

8. Matematičko modelovanje

Prof.dr Igor Radusinović

igorr@ucg.ac.me

Doc.dr Slavica Tomović

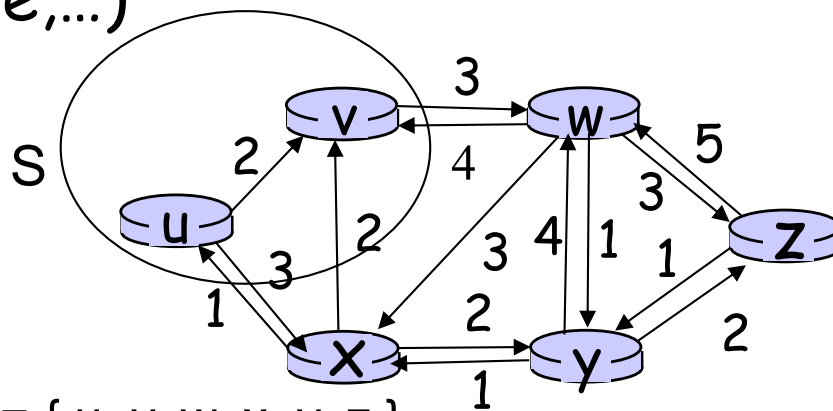
slavicat@ucg.ac.me

Matematičko modelovanje

- Grafovi
- Jackson-ove mreže
- Kontrola zagušenja
- Dinamičko rutiranje i kontrola zagušenja
- Bežično čvorište

Grafovi

- Čvorovi
- Grane
- Težinski faktori grana
- "Cut" (podskup skupa čvorova)
- Široka primjena (algoritmi rutiranja, modelovanje P2P komunikacije,...)



Graf: $G = (N, E)$

N = skup čvorova = $\{ u, v, w, x, y, z \}$

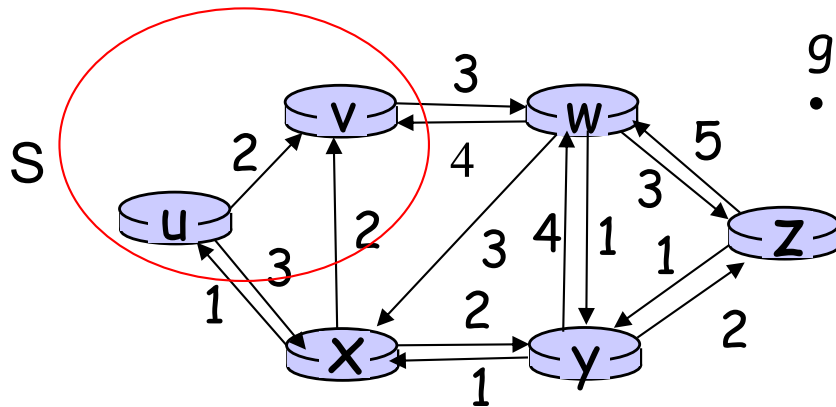
E = skup grana = $\{ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) \}$

S = cut = $\{ u, v \}$

$S^C = N/S = \{ x, y, w, z \}$

Grafovi

Kapacitet cut-a

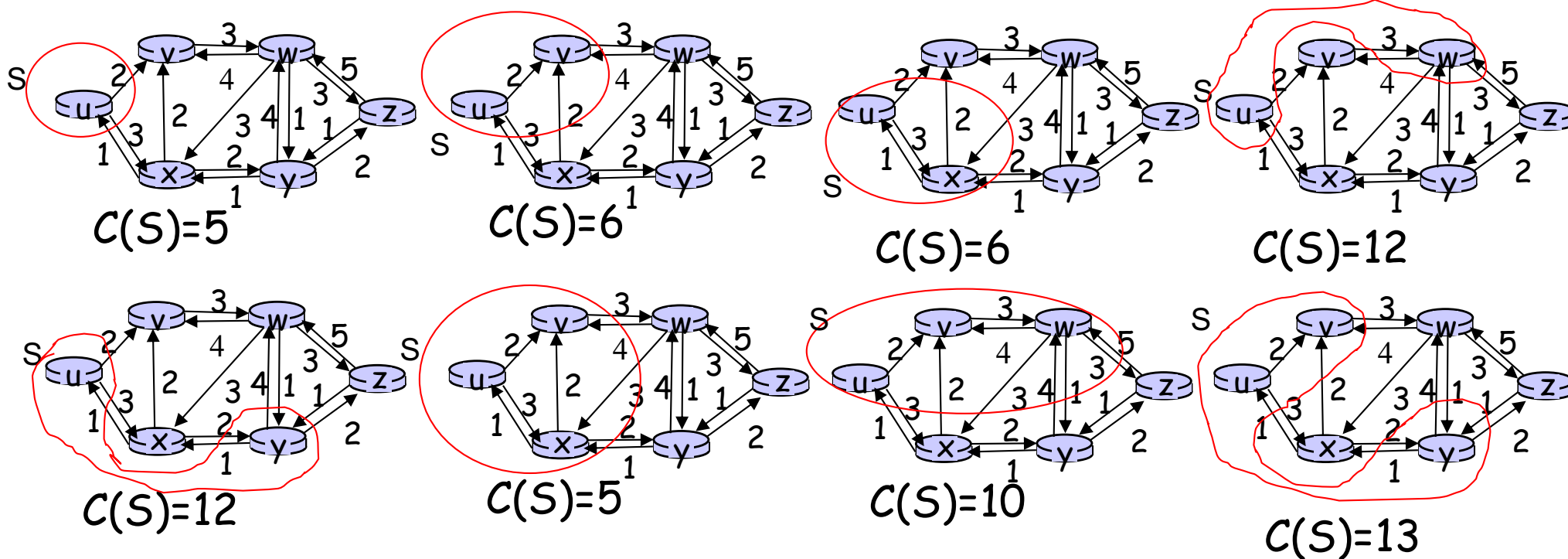


- Kapacitet cut-a S jednak je sumi težinskih grana koje izlaze iz njega i ulaze u S^c
- $c(S) = c(v, w) + c(u, x) = 3 + 3 = 6$

Grafovi

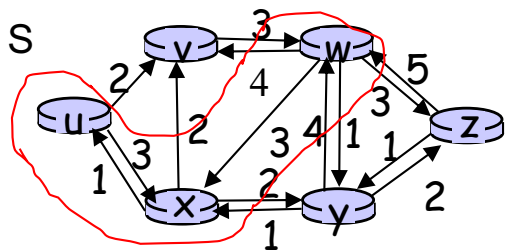
MAX-FLOW MIN-CUT teorema

- Rješava problem određivanja maksimalne propusnosti toka koji se može uspostaviti između dva čvorišta
- Neka težinski faktor predstavlja kapacitet linka
- Max-flow min-cut teorema kaže da je maksimalna vrijednost propusnosti toka koji se može uspostaviti između čvorova u i z jednaka minimalnom kapacitetu cut-ova S gdje $u \in S$ a $z \in S^c$

