

HTTP/1.1 vs HTTP/2 vs HTTP/3

Bevezetés

Ez a gyakorlati jegyzet a HTTP/1.1, HTTP/2 és HTTP/3 protokollok működését, különbségeit és teljesítménybeli jellemzőit bemutató előadáshoz kapcsolódik.

Szükséges eszközök

- Google Chrome vagy Firefox böngésző
- Wireshark (legfrissebb verzió ajánlott)
- curl parancssori eszköz
- Internet kapcsolat

Megjegyzés: A gyakorlat Windows, Linux és macOS rendszeren is elvégezhető.

1. gyakorlati óra (45 perc) – HTTP/1.1 vs HTTP/2

1. feladat – HTTP verzió felismerése (curl és böngésző)

Cél: Megállapítani, hogy egy adott weboldal milyen HTTP protokollt használ.

1. Nyiss meg egy terminált/parancssort.
2. Add ki az alábbi parancsot:

```
curl.exe -I https://example.com
```

3. Figyeld meg a válasz első sorát (pl. HTTP/2 200 vagy HTTP/1.1 200).

4. Kényszerítsd a HTTP/1.1 használatát:

```
curl.exe --http1.1 -I https://example.com
```

5. Kényszerítsd a HTTP/2 használatát:

```
curl.exe --http2 -I https://example.com
```

Magyarázat:

A kliens (curl vagy böngésző) és a szerver közösen dönti el, melyik HTTP verziót használják. A HTTPS kapcsolat esetén ez az ALPN (Application-Layer Protocol Negotiation) során történik.

2. feladat – HTTP/1.1 Head-of-Line Blocking megfigyelése

Cél: A HTTP/1.1 soros működésének és a HOL blocking jelenségnek a megfigyelése.

1. Nyisd meg a Chrome DevTools-t (F12).
2. Network fül → Throttling: 'Slow 3G'.
3. Nyiss meg egy weboldalt.

Ez a website HTTP/1.1 protokollt használ: <https://mysupport.enghouse.com/andtek-manuals/>

4. Figyeld meg a Waterfall diagramot.

Megfigyelés:

- HTTP/1.1 esetén több erőforrás várakozik egymásra.
- Egy lassú válasz késlelteti a mögötte lévőket.

Magyarázat:

Ez a head-of-line blocking jelenség, amely a HTTP/1.1 egyik legnagyobb teljesítményproblémája.

3. feladat – HTTP/2 multiplexing vizsgálata Wiresharkkal

Cél: Megfigyelni, hogy a HTTP/2 hogyan használ egyetlen TCP kapcsolatot több kéréshez.

1. Indítsd el a Wiresharkot.
2. Válaszd ki az aktív hálózati interfészt.
3. Indíts rögzítést.
4. Nyiss meg egy HTTP/2-t használó oldalt (pl. <https://nghttp2.org/>).

`curl.exe -v --http2 https://nghttp2.org/`

5. Állítsd le a rögzítést.

Hasznos Wireshark szűrők:

```
http2  
tcp.port == 443
```

Megfigyelés:

- Egy TCP kapcsolat
- Több HTTP/2 stream (Stream ID-k)

Magyarázat:

A HTTP/2 multiplexing lehetővé teszi, hogy több kérés és válasz párhuzamosan haladjon egyetlen TCP kapcsolaton.

4. feladat – Header compression (HPACK) hatásának megfigyelése

Cél: A HTTP/2 header compression működésének megértése.

1. Wiresharkban válassz ki egy HTTP/1.1 kérést.
2. Nézd meg a teljes HTTP header tartalmát.
3. Ezután válassz ki egy HTTP/2 HEADERS frame-et.

Megfigyelés:

- A HTTP/2 header adatok jóval kisebbek.

Magyarázat:

A HPACK statikus és dinamikus táblákat használ a header mezők tömörítésére. Ez jelentősen csökkenti az átvitt adatmennyiséget.

2. gyakorlati óra (45 perc) – HTTP/3 és QUIC

5. feladat – HTTP/3 forgalom felismerése

Cél: HTTP/3 és QUIC forgalom azonosítása.

1. Wiresharkban indíts új rögzítést.
2. Nyiss meg egy HTTP/3-at támogató oldalt (pl. <https://cloudflare-quic.com>).
3. Állítsd le a rögzítést.

Wireshark szűrők:

```
quic  
udp.port == 443
```

Megfigyelés:

- UDP csomagok
- QUIC protokoll

Magyarázat:

A HTTP/3 nem TCP felett működik, hanem a QUIC protokollt használja UDP felett.

6. feladat – QUIC stream-ek megfigyelése

Cél: A QUIC stream-alapú működésének megértése.

1. Wiresharkban bonts ki egy QUIC csomagot.
2. Figyeld meg a stream azonosítókat.

Megfigyelés:

- Több független stream
- Egy stream problémája nem állítja meg a többit

Magyarázat:

Ez a legnagyobb különbség a HTTP/2 + TCP és a HTTP/3 + QUIC között: nincs transport-szintű head-of-line blocking.

7. feladat – HTTP/2 vs HTTP/3 összehasonlítás

Cél: A két modern protokoll közti különbségek összefoglalása.

Kérdések:

- Melyik használ TCP-t?
- Melyik működik UDP felett?
- Hol jelenik meg HOL blocking?
- Melyik alkalmasabb mobil hálózatra?