

2. Preza - Procesi i komunikacija: model klijent - poslužitelj

Obilježja komunikacije:

- konekcijska - prvo se stvori konekcija pa se onda vrši komunikacija
- beskonkcijska - ne stvara se konekcija već se direktno šalju poruke
- perzistentna - osigurava se da poruka stigne na određeno mjesto tako da se prva poruka spremi u BP i onda od tamo šalje, a ako ne uspije, šalje se opet
- tranzijentna - nema nikakvog osiguranja, poruka se prenese samo ako su i posiljalatelj i primatelj online
- sinkrona - posiljalatelj blokiran dok ne dobije potvrdu o primitku poruke sa primatelja
- asinkrona - posiljalatelj nije blokiran

Obilježja procesa:

- procesi su mali programi koji se odvijaju svaki na nekoj dretvi i zato ih se može istovremeno odvijati više
- vremenski ovisni procesi - moraju biti istovremeno aktivni za komunikaciju
- vremenski neovisni procesi - ne moraju biti istovremeno aktivni
- (ne)ovisnost o referenci sugovornika

Komunikacija među raspodijeljenim procesima:

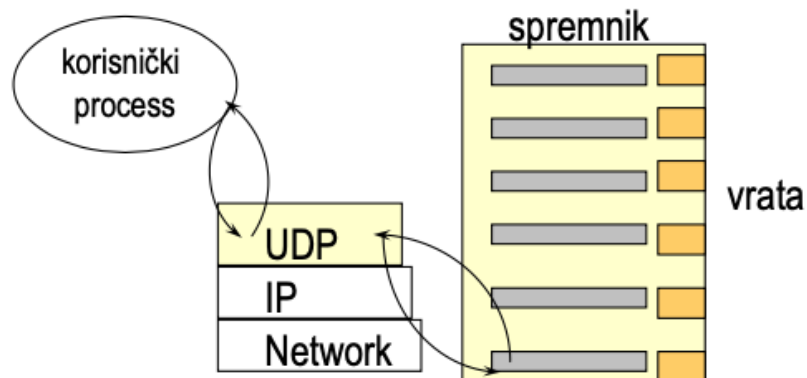
- Komunikacija korištenjem priključnica - Socket API
- Poziv udaljene procedure - RPC
- Raspodijeljeni objekti - poziv udaljene metode RMI

1. SOCKET API

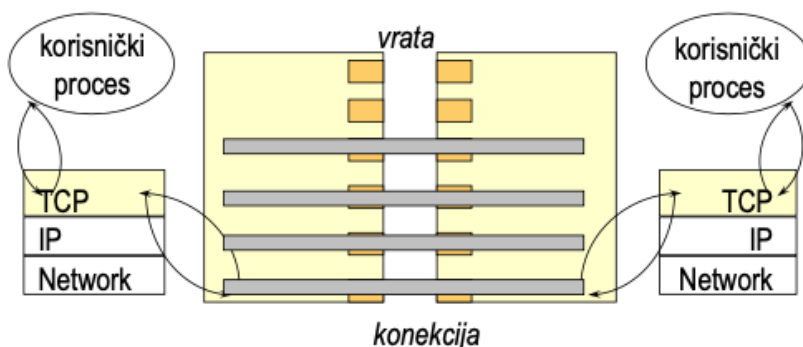
-koristi funkcionalnost transportnog sloja(TCP-pouzdana konekcijska komunikacija i UDP-nepouzdana beskonkcijska komunikacija)

-priključnica - pristupna točka preko koje aplikacija šalje podatke u mrežu i iz koje čita primljene podatke

UDP komunikacija preko priključnice - asinkrona, tranzijentna, beskonkcijska



TCP komunikacija preko priključnice - sinkrona, tranzijentna, konekcijska



Višedretveni poslužitelj i klijenti:

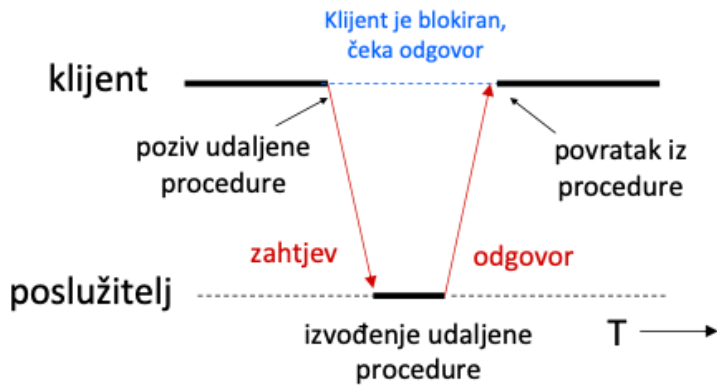
Poslužitelj: primanje konkurentnih klijentskih zahtjeva, složena obrada podataka, komunikacija s BP

Klijent: otvaranje mrežne konekcije i primanje podataka

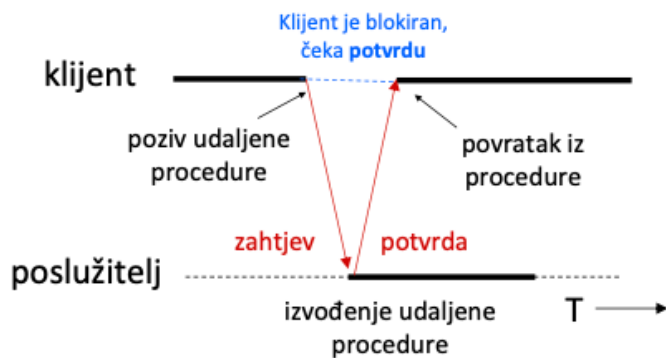
2. Poziv udaljene procedure(RPC)

-omogućuje procesima pozivanje i izvođenje procedura na udaljenom računalu

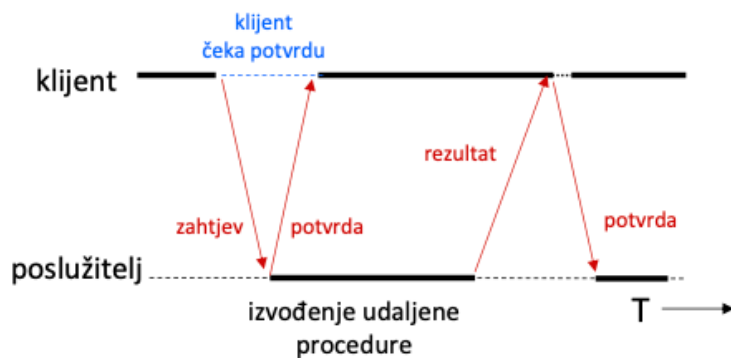
•Sinkroni RPC



•Asinkroni RPC



•Odgođeni sinkroni RPC



-primjer **gRPC**

- u .proto file se definira format protokola za poslužitelja i onda se te metode i objekti nasljeđuju u poslužitelju i implementiraju
- protokol je neovisan o prog. jeziku
- klijent pak samo treba kreirati konekciju prema poslužitelju i pozivati remote metodu na poslužitelju

3. Poziv udaljene metode

-unaprijeđeni rpc, samo se ne zovu remote procedure vec remote metode

-osim remote metoda, na udaljenim računalima imamo i udaljene objekte

-primjer **Java RMI**

- **prednosti**- visoka transparentnost, ista sintaksa, brza implementacija, čitljiviji kod
- **nedostatci** - poziv udaljene metode je puno sporiji od lokalne