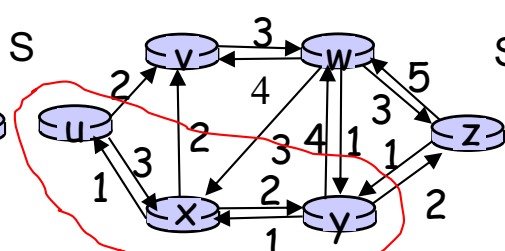


Grafovi

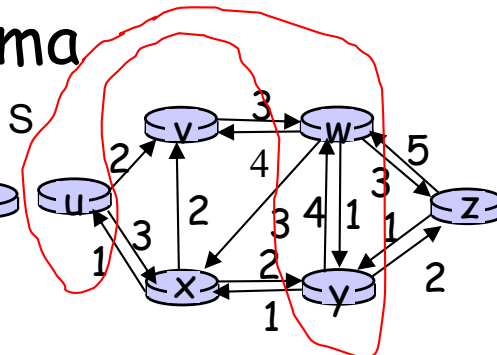
MAX-FLOW MIN-CUT teorema



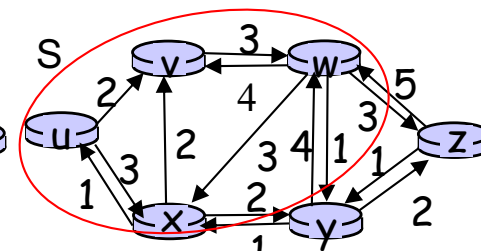
$$C(S)=14$$



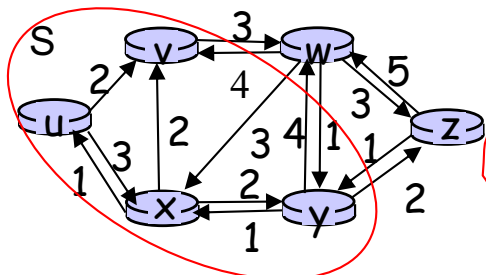
$$C(S)=10$$



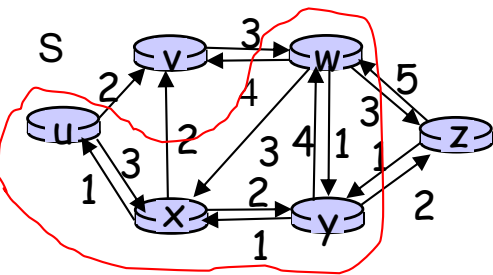
$$C(S)=18$$



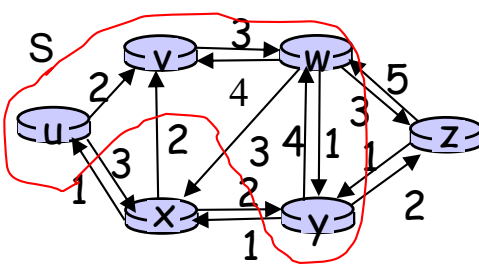
$$C(S)=6$$



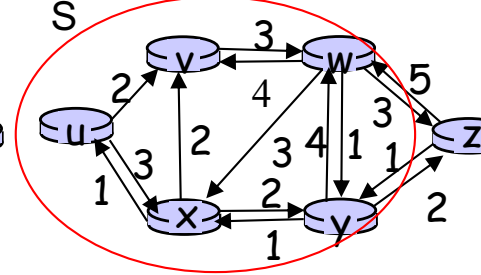
$$C(S) = 9$$



$$C(S)=13$$



$$C(S)=12$$



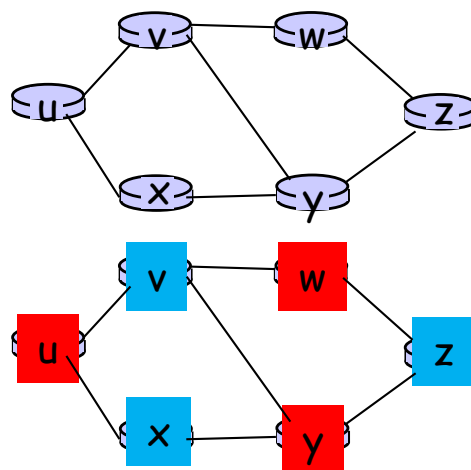
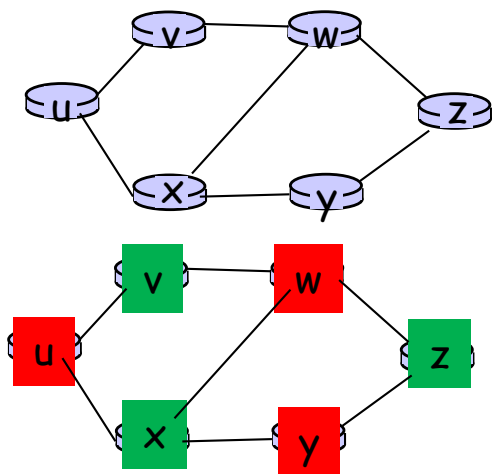
$$C(S)=5$$

Očigledno je maksimalna propusnost toka od u do z za mrežu abstrahovanu zadatim grafom jednaka 5.

Grafovi

Bojenje grafa i MAC protokoli

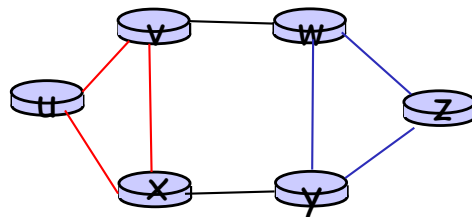
- ❑ Neusmjereni graf se sastoji od čvorova povezanih neusmjerenim granama.
- ❑ Dva čvora su susjedna ako su povezani granom
- ❑ Graf je povezan ako se svaki čvor može doseći iz svakog čvora
- ❑ Graf je potpuno povezan ako su bilo koja dva čvora povezana granom.
- ❑ Stepen čvora je jednak njegovom broju susjeda
- ❑ Maksimalni stepen grafa jednak je maksimalnom stepenu njegovih čvorova.
- ❑ Bojenje grafa predstavlja dodjelu boje svakom čvoru tako da nijedna dva susjeda nemaju istu boju.
- ❑ Minimalan potreban broj boja za bojenje grafa zove se hromatski broj grafa.



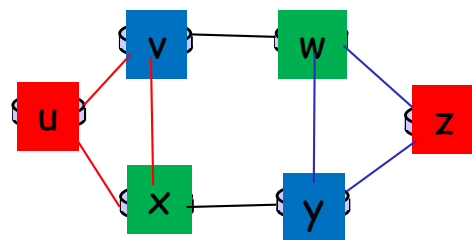
Grafovi

Bojenje grafa i MAC protokoli

- Clique je podskup neusmjerenog grafa kod koga su svaka dva čvora susjedi (potpuno povezani graf)
- Svi čvorovi u clique moraju biti različite boje
- Maksimalan broj čvorova u clique predstavlja donju granicu hromatskog broja grafa.

