

REPORT RISPOSTE HOMEWORK N.4

Telecomunicazioni 2022

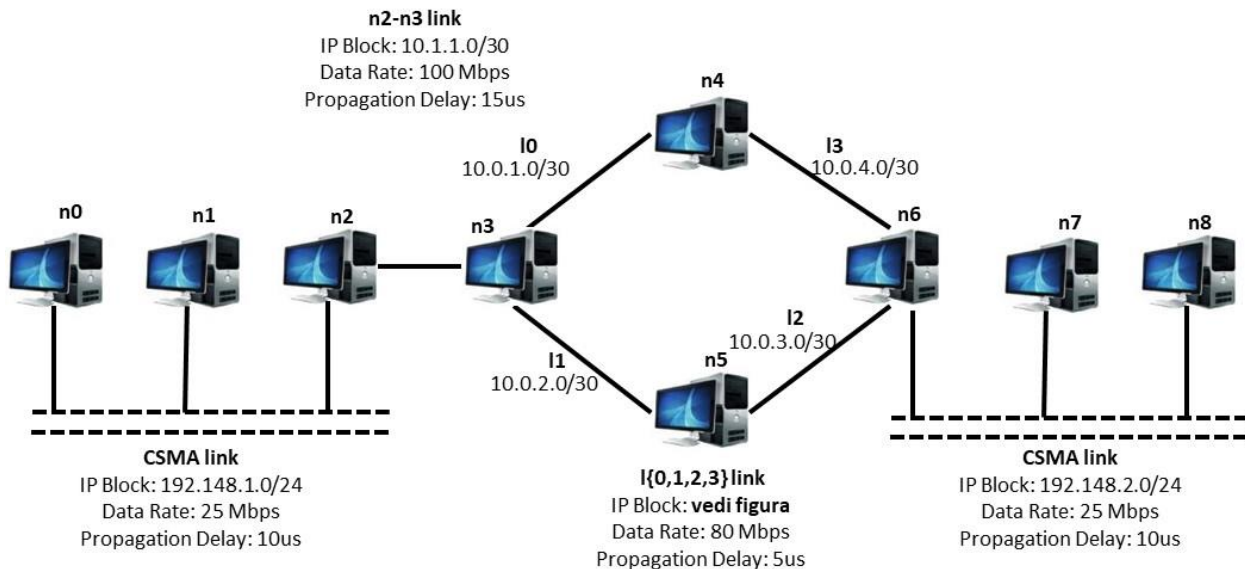


Fig.1 Rappresentazione grafica della rete simulata

Per tutte le risposte non è stato preso in considerazione il protocollo ZEBRA, perché come indicato in una risposta alle domande poste per i vari HW (03/11/2022, Q5-A5), nella struttura di rete considerata, tale protocollo opera esclusivamente per riempire i pacchetti che successivamente verranno affidati al protocollo di trasporto TCP.

«In questo caso però zebra se notate non sta venendo usato come protocollo di routing, ma piuttosto per "riempire" qualcosa.»

Configuration n.0

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

Prendendo come riferimento lo schema riportato in Fig.1, descriviamo la topologia della rete avendo il parametro "configuration=0".

Agli estremi della rete sono presenti due reti distinte di tipo CSMA, organizzate nel seguente raggruppamento:

- CSMA SX: n0, n1, n2;
- CSMA DX: n6, n7, n8;

Procedendo da sinistra verso destra, continuando nella descrizione della struttura della rete sono presenti:

- tra i nodi n2 e n3 un collegamento Point-to-Point;
- tra i nodi n3 e n4 un collegamento Point-to-Point (I0);
- tra i nodi n3 e n5 un collegamento Point-to-Point (I1);
- tra i nodi n6 e n4 un collegamento Point-to-Point (I3);
- tra i nodi n6 e n5 un collegamento Point-to-Point (I2);

Considerando la struttura di rete sopra descritta, sono presenti nella rete:

- TCP Sink installato sul nodo n0;
- TCP OnOff Client installato sul nodo n8;

REPORT RISPOSTE HOMEWORK N.4

Telecomunicazioni 2022

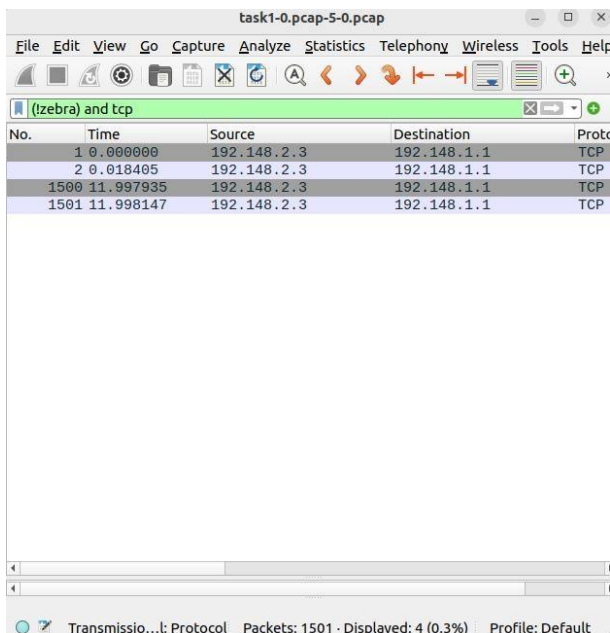
A2) Ricostruzione del percorso dei pacchetti attraverso la rete di tutti i flussi simulati usando WireShark evidenziando i filtri utilizzati per isolare i singoli flussi dello strato di trasporto tra le tracce.

Per rispondere a questa domanda sono stati generati dei file *pcap* su ogni singolo collegamento di ogni nodo. Tuttavia, nel codice allegato sono stati presi in considerazione solamente i nodi n3, n6 per la generazione dei file *pcap*, come da consegna.

Dal nodo 8 partono 755 pacchetti con protocollo TCP (*filtro: !(zebra) and tcp*), e non sono presenti pacchetti con protocollo UDP.

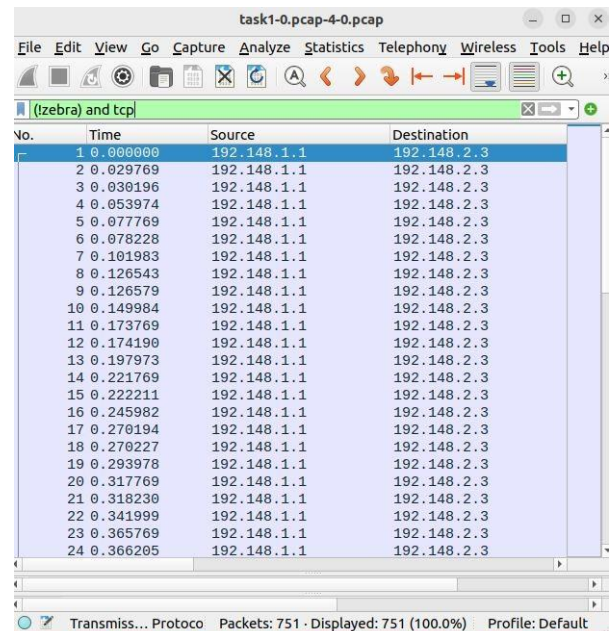
I 755 pacchetti passano per il nodo 7 e arrivano al nodo 6. Vengono poi indirizzati nella seguente maniera:

- 4 pacchetti arrivano sul nodo 5 (*Fig.2*);
- 751 pacchetti arrivano sul nodo 4 (*Fig.3*);



No.	Time	Source	Destination	Proto
1	0.000000	192.148.2.3	192.148.1.1	TCP
2	0.018405	192.148.2.3	192.148.1.1	TCP
1500	11.997935	192.148.2.3	192.148.1.1	TCP
1501	11.998147	192.148.2.3	192.148.1.1	TCP

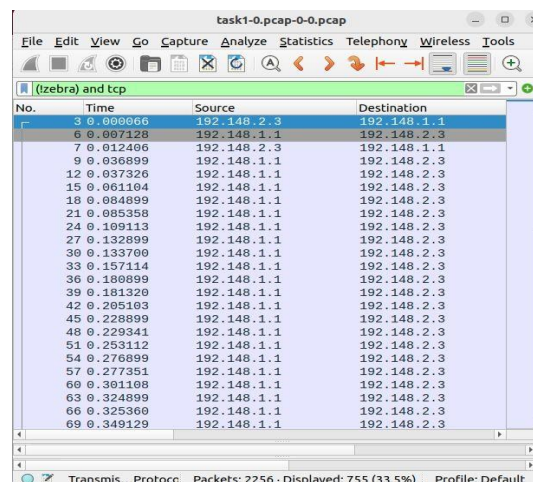
Fig.2 File pcap sul nodo 5, (Filtro: !(zebra) and tcp)



