HTTP/2

Redes y Comunicaciones

Qué es HTTP/2?

- Reemplazo de cómo HTTP se transporta.
- No es un reemplazo del protocolo completo.
- Se conservan métodos y semántica.
- Base del trabajo protocolo desarrollado por Google SPDY/2.
- Definido en:
  - RFC7540: Hypertext Transfer Protocol version 2.
  - RFC7540: HPACK - Header Compression for HTTP/2 RFC7541.

Problemas con HTTP/1.0, HTTP/1.1

- Un request por conexión, por vez, muy lento.
- Alternativas (evitar HOL):
  - Conexiones persistentes y pipelining.
  - Generar conexiones paralelas.

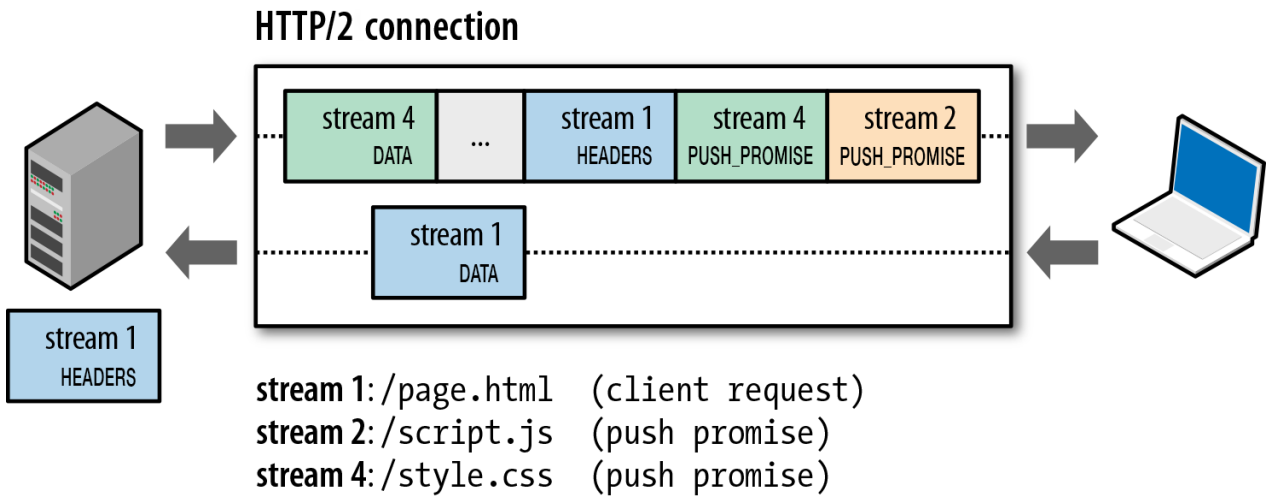
- Problemas:
  - Pipelining requiere que los responses sean enviado en el orden solicitado, HOL posible.
  - POST no siempre pueden ser enviados en pipelining.
  - Demasiadas conexiones genera problemas, control de congestión, mal uso de la red.
  - Muchos requests, muchos datos duplicados (headers).

Diferencias principales con HTTP/1.1

- Protocolo binario en lugar de textual(ASCII), binary framing: (más eficiente).
- Multiplexa varios request en una petición en lugar de ser una secuencia ordenada y bloqueante.
- Utilizar una conexión para pedir/traer datos en paralelos, agrega: datos fuera de orden, priorización, flow control por frame.
- Usa compresión de encabezado.
- Permite a los servidores “pushear” datos a los clientes.
- La mayoría de las implementaciones requieren TLS/SSL, no el estándar.

HTTP/2 mux stream, framing

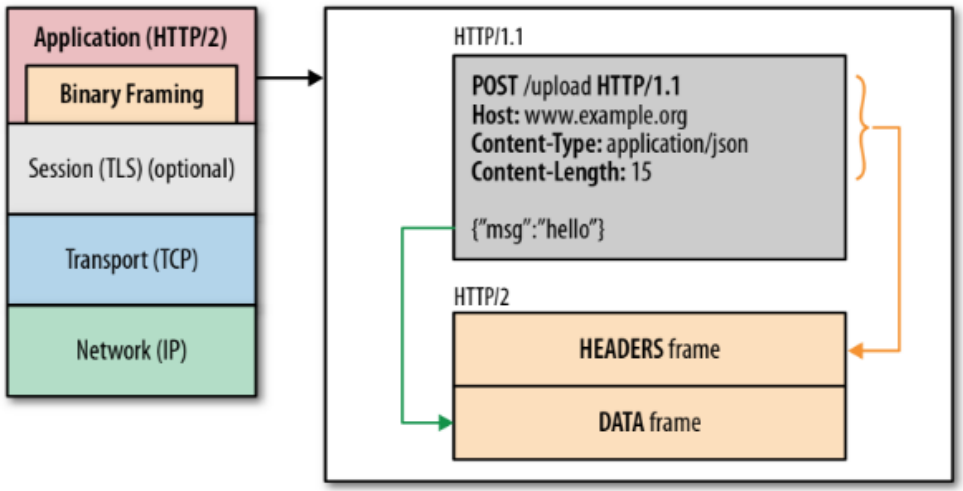
- Todos los streams en una misma conexión.
- Los streams son identificados y divididos en frames.



