



SAPIENZA
UNIVERSITÀ DI ROMA

Telecomunicazioni, a.a. 2021/2023

Report Homework N° 1

Gruppo N° 59

Risposte Gruppo A:

A1 conf 0,1,2)

Nella configurazione di rete proposta si possono distinguere due generi di topologie di rete:

- (1) Topologia a BUS, caratterizzata dall'implementazione del protocollo CSMA (Carrier Sense Multiple Access) allo strato MAC (Livello 2 modello ISO/OSI), con la quale si realizza una configurazione di rete nota come LAN (Local Area Network);
- (2) Topologia Point to Point, con la quale si realizzano collegamenti dedicati tra due nodi della rete, chiamati per tale motivo EndPoint.

Con riferimento alla Figura A.1, si possono associare le topologie sopra citate alle seguenti aree di network:

area 0: costituita dalle interfacce NIC interconnesse secondo la topologia (1), dei nodi n0, n1, n2;

area 1: costituita dalle interfacce NIC interconnesse secondo la topologia (2), dei nodi n2, n3, n4, n5, n6, che globalmente possono essere anche definite afferenti ad una topologia "parzialmente magliata";

area 2: costituita dalle interfacce NIC interconnesse secondo la topologia (1), dei nodi n6, n7, n8.

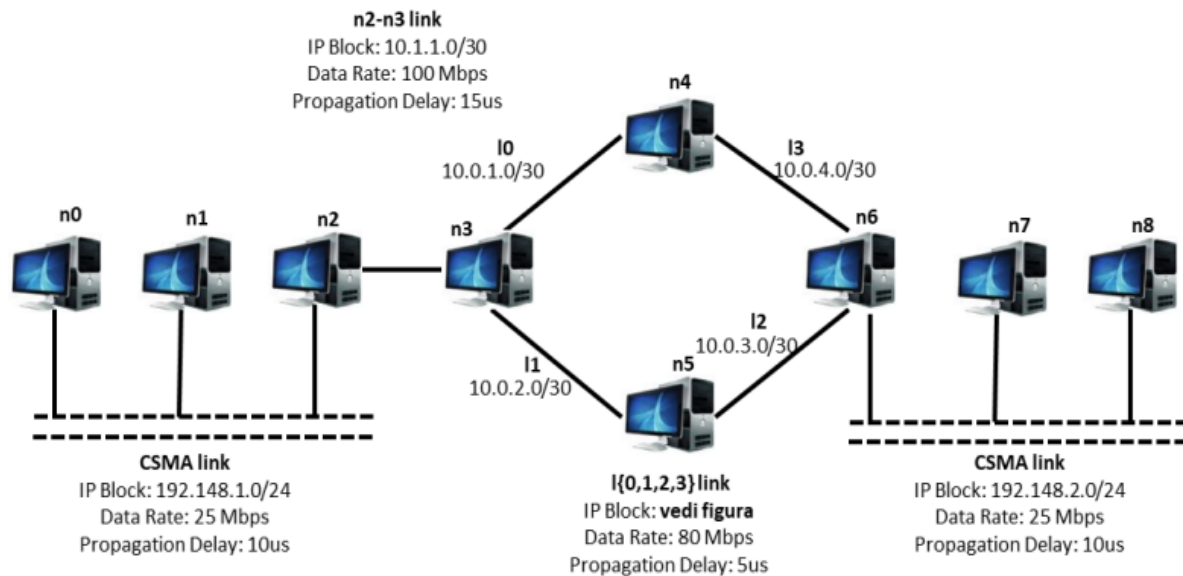


Figura A.1

A2 conf 0)

Per la configurazione 0 si evince da una disamina dei file .pcap, generati dal nodo n6 ed n3 su ciascuna delle proprie interfacce di rete, che la totalità (100%) dei segmenti TCP contenenti dati inviati da client a server passano per i link Point to Point N3-N4 N4-N6, mentre nei link N3-N5 N5-N6 passano solamente segmenti del protocollo di gestione TCP, ovvero segmenti con flag settati ad ACK / SYN / FIN.

A2 conf 1)

Per la configurazione 1 si evince da una disamina dei file .pcap, generati dal nodo n3 e n6, allo stesso modo della precedente risposta, che i due flussi di dati TCP afferenti alla comunicazione n8 -> n0 e n1 -> n7, compongono il 60% e 40% della totalità dei segmenti scambiati rispettivamente. Tale distinzione è stata fatta impostando il filtro "tcp.port == 7777".

