

HTTP 2.0

Revolucija web komunikacije ⚡



U ovoj lekciji:

- ⚡ Zašto je HTTP/1.1 problem?
- 🔄 Multipleksiranje & binarni protokol
- 📦 HPACK kompresija & Server Push
- 📊 Statistike koje te šokiraju

Sadržaj predavanja

01

Problemi HTTP/1.1

HOL blocking, spore glave, TCP slow start

02

SPDY → HTTP/2

Googleov odgovor, RFC 7540 (2015)

03

Ključne značajke

Binary, Multiplexing, HPACK, Server Push

04

Okviri i tokovi

10 vrsta okvira, streams, prioritizacija

05

Statistike & Demo

Performanse, adopcija, YouTube demo

01 Problemi HTTP/1.1

HTTP/1.1 je objavljen 1997.

1997

**Godina
HTTP/1.1**

Internet je tada bio
za 'turbo' dial-up

~100+

**Zahtjeva
po stranici**

Prosječna web stranica
2024. (vs ~10 u 1997.)

6×

**TCP konekcija
po domeni**

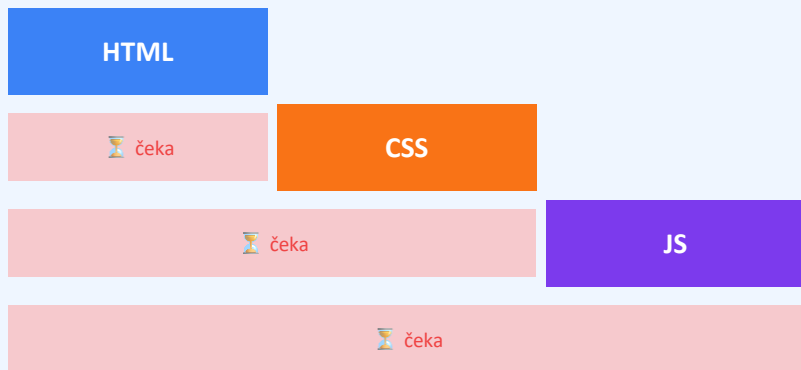
HTTP/1.1 workaround
za paralelne zahtjeve

💡 *Problem: web raste, ali protokol ostaje isti!*

01 Head-of-Line (HOL) Blocking

HTTP/1.1 = jedan zahtjev čeka drugi u redu. Kao blagajna s jednim blagajnikom! 🛒

HTTP/1.1 — Sekvencijalni zahtjevi



Vremenski slijed →

HTTP/2 — Multipleksiranje



Vremenski slijed →

🕒 HTTP/1.1 ukupno: ~800ms vs ⚡ HTTP/2 ukupno: ~200ms (4× brže!)

01 Problem: Opširna Zaglavlja (Verbose Headers)

HTTP/1.1 — Zaglavlje zahtjeva

```
GET /assets/logo.png HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 10...)
Accept: image/webp,image/apng,...
Accept-Language: hr-HR,hr;q=0.9,en
Accept-Encoding: gzip, deflate, br
Cookie: session=abc123; track=xyz
Cache-Control: no-cache
Connection: keep-alive
```

⚠ ~500 bajtova – SVAKI PUT iznova!



HTTP/2 + HPACK — Komprimirano zaglavlje

```
// Statična tablica – jednom poslano!
:method: GET
:path: /assets/logo.png
:authority: example.com
:scheme: https

// Duplikata nema! ✓
// Indeksi iz tablice: [idx:2, idx:5...]

✓ ~50 bajtova (smanjenje ~88%!)

```

01 Problem: TCP Slow Start & HOL na razini TCP-a

TCP Slow Start



Nova TCP konekcija kreće polako. Tek postupno povećava kapacitet — čak i ako je veza brza! HTTP/2 smanjuje broj konekcija, ali ih drži dulje.

Izgubljen paket?



Ako se izgubi paket, cijela TCP veza 'stane' i čeka retransmisiju. Ostali zahtjevi blokirani. HTTP/2 ne rješava ovo na TCP razini (→ HTTP/3!)

