

konzistentný broadcast

- ak dobrý proces i poslal (m, i, r) v kroku r , dobrí ju akceptujú najneskôr v $r + 1$
- ak dobrý proces i neposlal (m, i, r) v kroku r , nikto dobrý ju neakceptuje
- ak je správa (m, i, r) akceptovaná dobrým j v r' , najneskôr v $r' + 1$ ju akc. všetci dobrí

algoritmus

- i pošle $(init, m, i, r)$ v kole r
- ak dobrý dostane $(init, m, i, r)$ v kole r , pošle $(echo, m, i, r)$ všetkým dobrým v kole $r + 1$
- ak pred kolom $r' \geq r + 2$ dostane dobrý od $f + 1$ echo, pošle $(echo, m, i, r)$ v r'
- ak dostal echo od $n - f$, akceptuje

dohoda

- dvojkrovové fázy
- v prvom kole bcastujú všetci s 1
- v kole $2s - 1$ pošlú tí, čo akceptovali od $f + s - 1$ a ešte nebcastovali
- ak po $2(f + 1)$ kolách i akceptoval od $2f + 1$ procesov, tak 1, inak 0

vzájomné vylúčenie

Ricart-Agrawala

- logické hodiny T_i
- request \approx poslať všetkým (T_i, i)
- čakať na odpoveď od všetkých
- neodpovedá, ak:
 - je v CS
 - žiada prístup s vyššou prioritou (nižším časom)
- pri opustení CS pošle všetky odpovede

zmenšiť počet správ - token

- pre každý proces mám jeho posledný request
- pre vstup: pošle všetkým request a čaká na token
- pri opustení: zisti, kto čaká na token (v tokene je tabuľka posledných držaní)

Dijkstra: self-stabilizing token-ring

- procesy \mathbb{Z}_n , 0 je špeciálny
- pre 0: má token, ak $x_0 = x_{n-1}$, pri opustení CS $x_0 := (x_0 + 1) \bmod n$
- pre ostatných: má token, ak $x_i \neq x_{i-1}$, pri opustení CS $x_i := x_{i-1}$

detekcia terminácie

- pasívne vrcholy = čakajú na prijatie správy
- terminácia = všetky vrcholy sú pasívne
- základný vs. riadiaci algoritmus

Dijkstra-Scholten

- základný algoritmus má jedného iniciátora
- udržuje sa strom výpočtu: vnútorné vrcholy sú procesy, listy sú správy
- poslať správu: $sc_p := sc_p + 1$
- prijať správu od q : ak nie som v strome, $father_p = q$, inak send **sig** to $father_p$
- prijať **sig**: $sc_p := sc_p - 1$
ak $sc_p = 0$ a som pasívny: send **sig** to $father_p$ a vypoť sa
- prechod do pasívneho stavu:
ak $sc_p = 0$, send **sig** to $father_p$ a vypoť sa

- $S = \sum sc_p$: S rastie pri poslanej správe a klesá pri doručení **sig**, $S \geq 0$
- ak alg. terminuje, posielajú sa iba **sig**, S klesá \Rightarrow terminálna konfigurácia
- strom výpočtu je prázdny

počet správ je rovnaký ako v zákl. algoritme

lepšie to nejde

pre každý TD algoritmus a každé W existuje základný výpočet s W správami, kde treba W radiacích správ

- p a q sú aktívne (poslať jednu správu z p)
- obidva sa stanú pasívne \Rightarrow musí sa poslať radiaca správa, nech p
- q ostane aktívny, pošle správu a zaktívni p
- základný terminuje \Rightarrow radiaca správa

zovšeobecniť na viac iniciátorov:
les, keď skolabuje strom, ostane prázdny, wave