Anno Accademico 2021/2022

Laboratorio di Internet

Politecnico di Torino

Report IV

Analisi Microsoft Teams

Gruppo 14

Andreas Brummer (s270332)

Alessandro Ciullo (s269589)

Andrea Scamporrino (s270971)

1 Descrizione di Microsoft Teams

Abbiamo scelto come oggetto della nostra relazione questa applicazione poiché il vasto numero di funzioni condensate al suo interno ci permette di analizzare e descrivere un numero significativo di fenomeni diversi.

La versione del programma da noi utilizzato è quella istituzionale, in quanto, durante i nostri esperimenti ci siamo connessi a Teams facendo uso del pacchetto Microsoft 365 fornitoci dal nostro ateneo.

Microsoft Teams è una piattaforma di comunicazione e collaborazione unificata che combina chat di lavoro persistente, teleconferenza, condivisione di contenuti (incluso lo scambio e il lavoro simultaneo sui file) e integrazione delle applicazioni.

1.1 Servizi

1.1.1 Funzioni principali

- Messaggistica istantanea
- Voice over IP (VoIP)

Teams supporta le conferenze in rete telefonica generale (PSTN) consentendo agli utenti di chiamare numeri di telefono dal client.

1.1.2 Canali

All'interno di un team, i membri possono impostare canali. I canali sono argomenti di conversazione che consentono ai membri del team di comunicare senza l'uso di e-mail o SMS di gruppo. Gli utenti possono rispondere ai post con testo, immagini e GIF personalizzati. I messaggi diretti consentono agli utenti di inviare messaggi privati a un utente specifico anziché a un gruppo di persone.

1.1.3 Riunioni

Le riunioni possono essere programmate o create ad hoc e gli utenti che visitano il canale potranno vedere che una riunione è in corso. Teams ha anche un plugin per Microsoft Outlook per invitare altri a una riunione di Teams.

2 Configurazione di rete

Tutti i test, le analisi e le catture ottenute utilizzando il software Wireshark, sono state fatte da un computer con sistema operativo Linux con distribuzione Arch connesso tramite Wi-fi all'hotspot del telefono.

Il dispositivo utilizzato ha la seguente configurazione di rete:

Indirizzo di rete	192.168.115.48
Indirizzo di broadcast	192.168.115.255
Default Gateway	192.168.115.93
NetMask	255.255.255.0

Tabella 2.1: Configurazione di rete

Per tutti i test in cui serviva un secondo dispositivo come nel caso di scambio di messaggi, video-chiamata e invio e reinvio di un file è stato utilizzato un computer con sistema operativo MacOS Monterey sul quale è installato Microsoft Teams. Per ogni esperimento sono state effettuate più catture in modo da essere più precisi durante le analisi e confrontarle tra loro.

3 Esperimenti

3.1 Apertura dell'applicazione

All'apertura l'applicazione esegue una serie di richieste DNS per risolvere alcuni indirizzi. I siti richiesti in particolare sono i seguenti:

- mobile.pipe.aria.microsoft.com
- teams.microsoft.com
- statics.teams.cdn.office.net
- eu-api.asm.skype.com
- eu-prod.asyncgw.teams.microsoft.com
- emea.ng.msg.teams.microsoft.com
- config.teams.microsoft.com
- uksouth-prod.notifications.teams.microsoft.com
- login.microsoftonline.com
- teams.events.data.microsoft.com

Durante la fase di apertura dell'applicazione il server con cui abbiamo il maggior numero di byte scambiati in è teams.microsoft.com (circa il 90% dell'intero flusso TCP come mostrato figura 3.1).