No.	Time	Source	Destination
1	0.000000	192.148.1.1	192.148.2.3
2	0.029769	192.148.1.1	192.148.2.3
3	0.030196	192.148.1.1	192.148.2.3
4	0.053974	192.148.1.1	192.148.2.3
5	0.077769	192.148.1.1	192.148.2.3
6	0.078228	192.148.1.1	192.148.2.3
7	0.101983	192.148.1.1	192.148.2.3
8	0.126543	192.148.1.1	192.148.2.3
9	0.126579	192.148.1.1	192.148.2.3
10	0.149984	192.148.1.1	192.148.2.3
11	0.173769	192.148.1.1	192.148.2.3
12	0.174190	192.148.1.1	192.148.2.3
13	0.197973	192.148.1.1	192.148.2.3
14	0.221769	192.148.1.1	192.148.2.3
15	0.222211	192.148.1.1	192.148.2.3
16	0.245982	192.148.1.1	192.148.2.3
17	0.270194	192.148.1.1	192.148.2.3
18	0.270227	192.148.1.1	192.148.2.3
19	0.293978	192.148.1.1	192.148.2.3
20	0.317769	192.148.1.1	192.148.2.3
21	0.318230	192.148.1.1	192.148.2.3
22	0.341999	192.148.1.1	192.148.2.3
23	0.365769	192.148.1.1	192.148.2.3
24	0.366205	192.148.1.1	192.148.2.3

Fig.3 File pcap sul nodo 4, (Filtro: !(zebra) and tcp)

I pacchetti poi arrivano sul nodo 3, in totale 755, passano per i nodi 2,1 e arrivano al nodo 0 (*Fig.4*)



No.	Time	Source	Destination
3	0.000066	192.148.2.3	192.148.1.1
6	0.007128	192.148.1.1	192.148.2.3
7	0.012406	192.148.2.3	192.148.1.1
9	0.036999	192.148.1.1	192.148.2.3
12	0.037326	192.148.1.1	192.148.2.3
15	0.061104	192.148.1.1	192.148.2.3
18	0.084899	192.148.1.1	192.148.2.3
21	0.085358	192.148.1.1	192.148.2.3
24	0.109113	192.148.1.1	192.148.2.3
27	0.132899	192.148.1.1	192.148.2.3
30	0.133700	192.148.1.1	192.148.2.3
33	0.157114	192.148.1.1	192.148.2.3
36	0.180899	192.148.1.1	192.148.2.3
39	0.181320	192.148.1.1	192.148.2.3
42	0.205103	192.148.1.1	192.148.2.3
45	0.228899	192.148.1.1	192.148.2.3
48	0.229341	192.148.1.1	192.148.2.3
51	0.253112	192.148.1.1	192.148.2.3
54	0.276899	192.148.1.1	192.148.2.3
57	0.277351	192.148.1.1	192.148.2.3
60	0.301100	192.148.1.1	192.148.2.3
63	0.324899	192.148.1.1	192.148.2.3
66	0.325360	192.148.1.1	192.148.2.3
69	0.349129	192.148.1.1	192.148.2.3

Fig.4 File pcap sul nodo 0, (Filtro: !(zebra) and tcp)

REPORT RISPOSTE HOMEWORK N.4

Telecomunicazioni 2022

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

RTT è l'acronimo di Round Trip Time, ovvero il tempo che passa da quando viene inviato il segmento TCP a quando viene ricevuto l'ACK del segmento stesso.

Per stimare l'RTT abbiamo usato una funzionalità di Wireshark che mostra il tempo che passa dall'invio di un pacchetto alla ricezione dell'ACK.

L'HandShake inizia al secondo 0,018469, quando il nodo n0 riceve l'ACK dal nodo n8. Per il primo segmento inviato, finisce al secondo 0,04317, quando n8 riceve l'ACK mandato precedentemente, più la dimensione del segmento (ACK = 537, Fig.5). L'RTT è quindi $0,043173 - 0,018469 = 24,704$ ms.



Fig.5

Questo è un singolo caso, ma andando ad analizzare altri casi ed in più avvalendoci del grafico fornito sempre da Wireshark (Fig.6), è evidente come l'RTT sia un valore stimabile in un intorno di 25 ms.

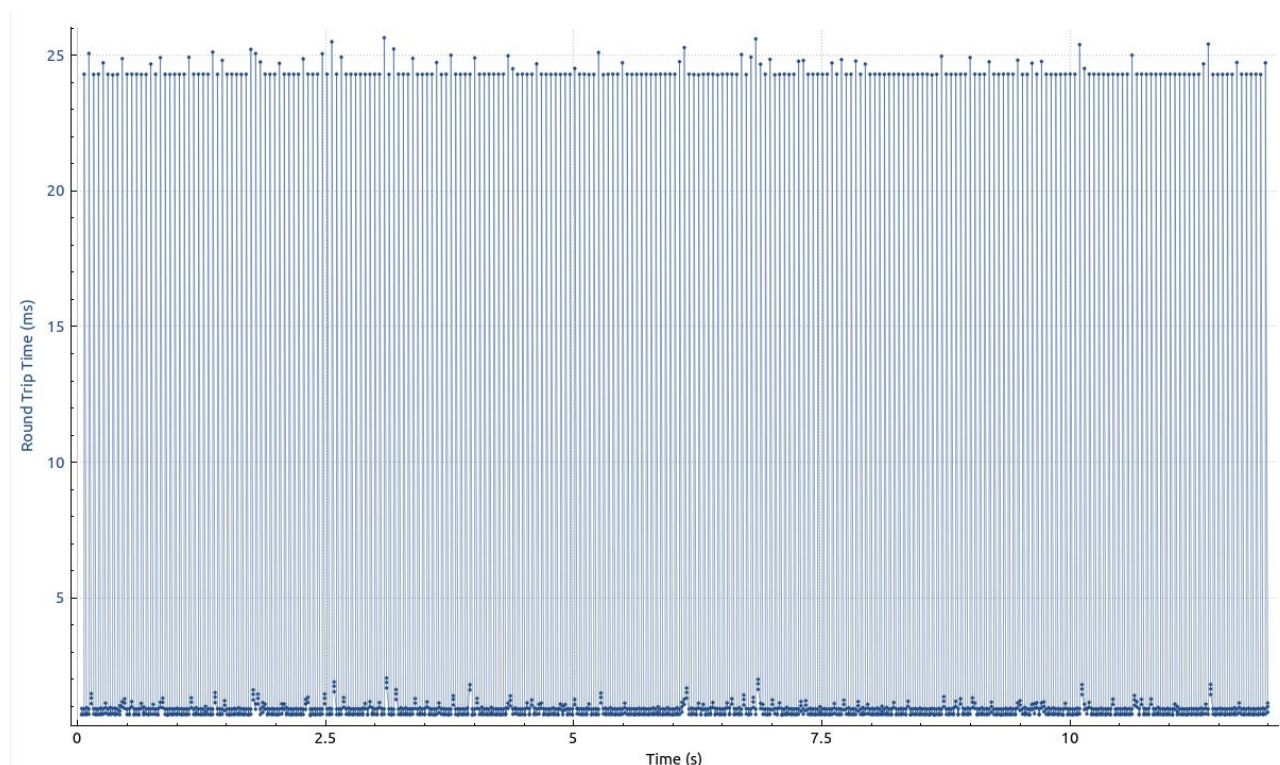


Fig.6

REPORT RISPOSTE HOMEWORK N.4

Telecomunicazioni 2022

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

In una rete il fenomeno del bottleneck si verifica quando un collegamento riceve una quantità di dati superiore al suo data rate.

Con il trasferimento di dati dal nodo n8 al nodo n0, i pacchetti dopo aver attraversato un collegamento con un data rate pari a 100Mbps (n2-n3 link) si ritrovano in un CSMA link con un data rate di soli 25Mbps. Questo è un bottleneck.

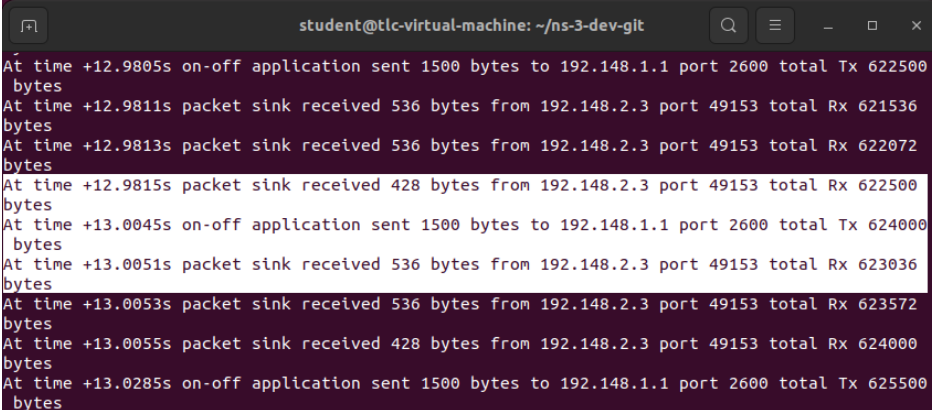
Per risolverlo, la soluzione principale consiste nel cambiare il collegamento con uno che permetta un data rate maggiore. Seppure sia la soluzione più efficace è anche quella più costosa. Invece una possibile contromisura è già presente nel protocollo TCP e consiste nel ridurre la sliding window o addirittura chiuderla momentaneamente una volta che viene individuato il verificarsi di una perdita di pacchetti, non è una vera e propria soluzione, ma è sicuramente una contromisura efficace.

Domande:

C01) Calcolare il throughput istantaneo del flusso TCP.