Per quanto concerne la distinzione dei percorsi seguiti dai segmenti all'interno della rete, dalla stessa analisi, ma su tracce .pcap di interfacce differenti, si conclude che



SAPIENZA
UNIVERSITÀ DI ROMA

il traffico generato dai nodi client e server è ripartito al 55% circa sui link N3-N4 N4-N6 e al 45% circa sui link N3-N5 N5-N6.

A2 conf 2)

Per la configurazione 2 si evince da una disamina dei file .pcap, generati dal nodo n3 e n6, allo stesso modo della precedente risposta, che i flussi di dati TCP ed UDP afferenti alle comunicazioni n8 -> n0 , n8 -> n7 e n8 -> n2, son composti dal 60% circa della totalità dai segmenti scambiati con il protocollo TCP, mentre solamente il 12% circa da quelli del protocollo UDP. I restanti pacchetti sono scambiati dagli strati inferiori della rete (ipv4 e ARP).

Questa situazione trova riscontro dal momento che è possibile notare, con un'attenzione particolare ai segmenti UDP, che questi ultimi, al contrario dei segmenti TCP, non vengono frammentati in pacchetti di dimensione inferiore a quella specificata in fase di progetto, bensì inviati in blocchi unici, come richiesto.

Per distinguere i due possibili percorsi che i pacchetti seguono all'interno della rete si è usato il filtro Wireshark "tcp || udp", che unitamente all'analisi dei file .pcap relativi alle altre interfacce di rete del nodo n6, hanno rivelato i seguenti risultati:

il 70% circa del totale del traffico utile scambiato tra i nodi in esame è imputabile ai link N3-N4 N4-N6,

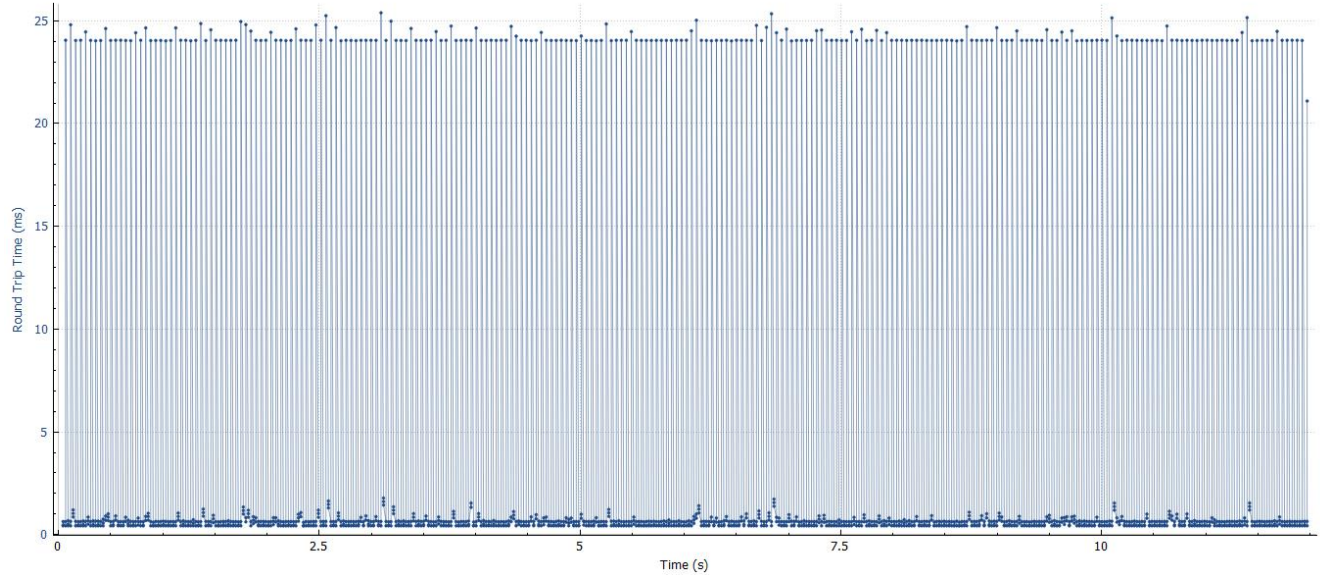
il restante 30% circa è invece imputabile ai link N3-N5 N5-N6.

A3 conf 0)

Secondo gli strumenti di statistica messi a disposizione dal tool Wireshark, impiegando un filtro sul protocollo TCP, si stima un RTT di 24 [ms] sul flusso TCP n8 - n0.