HTTP/1.1 Workaround



Preglednici otvaraju 6 TCP konekcija po domeni. Rješenje: domain sharding (razdvaja sadržaj na više poddomena). Ali: svaka konekcija = nova 'ruka', novi TLS handshake...

💡 HTTP/2 rješava HOL na razini aplikacijskog sloja — ali ne i na razini TCP-a. To rješava HTTP/3 (QUIC)!

02 Od SPDY-a do HTTP/2 – Kratka povijest

Zašto SPDY?

Google je 2009. primijetio da HTTP/1.1 ograničava performanse Gmaila, Searcha i Maps-a. Razvili su SPDY kao internu zamjenu. HTTP/2 standard je direktno preuzeo SPDY arhitekturu uz minimalne izmjene. Chrome je prestao podržavati SPDY krajem 2016. (zamijenjen HTTP/2-om).



03 HTTP/2 – 6 Ključnih Značajki



Binarni Protokol

Frames umjesto teksta



Multipleksiranje

Više streamova, 1 TCP



HPACK Kompresija

Zaglavlja ~88% manja



Server Push

Proaktivno slanje



Prioritizacija tokova

Važniji resursi prvi



TLS de facto

Svi preglednici traže

03 Binarni Protokol – Tekst vs Binarni Format

HTTP/1.1 — Tekstualni Format

```
GET /index.html HTTP/1.1
Host: www.example.com
Connection: keep-alive
Accept: text/html
```

```
// Čitljivo ljudima ✓
// Teže parserirati ✗
// Veće zaglavlje ✗
// Sporije ✗
```

✗ Human-readable, ali neučinkovito za strojeve



HTTP/2 — Binarni Okviri (Frames)

Duljina (24-bit)

Tip okvira (8-bit)

Oznake (8-bit)

Stream ID (31-bit)

Korisni sadržaj



Brže parsiranje, manje grešaka, manje podataka

03 Multipleksiranje – Više tokova, jedna konekcija

HTTP/1.1 — 6 zasebnih TCP konekcija

TCP #1 🚦 aktivan

TCP #2 🚦 aktivan

TCP #3 🚦 aktivan

TCP #4 ⌚ čeka

TCP #5 ⌚ čeka

TCP #6 ⌚ čeka

❌ 6× handshake, 6× slow-start, 6× memorija

HTTP/2 — 1 TCP konekcija, N streamova

🔗 Jedna TCP konekcija

Stream #1 → HTML ↔ Paralelno! ✓

Stream #3 → CSS ↔ Paralelno! ✓

Stream #5 → JS ↔ Paralelno! ✓

Stream #7 → IMG ↔ Paralelno! ✓

Stream #9 → Font ↔ Paralelno! ✓

03 Server Push — Proaktivno slanje resursa

HTTP/1.1 — Klijent mora sve tražiti

→ GET /index.html

← 200 OK + HTML

→ GET /style.css

← 200 OK + CSS

→ GET /app.js

← 200 OK + JS

🕒 **Ukupno: 6 round-trips za 3 resursa!**

HTTP/2 — Server šalje bez traženja!

→ GET /index.html

← 200 OK + HTML

← PUSH_PROMISE: style.css 📦 →bez čekanja!

← PUSH_PROMISE: app.js 📦 →bez čekanja!

← DATA: style.css ⚡

← DATA: app.js ⚡

⚡ **Ukupno: 1 round-trip za 3 resursa! (~60% brže)**

04 Uspostava HTTP/2 Konekcije

Metoda 1: HTTP Upgrade (h2c)

```
// Klijent šalje upgrade header
GET / HTTP/1.1
Host: example.com
Connection: Upgrade
Upgrade: h2c

// Odgovor poslužitelja
HTTP/1.1 101 Switch. Protocols
Upgrade: h2c

// Nastavak HTTP/2 komunikacijom
```

⚠ Samo za HTTP (bez TLS). Rijetko korišten u produkciji.

