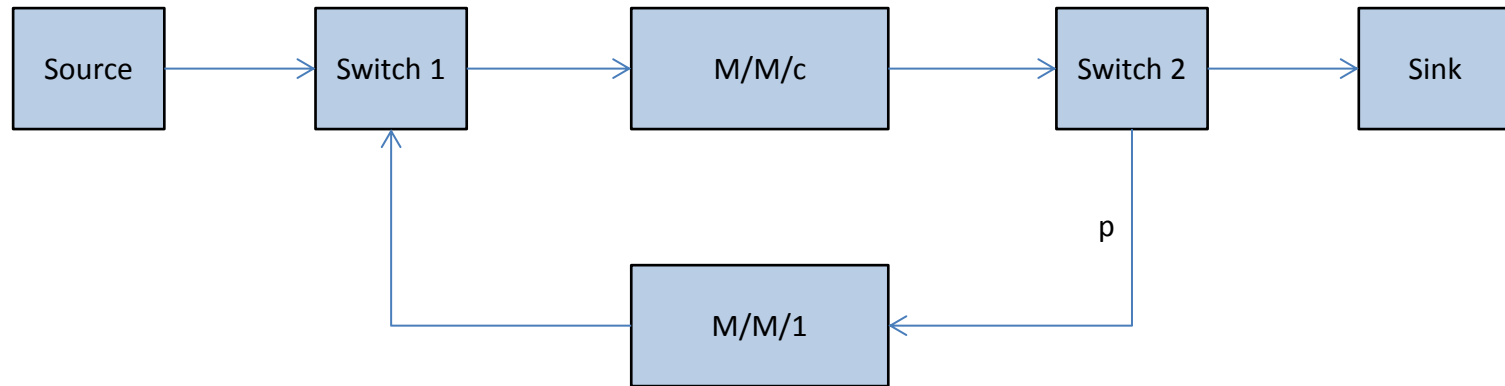


Teme za prvo seminarsko nalogo

Modeliranje računalniških omrežij

29/10/2013

1. Strežna enota z delno vrnitvijo zahtev v strežbo



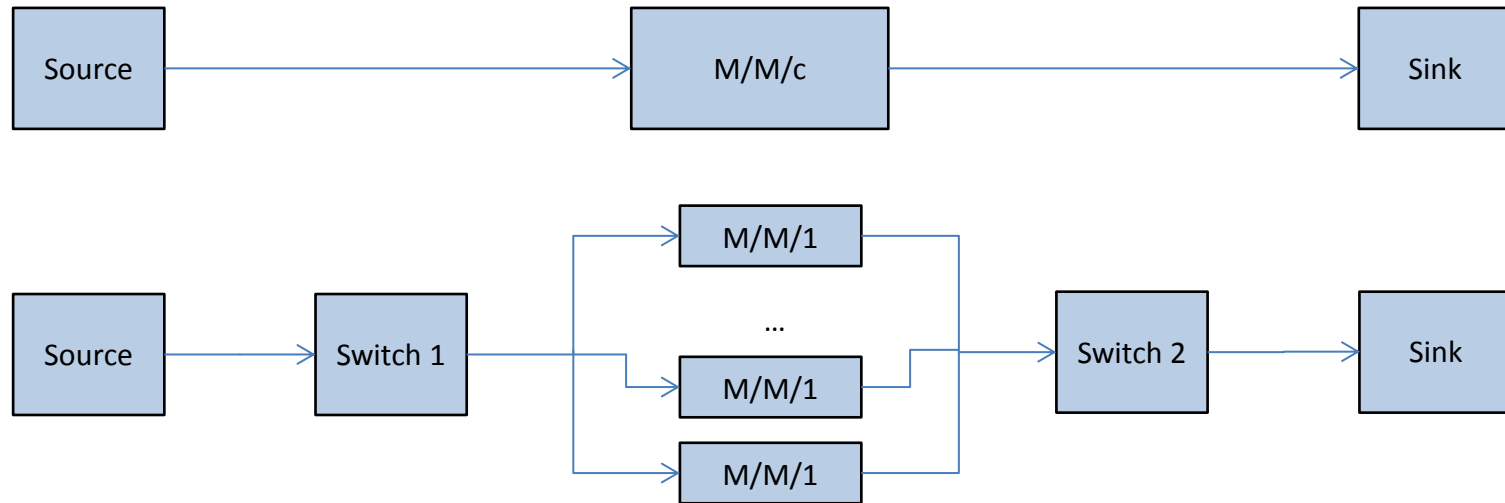
a) V simulacijskem okolju Omnet++ realizirajte podano omrežje:

- modul "Switch 2" naj bo delovalo stohastično, t.j. modul pošilja postreženo zahtevo iz modula M/M/c ponovno nazaj v strežbo z verjetnostjo p . Zahteva naj se v tem primeru ustrezno postreže tudi v dodatnem modulu M/M/1. Parametri c , p in dolžina obeh čakalnih vrst naj bodo podani v ini datoteki.

b) Simulirajte delovanje omrežja in zabeležite rezultate simulacij.

