Relazione Assignment 1 Gestione di reti

Analisi del traffico di rete generato dal sito web fast.com con l'utilizzo di wireshark

A cura di Luca Miglior (580671) e Federico Ramacciotti (582646) - Gruppo 8

Svolgimento della cattura

La cattura è stata avviata prima del collegamento al sito fast.com

Il collegamento al sito fast.com determina l'inizio dello speedtest, che dura circa 20 secondi. Al termine dello speedtest, la cattura su wireshark è stata interrotta ed è stato analizzato il file .pcap generato per filtrare esclusivamente i pacchetti generati sulla rete dal sito web.

Filtro applicato al traffico e Endpoints

Per creare i filtri da applicare alla cattura, si è ragionato sul funzionamento di un'applicazione di speedtest: questa deve inviare attraverso la rete grandi quantità di dati, ad un limitato numero di host, per poterne misurare la capacità. Dunque, sotto questa ipotesi, sarà sufficiente individuare gli indirizzi ip che hanno inviato (e ricevuto) un consistente volume di dati: per fare ciò si utilizza la funzione di analisi statistica di wireshark, in particolare lo strumento Endpoints (Menu -> Statistics -> Endpoints):

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
192.168.1.3	59,005	65M	22,180	19M	36,825	46M
151.5.17.51	15,227	17M	10,105	13M	5,122	4357k
151.5.17.59	16,776	16M	7,971	7656k	8,805	8637k
45.57.73.142	12,446	14M	8,326	10M	4,120	3640k
45.57.75.168	7,983	9799k	6,119	8992k	1,864	806k
23.246.51.156	6,573	7464k	4,304	5614k	2,269	1850k

Si vede come durante la cattura wireshark abbia individuato cinque host principali che hanno generato molto traffico da (e verso) la macchina utilizzata; si parla infatti di circa 65MB di dati totali scambiati fra fast.com e l'host locale `192.168.1.3`

Dunque, per filtrare solo i pacchetti interessanti (nulla infatti garantisce che wireshark possa avere catturato anche altro traffico, non rilevente) si sono applicati questi filtri:

```
((((ip.addr == 151.5.17.59) || (ip.addr == 151.5.17.51)) || (ip.addr ==
45.57.75.168)) || (ip.addr == 45.57.73.142)) || (ip.addr == 23.246.51.156) .
```

Da queste statistiche è anche possibile capire il funzionamento di FAST.com: l'applicazione apre più connessioni parallele dall'host locale verso host remoti da lui selezionati (in questo caso cinque, ma dalle impostazioni del servizio è possibile scegliere il numero minimo e massimo di connessioni desiderate per la misura) e misura il bitrate in funzione del tempo impiegato dal server (o dall'host locale) per ricevere una quantità nota di dati.

Dopo la cattura è possibile effettuare reverse dns su questi indirizzi ip, e si nota come questi appartengano tutti a domini del tipo `.nflxvideo.net` o `.netflix.com`

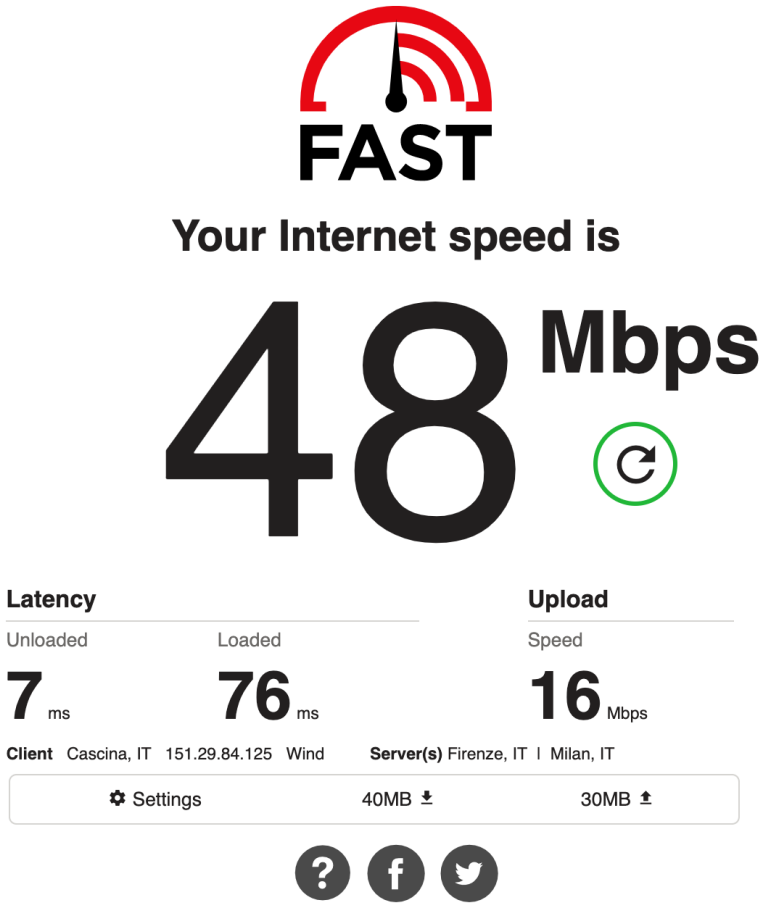
Analisi dei pacchetti

I pacchetti filtrati sono tutti e soli quelli generati dallo speedtest. Essi si presentano come pacchetti TCP/TLS, come atteso, dal momento che fast.com utilizza una connessione HTTPS per lo scambio dei dati col client. Si può dire quindi che applicare un filtro TCP o un filtro sulla porta 443 non avrebbe impedito la cattura di altro traffico, oltre a quello di fast.com.