Metoda 2: ALPN (TLS Extension) ✓

- 1 TCP Three-way Handshake
- 2 TLS ClientHello → ALPN ext. [h2, http/1.1]
- 3 TLS ServerHello → odabran 'h2'
- 4 HTTP/2 Connection Preface (magic bytes)
- 5 SETTINGS okvir razmjenjen ✓

✓ De facto standard. Svi browseri zahtijevaju TLS za HTTP/2!

04 10 Vrsta HTTP/2 Okvira (Frames)

0x0

DATA

Prijenos podataka (tijelo zahtjeva/odgovora)

0x1

HEADERS

HTTP zaglavlja zahtjeva/odgovora

0x2

PRIORITY

Prioritet toka za raspored

0x3

RST_STREAM

Prekid toka s greškom

0x4

SETTINGS

Konfiguracija konekcije

0x5

PUSH_PROMISE

Najava server push resursa

0x6

PING

Mjerenje latencije / keepalive

0x7

GOAWAY

Uredno zatvaranje konekcije

0x8

WINDOW_UPDATE

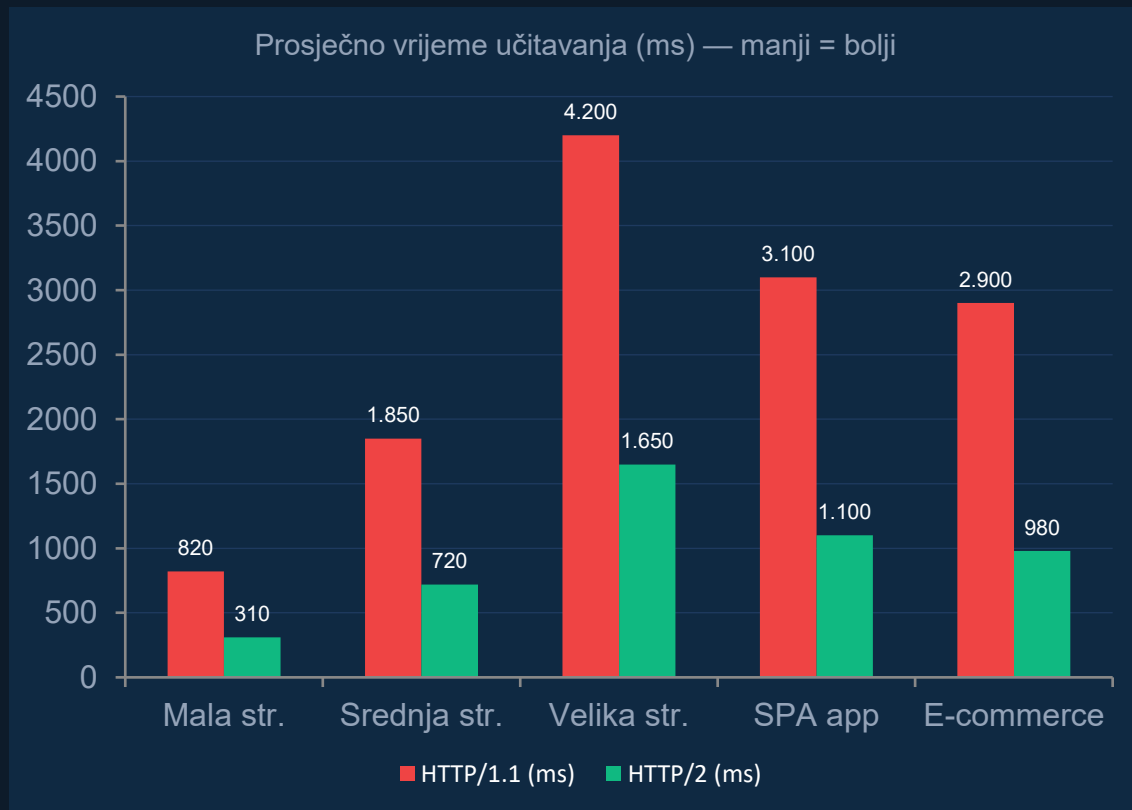
Kontrola protoka (flow control)