fuente: https://docs.google.com/presentation/d/1r7QXGYOLCh4fcUq0jDdDwKJWNqWK1o4xMtYpKZCJYjM/present?slide=id.p19

HTTP/2 mux stream, framing (Cont.)

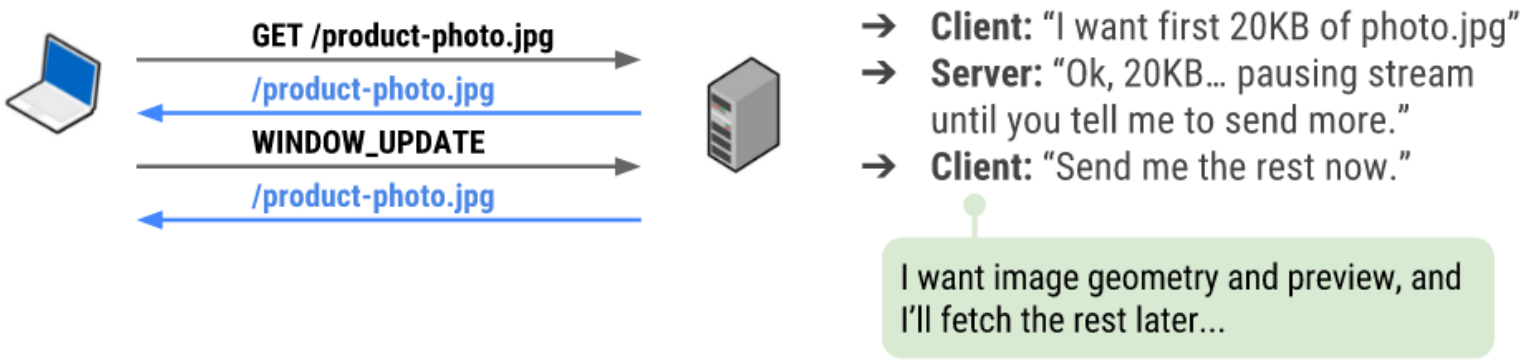
- Streams codificados en binario y cada frame header común fijo (9B).



fuente: https://docs.google.com/presentation/d/1r7QXGYOLCh4fcUq0jDdDwKJWNqWK1o4xMtYpKZCJYjM/present?slide=id.p19

HTTP/2 priorización y flow-control

- Los streams dentro de una misma conexión tienen flow-control individual.
- Los streams pueden tener un weight (prioridad).
- Los streams pueden estar asociados de forma jerárquica, dependencias.



fuente: https://docs.google.com/presentation/d/1r7QXGYOLCh4fcUq0jDdDwKJWNqWK1o4xMtYpKZCJYjM/present?slide=id.p19

