Számítógép hálózatok, osztott rendszerek

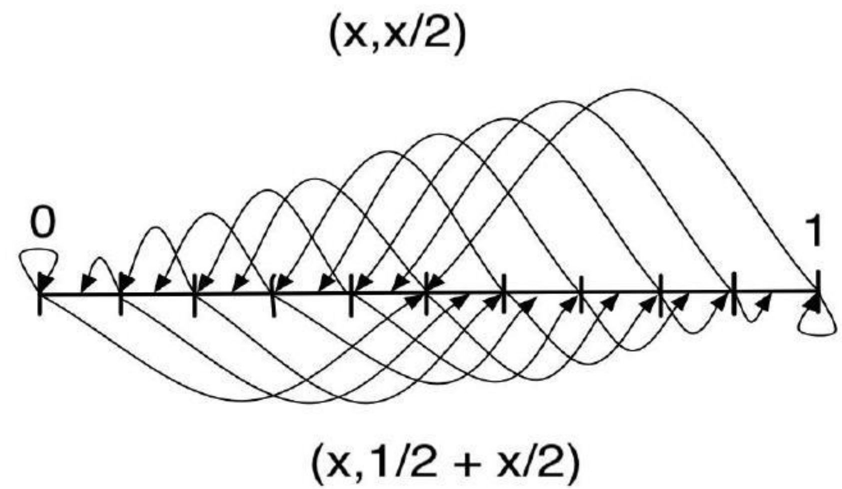
Distance Halving

Folytonos gráfok

- Végtelen gráfok folytonos csomópont halmazzal és élhalmazzal

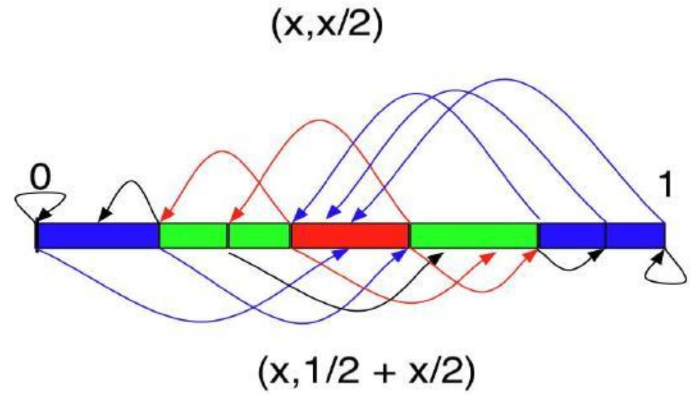
- A felhasznált gráf
  - $x \in [0,1)$
  - élek:
    - $(x,x/2)$, balra-élek
    - $(x,1/2+x/2)$, jobbra-élek

  - És a megfelelő élek az ellekező irányba
    - $(x/2,x)$
    - $(1/2+x/2,x)$

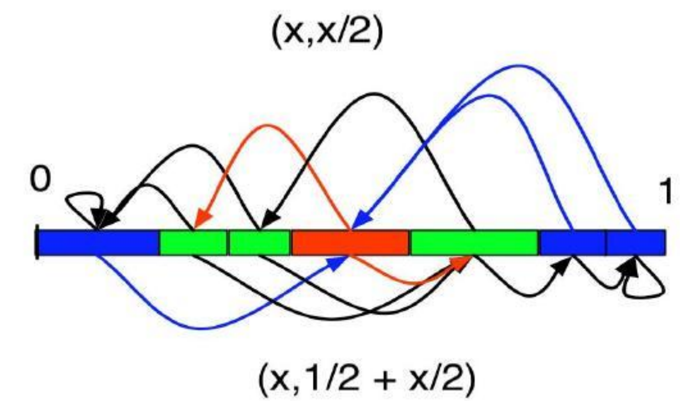


Folytonos esetről a diszkrét esetre

- Tekintsünk a [0,1) intervallum részintervallumait
- Adjunk egy élet az A és B intervallum közé,
  - ha létezik x ∈ A és y ∈ B, úgy hogy (x,y) egy él a folytonos gráfban



- A részintervallumok egymást követő felezésekkel állnak elő
- Ha a legnagyobb és a legkisebb intervallum hosszának aránya konstans, akkor a csomópontok foka konstans lesz
- Ez többszörös kiválasztás (multiple choice) által érjük el