Per rispondere alla domanda è stato abilitato il log "LogComponentEnable" per conoscere la dimensione dei byte trasmessi in ogni istante di tempo.

Consideriamo un pacchetto casuale, ad esempio il pacchetto spedito all'istante +13.0045 s (Fig.7):



```
student@tlc-virtual-machine: ~/ns-3-dev-git
At time +12.9805s on-off application sent 1500 bytes to 192.148.1.1 port 2600 total Tx 622500 bytes
At time +12.9811s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 621536 bytes
At time +12.9813s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 622072 bytes
At time +12.9815s packet sink received 428 bytes from 192.148.2.3 port 49153 total Rx 622500 bytes
At time +13.0045s on-off application sent 1500 bytes to 192.148.1.1 port 2600 total Tx 624000 bytes
At time +13.0051s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 623036 bytes
At time +13.0053s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 623572 bytes
At time +13.0055s packet sink received 428 bytes from 192.148.2.3 port 49153 total Rx 624000 bytes
At time +13.0285s on-off application sent 1500 bytes to 192.148.1.1 port 2600 total Tx 625500 bytes
```

Fig.7

La sua dimensione è di 1500 bytes=12.000 bit

Calcoliamo il suo throughput istantaneo come:

$$Th = 12.000 \text{ bit} / (13.0051 - 12.9815) \text{ s} = 508.474, 5763 \text{ bit/s}$$

REPORT RISPOSTE HOMEWORK N.4

Telecomunicazioni 2022

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

Per rispondere alla domanda è stato abilitato il log "LogComponentEnable" per conoscere la dimensione dei byte trasmessi in ogni istante di tempo.

All'istante di tempo 4s sono stati trasmessi $T_x = 61.500 \text{ bytes} = 492.000 \text{ bit}$ (Fig.8)

```
student@tlc-virtual-machine: ~/ns-3-dev-git
bytes
At time +3.95748s packet sink received 428 bytes from 192.148.2.3 port 49153 total Rx 58500
bytes
At time +3.98047s on-off application sent 1500 bytes to 192.148.1.1 port 2600 total Tx 60000
bytes
At time +3.98107s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 59036
bytes
At time +3.98128s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 59572
bytes
At time +3.98154s packet sink received 428 bytes from 192.148.2.3 port 49153 total Rx 60000
bytes
At time +4.00447s on-off application sent 1500 bytes to 192.148.1.1 port 2600 total Tx 61500
bytes
At time +4.00507s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 60536
bytes
At time +4.00532s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 61072
bytes
```

Fig.8

Calcolo il throughput medio come:

$$Th = 492.000 \text{ bit} / (4 - 3) \text{ s} = 492.000 \text{ bit/s} \text{ (Fig.9)}$$

Dove 3s è l'istante di tempo in cui l'on-off client inizia a trasmettere i dati.

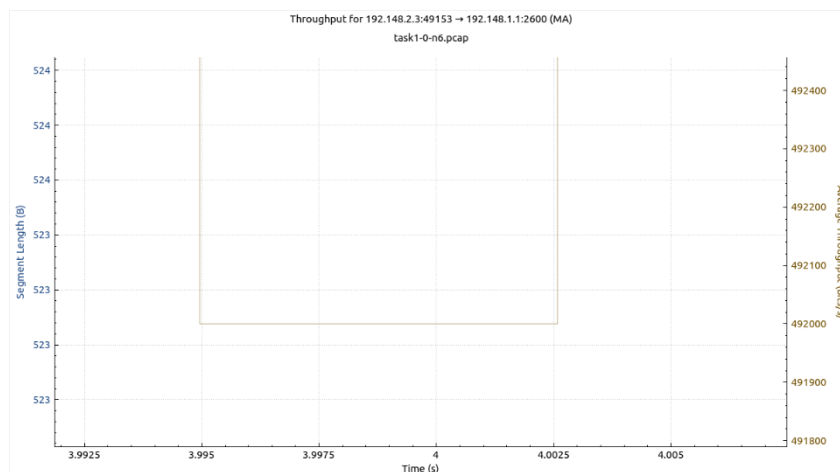


Fig.9

REPORT RISPOSTE HOMEWORK N.4

Telecomunicazioni 2022

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

Per rispondere alla domanda è stato abilitato il log "LogComponentEnable" per conoscere la dimensione dei byte trasmessi in ogni istante di tempo.

All'istante di tempo 7s sono stati trasmessi $T_x = 249.000 \text{ bytes} = 1.992.000 \text{ bit}$ (Fig.10)

```
student@tlc-virtual-machine: ~/ns-3-dev-git
bytes
At time +6.95728s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 245572
bytes
At time +6.95745s packet sink received 428 bytes from 192.148.2.3 port 49153 total Rx 246000
bytes
At time +6.98048s on-off application sent 1500 bytes to 192.148.1.1 port 2600 total Tx 247500
bytes
At time +6.98107s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 246536
bytes
At time +6.98131s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 247072
bytes
At time +6.98148s packet sink received 428 bytes from 192.148.2.3 port 49153 total Rx 247500
bytes
At time +7.00448s on-off application sent 1500 bytes to 192.148.1.1 port 2600 total Tx 249000
bytes
At time +7.00507s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 248036
bytes
```

Fig.10

Calcolo il throughput medio come:

$$Th = 1.992.000 \text{ bit} / (7 - 3) \text{ s} = 498.000 \text{ bit/s} \text{ (Fig.11)}$$

Dove 3s è l'istante di tempo in cui l'on-off client inizia a trasmettere i dati.

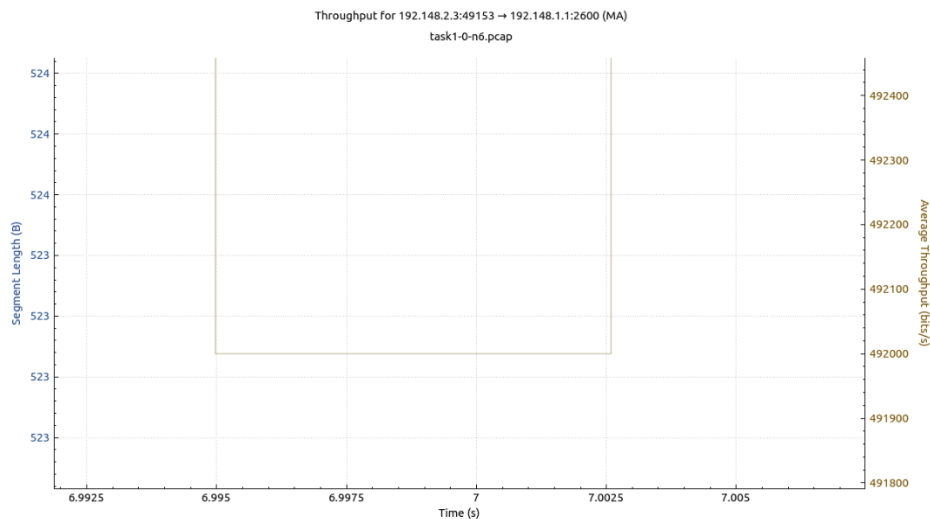


Fig.11

Il throughput aumenta a causa degli overhead di rete.

REPORT RISPOSTE HOMEWORK N.4

Telecomunicazioni 2022

C04) Calcolare il ritardo di trasferimento complessivo di tutti i pacchetti inviati.

Trascuriamo il ritardo di accodamento, elaborazione e i pacchetti relativi al protocollo zebra

$DELAY = L_i / R$ # ritardo di trasmissione dei singoli collegamenti

con:

L_i = numero bits che attraversano il collegamento i-esimo = 1500 bytes = 12000 bit

R = data rate collegamento, fornito dalla traccia

R1

$Delay1 = 12000 / 25000000 = 0,00048 \text{ s}$

R2

$Delay2 = 12000 / 80000000 = 0,00015 \text{ s}$

Oss. Essendo il ritardo uguale, consideriamo solo il caso in cui il pacchetto attraversi il collegamento n6-n5 rispetto a n6-n4

R3

$Delay3 = 12000 / 80000000 = 0,00015 \text{ s}$

R4

$Delay4 = 12000 / 100000000 = 0,00012 \text{ s}$

R5

$Delay5 = 12000 / 25000000 = 0,00048 \text{ s}$

D_{pi} = ritardo di propagazione i-esimo

$DELAY_{pacchetto} = Delay1 + Delay2 + Delay3 + Delay4 + Delay5 + D_{p1} + D_{p2} + D_{p3} + D_{p4} + D_{p5} =$

$= 0,001425 \text{ s}$ # ritardo di trasferimento di un solo pacchetto

$Delay_{tot} = 0,001425 * 4 = 0,0057 \text{ s}$ # ritardo totale pacchetti inviati da n8

$Delay_{tot} = 0,001425 * 755 = 1,075875 \text{ s}$ # ritardo totale pacchetti inviati da n8 e n0

REPORT RISPOSTE HOMEWORK N.4

Telecomunicazioni 2022

Configuration n.1

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

Prendendo come riferimento lo schema riportato in Fig. 1, descriviamo la topologia della rete avendo il parametro "configuration=1".