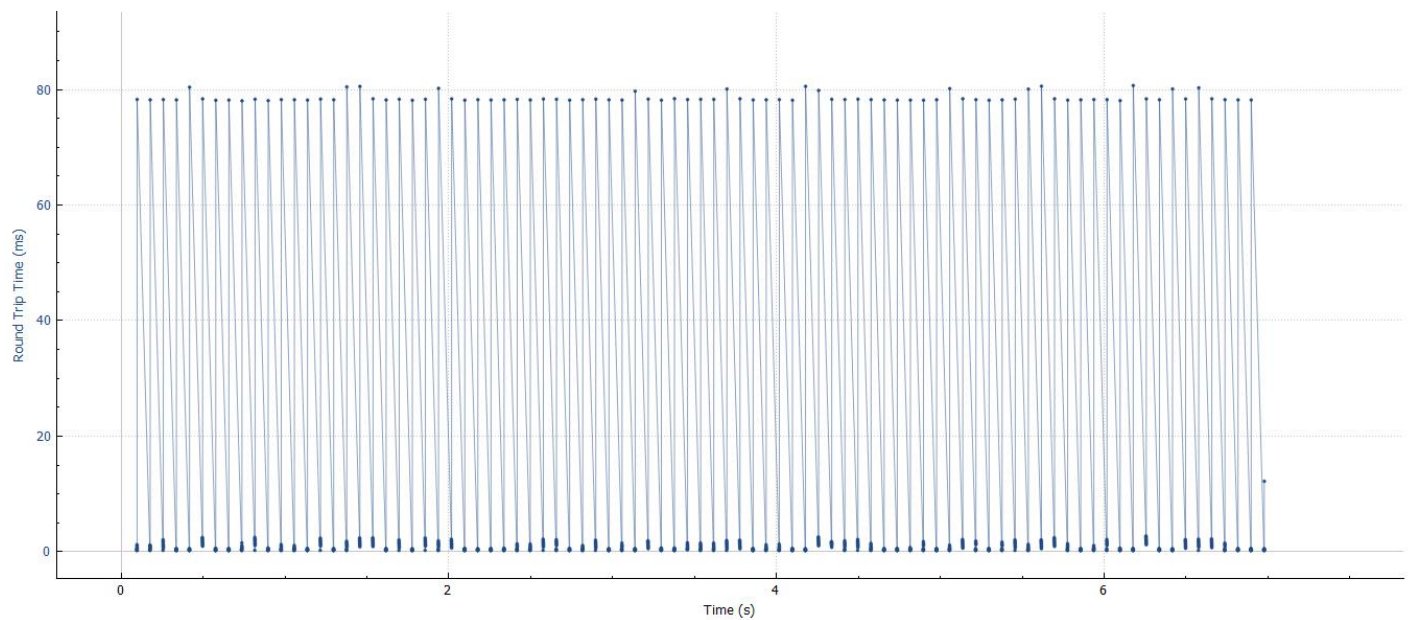
Round Trip Time for 192.148.2.3:49153 → 192.148.1.1:2600
task1-0-n6-6-2.pcap



A3 conf 1)

Secondo gli strumenti di statistica messi a disposizione dal tool Wireshark, impiegando un filtro sul protocollo TCP, si stima un RTT di 78 [ms] su entrambi i flussi TCP (n1 – n7 e n8 – n0).

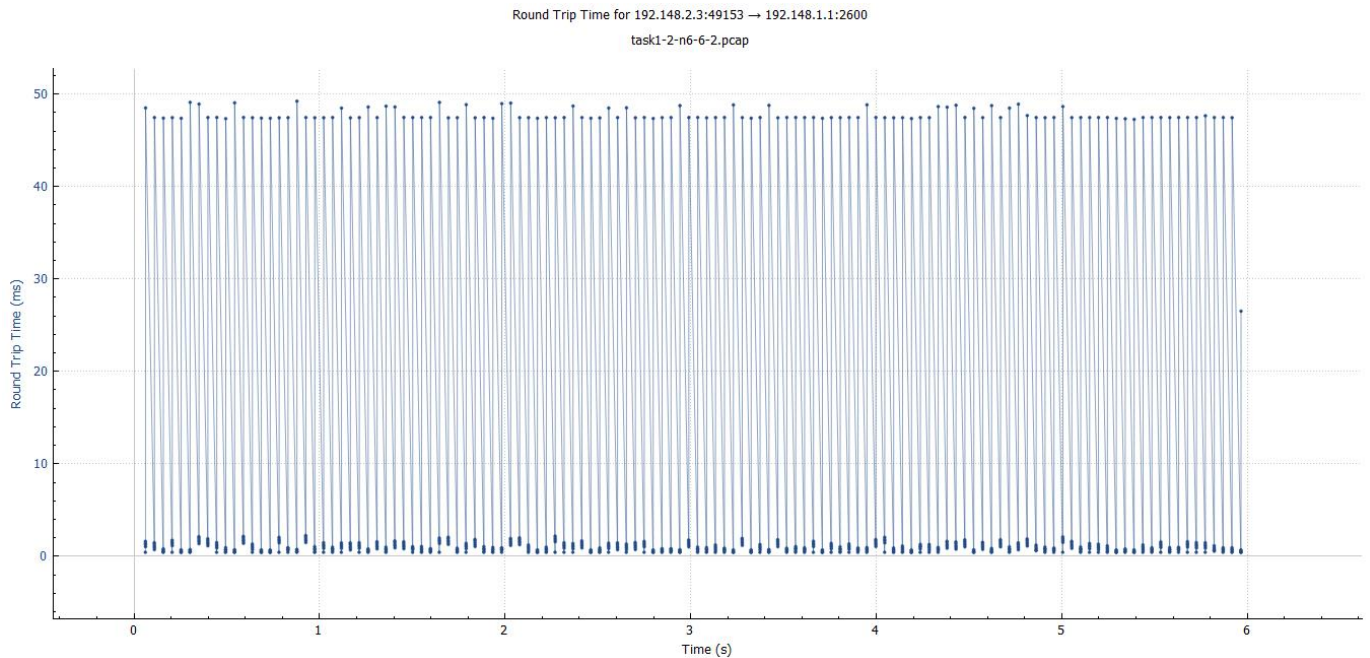
Round Trip Time for 192.148.1.2:49153 → 192.148.2.2:7777
task1-1-n6-6-2.pcap





