11	802.11	14	Acknowledgement, Flags=...	
37	0.913124	192.168.1.5	192.168.1.1	UDP	576	Ftp(21) → 49153 len=512	
38	0.913438	00:00:00_00:00:05 (... 802.11	192.168.1.1	802.11	14	Acknowledgement, Flags=...	
39	0.918592	192.168.1.5	192.168.1.1	UDP	576	Ftp(21) → 49153 len=512	
40	0.918706	00:00:00_00:00:06 (... 802.11	192.168.1.1	802.11	14	Acknowledgement, Flags=...	
41	0.920124	00:00:00_00:00:01	Broadcast	ARP	64	who has 192.168.1.5? Tel...	
42	0.920438	00:00:00_00:00:01	00:00:00_00:00:01 (... 802.11	802.11	14	Acknowledgement, Flags=...	
43	0.921170	00:00:00_00:00:01	Broadcast	ARP	64	who has 192.168.1.5? Tel...	
44	0.921049	00:00:00_00:00:06	Broadcast	802.11	71	Beacon frame, Sh=18, Fh=...	
45	0.922681	00:00:00_00:00:05	00:00:00_00:00:01	ARP	64	192.168.1.5 is at 00:00:...	
46	0.922995	00:00:00_00:00:05	00:00:00_00:00:05 (... 802.11	802.11	14	Acknowledgement, Flags=...	
47	0.923727	00:00:00_00:00:05	00:00:00_00:00:01	ARP	64	192.168.1.5 is at 00:00:...	
48	0.924661	00:00:00_00:00:06 (... 802.11	192.168.1.5	802.11	14	Acknowledgement, Flags=...	
49	0.928878	192.168.1.1	192.168.1.5	UDP	576	Ftp(21) → 49153 len=512	
50	0.929192	00:00:00_00:00:01 (... 802.11	192.168.1.5	802.11	14	Acknowledgement, Flags=...	
51	0.934805	192.168.1.1	192.168.1.5	UDP	576	Ftp(21) → 49153 len=512	
52	0.934579	00:00:00_00:00:06 (... 802.11	192.168.1.5	802.11	14	Acknowledgement, Flags=...	
53	1.023589	00:00:00_00:00:06	Broadcast	802.11	71	Beacon frame, Sh=21, Fh=...	
54	1.126309	00:00:00_00:00:06	Broadcast	802.11	71	Beacon frame, Sh=22, Fh=...	
55	1.228709	00:00:00_00:00:06	Broadcast	802.11	71	Beacon frame, Sh=23, Fh=...	
56	1.331189	00:00:00_00:00:06	Broadcast	802.11	71	Beacon frame, Sh=24, Fh=...	
57	1.433589	00:00:00_00:00:06	Broadcast	802.11	71	Beacon frame, Sh=25, Fh=...	
58	1.535989	00:00:00_00:00:06	Broadcast	802.11	71	Beacon frame, Sh=26, Fh=...	
59	1.638389	00:00:00_00:00:06	Broadcast	802.11	71	Beacon frame, Sh=27, Fh=...	
60	1.740789	00:00:00_00:00:06	Broadcast	802.11	71	Beacon frame, Sh=28, Fh=...	

Immagine 7

```

Frame Length: 576 bytes (4608 bits)
Capture Length: 576 bytes (4608 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: wlan:llc:ip:udp:data]
[Coloring Rule Name: UDP]
[Coloring Rule String: udp]

IEEE 802.11 Data, Flags: 00000000000000000000000000000000
Type/Subtype: Data (0x0020)
> Frame Control Field: 0x0001
  .0000 0001 0011 1010 = Duration: 314 microseconds
  Receiver address: 00:00:00_00:00:06 (00:00:00:00:00:06)
  Transmitter address: 00:00:00_00:00:05 (00:00:00:00:00:05)
  Destination address: 00:00:00_00:00:01 (00:00:00:00:00:01)
  Source address: 00:00:00_00:00:05 (00:00:00:00:00:05)
  BSS Id: 00:00:00_00:00:06 (00:00:00:00:00:06)
  STA address: 00:00:00_00:00:05 (00:00:00:00:00:05)
  .... 0000 = Fragment number: 0
  0000 0000 0010 .... = Sequence number: 2

Logical-Link Control
> DSAP: SNAP (0xaa)
> SSAP: SNAP (0xaa)
> Control field: U, func=UI (0x03)
  Organization Code: 00:00:00 (Officially Xerox, but
  Type: IPv4 (0x0800)

Internet Protocol Version 4, Src: 192.168.1.5 (192.168.1.5), Dst: 192.168.1.1 (192.168.1.1)

```

Immagine 8

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

Per evidenziare i beacon frames su Wireshark applichiamo il filtro

“*wlan.fc.type_subtype == 0x0008*” al pcap generato sull’AP. Le informazioni più rilevanti che contiene sono:

- Timestamp: consente la sincronizzazione tra tutti i nodi; in questo caso vuole che tutti i nodi settino come istante attuale il 143776 (Immagine 9).
- Beacon Interval: specifica l’intervallo di tempo tra due successivi beacon frames inviati dall’ AP; in questo caso il beacon frame successivo arriverà dopo 0,102400 secondi (ne abbiamo conferma nell’immagine 10)

- Capability information: contiene informazioni riguardo la capacità della rete, il tipo di network e informazioni riguardanti la privacy. Inoltre, è possibile notare tra i campi significativi la QoS (Quality of Service), che come da specifica non è abilitata.
- SSID: service set identifier usato per identificare una particolare rete wifi. In questo caso vediamo che il suo identifier è pari alla somma delle nostre matricole: 7753193 (Immagine 11).