Agli estremi della rete sono presenti due reti distinte di tipo CSMA, organizzate nel seguente raggruppamento:

- CSMA SX: n0, n1, n2;
- CSMA DX: n6, n7, n8;

Procedendo da sinistra verso destra, continuando nella descrizione della struttura della rete sono presenti:

- tra i nodi n2 e n3 un collegamento Point-to-Point;
- tra i nodi n3 e n4 un collegamento Point-to-Point (I0);
- tra i nodi n3 e n5 un collegamento Point-to-Point (I1);
- tra i nodi n6 e n4 un collegamento Point-to-Point (I3);
- tra i nodi n6 e n5 un collegamento Point-to-Point (I2);

Considerando la struttura di rete sopra descritta, sono presenti nella rete:

- TCP Sink installato sul nodo n0;
- TCP Sink installato sul nodo n7;
- TCP OnOff Client installato sul nodo n8;
- TCP OnOff Client installato sul nodo n1;

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.

Per rispondere a questa domanda sono stati generati dei file *pcap* su ogni singolo collegamento di ogni nodo. Tuttavia, nel codice allegato sono stati presi in considerazione solamente i nodi n3, n6 per la generazione dei file *pcap*, come da consegna.

Distinguo due flussi di pacchetti:

1. I pacchetti che partono dal nodo 8 e arrivano al nodo 0
2. I pacchetti che partono dal nodo 1 e arrivano al nodo 7

Caso 1:

Dal nodo 8 partono 4 pacchetti con protocollo TCP, non sono presenti pacchetti con protocollo UDP. I pacchetti passano poi attraverso il nodo 7 e il 6, e vengono poi indirizzati nella seguente maniera:

- 4 pacchetti arrivano al nodo 5 (Fig. 12);
- 0 pacchetti arrivano al nodo 4 (Fig. 13);

REPORT RISPOSTE HOMEWORK N.4

Telecomunicazioni 2022

No.	Time	Source	Destination	Protocol
187	2.990766	192.148.2.3	192.148.1.1	TCP
188	3.001171	192.148.2.3	192.148.1.1	TCP
1685	12.988701	192.148.2.3	192.148.1.1	TCP
1686	12.988913	192.148.2.3	192.148.1.1	TCP

No.	Time	Source	Destination	Protocol
-----	------	--------	-------------	----------

task1-1.pcap-5-0.pcap Packets: 1686 · Displayed: 4 (0.2%) Profile: Default

task1-1.pcap-4-0.pcap Packets: 1499 · Displayed: 0 (0.0%) Profile: Default

Fig.12 file pcap del nodo 5 (Filtro: !(zebra) and tcp and ip.src==192.148.2.3 and ip.dst==192.148.1.1)

Fig.13 file pcap del nodo 4 (Filtro: !(zebra) and tcp and ip.src==192.148.2.3 and ip.dst==192.148.1.1)

I pacchetti passano poi per i nodi 3,2,1 e arrivano al nodo 0 (Fig.14)

No.	Time	Source	Destination	Protocol
5	2.998204	192.148.2.3	192.148.1.1	TCP
9	3.008544	192.148.2.3	192.148.1.1	TCP
1878	12.996074	192.148.2.3	192.148.1.1	TCP
1880	12.996286	192.148.2.3	192.148.1.1	TCP

task1-1.pcap-0-0.pcap Packets: 1880 · Displayed: 4 (0.2%) Profile: Default

Fig.14 file pcap del nodo 0 (Filtro: !(zebra) and tcp and ip.src==192.148.2.3 and ip.dst==192.148.1.1)

REPORT RISPOSTE HOMEWORK N.4

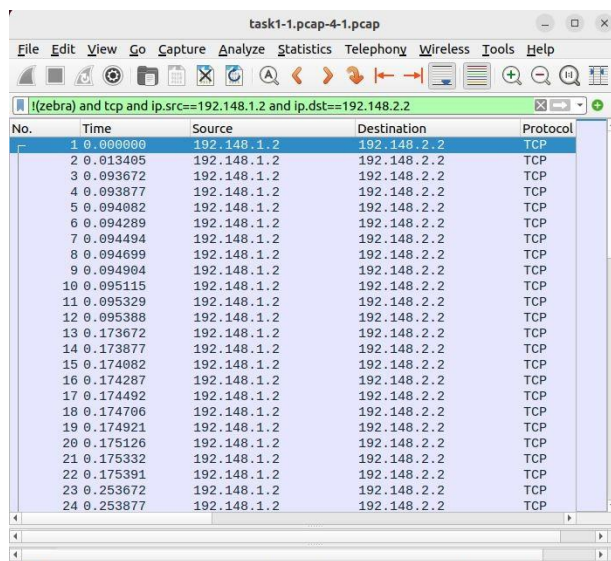
Telecomunicazioni 2022

Caso 2:

Dal nodo 1 partono 874 pacchetti con protocollo TCP, e non sono presenti pacchetti con protocollo UDP.

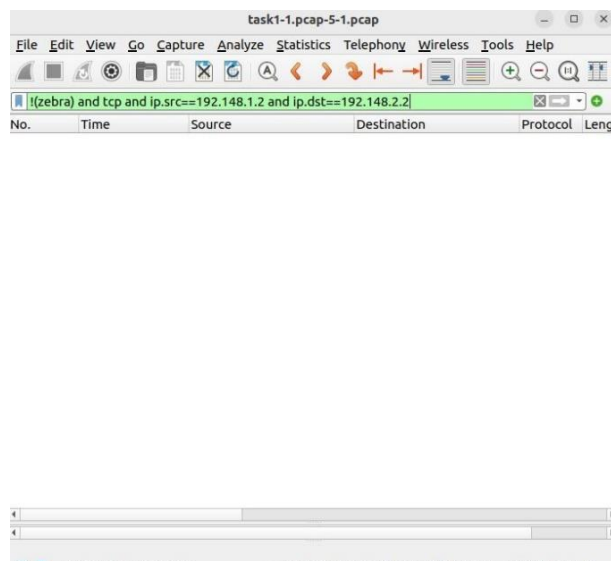
I pacchetti passano poi attraverso il nodo 2 e il 3, e vengono poi indirizzati nella seguente maniera:

- 874 pacchetti arrivano al nodo 4 (Fig.15);
- 0 pacchetti arrivano al nodo 5 (Fig.16);



No.	Time	Source	Destination	Protocol
1	0.000000	192.148.1.2	192.148.2.2	TCP
2	0.013405	192.148.1.2	192.148.2.2	TCP
3	0.093672	192.148.1.2	192.148.2.2	TCP
4	0.093877	192.148.1.2	192.148.2.2	TCP
5	0.094082	192.148.1.2	192.148.2.2	TCP
6	0.094289	192.148.1.2	192.148.2.2	TCP
7	0.094494	192.148.1.2	192.148.2.2	TCP
8	0.094699	192.148.1.2	192.148.2.2	TCP
9	0.094904	192.148.1.2	192.148.2.2	TCP
10	0.095115	192.148.1.2	192.148.2.2	TCP
11	0.095329	192.148.1.2	192.148.2.2	TCP
12	0.095388	192.148.1.2	192.148.2.2	TCP
13	0.173672	192.148.1.2	192.148.2.2	TCP
14	0.173877	192.148.1.2	192.148.2.2	TCP
15	0.174082	192.148.1.2	192.148.2.2	TCP
16	0.174287	192.148.1.2	192.148.2.2	TCP
17	0.174492	192.148.1.2	192.148.2.2	TCP
18	0.174706	192.148.1.2	192.148.2.2	TCP
19	0.174921	192.148.1.2	192.148.2.2	TCP
20	0.175126	192.148.1.2	192.148.2.2	TCP
21	0.175332	192.148.1.2	192.148.2.2	TCP
22	0.175391	192.148.1.2	192.148.2.2	TCP
23	0.253672	192.148.1.2	192.148.2.2	TCP
24	0.253877	192.148.1.2	192.148.2.2	TCP

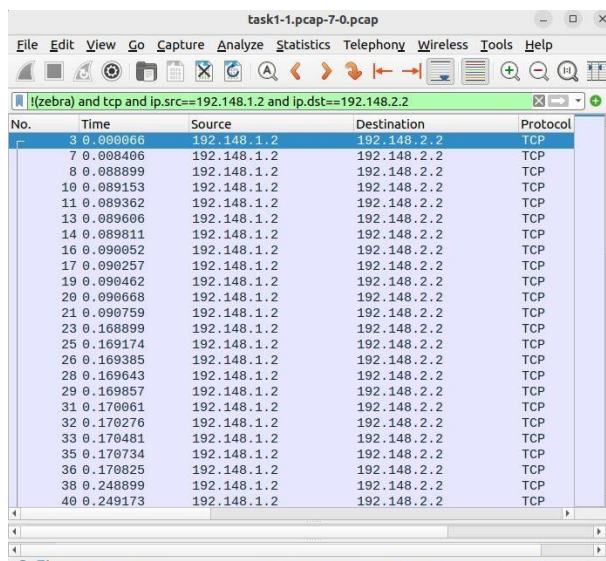
Fig.15 file pcap del nodo 4 (Filtro: !(zebra) and tcp and ip.src==192.148.1.2 and ip.dst==192.148.2.2)



No.	Time	Source	Destination	Protocol	Length
-----	------	--------	-------------	----------	--------

