Gestione di reti AA 2020/21

"Analisi speed test (Fast-com) con Wireshark"

Gruppo 5

Calogero Turco (558998) - Francesco Giuseppe Ziccolella (588922) - Salvo Firera (578018)

Cattura del Traffico durante un test della connessione:

Filtro di cattura: Il filtro utilizzato per scremare il traffico del pacchetti è:
“ tcp and port 443 and ip ”.
In questo modo sono stati catturati dalla scheda di rete solo i pacchetti tcp sulla porta 443 di HTTPS da indirizzi ipv4.

Filtri di display:
All’inizio della cattura andiamo a considerare un’insieme X di possibili candidati per le connessioni per raffinare il filtro. Analizzando il codice su github, abbiamo trovato un server Api Restfull che ci restituisce l’insieme dei server candidati alle connessioni (json) :

https://api.fast.com/netflix/speedtest/v2?https=true&token=YXNkZmFzZGxmbnNkYWZoYXNkZmhrYWxm&urlCount=10

dove nella query definiamo il numero massimo di candidati con il parametro urlCount.


```
client:          {…}
targets:
  0:
    name:        "https://ipv4-c001-fco001-linkem-isp.1.oca.nflxvideo.net/speedtest?c=it&n=198471&v=33&e=1616097759&t=PUYdLAomOtVbezQWpkf4P2XQf78UOmQrjORh5A"
    url:         "https://ipv4-c001-fco001-linkem-isp.1.oca.nflxvideo.net/speedtest?c=it&n=198471&v=33&e=1616097759&t=PUYdLAomOtVbezQWpkf4P2XQf78UOmQrjORh5A"
    location:
      city:      "Rome"
      country:   "IT"
  1:
    name:        "https://ipv4-c006-rom001-ix.1.oca.nflxvideo.net/speedtest?c=it&n=198471&v=33&e=1616097759&t=l3PT-LtQgIwxT5j7CZ25PboBP2jwNvt4EayR-A"
    url:         "https://ipv4-c006-rom001-ix.1.oca.nflxvideo.net/speedtest?c=it&n=198471&v=33&e=1616097759&t=l3PT-LtQgIwxT5j7CZ25PboBP2jwNvt4EayR-A"
    location:
      city:      "Rome"
      country:   "IT"
  2:
    name:        "https://ipv4-c031-nyc005-ix.1.oca.nflxvideo.net/speedtest?c=it&n=198471&v=33&e=1616097759&t=C3ulHcfnANcRzS0KgzI_sbb_3GTbUK2lKlLxyQ"
    url:         "https://ipv4-c031-nyc005-ix.1.oca.nflxvideo.net/speedtest?c=it&n=198471&v=33&e=1616097759&t=C3ulHcfnANcRzS0KgzI_sbb_3GTbUK2lKlLxyQ"
    location:
      city:      "Secaucus"
      country:   "US"
  3:
    name:        "https://ipv4-c051-nyc005-ix.1.oca.nflxvideo.net/speedtest?c=it&n=198471&v=33&e=1616097759&t=cD3zOA4tqpCR7TesPLfqORNdTb1ArjSd05H9dg"
    url:         "https://ipv4-c051-nyc005-ix.1.oca.nflxvideo.net/speedtest?c=it&n=198471&v=33&e=1616097759&t=cD3zOA4tqpCR7TesPLfqORNdTb1ArjSd05H9dg"
    location:
      city:      "Secaucus"
      country:   "US"
  4:
    name:        "https://ipv4-c003-was001-dev-ix.1.oca.nflxvideo.net/speedtest?c=it&n=198471&v=28&e=1616097759&t=tNynzW2ivrmE09fldGmNUt7VTnY066SbvtZEJg"
    url:         "https://ipv4-c003-was001-dev-ix.1.oca.nflxvideo.net/speedtest?c=it&n=198471&v=28&e=1616097759&t=tNynzW2ivrmE09fldGmNUt7VTnY066SbvtZEJg"
```