A3 conf 2)

Secondo gli strumenti di statistica messi a disposizione dal tool Wireshark, impiegando un filtro sul protocollo TCP, si stima un RTT di 48[ms].



A4 conf 0,1,2)

La rete presenta canali di trasmissione con differenti proprietà fisiche, le quali per loro natura, creano dei collegamenti a velocità e ritardo variabile all'interno della rete, al pari delle caratteristiche dei nodi.

Pertanto, si può parlare di colli di bottiglia nei collegamenti con velocità di trasmissione inferiore o, altresì, al pari di velocità, nei collegamenti che presentano un delay maggiore.

Data la configurazione di cui in figura A.1, è riscontrabile una caratteristica di prestazione nei collegamenti con topologia a BUS inferiore rispetto a quella che si ha invece nei collegamenti con topologia Point to Point, che, in ogni caso, per via della loro natura di collegamenti dedicati, hanno al pari delle risorse una capacità prestazionale maggiore nell'invio dei dati tra EndPoint.



SAPIENZA
UNIVERSITÀ DI ROMA

Nonostante le caratteristiche intrinseche della configurazione di rete proposta, la disposizione dei nodi che effettivamente prendono parte nello scambio di quantità di dati rilevanti al fine della comunicazione (client e server), fanno intuire un impiego dei canali “al centro della rete” in modo massivo, con conseguente possibile sovraccarico di tali strutture trasmissive. Ciò porta a ritenere che una disposizione degli apparati client e server in maniera da ridurre tale traffico molto focalizzato su tale parte della rete, sia una possibile contromisura al problema.

La contromisura sovraesposta, contiene comunque criticità notevoli, in quanto non sempre è possibile ovviare al problema della dislocazione degli apparati client e server. A tal proposito si ritiene che un'altra possibile soluzione al problema di congestione della sezione di rete in esame sia quella di migliorare le caratteristiche dei collegamenti e dell'infrastruttura di rete nel suo complesso, con l'aggiornamento di topologie e protocolli che ne consegue.

Risposte Gruppo C:

C01)

Il flusso TCP tra client n8 e server n0, ha un Throughput Istantaneo, calcolato secondo i tempi interconnessi dall'invio del primo byte utile di un pacchetto da parte del client all'invio dell'ultimo riscontro ACK su tale pacchetto da parte del server.

Con tale modello, si ottiene un **risultato di 1500 Bytes / 0.0008 s.**

(i dati sono stati ricavati da wireshark impostando il filtro TCP sul file .pcap relativo all'interfaccia di rete con indirizzo 192.148.2.3 del nodo n6):

8 0.043602 192.148.2.3 ->192.148.1.1 ZEBRA 606 Zebra
Request

12 0.044457 192.148.1.1 ->192.148.2.3 TCP 70 2600 → 49153
[ACK] Seq=1 Ack=1501 Win=131072 Len=0 TSval=3051 TSecr=3050