Fig.16 file pcap del nodo 5 (Filtro: !(zebra) and tcp and ip.src==192.148.1.2 and ip.dst==192.148.2.2)

I pacchetti passano poi per i nodi 6,7 e arrivano al nodo 8 (Fig.17)



No.	Time	Source	Destination	Protocol
3	0.000000	192.148.1.2	192.148.2.2	TCP
7	0.008406	192.148.1.2	192.148.2.2	TCP
8	0.008899	192.148.1.2	192.148.2.2	TCP
10	0.089153	192.148.1.2	192.148.2.2	TCP
11	0.089362	192.148.1.2	192.148.2.2	TCP
13	0.089606	192.148.1.2	192.148.2.2	TCP
14	0.089811	192.148.1.2	192.148.2.2	TCP
16	0.090052	192.148.1.2	192.148.2.2	TCP
17	0.090257	192.148.1.2	192.148.2.2	TCP
19	0.090462	192.148.1.2	192.148.2.2	TCP
20	0.090668	192.148.1.2	192.148.2.2	TCP
21	0.090759	192.148.1.2	192.148.2.2	TCP
23	0.168899	192.148.1.2	192.148.2.2	TCP
25	0.169174	192.148.1.2	192.148.2.2	TCP
26	0.169385	192.148.1.2	192.148.2.2	TCP
28	0.169643	192.148.1.2	192.148.2.2	TCP
29	0.169857	192.148.1.2	192.148.2.2	TCP
31	0.170061	192.148.1.2	192.148.2.2	TCP
32	0.170276	192.148.1.2	192.148.2.2	TCP
33	0.170481	192.148.1.2	192.148.2.2	TCP
35	0.170734	192.148.1.2	192.148.2.2	TCP
36	0.170825	192.148.1.2	192.148.2.2	TCP
38	0.248899	192.148.1.2	192.148.2.2	TCP
40	0.249173	192.148.1.2	192.148.2.2	TCP

Fig.17 file pcap del nodo 7 (filtro: !(zebra) and tcp and ip.src==192.148.1.2 and ip.dst==192.148.2.2)

REPORT RISPOSTE HOMEWORK N.4

Telecomunicazioni 2022

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

RTT è l'acronimo di Round Trip Time, ovvero il tempo che passa da quando viene inviato il segmento TCP a quando viene ricevuto l'ACK del segmento stesso.

Per calcolare l'RTT abbiamo usato una funzionalità di WireShark che mostra il tempo che passa dall'invio di un pacchetto alla ricezione dell'ACK.

Nel primo flusso di dati da n1 a n7: L'HandShake inizia al secondo 0,013469, quando n7 riceve l'ACK da n1. Per il primo segmento inviato, finisce al secondo 0,095759, quando n1 riceve l'ACK mandato precedentemente, più la dimensione del segmento (ACK = 537, Fig. 18). L'RTT è quindi $0,095759 - 0,013469 = 82,29$ ms.

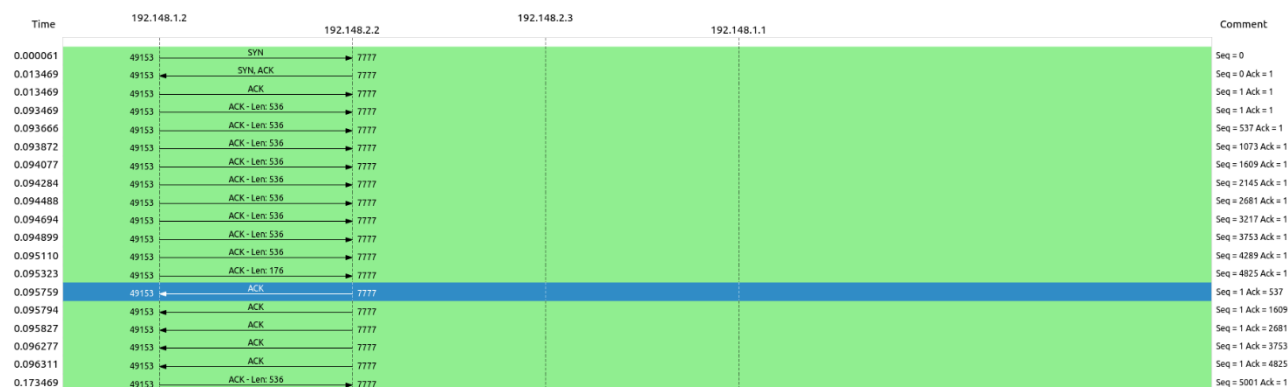


Fig.18

Questo è un singolo caso, ma andando ad analizzare altri casi ed in più avvalendoci del grafico fornito sempre da WireShark (Fig. 19) è evidente come l'RTT sia un valore stimabile in un intorno di 80 ms.

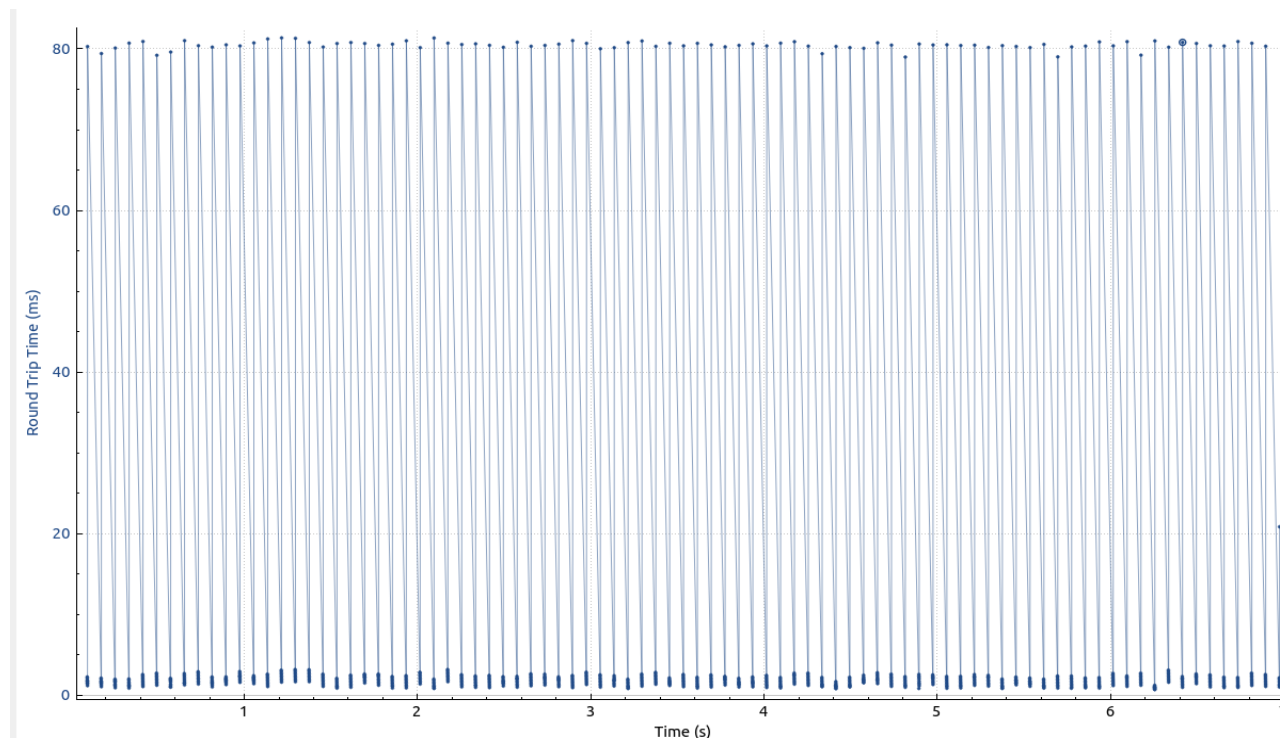


Fig.19

REPORT RISPOSTE HOMEWORK N.4

Telecomunicazioni 2022

Nel secondo flusso da n8 a n0: L'HandShake inizia al secondo 3.003301, quando n0 riceve l'ACK da n8. Finisce al secondo 3,044126, quando n8 riceve l'ACK mandato precedentemente, più la dimensione del segmento (ACK = 537, Fig.20). L'RTT è quindi $3,044126 - 3,003301 = 40,825$ ms.

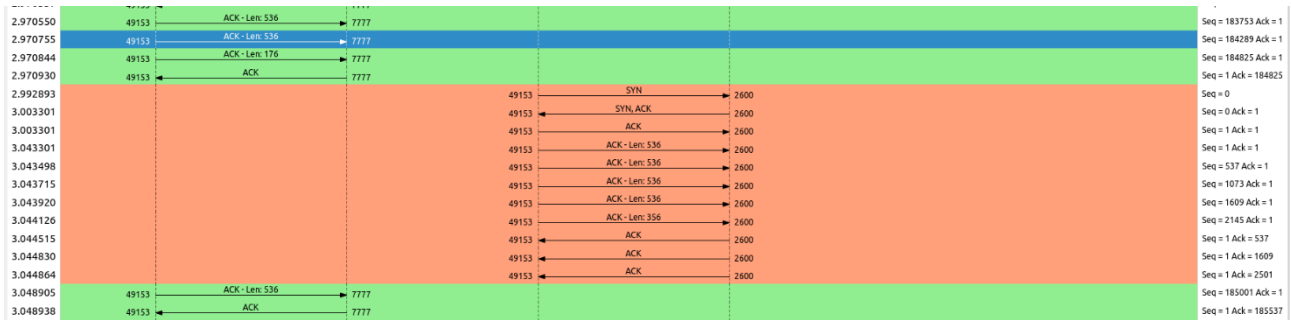


Fig.20

Questo è un singolo caso, ma andando ad analizzare altri casi ed in più avvalendoci del grafico fornito sempre da WireShark (Fig.21) è evidente come l'RTT sia un valore stimabile in un intorno di 40 ms.

