

Progetto GNS3 - Elementi di Reti di Telecomunicazione

Valperta Samuele

Gennaio 2024

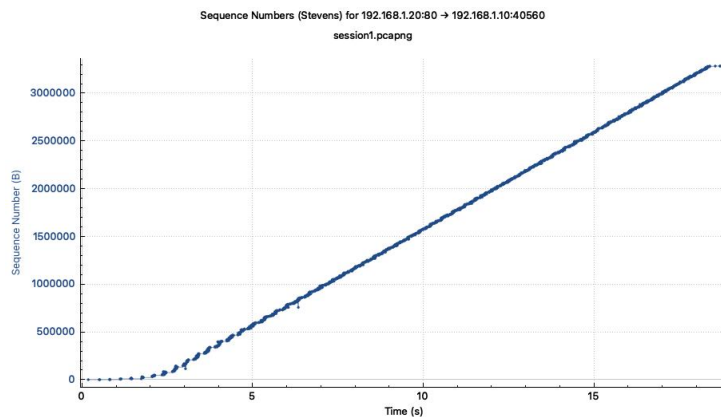
1 Introduzione

Abbiamo realizzato un semplice progetto in GNS3 dove troviamo 2 client (C1 e C2) e 2 server (S1 e S2), tutti collegati alla stessa rete tramite un unico switch. Procediamo con i seguenti esperimenti:

1. C1 richiede un documento a S1
2. C2 richiede un documento a S2
3. C1 e C2 richiedono simultaneamente un documento, rispettivamente ad S1 e S2

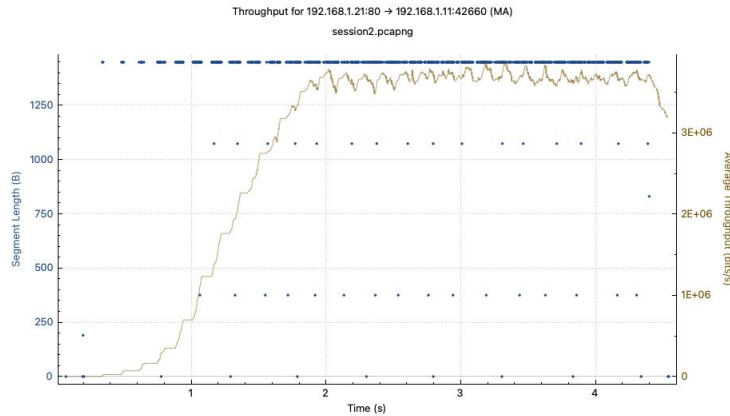
È importante notare che le richieste dei client sono state effettuate tramite HTTP, un protocollo applicativo che utilizza il protocollo TCP per la trasmissione dei dati. Le nostre catture infatti sono state fatte su tutti i pacchetti TCP tra i server e lo switch. Andiamo ora ad analizzare i diversi scenari sfruttando dei grafici generati Wireshark, uno strumento apposito per l'analisi del traffico di rete.

2 C1←S1



Analizziamo la prima cattura sfruttando il grafico appena mostrato dove troviamo il tempo su un asse e il sequence number sull'altro. Con questo grafico possiamo osservare l'andamento generale della comunicazione, notiamo che è un andamento lineare e senza particolari interruzioni, questo è derivato dal fatto che non ci sono state anomalie durante il trasferimento. Questo grafico purtroppo non ci permette di analizzare in modo dettagliato le varie fasi della comunicazione, a tale scopo utilizzeremo successivamente altri grafici. Notiamo però attraverso l'analisi dei pacchetti senza l'ausilio di grafici che per gran parte dei pacchetti abbiamo la dicitura "[TCP Out-Of-Order]", questo potrebbe indurci a pensare che ci siano stati errori durante la comunicazione quando invece non è così. Questa dicitura fa infatti riferimento alla situazione in cui i pacchetti arrivano a destinazione in un ordine differente da quello di partenza, questo non è un problema poichè il protocollo è strutturato in maniera tale da poter riassemblare correttamente i pacchetti prima inoltrarli al layer successivo.

