**Osnovno o redovima:**

Ukupno kašnjenje T = TRANS + PROP + QD + PROC

TRANS = (veličina paketa)/(brzina prijenosa)

PROP = (duljina prijenosnog medija)/(brzina prostiranja signala kroz prijenosni medij)

QD = kašnjenje (queueing delay) u npr.

PROC = trajanje obrade u npr. komutacijskom

Dolazni proces:

- međudolazno vrijeme između korisnika n i n+1, srednja vrijednost

- srednja brzina(intenzitet) dolazaka

Proces posluživanja:

-vrijeme posluživanja korisnika n, srednja vrijednost

- je brzina posluživanja

, Ts=Ttrans

**Važne oznake:**

- broj korisnika u sustavu u trenutku t

- broj korisnika u redu čekanja

- broj korisnika na posluživanju

N - srednji broj korisnika u sustavu

- srednji broj korisnika u redu čekanja

- srednji broj korisnika na posluživanju

- prosječno trajanje posluživanja

- vrijeme boravka korisnika n u sustavu

- vrijeme koje korisnik n provede u redu čekanja

T - srednje vrijeme boravka korisnika u sustavu (ukupno kašnjenje)

W - srednje vrijeme čekanja korisnika u redu čekanja

- vjerojatnost da je n korisnika u sustavu u trenutku t

- stacionarna vjerojatnost da je n korisnika u sustavu

- vjerojatnost da će korisnik provesti više od t vremena u sustavu

- vjerojatnost da će korisnik provesti više od t vremena u redu čekanja

- Ukupna srednja brzina dolazaka

– iskoristivost

a – intenzitet (ponuđeni) promet (jedinica erl – erlang)

- srednji broj zauzetih poslužitelja kad je sustav u stacionarnom stanju

E[] – očekivanje broja u redu čekanja

Formule poveznice:

, , , , ,

,

- očekivanje općenito

,očekivani broj korisnika u sustavu

, i=redni broj stanja u kojem ima reda, j = redni broj zadnjeg stanja

Little:

, T je ovdje srednje trasnimisjko kašnjenje

,a trajanje posluživanja, sustav prazan

, -očekivani broj korisnika u sustavu

Notacija procesa:

A/S/m/c/K/DP

A: dolazni proces, S: razdiobu posluživanja, m: broj paralelnih poslužitelja, c: je kapacitet sustava

Iskoristivost

svaki poslužitelj poslužuje prosječno korisnika,

prosječno trajanje posluživanja

Intenzitet prometa: - mjera ukupnog dolaznog prometa u sustav

,

Srednje kašnjenje paketa:

**Markovljev lanac:**

vremensko diskretni:

Prijelazna vjerojatnost u n koraka (Chapman-Kolmogorovljeve)

, je element (i, j) u matrici P na n-tu potenciju

Vremenski kontinuirani:

,brzina kad je u stanju i proces prelazi u stanje

, Ukupna brzina prijelaza iz stanja i

, prosječno vrijeme provedeno u stanju i

vjerojatnost zauzeća stanja j - - je proporcionalno vremenu provedenom u stanju j

JGR

, brzina prijelaza iz = u stanja j

JLR

**Redovi cekanja:**

Red M/M/1:

Ako je ρ≥ 1 sustav je nestabilan i nikad nije u stacionarnom stanju

Ti – vrijeme provedeno u stanju i, eksponencijalno parametrom

,

JLR:

, vjerojatnost da je u sustavu više od n korisnika

, , vjerojatnost da korisnik čeka

)

, srednji broj korisnika

, prosječno kašnjenje po korisniku (ček. + posl), vrijeme zadržavanja,odziva

*,* srednje vrijeme čekanja u redu

,broj korisnika u redu ( bez posluživanja)

, vrijeme boravka, zadržavanja

Red M/M/1: sustav s promjenljivim brzina

,

Faktori iskoristivosti ρ za oba su sustava isti ⇒ stacionarne razdiobe su jednake, prosječni broj paketa u sustavu je jednak

Statistički MUX: prijenosi se m Poissonova toka brzine λ/m, svi se tokovi stope u jedan (stat-mux) brzine λ

TDM ili FDM: podijeliti link na m kanala svaki kapaciteta 1/m i dodijeliti po jedan kanal svakom prometnom toku

M/G/1

, uvjet stabilnosti

PASTA:

Poisson Arrivals See Time Averages, vjerojatnost da je za M/M/1,

LAA (Lack of Anticipation Assumption): buduća međudolazna vremena i vremena posluživanja prethodno pristiglih korisnika su neovisna

Red M/M/\*

PASTA uvijek zadovoljeno

Red M/M/1/K: sustav s gubicima, uvijek stabilan

za

, , stac.razdoba

Ostale veličine Little =

Red M/M/c

Erlang C formula:

, P{čekanja u redu} = ,

stat-Mux

, Vjerojatnost da će korisnik provesti više od t u sustavu

, Vjerojatnost da će korisnik provesti više od t u redu čekanja

Red M/M/c/K

, za

, za

, za

,

, za

Red M/M/ ili M/G/

, Poisson brzine λ/μ (= a ≡ ponuđeni promet )

Red M/M/c/c: sustav s gubicima – Erlang B formula

, stac. vjerojatnosti, sustav je u stanju n

ERLANG-B formula: vjerojatnost blokiranja,odbacivanja, , PASTA svojstvo

,

,

Interpretacija u telefoniji (red M/M/c/c):