0.066775	vornao.local	151.5.17.59	TCP	66 58814 → 443 [ACK] Seq=10
0.066825	vornao.local	151.5.17.59	TCP	66 58814 → 443 [ACK] Seq=10
0.067386	vornao.local	151.5.17.59	TCP	66 [TCP Window Update] 5881
0.068013	151.5.17.59	vornao.local	TCP	1506 443 → 58814 [ACK] Seq=13
0.068021	151.5.17.59	vornao.local	TCP	1506 443 → 58814 [ACK] Seq=14
0.068024	151.5.17.59	vornao.local	TCP	1506 443 → 58814 [ACK] Seq=16
0.068114	vornao.local	151.5.17.59	TCP	66 58814 → 443 [ACK] Seq=10
0.068354	151.5.17.59	vornao.local	TCP	1506 443 → 58814 [ACK] Seq=17
0.068360	151.5.17.59	vornao.local	TCP	1506 443 → 58814 [ACK] Seq=19
0.068363	151.5.17.59	vornao.local	TLSv1.3	1506 Application Data [TCP se
0.068426	vornao.local	151.5.17.59	TCP	66 58814 → 443 [ACK] Seq=10
0.070119	vornao.local	151.5.17.59	TCP	66 [TCP Window Update] 5881

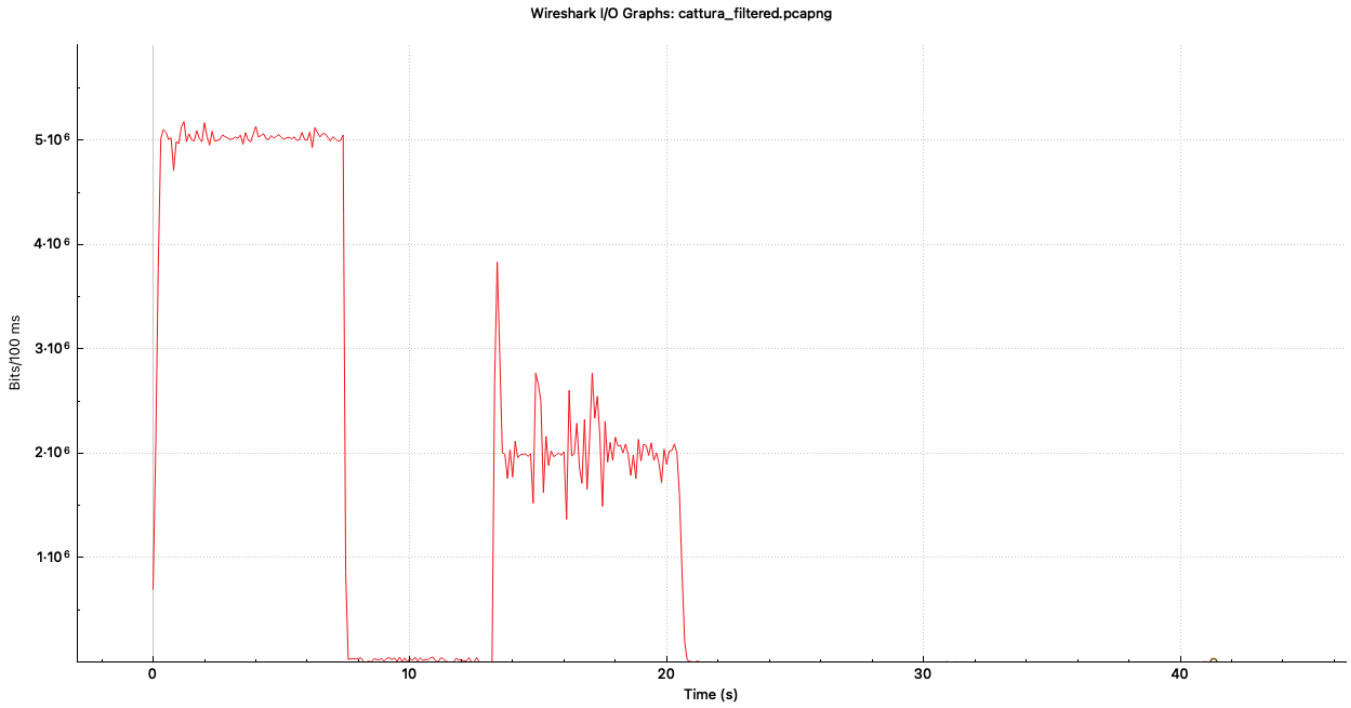
Confronto fra i risultati di FAST.com e la cattura Wireshark

Alla conclusione dello speedtest, FAST.com ha prodotto il seguente risultato:



48Mbit/s in Download, e 16Mbit/s in upload, scambiando circa 70MB di dati in totale fra client e server.

La macchina 192.168.1.3, con Wireshark, ha invece prodotto il seguente risultato:



Una media di 50Mbit/s in Download, e circa 20Mbit/s in Upload.

Si può notare quindi una differenza di 2-3Mbit/s tra la misurazione di FAST.com e di Wireshark. Questo è dovuto al fatto che FAST.com, essendo stato avviato da un browser web, è in grado di "catturare" e misurare esclusivamente il traffico del livello applicativo dello stack TCP/IP, al contrario di Wireshark che intercetta i pacchetti raw e può quindi analizzare tutti gli stati dello stack protocollare. La misura effettuata dallo speedtest dunque non tiene conto del traffico prodotto ai livelli sottostanti (cioè il livello di rete e collegamento). In particolare, Wireshark rileva gli header dei livelli IP e Ethernet, i quali, considerando 40Byte a pacchetto (20Byte per ciascun livello, circa il 3% del totale su pacchetti da 1500Byte), vanno ad aumentare il totale misurato di 2-3Mbit/s, ricavando la discrepanza che si era notata tra i dati di FAST.com e quelli misurati da Wireshark.