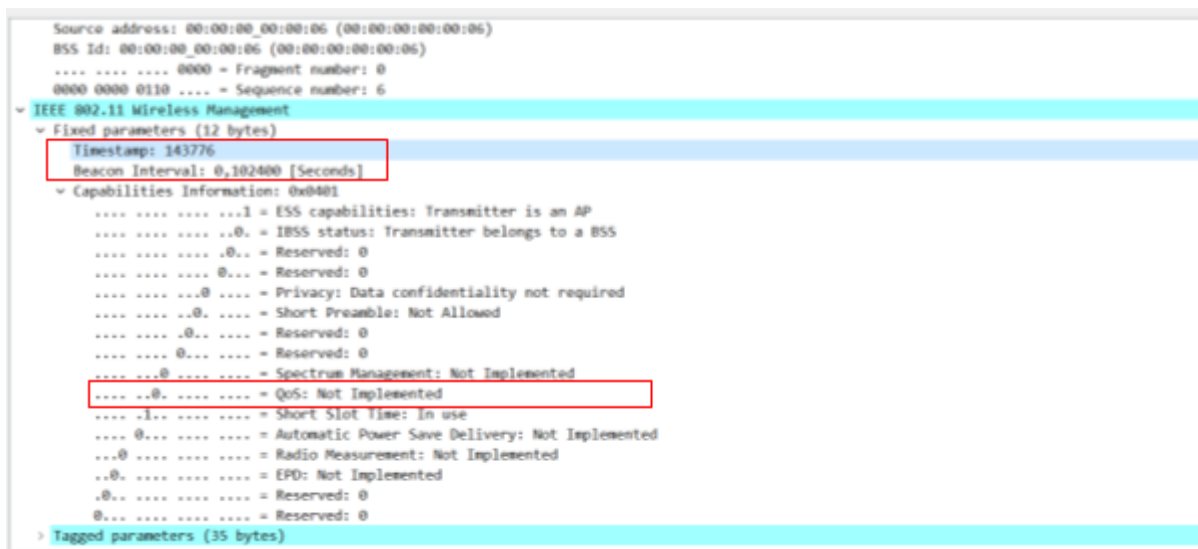


Immagine 9

No.	Time	Source	Destination	Protocol	Length	Info	Cumulative Bytes
1	0.000000	00:00:00:00:00:00	Broadcast	802.11	71	71 Beacon frame, SM=0, FN=0...	71
22	0.102389	00:00:00:00:00:00	Broadcast	802.11	71	71 Beacon frame, SM=0, FN=0...	142
23	0.204789	00:00:00:00:00:00	Broadcast	802.11	71	71 Beacon frame, SM=0, FN=0...	213

Immagine 10

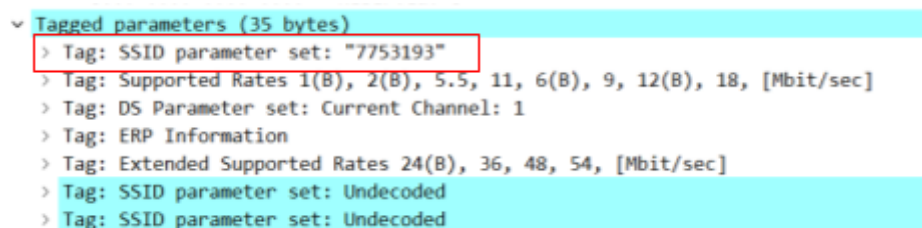


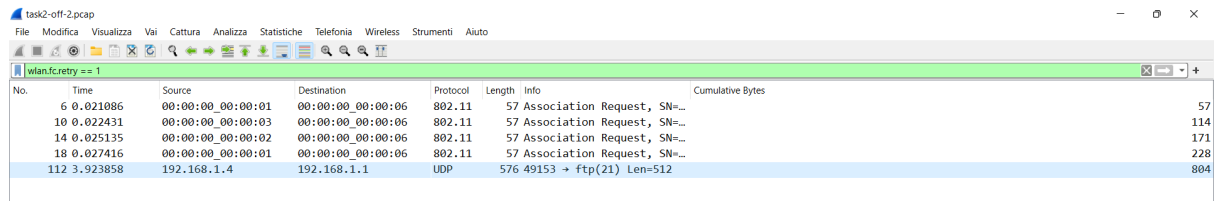
Immagine 11

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.