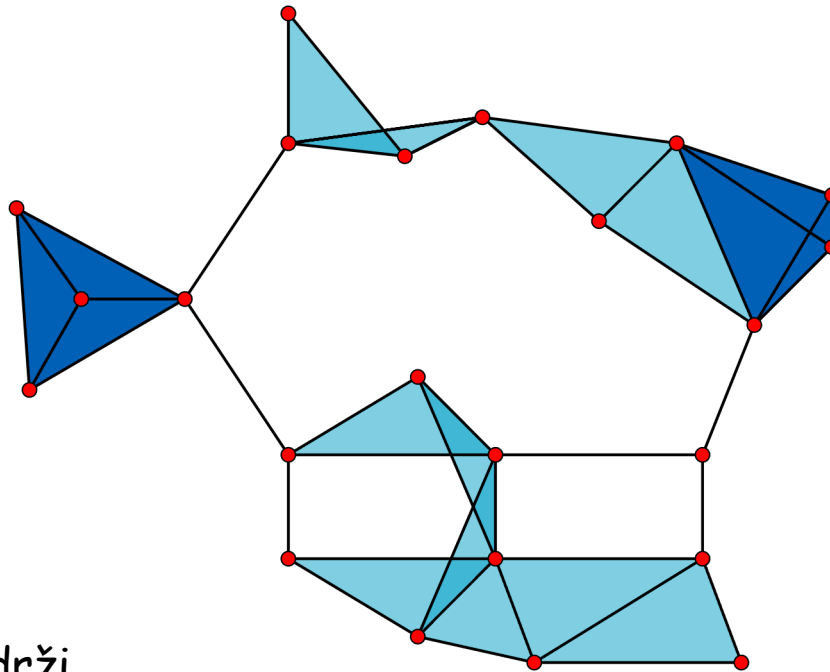


Koliki je hromatski broj ovog grafa (ima 2 clique sa po tri čvora)?



Grafovi

Bojenje grafa i MAC protokoli



Graf sa slike sadrži

- ❑ 23 jednočvorna clique-a (čvorovi),
- ❑ 42 dvočvorna clique-a (grane),
- ❑ 19 tročvornih clique-ova (svijetlo plavi trouglovi), i
- ❑ 2 četvoročvorna clique-a (tamno plave površine).

Grafovi

Bojenje grafa i MAC protokoli

Brook-ova teorema

U povezanom grafu u kome svaki čvor ima najviše Δ susjeda, čvorovi mogu biti obojeni sa svega Δ boja, izuzev u slučajevima:

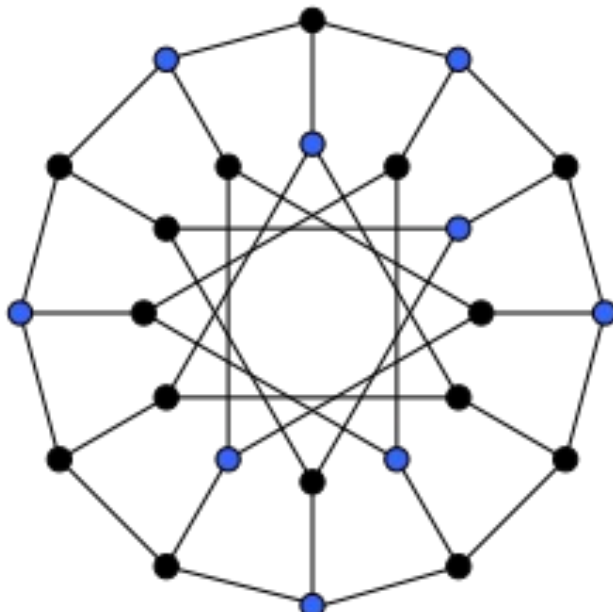
- Potpuno povezanog grafa
- Grafova u formi prstena sa neparnim brojem čvorova

kojima je potrebno $\Delta + 1$ boja.

Grafovi

Bojenje grafa i MAC protokoli

- Neka čvorovi predstavljaju bežična čvorišta
- Čvorovi su povezani granom ako jedan drugom izazivaju interferenciju
- Ovako definisan graf je graf konflikta
- Ako se na ovakvom grafu primijeni bojenje grafa onda svi čvorovi koji imaju istu boju mogu da šalju podatke bez interferencije i čine nezavisan skup.
- Maksimalni nezavisni skup grafa G je nezavisni skup sa najvećim brojem čvorova, koji predstavlja broj nezavisnosti grafa G .



Devet plavih čvorova formiraju maksimalni nezavisni podskup grafa (Generalizovani Petersonov graf)

Grafovi

Bojenje grafa i MAC protokoli

Primjena 1

- ❑ Neka čvorovi grafa predstavljaju WiFi access pointe.
- ❑ Boje predstavljaju različite kanale.
- ❑ Bojenje osigurava da access point-i međusobno ne interferiraju
- ❑ Slična primjena je i kod baznih stanica

Grafovi

Bojenje grafa i MAC protokoli

Primjena 2

- Alokaciji vremenskih slotova u bežičnoj mesh mreži sa J čvorišta.
- Neka paketi dolaze do bežičnih čvorišta koji ih moraju poslati.
- Radi jednostavnosti neka se paketi šalju samo jedanput a onda napuštaju mrežu.
- Neka je λ_i dolazna brzina paketa na čvoru i ($i=1,2,\dots,J$)
- Vektor $\lambda=(\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_J)$
- Postavlja se pitanje kada je moguće da bežični čvorovi šalju podatke ovim brzinama tako da nema interferencije (λ je *feasible* brzina)?
- Bežični čvorovi se mogu organizovati u nezavisne skupove S čiji članovi neinterferiraju.
- Neka su $(S_1, S_2, \dots, S_K)$ maksimalni nezavisni skupovi a $p=(p_1, p_2, \dots, p_K)$ vektor vjerovatnoća koje zadovoljavaju uslov normalizovanosti ($p_1 + p_2 + \dots + p_K = 1$)
- Ako čvorovi nezavisnog seta S_k šalju podatke tokom p_k dijela vremena tada čvor i ($i=1,\dots,J$) šalje podatke tokom

$$\sum_{k=1}^K p_k 1\{i \in S_k\}$$

dijela vremena

- Do interferencije neće doći ako za svako čvorište važi

$$\lambda_i \leq \sum_{k=1}^K p_k 1\{i \in S_k\}$$

Grafovi

Bojenje grafa i MAC protokoli

Primjena 2

Prethodno se implementira na sledeći način

- ❑ Izračuna se vektor p
- ❑ Kreira se slot koji je dovoljno velik za prenos paketa
- ❑ Odabere se neki broj M
- ❑ Periodično se alocira svakih M slotova približno $p_k M$ slotova čvorovima nezavisnog seta S_k ($k=1, \dots, K$)
- ❑ Ovaj TSMP (*Time-Synchronized Mesh Protocol*) protokol je pogodan kada se brzine slanja sporije mijenjaju u odnosu na vrijeme koje je potrebno za izračunavanje vektora p i realokacije vremenskih slotova.