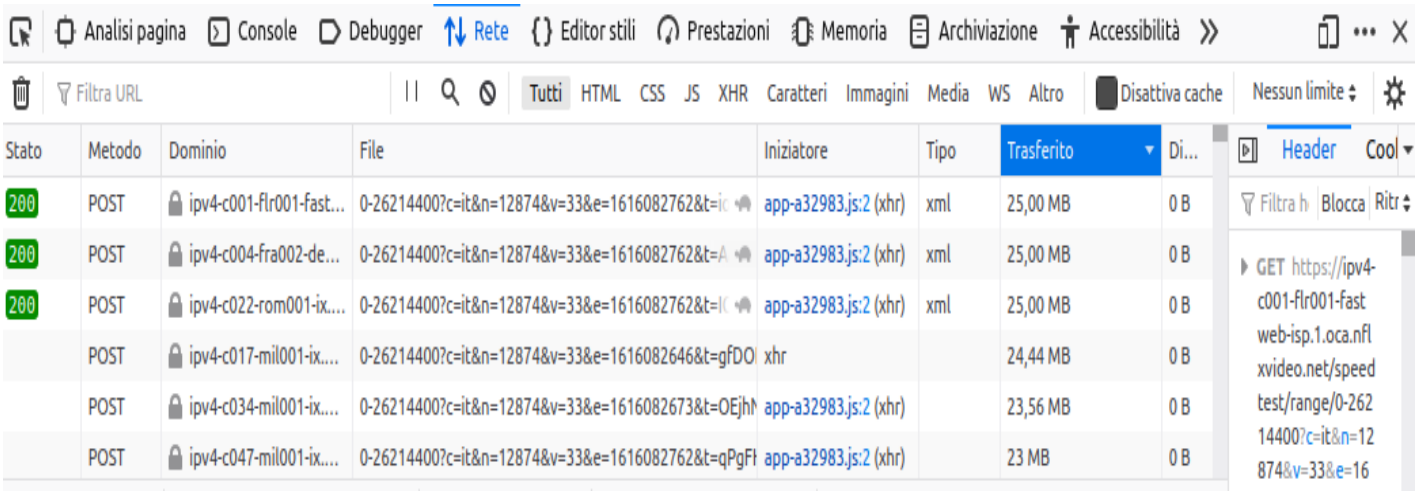
Allora è possibile usare un display filter che tenga conto di questo pattern (domini messi a fattor comune):
Display Filter=”ip.host contains "oca.nflxvideo.net"

N.B. Si noti che può essere sfruttato da python ,nel caso in cui si volesse automatizzare il filtro di cattura , evitando di dover ridefinire manualmente il display filter ad un cambio di dispositivo o ad un ulteriore test.

Assunzioni su come funziona il protocollo

Fase 1: HTTPS + TSLV 1.3
[Garantisce la sicurezza della connessione + controllo certificato]
Fase 2: Ping test
[Viene presa la latenza minima fra le connessioni]
Fase 3: Download Test
[Vengono instaurate X connessioni, di default un numero tra 1 e 8 server diversi, e diverse connessioni tcp per lo stesso server, vengono scaricati file di dimensione tra [0,25] MB, il risultato viene calcolato sulla macchina del client]
Fase 4: Upload Test
[Simile al download Test , upload di dim = [0,25] su X server]