HTTP/2 inline vs. push

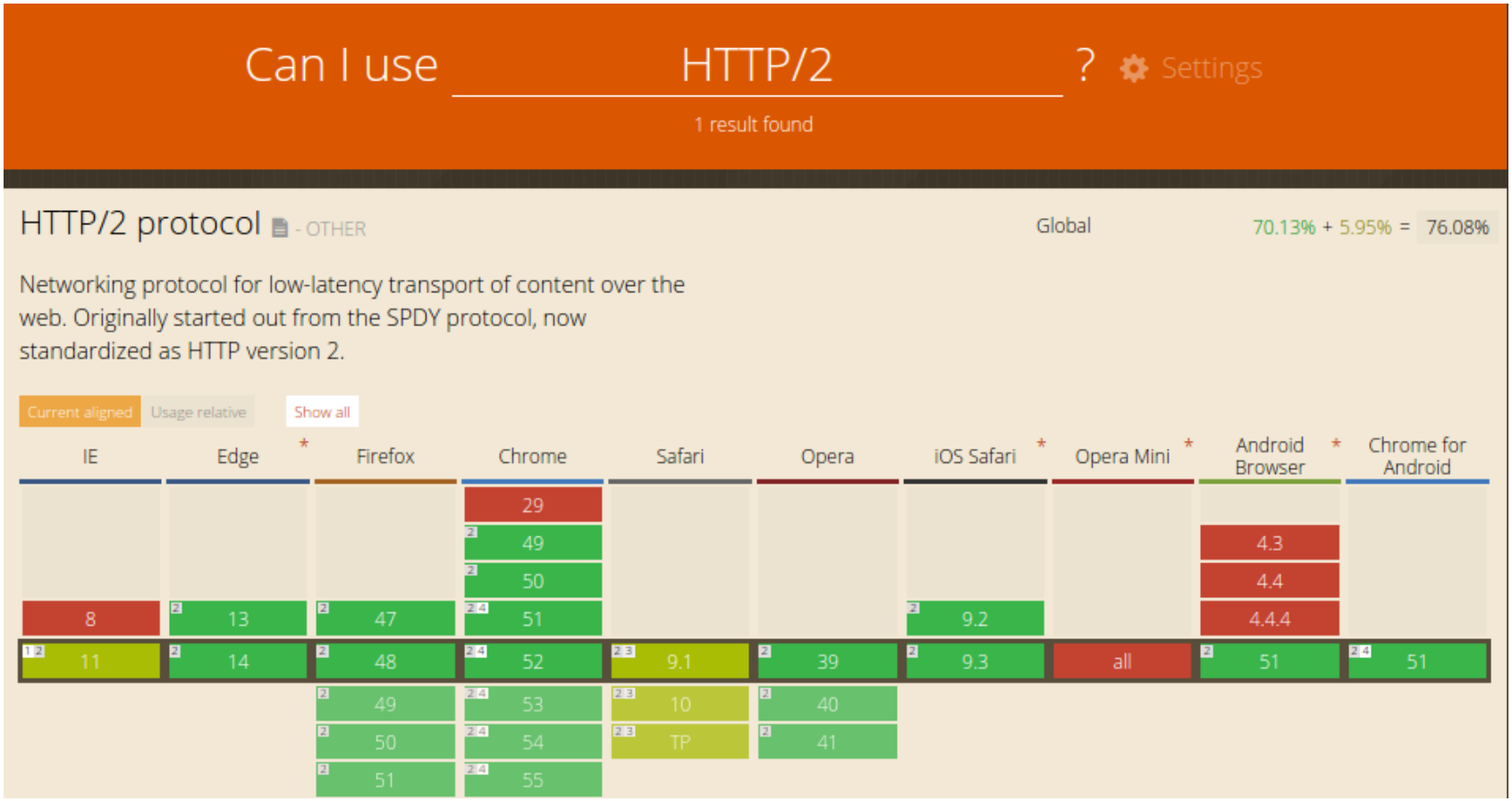
- Cuando el cliente solicita una página, “parsea” el primer response HTML luego solicita el resto.
- El server puede enviar el HTML más otros datos, por ejemplo CSS o Javascript.
- No siempre es lo que necesita el cliente, depende de que funcionalidad ofrece.

fuente: https://docs.google.com/presentation/d/1r7QXGYOLCh4fcUq0jDdDwKJWNqWK1o4xMtYpKZCJYjM/present?slide=id.p19

Compresión y Soporte

- Compresión de encabezados.
- SPDY/2 propone usar GZIP.
- GZIP + cifrado, tiene “bugs” utilizados por atacantes.
- Se crea un nuevo compresor de Headers: HPACK.
- H2 y SPDY, soportados en la mayoría de los navegadores.