- je dio vremena u kojem je n-ti kanal zauzet (stanje n), je dio vremena u kojem je n-ti kanal zauzet (stanje n). - je prosječni broj poziva u jedinici vremena, je srednje vrijeme zauzeća kanala

- je ponuđeni promet po poslužitelju (kanalu) i često se zove intenzitet prometa ( = ρ)

ZADACI:

kašnjenje od izvora do odredišta T = Ttrans+Tprop

broj paketa na linku: N=Tprop/Tpak , Tpak=Lpak/Vtrans

srednji broj zauzetih kanala jednak je prenesenom prometu a

Srednji broj korištenih telefona = prometno opterećenje

Očekivani broj korisnika na posluživanju = a =

srednje vrijeme posluživanja -

kad imamo promjenjive , računamo očekivanje i onda dalje računamo s njima

ALGORITMI:

ST - Algoritam za konstrukciju ST starta od jednog čvora i u svakoj iteraciji proširuje stablo za jedan čvor i jednu granu, čvorove gleda odpočetka i gleda dal postoji grana

MST:

Prim Dijsktra: Početni (trivijalni) fragment je neki proizvoljno odabrani čvor. Aktualni fragment se stalno povećava uzastopnim dodavanjem odlaznih grana minimalne težine

Algoritam manipulira s dva skupa čvorova, U svakoj iteraciji se jedan čvor prebacuje iz C u L

L: Označeni (labelirani) čvorovi (poznat je najkraći put), C: Kandidatski čvorovi (najkraći put još nije poznat)

Stablo = ∅

Izaberi proizvoljni vrh iz V(Graf) i stavi ga u V(Stablo).

Dok je Broj\_Vrhova(Stablo) < Broj\_Vrhova(Graf) ponavljaj

Izaberi vrh koji nije u V(Stablo) a susjedan je nekom vrhu iz

V(Stablo) i pri tome je težina brida koja ih spaja minimalna.

Stavi taj vrh zajedno s njemu pripadajućim bridom u Stablo

Kruskalov algoritam: Početno su svi čvorovi (trivijalni) fragmenti. Uzastopno se fragmenti kombiniraju u nove fragmente uključivanjem onih grana minimalne težine koje pri takvim

kombiniranjima ne stvaraju ciklus

Stablo = ∅

Bridove grafa poredaj u padajući niz.

Dok postoji brid čije dodavanje u Stablo ne tvori ciklus ponavljaj

Uzmi najmanji brid koji ne tvori ciklus s bridovima iz Stabla i dodaj ga u Stablo.

SP, najkraći put:

Dijkstra

Stablo = ∅

Stavi u stablo početni vrh i označi ga s udaljenosti 0.

Dok je Broj\_Vrhova(Stablo) < Broj\_Vrhova(Graf) ponavljaj

Za sve vrhove koji su susjedni nekom vrhu iz stabla izračunaj najmanju udaljenost od početnog vrha kao udaljenost vrha u stablu kojem je taj vrh susjedan + težina brida koji ih spaja

Od svih tih vrhova izaberi onaj koji ima najmanju vrijednost udaljenosti i stavi ga u stablo zajedno s pripadajućim bridom

Početno =0,

Tablica: redovi = iteracija, broj čvora i, stupci = Di

Bellman-Ford

D(i,j) = cijena za najbolju rutu od i do krajnjeg j

c(i,j) ili d(i,j) = cijena od čvora i do susjeda j

Ukoliko je i=j ili i i j nisu susjedi cijena se postavlja na beskonačnost.

D(i,j)= min {d(i,k) + D(k,j)}, pri cemu k predstavlja susjeda

dobijamo najkrači put za h+1 hop (link) između čvora i i j kao:

D(i,j)[h+1]=min{d(i,k)+D(k,j)[h]} , k – označava susjede, h=0,1,...H-2 , H = broj čvorova

Prvo se računa najkraći put sa jednim linkom za svaki čvor

D(i,j)[1]=min{d(i,k) + D(k,j)[0]} = d(i,j)

Drugi korak u rješavanju ovog problema je da se računa najkraći put za h=2 linkova

Tablica: redovi = čvorovi iz kojih se dolazi, stupci = , zadnji stupac = graneSPT

Floyd-Warshall

Za vrh\_kroz = 1 do broj\_vrhova

Za vrh\_od = 1 do broj\_vrhova

Ako G[vrh\_od,vrh\_kroz]!=∞

Za vrh\_do = 1 do broj\_vrhova

Ako G[vrh\_kroz,vrh\_do]!=∞

Ako G[vrh\_od,vrh\_do]==∞ ili G[vrh\_od,vrh\_do]>G[vrh\_od,vrh\_kroz]+G[vrh\_kroz,vrh\_do]

G[vrh\_od,vrh\_do]=G[vrh\_od,vrh\_kroz]+G[vrh\_kroz,vrh\_do]

Za svaki vrh, vrh\_kroz, iz grafa pokušavamo poboljšati put između neka druga dva vrha, vrh\_od i

vrh\_do tako da provjerimo da li se udaljenost smanjuje ako put između ta dva vrha prođe

kroz vrh\_kroz. Algoritam tokom izvođenja mijenja tablicu incidencije te na kraju izvođenja

u tablici na mjestu *i,j* imamo najmanju udaljenost između vrhova *i* i *j*.