
Aplikacijski sloj II: DNS, e-pošta, P2P i socketi

Poglavlje 3 — Skripta kolegija *Računalne mreže*

doc. dr. sc. Sandi Baressi Šegota

Fakultet informatike Sveučilišta Jurja Dobrile u Puli

Poglavlje 3

Aplikacijski sloj II: DNS, e-pošta, P2P i sock- eti

Nakon weba i HTTP-a, u ovom poglavlju upoznajemo još nekoliko temeljnih aplikacijskih usluga: **DNS** koji imena prevodi u adrese, **elektroničku poštu**, distribuciju sadržaja **P2P** mrežama i **CDN**-ovima te **prilagodljivi streaming**. Na kraju upoznajemo **socket API** — programsko sučelje kojim aplikacije zapravo pristupaju mreži.

Ishodi učenja

Po završetku poglavlja student će moći: objasniti hijerarhiju i tijek razlučivanja DNS-a te ulogu predmemorije; opisati arhitekturu e-pošte i protokole SMTP, IMAP/POP3 i MIME; obrazložiti zašto se P2P distribucija bolje skalira; opisati prilagodljivi streaming (DASH) i ulogu CDN-a; te skicirati jednostavan UDP i TCP socket program.

3.1 DNS — sustav imena domena

Ljudi lakše pamte imena poput `www.fipu.hr` nego numeričke IP adrese, no mrežni sloj radi isključivo s adresama. **DNS** (*Domain Name System*) je distribuirana, hijerarhijska baza podataka i aplikacijski protokol koji ime hosta prevodi u IP adresu. Osim toga, DNS pruža *aliase* (kanonsko ime i nadimke), usmjeravanje e-pošte putem MX zapisa te *raspodjelu opterećenja* (za isto ime vraća više adresa).

3.1.1 Hijerarhija i vrste poslužitelja

DNS je organiziran hijerarhijski. Na vrhu su **korijenski** poslužitelji koji upućuju na poslužitelje **vršnih domena** (TLD, npr. `.hr`, `.com`). Ispod njih su **mjerodavni** (*authoritative*) poslužitelji koji čuvaju zapise konkretne organizacije. Uz njih djeluje i **lokalni** (resolving) poslužitelj, tipično kod ISP-a, koji obavlja upite u ime hosta i *predmemorira* odgovore.

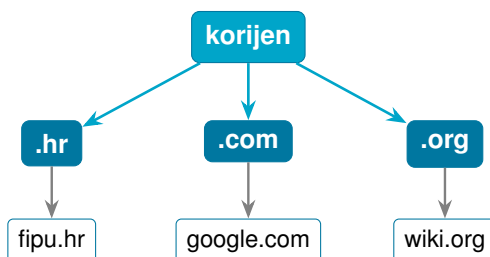


Figure 3.1: Hijerarhija DNS-a: korijen, TLD i mjerodavni poslužitelji.

3.1.2 Razlučivanje imena i predmemorija

Razlučivanje obično kombinira *rekurzivni* upit (host prepušta cijeli posao lokalnom poslužitelju) i niz *iterativnih* upita (lokalni poslužitelj redom pita korijen, TLD i mjerodavni poslužitelj). Dobiveni se odgovor vraća hostu i **predmemorira** uz vrijednost TTL. Predmemorija je ključna: drastično smanjuje promet prema korijenu i TLD-u te ubrzava razlučivanje popularnih imena; dulji TTL znači manje prometa, ali sporije širenje promjena.

Vrste zapisa

A (ime → IPv4), AAAA (ime → IPv6), NS (mjerodavni poslužitelj domene), CNAME (alias → kanonsko ime) i MX (poslužitelj e-pošte). DNS koristi UDP na portu 53, a za velike odgovore i zonske prijenose TCP.

