

REPORT TEAM 2 - HOMEWORK 3:

Camilla Santoro - Emanuele Mule - Simone Palumbo - Isidoro Tamassia

Disclaimer: se non diversamente specificato i pcap considerati nelle analisi sono, per ogni configurazione, quelli generati sul nodo n1 della CSMA IP Block: 192.138.1.0 e sul nodo n3 dell'interfaccia I0.

CONFIGURAZIONE 0:

A1) Individuare le varie topologie note che compongono la rete:

La struttura della rete è costituita da una serie di point-to-point e da dei bus chiamati CSMA:

- 4 interfacce ptp. Ognuna di esse rappresenta l'astrazione di un collegamento diretto tra due nodi, ovvero senza alcun nodo intermedio tra di loro.
- 2 bus CSMA. Tale struttura si presta al passaggio di pacchetti, veicolati tramite tale bus, tra un nodo N_v ed una lista di nodi $\{N_1, N_2, \dots, N_n\}$. N_v è collegato ad uno dei nodi presenti nel bus tramite un collegamento, ma tali pacchetti possono poi essere passati a tutti i nodi della lista.

La topologia della rete è di tipo albero (Immagine 1): n3 è la radice e le interfacce I0 e I1 sono i rami che connettono la radice ai figli, ovvero le due CSMA. In particolare la CSMA con IP Block: 192.138.1.0 non ha figli, mentre quella con IP Block: 192.138.2.0 ha figlio sinistro n5, a sua volta n5 ha figlio n4. Le foglie dell'albero sono la prima CSMA e il nodo n4.

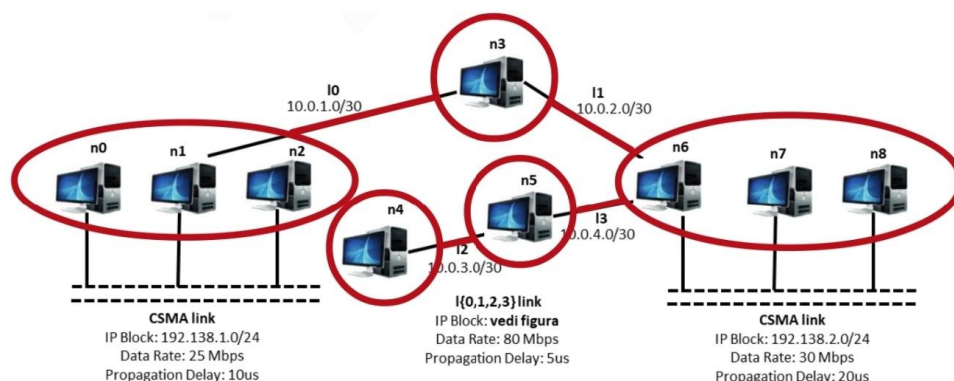


Immagine 1: Visualizzazione della struttura ad albero della rete

A2) Ricostruzione del percorso dei pacchetti attraverso la rete di tutti i flussi simulati usando wireshark evidenziando i filtri utilizzati per isolare i singoli flussi dello strato di trasporto tra le tracce:

In questa configurazione i pacchetti vanno dal host n4 al host n2 e percorrono la rete attraversando in ordine le seguenti interfacce: n5->n6->n3->n1. Confermiamo con Wireshark: sapendo che gli indirizzi IP di n4 e n2 sono rispettivamente “10.0.3.1” e “192.138.1.3” possiamo applicare il filtro “*ip.addr==10.0.3.1 && ip.addr==192.138.1.3*” sui pcap generati dai nodi non terminali n5, n6, n3, n1 e vedere che il numero di pacchetti intercettati è sempre lo stesso, 2252 pacchetti. Questo conferma l’evidenza grafica, dato che non esistono possibili percorsi alternativi che i pacchetti potrebbero intraprendere per arrivare da tale sorgente a tale destinazione. Inoltre si può vedere come i pcap generati su n7 e n8, non facendo parte del percorso che intercorre tra i nodi n4 e n2, non contengono alcun pacchetto. A differenza di questi ultimi, il nodo n0, essendo soggetto al protocollo ARP della CSMA, contiene due messaggi ARP.

A3) Calcolo e grafico di round trip time (RTT) e commento:

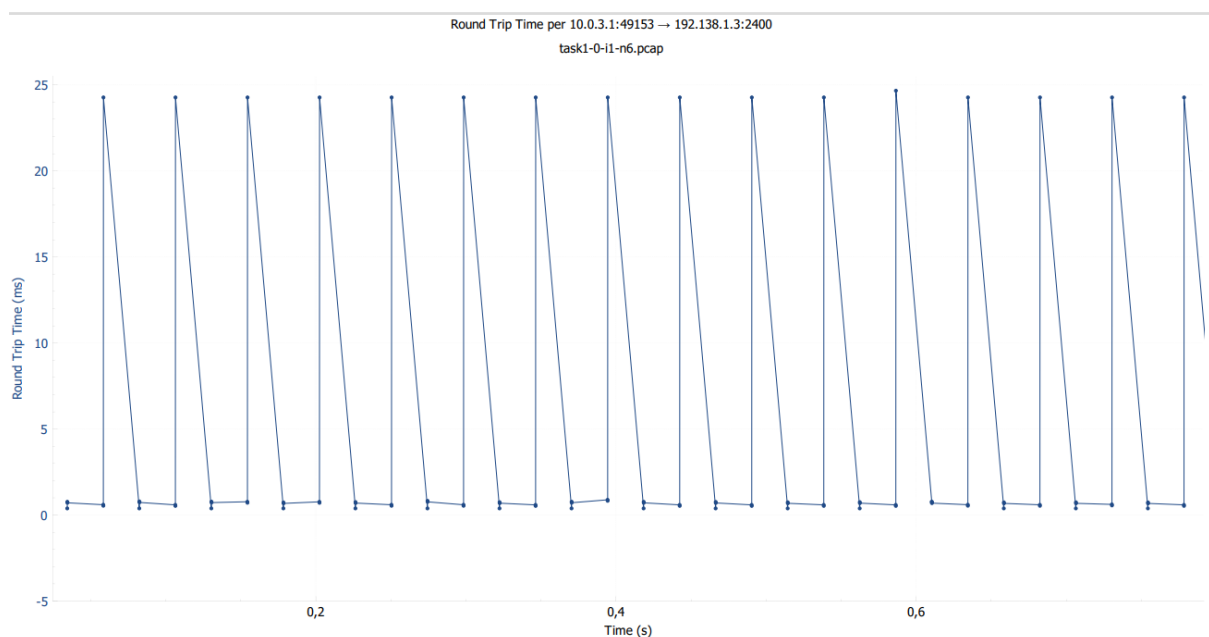


Immagine 2 - Grafico RTT

Dalla schermata principale di WS, è possibile, andando su Statistiche -> Grafici dei Flussi TCP -> Round Trip Time -> Inverti direzione, visualizzare il grafico dell’ RTT (Immagine 2). Premendo ripetutamente x si allarga l’asse x permettendo una

visualizzazione come quella in Immagine 2. Si può vedere che i picchi dell' RTT sono di circa 24 ms. Il grafico dell' RTT denota inoltre il comportamento che possiamo vedere nel grafico del flusso (Immagine 3) relativo al protocollo TCP (andando su Statistiche -> Grafico del Flusso -> Tipo di Flusso: TCP flow), in quanto la rete ha un approccio conservativo per cui a tre pacchetti inviati corrisponde un unico ACK cumulativo.

