

2025 - 2026



Protocollo HTTP



Sul libro...

- Unità 1 - Lezione 3
- Da pag. 22 a pag. 32





Protocollo HTTP

- HTTP (*HyperText Transfer Protocol*) è il protocollo fondamentale del web.
- Sviluppato da Tim Berners-Lee nel 1989-1991



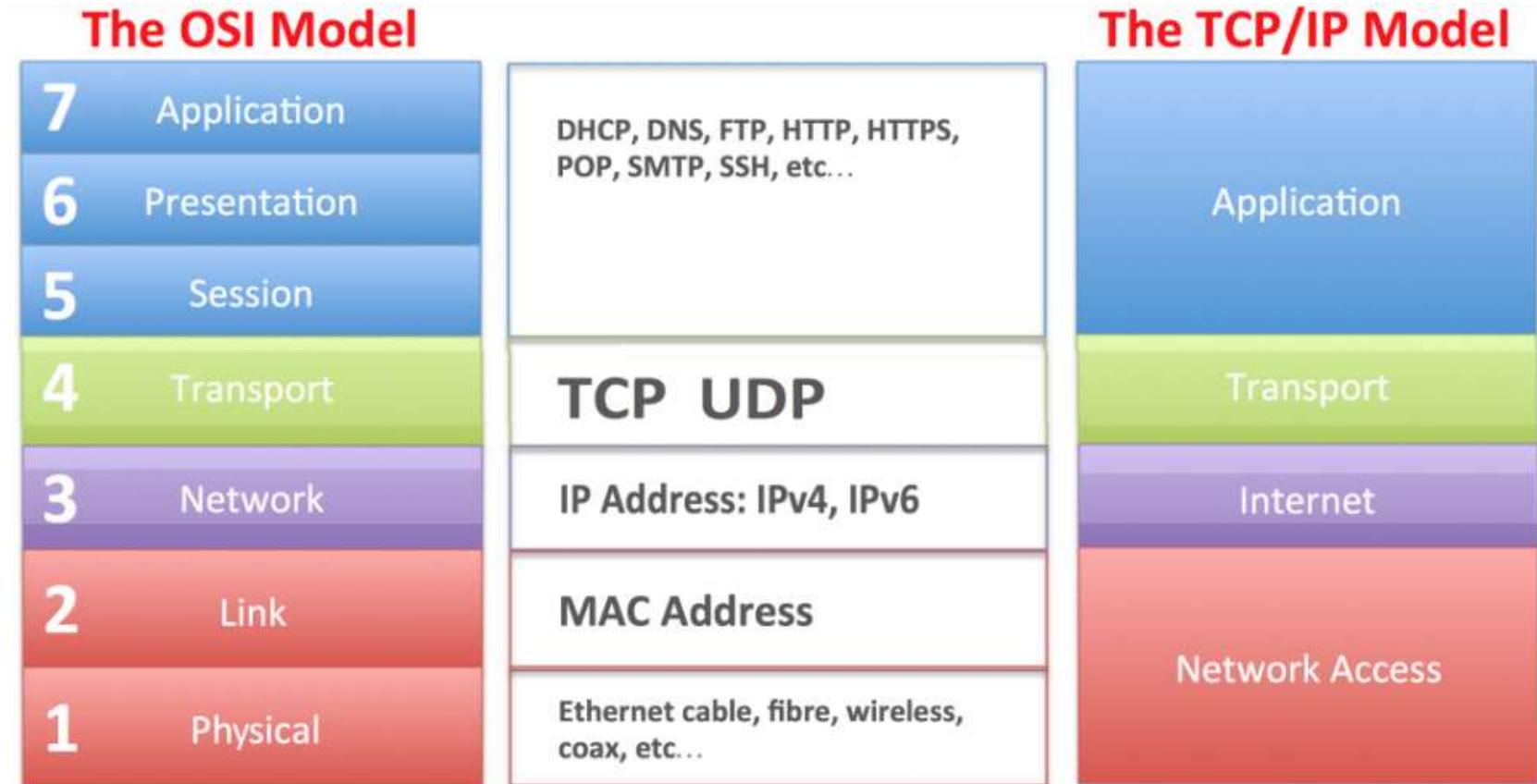
RFC 2610





Protocollo HTTP

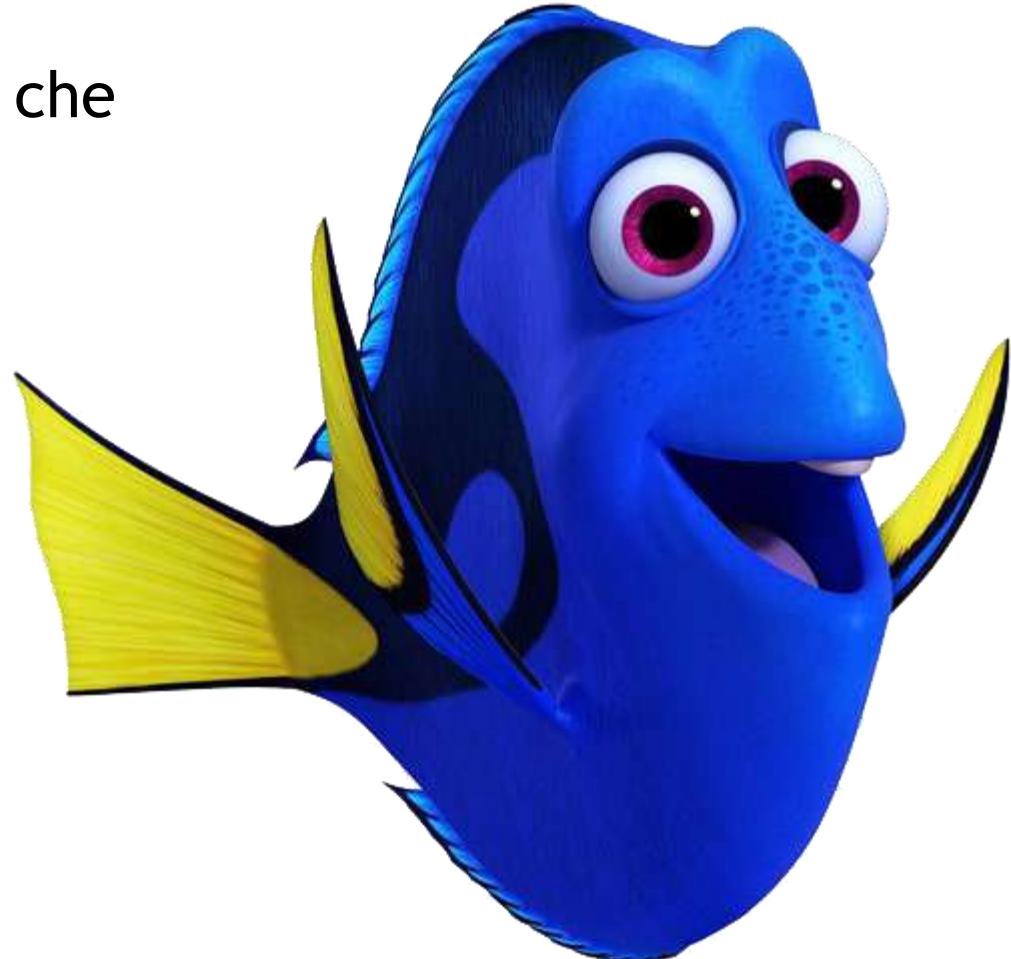