3.1.3 Primjer razlučivanja i sigurnost

Pretpostavimo da preglednik traži `www.fipu.hr`. Host šalje rekurzivni upit lokalnom poslužitelju; ako odgovor nije u predmemoriji, lokalni poslužitelj pita korijenski poslužitelj koji vraća adresu `.hr` TLD poslužitelja, zatim TLD poslužitelj koji vraća adresu mjerodavnog poslužitelja domene `fipu.hr`, i konačno mjerodavni poslužitelj koji vraća traženu IP adresu. Odgovor putuje natrag i pohranjuje se u predmemoriju svih posrednika. U praksi je većina ovih koraka preskočena zahvaljujući predmemoriji.

DNS je i meta napada: kod *trovanja predmemorije* napadač pokušava ubaciti lažne zapise. Zaštitu pružaju **DNSSEC** (digitalni potpisi zapisa koji jamče integritet i autentičnost) te **DNS-over-HTTPS/TLS** (DoH/DoT), koji šifriraju same upite i tako čuvaju privatnost korisnika.

3.2 Elektronička pošta

Sustav e-pošte čine tri komponente: **MUA** (*mail user agent*, korisnički klijent), **MTA** (*mail transfer agent*, poslužitelji koji prenose poštu) i **poštanski sandučić** na poslužitelju primatelja. Za *slanje* pošte koristi se **SMTP**, protokol tipa *push* koji gura poruku prema poslužitelju (TCP, port 25); za *dohvat* pošte koriste se **IMAP** ili **POP3**.



Figure 3.2: Put poruke e-pošte od pošiljatelja do primatelja.

IMAP ostavlja poruke na poslužitelju, podržava mape i sinkronizaciju s više uređaja, dok **POP3** obično preuzima i briše poruke i jednostavniji je. Poruka ima zaglavlja (From, To, Subject, Date) i tijelo; standard **MIME** omogućuje ne-ASCII sadržaj i privitke pomoću zaglavlja Content-Type i Content-Transfer-Encoding. Sigurnost pošiljatelja podižu mehanizmi SPF, DKIM i DMARC, a prijenos se štiti TLS-om.

Sam SMTP je tekstualni protokol: nakon uspostave TCP veze poslužitelj se javlja kodom 220, klijent se predstavlja naredbom HELO (ili EHLO), zatim slijede MAIL FROM (pošiljatelj), RCPT TO (primatelj) i DATA nakon koje se šalje tijelo poruke završeno retkom s jednom točkom. Poslužitelj svaki korak potvrđuje kodom (npr. 250 OK), a veza se zatvara naredbom QUIT. Danas se pošti sve češće pristupa i preko *web-pošte* u pregledniku, uz IMAP u pozadini, čime su poruke i privitci dostupni s više uređaja.

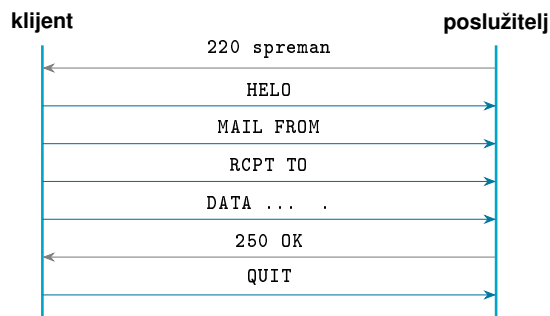


Figure 3.3: Tijek SMTP dijaloga pri predaji poruke.

3.3 P2P, streaming i CDN

Kod distribucije velikoj publici arhitektura *klijent–poslužitelj* pati jer je izvor usko grlo: vrijeme distribucije raste s brojem korisnika. U **P2P** pristupu svaki novi *peer* donosi i vlastiti kapacitet slanja, pa se sustav znatno bolje skalira.

Razliku je korisno promotriti kroz *vrijeme distribucije* datoteke velike populacije korisnika. Kod klijent–poslužitelja poslužitelj mora poslati svaku kopiju, pa potrebno vrijeme raste *linearно* s brojem korisnika — ograničava ga isključivo izlazna brzina poslužitelja. Kod P2P-a, čim korisnik primi dio datoteke, i sam ga počinje dijeliti; ukupni kapacitet slanja tako raste zajedno s brojem sudionika, pa vrijeme distribucije za veliku publiku raste mnogo sporije i ostaje ograničeno. Upravo zato P2P dobro odgovara scenarijima masovne distribucije istog sadržaja. **BitTorrent** dijeli datoteku na *dijelove* (chunks); *tracker* vodi popis *peerova*, a oni izravno razmjenjuju dijelove. Mehanizmi *rarest first* (prvo najrjeđi dijelovi), *tit-for-tat* (šalji onima koji šalju tebi) i *optimistic unchoke* čine razmjenu učinkovitom i poštenom.