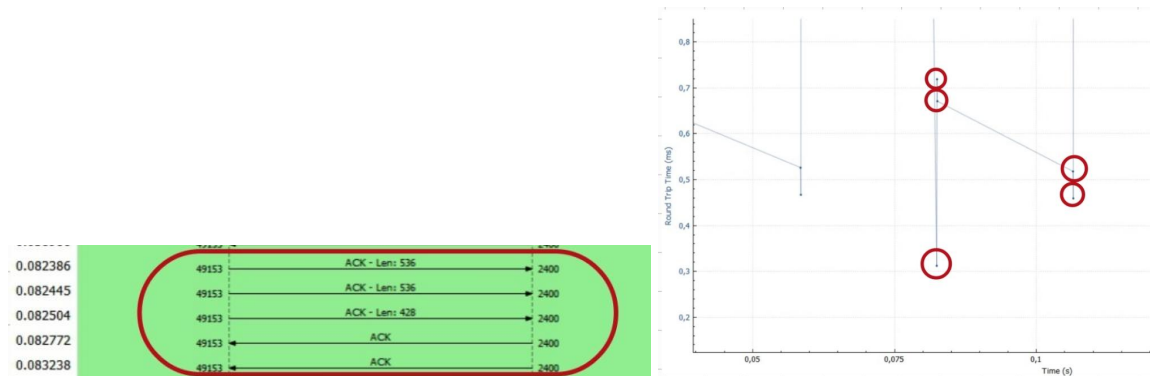


Immagine 3 - Confronto tra pacchetti sul grafico RTT e stessi pacchetti sul grafico TCP Flow

A4) Vi sono dei bottleneck nella rete? Se si, individuare gli eventuali link e discutere eventuali contromisure e soluzioni:

Dato che in ns3 il valore di default del data-rate dei client on-off è 0.5 Mbps, questi influiscono totalmente sul throughput di tutte e tre le configurazioni. Nella CSMA la dimensione media dei pacchetti aumenta e con lei il bitrate medio; questo è visibile dalle statistiche di WS. Per client aventi un data-rate di ordine maggiore le point-to-point riuscirebbero a sostenere il carico di rete purché si stia sotto gli 80 Mbps di traffico complessivo, mentre le due CSMA, a causa delle loro capacità, non ci riuscirebbero, risultando quindi in un *bottleneck*. Le soluzioni per il bottleneck sono molteplici, ma in questa topologia e con queste configurazioni l'unica possibile è aumentare il data-rate del client, in questa maniera il throughput non verrà limitato dal client. Altre possibili soluzioni, in generale, sono:

- installare un server proxy (dipende dal servizio fornito, es. HTTP)
- aumentare la capacità delle interfacce problematiche (es. CSMA)
- fornire un meccanismo di routing per massimizzare il throughput

Wireshark - Proprietà del file di cattura - task1-0-csma1-n2.pcap

Dettagli

File

Nome:

H:\8 mio Drive\III Anno I Semestre\Telecomunicazioni\homework\Cartelle .pcap e .tr\Cartelle all\Configurazione 0-all\task1-0-csma1-n2.pcap

Lunghezza:

942 kB

Hash (SHA256):

99f38991a66815c47b5a273bfa655d0581864b358384b174c21480c9

Hash (RMD160):

c1292c81314a7af8a445252a7b8f7ed426a2

Hash (SHA1):

27ced1a86a3995675b714c2ed5b780119ba

Formato:

Wireshark/tcpdump/... - pcap

Incap슐amento:

Ethernet

Lunghezza istantanea:

65535

Tempo

Primo pacchetto:

1970-01-01 01:00:03

Ultimo pacchetto:

1970-01-01 01:00:15

Trascorso:

00:00:11

Cattura

Hardware:

Sconosciuto

OS:

Sconosciuto

Applicazione:

Sconosciuto

Interfaccia

Interfaccia

Pacchetti persi

Filtro di cattura

Tipo di link

Dimensione limite del pacchetto (snaplen)

Sconosciuto

Sconosciuto

Sconosciuto

Ethernet

65535 byte

Statistiche

Misure

Catturati

Visualizzati

Marcati

Pacchetti

2258

2258 (100.0%)

—

Tempo, s

11.996

11.996

—

PPS medi

188.1

188.1

—

Dimensione media dei pacchetti, B

402

402

—

Byte

90404

90404 (100.0%)

0

Byte/s medi

75 k

75 k

—

Bit/s medi

604 k

604 k

—

Wireshark - Proprietà del file di cattura - task1-0-i1-n3.pcap

Dettagli

File

Nome:

H:\8 mio Drive\III Anno I Semestre\Telecomunicazioni\homework\Cartelle .pcap e .tr\Cartelle all\Configurazione 0-all\task1-0-i1-n3.pcap

Lunghezza:

906 kB

Hash (SHA256):

3d3c3d6b9423ab9f36224edfcb8314edc7b0e11747753ce3ad6f340c

Hash (RMD160):

a88537e3d01da59450e4218a05e59c8b483

Hash (SHA1):

a8bda48644046bda6e4402e8183dca235383

Formato:

Wireshark/tcpdump/... - pcap

Incap슐amento:

PPP

Lunghezza istantanea:

65535

Tempo

Primo pacchetto:

1970-01-01 01:00:03

Ultimo pacchetto:

1970-01-01 01:00:15

Trascorso:

00:00:12

Cattura

Hardware:

Sconosciuto

OS:

Sconosciuto

Applicazione:

Sconosciuto

Interfaccia

Interfaccia

Pacchetti persi

Filtro di cattura

Tipo di link

Dimensione limite del pacchetto (snaplen)

Sconosciuto

Sconosciuto

Sconosciuto

PPP

65535 byte

Statistiche

Misure

Catturati

Visualizzati

Marcati

Pacchetti

2252

2252 (100.0%)

—

Tempo, s

12.000

12.000

—

PPS medi

187.7

187.7

—

Dimensione media dei pacchetti, B

386

386

—

Byte

870116

870116 (100.0%)

0

Byte/s medi

72 k

72 k

—

Bit/s medi

580 k

580 k

—

Immagine 4 - Throughput medio configurazione 0

Wireshark - Proprietà del file di cattura - task1-1-csma1-n1.pcap

Dettagli

File

Nome:

H:\8 mio Drive\III Anno I Semestre\Telecomunicazioni\homework\Cartelle .pcap e .tr\Cartelle all\Configurazione 1-all\task1-1-csma1-n1.pcap

Lunghezza:

1328 kB

Hash (SHA256):

680c578f44da278e8dc0bb227bc9f68705853f91a5afeb347a8Secid3ae9b

Hash (RMD160):

e206ec14e5278fec052900464c0a84e73157c0d

Hash (SHA1):

cb01e0b217a8123a7c8e11a0f9a9c9c621b94d

Formato:

Wireshark/tcpdump/... - pcap

Incap슐amento:

Ethernet

Lunghezza istantanea:

65535

Tempo

Primo pacchetto:

1970-01-01 01:00:02

Ultimo pacchetto:

1970-01-01 01:00:15

Trascorso:

00:00:12

Cattura

Hardware:

Sconosciuto

OS:

Sconosciuto

Applicazione:

Sconosciuto

Interfaccia

Interfaccia

Pacchetti persi

Filtro di cattura

Tipo di link

Dimensione limite del pacchetto (snaplen)

Sconosciuto

Sconosciuto

Sconosciuto

Ethernet

65535 byte

Statistiche

Misure

Catturati

Visualizzati

Marcati

Pacchetti

3184

3184 (100.0%)

—

Tempo, s

12.989

12.989

—

PPS medi

245.1

245.1

—

Dimensione media dei pacchetti, B

401

401

—

Byte

1277348

1277348 (100.0%)

0

Byte/s medi

98 k

98 k

—

Bit/s medi

786 k

786 k

—

Wireshark - Proprietà del file di cattura - task1-1-i0-n3.pcap

Dettagli

File

Nome:

H:\8 mio Drive\III Anno I Semestre\Telecomunicazioni\homework\Cartelle .pcap e .tr\Cartelle all\Configurazione 1-all\task1-1-i0-n3.pcap

Lunghezza:

1276 kB

Hash (SHA256):

b590cd3f56f66bb4eba4cab772ec7581cd0bb6220196c6e9db78908c9ee03

Hash (RMD160):

a5c618a0b36987dc557ca3ba8544132ecd8b7d1

Hash (SHA1):

7ba2e798baa1a1c13d4e8d77c7b7d5fdcd461e6a

Formato:

Wireshark/tcpdump/... - pcap

Incap슐amento:

PPP

Lunghezza istantanea:

65535

Tempo

Primo pacchetto:

1970-01-01 01:00:02

Ultimo pacchetto:

1970-01-01 01:00:15

Trascorso:

00:00:12

Cattura

Hardware:

Sconosciuto

OS:

Sconosciuto

Applicazione:

Sconosciuto

Interfaccia

Interfaccia

Pacchetti persi

Filtro di cattura

Tipo di link

Dimensione limite del pacchetto (snaplen)

Sconosciuto

Sconosciuto

Sconosciuto

PPP

65535 byte

Statistiche

Misure

Catturati

Visualizzati

Marcati

Pacchetti

3176

3176 (100.0%)

—

Tempo, s

12.993

12.993

—

PPS medi

244.4

244.4

—

Dimensione media dei pacchetti, B

386

386

—

Byte

1226020

1226020 (100.0%)

0

Byte/s medi

94 k

94 k

—

Bit/s medi

754 k

754 k

—

Immagine 5 - Throughput medio configurazione 1

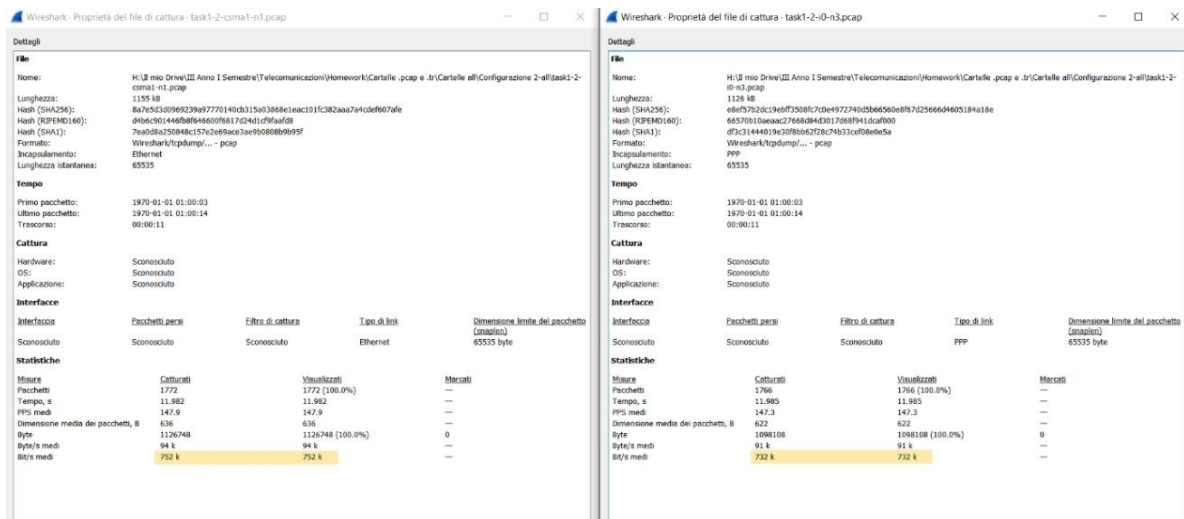


Immagine 6 - Throughput medio configurazione 2

C01) Calcolare il throughput istantaneo del flusso TCP:

Per il calcolo del throughput istantaneo possiamo utilizzare Wireshark, in particolare andremo a selezionare Statistiche->I/O Graphs, selezionando come asse y bit/s.

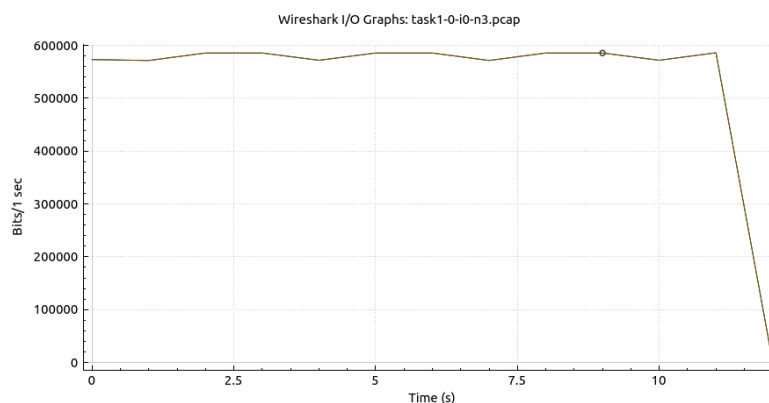


Immagine 7 - Grafico I/O dell'interfaccia i0

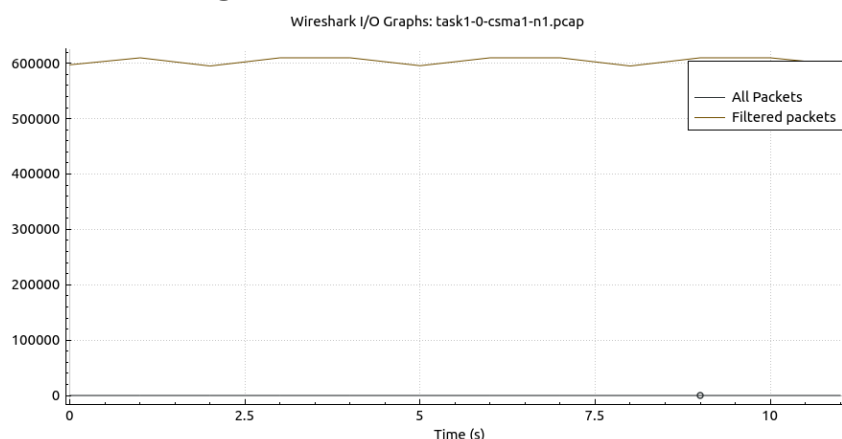


Immagine 8 - Grafico I/O della CSMA

Possiamo dunque verificare immediatamente che il throughput è più elevato sulla interfaccia CSMA (nonostante la capacità ridotta), questo è dovuto al fatto che, seppur

il traffico sia lo stesso, i segmenti che giungono alla CSMA avranno dimensione maggiore (header maggiori), di conseguenza il numero complessivo di bit sulla interfaccia CSMA sarà complessivamente maggiore, e ciò si ripercuoterà su tutto il report.