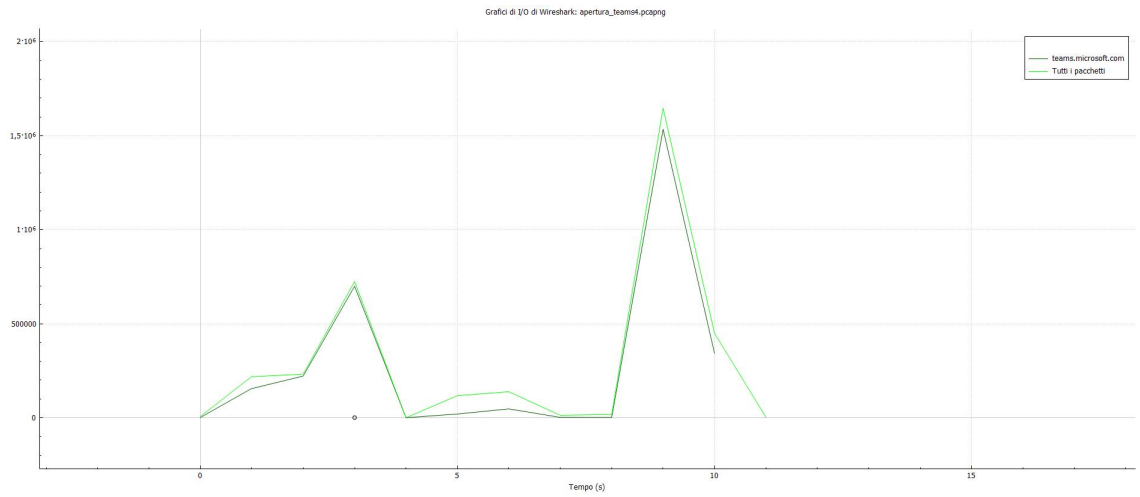

Figura 3.1: Confronto flusso TCP verso teams.microsoft.com ed intero flusso TCP

Analizzando le query DNS per tale hostname scopriamo che è un alias del server s-0005.s-msedge.net che ha indirizzo di rete 52.113.194.132 e, secondo alcuni database online appartiene, come potevamo immaginare, a Microsoft ed è geolocalizzato in Nord America: quest'ultima assunzione è falsa poichè il RTT misurato durante l'handshake TCP è di appena 40ms ed il server è quindi fisicamente in Europa.

Buona parte del restante flusso viene invece trasmesso verso il server login.microsoftonline.com per la fase di accesso al nostro account che però, analizzeremo in maniera approfondita nella sezione 3.2.

In linea generale, con ciascuno degli host elencati, una volta risolti i loro indirizzi di rete, vengono istanziate una o più connessioni TCP. Per la comunicazione viene utilizzato TLSv1.2, un protocollo che viaggia su TCP e rende le comunicazioni sicure per mezzo della crittografia asimmetrica. L'handshake viene

avviato tramite un 'Client Hello': il messaggio includerà la versione TLS supportata dal client, le suite di crittografia supportate e una stringa di byte casuali nota come "casuale client". Una volta terminato l'handshake vengono scambiati pacchetti di dati ed i relativi ACK(che usano solamente TCP).

Durante la fase di apertura vengono anche scaricati i backup delle nostre chat che possono contenere oggetti collocati in server esterni a Microsoft, come il servizio di host di GIF osservato nelle nostre catture: media0.gyphy.com (API del noto sito giphy.com), corrispondente all'indirizzo 151.101.242.2 e localizzato in Italia.

In totale, durante i 18 secondi di durata dell'esperimento, vengono contattati 17 diversi indirizzi IP e vengono istanziate 25 connessioni TCP. Sono riportati nella tabella 3.1 le 10 principali conversazioni IPv4 e il numero di byte scambiato con ciascuno degli host.

Address A	Address B	Bytes
192.168.115.48	52.113.194.132	3219113
192.168.115.48	40.126.32.135	107837
192.168.115.48	20.189.173.11	45577
192.168.115.48	52.111.255.0	22694
192.168.115.48	52.113.199.174	21458
192.168.115.48	23.12.130.220	18340
192.168.115.48	52.113.199.175	17675
192.168.115.48	52.113.205.25	15053
192.168.115.48	52.114.92.0	13844
192.168.115.48	52.114.92.158	13133
...	...	...

Tabella 3.1: Conversazioni IPv4 durante l'apertura di Microsoft Teams

3.2 Login

In questa sezione analizziamo il processo di login all'applicazione Microsoft Teams. Nell'esperienza in questione effettuiamo l'accesso mediante l'account studentesco per i servizi Microsoft fornitoci dall'ateneo.