Grafovi

Bojenje grafa i MAC protokoli

Primjena 3

- ❑ Q-CSMA (Queue-based Carrier Sense Multiple Access)
- ❑ Kada čvorište želi da šalje podatke čeka dok se kanal ne oslobodi
- ❑ Kada se kanal oslobodi čvorište odbrojava timer čija je početna vrijednost slučajan broj čija srednja vrijednost, između ostalog, zavisi od broja čvorova koji se takmiče za kanal.
- ❑ Timer odbrojava samo kada je kanal slobodan.
- ❑ Kada timer istekne čvorište šalje frejm.
- ❑ Može se pokazati da i za ovaj MAC protocol važi

$$\lambda_i \leq \sum_{\substack{k=1 \\ i \in S_K}}^K p_k$$

- ❑ Sa nekim pozitivnim gap-om ε .
- ❑ Na žalost, kašnjenje paketa raste kada je ε malo.

Grafovi

Bojenje grafa i MAC protokoli

Primjena 3

- ❑ Q-CSMA se kombinuje sa kontrolom prihvatanja paketa u bafer koja prestaje sa prihvatanjem paketa ako popunjenost bafera pređe neku vrijednost koja zavisi od nivoa prioriteta.
- ❑ Kontrola prihvatanja efikasno podešava brzinu tako da je ona feasible i favorizuje pakete visokog prioriteta.
- ❑ Može da bira kašnjenja koja su prihvatljiva visoko prioritetnim paketima
- ❑ Prednost Q-CSMA u odnosu na TSMP je što ne zahtijeva trenutno prepoznavanje promjene uslova i automatsku adaptaciju i ne troši resurse prilikom alociranja slotova.

Jackson-ove mreže

- Najjednostavniji model mreže koju čine M/M/1 redovi čekanja
- Neka na ulaze Jackson-ove mreže sa J redova čekanja dolazi C različitih klasa saobraćaja koje imaju Poasonove raspodjele parametara λ_c ($c=1,2,\dots,C$)
- Paket klase c se prenosi preko nekog skupa redova čekanja $S(c) \subset \{1,2,\dots,J\}$ pri čemu se vodi računa o redosledu prolaska
- Pri tome pretpostavka je da paket prolazi kroz red čekanja samo jedanput
- Ukupna dolazna brzina je

$$\lambda = \sum_{c=1}^C \lambda_c$$

- Brzina kojim paketi dolaze na neki red čekanja j ($j=1,\dots,J$) iznosi

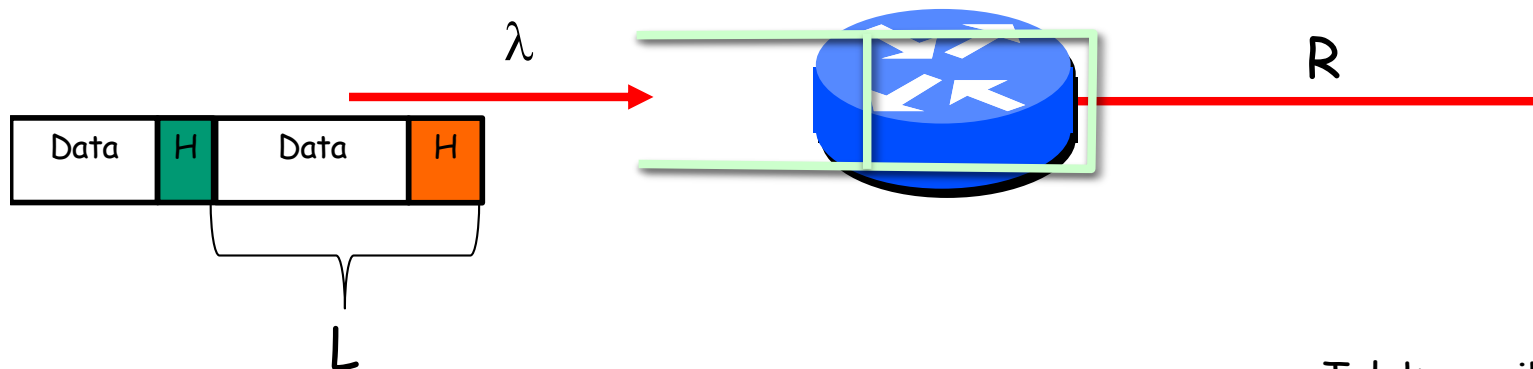
$$\gamma_j = \sum_{\substack{c=1 \\ j \in S(c)}}^C \lambda_c$$

- Odnosno jednaka je sumi Poasonovih brzina klasa koje prolaze kroz njega

Primjena redova čekanja

Primjer 1

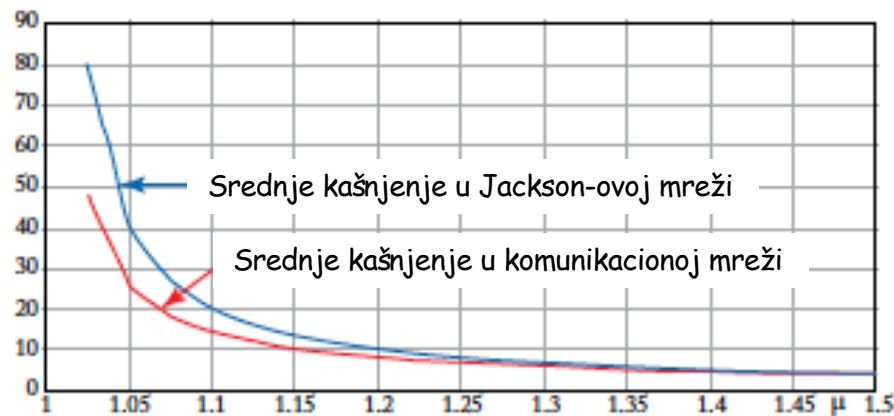
- U telekomunikacionim mrežama vrijeme posluživanja paketa je jednako kašnjenju uslijed prenosa
- Neka je dužina paketa L promjenljiva i može se opisati nezavisnom eksponencijalnom slučajnom promjenljivom srednje dužine $\bar{L}$ bita
- Ako je brzina prenosa na linku R onda vrijeme posluživanja ima eksponencijalnu raspodjelu srednje vrijednosti $\mu = \frac{\bar{L}}{R}$.
- Dodatno ako se dolazak paketa može opisati Poasonovim dolaznim procesom parametra λ , onda se red čekanja u baferu može približno modelovati $M/M/1$ redom čekanja.