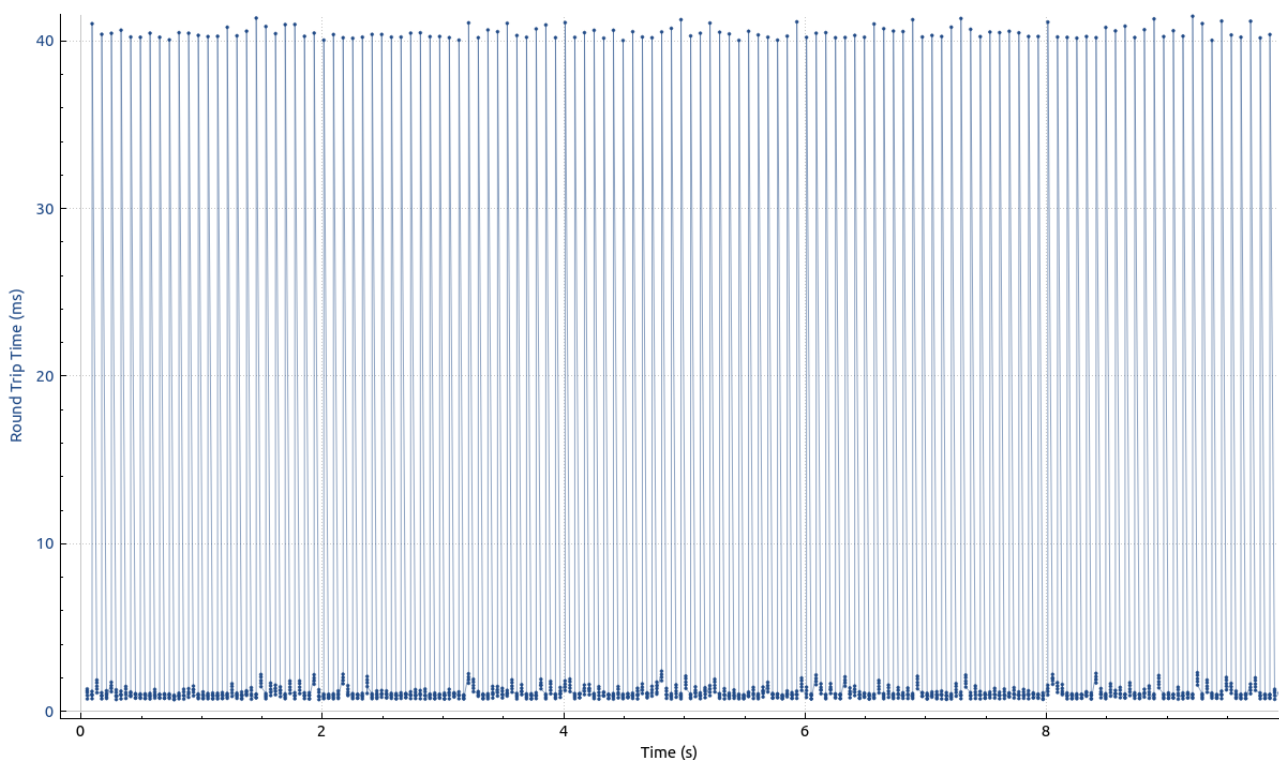


Fig.21

REPORT RISPOSTE HOMEWORK N.4

Telecomunicazioni 2022

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

In una rete il fenomeno del bottleneck si verifica quando un collegamento riceve una quantità di dati superiore al suo data rate.

I dati vengono trasferiti in un flusso dal nodo n8 al nodo n0 e nel secondo flusso da n1 a n7.

Nel primo flusso si verifica un bottleneck nel passaggio dal link n2-n3 al link CSMA.

Nel secondo flusso, si verifica un bottleneck nel link n6-n7 con data rate 25Mbps, dato che i dati arrivano a n6 tramite un collegamento con data rate 80Mbps.

Per risolvere un bottleneck la soluzione principale consiste nel cambiare il collegamento con uno che permetta un data rate maggiore. È un'ottima soluzione, ma molto costosa. Invece una possibile contromisura è già presente nel protocollo TCP e consiste nel ridurre la sliding window o addirittura chiuderla momentaneamente una volta che viene individuato il verificarsi di una perdita di pacchetti, non è una vera e propria soluzione, ma è sicuramente una contromisura efficace.

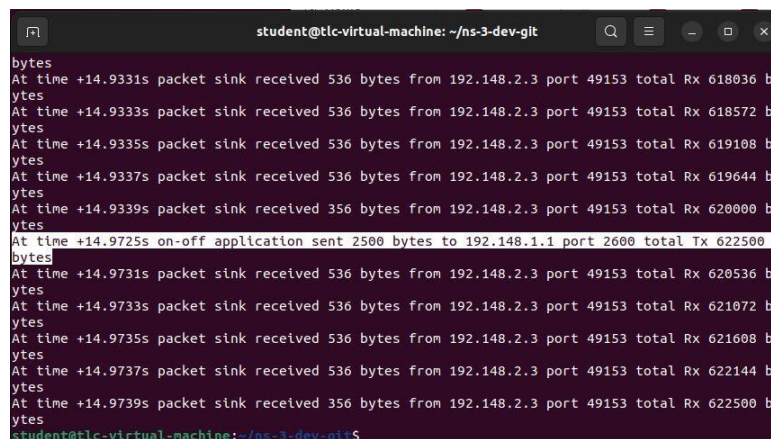
Domande:

C11) Calcolare il throughput medio dei flussi TCP.

Per rispondere alla domanda è stato abilitato il log "LogComponentEnable" per conoscere la dimensione dei byte trasmessi in ogni istante di tempo.

Considero il flusso n8 -> n0:

Sono stati trasmessi Tx = 622.500 bytes = 4.980.000 bit (Fig.22)



```
student@tlc-virtual-machine: ~/ns-3-dev-git
bytes
At time +14.9331s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 618036 b
ytes
At time +14.9333s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 618572 b
ytes
At time +14.9335s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 619108 b
ytes
At time +14.9337s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 619644 b
ytes
At time +14.9339s packet sink received 356 bytes from 192.148.2.3 port 49153 total Rx 620000 b
ytes
At time +14.9725s on-off application sent 2500 bytes to 192.148.1.1 port 2600 total Tx 622500
bytes
At time +14.9731s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 620536 b
ytes
At time +14.9733s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 621072 b
ytes
At time +14.9735s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 621608 b
ytes
At time +14.9737s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 622144 b
ytes
At time +14.9739s packet sink received 356 bytes from 192.148.2.3 port 49153 total Rx 622500 b
ytes
student@tlc-virtual-machine:~/ns-3-dev-git$
```

Fig.22

Calcolo il throughput medio come:

$$Th = 3.480.000 \text{ bit} / (9 - 2) \text{ s} = 497.142,857 \text{ bit/s}$$

Dove:

- 2s è l'istante di tempo in cui l'on-off client inizia a trasmettere i dati
- 9s è l'istante di tempo in cui l'on-off client smette di trasmettere dati

REPORT RISPOSTE HOMEWORK N.4

Telecomunicazioni 2022

Considero ora il flusso n1 -> n7:

Sono stati trasmessi Tx = 435.000 bytes = 3.480.000 bit (Fig.23)

```
student@tlc-virtual-machine: ~/ns-3-dev-git
At time +8.97307s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 245536 b
ytes
At time +8.97342s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 246072 b
ytes
At time +8.97363s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 246608 b
ytes
At time +8.97383s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 247144 b
ytes
At time +8.97398s packet sink received 356 bytes from 192.148.2.3 port 49153 total Rx 247500 b
ytes
At time +8.97748s on-off application sent 5000 bytes to 192.148.2.2 port 7777 total Tx 435000
bytes
At time +8.97808s packet sink received 536 bytes from 192.148.1.2 port 49153 total Rx 430536 b
ytes
At time +8.97833s packet sink received 536 bytes from 192.148.1.2 port 49153 total Rx 431072 b
ytes
At time +8.97855s packet sink received 536 bytes from 192.148.1.2 port 49153 total Rx 431608 b
ytes
At time +8.97883s packet sink received 536 bytes from 192.148.1.2 port 49153 total Rx 432144 b
ytes
At time +8.97903s packet sink received 536 bytes from 192.148.1.2 port 49153 total Rx 432680 b
ytes
At time +8.97934s packet sink received 536 bytes from 192.148.1.2 port 49153 total Rx 433216 b
ytes
```

Fig.23

Calcolo il throughput medio come:

$$Th = 3.480.000 \text{ bit} / (9 - 2) \text{ s} = 497.142,857 \text{ bit/s}$$

Dove:

- 2s è l'istante di tempo in cui l'on-off client inizia a trasmettere i dati
- 9s è l'istante di tempo in cui l'on-off client smette di trasmettere dati

C12) Calcolare il throughput medio del flusso TCP n8 verso n0 a tempo t=6s.