Osservando le catture ottenute tramite Wireshark, e confrontandole con quelle ottenute durante un avvio dell'applicazione con account già connesso, possiamo intuire i passaggi aggiuntivi che vengono effettuati. Dopo un primo flusso di dati iniziali scambiati durante l'apertura del programma viene constatato lo stato dell'accesso contattando il server login.microsoftonline.com il cui indirizzo IP 40.126.32.135 viene ottenuto tramite una DNS query di tipo A: secondo "whois" il server è localizzato in Olanda, e a riprova di ciò otteniamo, come mostrato in figura 3.2, un RTT di circa 50ms durante l'handshake TCP.

No.	Time	Source	Destination	Protocol	Length	Info
12344	5.800977607	192.168.115.48	40.126.32.135	TCP	74	33762 → 443 [SYN] Seq=0
12346	5.857564274	40.126.32.135	192.168.115.48	TCP	66	443 → 33762 [SYN, ACK]

Figura 3.2: RTT login.microsoft.com misurato durante TCP handshake

Viene instaurata una connessione TLS con il server in questione e, se l'account è già "loggato" nell'applicazione la procedura di login termina qui, altrimenti si viene reindirizzati verso un nuovo server. Segue quindi una coppia di query DNS verso login.live.com, ottenendo gli indirizzi 13.107.213.60 e 13.107.246.60. Viene, quindi, stabilita una connessione TLS con il primo dei due indirizzi, mentre il secondo viene tenuto come backup o scartato. Il server in questione secondo diversi siti di analytics appartiene a Microsoft ed è localizzato negli Stati Uniti ma, l'handshake TCP dura appena 50ms: possiamo assumere che il server in realtà si trovi in Europa ed i certificati (Baltimore CyberTrust) scambiati durante l'apertura TLS ne confermano il proprietario.

Segue una query DNS per privacy.microsoft.com e www.microsoft.com attraverso le quali si ottiene l'indirizzo ipv4 2.20.205.172 per entrambi, che corrisponde ad un server, localizzato a Milano (secondo ipinfo.io),

facente parte del Content Distribution Network della nota azienda Akamai. Con essi non viene stabilita nessuna connessione, in quanto il login viene effettuato attraverso il sito del Politecnico di Torino e non attraverso un account Microsoft.

Inserendo nella connessione TLS instaurata con login.live.com come credenziali la mail istituzionale del nostro ateneo, veniamo reindirizzati ad una pagina di login del Politecnico. Osservando le catture, infatti, notiamo una query DNS per adfs.polito.it che ottiene come risposta l'indirizzo 130.192.182.46 e con cui viene instaurata una connessione TLS. Attraverso tale connessione vengono scambiate delle chiavi e viene fatto un nuovo handshake criptato; solo dopo, le credenziali vengono inviate e il login viene effettuato.

Dopo aver terminato la comunicazione con il server adfs.polito.it, avviene un ultimo scambio di messaggi con login.microsoftonline.com e login.live.com con cui si conclude la procedura di accesso al programma.

Address A	Port A	Address B	Port B	Bytes	Start(s)	Duration(s)
192.168.115.48	33762	40.126.32.135	443	215779	5.800	35.989
192.168.115.48	39896	13.107.213.60	443	212071	8.0617	18.347
192.168.115.48	39900	13.107.213.60	443	121052	8.453	18.163
192.168.115.48	49492	130.192.182.46	443	265151	20.169	5.322
192.168.115.48	49494	130.192.182.46	443	41756	20.242	0.8654
192.168.115.48	33764	40.126.32.135	443	5686	41.442	0.260

Tabella 3.2: Conversazioni TCP per login

La tabella 3.2, ottenuta tramite l'analisi delle conversazioni di Wireshark, elenca in ordine cronologico le connessioni TCP aperte con i server discussi prima. Si osserva che per entrambi i server login.live.com (13.107.213.60) ed adfs.polito.it (130.192.182.46) per effettuare le comunicazioni viene aperta una coppia di connessioni simultanee, probabilmente per ottenere una connessione più robusta e veloce e/o per differenziare flussi di dati di tipologie diverse. Le due connessioni verso il server login.microsoftonline.com (40.126.32.135) non vengono invece istanziate in contemporanea: la prima viene aperta durante la fase iniziale di login e, resta attiva per quasi l'intera durata dell'operazione, mentre la seconda viene aperta nella fase finale e dura per pochi istanti. Osserviamo infine che il protocollo di trasmissione utilizzato è sempre HTTP over TLS.