Soporte en clientes

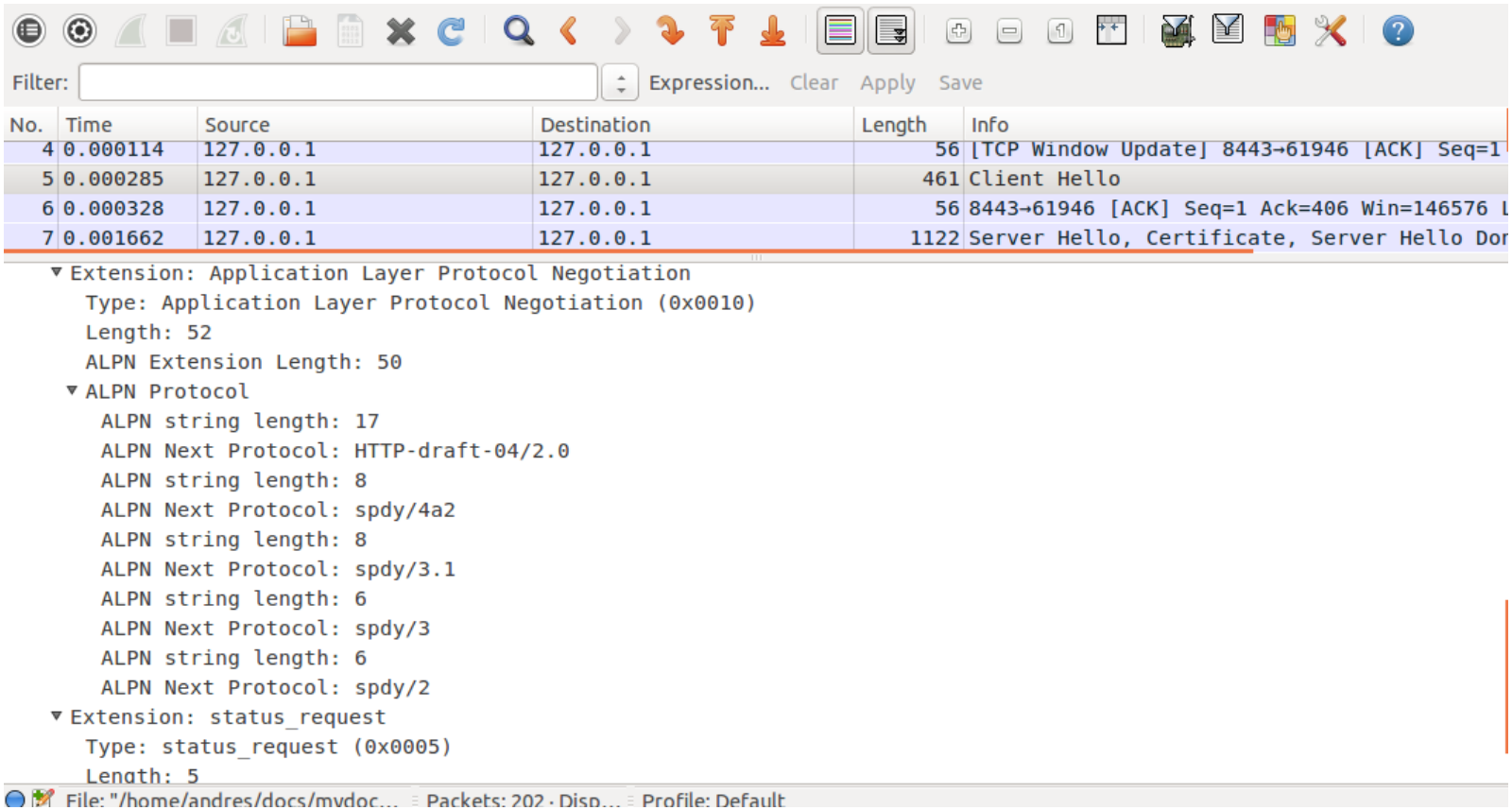


fuente: http://caniuse.com/#search=HTTP %2F2

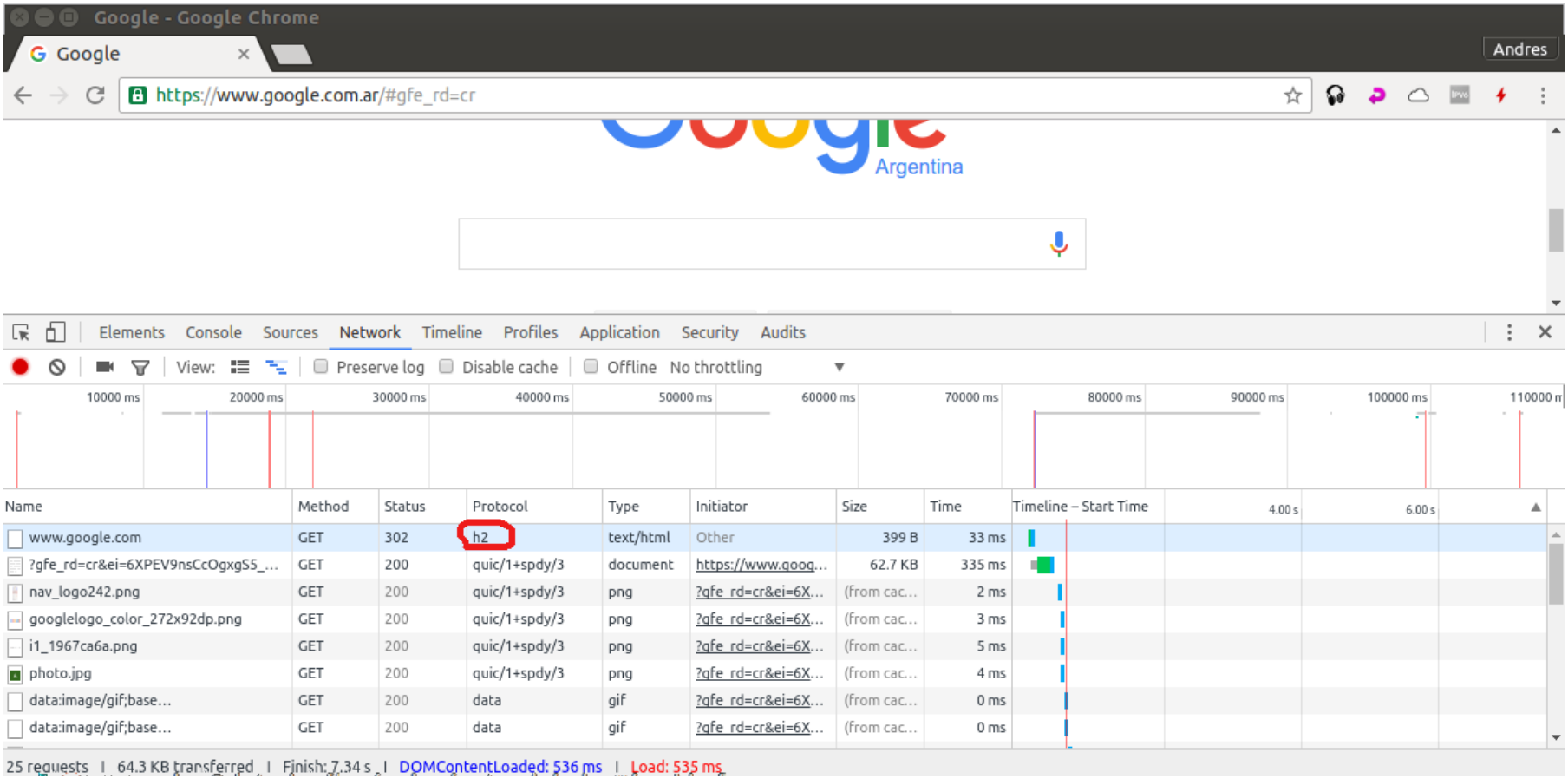
Otras Características

- HTTP/1.1, posibilidad de hacer un upgrade durante la conexión: Upgrade Header.
- Negociar el protocolo de aplicación:
  ALPN: Application-Layer Protocol Negotiation.
  Se negocia como extensión de SSL en Hello (Anteriormente NPN).
- Posibilidad de negociar protocolo alternativo:
  Alterantive Service: `alt-svc`.

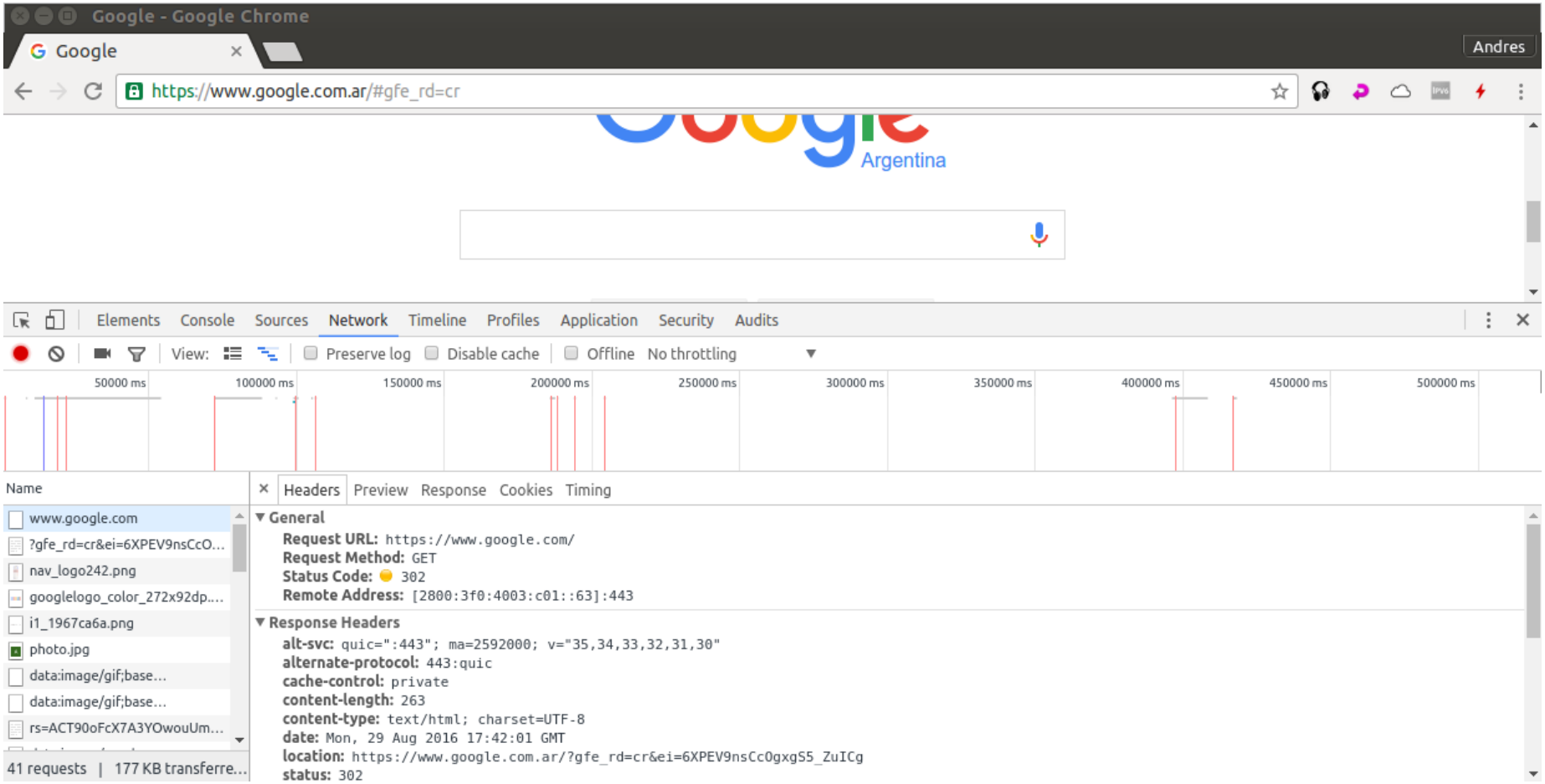
Application-Layer Protocol Neg.