Primjena redova čekanja

Primjer 2

- Pretpostaviti da nakon napuštanja reda čekanja (mrežnog čvorišta) paketi stižu na sledeći red čekanja (mrežno čvorište) i dalje nastavljaju saglasno *store & forward* komutaciji
- Jasno je da vremena dolazaka na drugi red čekanja nijesu nezavisna od vremena posluživanja paketa na prvom redu čekanja
- Vrijeme dolaska između dva susjedna paketa n i $n+1$ ne može biti manje od trajanja prenosa paketa n na prvom linku
- Prema tome, vrijeme dolaska paketa $n+1$ zavisi od vremena dolaska paketa n u naredni red čekanja
- Mreža sa dva redno povezana reda čekanja zato nije prava Jackson-ova mreža.
- Međutim, rezultati simulacija pokazuju da su odstupanja mala tako da se dva redno povezana reda čekanja mogu aproksimirati i modelovati Jackson-ovom mrežom.



Primjena redova čekanja

Primjer 3

- Balansiranje saobraćaja je važna funkcija mreže koja se može modelovati Jackson-ovom mrežom
- Neka postoje dva puta kroz mrežu za prenos paketa od izvora do destinacije preko kojih se odgovarajućim balansiranjem saobraćaja može smanjiti kašnjenje koje unosi mreža.
- Neka paketi dolaze saglasno Poasonovoj raspodjeli parametra λ i mogu nastaviti preko jednog od dva reda čekanja čije su brzine posluživanja μ_1 , odnosno μ_2 .
- Pretpostavka je da sa vjerovatnoćom p paketi odlaze na prvi red čekanja a sa vjerovatnoćom $1-p$ na drugi, pri čemu je izbor nezavisan.
- Prethodno znači da paketi na ulaz prvog bafera dolaze saglasno Poasonovoj raspodjeli parametra λp , odnosno na ulaz drugog bafera saglasno Poasonovoj raspodjeli parametra $\lambda(1-p)$
- Ako je $\lambda p < \mu_1$ i $\lambda(1-p) < \mu_2$ onda su srednja kašnjenja koje unose prvi i drugi red čekanja

$$\frac{1}{\mu_1 - \lambda p}, \text{ odnosno } \frac{1}{\mu_2 - \lambda(1-p)}$$

- Pošto paket prvo kašnjenje ima sa vjerovatnoćom p , a drugo sa vjerovatnoćom $1-p$, srednje kašnjenje koje unosi balanser saobraćaja iznosi

$$\bar{T} = \frac{p}{\mu_1 - \lambda p} + \frac{1-p}{\mu_2 - \lambda(1-p)}$$

- Može se pokazati da optimalno balansiranje ($p=0,5$) ne garantuje minimalno $\bar{T}$
- Ako se pretpostavi da je $\lambda < \mu_1 + \mu_2$ minimalno $\bar{T}$ se dobija za

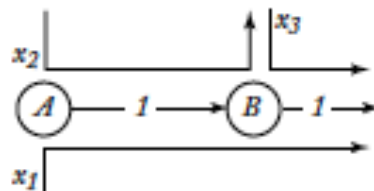
$$p = \frac{\lambda\sqrt{\mu_1} + \mu_1\sqrt{\mu_2} - \mu_2\sqrt{\mu_1}}{\lambda(\sqrt{\mu_1} + \sqrt{\mu_2})}$$

Kontrola zagušenja

- Već je rečeno da kontrola zagušenja predstavlja sprečavanje zagušenja mreže ili nekog njenog dijela
- Pokazano je da TCP kontrola zagušenja može veoma efikasno pristupom od kraja do kraja spriječiti pojavu velikih zagušenja u mreži
- Takođe, može se pokazati da TCP kontrola zagušenja garantuje fer podjelu resursa linka koji dijeli više TCP saobraćajnih tokova.
- Neka na ulaz mreže sa dva čvora (A i B) dolaze tri saobraćajna toka brzina x_1 , x_2 i x_3 . Brzine tokova su *feasible* ako važe sledeće relacije

$$g_a(x) := x_1 + x_2 - 1 \leq 0 \text{ i } g_b(x) := x_1 + x_3 - 1 \leq 0$$

- a i b su prvi i drugi link, a $x = (x_1, x_2, x_3)$
- Brzine $x_1 = 0$, $x_2 = 1$ i $x_3 = 1$ su *feasible*, a može se pokazati da obezbjeđuju maksimalnu propusnost ($x_1 + x_2 + x_3$) mreže koja je jednaka 2
- Ova maksimalna propusnost nije fer prema prvom toku koji ne dobija ni jedan dio kapaciteta mreže
- Brzine 1/2, 1/2 i 1/2 su *feasible*, obezbjeđuju ravnopravnost svih tokova ali je propusnost mreže svega 1,5



Kontrola zagušenja

- Treba izbalansirati maksimizovanje propusnosti mreže i fer raspodjelu njenih resursa
- Neka je $u(x_i)$ utility funkcija za tok i koja predstavlja koliko je korisniku vrijedno da brzina toka i iznosi x_i
- Utility funkcija raste sa povećanjem x_i ali je taj rast sve manji kako se x_i povećava (konkavna funkcija)
- Cilj je da se pronade $x = (x_1, x_2, x_3)$ tako da je $f(x) = u(x_1) + u(x_2) + u(x_3)$ maksimalno
- Neka je

$$u(x) = \begin{cases} \frac{1}{1-\alpha} x^{1-\alpha}, & \alpha \neq 1 \\ \log(x), & \alpha = 1 \end{cases}$$

- Za $\alpha \neq 1$ cilj je maksimizovati funkciju

$$f(x) = \frac{1}{1-\alpha} (x_1^{1-\alpha} + x_2^{1-\alpha} + x_3^{1-\alpha})$$

Uz uslov da važi

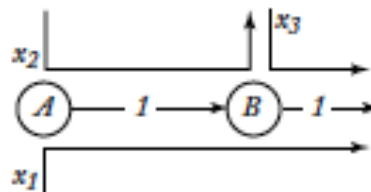
$$x_1 + x_2 \leq 1 \text{ i } x_1 + x_3 \leq 1$$

- Kako $f(x)$ raste za svako x_i jasno je da rješenje mora biti tako da važi

$$x_1 + x_2 = 1 = x_1 + x_3$$

- Odnosno

$$x_2 = x_3 = 1 - x_1$$



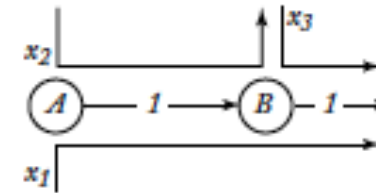
Kontrola zagušenja

- Na osnovu prethodnog

$$f(x) = \frac{1}{1-\alpha} (x_1^{1-\alpha} + 2(1-x_1)^{1-\alpha})$$

je maksimalna za

$$x_1 = \frac{1}{1 + 2^{\frac{1}{\alpha}}}$$



- Rješenje $(x_1, 1 - x_1, 1 - x_1)$ se kreće od $(0,1,1)$ koje obezbjeđuje maksimalnu propusnost do $(1/2, 1/2, 1/2)$ koja obezbjeđuje fer podjelu resursa
- Zanimljivo je vidjeti vrijednosti x_1, x_2 i x_3 koje maksimizuju funkciju $f(x)$ u zavisnosti od α
- Vidi se da promjena α od nule do beskonačnosti znači promjenu od maksimizovane sume brzina do maksimizovanja minimalne pojedinačne brzine