Questa ipotesi viene confermata dai calcoli effettuati sui pacchetti trasmessi in un intervallo sul nodo n1 della CSMA. Preso ad esempio il pacchetto con SEQ=3537 possiamo vedere che:

$$\text{cumulative bytes} * 8 / \text{Time} = 5316 * 8 / 0.078690 = 539180,775 \text{ bit/s.}$$

C02) Calcolare il throughput medio del flusso TCP a tempo t=4.0s.:

Bisogna considerare che Wireshark lavora con un tempo relativo (TRel) mentre noi dobbiamo rispettare richieste in tempo assoluto (TAss), quello della simulazione.

Dunque al secondo t=4, per WS sarà passato solo un secondo. Con queste premesse, possiamo andare ad applicare il filtro wireshark “tcp.time_relative<=1”, e mediante l’ausilio della colonna “cumulative bytes”(Immagine 9), calcoliamo il throughput medio. La somma dei bytes dei pacchetti inviati nell’intervallo temporale scelto è di 74598 bytes (basterà andare a vedere i cumulative bytes relativi all’ultimo pacchetto filtrato), dunque il Throughput medio sarà pari a $74598 * 8 / 0.991163 = 602669 \text{ bit/s}$. Da notare che nelle immagini 9 e 10, ed equivalenti a seguire, i pacchetti “**Visualizzati**” sono quelli considerati tramite i filtri. Questo calcolo considera anche gli header. Inoltre tra l’interfaccia CSMA e le altre interfacce di interesse vi è una differenza di throughput dovuta all’aumento della dimensione dei pacchetti nella CSMA. In particolare applicando lo stesso calcolo sulla interfaccia 0 ad esempio, otteniamo un throughput medio di $71660 * 8 / 0.995185 = 576053 \text{ bit/s}$.

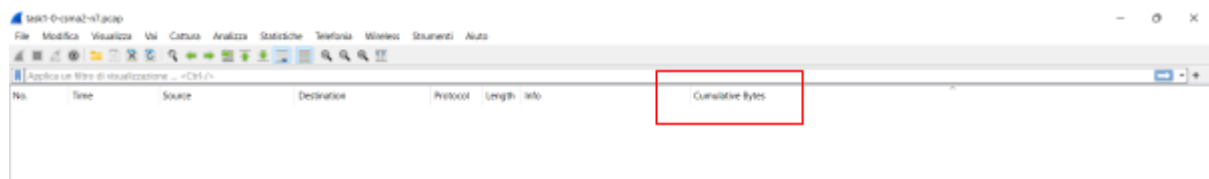


Immagine 9 - Cumulative bytes

Statistiche			
Misure	Catturati	Visualizzati	Marcati
Pacchetti	2256	188 (8.3%)	—
Tempo, s	11.996	0.991	—
PPS medi	188.1	189.7	—
Dimensione media dei pacchetti, 402 B		397	—
Byte	906404	74668 (8.2%)	0
Byte/s medi	75 k	75 k	—
Bit/s medi	604 k	602 k	—

Immagine 10 - Throughput Medio sulla CSMA

Statistiche			
Misure	Catturati	Visualizzati	Marcati
Pacchetti	2252	188 (8.3%)	—
Tempo, s	12.000	0.995	—
PPS medi	187.7	188.9	—
Dimensione media dei pacchetti, B	386	381	—
Byte	870116	71660 (8.2%)	0
Byte/s medi	72 k	72 k	—
Bit/s medi	580 k	576 k	—

Immagine 11 - Throughput Medio sull'interfaccia i0

C03) Calcolare il throughput medio del flusso TCP a tempo $t=7.0s$. Commentare eventuali cambiamenti rispetto a C02:

Come prima, valgono le considerazioni fatte su tempo assoluto e relativo, dunque al secondo $t=7$ per WS saranno passati solo 4 secondi. Basterà dunque applicare il filtro “tcp.time_relative ≤ 4 ” e andare a controllare il valore di cumulative bytes per l'ultimo pacchetto dell'intervallo. In questo caso la somma dei bytes sarà, nella CSMA, 301508 bytes e dunque il TH medio sarà pari a $301508 \cdot 8 / 3.990941 = 604384$ bit/s contro i 579016 bit/s dell'interfaccia 0. Il cambiamento del valore medio rispetto a C02 è dovuto al fatto che al tempo $t=7s$ la trasmissione è a regime e dunque tende ai valori limite riportati sopra (vedi Immagine 7 e Immagine 8).

Statistiche			
Misure	Catturati	Visualizzati	Marcati
Pacchetti	2256	750 (33.2%)	—
Tempo, s	11.996	3.991	—
PPS medi	188.1	187.9	—
Dimensione media dei pacchetti, B	402	402	—
Byte	906404	301508 (33.3%)	0
Byte/s medi	75 k	75 k	—
Bit/s medi	604 k	604 k	—

Immagine 12 - Throughput Medio sul CSMA

Statistiche			
Misure	Catturati	Visualizzati	Marcati
Pacchetti	2252	750 (33.3%)	—
Tempo, s	12.000	3.995	—
PPS medi	187.7	187.7	—
Dimensione media dei pacchetti, B	386	386	—
Byte	870116	289508 (33.3%)	0
Byte/s medi	72 k	72 k	—
Bit/s medi	580 k	579 k	—

Immagine 13 - Throughput Medio sull'interfaccia i0

C04) Calcolare il ritardo di trasferimento complessivo di tutti i pacchetti inviati:
Ipotizzando che il ritardo di accodamento ed elaborazione sia 0

$$D_{MES} = d_{prop,AB} + \frac{H+L}{C} \cdot N + \frac{1}{C} \left\{ \left\lceil \frac{X}{L} \right\rceil \cdot (H+L) \right\} - \frac{H+L}{C}$$

Immagine 14 - Formula delay messaggio che abbiamo generalizzato

Per calcolare il ritardo complessivo di tutti i pacchetti (compresi gli ack) consideriamo i ritardi di trasmissione e di propagazione sulle diverse interfacce attraversate dal flusso. La dimensione dei pacchetti è decisa dal protocollo TCP in quanto un pacchetto di 1500 byte viene segmentato in tre segmenti poiché supera l'MTU. A questo punto, mediante WS, possiamo individuare che diverse dimensioni hanno i pacchetti sulle interfacce I0, I1, I2, I3: 58, 54, 590, 482. Si può calcolare, tramite WS, qual è il numero di pacchetti per ognuna di queste dimensioni e, avendo capacità specificata costante (80 Mbps) e tempo di propagazione definito (5 microsecondi per interfaccia), allora possiamo calcolare il ritardo end-to-end tra n4 ed n1, mediante la formula sopra (Immagine 9). A quest'ultimo, andrà aggiunto il ritardo relativo alla CSMA. I pacchetti sopra citati, nel momento in cui attraversano la CSMA, assumono dimensioni maggiori, precisamente: 64, 74, 606, 70, 498 (si considerano anche i pacchetti ARP). Come prima, calcolando mediante WS il numero di pacchetti per ognuna di queste dimensioni, ma considerando questa volta capacità e ritardo di propagazione diversi (quelli relativi alla CSMA) si ottiene il ritardo tra n1 e n2. Sommando il delay end-to-end tra n4 e n1 e quello tra n1 e n2 otteniamo il ritardo richiesto. Il valore numerico ottenuto è 0,37745s.

