**Entrega:** 30/11/2023

**Alumnes:** Francesco Oncins Spedo

Mariona Farré Tapias

Pau Alcázar Perdomo

**INTERNET:**

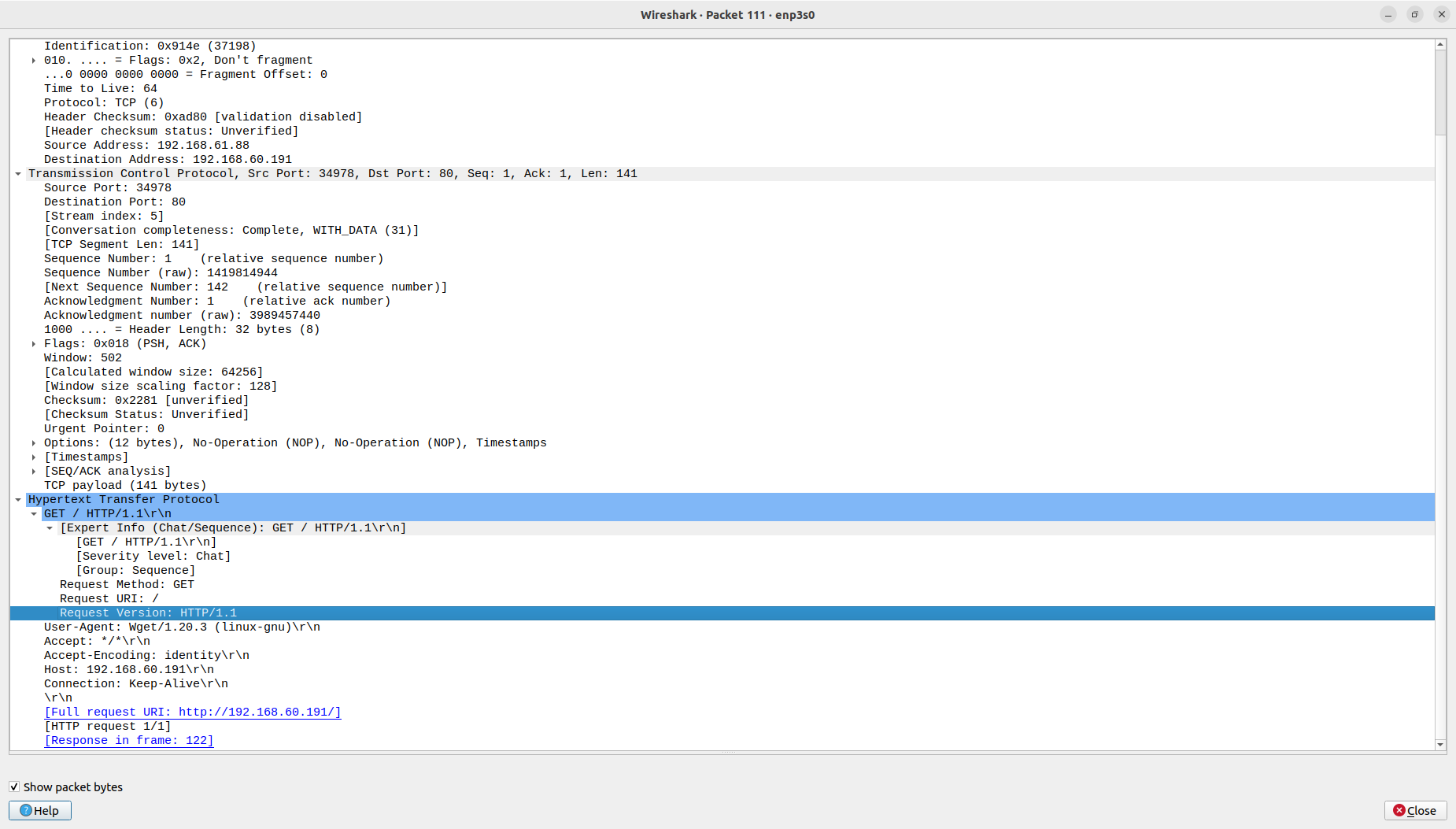
**P2- Qüestionari sessió 3 - Anàlisis dels protocols HTTP**

**Preguntes HTTP bàsic**

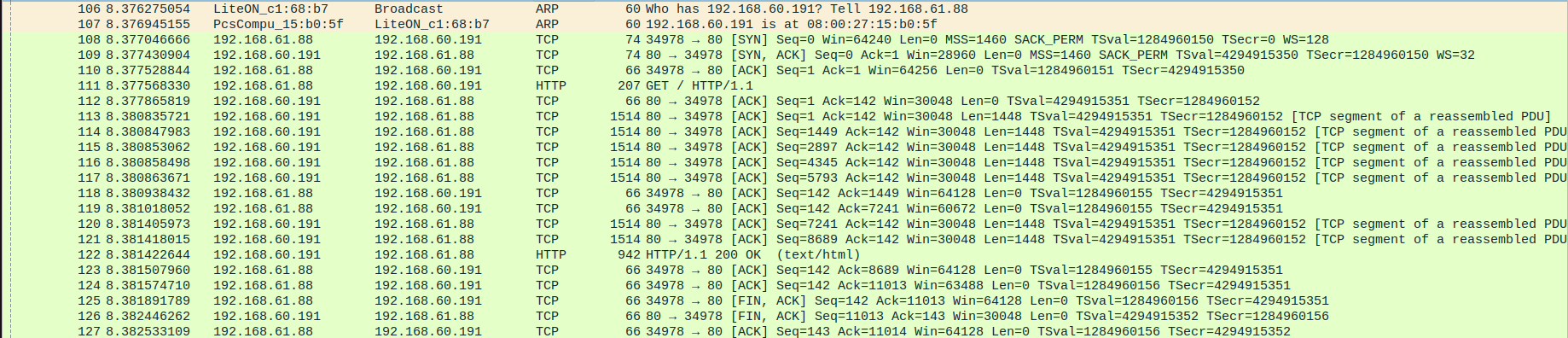
Sí, és possible fer més d'una petició HTTP en una mateixa connexió TCP. Això depèn de la versió d'HTTP que s'estigui utilitzant.

A la versió que utilitzem, la 1.1, les connexions TCP queden obertes després d'una petició HTTP que permet peticions addicionals sense necessitat d'establir una nova connexió.

Com podem veure a la següent imatge, que conté la configuració d’un paquet HTTP que hem capturat amb wireshark, es pot veure la versió d’HTTP, dins del seu quadre de dades *Hypertext Transfer Protocol*:



Nosaltres hem vist qu￼e s’utilitza és la 1.1 del protocol HTTP.



El mètode que es pot veure en la petició és "GET", que s'utilitza per sol·licitar dades des d'un recurs específic.

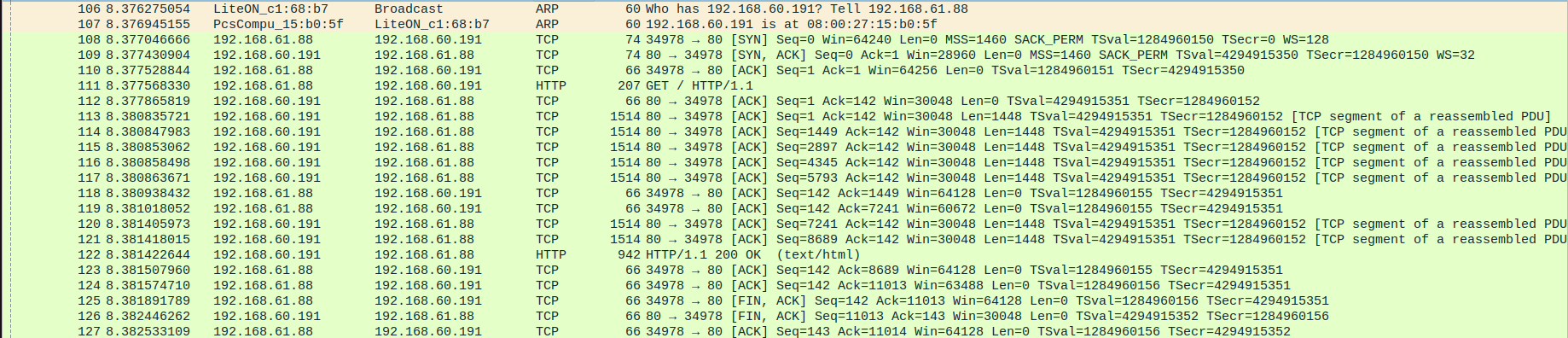
Pel que fa als codis de resposta, en una de les línies es pot veure "HTTP/1.1 200 OK", que és un codi de resposta HTTP. El codi 200 significa que la petició ha estat exitosa i que el contingut sol·licitat s'està enviant al client.

Altres mètodes que podem trobar són:

* HEAD: Només retorna els headers del servidor, però no el contingut (body) del document especificat.
* PUT: Demana al client que accepti i guardi un recurs amb la request-URI que demana el client.
* DELETE: A l’inrevés que PUT, per a esborrar un recurs.
* POST: Petició utilitzada per a per enviar correus electrònics, news, o formularis que poden ser omplerts per un usuari interactiu.
* OPTIONS: Per a preguntar al servidor per les capacitats d’un recurs determinat o del servidor en general
* TRACE: Utilitzat per debugging a nivell d’aplicació.

Alguns dels codis de resposta més utilitzats:

* 200:Ok petició acceptada



* 400: Error del client
* 401: Client error: petició incorrecta
* 500: Server error: error intern del servidor

Els diferents recursos que composen la pàgina, són:

* **text**, és el recurs més bàsic i més utilitzat.
* **imatges**, pot haver-hi un directori amb imatges que s'utilitzen dins de la web.
* **link**, conté la direcció d’un recurs a la web i com accedir-hi.

Amb la propietat d’inspeccionar la web podem trobar els recursos que hem mencionat:

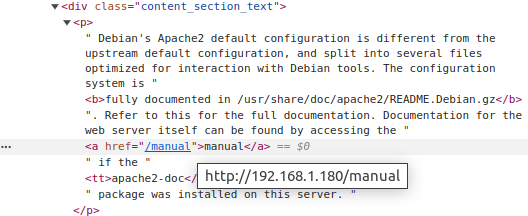
**text**:



**imatge**:



**link**:



i com accedir als recursos anteriors a partir d’un link:

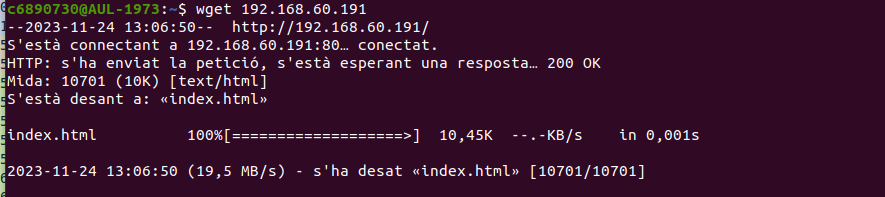
recurs: URL relativa URL absoluta

**text** /var/www/html/index.html [http://192.168.1.180/](http://192.168.1.180/icons/openlogo-75.png)

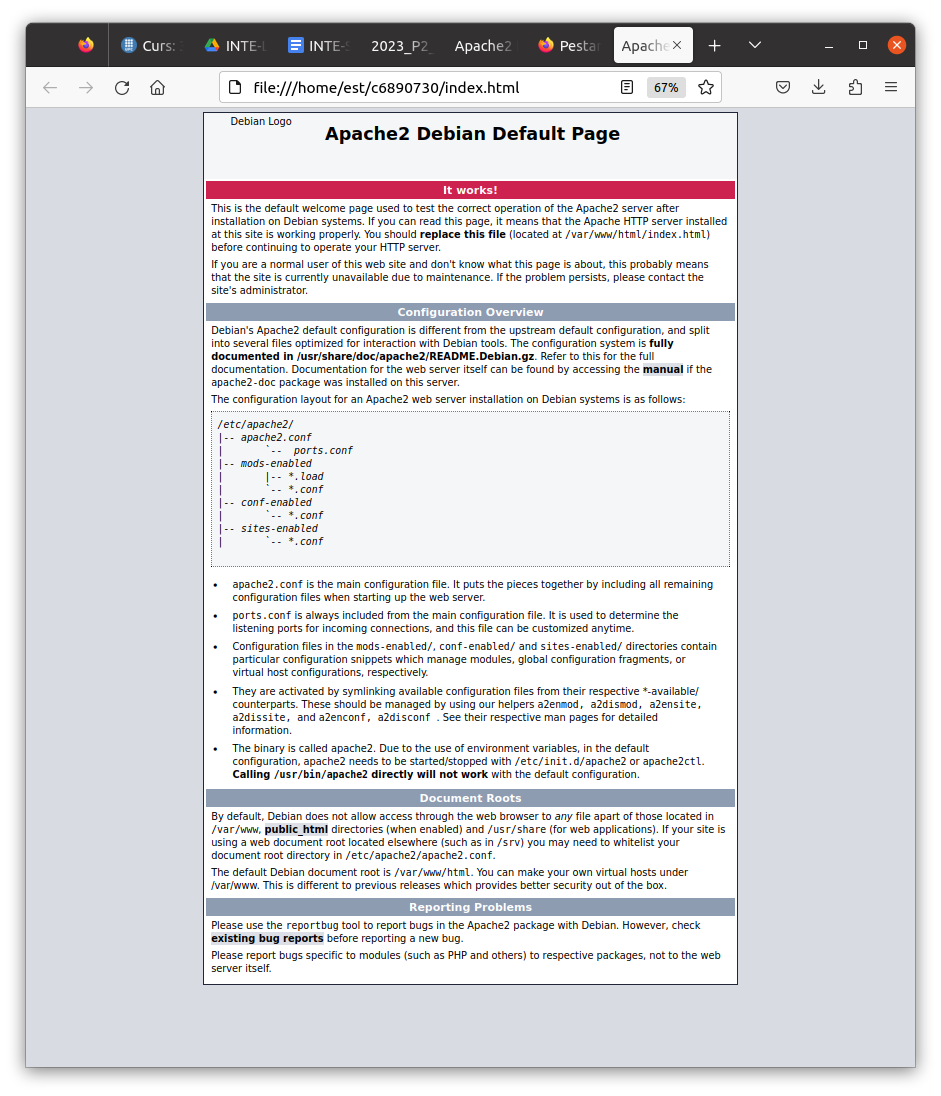
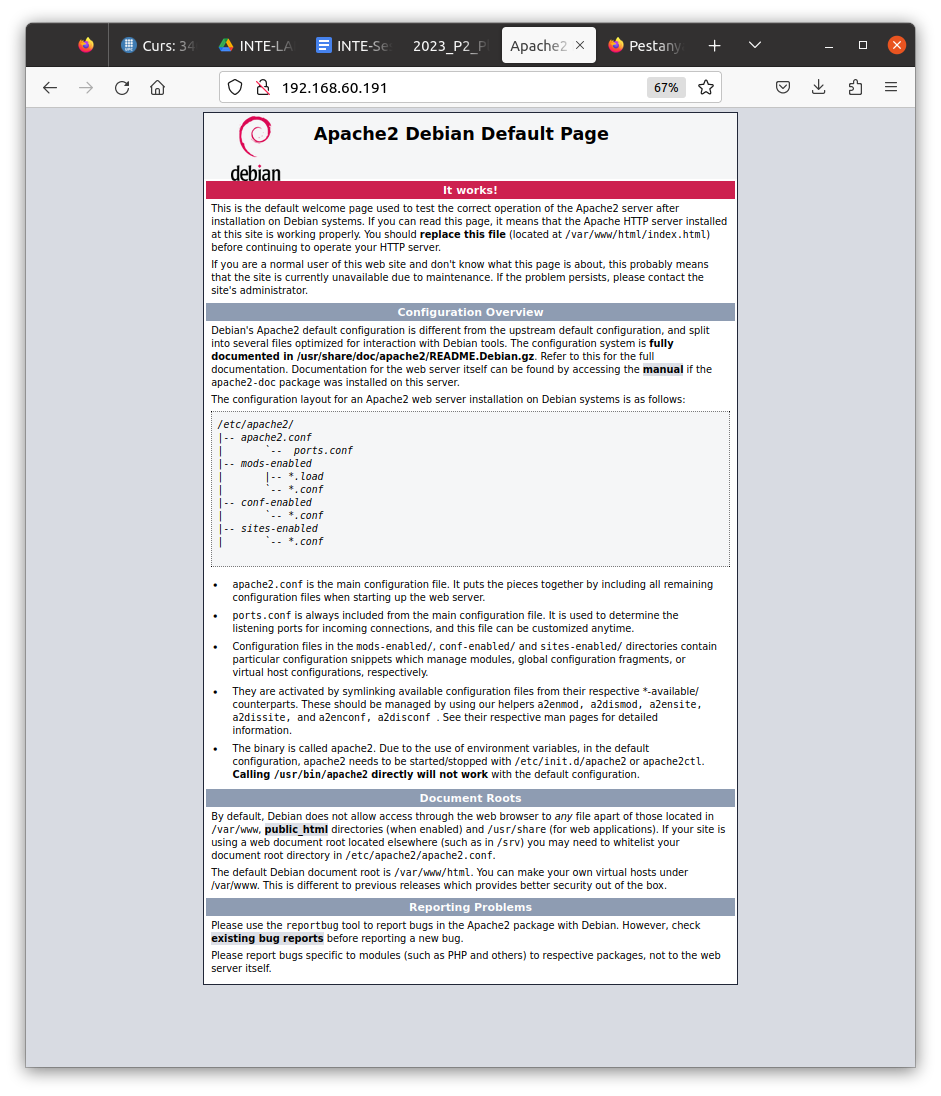
**imatge** [/icons/openlogo-75.pn](http://192.168.1.180/icons/openlogo-75.png)g <http://192.168.1.180/icons/openlogo-75.png>

**link** [/manual](http://192.168.1.180/icons/openlogo-75.png) [http://192.168.1.180/manual](http://192.168.1.180/icons/openlogo-75.png)

Fent el wget d’una web (que hem activat dins d’un servidor) a través de la seva adreça ip, el que ens retornarà es un fitxer annomenat “index.html” amb els continguts de la web:



I per entrar al navegador, només ens cal entrar l’adreça ip a qualsevol navegador del sistema operatiu.

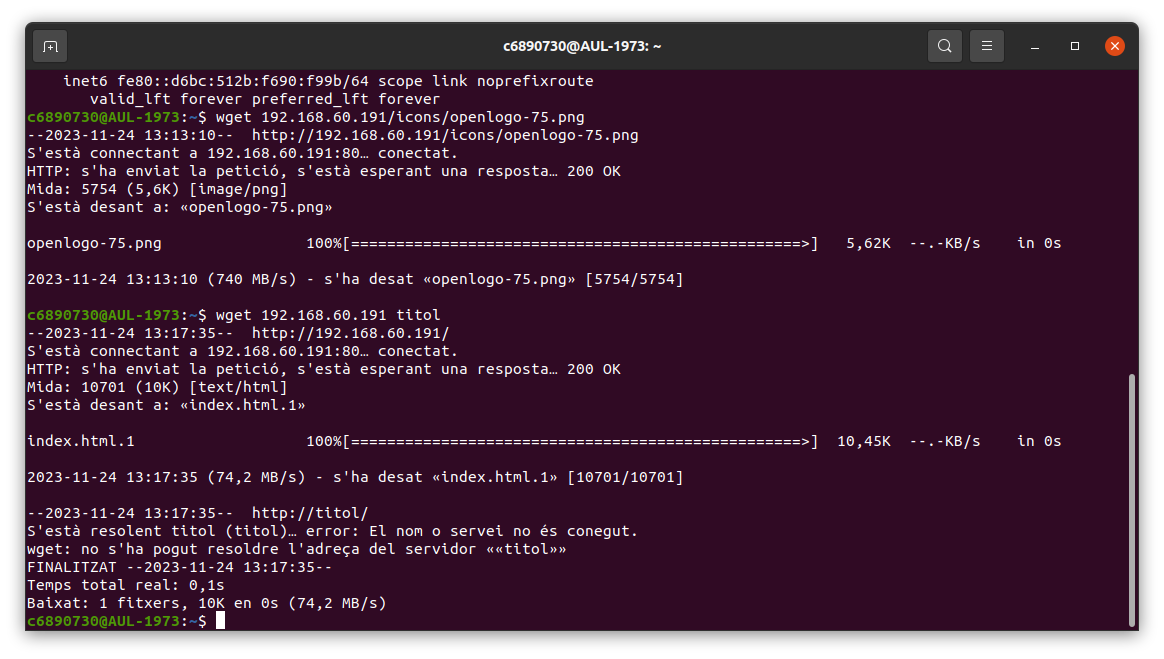
 

￼La primera captura és el fitxer del wget, podem veure que només hi ha un format de recurs el text, ja que el wget si no es especificat només guardarà el recurs tipus text de l’adreça.

La segona captura és des del navegador al servidor on tenim la web, podem observar que ara si que tenim diferents recursos com text imatges i links utilitzables.

El navegador està pensat per ser interactiu amb els usuaris i tenir un contingut dinàmic, pel contrari el wget descarrega un recurs estàtic de la web.