3.3 Video-Chiamata

In questa sezione analizziamo il processo di video-chiamata dell'applicazione Microsoft Teams analizzando sia il caso di ricezione di essa e sia il caso di invio.

In entrambe le situazioni la chiamata avviene tramite scambio di pacchetti con protocollo RTP. Quest'ultimo è l'acronimo di Real-time Transport Protocol (Protocollo di trasporto in tempo reale), è un protocollo di livello applicazione e definisce il formato di un pacchetto standard per la consegna audio e video su internet. Viene utilizzato in combinazione con il protocollo di controllo RTCP. Mentre RTP trasporta i flussi multimediali, audio e video, RTCP viene utilizzato per monitorare le statistiche di trasmissione e la qualità del servizio e aiuta la sincronizzazione di più flussi. RTP viene originato e ricevuto su numeri di porta pari e la comunicazione RTCP associata utilizza il numero di porta dispari immediatamente superiore. L'obiettivo di progettazione di RTP è lo streaming end-to-end in tempo reale dei dati relativi ai media. RTP include meccanismi per il rilevamento della perdita di pacchetti, così come la consegna di pacchetti di dati fuori servizio, problemi che sono particolarmente comuni nelle trasmissioni UDP (User Datagram Protocol) su IP. Poiché RTP consente il trasferimento di dati a più end-point di destinazione in parallelo tramite IP multicast, è lo standard primario impiegato per i trasferimenti di rete IP audio e video. Anche il TCP (Transmission Control Protocol) è standardizzato per l'uso di RTP, anche se non è tipicamente impiegato nelle applicazioni a causa dei suoi meccanismi di controllo degli errori che possono causare ritardi e influenzare la consegna puntuale dei pacchetti. Per questo motivo, la maggior parte delle applicazioni RTP basano le loro implementazioni su UDP.

Osservando le catture ottenute tramite Wireshark durante una video-chiamata, notiamo che i flussi RTP avvengono con 2 indirizzi IP differenti, quest'ultimi ricevuti dall'host "trouter2-azsc-euwe-8-a.trouter.teams.microsoft.com" tramite un messaggio TLS. Questo perchè audio e video nella stessa conferenza vengono trasmessi come due

diverse sessioni RTP ciascuna gestita attraverso pacchetti RTCP e vengono sfruttate due differenti coppie di porte UDP; una delle ragioni di questa separazione è di permettere ai partecipanti di ricevere a richiesta un solo flusso di dati. Nell'RTP Video/Audio viene stabilita una corrispondenza standard tra i numeri dei payload types e le più usate codifiche dati; non a tutte le codifiche usate da RTP deve essere assegnato un payload type statico, infatti i codici dal 96 al 127 possono essere definiti dinamicamente. Nel nostro caso i numeri di payload types usati sono il 122 e 123 per il video e il 104 per l'audio.

Come si può vedere dal grafico 3.3 che rappresenta i pacchetti RTP scambiati durante una video chiamata, i pacchetti video, definiti dalla linea rossa sono molto più pesanti e frequenti rispetto ai pacchetti audio, definiti dalla linea verde. Inoltre si può notare che durante la video-chiamata il video è stato attivato circa al secondo 18 mentre la chiamata vocale era già iniziata da qualche secondo.