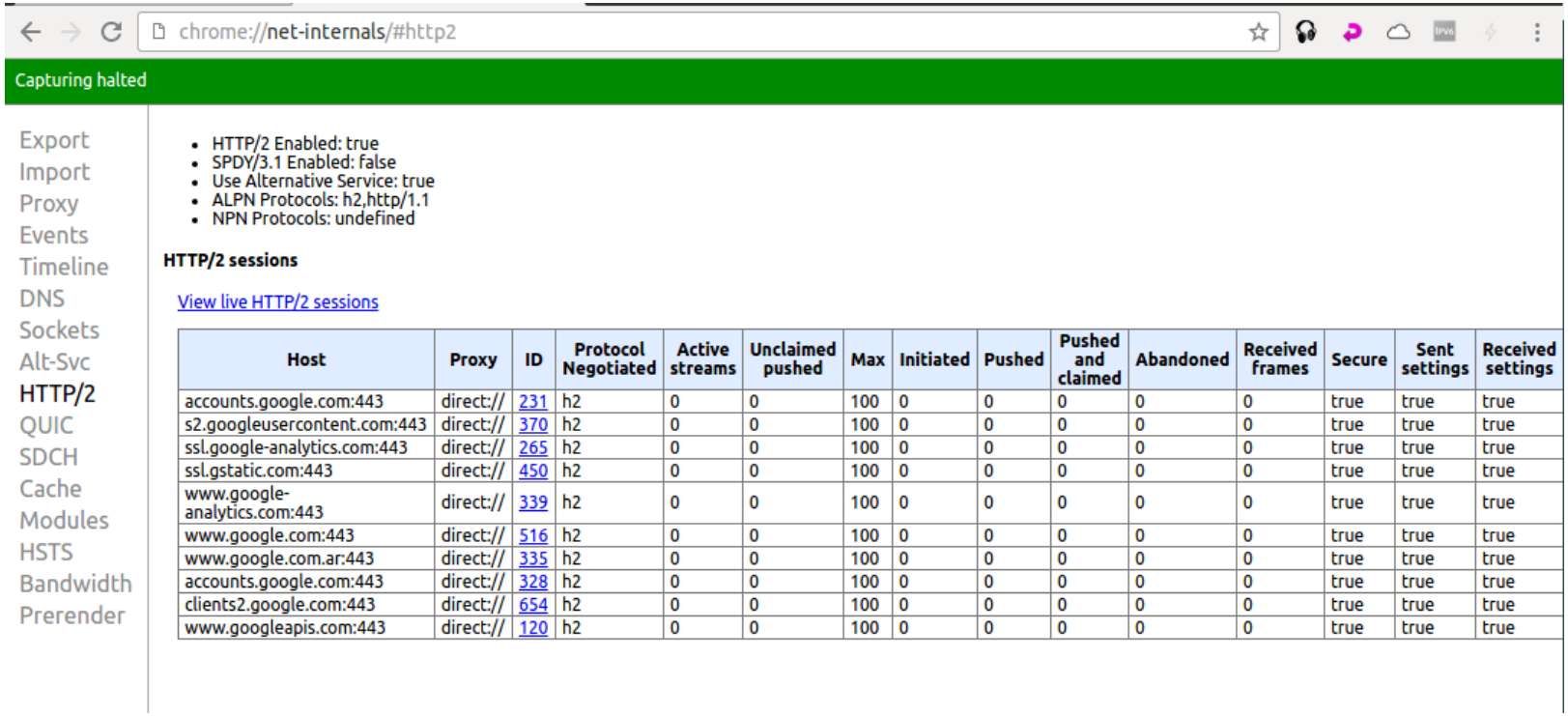
Debugging


Google - Google Chrome
Google
Andres
https://www.google.com.ar/#gfe_rd=cr
Argentina
Elements Console Sources Network Timeline Profiles Application Security Audits
View: Preserve log Disable cache Offline No throttling
10000 ms 20000 ms 30000 ms 40000 ms 50000 ms 60000 ms 70000 ms 80000 ms 90000 ms 100000 ms 110000 m
Name Method Status Protocol Type Initiator Size Time Timeline – Start Time 4.00 s 6.00 s
www.google.com GET 302 h2 text/html Other 399 B 33 ms
?gfe_rd=cr&ei=6XPEV9nsCcOgxgS5_... GET 200 quic/1+spdy/3 document https://www.goog... 62.7 KB 335 ms