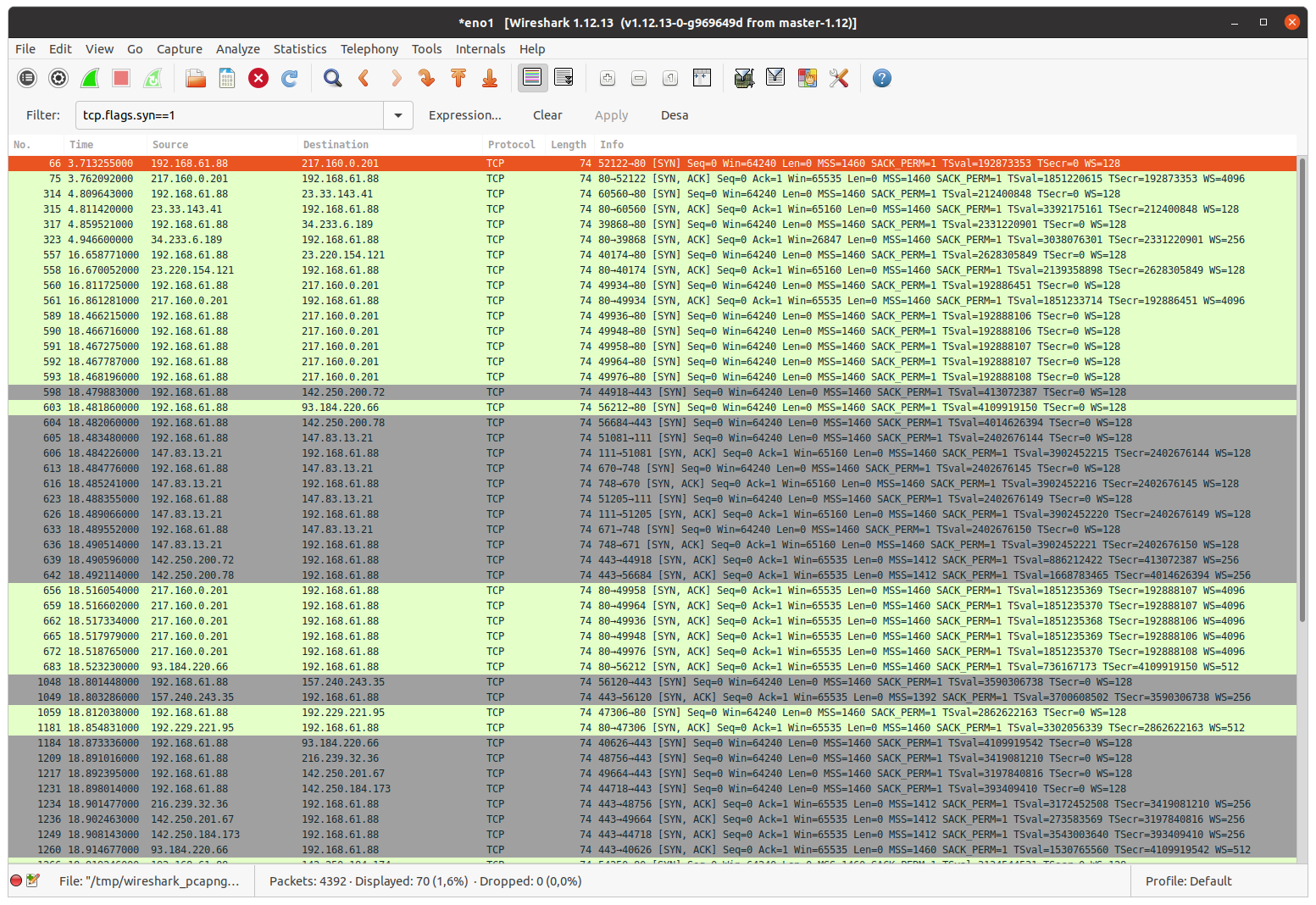
Si es fa un wget dins del servidor un recurs que no existeix, primer enviarà la petició al servidor, s’esperarà una resposta d'aquest, aquest retornarà un índex, però com que el recurs “titol” no existeix la petició wget no podrà ser completada i el index estarà buit.



**Preguntes HTTP**

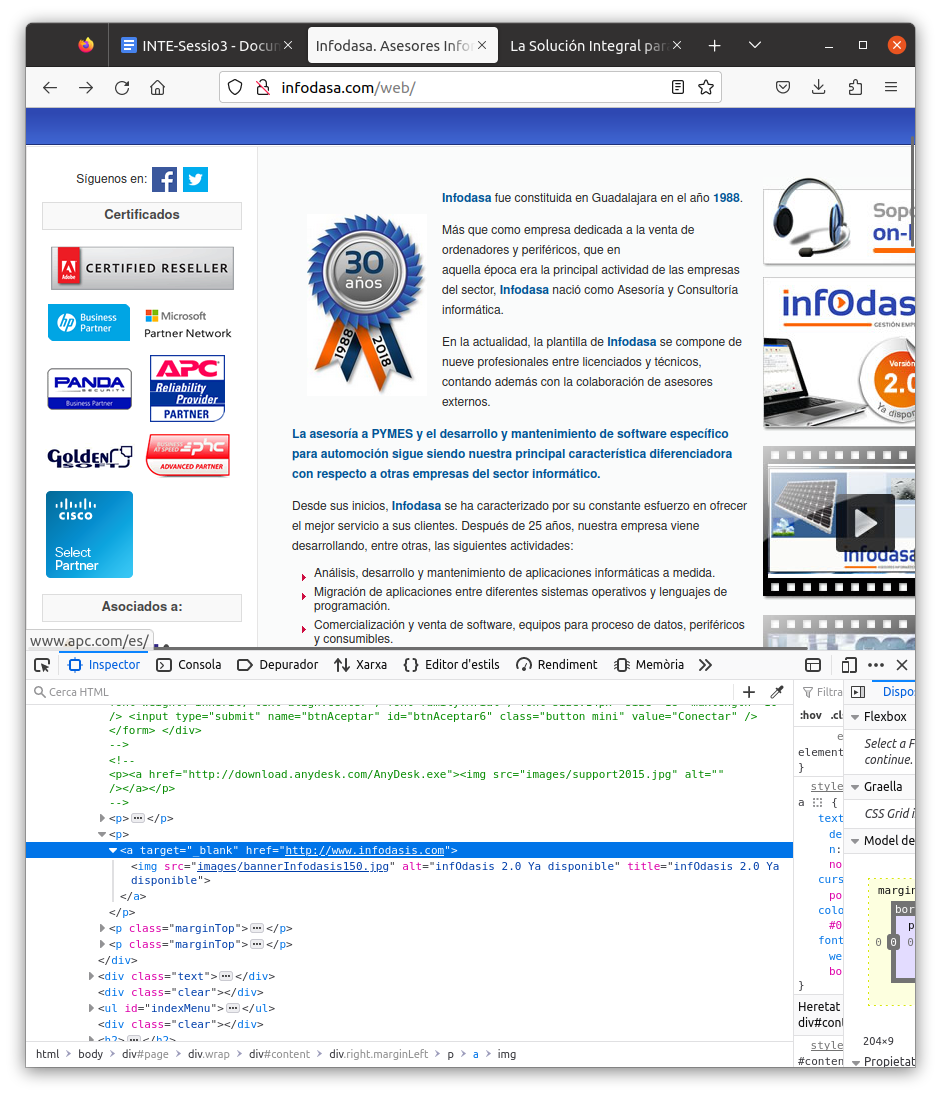
Per comptar les peticions fetes, el que podem fer és des de Wireshark posar el filtre de flags de paquets tcp.

En aquest cas al voler peticions les flags que volem activades es la de syn, ja que en el primer paquet per tenir una petició de http serà amb la flag syn activada, per començar la connexió.

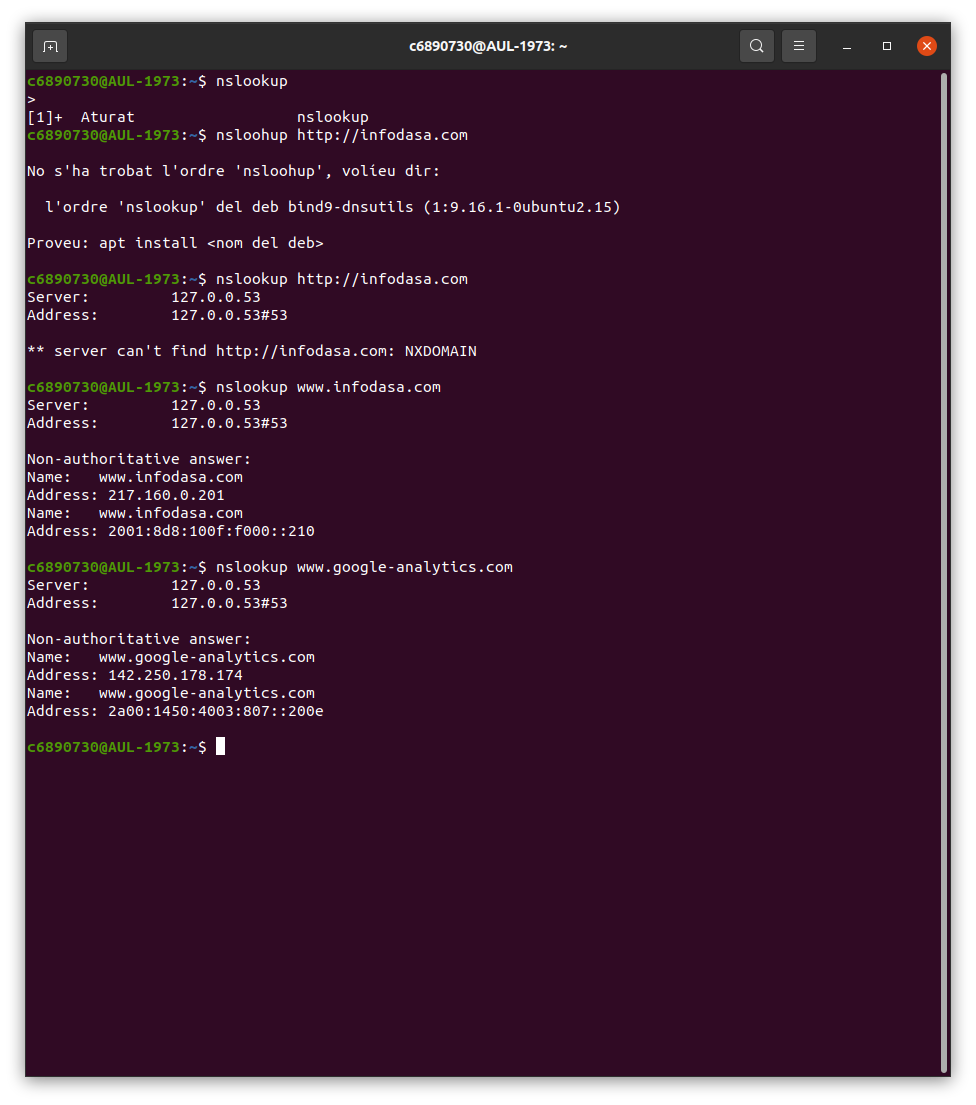


Podem mirar el nombre de connexions, podem mirar gràcies el protocol DNS l'adreça ip i després filtrar els paquets amb aquesta i contar quants s’han intercanviat.

Dins d’una web, podem inspeccionar i trobar els recursos que utilitza, en aquest cas, al entrar a la web infodasa.com/web/web, al inspeccionar la web el url inicial que ens dona es: <http://www.infodasis.com>