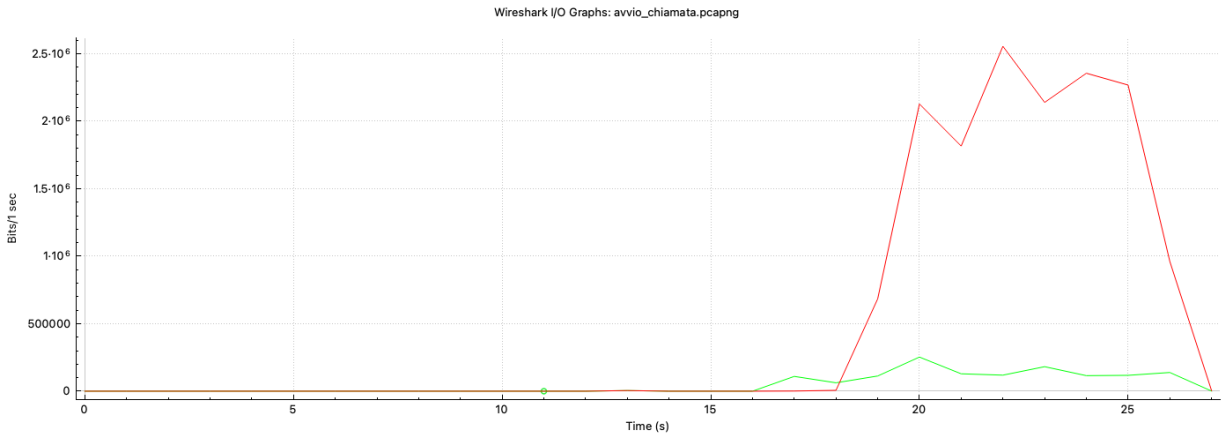

Figura 3.3: Grafico UDP durante una video-chiamata.

Nei due test effettuati con Wireshark gli indirizzi IP con cui comunichiamo durante la video-chiamata sono sempre diversi tra loro e con posizioni geografiche differenti che sono state verificate sia da più siti internet e sia dal comando traceroute del terminale dal quale si può trovare la distanza in millisecondi dall'host target. In particolare nel test di avvio di una video-chiamata gli indirizzi IP osservati sono provenienti dai Paesi Bassi (circa 20 millisecondi con traceroute), mentre nel test di ricezione sono provenienti dall'Inghilterra (circa 30 millisecondi).

É interessante osservare come la chiamata vera è propria sia anticipata, su entrambi i client, da un breve flusso UDP indirizzato ad un server Microsoft che termina prima dell'inzio della trasmissione Audio/Video. Seppur non possiamo affermalo con certezza, abbiamo le evidenze necessarie a sostenere che questi pacchetti UDP corrispondano allo "squillo" del telefono, ed il loro flusso termina, infatti, nel momento in cui la chiamata viene accettata oppure rifiutata come evidenziato attraversi il software Wireshark.

Analizzando i pacchetti catturati abbiamo osservato che durante la fase di apertura della chiamata vengono inoltrate una serie richieste STUN verso un server relay di Microsoft Teams (euaz.relay.teams.microsoft.com) e verso il NAT dell'utente con cui si vuole aprire la comunicazione da entrambi i client per verificare se è possibile stabilire una connessione diretta tra i due dispositivi. STUN è un protocollo che permette alle applicazioni in esecuzione su un computer di scoprire la presenza ed i tipi di NAT e firewall che si interpongono tra il computer e la rete pubblica. STUN permette a queste applicazioni di conoscere gli indirizzi IP e le porte con cui il dispositivo NAT li sta rendendo visibili sulla rete pubblica. STUN è un protocollo client-server. Un telefono o un software VoIP, nel nostro caso Microsoft Teams, può includere un client STUN, che invierà una richiesta ad un server STUN. Il server riporterà al client STUN l'indirizzo IP pubblico e la porta UDP che il dispositivo NAT (per esempio il router) sta associando al client per il traffico entrante nella rete. Le risposte permettono anche al client STUN di determinare che tipo di NAT è in uso. Ci sono tre tipi di NAT che è possibile attraversare tramite STUN: Full Cone, Restricted Cone e Port Restricted Cone. STUN non lavora con il quarto tipo di NAT, detto simmetrico o bidirezionale, questo a causa del fatto che i dati trovati

dal server STUN non saranno validi per terze parti, in quanto il NAT bidirezionale non permette a terzi di riusare IP e porte abilitate, differenziando le associazioni a seconda dell'host contattato.