nav_logo242.png GET 200 quic/1+spdy/3 png ?gfe_rd=cr&ei=6X... (from cac... 2 ms
googlelogo_color_272x92dp.png GET 200 quic/1+spdy/3 png ?gfe_rd=cr&ei=6X... (from cac... 3 ms
i1_1967ca6a.png GET 200 quic/1+spdy/3 png ?gfe_rd=cr&ei=6X... (from cac... 5 ms
photo.jpg GET 200 quic/1+spdy/3 png ?gfe_rd=cr&ei=6X... (from cac... 4 ms
data:image/gif;base... GET 200 data gif ?gfe_rd=cr&ei=6X... (from cac... 0 ms
data:image/gif;base... GET 200 data gif ?gfe_rd=cr&ei=6X... (from cac... 0 ms
25 requests | 64.3 KB transferred | Finish: 7.34 s | DOMContentLoaded: 536 ms | Load: 535 ms

Debugging (Cont.)

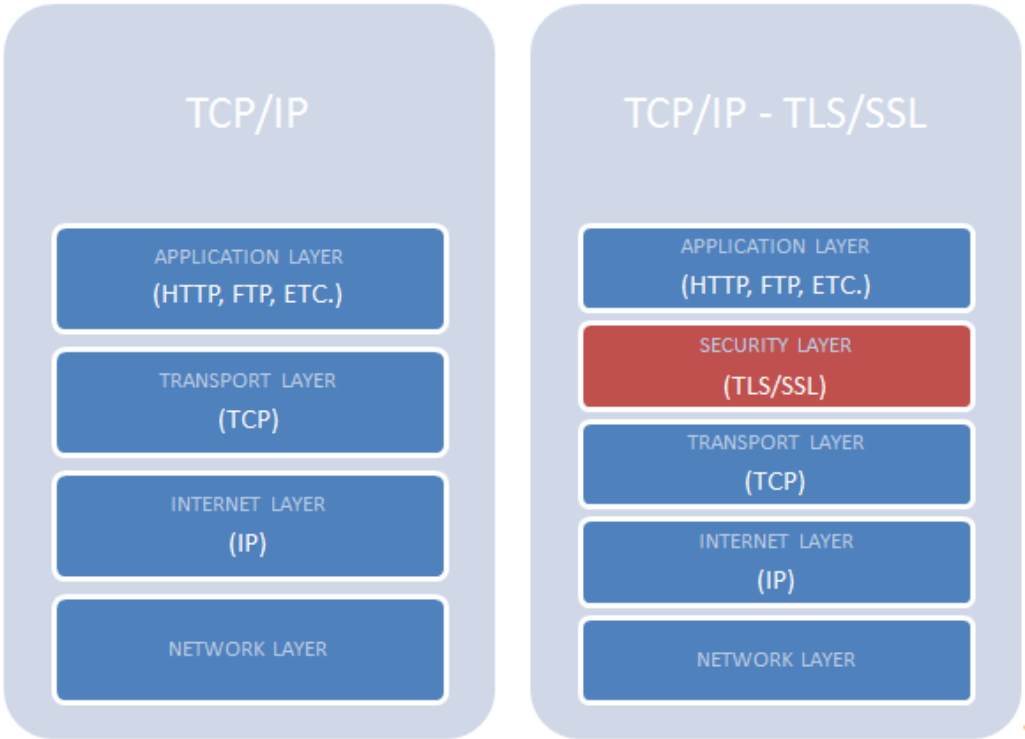


Debugging (Cont.)