- HTTP è un protocollo **testuale** (utilizza caratteri ASCII standard) di **livello applicativo**





Protocollo HTTP

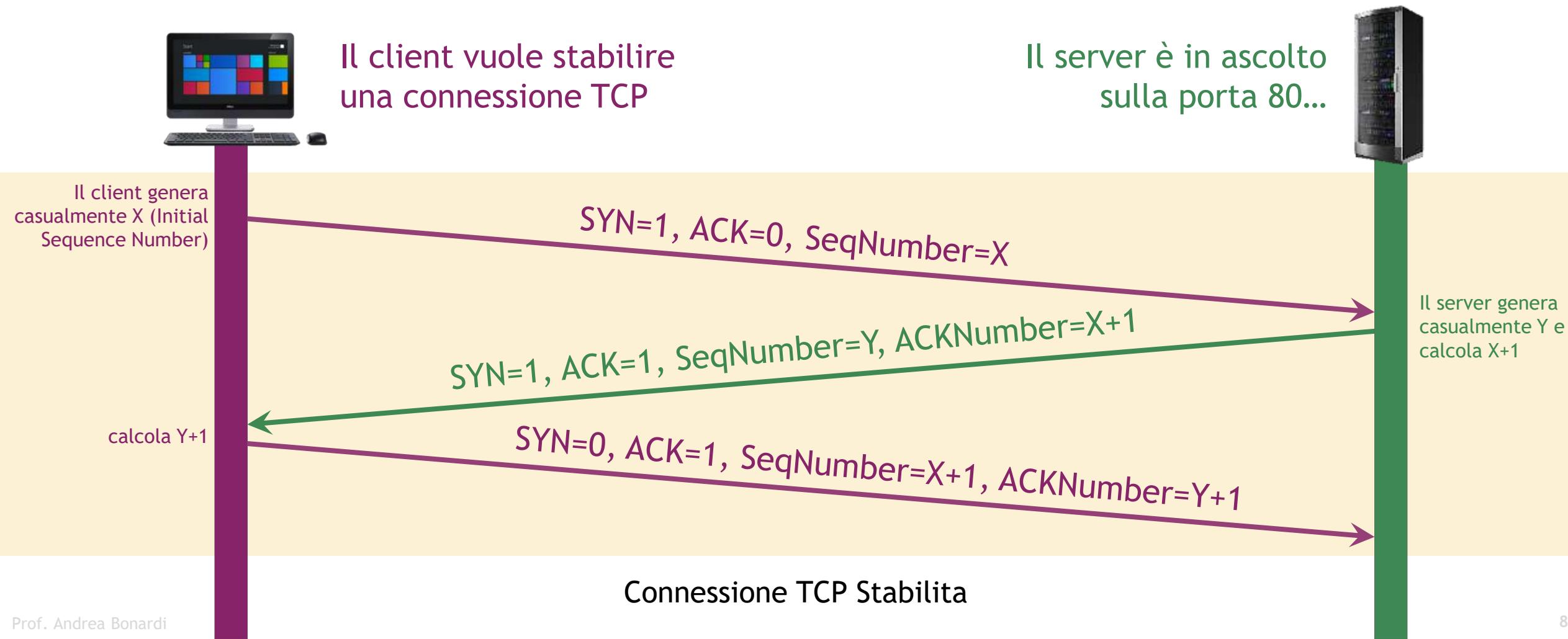
- HTTP è un protocollo **stateless** ("senza memoria"): ogni richiesta è indipendente
 - Una richiesta HTTP non sa nulla di quelle che l'hanno preceduta.