3 C2←S2



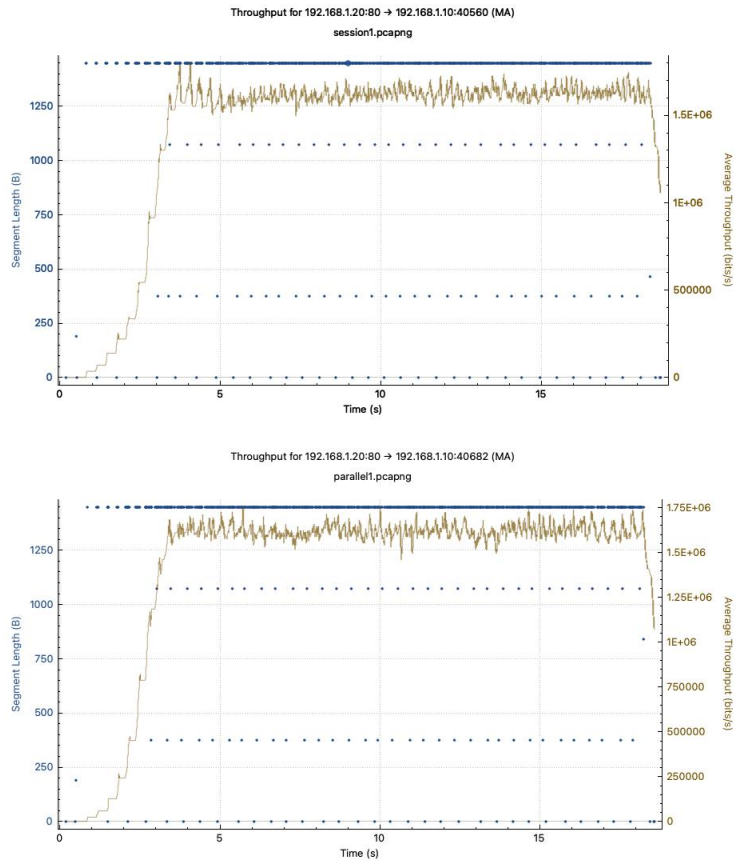
Passiamo ora all'analisi del secondo esperimento sfruttando il grafico di Throughput. Questo grafico ci permette un'analisi molto più dettagliata in quando abbiamo la quantità di dati inviati in relazione al tempo. La prima cosa che notiamo seguendo la linea gialla è che segue due andamenti molto differenti, questa differenza è data dai diversi algoritmo di gestione della dimensione della finestra (CWND). Viene utilizzato per i primi due secondi lo Slow Start, in questo caso per ciascun riscontro positivo ricevuto dopo l'invio di un pacchetto la CWND viene aumentata di una quantità definita dal MSS (Maximum Segment Size), questo fino al raggiungimento del SSTHRESH (Slow Start Treshold) quando la gestione della CWND viene regolata dall'algoritmo Congestion Avoidance che applica un incremento della CWND molto più lineare e controllato.

Notiamo inoltre come nei primi attimi della comunicazione non vengano trasferite grosse quantità di dati, questo è dato dalla Three-Way Handshake, ovvero una tecnica utilizzata per instaurare la connessione tra due host. Un

altro particolare che osserviamo da questo grafico è la strana ripetizione di pacchetti la cui segment length è 1073, 375 o 0 e questo è dato da un processo di frammentazione dei pacchetti legato al MTU (Maximum Transmission Unit).

4 C1←S1 e C2←S2 in parallelo

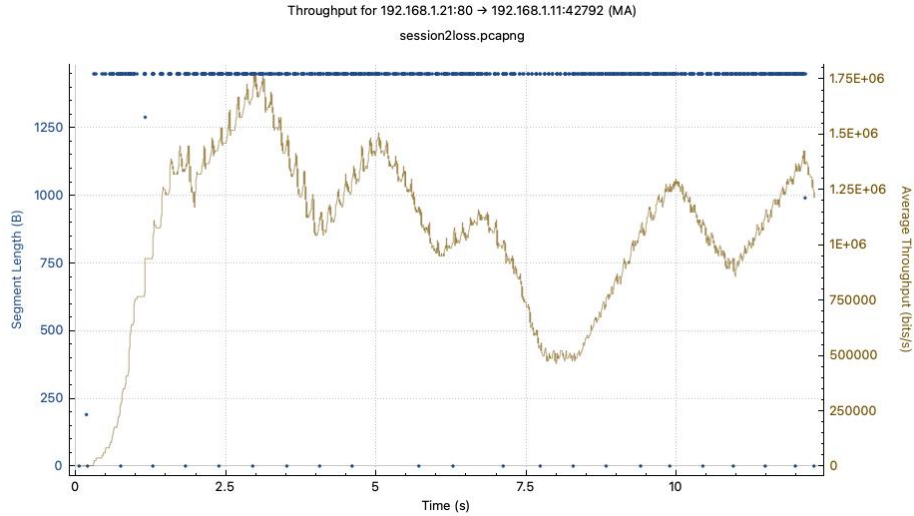
Per questo esempio prendiamo in considerazione i due grafici di throughput della trasmissione tra S1←C1 e li mettiamo a confronto.



Come possiamo facilmente notare non vi è quasi nessuna differenza tra i due esperimenti, questo poichè lo switch, in quanto collegato a pochi dispositivi non ha problemi di gestione del traffico, è in grado di lavorare in modalità full-duplex, ovvero può inviare e ricevere pacchetti in contemporanea e data la poca quantità di dati trasmessi non vi sono problemi relativi a collisioni.

5 C2←S2 con packet loss

Abbiamo eseguito anche un ulteriore esperimento, riconfigurando i vari dispositivi aggiungendo una piccola percentuale di possibilità di perdita dei pacchetti, pari all'1%.



Per quanto possa sembrare un'insignificante percentuale notiamo come il tempo di trasferimento totale nel nostro caso venga molto più che raddoppiato, questo è causato dagli algoritmi Fast Retransmit e Fast Recovery, utilizzati per correggere rapidamente la perdita di pacchetti. Questi algoritmi entrano in atto dopo la ripetuta ricezione di pacchetti "TCP Dup ACK", ad esempio i pacchetti da 684 a 689 (688 escluso) inviando un pacchetto FR, il 690. In questi casi viene dimezzata la CWND e ciò spiega il particolare andamento del grafico di throughput e l'aumento così significativo del tempo di trasmissione.

6 Conclusioni

Segue una tabella riassuntiva del tempo dei vari trasferimenti, è importante in generale notare come tutti questi valori siano dipendenti anche dalle latenze che vi sono tra i vari host, che noi abbiamo impostato manualmente per simulare il più possibile la realtà.

Session	Duration [s]	TPUT [kbit/s]
s1	18.83	1526
s2	4.62	3148
s1-par	18.70	1538
s2-par	4.53	3207