0x9

CONTINUATION

Nastavak HEADERS okvira (dulja zaglavlja)

05 Performanse — HTTP/1.1 vs HTTP/2



 **~60%**

Brže učitavanje

 **~88%**

Manja zaglavlja

 **1×**

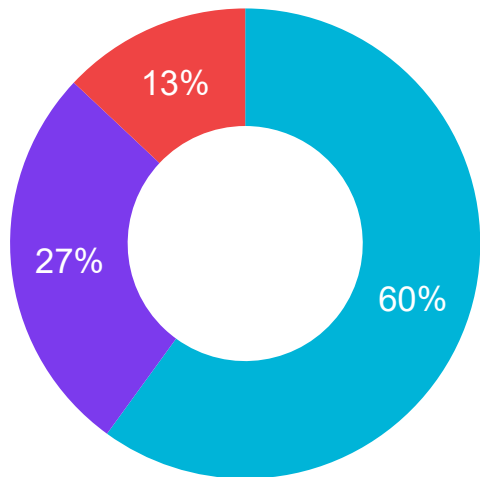
TCP konekcija
zamjesto 6×

 **~60%**

HTTP/2 weba

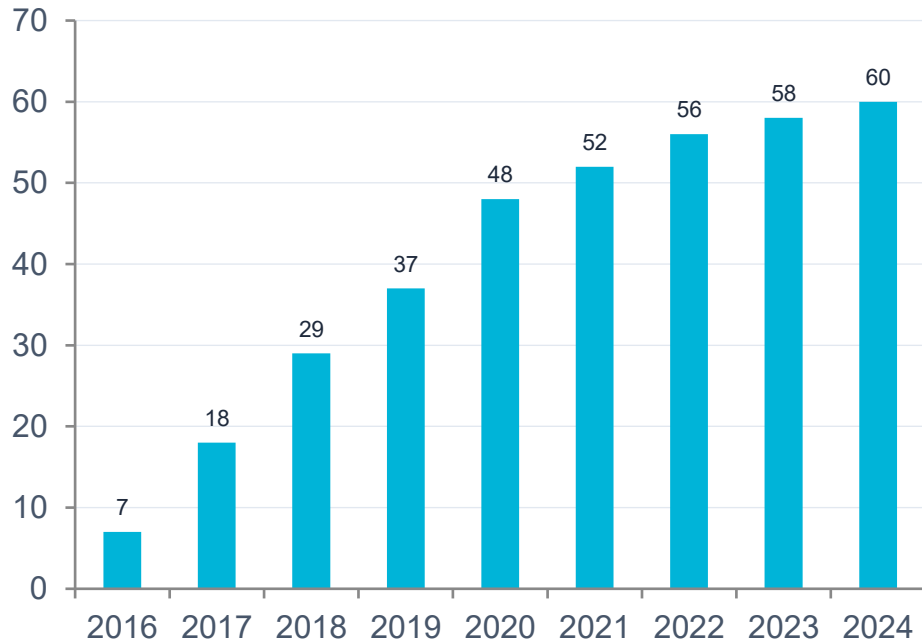
05 Globalna Adopcija HTTP/2 (2024)

Distribucija protokola (web stranice, 2024)



■ HTTP/2 ■ HTTP/3 ■ HTTP/1.1

Rast adopcije HTTP/2 (% web stranica)



Izvor: W3Techs, Can I Use, HTTP Archive (2024) • Gotovo svaki moderni CDN, Cloud provider i web server podržava HTTP/2 po defaultu

05 Podrška preglednika i poslužitelja



Chrome

od v41+

~65% korisnika



Firefox

od v36+

~3% korisnika



Safari

od v9+

~19% korisnika



Edge

od v12+

~5% korisnika



Samsung

od v4+

~3% korisnika



Opera

od v28+

~2% korisnika



Poslužitelji koji podržavaju HTTP/2:

nginx (1.9.5+) • Apache (2.4.17+) • Node.js (http2) • IIS (Win 10+) • Caddy • H2O • LiteSpeed • Tomcat 8.5+

05 Demo & Korisni Resursi



YouTube Videi — Obavezno pogledaj!