Protocollo HTTP

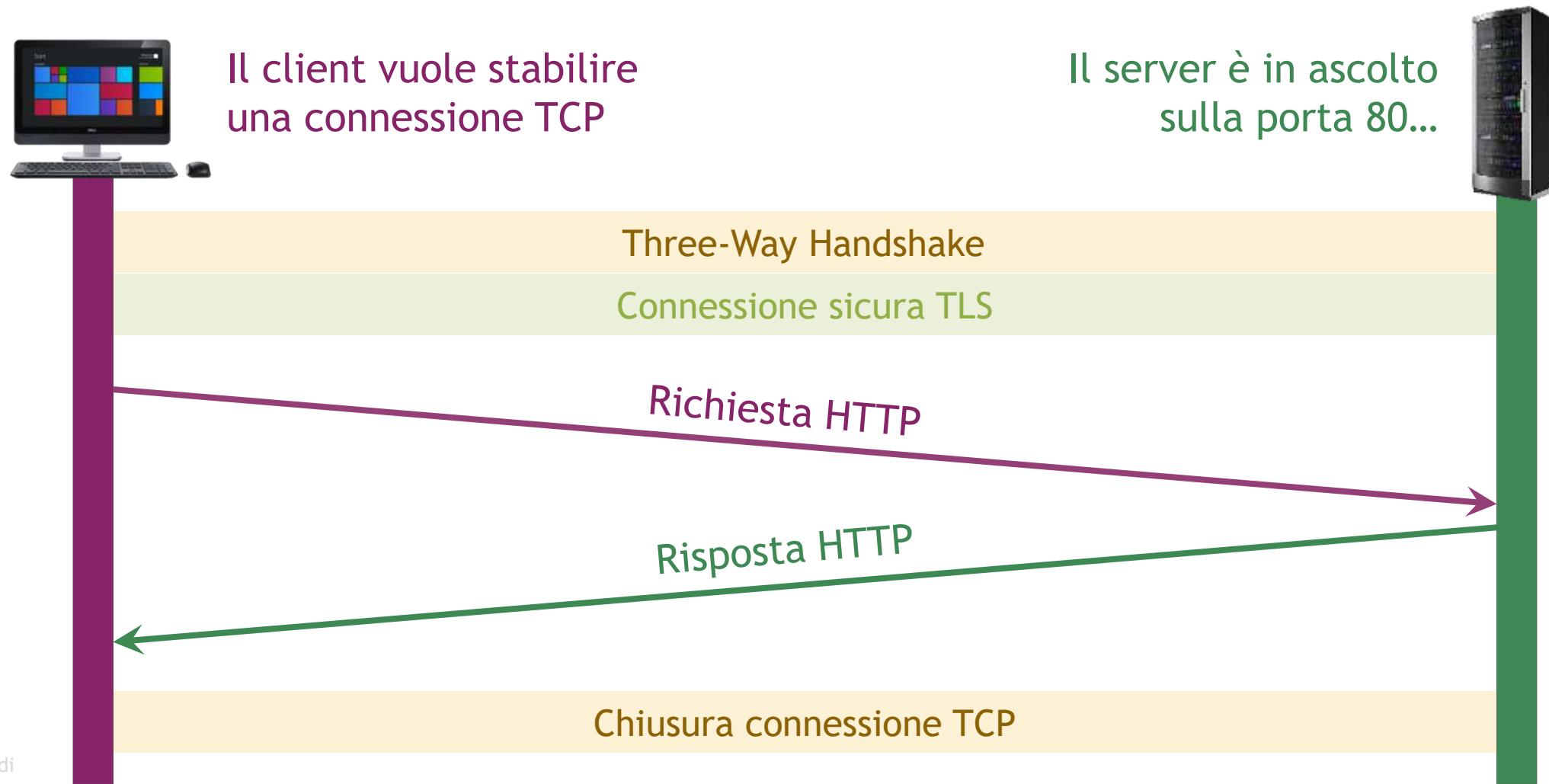
- HTTP/1 e 2 per comunicare usano **sessioni basate su TCP**