Nei tentativi effettuati siamo riusciti a stabilire un collegamento diretto solo quando connessi alla stessa rete locale tramite Wi-fi. In tutti gli altri casi, invece, i pacchetti sono stati inviati ad alcuni server di Microsoft (non uguali per i due client in comunicazione, ma abbastanza simili da ipotizzare appartenghino allo stesso data center) per poi essere inoltrati al destinatario.

3.4 Invio e reinvio di un File

In questa sezione abbiamo testato il comportamento di Microsoft Teams durante l'invio di file in una chat all'interno del programma.

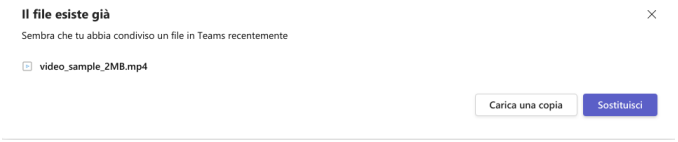

Figura 3.4: Opzioni reinvio file

Il file di test utilizzato è un file video MP4 ed ha un peso poco superiore a 2MB. Il test è stato svolto nei seguenti passi:

- Caricamento del file in chat per la prima volta.
- Annullamento e cancellazione del messaggio.
- reinvio del file selezionando l'opzione "Sostituisci".
- Nuovamente annullamento e cancellazione del messaggio.
- reinvio del file selezionando l'opzione "Carica una copia".

Utilizzando il software Wireshark possiamo isolare la conversazione TCP in cui è stato trasmesso il file utilizzando lo strumento di statistiche fornito per valutare il numero di byte trasmessi dal nostro host in ciascuna connessione. La conversazione è quindi facilmente individuata e mostrata in tabella 3.3.

Address A	Port A	Address B	Port B	Bytes A → B
192.168.115.48	37734	13.107.136.9	443	6658483

Tabella 3.3: Conversazione TCP in cui è stato trasmesso il file (per 3 volte)

É di fatti l'unica connessione in cui sono stati trasmessi più di 2MB e perciò la scelta è stata semplice. Notiamo subito che il valore di byte trasmessi è circa il triplo di quello del singolo file ed osservando il grafico I/O in figura 3.5 dei bit trasmessi attraverso connessione TLS nel flusso appena individuato è palese che il programma non offre nessun sistema di recupero del file ma, indipendentemente dall'opzione selezionata durante l'operazione di reinvio, il file campione viene ritrasmesso interamente. Non viene inoltre effettuata alcuna operazione di compressione video poiché i dati trasmessi, seppur inficiati dagli header di ciascun pacchetto, sono in dimensione riconducibili al file originale di 2MB.

Cercando nelle catture Wireshark il server corrispondente all'indirizzo IP 13.107.136.9 (mostrato in tabella 3.3) troviamo l'hostname politoit-my.sharepoint.com. Utilizzando la mail istituzionale del nostro Ateneo la trasmissione avviene caricando su una pagina personale di storage online, supportata dal servizio Microsoft OneDrive in collaborazione con il Politecnico di Torino, il file target. Il server 13.107.136.9, contrariamente a quanto il suo hostname lasci presumere, non appartiene alla nostra università, bensì a Microsoft. Secondo alcuni database online, l'host è localizzato negli Stati Uniti, ma il breve RTT letto dalle catture ci suggerisce che il server in questione sia situato in Europa ed appartenga al servizio di storage online OneDrive.