- `serviceTime` modula M/M/c naj bo porazdeljen normalno, `mean` se spreminja od 1s do 3s, `stddev` pa je vedno enak `mean*0.5`.
- `interArrivalTime` modula Source pa naj bo porazdeljen eksponentno, `mean` se spreminja od 0.5s do 1.5s

2. Primerjava M/M/c in c - M/M/1



a) V simulacijskem okolju Omnet++ realizirajte podana omrežja:

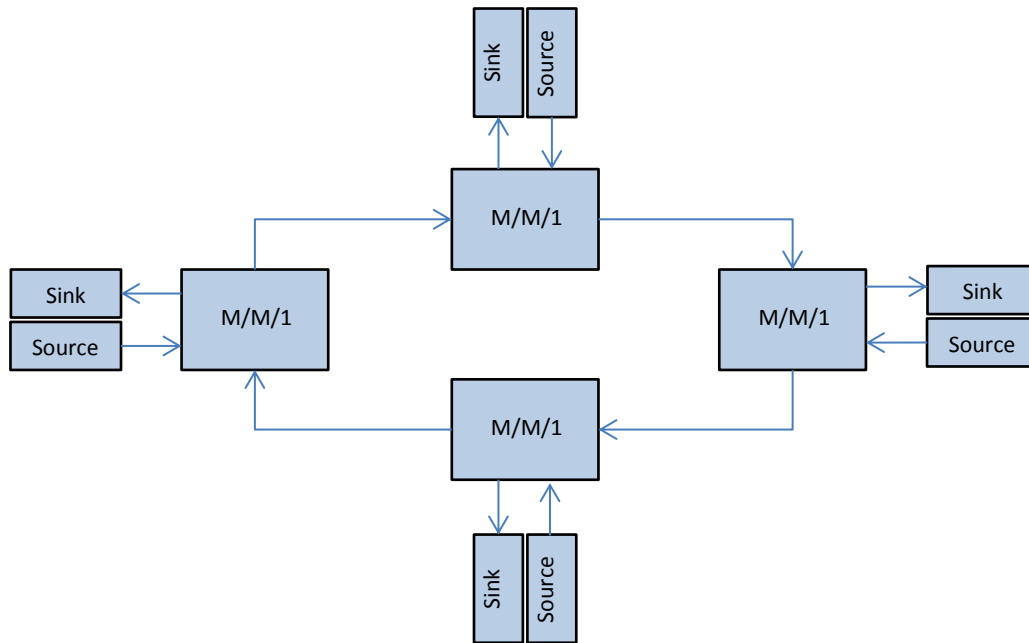
- parametra c in dolžina čakalnih vrst naj bosta podana v ini datoteki.

b) Simulirajte delovanje omrežij, zabeležite rezultate simulacij in primerjajte rezultate.

- `serviceTime` vseh strežnikov naj bo enak. Simulirajte vsaj 10 različnih konfiguracij, pri čemer se bo `serviceTime` spremenil od 1s do 2.5s
- `interArrivalTime` modula `Source` pa naj bo porazdeljen normalno, `mean` se spreminja od 0.5s do 1.5s, medtem ko `stdev` naj bo vedno enak `mean*0.5`.

c) Kateri strežniški sistem je učinkovitejši?

3. P2P torrent omrežje



a) V simulacijskem okolju Omnet++ realizirajte podano omrežje:

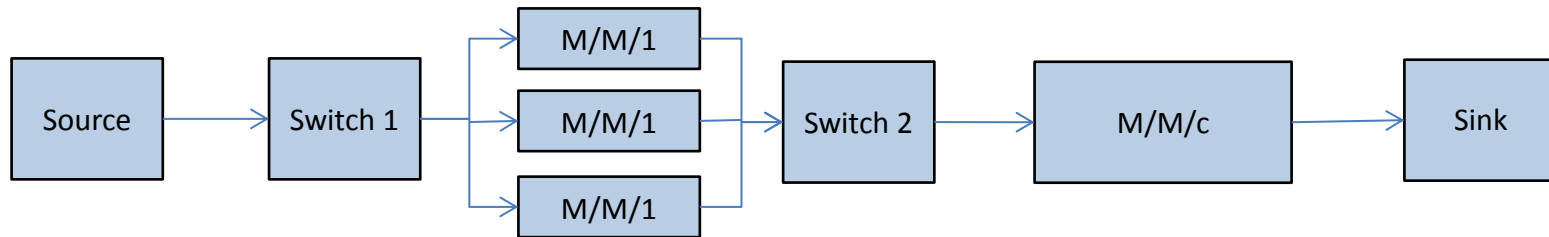
- Parameter dolžina čakalnih vrst naj bo podan v ini datoteki.

b) Simulirajte delovanje omrežja P2P, zabeležite rezultate simulacij in prikažite rezultate.

