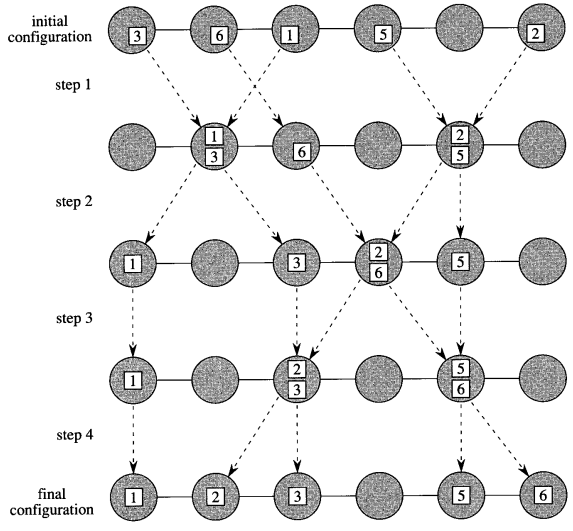
packet routing

- synchrónny režim
- vrcholy majú pakety (uložené v bufferoch)
- v jednom kroku po jednej linke ide max. jeden paket
- algoritmus = odchádzajúce linky + priorita bufferov
- celkový čas



packet routing na mriežke $\sqrt{N} \times \sqrt{N}$

Každý vrchol má 1 paket, do každého smeruje 1 paket (permutation routing)

Najprv riadok, potom stĺpec. Prednosť má ten s najdlhšou cestou.

analýza: stačí $2\sqrt{N} - 2$ krokov

- po $\sqrt{N} - 1$ krokoch je každý v správnom stĺpci (nebrzdia sa)
- routovanie v stĺpci ide v $\sqrt{N} - 1$ krokoch
  - pre každé i platí: po $N - 1$ krokoch sú koncové pakety na koncových miestach
  - dôvod: zdržujú sa iba navzájom

veľkosť buffra v najhoršom prípade: $2/3\sqrt{N} - 3$

veľkosť buffra: priemerný prípad I

setting

Každý vrchol má jeden paket s **náhodným cieľom**

max. veľkosť buffra $\approx$ počet zahnutí vo vrchole

psť, že aspoň r zahne $\leq \binom{\sqrt{N}}{r} \left(\frac{1}{\sqrt{N}}\right)^r < \left(\frac{e}{r}\right)^r$

pre $r = \frac{e \log N}{\log \log N}$ je psť $o(N^{-2})$

veľkosť buffra: priemerný prípad II

wide-channel: nepredbiehajú sa

lema

psť, že vo wch prejde aspoň $\alpha\Delta/2$ paketov cez hranu e počas $t+1, t+2, \ldots, t+\Delta$ je najviac $e^{(\alpha-1-\alpha \ln \alpha)\Delta/2}$

očakávaný počet paketov na hrane $(i,j) \mapsto (i+1,j)$ je

$$\frac{2i(\sqrt{N}-i)\Delta}{N} \leq \frac{\Delta}{2}$$

chceme ukázať, že s veľkou psťou ich neprejde príliš viac

Černovov odhad

lema

Majme n nezávislých Bernoulihho náh. prem. $X_1, \ldots, X_n$, pričom $Pr[X_k = 1] \leq P_k$. Potom

$$Pr[X \geq \beta P] \leq e^{(1-\frac{1}{\beta}-\ln \beta)\beta P}$$

kde $X = \sum X_i$, $P = \sum P_i$

$$E[e^{\lambda X_k}] \leq 1 + P_k(e^{\lambda} - 1) \leq e^{P_k(e^{\lambda}-1)}$$

$$E[e^{\lambda X}] \leq e^{P(e^{\lambda}-1)}$$

$$P[e^{\lambda X} \geq e^{\lambda \beta P}] \leq \frac{E[e^{\lambda X}]}{e^{\lambda \beta P}} \leq e^{P(e^{\lambda}-1)-\lambda \beta P}$$