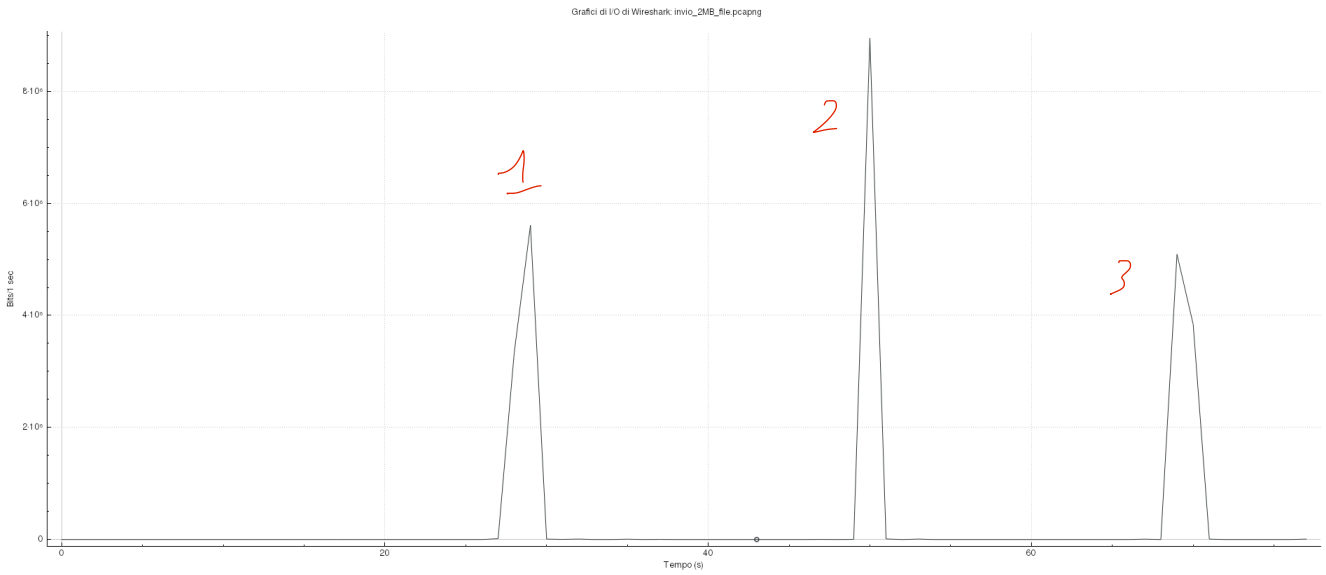

Figura 3.5: Grafico del flusso TLS durante l'invio ripetuto 3 volte di un file

4 Conclusione

Attraverso la stesura di questo report abbiamo avuto l'opportunità di utilizzare gli strumenti conosciuti ed appresi durante l'intero corso "Laboratorio di Internet" per analizzare in maniera concreta un'applicazione di uso comune. Avendo a che fare con pacchetti, quasi sempre, criptati dal protocollo TLS, durante la nostra esperienza abbiamo dovuto far uso di tutti gli elementi a nostra disposizione per poter dedurre il comportamento del programma e riportare una descrizione più fedele possibile del funzionamento dei servizi offerti da Microsoft Teams.

La grande mole di messaggi scambiati durante l'utilizzo di un'applicazione "reale" complica ulteriormente le cose. Ciò nonostante, mediante un'attenta analisi dei pacchetti scambiati e sfruttando a pieno le conoscenze apprese durante il corso possiamo inquadrare i punti cardine del programma analizzato.

L'applicazione Microsoft Teams fa largo uso del protocollo TCP, attraverso cui vengono scambiati praticamente tutti dati necessari al corretto funzionamento dell'applicazione e che permettono la corretta interazione tra il client e i server con cui continuamente si interfaccia. I messaggi inviati sono quasi interamente criptati attraverso il protocollo TLS, in modo da fornire all'utente finale un'esperienza priva dei rischi che, una connessione non sicura comporta.

Il protocollo di trasmissione dati utilizzato è HTTPS, essendo Teams anche un'applicazione web ed, essendo essenziale la comunicazione con server esterni a quelli della nostra applicazione, utilizzare un protocollo standard così diffuso è la scelta migliore.

L'unica funzione che il programma fornisce mediante una connessione UDP è la video-chiamata. Teams fa, di fatti, uso del protocollo RTP, molto diffuso per questa tipologia di servizi poiché, utilizzando una connessione UDP, è estremamente adatto alla trasmissione in tempo reale in quanto gli errori di trasmissione risultano trascurabili per queste operazioni e la velocità del flusso è la più veloce possibile.