HTTP/2 vs HTTP/1.1 — Vizualna demonstracija — High Performance Browser Networking

 <https://www.youtube.com/watch?v=WfA1uDAgXf8>



HTTP/2 101 — Chrome Developers (Google I/O) — Google Chrome Developers

 https://www.youtube.com/watch?v=r5oT_2ndjms



HTTP/2 in 100 Seconds — Fireship — Fireship (preporučeno!)

 <https://www.youtube.com/watch?v=a-sBfyiXysl>



HTTP/1.1 vs HTTP/2 vs HTTP/3 — Hussain Nasser — Hussein Nasser

 <https://www.youtube.com/watch?v=0OrmKCB0UrQ>



Live Demo: <https://http2.akamai.com/demo> | <https://www.http2demo.io>

Nedostaci i Ograničenja HTTP/2



TLS u praksi obavezan

Svi preglednici podržavaju HTTP/2 SAMO preko HTTPS-a. Certifikat je neophodan (ali Let's Encrypt je besplatan!).



TCP HOL Blocking ostaje

HTTP/2 rješava HOL na aplikacijskom sloju, ali TCP HOL ostaje. Jedan izgubljen paket blokira sve streamove na konekciji.



Složenost implementacije

HTTP/2 je kompleksniji za debugiranje i analizu prometa. Binarni format nije čitljiv bez alata (Wireshark, Chrome DevTools).



Server Push zloupotrijebljen

Server Push se pokazao teško koristivim ispravno. Chrome je od 2022. ograničio podršku. Alternativa: 103 Early Hints.



Proxy/Firewall problemi

Stariji proxy serveri i vatrozidi ne razumiju HTTP/2 pa blokiraju konekcije. U enterprise okruženjima može biti problem.

HTTP/2 vs HTTP/3 — Što slijedi?

HTTP/3 (RFC 9114, 2022) zamjenjuje TCP s QUIC protokolom temeljenim na UDP-u 🚀

Svojstvo	HTTP/1.1	HTTP/2	HTTP/3 (QUIC)
Transport	TCP	TCP	UDP (QUIC)
Multiplexiranje	❌ Ne	✅ Da (TCP HOL)	✅ Da (bez HOL!)
HOL Blocking	❌ Aplikacijsko	⚠️ TCP ostaje	✅ Riješen!
Kompresija zaglavlja	❌ Nema	✅ HPACK	✅ QPACK
TLS	Opcijsko	De facto	Obavezno (TLS 1.3)
Uspostava konekcije	1-2 RTT	1-2 RTT	0-1 RTT (0-RTT!)
Adopcija (2024)	~13%	~60%	~27%

💡 HTTP/3 koriste: YouTube, Google, Facebook, Cloudflare, Instagram — 27% weba već 2024.!

Sažetak — Što smo naučili danas?

1

HTTP/1.1 (1997.) je spor jer zahtjevi čekaju u redu (HOL blocking), headeri se šalju svaki put iznova, a preglednici trebaju 6 TCP konekcija kao workaround.

2

HTTP/2 (2015.) uvodi binarni protokol, multipleksiranje (više streamova na 1 TCP), HPACK kompresiju zaglavlja (−88%) i Server Push.

3

HTTP/2 je ~60% brži od HTTP/1.1 u realnim scenarijima. Koristi ga ~60% weba (2024). Nginx, Apache, Node.js, IIS — svi podržavaju.

4

TCP HOL blocking ostaje problem → rješenje: HTTP/3 (QUIC, UDP). HTTP/3 već koriste YouTube, Google, Netflix i ~27% weba.

Pitanja?