Test su Linux

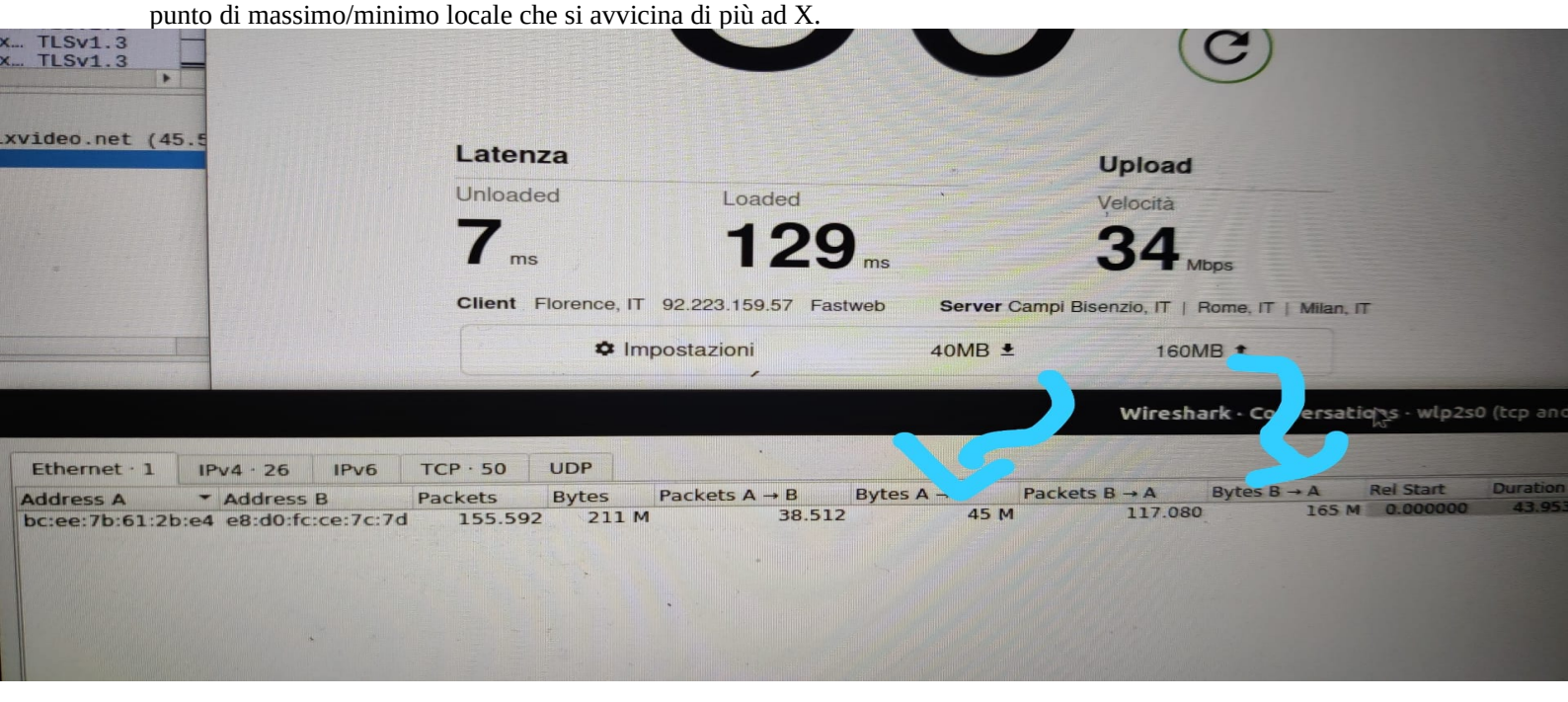


...i clic per selezionare il pacchetto 27590 (22s = 1.401e+7).

Enabled	Graph Name	Display Filter	Color	Style	Y Axis	Y Field	SMA Peric
☑	Tutti i pacchetti			Square	Bits		None
☑	ME -> Fast	ip.src == 192.1...		Line	Bits		None
☑	Me <- Fast	ip.dst == 192.1...		Line	Bits		None

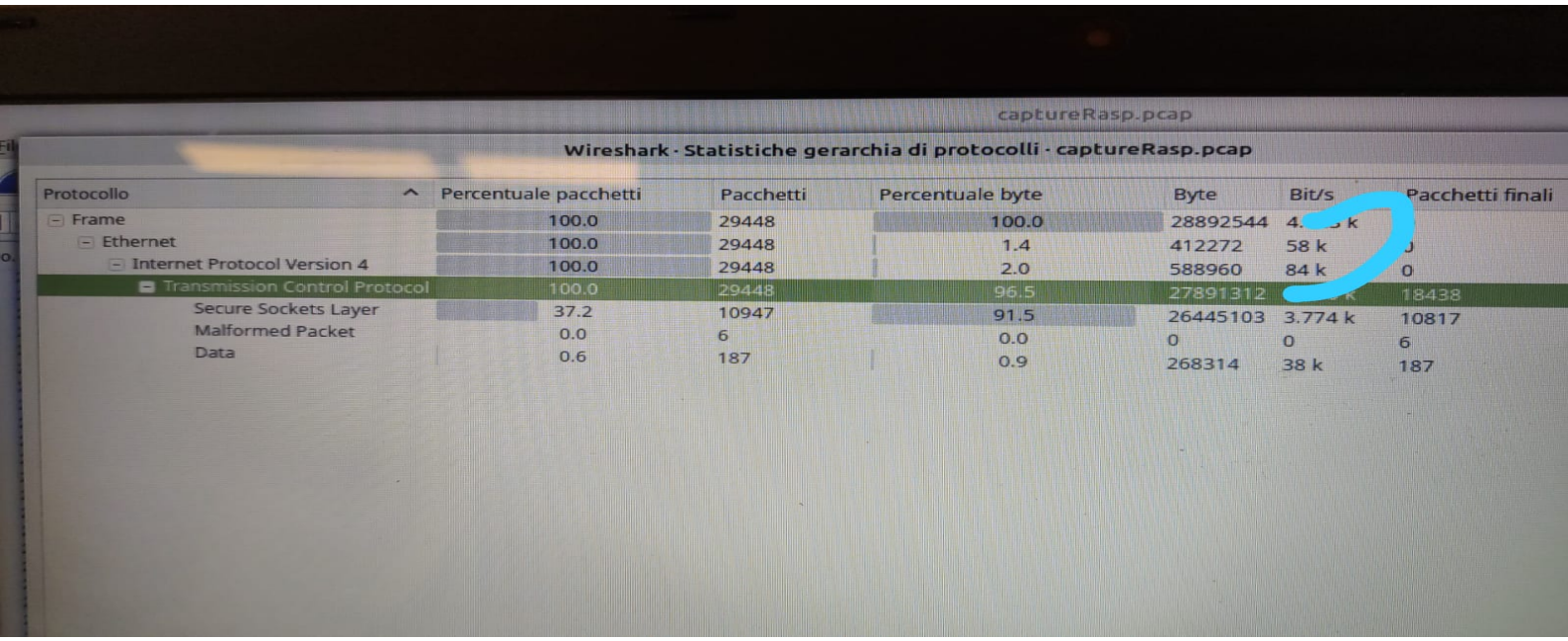
Analisi Risultati :