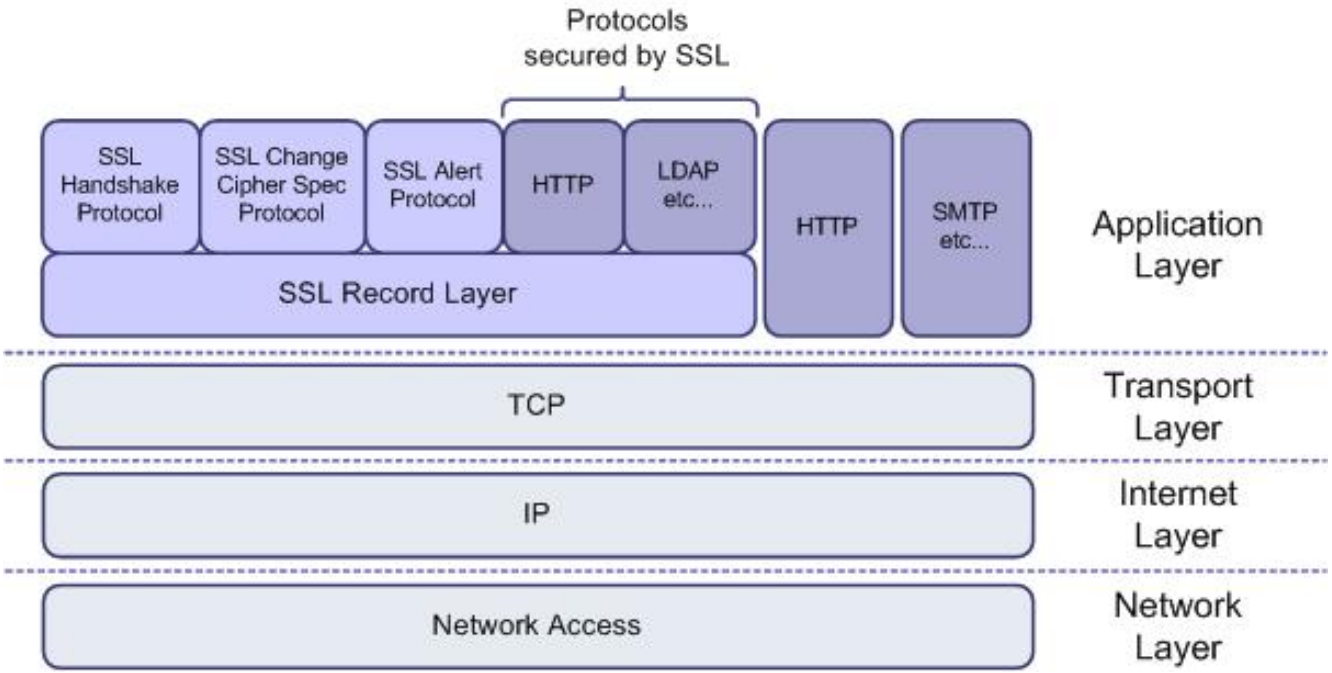
Host	Proxy	ID	Protocol Negotiated	Active streams	Unclaimed pushed	Max	Initiated	Pushed	Pushed and claimed	Abandoned	Received frames	Secure	Sent settings	Received settings
accounts.google.com:443	direct://	231	h2	0	0	100	0	0	0	0	0	true	true	true
s2.googleusercontent.com:443	direct://	370	h2	0	0	100	0	0	0	0	0	true	true	true
ssl.google-analytics.com:443	direct://	265	h2	0	0	100	0	0	0	0	0	true	true	true
ssl.gstatic.com:443	direct://	450	h2	0	0	100	0	0	0	0	0	true	true	true
www.google-analytics.com:443	direct://	339	h2	0	0	100	0	0	0	0	0	true	true	true
www.google.com:443	direct://	516	h2	0	0	100	0	0	0	0	0	true	true	true
www.google.com.ar:443	direct://	335	h2	0	0	100	0	0	0	0	0	true	true	true
accounts.google.com:443	direct://	328	h2	0	0	100	0	0	0	0	0	true	true	true
clients2.google.com:443	direct://	654	h2	0	0	100	0	0	0	0	0	true	true	true
www.googleapis.com:443	direct://	120	h2	0	0	100	0	0	0	0	0	true	true	true

SSL/TLS



fuente: https://www.simple-talk.com/dotnet/net-framework/tlssl-and-net-framework-4-0/

SSL/TLS



fuente: http://nicolascormier.com/documentation/bin/apache/apache2_with_ssl_tls/part1.htm

[HTTP/2] https://http2.github.io/.

[Ilya Grigorik] HTTP/2 is here, let's optimze!
https://docs.google.com/presentation/d/1r7QXGYOLCh4fcUq0jDdDwKJWNqWK1o4xMtYpKZCJYjM/present?slide=id.p19.