veľkosť buffra: priemerný prípad II

lema

Majme n nezávislých Bernoulihho náh. prem. $X_1, \ldots, X_n$, pričom $Pr[X_k = 1] \leq P_k$. Potom $Pr[X \geq \beta P] \leq e^{(1-\frac{1}{\beta}-\ln\beta)\beta P}$ kde $X = \sum X_i$, $P = \sum P_i$

lema

psť, že vo wch prejde aspoň $\alpha\Delta/2$ paketov cez hranu e počas $t+1, t+2, \ldots, t+\Delta$ je najviac $e^{(\alpha-1-\alpha\ln\alpha)\Delta/2}$

očakávaný počet paketov na hrane $(i,j) \mapsto (i+1,j)$ je

$$\frac{2i(\sqrt{N}-i)\Delta}{N} \leq \frac{\Delta}{2}$$

chceme ukázať, že s veľkou psťou ich neprejde príliš viac $n = 2i\Delta$, $P_k = \frac{\sqrt{N}-i}{N}$, $P = \frac{2i(\sqrt{N}-i)\Delta}{N}$, $\beta = \frac{\alpha N}{4i(\sqrt{N}-i)}$

veľkosť buffra: priemerný prípad II

lema

ak je paket vo vzd. d od hrany e v čase T, a p prejde cez e v čase $T + d + \delta$, potom v každom kroku $[T + d, T + d + \delta]$ prejde paket cez e

dosledok

ak paket prejde cez e v čase T vo wch, a prejde cez e v čase $T + \delta$ v št. , tak v každom kroku $[T, T + \delta]$ prejde paket

lema

ak počas $[T + 1, T + \Delta]$ prejde cez e x paketov v št., tak pre nejaké t prejde $x + t$ paketov cez e v čase $[T + 1 - t, T + \Delta]$ vo wch.

lema

psť, že cez e prejde viac ako $\alpha\Delta/2$ paketov počas konkrétneho okna Δ krokov je najviac $O(e^{(\alpha-1-\alpha \ln \alpha)\Delta/2})$

dynamické routovanie

model

V každom kroku sa v každom vrchole s psťou λ narodí paket s náhodným cieľom.

stabilita

Pre $\lambda \geq 4/\sqrt{N}$ je systém nestabilný

veta

Ak je $\lambda \leq 0.99\frac{4}{\sqrt{N}}$, tak psť zdržania konkrétneho paketu o Δ krokov je $e^{-O(\Delta)}$.
W.h.p. stačí buffer $O(1 + \frac{\log T}{\log N})$.

hierarchické routovanie

cieľ: minimalizovať počet rozhodnutí

veta

Pre sieť s N vrcholmi stačí $O(\sqrt{N})$ rozhodnutí pri použití 3 farieb.

s-klastre:

- každý je súvislý, pokrývajú všetky vrcholy
- každý obsahuje aspoň s vrcholov a má polomer najviac $2s$

kostra spájajúca centrá klastrov: m listov $\Rightarrow$ $m - 2$ vetvení

hierarchické routovanie

veta

Pre sieť s N a pre $f \leq \log N$ stačí $O(f \cdot N^{1/f})$ rozhodnutí a $2f + 1$ farieb

po i klastrovaniach s parametrom s: m_i listov, max. $m_i - 2$ vetvení $\Rightarrow$
$m_{i+1} = m_i(2/s)$
$m_f + fs$ rozhodnutí
$s \approx 2N^{1/f}$

cvičenia

mriežka $\sqrt{n} \times \sqrt{n}$, prednosť má hocikto; ukážte, že v najhoršom prípade treba viac ako $2\sqrt{n}$ krokov, ale stačí $O(\sqrt{n})$

mriežka $\sqrt{n} \times \sqrt{n}$, v každom vrchole správa do náhodného. Ukážte, že w.h.p. do žiadnoho vrchola nesmeruje viac ako $3 \log n / \log \log n$ správ.