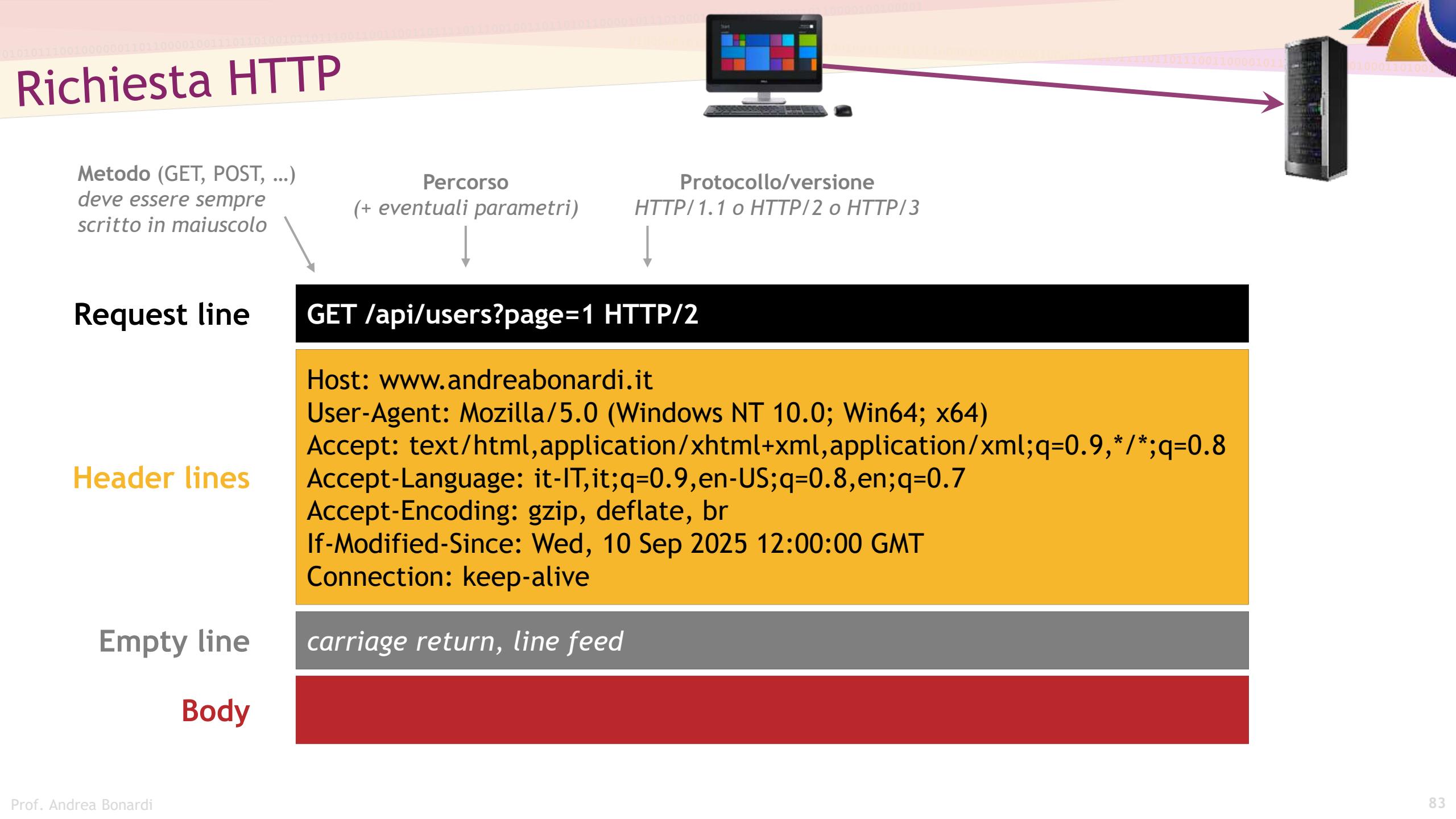




Protocollo HTTP

- HTTP/1 e 2 per comunicare usano **sessioni basate su TCP**







Metodi HTTP

GET	È usato quando il client vuole scaricare un documento dal server. Il documento richiesto è specificato nell'URL. Il server normalmente risponde con il documento richiesto nel corpo del messaggio di risposta.
HEAD	È usato quando il client non vuole scaricare il documento ma solo alcune informazioni sul documento (come ad esempio la data dell'ultima modifica). Nella risposta il server non inserisce il documento ma solo degli header informativi.
POST	È usato per fornire degli input al server da utilizzare per un particolare oggetto (di solito un applicativo) identificato nell'URL.
PUT	È utilizzato per memorizzare un documento nel server. Il documento viene fornito nel corpo del messaggio e la posizione di memorizzazione nell'URL.
DELETE	cancella il documento specificato nella URL



GET vs POST

Caratteristica	GET	POST
Nel form HTML	<form method="get"> (<i>default</i>)	<form method="post">
Visibilità dati inviati	Nella prima riga della richiesta HTTP, in coda all'URL (?key=val)	Nel body della richiesta HTTP
Lunghezza dei dati	Limitata (dipende dal browser, ~2000 caratteri)	Illimitata
Sicurezza	Meno sicuro (dati visibili nell'URL)	Più sicuro (dati nel body)
Tipi di dati inviati	Solo caratteri ASCII	Qualsiasi formato (testo, video, immagini)
Cache	Le risposte possono essere memorizzate dal browser	Di norma non vengono memorizzate
Bookmark	L'URL con i dati può essere salvato nei segnalibri	Non è possibile salvare l'URL con i dati
Ripetizione invio (back/refresh)	Sicura da ripetere (es. ricarica pagina)	Mostra avviso di conferma. Può causare duplicazione di dati