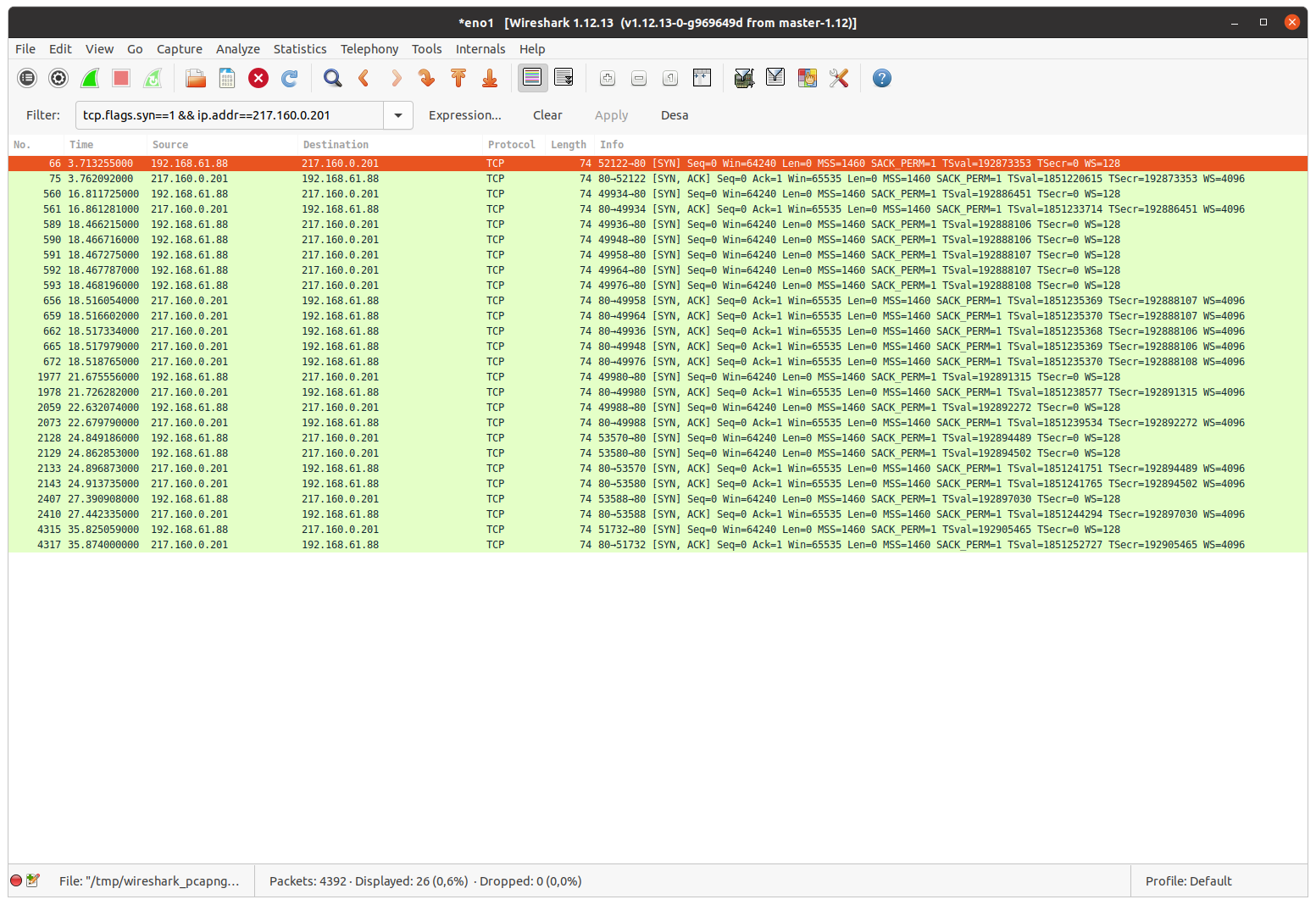


Amb la url de la pàgina web, busquem quina adreça ip té assignada pel protocol DNS amb *nslookup url* des de la terminal:



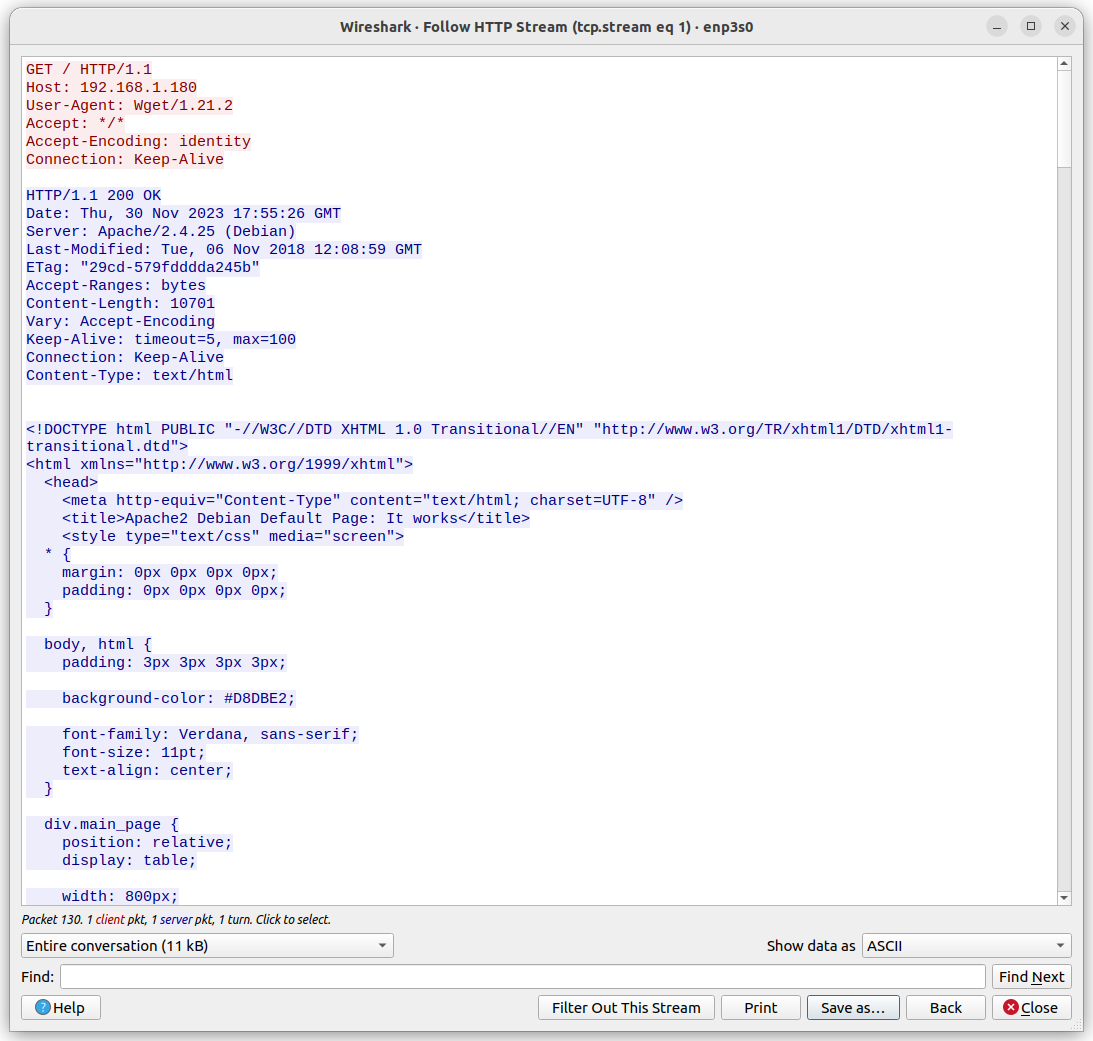
Després a la part de filtres de wireshark filtra per la flag tcp syn activada i l’adreça ip que ens ha donat la comanda nslookup.

Si ens hem connectat sortiran tots els paquets enviats per aquesta connexió, sino el llistat de paquets serà buit.



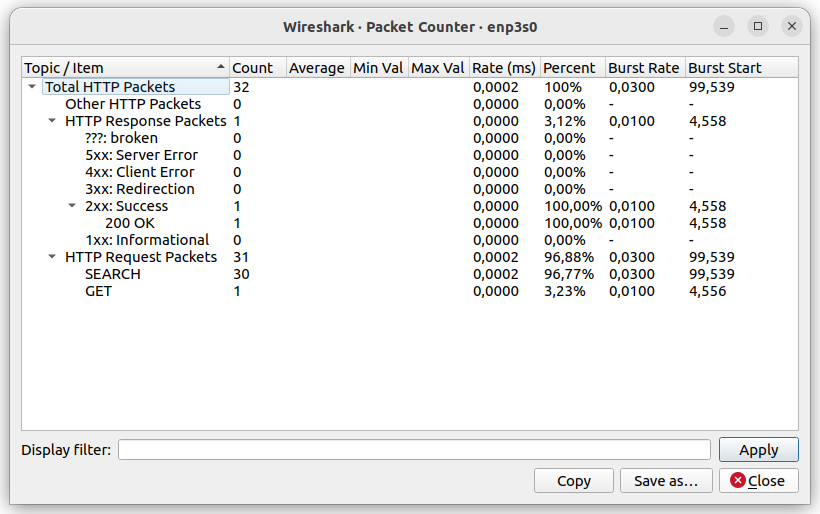
Podem veure i contar les diferents connexions fetes al estar una estona navegant dins de la pàgina [www.infodasa.com](http://www.infodasa.com) ( amb adreça ip 127.160.0.201)

Amb la eina HTTP Stream, podem veure tota la pàgina descarregada:

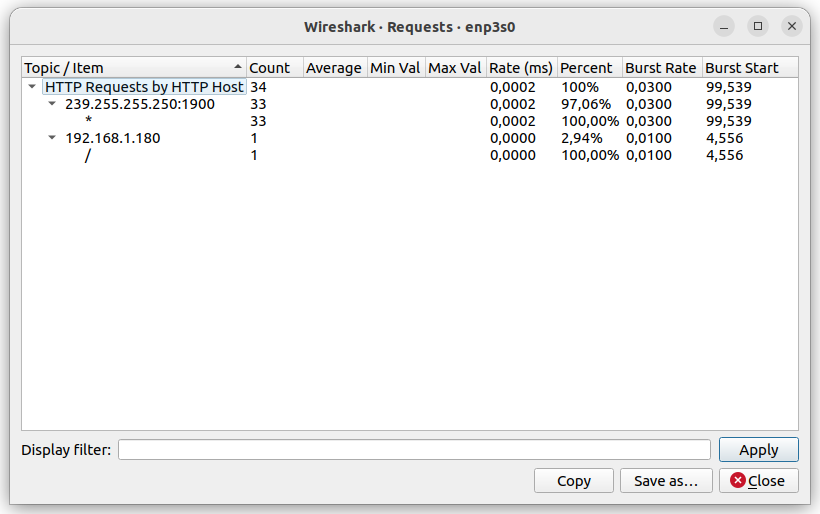


Mentres que la eina Statistics →HTTP dona la informació en forma d'estadístiques de la connexió tcp, com per exemple:

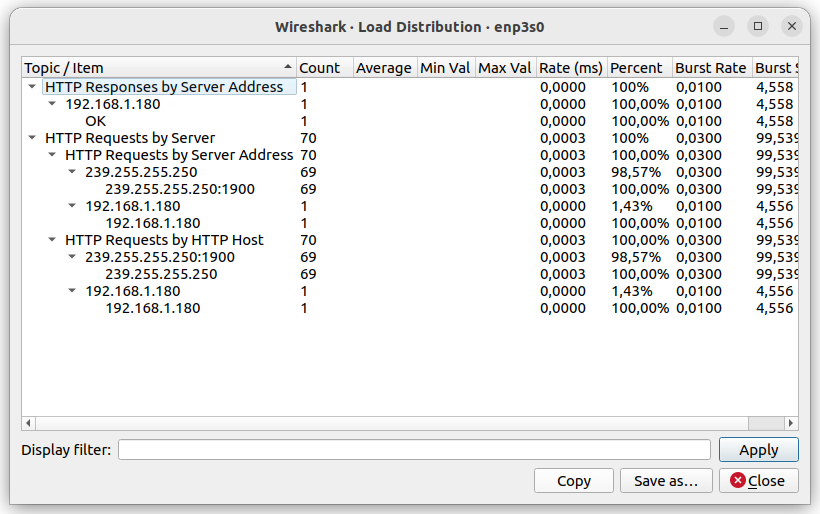
* Contador dels paquets intercanviats:



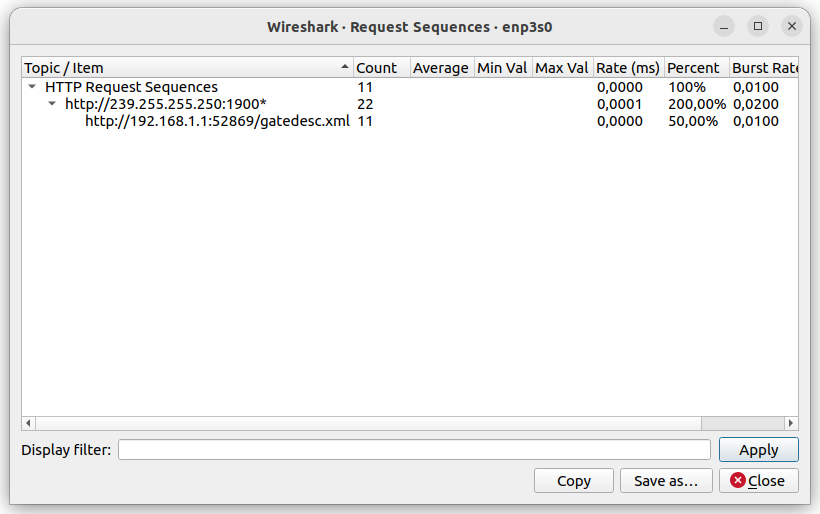
* Estadístiques dels paquets petició:



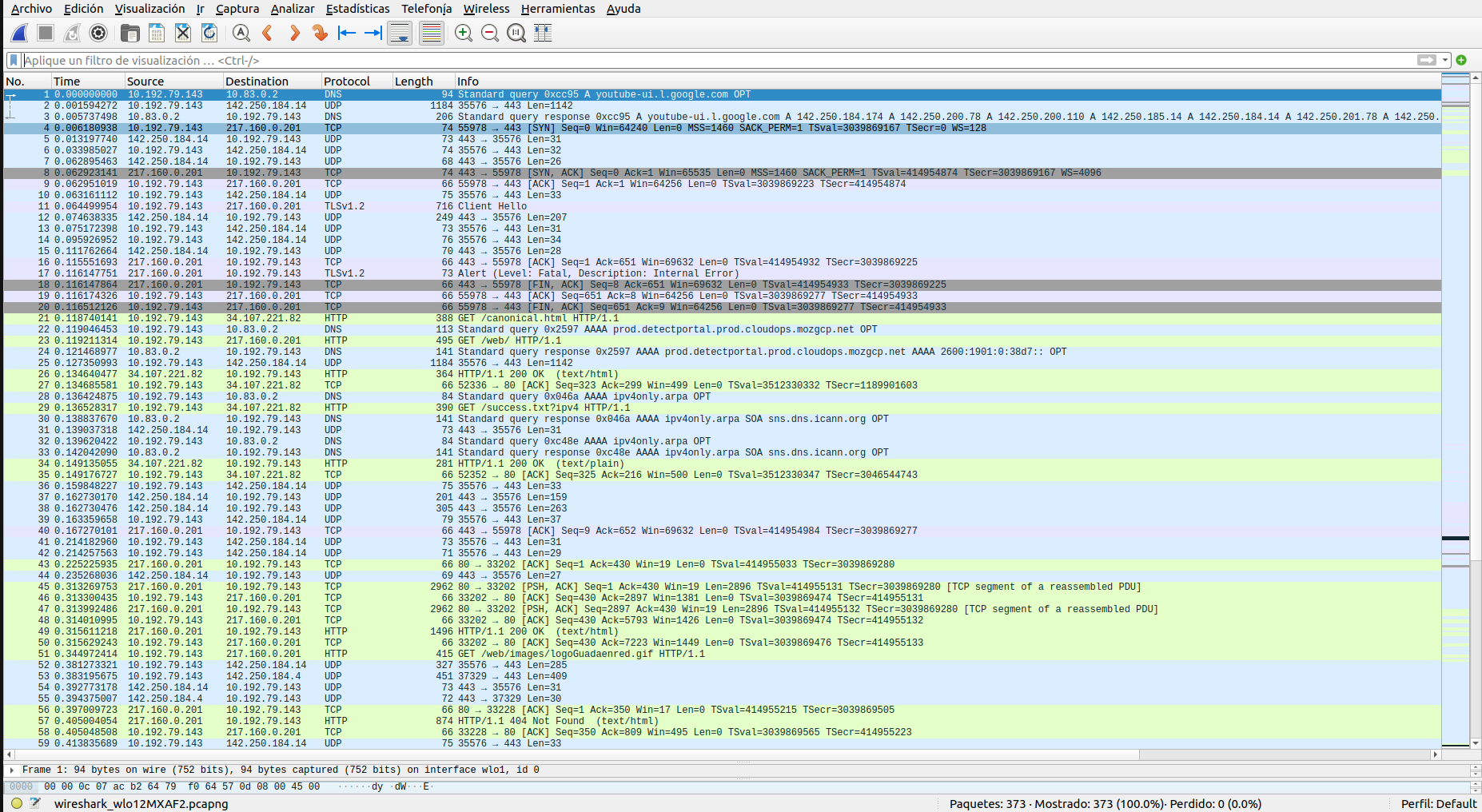
* De la distribució de càrrega de la connexió:



* Les seqüències de peticions:



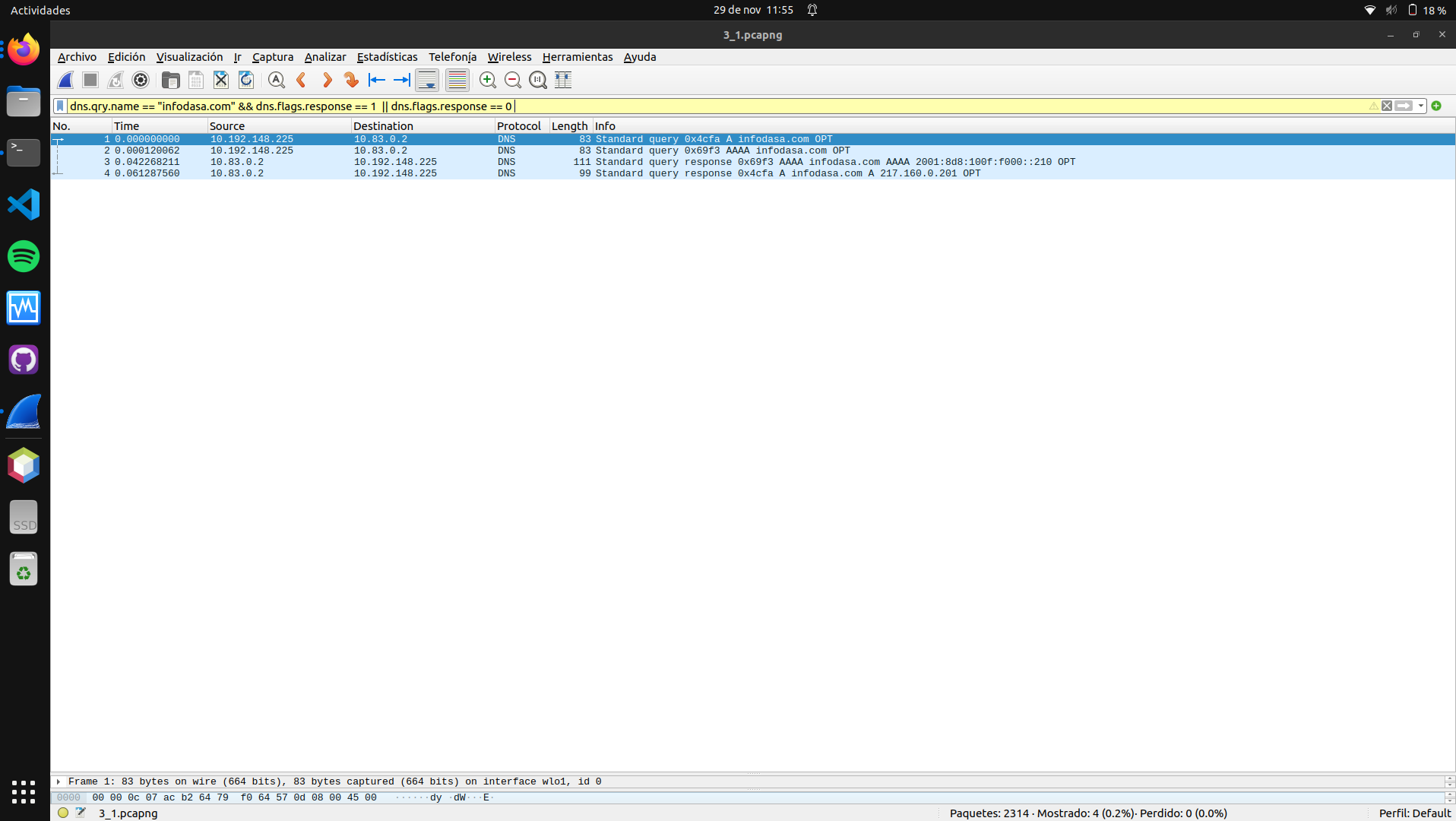
**Preguntes anàlisis de la latència**

Des de Wireshark podem mirar l’intercanvi de paquets quan s’entra a un url des d’un navegador:

Quan es po￼sa la URL http://infodasa.com/ al navegador i es carrega la pàgina, es produeix una sèrie d'esdeveniments de xarxa que es poden rastrejar mitjançant una captura de Wireshark.

1. **Resolució DNS**:
   * La captura comença amb una sol·licitud DNS (paquet #1) que no sembla estar relacionada amb infodasa.com. No hi ha cap sol·licitud DNS visible per a infodasa.com en la seqüència de paquets que has proporcionat, però normalment, aquest seria el primer pas: el navegador pregunta per l'adreça IP associada amb el nom de domini infodasa.com.
2. **Establiment de Connexió TCP**:
   * El navegador inicia una connexió TCP amb el servidor que allotja la pàgina web, com es mostra en els paquets #4, #8 i #9. Això s'indica amb l'intercanvi de paquets TCP amb les banderes [**SYN**] i **[SYN, ACK]**, establint la connexió.
3. **Negociació TLS**:
   * Una vegada establerta la connexió TCP, el navegador comença un handshake TLS amb un "**Client Hello**" (paquet #11) per assegurar la connexió.
4. **Sol·licituds HTTP**:
   * El navegador envia sol·licituds HTTP **GET** (paquets #21 i #23) per obtenir recursos com arxius HTML, imatges, etc. Aquests recursos podrien estar allotjats en diferents servidors, com es pot deduir per les múltiples adreces IP a les que es fan les peticions.
5. **Respostes del Servidor**:
   * El servidor respon amb els recursos sol·licitats, com s'indica en les respostes HTTP 200 **OK** (paquets #26 i #34), que significa que la pàgina web s'ha trobat i es transmet al navegador.
6. **Error en la Connexió**:
   * Hi ha un error de TLS (paquets #17 i #18) amb un servidor específic (217.160.0.201), que podria ser un recurs relacionat amb infodasa.com o un altre element que el navegador intentava carregar.
7. **Càrrega Addicional de Continguts**:
   * La seqüència de paquets mostra múltiples intents de càrrega de recursos addicionals (paquets #21 en endavant), que inclouen sol·licituds HTTP **GET** i comunicacions QUIC, una alternativa més ràpida a TCP i TLS per HTTPS.
8. **Errors de Resolució DNS**:
   * Diversos intents de resolució DNS (paquets #171 en endavant) fallen amb respostes "No such name", suggerint que hi ha sol·licituds per noms de domini que no es poden resoldre, possiblement a causa de dominis mal escrits o que no existeixen.
9. **Finalització de la Càrrega**:
   * Finalment, després de rebre els recursos necessaris, el navegador processa i renderitza la pàgina web per a la visualització de l'usuari.

Des de wireshark podem filtrar per només tenir paquets del protocol DNS:



Per fer això, mirem la diferència de temps entre la sol·licitud de consulta DNS i la resposta DNS. En la captura de pantalla, podem veure:

* Paquet 2: Sol·licitud de consulta DNS (Standard query) per infodasa.com.
* Paquet 4: Resposta DNS (Standard query response) per infodasa.com.