majme cestu z n procesorov, každý chce routovať práve dva pakety (červený a modrý), pričom červené aj modré tvoria permutáciu. ukážte, že stačí n krokov

odolnosť voči chybám – strata správ

Problém dohody

- synchrónny systém
- známe identifikátory
- každý má na vstupe 0/1
- správy sa môžu strácať
- každý proces sa musí rozhodnúť
- treba zaručiť
  - **Dohoda:** všetky procesy sa rozhodnú na tú istú hodnotu
  - **Terminácia:** každý proces sa rozhodne v konečnom čase
  - **Netrivialita:**
    1. Ak všetci začnú s hodnotou 0, musia sa dohodnúť na 0.
    2. Ak všetci začnú s hodnotou 1 a správy sa nestrácajú, musia sa dohodnúť na 1.

neexistuje deterministické riešenie

- 2 vrcholy, 1 linka
- sporom, nech existuje a trvá r kôl
- výpočet, kde začnú obaja s hodnotou 1 a nestrácajú sa správy
- dohodnú sa na 1
- stratí sa posledná správa, jeden z nich to nezistí
- výpočet, kde neprejde ani jedna správa a dohodnú sa na 1
- jeden z nich dostane na vstup 0
- aj druhý

randomizované riešenie (úplný graf)

komunikačný pattern

zoznam trojíc (i, j, t): v čase t sa nestratí správa z $i \mapsto j$

(fixný) adversary = vstup a komunikačný pattern
Dohoda: $Pr[$ nejaké dva procesy sa rozhodnú na rôznu hodnotu$] \leq \varepsilon$

algoritmus s $\varepsilon = 1/r$

daný adversary γ: dvojice (i, t), kde i-procesor, t-čas majme usporiadanie:

1. $(i, t) \leq_\gamma (i, t')$, kde $t \leq t'$
2. ak $(i, j, t) \in \gamma$, tak $(i, t-1) \leq_\gamma (j, t)$
3. tranzitivita

úroveň informovanosti

1. $level_\gamma(i, 0) = 0$
2. ak $t > 0$ a existuje $j \neq i$ také, že $(j, 0) \not\leq_\gamma (i, t)$, tak $level_\gamma(i, t) = 0$
3. nech l_j je $\max\{level_\gamma(j, t') \mid (j, t') \leq_\gamma (i, t)\}$
   potom $level_\gamma(i, t) = 1 + \min\{l_j \mid j \neq i\}$

algoritmus

- prvý proces vygeneruje náhodný kľúč
- procesy si počítajú level
- rozhodnutie 1, ak všetci majú 1 a môj level je aspoň kľúč

$rounds := rounds + 1$
let (L_j, V_j, k_j) be the message from j, for each j from which a message arrives
if some $k_j \neq undefined$ then $key := k_j$
for all $j \neq i$ do
    if some $V_{i'}(j) \neq undefined$ then $val(j) := V_{i'}(j)$
    if some $L_{i'}(j) > level(j)$ then $level(j) := \max\{L_{i'}(j)\}$
$level(i) := 1 + \min\{level(j) : j \neq i\}$
if $rounds = r$ then
    if $key \neq undefined$ and $level(i) \geq key$ and $val(j) = 1$ for all j then
        $decision := 1$
    else $decision := 0$

dôkaz

- **Dohoda:** $Pr[$ nejaké dva procesy sa rozhodnú na rôznu hodnotu$] \leq \varepsilon$
- **Terminácia:** každý proces sa rozhodne v konečnom čase
- **Netrivialita:**
  1. Ak všetci začnú s hodnotou 0, musia sa dohodnúť na 0.
  2. Ak všetci začnú s hodnotou 1 a správy sa nestrácajú, musia sa dohodnúť na 1.

terminácia a netrivialita sú zrejmé
pre fixný pattern, aká je pravdepodobnosť nezhody?
levely sa líšia max o 1, preto jediný problém je ak $key = \max\{l_i\}$

dolný odhad

ľubovoľný r-kolový algoritmus má pravdepodobnosť nezhody aspoň $\frac{1}{r+1}$

orez

pre adversary B s patternom γ a proces i, $B' = \mathit{prune}(B, i)$

1. ak $(j, 0) \leq_\gamma (i, r)$ tak sa vstup j zachová, inak znuluje
2. trojica (j, j', t) je v kom. patterne B', akk je v γ a $(j', t) \leq_\gamma (i, r)$