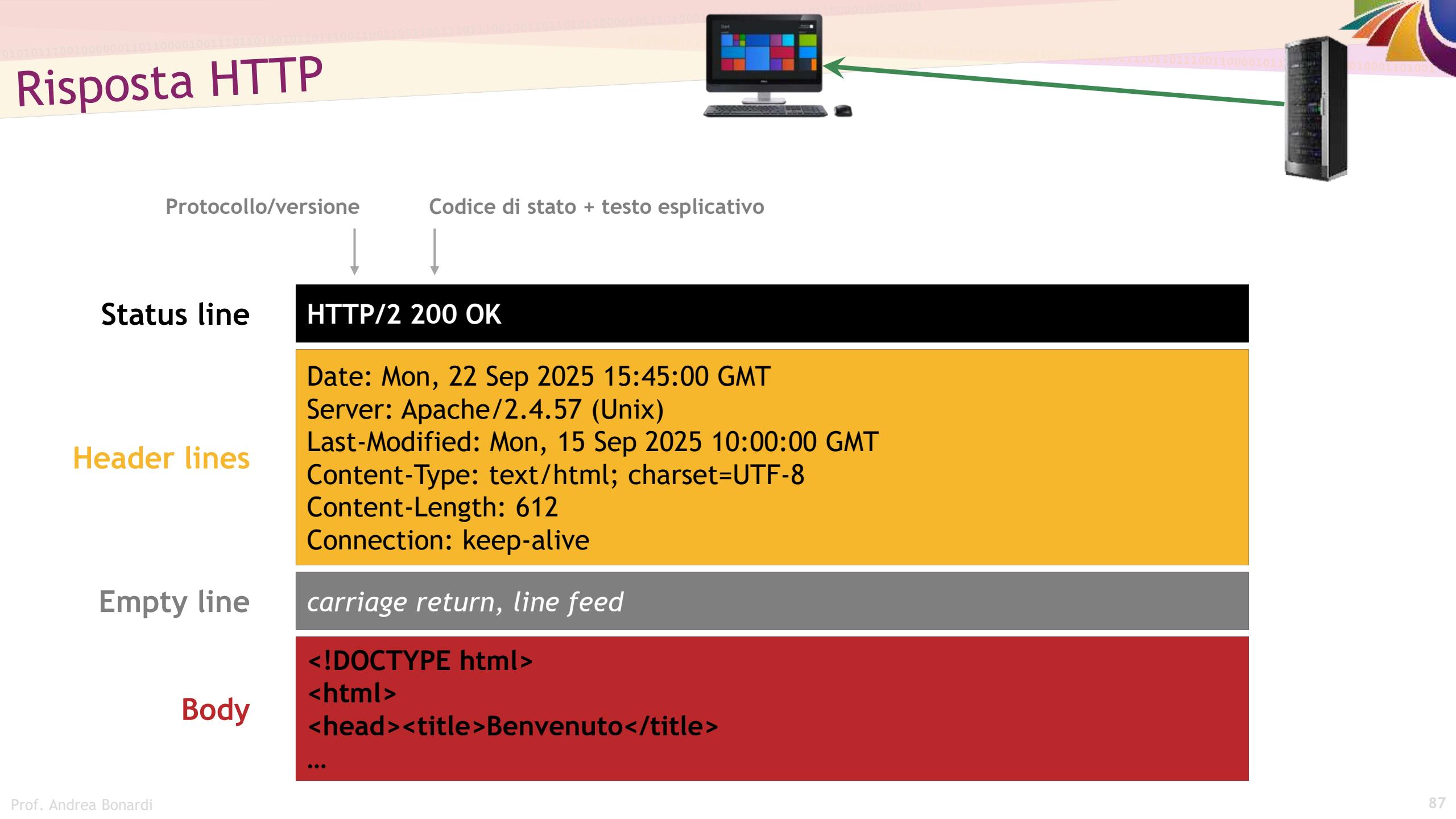
GET vs POST

- Usa GET quando:
 - Recuperi dati senza modificare lo stato del server
 - Vuoi che l'URL sia condivisibile
 - L'operazione è sicura e ripetibile
- Usa POST quando:
 - Invi dati sensibili (password, dati personali)
 - Modifichi dati sul server (creazione, aggiornamento)
 - L'operazione non è idempotente
(ogni invio produce risultati diversi)
 - Carichi file



The image shows a simplified login form. At the top is a red bar with the word "LOGIN" in white capital letters. Below it is a light gray area containing two input fields: one for "Username" with a user icon and one for "Password" with a lock icon. At the bottom is a dark gray button labeled "SIGN IN".





Risposta HTTP

Protocollo/versione

Codice di stato + testo esplicativo

Status line

HTTP/2 200 OK

Header lines

Date: Mon, 22 Sep 2025 15:45:00 GMT
Server: Apache/2.4.57 (Unix)
Last-Modified: Mon, 15 Sep 2025 10:00:00 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 612
Connection: keep-alive

Empty line

carriage return, line feed

Body

<!DOCTYPE html>
<html>
<head><title>Benvenuto</title>
...