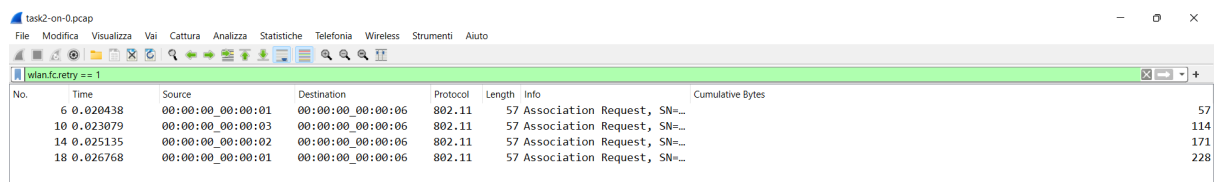
Senza l'utilizzo di Rts/Cts possiamo notare ben 5 collisioni (Immagine 12). In particolare applicando il filtro "wlan.fc.retry == 1" notiamo che i nodi 0, 1 e 2 devono eseguire delle ritrasmissioni di Association Request mentre il nodo 3 esegue la ritrasmissione di un pacchetto UDP. Applicando la funzione useRtsCts notiamo

invece che la ritrasmissione del pacchetto UDP non si verifica più, mentre le Association Request collidono come nel caso precedente (Immagine 13). Questo succede perché le Association Request non inviano un Clear to Send prima di inviare il proprio pacchetto e quindi sono ancora soggette a collisioni. Per il pacchetto UDP invece adesso, non appena il nodo 3 invia la sua Request to Send, gli altri sono a conoscenza della comunicazione in corso sulla rete e si mettono in stand-by. Questi infatti settano il proprio NAV=Duration e ritentano l'invio successivamente.



No.	Time	Source	Destination	Protocol	Length	Info	Cumulative Bytes
6	0.021086	00:00:00_00:00:01	00:00:00_00:00:06	802.11	57	Association Request, SN=...	57
10	0.022431	00:00:00_00:00:03	00:00:00_00:00:06	802.11	57	Association Request, SN=...	114
14	0.025135	00:00:00_00:00:02	00:00:00_00:00:06	802.11	57	Association Request, SN=...	171
18	0.027416	00:00:00_00:00:01	00:00:00_00:00:06	802.11	57	Association Request, SN=...	228
112	3.923858	192.168.1.4	192.168.1.1	UDP	576	49153 → ftp(21) Len=512	804

Immagine 12



No.	Time	Source	Destination	Protocol	Length	Info	Cumulative Bytes
6	0.020438	00:00:00_00:00:01	00:00:00_00:00:06	802.11	57	Association Request, SN=...	57
10	0.023079	00:00:00_00:00:03	00:00:00_00:00:06	802.11	57	Association Request, SN=...	114
14	0.025135	00:00:00_00:00:02	00:00:00_00:00:06	802.11	57	Association Request, SN=...	171
18	0.026768	00:00:00_00:00:01	00:00:00_00:00:06	802.11	57	Association Request, SN=...	228

Immagine 13

NOTA: Il fatto che avvenga una sola ritrasmissione nel caso della collisione tra i due pacchetti udp (nodo 3 e 4) potrebbe essere motivato dalla minore distanza tra il nodo 4 e l'AP, infatti il pacchetto riesce a raggiungere quest'ultimo prima che avvenga la collisione qualche istante dopo (Immagine 14).

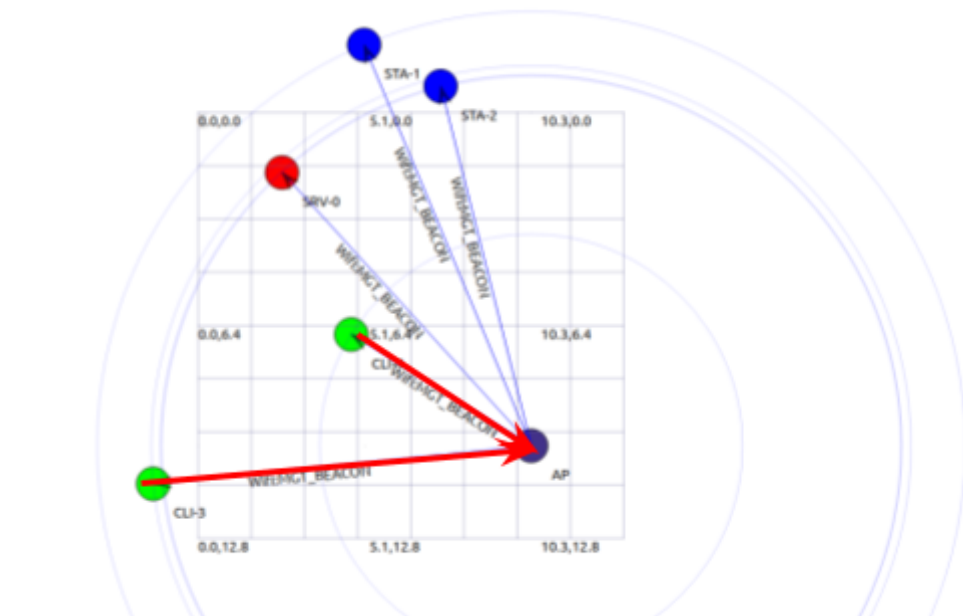


Immagine 14