Considerazione: I dati ottenuti facendo riferimento ai modelli grafici dello strumento Wireshark, trovano altresì riscontro con la teoria:

$$X(t) = \text{Window}(t) / RTT$$

Throughput for 192.148.2.3:49153 → 192.148.1.1:2600 (MA)
task1-0-n6-6-2.pcap

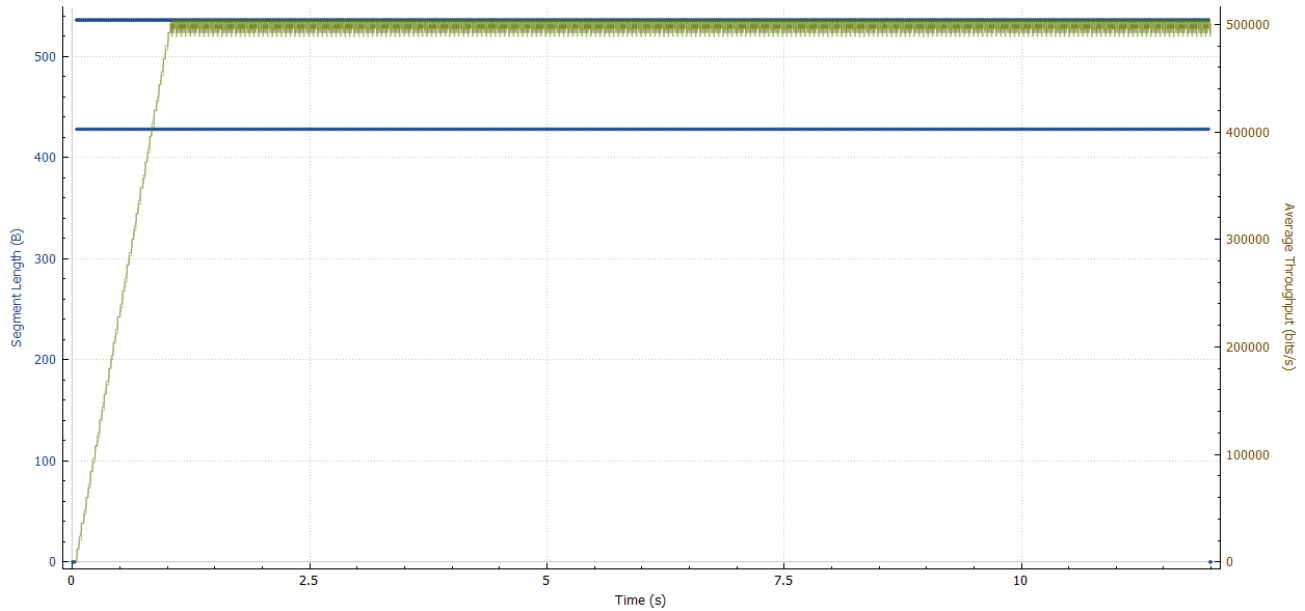


Figura C.0.1: Whireshark statistics > tcp stream graph > throughput

C02)

All'istante $t = 4.0$ s, si ottiene un Throughput medio, calcolato a partire dal file .pcap di tracing, dividendo il valore del sequence number dei bytes trasmessi dal client al server per il valore di istante di tempo in considerazione.

188 1.003602 192.148.2.3 192.148.1.1 ZEBRA 606 Zebra Request
Seq. Number: 60001

Ottenendo così 60001 Bytes/s.

Nota: nel record il tempo 1.003602 è inteso come relativo dall'istante di inizio di trasmissione del primo byte da parte del client, il quale è da intendersi come tempo



SAPIENZA
UNIVERSITÀ DI ROMA

assoluto $t = 4s$. Nel seguito del presente report si intenderà la suddetta considerazione come implicita dovuta al contesto.

C03)

Considerata l'intuizione precedente sul calcolo del Throughput medio all'istante richiesto, si effettua il calcolo seguendo lo stesso procedimento all'istante di tempo $t=7.0s$.

Si ottiene il risultato: 61875 Bytes/s.

Commento: la differenza del valore di throughput ai due istanti di tempo selezionati è plausibilmente imputabile alle condizioni di controllo di flusso e congestione della rete, nonché ai differenti percorsi che i dati possono seguire nei link dell'infrastruttura in questione.

C04)

RP_i = ritardo di propagazione link i [s]

L = lunghezza media di un pacchetto [bit]

BR_i = bit rate del link i [bit/s]

$$\text{Ritardo di Trasferimento} = \sum_{i=\text{link}} (RP_i) + (L) / (BR_i)$$

Considerando Ritardi di elaborazione e accodamento trascurabili.

Il ritardo di trasferimento per ogni pacchetto è quindi:

$$0.00017 + 0.00017 + 0.000055 + (0.0000505 * 2) [s]$$

Tale risultato è riscontrabile graficamente facendo il confronto tra il timestamp dell'invio di un segmento da parte del client con il timestamp al momento in cui il server lo riceve.