Codici di stato

➤ Codici di stato HTTP restituiti dai server:

Intervallo codici	Categoria	Significato
100 - 199	Informational	Risposte informative temporanee alla richiesta (oggi sconsigliate dall'HTTP 1.0 in poi)
200 - 299	Successful	La richiesta è stata completata con successo
300 - 399	Redirection	Richiede al browser di eseguire un'azione aggiuntiva (es. seguire un nuovo URL)
400 - 499	Client Error	Errore dovuto al client (richiesta errata, risorsa non trovata, accesso negato...)
500 - 599	Server Error	Errore dovuto al server (malfunzionamenti, overload, configurazioni errate...)



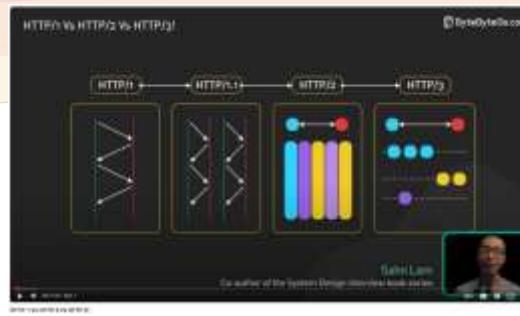
Codici di stato

- Esempi di codici:
 - **200 OK:** la richiesta di GET o POST ha avuto esito positivo
 - **301 Moved permanently:** la richiesta dovrà essere diretta a un altro URI
 - **400 Bad Request:** la richiesta non può essere soddisfatta a causa di errori di sintassi.
 - **404 Not Found:** la risorsa richiesta non è stata trovata
 - **500 Internal Server Error:** messaggio di errore generico