Utilizzando il valore medio come stima della dimensione dei pacchetti, lo stesso calcolo ci avrebbe restituito il valore 0,3652s con un errore quindi di $0,37745 - 0,3652 = 0,01225$.

In fondo al pdf abbiamo riportato i calcoli relativi a questa risposta, per completezza.

NOTA:

In un primo momento potrebbe venire in mente di considerare il ritardo di trasferimento complessivo come la differenza tra l'istante relativo alla ricezione dell'ultimo pacchetto inviato e l'invio del primo, tale approccio tuttavia non sarebbe corretto perchè in questa simulazione il numero di pacchetti inviati (e quindi la dimensione del messaggio) dipendono direttamente dall'intervallo di tempo in cui il

client rimane acceso, dunque il calcolo sopracitato non calcolerebbe il ritardo di trasmissione dei pacchetti complessivo ma solo l'intervallo di tempo in cui questi sono stati inviati. Tale approccio sarebbe corretto se il messaggio da inviare avesse una dimensione fissata e non dipendesse dal tempo in cui il client rimane acceso.

CONFIGURAZIONE 1:

A1) Individuare le varie topologie note che compongono la rete:

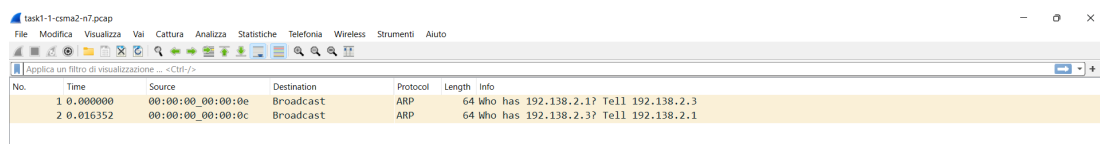
Essendo la risposta a questa domanda indipendente dalla configurazione, vale la risposta già riportata nella A1 della configurazione 0 (vedi sopra).

A2) Ricostruzione del percorso dei pacchetti attraverso la rete di tutti i flussi simulati usando wireshark evidenziando i filtri utilizzati per isolare i singoli flussi dello strato di trasporto tra le tracce:

In questa configurazione abbiamo due flussi distinti: nel primo ($n4 \rightarrow n0$), i pacchetti attraversano i seguenti nodi: $n5 \rightarrow n6 \rightarrow n3 \rightarrow n1$. Confermiamo con WS: conoscendo gli indirizzi IP di $n4$ e $n0$ applichiamo il filtro “`ip.addr==10.0.3.1 && ip.addr==192.138.1.1`” sui pcap dei nodi non terminali e vediamo che il numero di pacchetti intercettati è sempre 1874 pacchetti.

Nel secondo ($n8 \rightarrow n2$), i pacchetti attraversano i seguenti nodi: $n6 \rightarrow n3 \rightarrow n1$. Su $n7$ avvengono solo delle query broadcast dove $n8$ chiede l'indirizzo IP di $n6$ (Immagine 15). Confermiamo con WS: conoscendo gli indirizzi IP di $n8$ e $n2$ applichiamo il filtro “`ip.addr==192.138.2.3 && ip.addr==192.138.1.3`” sui pcap dei nodi non terminali: il numero di pacchetti intercettati è sempre 1302 pacchetti.

Ciò conferma l'evidenza grafica, dato che non esistono possibili percorsi alternativi che i pacchetti potrebbero intraprendere per arrivare da tale sorgente a tale destinazione.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	00:00:00_00:00:0e	Broadcast	ARP	64	who has 192.138.2.1? Tell 192.138.2.3
2	0.016352	00:00:00_00:00:0c	Broadcast	ARP	64	who has 192.138.2.3? Tell 192.138.2.1

Immagine 15 - sul nodo $n7$ notiamo solamente delle query broadcast inviate da $n8$ che chiede l'indirizzo IP di $n6$

A3) Calcolo e grafico di round trip time (RTT) e commento:

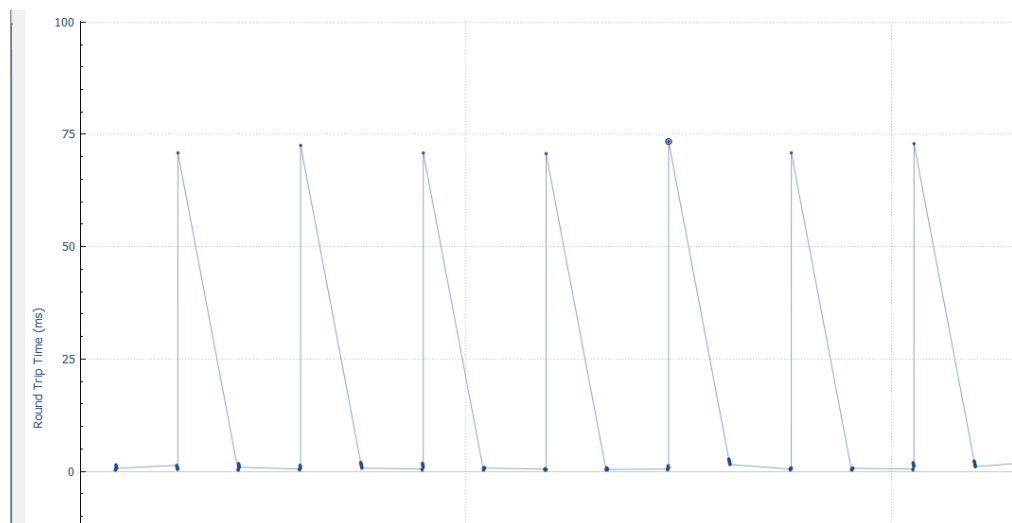


Immagine 16 - Grafico RTT

Dal grafico di WS (Immagine 16) si vede che i picchi dell' RTT sono di circa 74 ms.