$P^B[i \text{ sa rozhodne 1}] = P^{\mathit{prune}(B,i)}[i \text{ sa rozhodne 1}]$

lema

Ak majú na vstupe všetci 1, $P[i \text{ sa rozhodne 1}] \leq \varepsilon(\mathit{level}(i, r) + 1)$

lema

Ak majú na vstupe všetci 1, $P[i \text{ sa rozhodne 1}] \leq \varepsilon(level(i,r)+1)$

- indukcia na $level(i,r)$: nech $level(i,r) = 0$:
- $B' = prune(B,i) = prune(B',i)$
- $P^B[i \text{ sa rozhodne 1}] = P^{B'}[i \text{ sa rozhodne 1}]$
- od j-čka neprišla správa, $B'' = prune(B',j) = prune(B'',j)$ je triviálny adversary
- $P^{B'}[j \text{ sa rozhodne 1}] = P^{B''}[j \text{ sa rozhodne 1}]$
- lenže $P^{B''}[j \text{ sa rozhodne 1}] = 0$, takže $P^{B'}[j \text{ sa rozhodne 1}] = 0$
- psť nezhody je ε, $\Rightarrow |P^{B'}[i \text{ sa rozhodne 1}] - P^{B'}[j \text{ sa rozhodne 1}]| \leq \varepsilon$
- preto $P^{B'}[i \text{ sa rozhodne 1}] \leq \varepsilon$ a $P^B[i \text{ sa rozhodne 1}] \leq \varepsilon$

lema

Ak majú na vstupe všetci 1, $P[i \text{ sa rozhodne 1}] \leq \varepsilon(level(i,r)+1)$

- indukcia na $level(i,r)$: nech $level(i,r) > 0$
- $B' = prune(B,i) = prune(B',i)$
- existuje j, že $level_{B'}(j,r) \leq l-1$
- podľa i.p. $P^{B'}[j \text{ sa rozhodne 1}] \leq \varepsilon(level(j,r)+1) \leq \varepsilon l$
- psť nezhody je ε, $\Rightarrow |P^{B'}[i \text{ sa rozhodne 1}] - P^{B'}[j \text{ sa rozhodne 1}]| \leq \varepsilon$
- preto $P^{B'}[i \text{ sa rozhodne 1}] \leq \varepsilon(l+1)$ a $P^{B}[i \text{ sa rozhodne 1}] \leq \varepsilon(l+1)$

chyby procesov – stop chyby

Problém dohody

- synchrónny systém
- známe identifikátory
- každý má na vstupe 0/1
- proces môže havarovať (uprostred posielania správ)
- maximálne f havarovaných procesov
- každý proces sa musí rozhodnúť
- treba zaručiť
  - **Dohoda:** všetky procesy (ktoré nehavarovali) sa rozhodnú na tú istú hodnotu
  - **Terminácia:** každý proces (ktorý nehavaroval) sa rozhodne v konečnom čase
  - **Netrivialita:** ak všetci začnú s rovnakou hdonotou i, musia sa dohodnúť na i.