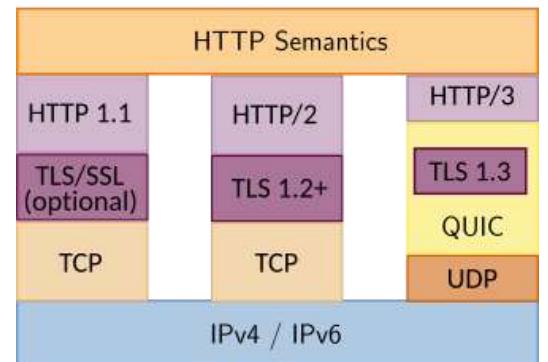


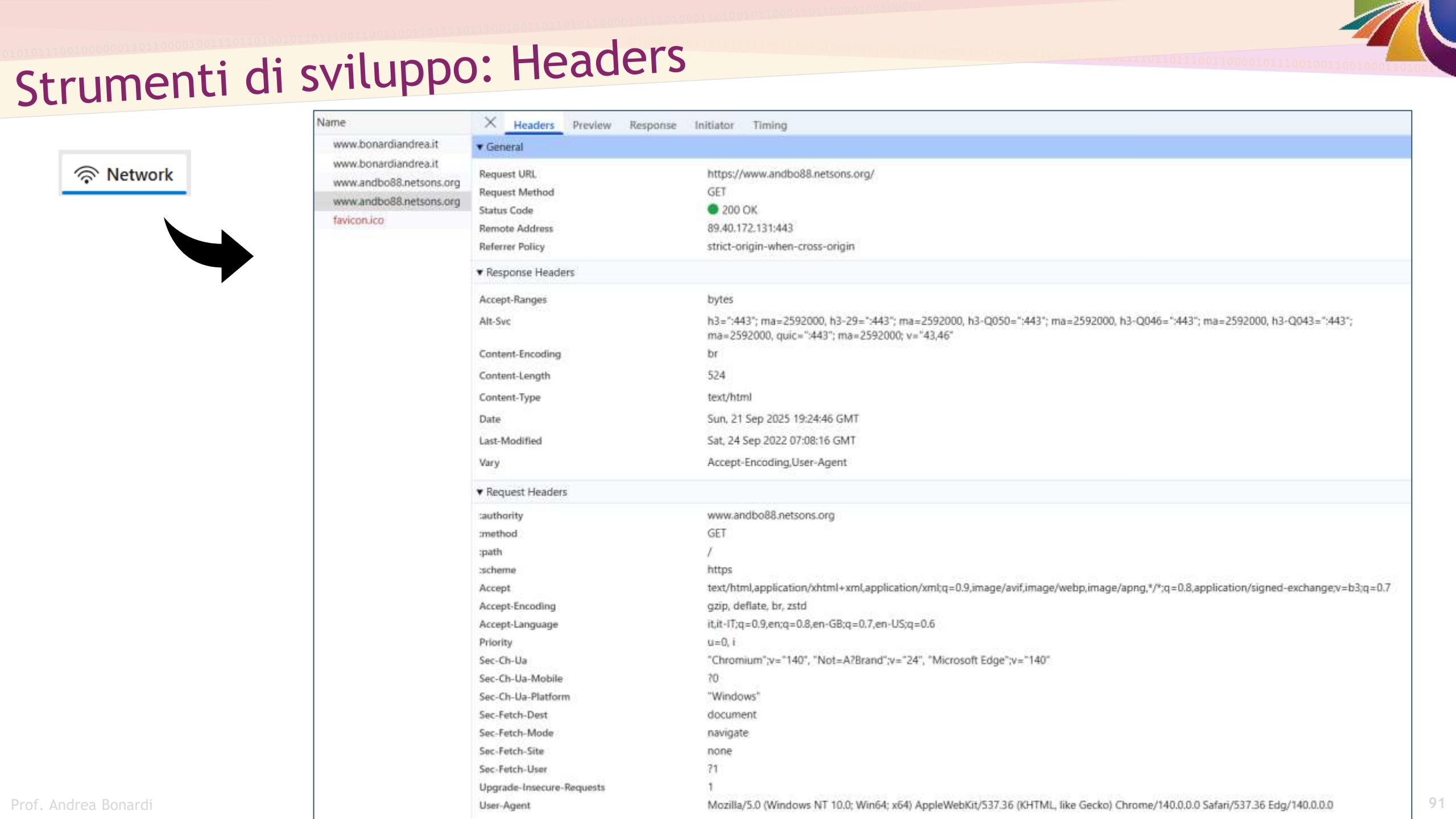
Protocollo HTTP: versioni

- 1996: HTTP/1.0
 - Una connessione TCP separata per ogni richiesta/risorsa.
- 1999: HTTP/1.1
 - Connessioni TCP persistenti e riutilizzabili per più richieste.
- 2015: HTTP/2
 - Invio e ricezione parallela di più richieste/risposte su una singola connessione.
 - Compressione degli header
- 2022: HTTP/3
 - Basato sul protocollo QUIC: sostituisce TCP con UDP



<https://www.youtube.com/watch?v=UMwQjFzTQXw>
<https://www.youtube.com/watch?v=ocGtt0IX0Js>





Strumenti di sviluppo: Headers

Network

Name	Headers	Preview	Response	Initiator	Timing
www.bonardiandrea.it					
www.bonardiandrea.it					
www.andbo88.netsons.org					
www.andbo88.netsons.org					
favicon.ico					
	General				
	Request URL	https://www.andbo88.netsons.org/			
	Request Method	GET			
	Status Code	200 OK			
	Remote Address	89.40.172.131:443			
	Referrer Policy	strict-origin-when-cross-origin			
	Response Headers				
	Accept-Ranges	bytes			
	Alt-Svc	h3=":443"; ma=2592000, h3-29=":443"; ma=2592000, h3-Q050=":443"; ma=2592000, h3-Q046=":443"; ma=2592000, h3-Q043=":443"; ma=2592000, quic=":443"; ma=2592000; v="43,46"			
	Content-Encoding	br			
	Content-Length	524			
	Content-Type	text/html			
	Date	Sun, 21 Sep 2025 19:24:46 GMT			
	Last-Modified	Sat, 24 Sep 2022 07:08:16 GMT			
	Vary	Accept-Encoding,User-Agent			
	Request Headers				
	:authority	www.andbo88.netsons.org			
	:method	GET			
	:path	/			
	:scheme	https			
	Accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7			
	Accept-Encoding	gzip, deflate, br, zstd			
	Accept-Language	it,it-IT;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6			
	Priority	u=0, i			
	Sec-Ch-Ua	"Chromium";v="140", "Not=A?Brand";v="24", "Microsoft Edge";v="140"			
	Sec-Ch-Ua-Mobile	?0			
	Sec-Ch-Ua-Platform	"Windows"			
	Sec-Fetch-Dest	document			
	Sec-Fetch-Mode	navigate			
	Sec-Fetch-Site	none			
	Sec-Fetch-User	?1			
	Upgrade-Insecure-Requests	1			
	User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36 Edg/140.0.0.0			

Prof. Andrea Bonardi

91



Strumenti di sviluppo: Headers

General	
Request URL	https://www.andbo88.netsons.org/
Request Method	GET
Status Code	200 OK
Remote Address	89.40.172.131:443
Referrer Policy	strict-origin-when-cross-origin

- L'URL richiesto è in HTTPS, quindi la connessione è cifrata
- Il metodo HTTP usato è GET
- Il server ha risposto correttamente con la risorsa (200 OK)
- Il server si trova all'indirizzo IP 89.40.172.131 e ha risposto usando la porta 443
- La Referrer Policy si riferisce a cosa il server invierà verso altri domini (cross origin) nelle richieste successive: solo l'origine, non il percorso né le query string



Strumenti di sviluppo: Headers

▼ Request Headers	
:authority	www.andbo88.netsons.org
:method	GET
:path	/
:scheme	https
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding	gzip, deflate, br, zstd
Accept-Language	it,it-IT;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Priority	u=0, i
Sec-Ch-Ua	"Chromium";v="140", "Not=A?Brand";v="24", "Microsoft Edge";v="140"
Sec-Ch-Ua-Mobile	?0
Sec-Ch-Ua-Platform	"Windows"
Sec-Fetch-Dest	document
Sec-Fetch-Mode	navigate
Sec-Fetch-Site	none
Sec-Fetch-User	?1
Upgrade-Insecure-Requests	1
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36 Edg/140.0.0.0

Tipi di contenuti che il browser accetta (HTML, immagini, formati compressi, ecc.)