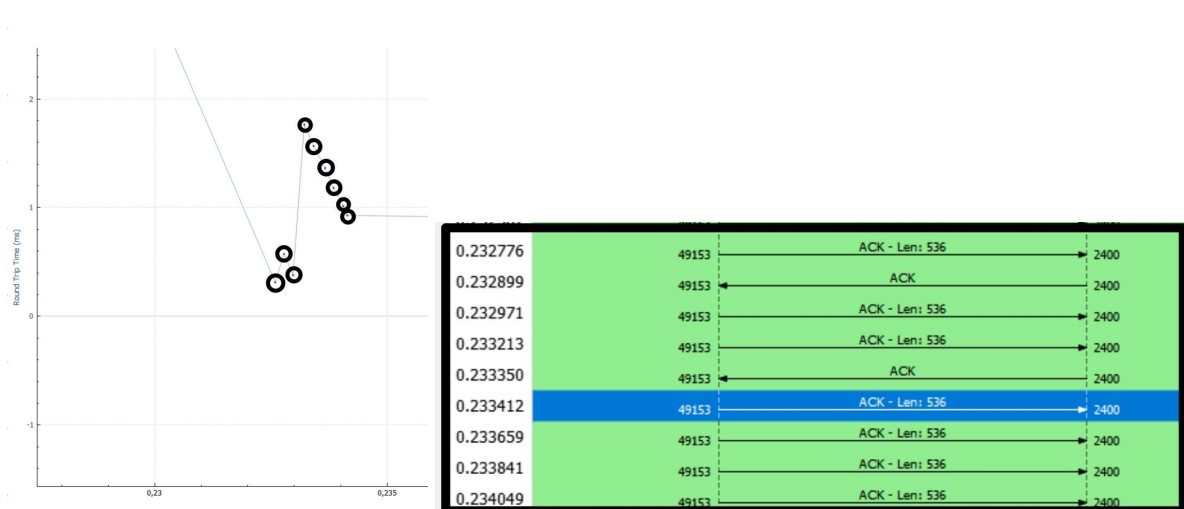


Immagine 17 - Confronto tra pacchetti sul grafico RTT e stessi pacchetti sul grafico TCP Flow

Come per la configurazione precedente, è possibile vedere (Immagine 17) l'analogia tra il dettaglio del grafico dell' RTT (sulla sinistra) e il grafico di flusso (sulla destra)

A4) Vi sono dei bottleneck nella rete? Se si, individuare gli eventuali link e discutere eventuali contromisure e soluzioni:

Essendo la risposta a questa domanda indipendente dalle configurazione, vale

la risposta già riportata nella A4 della configurazione 0 (vedi sopra).

C11) Calcolare il throughput medio dei flussi TCP:

Questa configurazione è caratterizzata da due flussi: uno TCP che va da n4 a n0 ed un altro TCP che va da n8 a n2. Possiamo quindi come prima, mediante l'utilizzo della colonna cumulative bytes (Immagine 9) , andare a calcolare il throughput medio dei due flussi dapprima sulla CSMA e poi su una delle interfacce point to point, ad esempio l'interfaccia 1. Applichiamo il filtro "tcp" per eliminare dai pacchetti catturati dal nodo n1 sulla interfaccia CSMA i pacchetti ARP. In questo caso, naturalmente, l'intervallo di interesse è dei 13 secondi di cui si compone la connessione tra i diversi client con i rispettivi server. Il valore finale di cumulative bytes filtrato sui flussi TCP è di 1276836 bytes, questo si traduce in un throughput medio di $1276836 * 8 / 13 = 785745,23$ bit/s sulla CSMA contro i 754473,84 bit/s della interfaccia 0.

Statistiche			
Misure	Catturati	Visualizzati	Marcati
Pacchetti	3184	3176 (99.7%)	—
Tempo, s	12.989	12.989	—
PPS medi	245.1	244.5	—
Dimensione media dei pacchetti, B	401	402	—
Byte	1277348	1276836 (100.0%)	0
Byte/s medi	98 k	98 k	—
Bit/s medi	786 k	786 k	—

Immagine 18 - Throughput Medio sulla CSMA

Statistiche			
Misure	Catturati	Visualizzati	Marcati
Pacchetti	3176	3176 (100.0%)	—
Tempo, s	12.993	12.993	—
PPS medi	244.4	244.4	—
Dimensione media dei pacchetti, B	386	386	—
Byte	1226020	1226020 (100.0%)	0
Byte/s medi	94 k	94 k	—
Bit/s medi	754 k	754 k	—

Immagine 19 - Throughput Medio dell'interfaccia i0

C12) Calcolare il throughput medio del flusso TCP n8 verso n2 a tempo t=6s.:

Per stimare il throughput medio del flusso TCP richiesto andiamo ad applicare in prima battuta il filtro WS:

"tcp&&ip.addr==192.138.1.3&&ip.addr==192.138.2.3&&tcp.time_relative<=4".

Come prima vale l'osservazione per cui Tass=6s, TRel=4s. Dunque andando a

prendere l'ultimo valore di cumulative bytes otteniamo un throughput medio sulla CSMA del flusso in considerazione di $299728 * 8 / 3.974607 = 603285$ bit/s contro i $287792 * 8 / 3.978629 = 578675$ bit/s sulla interfaccia 0. Per questi calcoli stiamo appositamente considerando il flusso TCP compreso di ACK da parte del sink n2.

Statistiche			
Misure	Catturati	Visualizzati	Marcati
Pacchetti	3184	746 (23.4%)	—
Tempo, s	12.989	3.975	—
PPS medi	245.1	187.7	—
Dimensione media dei pacchetti, B	401	402	—
Byte	1277348	299728 (23.5%)	0
Byte/s medi	98 k	75 k	—
Bit/s medi	786 k	603 k	—

Immagine 20 - Throughput Medio sulla CSMA

Statistiche			
Misure	Catturati	Visualizzati	Marcati
Pacchetti	3176	746 (23.5%)	—
Tempo, s	12.993	3.979	—
PPS medi	244.4	187.5	—
Dimensione media dei pacchetti, B	386	386	—
Byte	1226020	287792 (23.5%)	0
Byte/s medi	94 k	72 k	—
Bit/s medi	754 k	578 k	—

Immagine 21 - Throughput Medio dell'interfaccia i0

C13) Calcolare il throughput medio del flusso TCP n8 verso n2 a tempo $t=8s$.

Commentare eventuali cambiamenti rispetto a C12:

Per stimare il throughput medio del flusso TCP richiesto andiamo ad applicare in prima battuta il filtro WS:

“tcp&&ip.addr==192.138.1.3&&ip.addr==192.138.2.3&&tcp.time_relative<=6”.

Come prima vale l'osservazione per cui $T_{ass}=8s$, $T_{Rel}=6s$. Dunque andando a prendere l'ultimo valore di cumulative bytes otteniamo un throughput medio sulla CSMA del flusso in considerazione di $452188 * 8 / 5.9908 = 603843$ bit/s contro i $434204 * 8 / 5.9948 = 579440,85$ bit/s sulla interfaccia 0. Per questi calcoli stiamo appositamente considerando il flusso TCP compreso di ACK da parte del sink n2.

Il cambiamento del valore medio rispetto a C02 è dovuto al fatto che al tempo $t=8s$ la trasmissione è a regime.