*** CONFRONTO DOWNLOAD /UPLOAD:**
Come valore sul sito viene scelto il picco massimo, mentre per l'upload viene fatta una media X e viene scelto il punto di massimo/minimo locale che si avvicina di più ad X.



*** CONFRONTO DATI RICEVUTI / DICHIARATI SUL SITO**

Poichè si voleva controllare che tutto il traffico fosse stato catturato , abbiamo confrontato il valore dichiarato con il traffico catturato.
Ne deduciamo che in media ho il 3-4% di byte in più ,poiche ci sono gli header del livello ip ed ethernet da considerare e il bit (NOT FRAGMENT) è settato ad 1.
Il pacchetto ha dimensione 1506 byte , la parte di Segment Data cioè il Payload è 1440= 1506-1440=66 byte di overhead.



Protocollo	Percentuale pacchetti	Pacchetti	Percentuale byte	Byte	Bit/s	Pacchetti finali
Frame	100.0	29448	100.0	28892544	4.[illegible] k	
Ethernet	100.0	29448	1.4	412272	58 k	0
Internet Protocol Version 4	100.0	29448	2.0	588960	84 k	0
Transmission Control Protocol	100.0	29448	96.5	27891312	[illegible] k	18438
Secure Sockets Layer	37.2	10947	91.5	26445103	3.774 k	10817
Malformed Packet	0.0	6	0.0	0	0	6
Data	0.6	187	0.9	268314	38 k	187

*** CONNESSIONI :**
Il numero di connessioni e la quantità di dati scambiati si determina dalla configurazione scelta su Fast , le possiamo vedere nelle conversazioni

Ethernet · 1 | IPv4 · 7 | IPv6 | TCP · 23 | UDP

Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
23.246.50.132	192.168.0.128	15.781	20 M	5.917	7.160 k	9.864	13 M	0.993231	103.1340	555 k	1.027 k
23.246.51.155	192.168.0.128	18.172	30 M	6.568	15 M	11.604	15 M	0.355372	103.7173	1.201 k	1.160 k
45.57.74.168	192.168.0.128	10.555	13 M	2.246	1.609 k	8.309	11 M	1.526695	57.0408	225 k	1.661 k
45.57.75.168	192.168.0.128	10.832	14 M	2.201	2.398 k	8.631	12 M	1.532150	56.9393	336 k	1.722 k
54.154.59.168	192.168.0.128	628	563 k	229	29 k	399	534 k	8.328650	96.3931	2.415	44 k
54.154.202.35	192.168.0.128	34	8.401	15	6.047	19	2.354	0.000000	61.3063	789	307
192.168.0.128	198.38.121.175	16.241	24 M	10.262	13 M	5.979	11 M	2.133690	102.2524	1.084 k	871 k

Fonti:
- https://netflixtechblog.com/building-fast-com-4857fe0f8adb
- https://ripe74.ripe.net/presentations/105-Ripe74-Fast.com-1.pdf
- https://ipinfo.io/AS2906/23.246.50.0/24