chyby procesov – stop chyby

algoritmus

flood počas $f + 1$ kôl; ak je iba jedna hodnota, rozhodni sa, inak (default) 0

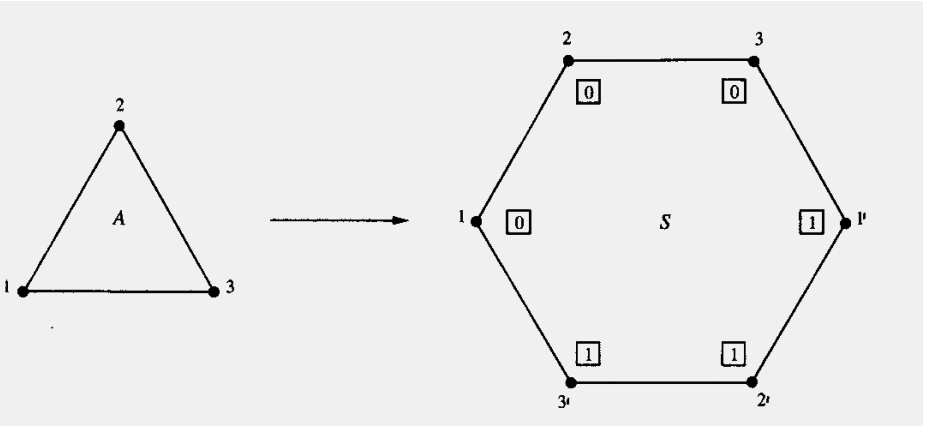
- existuje kolo, v ktorom nikto nehavaruje; potom sa udržiavajú rovnaké hodnoty
- $(f + 1)n^2$ správ
- zlepšenie: posielať iba keď sa zmení hodnota $\Rightarrow O(n^2)$ správ