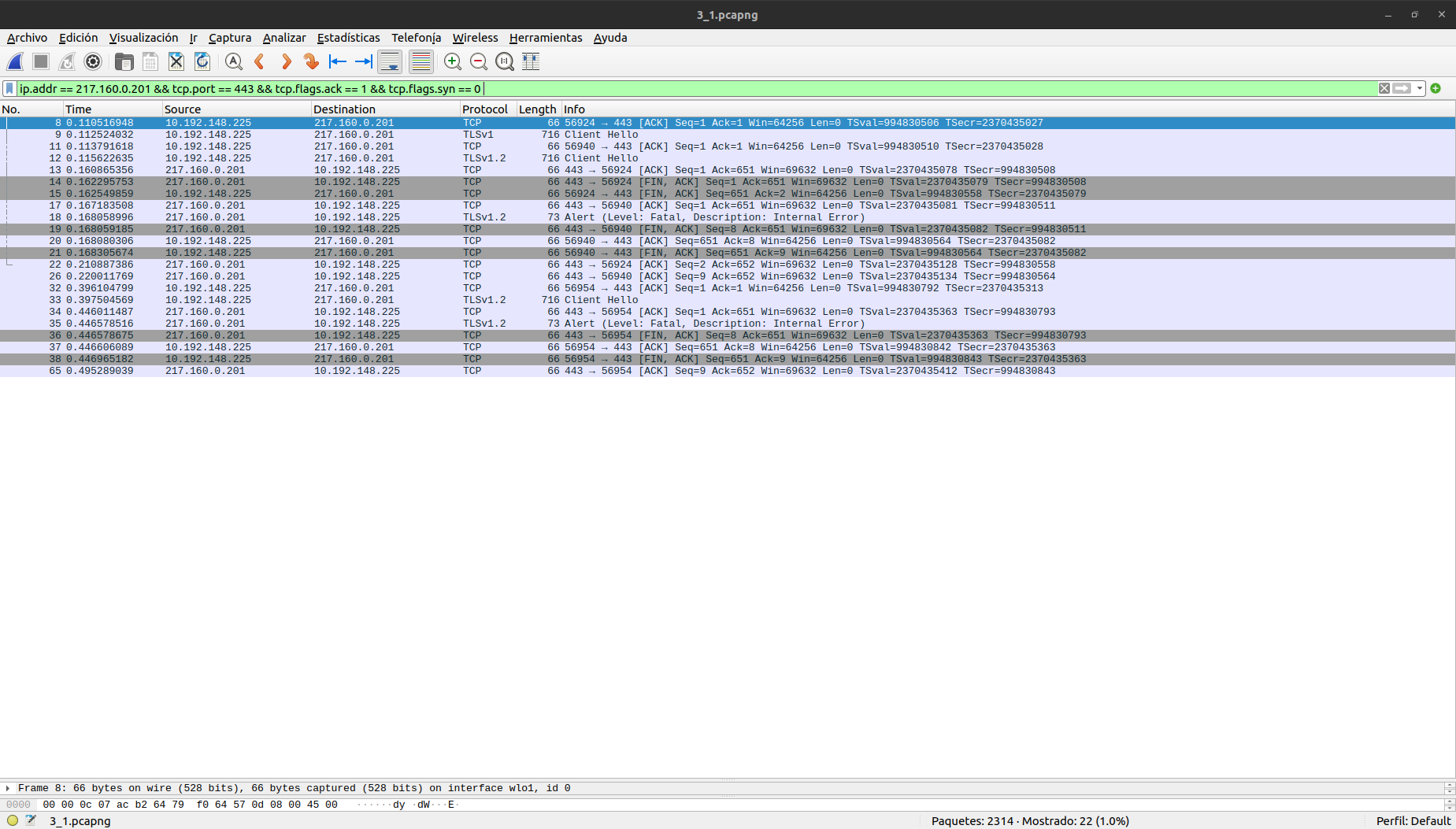
Els temps per aquests paquets són:

* Paquet 2 (Consulta): 0.000180962 segons
* Paquet 4 (Resposta): 0.061287560 segons

La diferència entre aquests dos temps ens dona el temps de resolució DNS. Restant el temps de la consulta del temps de la resposta, obtenim: 0.061287560 segons - 0.000180962 segons = 0.061106598 segons

Això significa que el sistema operatiu va trigar aproximadament 0.0611 segons a resoldre el nom del servidor web infodasa.com.

Des de Wireshark filtrem per aconseguir els paquets relacionats amb l'establiment d'una connexió TCP al servidor amb adreça IP 217.160.0.201 pel port 443, que és el port estàndard per a HTTPS.

Per calcular el temps que triga a establir la connexió, necessitem trobar una seqüència completa de three-way handshake, entre el client (10.192.148.225) i el servidor (217.160.0.201), els paquets trobats son:

* Paquet 10: SYN enviat des del client al servidor.
* Paquet 12: SYN-ACK responent des del servidor al client.
* Paquet 14: ACK final des del client al servidor.

Els temps d'arribada d'aquests paquets són:

* Paquet 10 (SYN): 0.110736821 segons
* Paquet 12 (SYN-ACK): 0.113921631 segons
* Paquet 14 (ACK): 0.162255753 segons

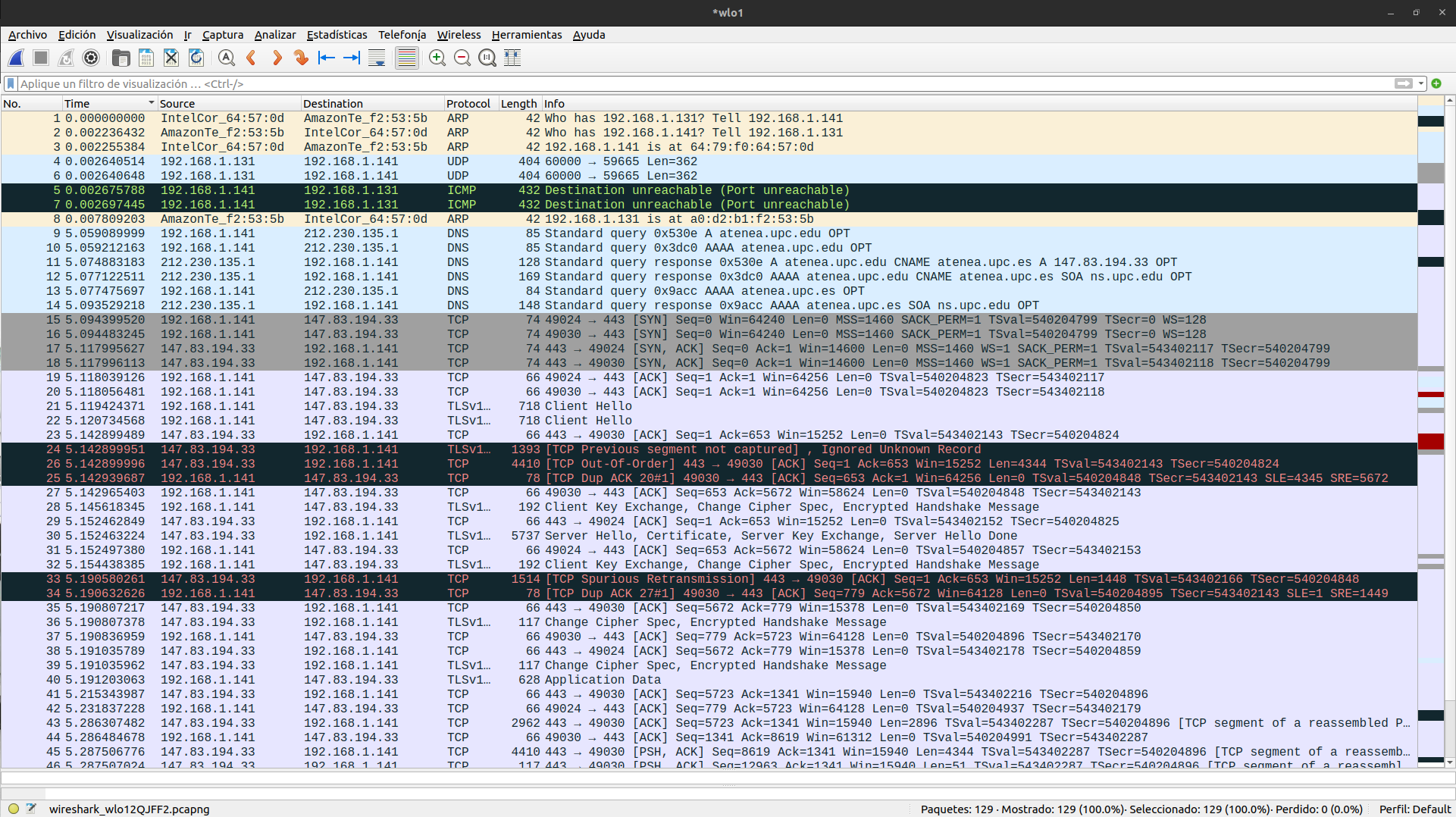
Per tant, el temps per establir la connexió és la diferència entre el temps del paquet SYN i el temps del paquet ACK final:

0.162255753 segons (ACK) - 0.110736821 segons (SYN) = 0.051518932 segons

Així que, segons aquesta captura de paquets, s’ha trigat aproximadament 0.0515 segons a establir la connexió TCP al servidor web.

**Preguntes HTTPS**

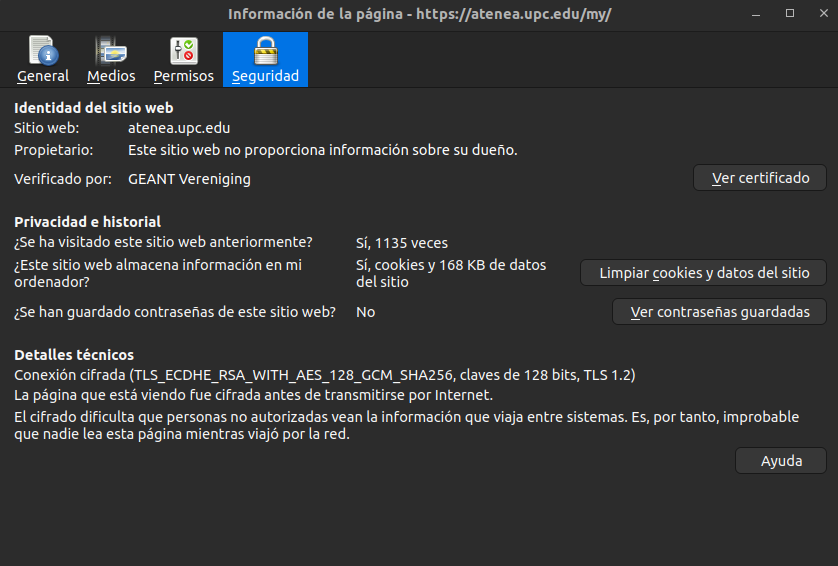
Des de Wireshark observem la connexió i possibles redireccions:

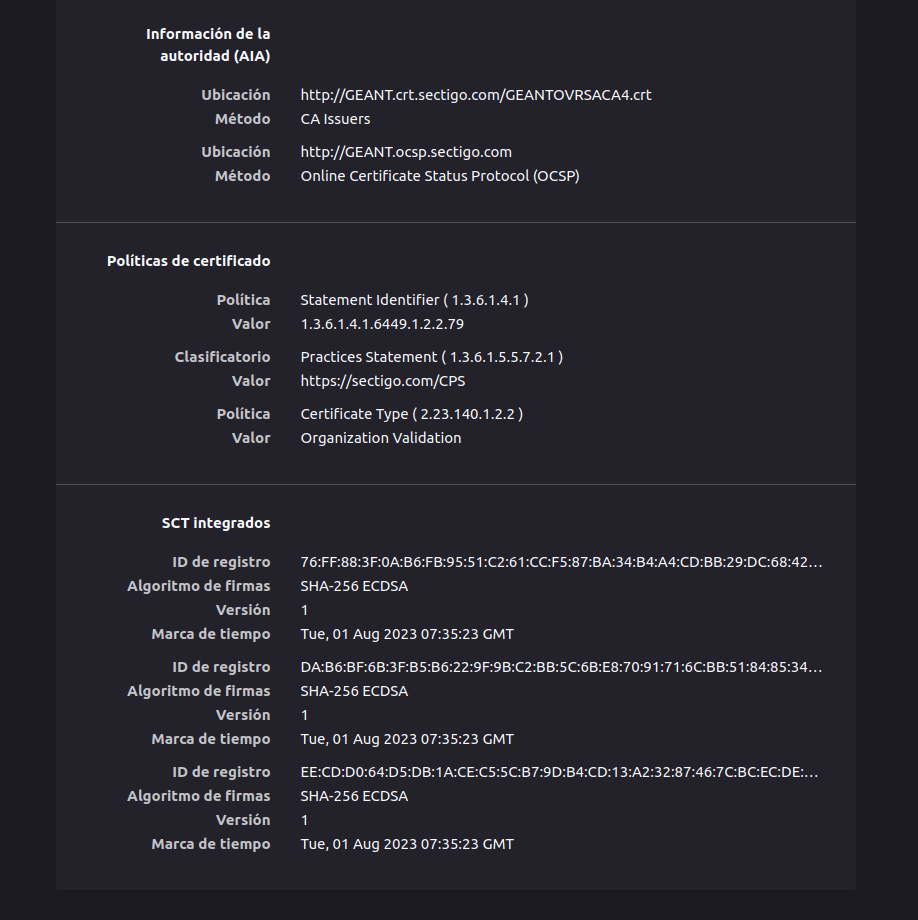


Podem fer un seguiment del procés que probablement ha portat a la redirecció del servidor web d'Atenea a la seva versió segura:

1. Els paquets 9 i 10 mostren **consultes DNS** per al domini atenea.upc.edu. Aquestes són les consultes que el teu dispositiu envia per resoldre l'adreça IP associada amb el nom de domini.
2. Els paquets 11 i 12 són les **respostes** **DNS**. Indiquen que atenea.upc.edu és un àlies (CNAME) per atenea.upc.es, i proporcionen l'adreça IP 147.83.194.33 per a aquest últim nom de domini.
3. Amb aquesta adreça IP resolta, els paquets 15 i 16 mostren intents de connectar-se al servidor utilitzant el protocol TCP al port 443, que és el port estàndard per **HTTPS**, la versió segura de HTTP.
4. Els paquets 17 i 18 són les respostes del servidor (**SYN, ACK**) als intents de connexió, i els paquets 19 i 20 són els reconeixements d'aquestes respostes.
5. Finalment, els paquets 21 i 22 mostren missatges "Client Hello" de TLSv1.2, que són el començament del "handshake" TLS que estableix una connexió segura.

Aquesta seqüència suggereix que el navegador ja està intentant connectar-se de manera segura des del principi, possiblement perquè l'usuari ha entrat directament l'URL amb "https://" o perquè el lloc web ha estat configurat per fer servir HSTS (HTTP Strict Transport Security), que indica als navegadors que sempre utilitzin HTTPS quan es connectin.





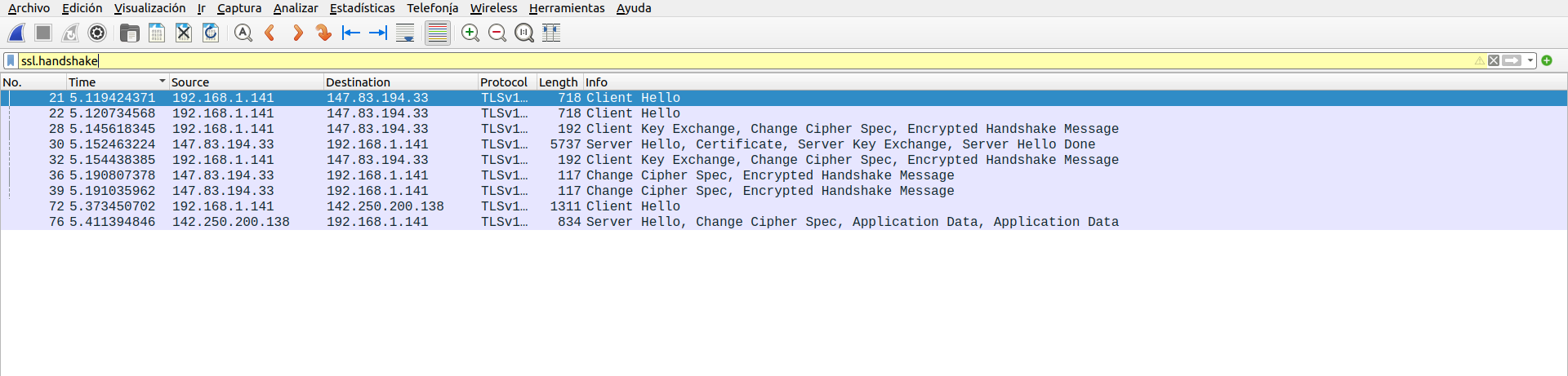
Basant-nos en les captures de pantalla, que mostren la informació de seguretat del navegador per a la pàgina de atenea.upc.edu:

El navegador mostra que la connexió és segura de diverses maneres:

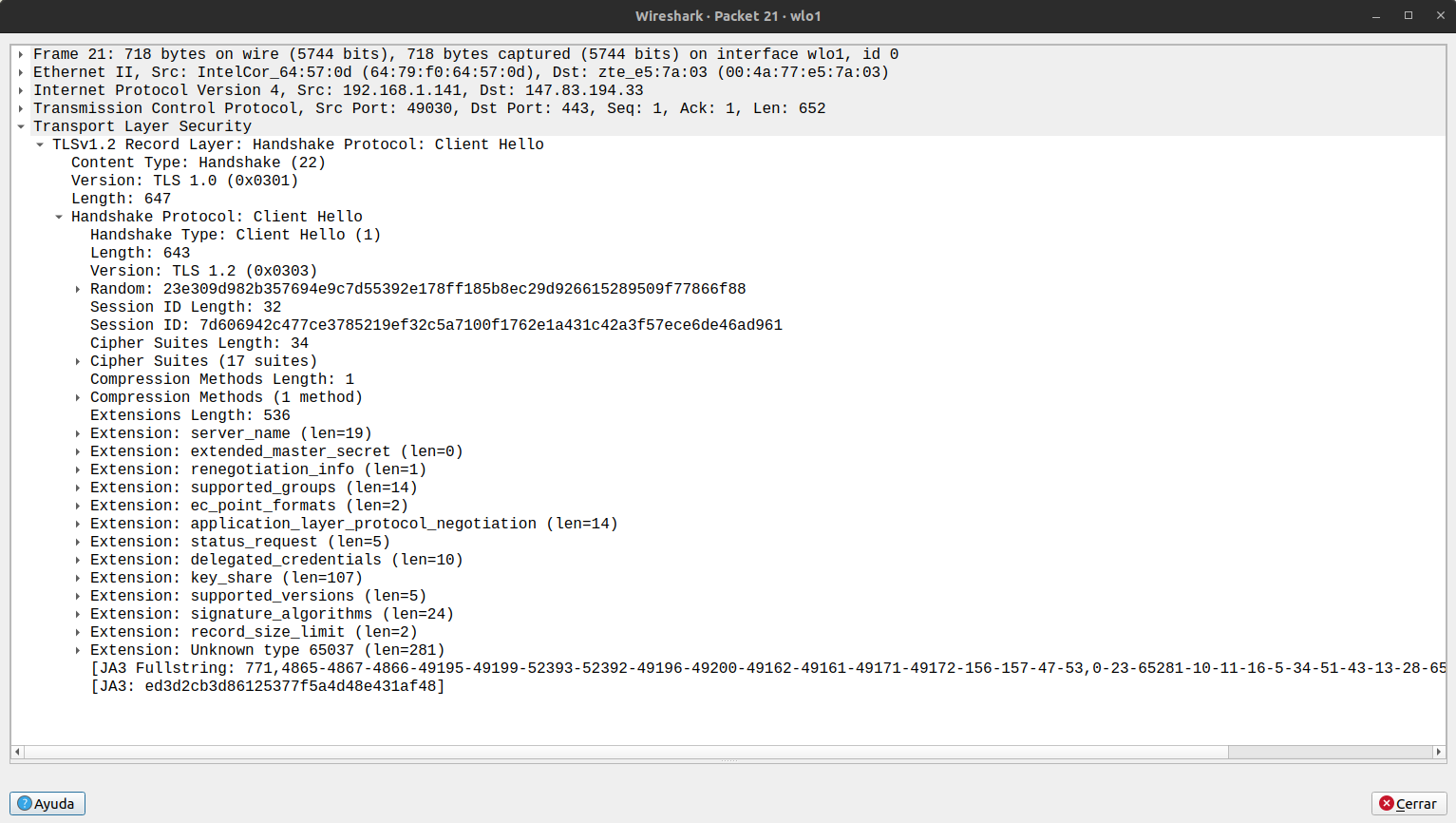
* La URL comença amb "https://", indicant que s'utilitza el protocol HTTPS.
* El navegador mostra que el lloc web està "Verificat per: GEANT Vereniging", que és probablement l'autoritat de certificació (CA) que ha emès el certificat SSL/TLS.
* A més, s'indica que s'utilitza una connexió xifrada (TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256, claus de 128 bits, TLS 1.2), que és una xifra forta per a la comunicació segura.

D'aquesta informació, podem treure que la connexió és segura perquè s'utilitza una xifra forta de TLS, que assegura la confidencialitat i integritat de les dades. El certificat SSL/TLS utilitzat ha estat emès per una autoritat de certificació reconeguda i confiable, que valida l'autenticitat del servidor al qual està connectat el navegador.