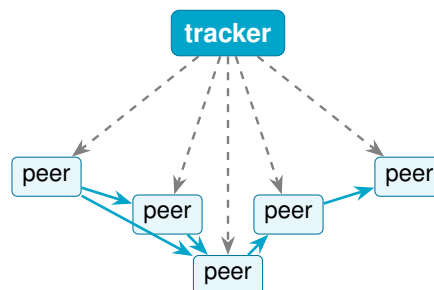


Figure 3.4: BitTorrent: tracker povezuje peerove koji razmjenjuju dijelove datoteke.

Za video se koristi **prilagodljivi streaming** (DASH): video je kodiran u više *kvaliteta* i podijeljen na *segmente*, a *manifest* (MPD) opisuje dostupne razine. Klijent (player) tijekom reprodukcije bira kvalitetu prema *trenutnoj propusnosti*, čime održava neprekidnu reprodukciju. **CDN** (*content delivery network*) drži kopije sadržaja na *rubnim* poslužiteljima blizu korisnika; korisnik se — najčešće preko DNS-a — usmjerava na najbliži edge, čime se smanjuje kašnjenje i rasterećuje izvor.

Odabir edge poslužitelja obično se izvodi tako da mjerodavni DNS CDN-a, na temelju adrese lokalnog DNS poslužitelja korisnika, procijeni njegovu geografsku i mrežnu blizinu te vrati adresu prikladnog edge-a; uzimaju se u obzir i trenutno opterećenje te dostupnost. CDN-ovi se razlikuju i po strategiji punjenja: kod *pull* pristupa edge dohvaća sadržaj s izvora tek pri prvom zahtjevu (i potom ga predmemorira), dok se kod *push* pristupa sadržaj unaprijed raspoređuje na rubne poslužitelje. Time se postiže ravnoteža između svježine sadržaja i učinkovitosti isporuke.

3.4 Socket programiranje

Socket je programsko sučelje između aplikacije i transportnog sloja. Razlikujemo **UDP** sockete (bez veze) i **TCP** sockete (s vezom). Kod UDP-a svaka poruka nosi određenu adresu i nema jamstva dostave; kod TCP-a se prije prijenosa uspostavlja veza, a poslužitelj koristi *welcome* socket za primanje novih veza i po jedan *connection* socket za svakoga klijenta.

Skica (Python, pojednostavljeno)

UDP poslužitelj: `socket(AF_INET, SOCK_DGRAM), bind(...)`, u petlji `recvfrom()` pa `sendto()`.

TCP poslužitelj: `socket(AF_INET, SOCK_STREAM), bind(...), listen()`, u petlji `accept()` → `recv()/send()` → `close()`.

Poslužitelj u praksi mora posluživati *više klijenata istovremeno*, bilo pomoću dretvi/procesa po vezi, bilo asinkronim (event-driven) modelom.

UDP socket	TCP socket
bez uspostave veze	uspostava veze prije prijenosa
<code>sendto / recvfrom</code>	<code>connect / accept, send / recv</code>
određena adresa uz svaku poruku	adresa jednom, pri spajanju
brz, bez jamstva dostave	pouzdan tok bajtova

Table 3.1: Usporedba UDP i TCP socketa.

Sažetak poglavlja

DNS hijerarhijski i uz predmemoriju prevodi imena u adrese. **E-pošta** koristi SMTP za slanje te IMAP/POP3 za dohvat, uz MIME za sadržaj. **P2P** (BitTorrent) skalira distribuciju jer resursi rastu s brojem peerova, dok **DASH** i **CDN** omogućuju kvalitetan streaming velikoj publici. **Socket API** (UDP i TCP) temelj je izgradnje mrežnih aplikacija.