- Izbor parametra α za razmatrani slučaj obezbjeđuje mogućnost balansiranja dva oprečna zahtjeva: maksimalne propusnosti mreže i fer podjele resursa

Kontrola zagušenja

- U prethodnom slučaju je razmatrana u potpunosti poznata mreža što je rijedak slučaj
- U opštem slučaju se koristi Lagranžova funkcija

Teorema: Neka je $f(x)$ konkavna funkcija, a $g_j(x)$ ($j=a,b$) konveksna funkcija po x . Tada x^* predstavlja rješenje optimizacionog problema

$$\text{Maksimizuj } f(x)$$

$$\text{pod uslovom da je } g_j(x) \leq 0, j = a, b$$

Ako i samo ako x^* zadovoljava prethodno imaksimizuje

$$L(x, \lambda^*) := f(x) - \lambda_a^* g_a(x) - \lambda_b^* g_b(x)$$

za $\lambda_a^* \geq 0$ i $\lambda_b^* \geq 0$ takve da je $\lambda_j^* g_j(x^*) = 0$, za $j = a, b$

Dodatno ako je

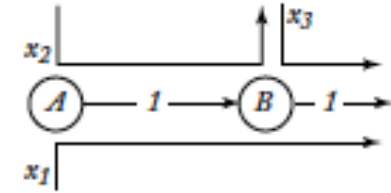
$$L(x(\lambda), \lambda^*) = \max_x L(x, \lambda)$$

Onda λ_a^* i λ_b^* minimizuju

$$L(x(\lambda), \lambda)$$

- L se naziva Lagranžeovom funkcijom a λ_a^* i λ_b^* su Lagranžeovi množioci

Kontrola zagušenja



- Primjenom prethodnog na ranije razmatranu mrežu dobija se

$$L(x, \lambda) = f(x) - \lambda_a g_a(x) - \lambda_b g_b(x) = (u(x_1) - (\lambda_a + \lambda_b)x_1) + (u(x_2) - \lambda_a x_2) + (u(x_3) - \lambda_b x_3)$$

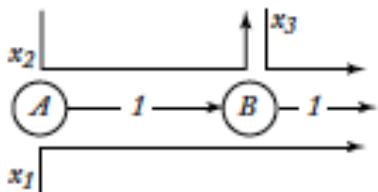
- Treba pronaći $x(\lambda)$ koja maksimizuje $L(x, \lambda)$ za fiksnu vrijednost $\lambda = (\lambda_a, \lambda_b)$
- x_1 maksimizuje $u(x_1) - (\lambda_a + \lambda_b)x_1$, x_2 maksimizuje $u(x_2) - \lambda_a x_2$, a x_3 e $u(x_3) - \lambda_b x_3$
- Fizički smisao prethodnog je da svaki link j „tarifira“ korisniku svakog toka „cijenu“ λ_j po jedinici brzine.
- Na taj način korisnik prvog toka „plaća“ $(\lambda_a + \lambda_b)x_1$ zato što tok ide preko dva linka. Zbog toga korisnik bira x_1 tako da maksimizuje utility funkciju $u(x_1) - (\lambda_a + \lambda_b)x_1$
- Slično važi i za ostala dva toka
- Ključno zapažanje je da je maksimizacija L podijeljena na maksimizaciju za svakog korisnika!
- Uparivanje varijabli x u originalni problem je urađeno zbog uslova ograničenja.
- Maksimizacija L nije uslovljena ograničenjima i rastavljena je na posebne probleme za svaku varijablu.
- Prethodna dekompozicija se dešava jer su uslovna ograničenja linearna u tokovima, tako da se varijabla x_i pojavljuje u različitim izrazima sume L .
- Takođe, treba primijetiti da je cijena koji svaki link „tarifira“ za svaki od svojih tokova ista.

Kontrola zagušenja

- Da bi se pronašli množitelji λ^* koristi se gradijentni algoritam za minimizaciju $L(x, \lambda)$

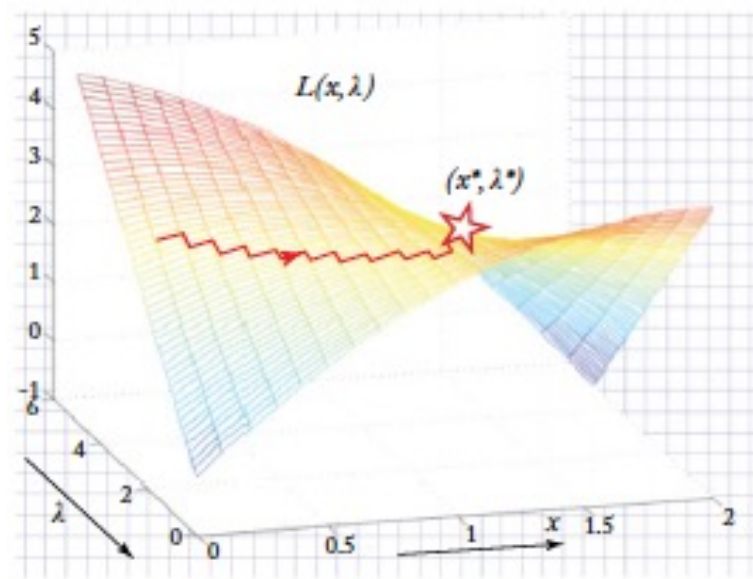
$$\lambda_j(n+1) = \lambda_j(n) - \beta \frac{d}{d\lambda_j} L(x(\lambda), \lambda) = \lambda_j(n) + \beta g_j(x)$$

- Gdje je x vektor brzina u koraku n a β je parametar koji kontroliše korak algoritma
- U svakom koraku n algoritam podešava λ u smjeru suprotnom od gradijenta funkcije $L(x(\lambda), \lambda)$ koji predstavlja najstrmiji pad funkcije.
- Za novu vrijednost cijene korisnici podešavaju svoje brzine tako da se približavaju $x(\lambda)$ za novu cijenu.
- To znači da korisnik 1 u koraku n određuje svoju brzinu na sledeći način
$$x_1(n) \text{ maksimizuje } u(x_1(n)) - (\lambda_a(n) + \lambda_b(n))x_1(n)$$
- Slično važi za $x_2(n)$ i $x_3(n)$



Kontrola zagušenja

- ❑ Slika ilustruje funkcionisanje gradijentnog algoritma
- ❑ Vidi se da algoritam podešava x u pravcu gradijenta funkcije $L(x(\lambda), \lambda)$ po x , zatim λ u pravcu koji je suprotan gradijentu funkcije $L(x(\lambda), \lambda)$ po λ
- ❑ Algoritam traži tačku „sedla“ koja je označena na slici
- ❑ Tačka „sedla“ (x^*, λ^*) je minimum funkcije $L(x(\lambda), \lambda)$ po λ od maksimuma funkcije $L(x(\lambda), \lambda)$ po x .