Statistiche			
Misure	Catturati	Visualizzati	Marcati
Pacchetti	3184	1124 (35.3%)	—
Tempo, s	12.989	5.991	—
PPS medi	245.1	187.6	—
Dimensione media dei pacchetti, B	401	402	—
Byte	1277348	452188 (35.4%)	0
Byte/s medi	98 k	75 k	—
Bit/s medi	786 k	603 k	—

Immagine 22 - Throughput Medio sulla CSMA

Statistiche			
Misure	Catturati	Visualizzati	Marcati
Pacchetti	3176	1124 (35.4%)	—
Tempo, s	12.993	5.995	—
PPS medi	244.4	187.5	—
Dimensione media dei pacchetti, B	386	386	—
Byte	1226020	434204 (35.4%)	0
Byte/s medi	94 k	72 k	—
Bit/s medi	754 k	579 k	—

Immagine 23 - Throughput Medio dell'interfaccia i0

C14) [Extra] Ritardo di accodamento vs congestione: Disegnare un grafico che mostri il ritardo di accodamento in funzione del livello di congestione in rete:

CONFIGURAZIONE 2:

A1) Individuare le varie topologie note che compongono la rete:

Essendo la risposta a questa domanda indipendente dalle configurazioni, vale la risposta già riportata nella A1 della configurazione 0 (vedi sopra).

A2) Ricostruzione del percorso dei pacchetti attraverso la rete di tutti i flussi simulati usando Wireshark evidenziando i filtri utilizzati per isolare i singoli flussi dello strato di trasporto tra le tracce:

Abbiamo tre flussi:

Nel primo ($n7 \rightarrow n0$), i pacchetti UDP attraversano le seguenti interfacce: $n6 \rightarrow n3 \rightarrow n1$. Confermiamo con WS: conoscendo gli indirizzi IP di $n7$ e $n0$ e la porta di ricezione su $n0$ applichiamo il filtro “`ip.addr==192.138.2.2 && ip.addr==192.138.1.1 &&`

udp.port==2500” sui pcap dei nodi non terminali e vediamo che il numero di pacchetti intercettati è sempre lo stesso, 208 pacchetti.

Nel secondo (n8→n2), i pacchetti UDP attraversano le seguenti interfacce: n7→n6->n3->n1. Su n7 tuttavia avvengono solo delle query broadcast dove n8 chiede l’indirizzo IP di n6. Confermiamo con WS: conoscendo gli indirizzi IP di n8 e n2 e la porta di ricezione su n2 applichiamo il filtro “ip.addr==192.138.2.3 && ip.addr==192.138.1.3 && udp.port==63” sui pcap dei nodi non terminali n6, n3, n1 e vediamo che il numero di pacchetti intercettati è sempre lo stesso, 10 pacchetti.

Nel terzo (n4→n2), i pacchetti TCP attraversano le seguenti interfacce: n5→n6->n3->n1. Confermiamo con WS: conoscendo gli indirizzi IP di n4 e n2 e la porta di ricezione su n2 applichiamo il filtro “ip.addr==10.0.3.1 && ip.addr==192.138.1.3 && tcp.port==2600” sui pcap dei nodi non terminali e vediamo che il numero di pacchetti intercettati è sempre lo stesso, 1122 pacchetti.

A3) Calcolo e grafico di round trip time (RTT) e commento:

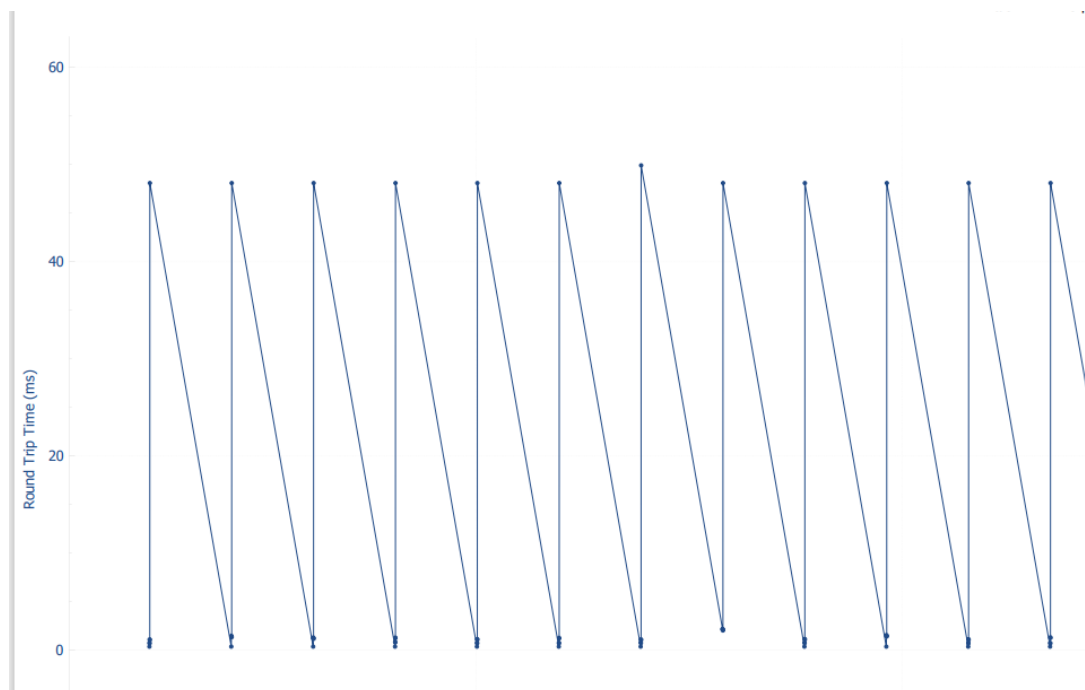


Immagine 24 - Grafico RTT

Il grafico RTT (Immagine 24) mostra che i picchi valgono circa 48ms

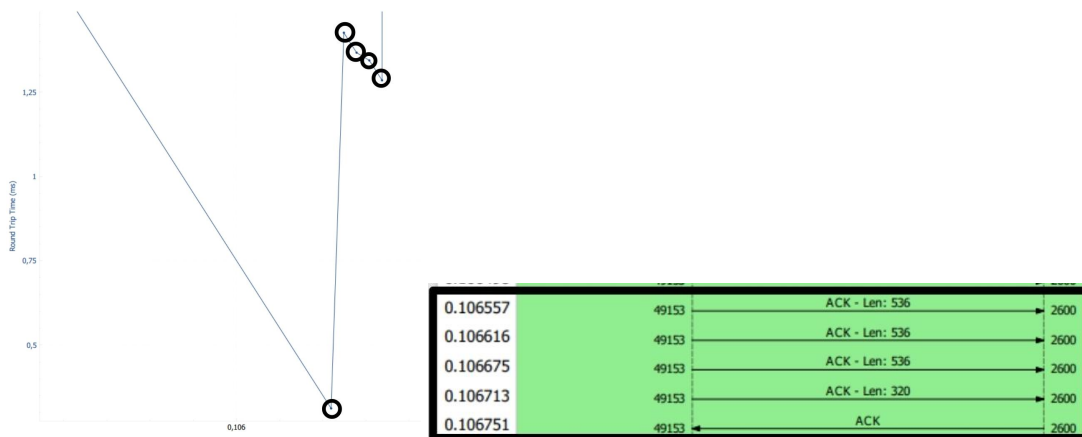


Immagine 25 - Confronto tra pacchetti sul grafico RTT e stessi pacchetti sul grafico TCP Flow

Ancora una volta, è possibile notare (Immagine 25) come ci sia un'analogia tra il grafico dell' RTT e il grafico di flusso

A4) Vi sono dei bottleneck nella rete? Se sì, individuare gli eventuali link e discutere eventuali contromisure e soluzioni:

Essendo la risposta a questa domanda indipendente dalle configurazioni, vale la risposta già riportata nella A4 della configurazione 0 (vedi sopra).