Per rispondere alla domanda è stato abilitato il log "LogComponentEnable" per conoscere la dimensione dei byte trasmessi in ogni istante di tempo.

All'istante di tempo 6s sono stati trasmessi Tx = 62.500 bytes = 500.000 bit (Fig.24)

```
student@tlc-virtual-machine: ~/ns-3-dev-git
At time +5.93988s packet sink received 536 bytes from 192.148.1.2 port 49153 total Rx 244824
bytes
At time +5.93997s packet sink received 176 bytes from 192.148.1.2 port 49153 total Rx 245000
bytes
At time +5.97247s on-off application sent 2500 bytes to 192.148.1.1 port 2600 total Tx 60000
bytes
At time +5.97307s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 58036
bytes
At time +5.97327s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 58572
bytes
At time +5.97352s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 59108
bytes
At time +5.97373s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 59644
bytes
At time +5.97394s packet sink received 356 bytes from 192.148.2.3 port 49153 total Rx 60000
bytes
At time +6.01247s on-off application sent 2500 bytes to 192.148.1.1 port 2600 total Tx 62500
bytes
At time +6.01307s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 60536
bytes
At time +6.01333s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 61072
bytes
At time +6.01353s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 61608
bytes
```

Fig.24

REPORT RISPOSTE HOMEWORK N.4

Telecomunicazioni 2022

Calcolo il throughput medio come:

$$Th = 500.000 \text{ bit} / (6 - 5) \text{ s} = 500.000 \text{ bit/s (Fig.25)}$$

Dove 5s è l'istante di tempo in cui l'on-off client inizia a trasmettere i dati.

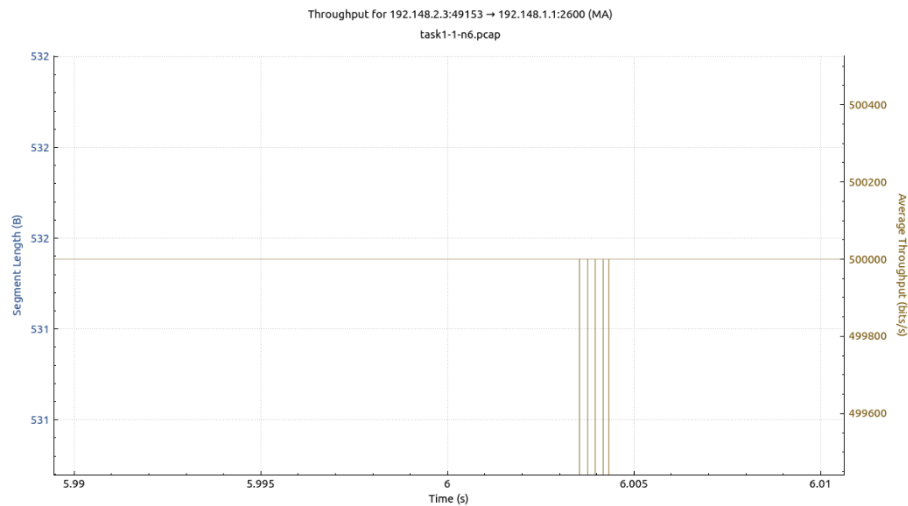


Fig.25

C13) Calcolare il throughput medio del flusso TCP n8 verso n0 a tempo t=8s. Commentare eventuali cambiamenti rispetto a C12.

Per rispondere alla domanda è stato abilitato il log "LogComponentEnable" per conoscere la dimensione dei byte trasmessi in ogni istante di tempo.

All'istante di tempo 8s sono stati trasmessi Tx = 187.500 bytes = 1.500.000 bit (Fig.26)

```
student@tlc-virtual-machine: ~/ns-3-dev-glt
At time +7.97328s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 183572
bytes
At time +7.97354s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 184108
bytes
At time +7.97375s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 184644
bytes
At time +7.97397s packet sink received 356 bytes from 192.148.2.3 port 49153 total Rx 185000
bytes
At time +8.01248s on-off application sent 2500 bytes to 192.148.1.1 port 2600 total Tx 187500
bytes
At time +8.01307s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 185536
bytes
At time +8.01331s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 186072
bytes
At time +8.01352s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 186608
bytes
At time +8.01379s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 187144
bytes
At time +8.01394s packet sink received 356 bytes from 192.148.2.3 port 49153 total Rx 187500
bytes
At time +8.01748s on-off application sent 5000 bytes to 192.148.2.2 port 7777 total Tx 375000
bytes
At time +8.01808s packet sink received 536 bytes from 192.148.1.2 port 49153 total Rx 370536
bytes
```

Fig.26

Calcolo il throughput medio come:

$$Th = 1.500.000 \text{ bit} / (8 - 5) \text{ s} = 500.000 \text{ bit/s (Fig.27)}$$

Dove 5s è l'istante di tempo in cui l'on-off client inizia a trasmettere i dati

REPORT RISPOSTE HOMEWORK N.4

Telecomunicazioni 2022

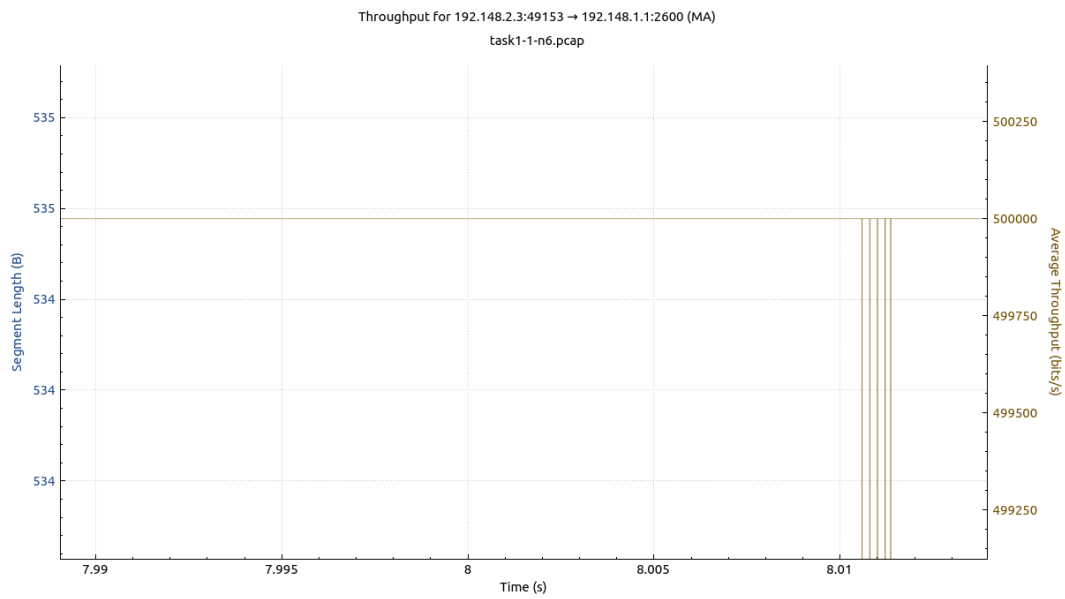


Fig.27

REPORT RISPOSTE HOMEWORK N.4

Telecomunicazioni 2022

Configuration n.2

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

Prendendo come riferimento lo schema riportato in Fig. 1, descriviamo la topologia della rete avendo il parametro "configuration=2".

Agli estremi della rete sono presenti due reti distinte di tipo CSMA, organizzate nel seguente raggruppamento:

- CSMA SX: n0, n1, n2;
- CSMA DX: n6, n7, n8;

Procedendo da sinistra verso destra, continuando nella descrizione della struttura della rete sono presenti:

- tra i nodi n2 e n3 un collegamento Point-to-Point;
- tra i nodi n3 e n4 un collegamento Point-to-Point (I0);
- tra i nodi n3 e n5 un collegamento Point-to-Point (I1);
- tra i nodi n6 e n4 un collegamento Point-to-Point (I3);
- tra i nodi n6 e n5 un collegamento Point-to-Point (I2);

Considerando la struttura di rete sopra descritta, sono presenti nella rete:

- UDP Echo Server installato sul nodo n2;
- UDP Echo Client installato sul nodo n8;
- TCP Sink installato sul nodo n0;
- UDP Sink installato sul nodo n7;
- TCP OnOff Client installato sul nodo n8;
- UDP OnOff Client installato sul nodo n8;

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.

Per rispondere a questa domanda sono stati generati dei file *pcap* su ogni singolo collegamento di ogni nodo. Tuttavia, nel codice allegato sono stati presi in considerazione solamente i nodi n3, n6 per la generazione dei file *pcap*, come da consegna. Sono state omesse le immagini dell'analisi su Wireshark per non essere pedanti, e non sono stati presi in considerazione i pacchetti con protocollo *zebra*.

Distinguo 3 flussi:

- n8 -> n0

n8 spedisce a n0 4 pacchetti con protocollo TCP

(filtro: `!(zebra) and tcp and ip.src==198.142.2.3 and ip.dst==198.142.1.1`)

Passando attraverso il nodo 5.