Kontrola zagušenja

- ❑ Radi ilustracije kako čvorišta izračunavaju cijenu, neka je $q_j(t)$ popunjenost reda čekanja na čvoru j u trenutku t
- ❑ Ovaj red čekanja se povećava brzinom koja je jednaka razlici između ukupne dolazne brzine i brzine posluživanja.

- ❑ U kratkom intervalu vremena τ

$$q_j((n+1)\tau) = q_j(n\tau) + \tau g_j(x)$$

- ❑ Gdje je x vektor brzina u trenutku $n\tau$
- ❑ Ako se napravi poređenje prethodne relacije sa relacijom

$$\lambda_j(n+1) = \lambda_j(n) - \beta \frac{d}{d\lambda_j} L(x(\lambda), \lambda) = \lambda_j(n) + \beta g_j(x)$$

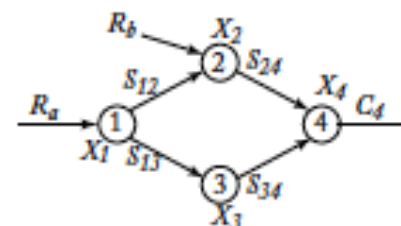
- ❑ za $\lambda_j(n+1)$ može se zaključiti da ako se gradijentni algoritam izvršava svakih τ sekundi onda

$$\lambda_j \approx \left(\frac{\beta}{\tau}\right) q_j$$

- ❑ Prema ovom izrazu trebalo bi uzimati da je cijena linka proporcionalna zauzetosti reda čekanja.
- ❑ To znači da kada se red čekanja puni link bi trebao da bude „skuplji“ što korisnike prisiljava da smanje brzinu. Jasno je da važi slično i za pražnjenje reda čekanja.
- ❑ TCP kontrola zagušenja radi sličnu stvar pri čemu je razlika u tome što mnogo zavisi od RTT i broja tokova koji dijele link

Dinamičko rutiranje i kontrola zagušenja

- Na slici je prikazana mreža sa 4 čvora čija je trenutna popunjenost bafera X_1, X_2, X_3 i X_4
- Dva toka se prenose mrežom i imaju brzine R_a i R_b .
- S_{ij} predstavlja brzinu kojom čvor i šalje podatke čvoru j
- C_i predstavlja ukupni prenosni kapacitet čvora i
- Suma brzina tokova koji izlaze iz čvora i mora biti manja od C_i



- Cilj je maksimizovati sumu utility funkcija

$$u_a(R_a) + u_b(R_b)$$

- Pri čemu treba spriječiti nagli porast popunjenosti bafera
- Funkcije u_a i u_b su pozitivne i konkavne što znači da korisnici imaju veću korist od porasta brzine ali se ta korist smanjuje sa tim povećanjem
- Neka je

$$u_j(x) = k_j \frac{1}{1 - \alpha} x^{1-\alpha}, j = a, b, \alpha > 0, \alpha \neq 1, k_j > 0$$

- Da bi se postigao zacrtani cilj treba maksimizovati brzine $R_a, R_b, S_{12}, S_{13}, S_{24}$ i S_{34} za neko $\beta > 0$.

$$\phi := \beta [u_a(R_a) + u_b(R_b)] - \frac{d}{dt} \left[\frac{1}{2} \sum_{i=1}^4 X_i^2(t) \right]$$

- Kako bi se maksimizovao ovaj izraz treba odabrati brzine koje obezbjeđuju veliku vrijednost $u_a(R_a) + u_b(R_b)$ i veliko smanjenje sume kvadrata zauzetosti bafera
- Parametar β određuje kompromis između velike utility funkcije i velike zauzetosti bafera. Za veliko β više se težira utility funkcija od zauzeća.

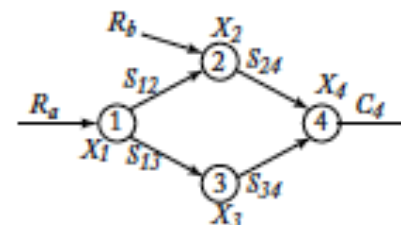
Dinamičko rutiranje i kontrola zagušenja

$$\frac{d}{dt} \left[\frac{1}{2} X_2^2(t) \right] = X_2(t) \frac{d}{dt} X_2(t) = X_2(t) [R_b + S_{12} - S_{24}]$$

- Zato što je brzina promjene brzine $X_2(t)$ jednaka razlici dolazne brzine $R_b + S_{12}$ i odlazne brzine S_{24}

- Slično važi i sa ostale čvorove tako da se može izvesti da je

$$\begin{aligned} \phi &= \\ &= \beta [u_a(R_a) + u_b(R_b)] - X_1(t) [R_a - S_{12} - S_{13}] - X_2(t) [R_b + S_{12} - S_{24}] \\ &\quad - X_3(t) [S_{13} - S_{34}] - X_4(t) [S_{24} + S_{34} - C_4] \end{aligned}$$



$$\begin{aligned} \phi &= \\ &= [\beta u_a(R_a) - X_1(t) R_a] + [\beta u_b(R_b) - X_2(t) R_b] + S_{12} [X_1(t) - X_2(t)] + S_{13} [X_1(t) - X_3(t)] \\ &\quad + S_{24} [X_2(t) - X_4(t)] + S_{34} [X_3(t) - X_4(t)] + C_4 X_4(t) \end{aligned}$$

- Maksimizacija je laka jer različiti izrazi uključuju različite varijable, pri tome poslednji član sume ne sadrži nijednu varijablu za odlučivanje

Dinamičko rutiranje i kontrola zagušenja

- Prema tome

R_a maksimizuje $\beta u_a(R_a) - X_1(t)R_a$

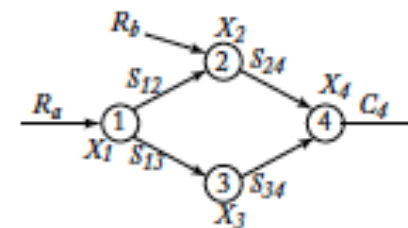
R_b maksimizuje $\beta u_b(R_b) - X_2(t)R_b$

$S_{12} = C_1$ ako je $X_1(t) > X_2(t)$ i $X_2(t) < X_3(t)$

$S_{13} = C_1$ ako je $X_1(t) > X_3(t)$ i $X_3(t) < X_2(t)$

$S_{24} = C_2$ ako je $X_2(t) > X_4(t)$

$S_{34} = C_3$ ako je $X_3(t) > X_4(t)$

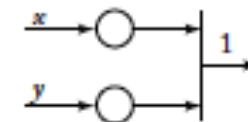


- Izvor a (b) podešava svoju brzinu R_a (R_b) na bazi zauzetosti bafera $X_1(t)$ ($X_2(t)$) na čvoru 1 (2)
- Konkavnost funkcije u_j utiče da se R_a (R_b) smanjuje sa $X_1(t)$ ($X_2(t)$)
- Ako se uzme ranije definisana utility funkcija, može se pokazati da važi

$$R_a = \left[\frac{k_a}{X_1(t)} \right]^{\frac{1}{\alpha}} \text{ i } R_b = \left[\frac{k_b}{X_2(t)} \right]^{\frac{1}{\alpha}}$$