Compressioni supportate dal browser

Chiede al server di fare upgrade automatico a HTTPS se possibile

Altre informazioni su client: browser, versione, piattaforma, mobile/desktop

Strumenti di sviluppo: Headers

Il server supporta richieste parziali
(utile per riprendere download interrotti o lo streaming)

Response Headers	
Accept-Ranges	bytes
Alt-Svc	h3=":443"; ma=2592000, h3-29=":443"; ma=2592000, h3-Q050=":443"; ma=2592000, h3-Q046=":443"; ma=2592000, h3-Q043=":443"; ma=2592000, quic=":443"; ma=2592000; v="43,46"
Content-Encoding	br
Content-Length	524
Content-Type	text/html
Date	Sun, 21 Sep 2025 19:24:46 GMT
Last-Modified	Sat, 24 Sep 2022 07:08:16 GMT
Vary	Accept-Encoding,User-Agent

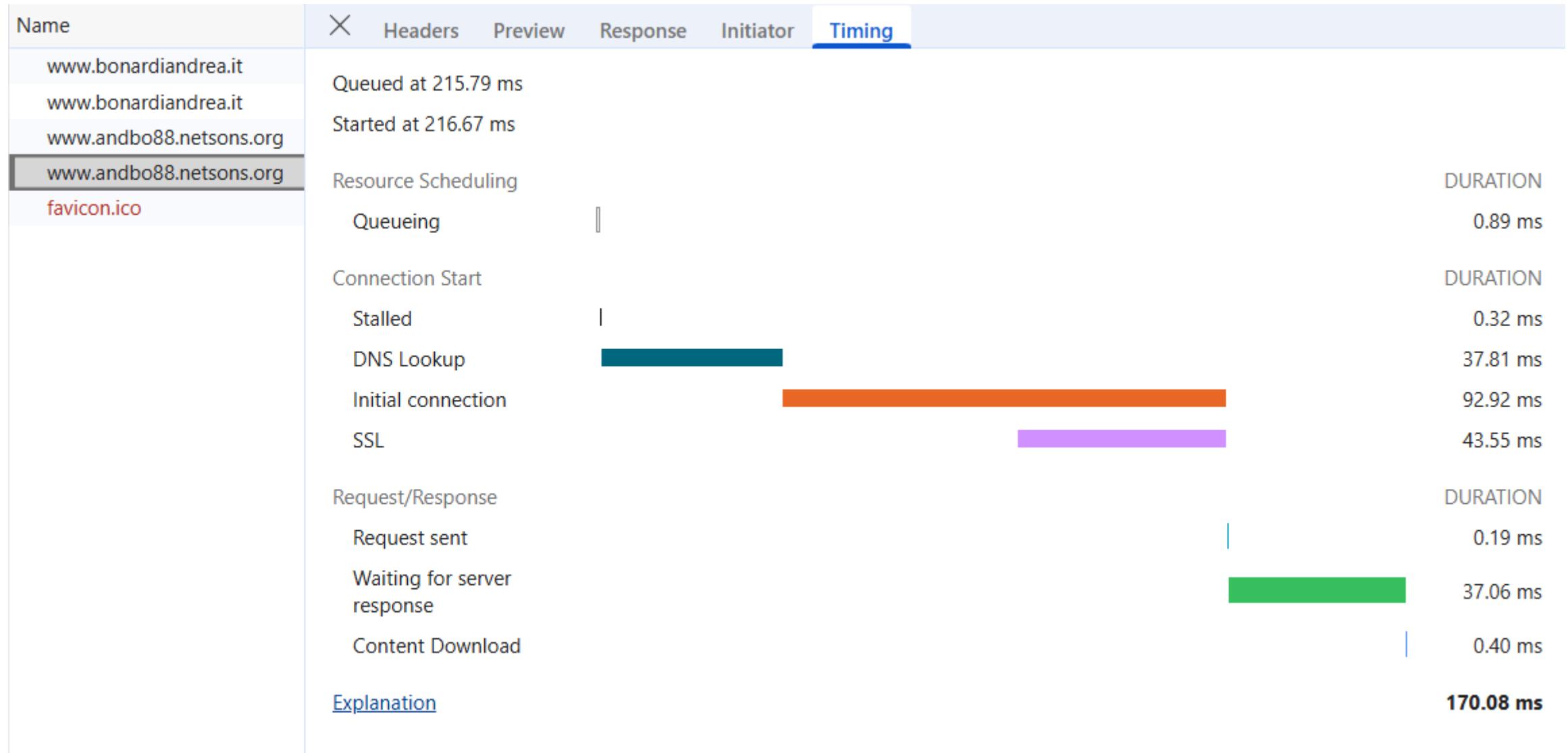
Il server segnala che supporta HTTP/3 (QUIC) sulla porta 443

Il server ha scelto l'algoritmo Brotli per comprimere le risposte
(algoritmo di compressione moderno, più efficiente di gzip)

Il server dichiara che le risposte possono cambiare a seconda della
compressione (Accept-Encoding) e del browser (User-Agent)



Strumenti di sviluppo: Timing





Error 418

- Documentazione ufficiale: [RFC2324](#)
- [Pagina Google](#)



418. I'm a teapot.

The requested entity body is short and stout.
Tip me over and pour me out.





Shopping advice



già in vostro
possesso...