Considerando inoltre i timestamp dei pacchetti loggati nei file .tr dei nodi client e server, in aggiunta al considerare il tempo di simulazione effettivo intercorso



dall'istante in cui il client inizia ad inviare il primo byte all'istante in cui il server lo riceve, si evince che il **Ritardo di Trasferimento complessivo** della totalità dei pacchetti inviati tra client e server con protocollo TCP, sia di circa 11.993 [s].

C11)

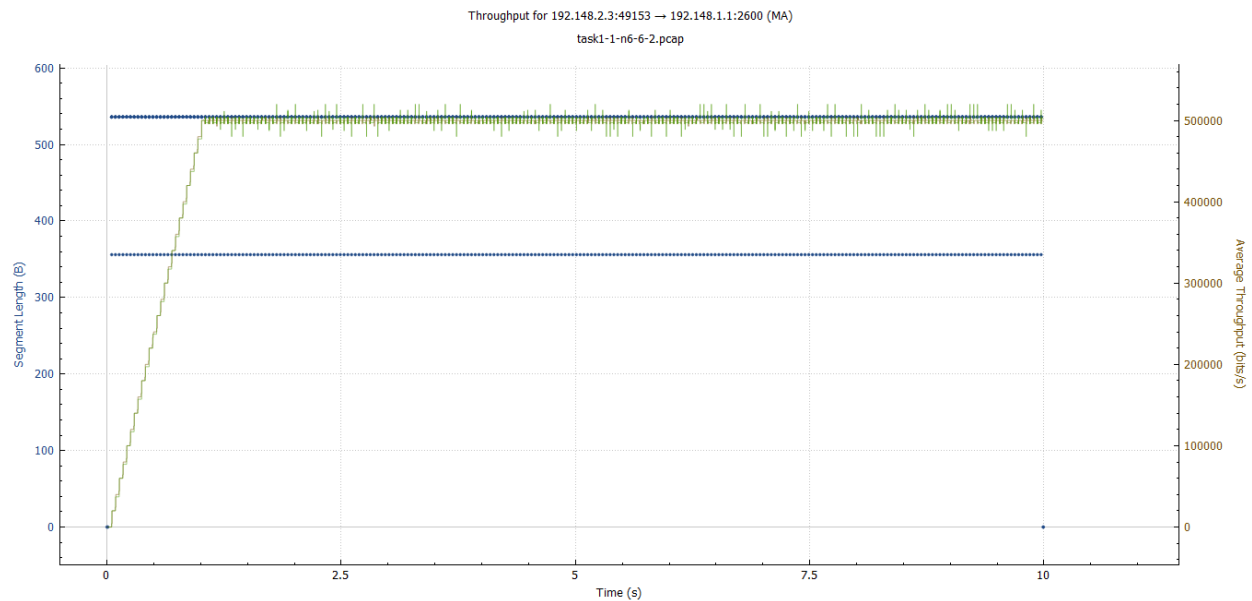


Figura C.1.1: Wireshark statistics > tcp stream graph > throughput

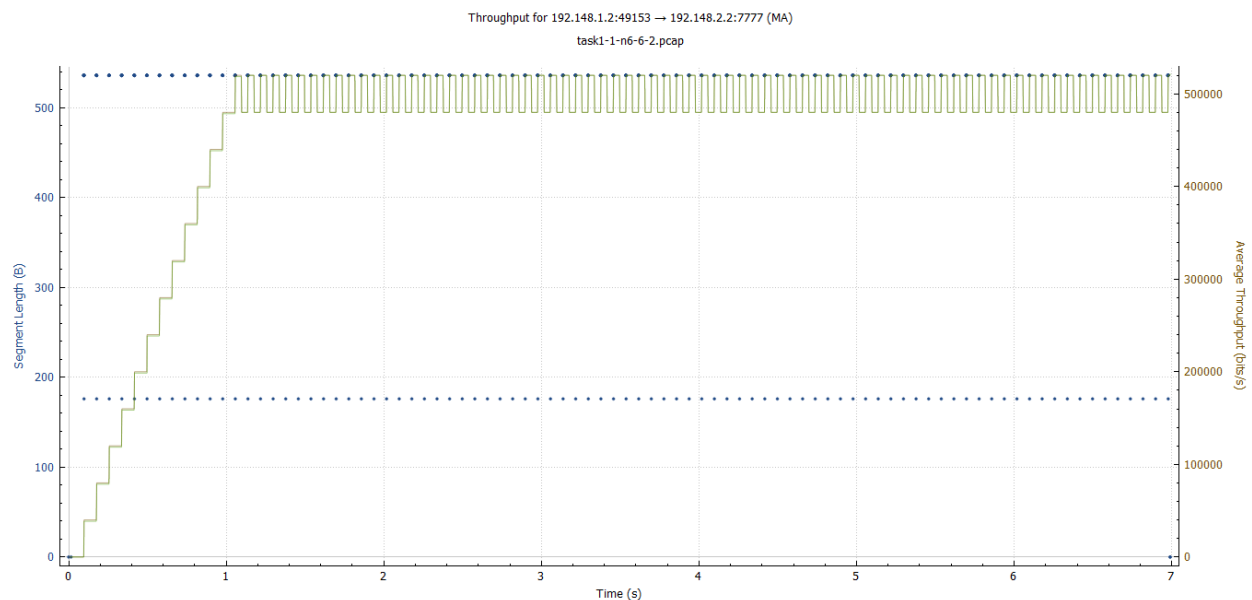


Figura C.1.2: Wireshark statistics > tcp stream graph > throughput



I due flussi di comunicazione basati su protocollo TCP permettono di ricavare il Throughput totale medio, operando la media aritmetica dei singoli flussi individuabili dalle comunicazioni client server afferenti ai nodi in questione ($n1 \rightarrow n7$, $n8 \rightarrow n0$).

A tale scopo, si opera su Wireshark con il filtro "tcp.port==2600" per estrapolare uno dei due flussi sopra citati ($n8 \rightarrow n0$), grazie al quale si può riscontrare, facendo anche riferimento alla Figura C.1.1, un throughput di circa 60000 Bytes/s.

Con ragionamento analogo si impiega il filtro "tcp.port==7777", che con la medesima ricostruzione, verificabile in Figura C.1.2, restituisce per tale flusso, al pari del precedente, un throughput medio richiesto di 60000 Bytes/s.

C12)

All'istante di tempo richiesto $t = 6.0s$, da una disamina delle tracce .pcap inerenti al nodo $n6$, con particolare riferimento ai Sequence Number dei segmenti inviati dal client $n8$ al server $n0$, è possibile calcolare, come già proposto nella configurazione precedente il throughput medio al dato istante come rapporto tra lo stesso Sequence Number per il tempo di simulazione.