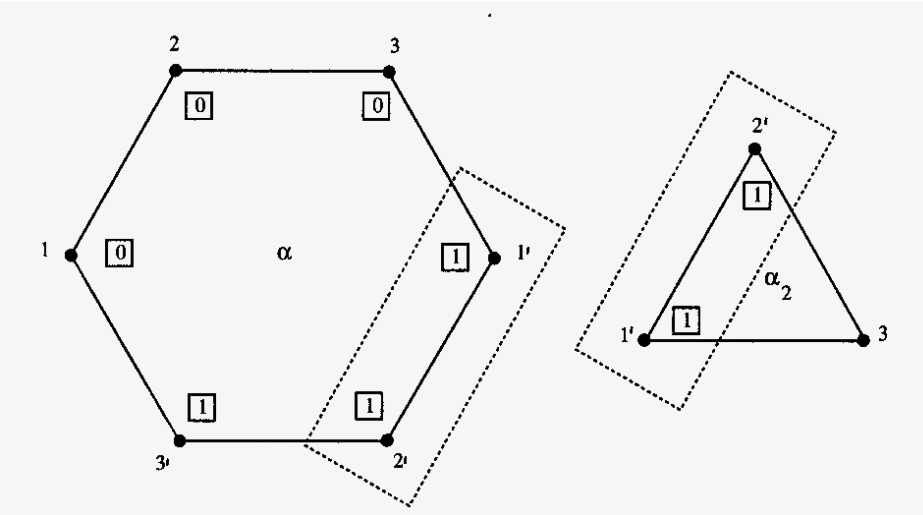
chyby procesov – byzantínske chyby

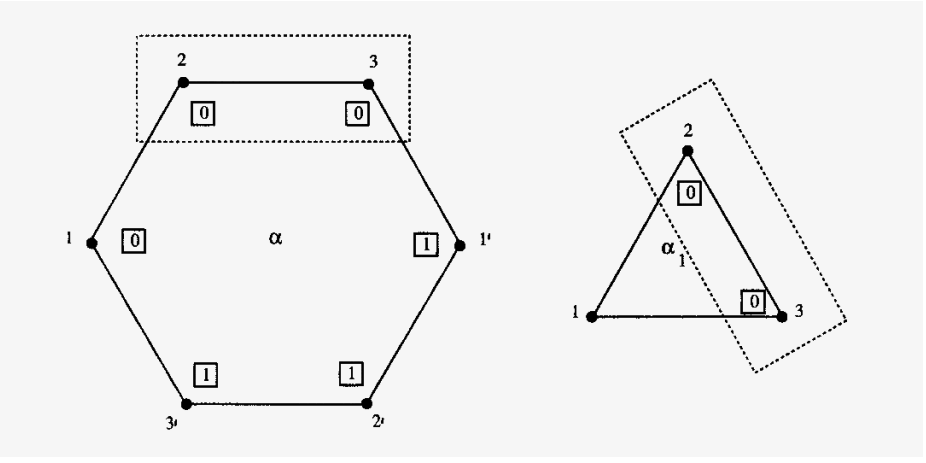
Problém dohody

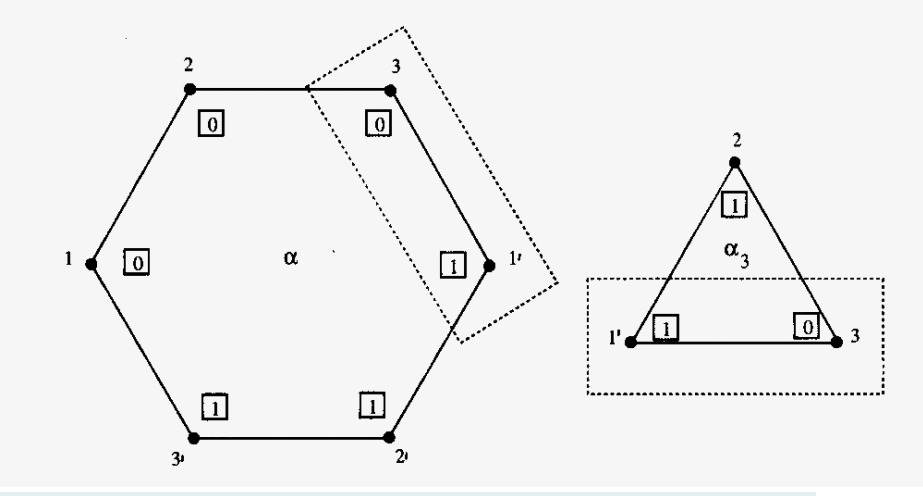
- synchrónny systém
- známe identifikátory
- každý má na vstupe 0/1
- niektoré procesy sú *zlé*
- maximálne f zlých procesov
- každý proces sa musí rozhodnúť
- treba zaručiť
  - **Dohoda:** všetky dobré procesy sa rozhodnú na tú istú hodnotu
  - **Terminácia:** každý dobrý proces sa rozhodne v konečnom čase
  - **Netrivialita:** ak všetci dobrí začnú s rovnakou hdonotou i, všetci dobrí sa musia dohodnúť na i.