C21) Calcolare il throughput medio del flusso TCP a tempo $t=5s$:

Siamo interessati al throughput medio del flusso TCP n4 verso n2 al tempo $t=5s$.

Applichiamo un filtro sugli IP che ci interessano, quelli di n4 e n2, rispettivamente 10.0.3.1 e 192.138.1.3. In questo caso, si possono notare dei pacchetti relativi al protocollo Zebra, considerando questi parte del flusso TCP possiamo filtrare come segue:

“(tcp||zebra)&&ip.addr==10.0.3.1&&ip.addr==192.138.1.3&&tcp.time_relative<=2”, considerando come prima che la richiesta è espressa in TAss=5s e TRel=2s, andando a vedere i cumulative bytes dell'ultimo pacchetto nell'intervallo richiesto si ha un throughput medio sulla interfaccia CSMA di $149048 \times 8 / 1.975 = 603738,73$ bit/s contro i 578457,457 bit/s sulla interfaccia 0.

Statistiche			
Misure	Catturati	Visualizzati	Marcati
Pacchetti	1772	372 (21.0%)	—
Tempo, s	11.982	1.976	—
PPS medi	147.9	188.3	—
Dimensione media dei pacchetti, B	636	401	—
Byte	1126748	149048 (13.2%)	0
Byte/s medi	94 k	75 k	—
Bit/s medi	752 k	603 k	—

Immagine 26 - Throughput Medio sulla CSMA

Statistiche			
Misure	Catturati	Visualizzati	Marcati
Pacchetti	1766	372 (21.1%)	—
Tempo, s	11.985	1.980	—
PPS medi	147.3	187.9	—
Dimensione media dei pacchetti, B	622	385	—
Byte	1098108	143096 (13.0%)	0
Byte/s medi	91 k	72 k	—
Bit/s medi	732 k	578 k	—

Immagine 27 - Throughput Medio sull'interfaccia i0

C22) Calcolare il throughput medio del flusso TCP a tempo $t=7s$. Commentare eventuali cambiamenti rispetto a C21:

Siamo interessati al throughput medio del flusso TCP n4 verso n2 al tempo $t=7s$. Applichiamo un filtro sugli IP che ci interessano, quelli di n4 e n2, rispettivamente 10.0.3.1 e 192.138.1.3. In questo caso, si possono notare dei pacchetti relativi al protocollo Zebra, considerando questa parte del flusso TCP possiamo filtrare come segue:

“(tcp||zebra)&&ip.addr==10.0.3.1&&ip.addr==192.138.1.3&&tcp.time_relative<=4”, considerando come prima che la richiesta è espressa in $T_{Ass}=7s$ e $T_{Rel}=4s$, andando a vedere i cumulative bytes dell'ultimo pacchetto nell'intervallo richiesto si ha un throughput medio sulla interfaccia CSMA di $301508 \cdot 8 / 3.99 = 604527,32$ bit/s contro i $289508 \cdot 8 / 3.99 = 580467,16$ bit/s sulla interfaccia 0.

Il cambiamento del valore medio rispetto a C21 è dovuto al fatto che al tempo $t=7s$ la trasmissione è a regime.

Statistiche			
Misure	Catturati	Visualizzati	Marcati
Pacchetti	1772	750 (42.3%)	—
Tempo, s	11.982	3.992	—
PPS medi	147.9	187.9	—
Dimensione media dei pacchetti, 636 B		402	—
Byte	1126748	301508 (26.8%)	0
Byte/s medi	94 k	75 k	—
Bit/s medi	752 k	604 k	—

Immagine 28 - Throughput Medio sul CSMA

Statistiche			
Misure	Catturati	Visualizzati	Marcati
Pacchetti	1766	750 (42.5%)	—
Tempo, s	11.985	3.996	—
PPS medi	147.3	187.7	—
Dimensione media dei pacchetti, 622 B		386	—
Byte	1098108	289508 (26.4%)	0
Byte/s medi	91 k	72 k	—
Bit/s medi	732 k	579 k	—

Immagine 29 - Throughput Medio sull'interfaccia i0

C23) [Extra] Ritardo di accodamento vs congestione: Disegnare un grafico che mostri il ritardo di accodamento in funzione del livello di congestione in rete:

ADDENDUM: CALCOLI RISPOSTA C04

Per completezza, il calcolo è mostrato qua sotto:

$d_{prop} n_4 \rightarrow n_2 = 3 \cdot 10^{-5} \text{ sec}$

$N = 4, C = 80 \cdot 10^6$

per len 58: $X/L = 2, H + L = 464,$

$((H+L)/C1) \cdot 4 = 2,32 \cdot 10^{-5}$

$(H + L) \cdot \text{num_pack} / C1 = 1,16 \cdot 10^{-5}$

$-(H + L) / C1 = 5,8 \cdot 10^{-6}$

Somma58 = $4,06 \cdot 10^{-5}$

per len 54: $X/L = 753, H + L = 432$

$((H+L)/C1) \cdot 4 = 2,16 \cdot 10^{-5}$

$(H + L) \cdot \text{num_pack} / C1 = 4,066 \cdot 10^{-3}$

$-(H + L) / C1 = -5,4 \cdot 10^{-6}$

$$\text{Somma54} = 4,082 * 10^{-3}$$

$$\text{per len 590: } X/L = 998, H + L = 4720$$

$$((H+L)/C1)^4 = 2,36 * 10^{-4}$$

$$(H + L) * \text{num_pack} / C1 = 0,058882$$

$$- (H + L) / C1 = - 5,9 * 10^{-5}$$

$$\text{Somma590} = 0,059059$$

$$\text{per len 482: } X/L = 499, H + L = 3856$$

$$((H+L)/C1)^4 = 1,928 * 10^{-4}$$

$$(H + L) * \text{num_pack} / C1 = 0,02405$$

$$- (H + L) / C1 = - 4,82 * 10^{-5}$$

$$\text{Somma482} = 0,02419$$

$$\text{Somma totale p2p + dprop n4} \rightarrow n2 = 0,0874016$$

CSMA:

$$N = 1, C = 25 * 10^6$$

$$\text{per len 64: } X/L = 4, H + L = 512$$

$$(H + L) * \text{num_pack} / C = 8,192 * 10^{-5}$$

$$\text{per len 74: } X/L = 2, H + L = 592$$

$$(H + L) * \text{num_pack} / C = 4,736 * 10^{-5}$$

$$\text{per len 606: } X/L = 998, H + L = 4848$$

$$(H + L) * \text{num_pack} / C = 0,193532$$

$$\text{per len 70: } X/L = 753, H + L = 560$$

$$(H + L) * \text{num_pack} / C = 0,0168672$$

$$\text{per len 498: } X/L = 499, H + L = 3984$$

$$(H + L) * \text{num_pack} / C = 0,07952$$

$$\text{Somma totale CSMA} = 0,290049$$

$$\text{Somma totale p2p + dprop n4} \rightarrow n2 = 0,0874016$$

$$\text{Somma totale} = 0,37745$$