Si ottiene con tale criterio il valore di throughput medio pari a 60001 Bytes/s.

C13)

All'istante di tempo richiesto $t = 8.0s$, facendo riferimento al metodo di calcolo proposto nella risposta precedentemente illustrata, si ottiene, per il medesimo flusso TCP client/server, il valore di Throughput medio pari a 61667 Bytes/s.

Da notare, come anche riscontrato nella configurazione precedente, che si assiste ad un incremento del valore di cui si è richiesto il calcolo. Questo è plausibilmente imputabile alle specifiche di protocollo del TCP, che opera modifiche al flusso in base alle condizioni dei nodi in comunicazione e allo stato della rete.

Il Seq. Number utilizzato è 185001.



C14)

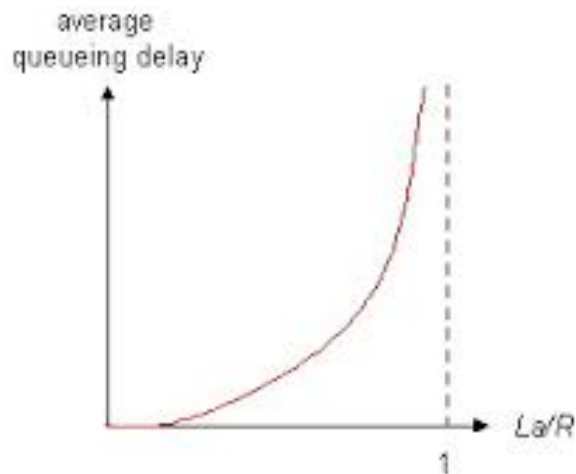
Considerato:

L = lunghezza pacchetto [bit]

R = larghezza di banda del canale [bit/s]

a = velocità media di arrivo dei pacchetti

Dalla teoria, si richiama al grafico sottostante, in cui si può notare come il **tempo di attesa nelle code dei dispositivi** aumenti in modo esponenziale all'aumentare della congestione della rete di trasmissione.



Secondo il protocollo TCP, il quale fa uso del controllo di congestione per adeguare la velocità trasmissiva in capo al mittente della connessione, si può inoltre riscontrare nei grafici messi a disposizione dal tool Wireshark un andamento di tale parametro che è in accordo con quanto descritto dalla teoria. Si riscontra cioè un andamento esponenziale della velocità trasmissiva fino al raggiungimento della soglia (Slow Start Threshold) seguita da un andamento lineare fino alla prima perdita della conferma sul pacchetto trasmesso.