dolný odhad na počet zlých: jeden zlý spomedzi troch



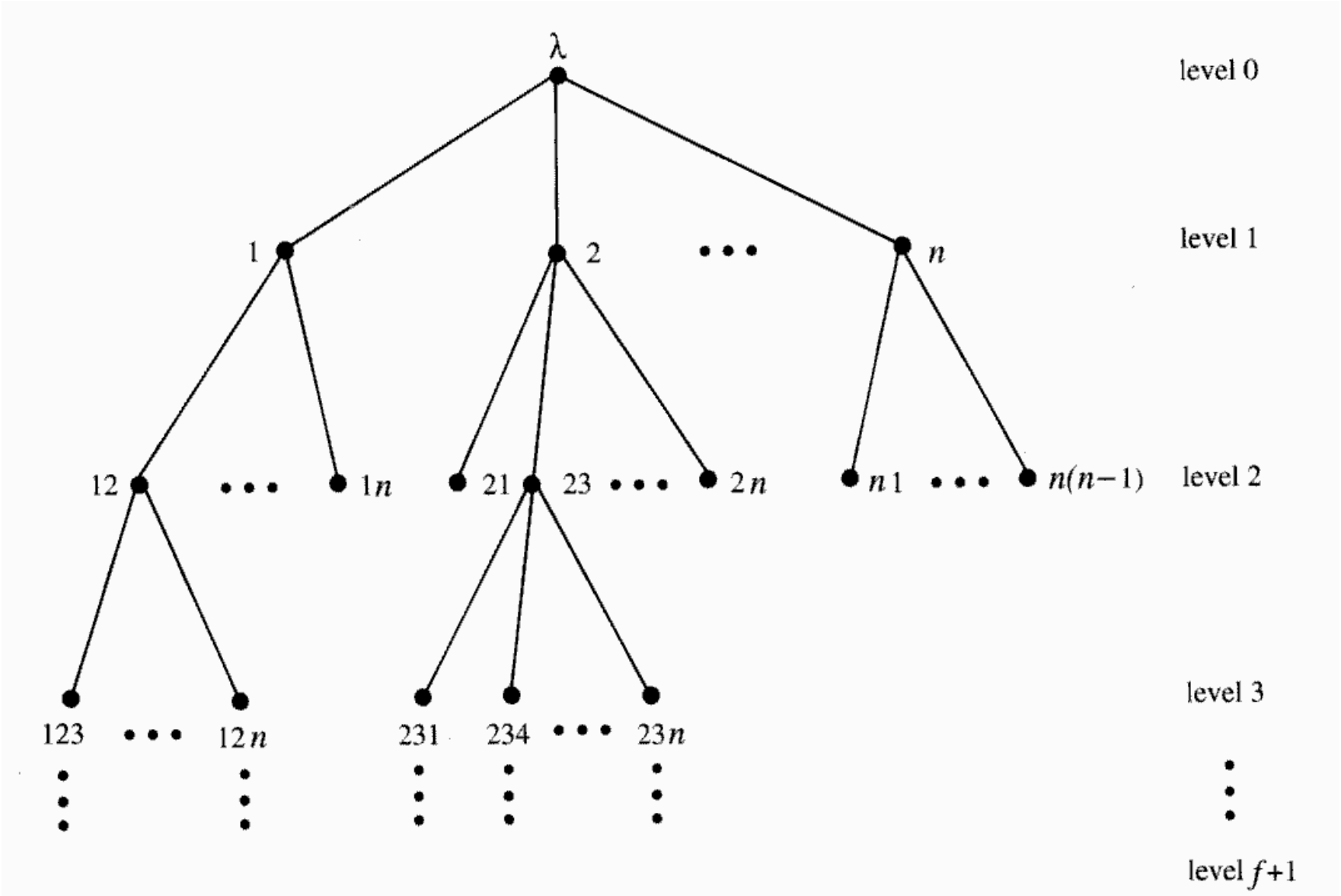






pre viac: simulácia

EIG algoritmus



newval(x): väčšina z *newval*(xj)

dôkaz

lema

Po $f + 1$ kolách platí: nech j, j, k, $i \neq j$ sú tri dobré procesy. Potom $val(xk)_i = val(xk)_j$ pre všetky x.

lema

Po $f + 1$ kolách platí: nech k je dobrý proces. Potom existuje v, že $val(xk)_i = newval(xk)_i = v$ pre všetky dobré procesy i

lema

keď všetci začnú s rovnakou hodnotou, musia sa na nej dohodnúť

dôkaz

vrchol x je *spoľahlivý*, ak všetky dobré procesy i majú po $f+1$ kolách $newval(x)_i = v$ pre nejaké v

lema

Po $f+1$ kolách je na každej ceste z koreňa do listu spoľahlivý vrchol

lema

Po $f+1$ kolách: ak existuje pokrytie podstromu vo vrchole x dobrými vrcholmi, potom x je dobrý.

polynomiálny počet správ

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

polynomiálny počet správ

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

polynomiálny počet správ

dohoda

- dvojkrokové 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

komunikačné protokoly

alternating bit

- A opakovane posiela správu so “sequence number” 0
- B začne posielať ACK0
- A prejde na sequence number 1

problém: latencia

sliding window

- A posiela naraz niekoľko správ (frames)
- keď dostane ACK, posunie “okno”

dvojsmerné vyvážené posúvanie okna

- konštanta l – veľkosť okna
- i-ty frame sa môže poslať, keď sa doručili $0..i-l$
- zároveň je to acknowledgement