- n8 -> n7

n8 spedisce a n7 208 pacchetti con protocollo UDP

(filtro: `!(zebra) and udp and ip.src==198.142.2.3 and ip.dst==198.142.2.2`)

Il nodo 7 riceve 208 pacchetti udp, quindi non si verificano perdite.

REPORT RISPOSTE HOMEWORK N.4

Telecomunicazioni 2022

- n8 -> n2

n8 spedisce a n2 5 pacchetti con protocollo UDP

(filtro: *!(zebra) and udp and ip.src==198.142.2.3 and ip.dst==198.142.1.3*)

I pacchetti passano attraverso il nodo 5 e arrivano al nodo 2 senza perdite

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

RTT è l'acronimo di Round Trip Time, ovvero il tempo che passa da quando viene inviato il segmento TCP a quando viene ricevuto l'ACK del segmento stesso.

Per calcolare l'RTT abbiamo usato una funzionalità di WireShark che mostra il tempo che passa dall'invio di un pacchetto alla ricezione dell'ACK.

Nel primo flusso di dati da n8 ad n0: L'HandShake inizia al secondo 0,018469, quando n0 riceve l'ACK da n8. Per il primo segmento inviato, finisce al secondo 0,067955, quando n8 riceve l'ACK mandato precedentemente, più la dimensione del segmento (ACK = 537, Fig.28). L'RTT è quindi $0,067955 - 0,018469 = 49,486$ ms.



Fig.28

Questo è un singolo caso, ma andando ad analizzare altri casi ed in più avvalendoci del grafico fornito sempre da WireShark (Fig.29) è evidente come l'RTT sia un valore stimabile in un intorno di 49 ms.



Fig.29

REPORT RISPOSTE HOMEWORK N.4

Telecomunicazioni 2022

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

In una rete il fenomeno del bottleneck si verifica quando un collegamento riceve una quantità di dati superiore al suo data rate.

In questa configurazione i 3 flussi esistenti partono tutti da n8 e arrivano a n0, a n7 e a n2. Mentre il flusso tra n8 e n7 non incontra alcun bottleneck, i restanti due flussi invece ne incontrano entrambi uno, già discusso nelle due configurazioni precedenti (si veda A04 e A14), presente nel passaggio dal link n2-n3 al link CSMA.

Per risolvere un bottleneck la soluzione principale consiste nel cambiare il collegamento con uno che permetta un data rate maggiore. È un'ottima soluzione, ma molto costosa. Invece una possibile contromisura è già presente nel protocollo TCP e consiste nel ridurre la sliding window o addirittura chiuderla momentaneamente una volta che viene individuato il verificarsi di una perdita di pacchetti, non è una vera e propria soluzione, ma è sicuramente una contromisura efficace

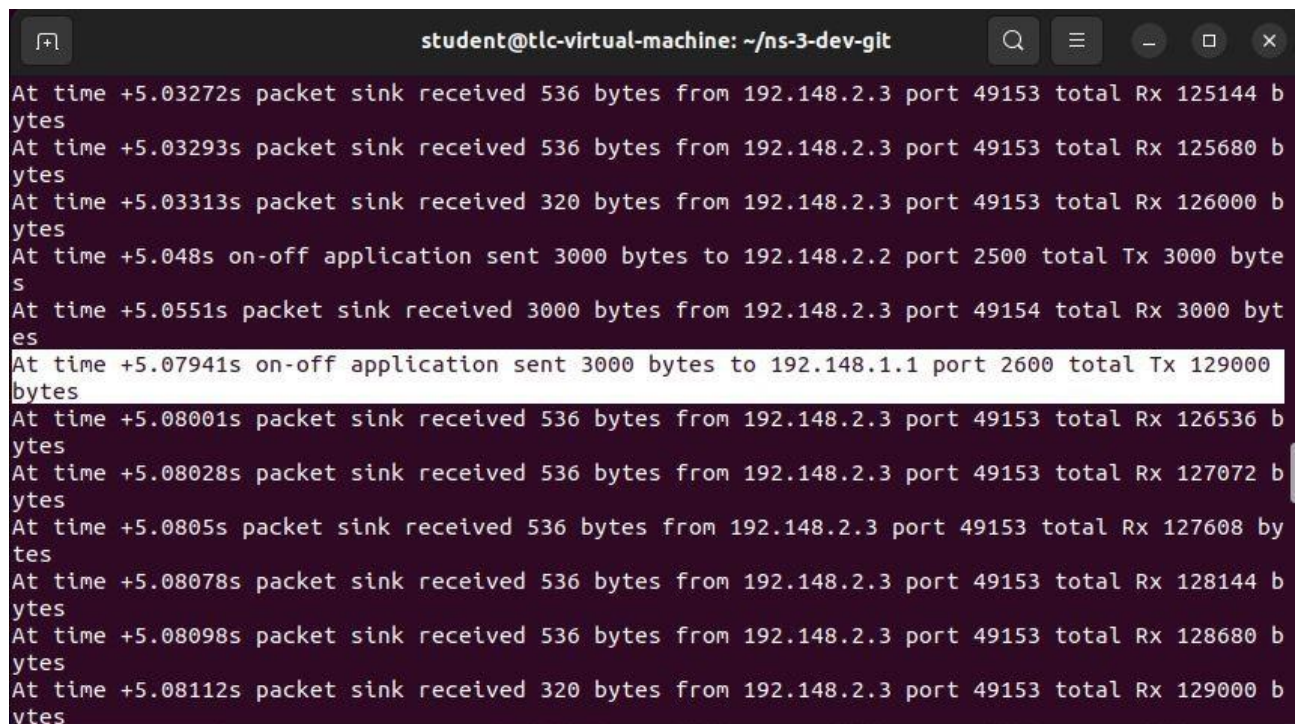
Domande:

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

Per rispondere alla domanda è stato abilitato il log "LogComponentEnable" per conoscere la dimensione dei byte trasmessi in ogni istante di tempo.

- n8 -> n0

All'istante di tempo 5s sono stati trasmessi $T_x = 129.000 \text{ bytes} = 1.032.000 \text{ bit}$ (Fig.30)



```
student@tlc-virtual-machine: ~/ns-3-dev-git
At time +5.03272s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 125144 bytes
At time +5.03293s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 125680 bytes
At time +5.03313s packet sink received 320 bytes from 192.148.2.3 port 49153 total Rx 126000 bytes
At time +5.048s on-off application sent 3000 bytes to 192.148.2.2 port 2500 total Tx 3000 bytes
At time +5.0551s packet sink received 3000 bytes from 192.148.2.3 port 49154 total Rx 3000 bytes
At time +5.07941s on-off application sent 3000 bytes to 192.148.1.1 port 2600 total Tx 129000 bytes
At time +5.08001s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 126536 bytes
At time +5.08028s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 127072 bytes
At time +5.0805s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 127608 bytes
At time +5.08078s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 128144 bytes
At time +5.08098s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 128680 bytes
At time +5.08112s packet sink received 320 bytes from 192.148.2.3 port 49153 total Rx 129000 bytes
```

Fig.30

REPORT RISPOSTE HOMEWORK N.4

Telecomunicazioni 2022

Calcolo il throughput medio come:

$$Th = 1.032.000 \text{ bit} / (5 - 3) \text{ s} = 516.000 \text{ bit/s}$$

Dove 3s è l'istante di tempo in cui l'on-off client inizia a trasmettere i dati.

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

Per rispondere alla domanda è stato abilitato il log "LogComponentEnable" per conoscere la dimensione dei byte trasmessi in ogni istante di tempo.

- n8 -> n0

All'istante di tempo 7s sono stati trasmessi Tx = 252.000 bytes = 2.016.000 bit (Fig.31)

```

student@tlc-virtual-machine: ~/ns-3-dev-git
At time +7.00063s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 248144 b
ytes
At time +7.00084s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 248680 b
ytes
At time +7.00098s packet sink received 320 bytes from 192.148.2.3 port 49153 total Rx 249000 b
ytes
At time +7.00178s server received 2560 bytes from 192.148.2.3 port 49153
At time +7.00178s server sent 2560 bytes to 192.148.2.3 port 49153
At time +7.00311s client received 2560 bytes from 10.1.1.1 port 63
At time +7.016s on-off application sent 3000 bytes to 192.148.2.2 port 2500 total Tx 126000 by
tes
At time +7.01704s packet sink received 3000 bytes from 192.148.2.3 port 49154 total Rx 126000
bytes
At time +7.04741s on-off application sent 3000 bytes to 192.148.1.1 port 2600 total Tx 252000
bytes
At time +7.04801s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 249536 b
ytes
At time +7.04822s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 250072 b
ytes
At time +7.04843s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 250608 b
ytes
At time +7.04864s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 251144 b
ytes
At time +7.04885s packet sink received 536 bytes from 192.148.2.3 port 49153 total Rx 251680 b

```

Fig.31

Calcolo il throughput medio come:

$$Th = 2.016.000 \text{ bit} / (7 - 3) \text{ s} = 504.000 \text{ bit/s}$$

Dove 3s è l'istante di tempo in cui l'on-off client inizia a trasmettere i dati.