C21)

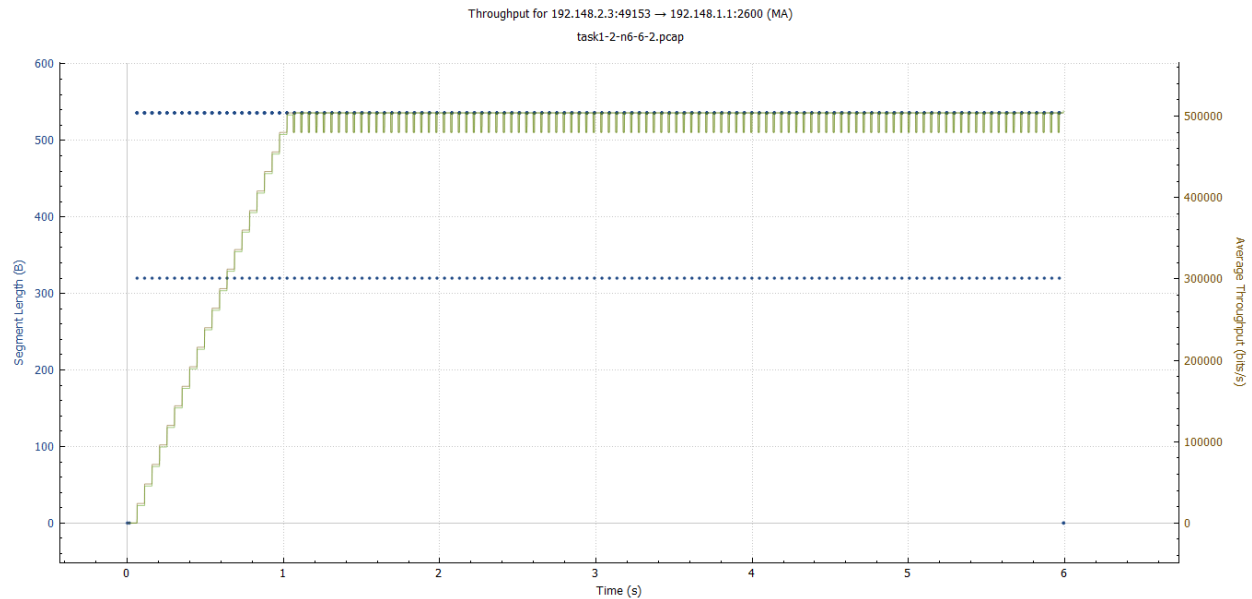


Figura C.2.1: Wireshark statistics > tcp stream graph > throughput

Come anche nelle risposte precedenti in merito al calcolo del Throughput medio dei flussi TCP, si opera a partire dalla disamina delle tracce .pcap, che con lo stesso metodo che ha portato ai risultati precedenti, ovvero calcolare il Throughput medio dividendo il Sequence Number dei Bytes dei segmenti trasmessi fino all'istante di tempo richiesto, per l'istante di tempo stesso, porta al seguente risultato:

al tempo $t = 5.0s$, impiegando il segmento con Seq. Number 123001, si ottiene il valore di Throughput di 61500 Bytes/s.

C22)

Come nei casi precedenti, applicando lo stesso modello di calcolo con Seq. Number di riferimento 249001, si ottiene il valore di Throughput medio di 61667 Bytes/s, con leggero scostamento dal valore calcolato all'istante di cui alla C22 dovuto ai motivi ampiamente esposti nelle precedenti discussioni dello stesso ambito.

C23)

Per questa domanda si rimanda alla risposta C14).



SAPIENZA
UNIVERSITÀ DI ROMA

Considerazioni Generali in merito al protocollo TCP:

nell'ambito della presente simulazione di infrastruttura di rete realizzata con l'applicativo NS3, si è riscontrato un uso massivo di un protocollo afferente allo strato applicativo, denominato ZEBRA.

Tale protocollo è stato inteso, a seguito di una attenta e premurosa disamina pacchetto per pacchetto all'interno dello strumento Wireshark, di vitale importanza al fine dell'inoltro dei segmenti in protocollo TCP da parte dei client ai server.

È stato infatti possibile notare, che per molte, se non tutte, le comunicazioni che avvengono secondo TCP, i dati vengono mandati con il suddetto protocollo ZEBRA, il quale, quindi, si è dovuto intendere come TCP al fine di espletare i calcoli di cui nelle risposte sopra riportate si è visto maggior dettaglio.