- a – prvý (odoslaný) frame, na ktorý neprišiel ACK
- s – prvý neprijatý frame
- posiela sa z intervalu $a \leq i < s + l$
- pri prijatí i : $a := \max(a, i - l + 1)$

fairness

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

- 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 tabulka posledných držaní)

detekcia terminácie

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

detekcia terminácie

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 q
- prijať **sig**: $sc_p := sc_p - 1$
  ak $sc_p = 0$ a som pasívny: send **sig** to $father_p$ a vypoj sa
- prechod do pasívneho stavu:
  ak $sc_p = 0$, send **sig** to $father_p$ a vypoj sa

- $S = \sum sc_p$: S rastie pri poslanej správe a klesá pri doručenom **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

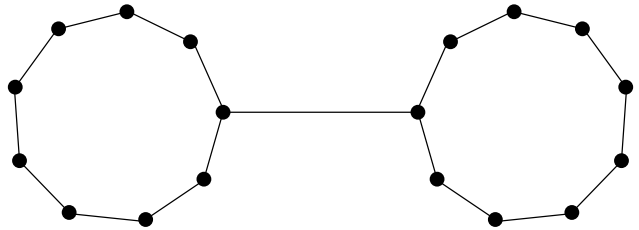
pre každý TD algoritmus a každé W existuje základný výpočet s W správami, kde treba W riadiacich správ

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

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

demo písomka

- Navrhnite algoritmus na problém dohody so stop-chybami procesorov podľa definície z prednášky (t.j. procesy majú vstup 0/1; poznajú svoje ID aj ID všetkých susedov; môžu komunikovať každý s každým; je najviac f havárií; v konečnom čase sa každý preživší rozhodne; musia sa rozhodnúť rovnako; ak majú na začiatku všetci hodnotu i, je i jediné prípustné rozhodnutie), ktorý má navyše "early stopping" vlastnosť: existuje funkcia $h(x) \geq x + 1$ taká, že ak počas daného výpočtu zhavarovalo $f' < f$ procesorov, každý preživší procesor sa rozhodol v čase $\min\{h(f'), f + 1\}$. Snažte sa dosiahnuť čo najlepšiu funkciu $h(\cdot)$.
- Uvažujme n^2, $n > 2$ procesov spojených obojsmernými linkami do torusu. Hrany sú označené tak, že tvoria zmysel pre orientáciu (t.j. procesy nepoznajú svoje súradnice, ale vedia, ktorá linka ide ktorým smerom). Procesy majú identifikátory, štartujú naraz a pracujú v asynchrónnom režime. V grafe je zvolený šéf, t.j. každý proces má logickú premennú `som_sef`, ktorá má hodnotu `true` práve v jednom procese.
  V sieti je jedna chybná linka, ktorá nikdy neprepúšťa správy: ak ľubovoľný z koncových vrcholov chybnej linky po nej pošle správu, správa sa bez akejkoľvek notifikácie stratí (takže žiaden proces nevie, či sa správa stratila, alebo je iba pomalá). Cieľom šéfa je zistiť, medzi ktorými dvomi vrcholmi vedie chybná linka.
  Navrhnite čo najefektívnejší algoritmus, pomocou ktorého šéf lokalizuje chybnú linku (t.j. zistí identifikátory vrcholov susediacich s chybnou linkou).
- Majme **synchrónny** torus rozmerov $n \times n$ so zmyslom pre orientáciu ako v predchádzajúcom príklade. Procesy majú identifikátory, ktoré sú celé čísla, poznajú n a štartujú naraz. Je možné zvoliť šéfa s komunikáciou $O(n^2)$ bitov?
- Majme synchrónnu sieť zloženú z dvoch n-vrcholových kruhov (dokopy má $2n$ vrcholov) spojených hranou takto:



V každom kroku sa v každom vrchole narodí s pravdepodobnosťou p paket, ktorý je určený do náhodného cieľa. Ukážte, že ak $p > \frac{2}{n}$, sieť nemôže byť stabilná, t.j. ak v každom kroku môže prejsť po jednej linke v jednom smere jeden paket, maximálne časy doručenia paketov budú neobmedzene rásť.