- Primjećuje se da čvor 1 šalje pakete sledećem čvoru (2 ili 3) koji ima manju zauzetost obezbeđujući da je ona manja od $X_1(t)$ što ako nije ispunjeno nalaže čvoru 1 da prestane sa slanjem
- Čvor 2 i čvor 3 šalju podatke sve dok njihova zauzetost bafera veća od zauzetosti bafera u čvoru 4
- Pošto brzine R_a i R_b teže nuli kada $X_1(t)$ i $X_2(t)$ rastu, jasno je da će mreža biti stabilna
- U realnim mrežama postoji kašnjenje u informisanju stanja bafera i čvorovi ne šalju bite nego pakete, što ne utiče na veliko odstupanje stvarnih rezultata od modela
- Ovaj mehanizam kontrole zagušenja se naziva *backpressure* mehanizmom i lako se implementira jer je algoritmu za odlučivanje dovoljna informacija o popunjenosti bafera njegovih susjednih čvorova kojima prosleđuje saobraćaj

Modelovanje bežičnog čvorišta



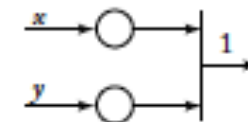
- U žičnim mrežama link „vidi“ zauzetost svog bafera
- U bežičnim mrežama ako dva čvora dijele isti kanal oni nijesu „svjesni“ o ukupnom broju paketa koji čekaju na prenos preko linka
- Neka dva čvora dijele isti kanal i neka je utility funkcija za tok 1 $\log(x)$ a za tok 2 je $2\log(y)$
- Optimizacioni problem je

$$\text{Maksimizovati } f(x, y) = \log(x) + 2\log(y) \\ \text{uz ograničenje da je } x + y \leq 1$$

- Ako se koristi već opisani dualni algoritam može se zaključiti da bi čvorovi trebali da podešavaju x i y na bazi ukupnog broja paketa koji čekaju na prenos preko linka. Međutim čvorovi nemaju tu informaciju.
- Može se uočiti i pristup po kome bi čvorovi razmjenjivali informacije o zauzetostima svojih bafera što je veoma kompleksno posebno u velikim mrežama
- Pristup koji je realan je da se za svaki čvor prati procenat vremena p_1 , odnosno p_2 , tokom kojeg koristi link. U tom slučaju

$$\text{Maksimizovati } f(x, y) = \log(x) + 2\log(y) \\ \text{uz ograničenje da je } x \leq p_1 \\ \text{uz ograničenje da je } y \leq p_2 \\ \text{uz ograničenje da je } p_1 + p_2 \leq 1$$

Modelovanje bežičnog čvorišta



- Da bi se odredio ovaj distribuirani algoritam uvodi se sledeća modifikacija

$$\text{Maksimizovati } f(x, y) = \log(x) + 2\log(y) + \beta\phi(p_1) + \beta\phi(p_2)$$

uz ograničenje da je $x \leq p_1$

uz ograničenje da je $y \leq p_2$

uz ograničenje da je $p_1 + p_2 \leq 1$

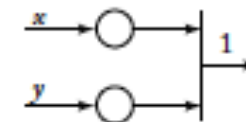
- Gdje je $\phi(z)$ ograničena rastuća konkavna funkcija
- Ako se uzme da je β malo onda je to veoma mala modifikacija prethodne jednačine
- Ako se dvije nejednačine ograničenja zamijene *penalty* funkcijama dobija se sledeći cilj optimizacije

$$\text{Maksimizovati } h(x, y, p_1, p_2; \lambda_1, \lambda_2)$$

$$= \log(x) + 2\log(y) + \beta\phi(p_1) + \beta\phi(p_2) - \lambda_1(x - p_1) - \lambda_2(y - p_2)$$

- Uslov normalizovanosti se uzima za kasnije

Modelovanje bežičnog čvorišta



- Ako se maksimizuje funkcija h po x, y, p_1, p_2 a minimizuje po λ_1, λ_2 gradijentnim algoritmom dobija se

$$\frac{1}{x} = \lambda_1$$

$$\frac{2}{y} = \lambda_2$$

$$\beta \phi'(p_1) + \lambda_1 = 0$$

$$\beta \phi'(p_2) + \lambda_2 = 0$$

$$\lambda_1(n+1) = \lambda_1(n) + \gamma(x(n) - p_1(n))$$

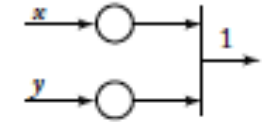
$$\lambda_2(n+1) = \lambda_2(n) + \gamma(y(n) - p_2(n))$$

- Kao i ranije poslednje dvije jednačine pokazuju da su λ_1 i λ_2 proporcionalne zauzetosti bafera na čvorovima 1 i 2, respektivno.
- Prve dvije jednačine objašnjavaju vezu kontrole toka i zauzetosti bafera
- Treća i četvrta jednačina se mogu transformisati u

$$p_1 = \psi\left(\frac{\lambda_1}{\beta}\right) \text{ i } p_2 = \psi\left(\frac{\lambda_2}{\beta}\right), \text{ pri čemu je } p_1 + p_2 = 1$$

- ψ je izvod funkcije ϕ . Postoji mnogo funkcija koje zadovoljavaju prethodno jednačine i potrebu da p_1 raste sa povećanjem zauzetosti bafera na čvoru 1, odnosno da p_2 raste sa rastom zauzetosti bafera na čvoru 2.

Modelovanje bežičnog čvorišta



- Jedno rješenje za pronalaženje p_1 i p_2 može da bude CSMA.
- Neka čvor i pokušava da pristupi slobodnom kanalu sa vjerovatnoćom a_i tokom kratkog slot-a u kome se takmiči sa ostalim čvorovima.
- Ako je a_i malo tako da se $a_1 a_2$ može zanemariti, jasno je da čvor i zauzima kanal sa vjerovatnoćom $\frac{a_i}{a_1 + a_2}$
- Prema tome, a_i bi trebalo da raste sa povećanjem broja paketa u baferu čvora i .
- CSMA se brine da je $p_1 + p_2 = 1$
- Na kraju algoritam bi trebao da izgleda ovako:
 - Svaki čvor pokušava da šalje sa vjerovatnoćom koja raste sa rastom zauzetosti njegovog bafera
 - Svaki čvor prima novi paket brzinom koja se smanjuje sa rastom zauzetosti njegovog bafera
- Ako se uzme da je
- Dobija se

$$a_1 = e^{2Q(1,n)} \text{ i } a_2 = e^{2Q(2,n)}$$