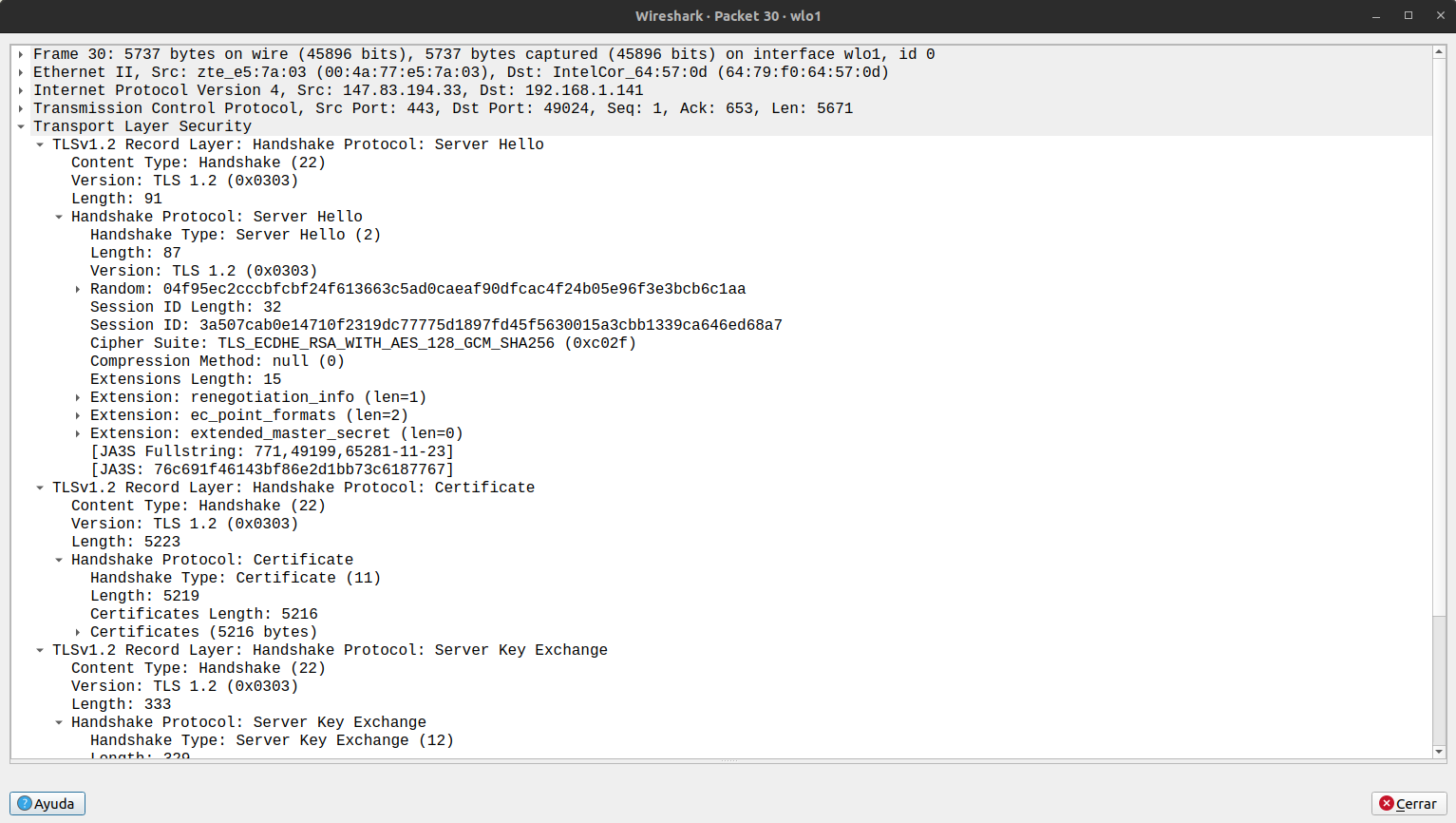
En el Wireshark podem veure la informació específica dels paquets al establir una connexió segura:



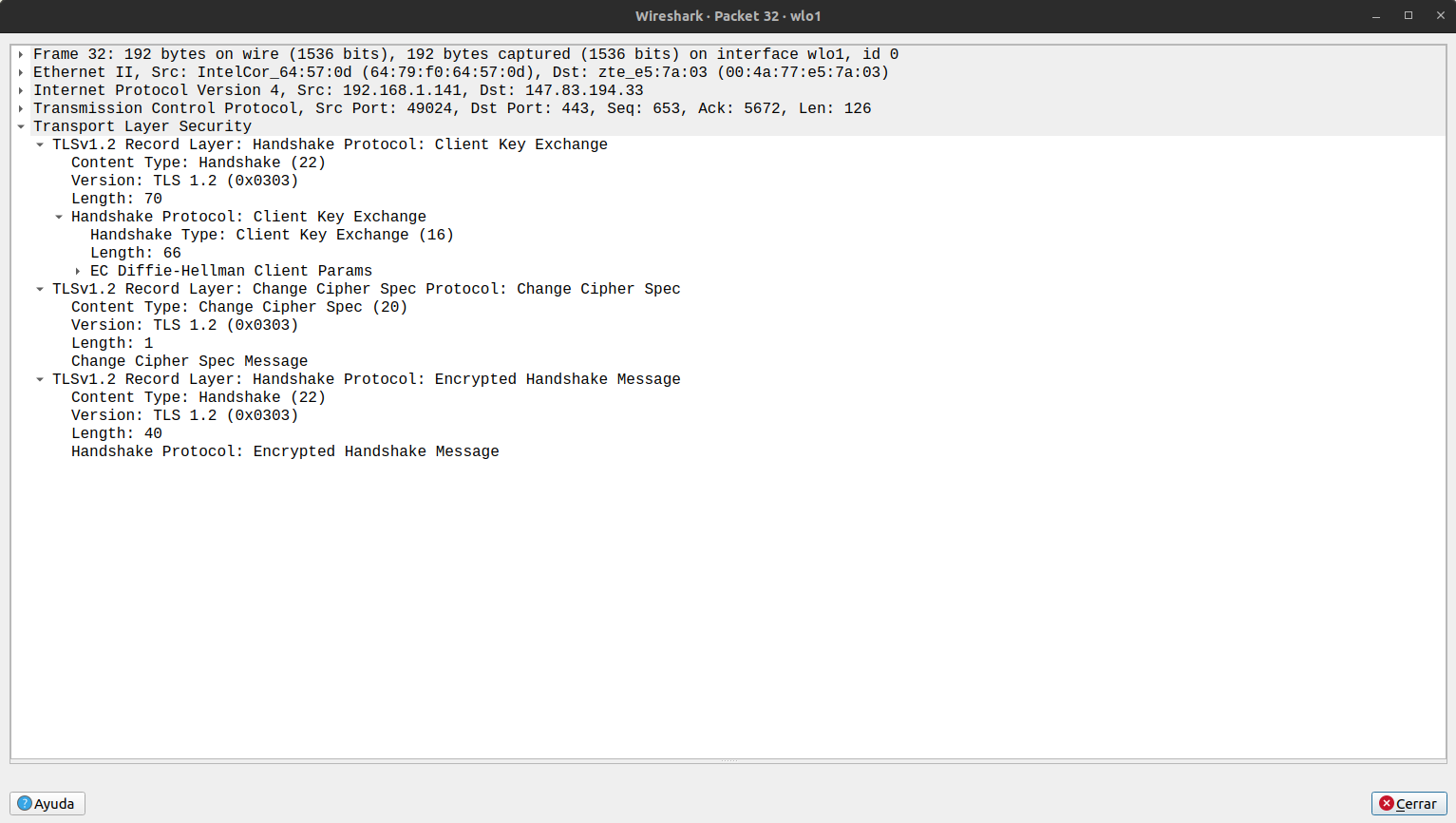
1. **Paquets de "Client Hello" (paquet 21 i 22):** Aquests paquets inicien la negociació de seguretat i contenen informació sobre els protocols i xifres que el client està disposat a utilitzar. Indica que la connexió entre el navegador i el servidor web està xifrada i és segura.

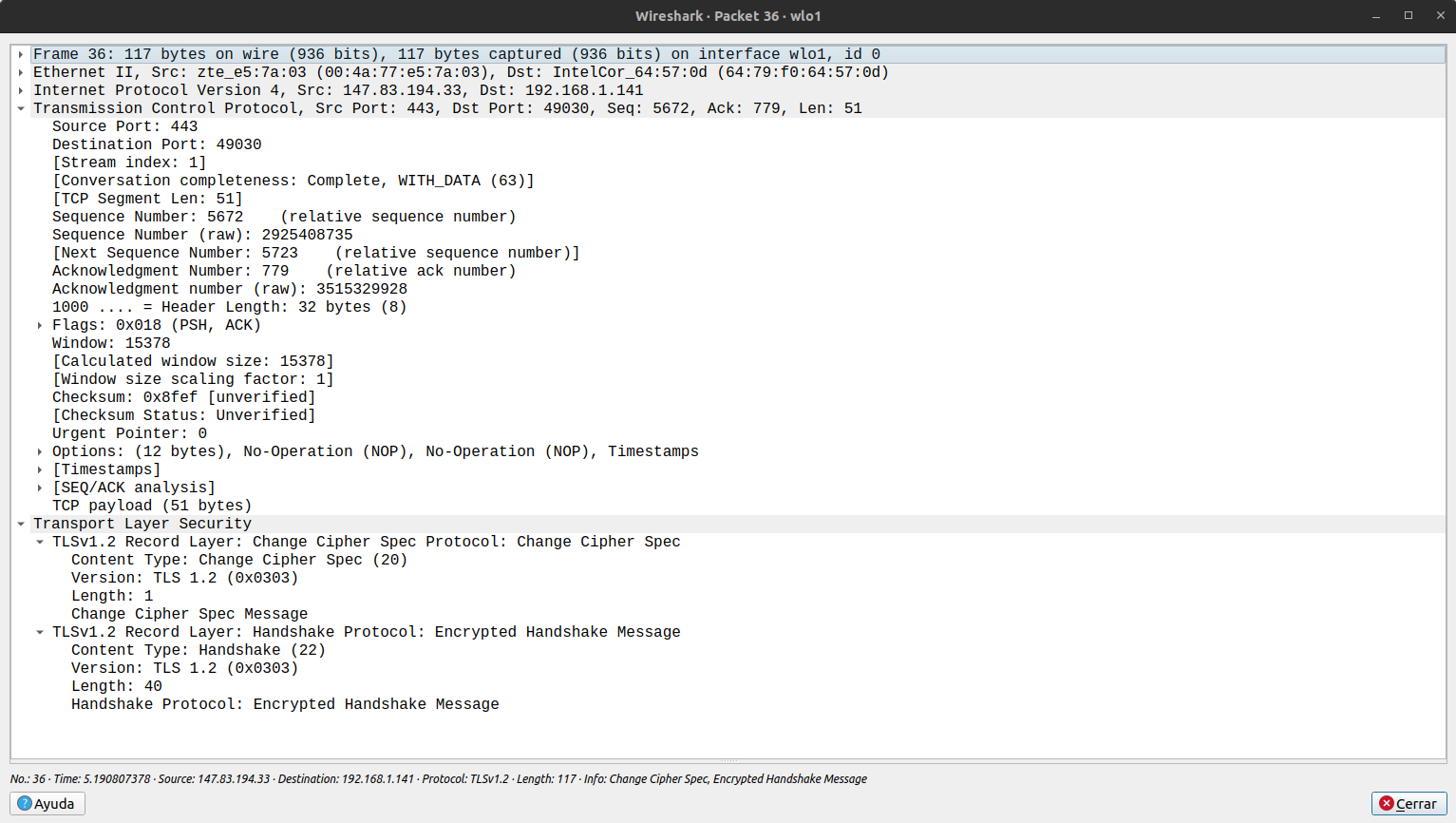


1. **Paquet de "Server Hello, Certificate" (paquet 30):** Aquest paquet conté la resposta del servidor, el certificat SSL/TLS, i els detalls de la xifra que el servidor ha seleccionat per utilitzar en la connexió segura.Es mostra qui ha emès el certificat (GEANT Vereniging), la política del certificat, i altres detalls tècnics com ara els algoritmes de xifrat utilitzats i les marques de temps.



1. **Paquets de "Client Key Exchange, Change Cipher Spec" (paquet 32):** Aquests paquets indiquen que el client està enviant informació criptogràfica al servidor (com ara la clau premaster encriptada) i que està a punt de canviar a l'encriptació per a la resta de la sessió.



1. **Paquets de "Change Cipher Spec, Encrypted Handshake Message" (paquet 36):** Aquests indiquen que tant el client com el servidor han acordat utilitzar una xifra específica i estan començant a comunicar-se utilitzant l'encriptació establerta.
2. **Paquets amb "Application Data" encriptat (paquet 76):** Aquests paquets contenen dades de l'aplicació que ja estan encriptades i segures, indicant que la connexió ha estat establerta amb èxit i que les dades estan protegides.