- `serviceTime` vseh strežnikov naj bo porazdeljen normalno, mean se spreminja od 0.8s do 1.5s (za vsak strežnik si izberite drugačen interval), medtem ko `stdev` naj bo vedno enak `mean*0.5`.
- `interArrivalTime` modula `Source` pa naj bo porazdeljen normalno, mean se spreminja od 0.5s do 1s, medtem ko `stdev` naj bo vedno enak `mean*0.5`.

c) Kolikšen je povprečni življenjski čas zahteve v sistemu P2P?

4. Strežniki s prioriteto čakalno vrsto (Priority queue)



a) V simulacijskem okolju Omnet++ realizirajte podano omrežje:

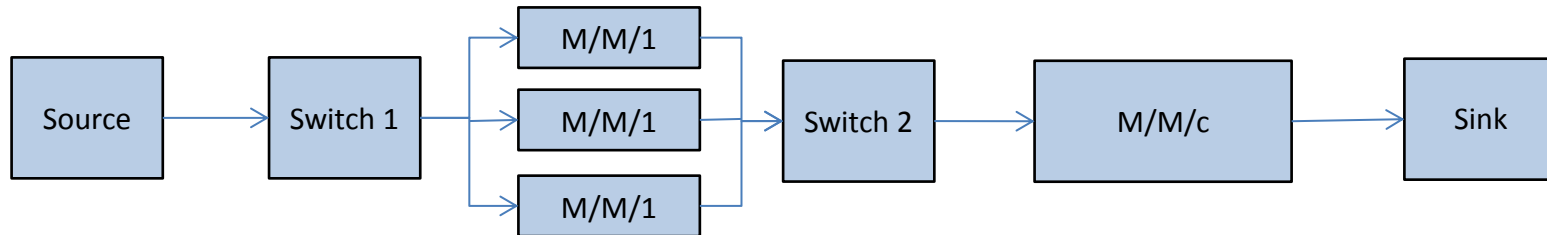
- parametra c in dolžina čakalnih vrst naj bosta podana v ini datoteki.
- Sporočila v omrežju naj bodo imela prioritetni indeks od 1 do 10, pri čemer imajo višjo prioriteto sporočila z višjim indeksom.
- Strežnik $M/M/c$ naj vsebuje PRIORITETNO čakalno vrsto. Namig: spremenite (oz. dopolnite) rešitev iz naloge2, tako da boste implementirali prioriteto čakalno vrsto v modulu $M/M/c$.

b) Simulirajte delovanje omrežij, zabeležite rezultate simulacij in primerjajte rezultate.

- `serviceTime` vseh strežnikov $M/M/1$ naj bo enak (recimo 0.8 s). `serviceTime` strežnika $M/M/c$ pa naj bo porazdeljen normalno, `mean` se spreminja od 1 s do 2 s, medtem ko `stdev` naj bo vedno enak `mean*0.5`.
- `interArrivalTime` modula `Source` pa naj bo porazdeljen normalno, `mean` se spreminja od 0.1 s do 0.8 s, medtem ko `stdev` naj bo vedno enak `mean*0.5`.

c) Za posamezni prioritetni razred sporočil izrišite stolpčni diagram povprečnih časov bivanja zahtev v čakalni vrsti strežnika $M/M/c$.

5. Strežniki s prioriteto čakalno vrsto (Shortest job first - SJF)



a) V simulacijskem okolju Omnet++ realizirajte podano omrežje:

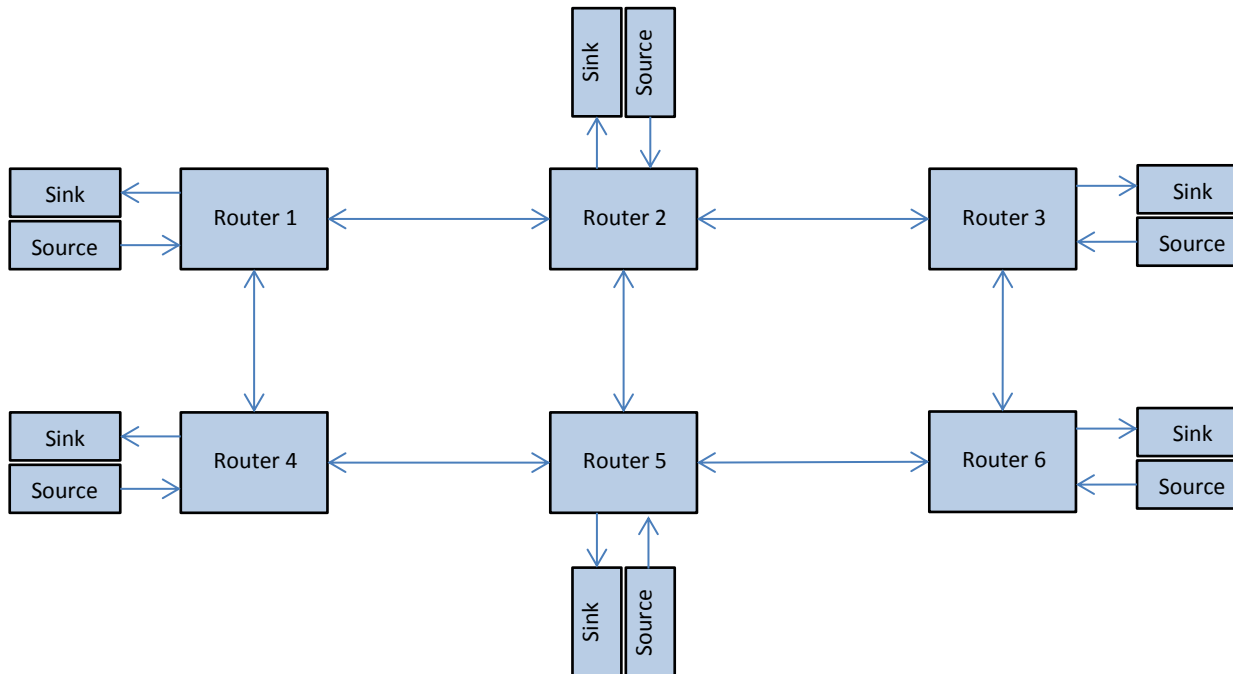
- parametra c in dolžina čakalnih vrst naj bosta podana v ini datoteki.
- Sporočila v omrežju naj bodo imela prioritetni indeks od 1 do 10, pri čemer imajo višjo prioriteto sporočila z nižjim indeksom. Indeks naj predstavlja dolžino sporočila (angl. *Job length*) in hkrati tudi čas, v sekundah, potreben za njegovo uspešno strežbo (recimo, če ima sporočilo prioriteto 5, potem bo tudi čas strežbe enak 5 sekund).
- Strežnik M/M/c naj vsebuje PRIORITETNO čakalno vrsto. Namig: spremenite (oz. dopolnite) rešitev iz naloge2, tako da boste implementirali prioriteto čakalno vrsto v modulu M/M/c.

b) Simulirajte delovanje omrežij, zabeležite rezultate simulacij in primerjajte rezultate.

- `serviceTime` vseh strežnikov M/M/1 naj bo enak (recimo 0.8 s). `serviceTime` strežnika M/M/c pa naj bo porazdeljen normalno, mean se spreminja od 1s do 2s, medtem ko `stdev` naj bo vedno enak `mean*0.5`.
- `interArrivalTime` modula Source pa naj bo porazdeljen normalno, mean se spreminja od 0.1s do 0.8s, medtem ko `stdev` naj bo vedno enak `mean*0.5`.

c) Za posamezni prioritetni razred sporočil izrišite stolpčni diagram povprečnih časov bivanja zahtev v čakalni vrsti strežnika M/M/c.

6. Poenostavljen IP protokol



a) V simulacijskem okolju Omnet++ realizirajte podano omrežje:

- Vsak modul *Router* naj vsebuje unikatni parameter `IP_address` tipa `string`, ki ga enolično razlikuje med vsemi ostalimi moduli v omrežju.
- Napišite nov razred sporočil `IP_Message`, ki bo dedoval iz razreda `cMessage`. Sporočila tipa `IP_Message` naj vsebujejo še naslednja dodatna polja: `IP_source` (`string`), `IP_destination` (`string`) in `Content` (`string`). V podanem omrežju uporabljate izključno samo ta novi tip sporočil.
- Naslavljanje sporočil v omrežju (v modulu tipa *Router*) realizirate tako, da dinamično implementirate seznam IP naslovov, ki bo vseboval za posamezni IP naslov omrežja tudi *gate* trenutnega modula *Router*, na katerem je potrebno dostaviti sporočilo. Če se IP naslov modula *Router* ujema z `IP_destination` sporočila, potem naj se zgenerira *Aknowledgment* sporočilo (s praznim poljem `Content`).

- b) Simulirajte delovanje omrežij, zabeležite rezultate simulacij in primerjajte rezultate. Uporabljajte podobne nastavitve za medprijodne čase zahtev kot pri prejšnjih temah.