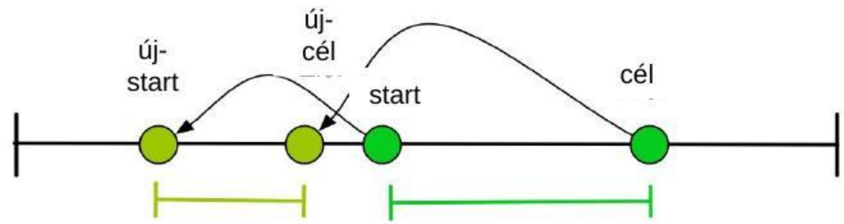


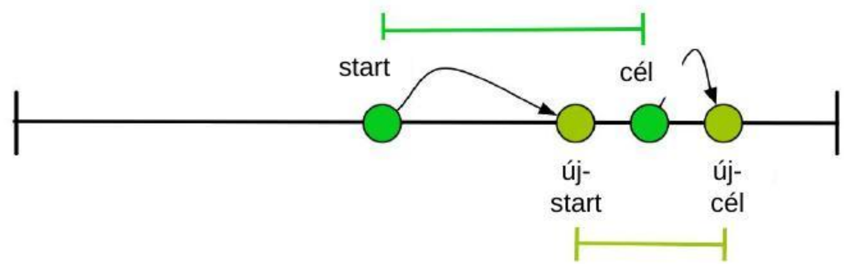
A Distance-Halving hálózat felépítése

- A Peerekhez részintervallumot rendelünk
- Szomszédos részintervallumokat egy gyűrűbe fűzzük össze
- A legnagyobb intervallum hossza nagy valószínűséggel: 2/n
- A legkisebb intervallum hossza nagy valószínűséggel: 1/(2n)
  - Ezt a többszörös kiválasztással (multiple choice) érjük el
- Ezáltal a be- és ki-fok konstans lesz
- Az átmérő logaritmikus

Keresés a Distance-Halving hálózatban

- **left-routing**(start,cél)
  - **if** start és cél szomszédos **then**
    - küldjük az üzenetet a start-tól a cél-nak
  - **else**
    - új-start ← bal-mutató(start)
    - új-cél ← bal-mutató(cél)
    - küldjük az üzenetet a start-tól az új start-nak
    - **left-routing**(új-start,új-cél)
    - Küldjük az üzenetet az új-céltól a cél-nak

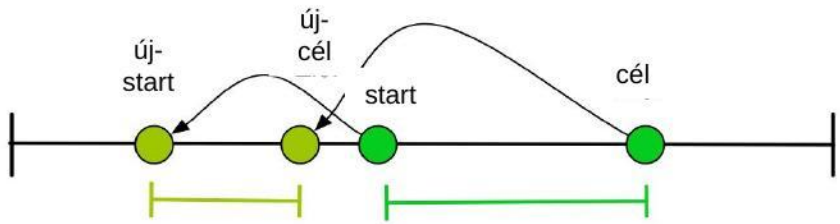




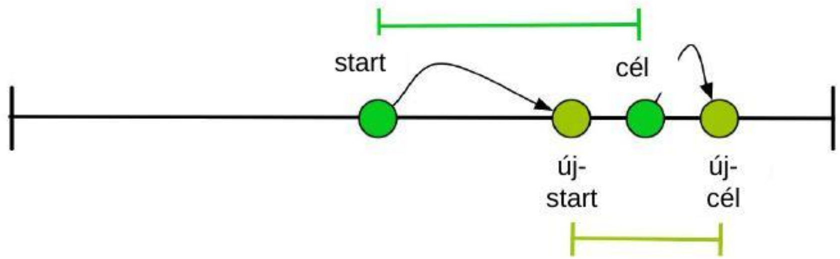
- **right-routing**(start, cél)
  - …

Keresés a Distance-Halving hálózatban

- A bal-éleket követve a start-cél távolság minden lépés után feleződik
- 2+log n lépésen keresztül követjük a bal-éleket
- Ezután az aktuális cél egy lépésben elérhető
- Végül a bal-éleken visszafelé haladva elérjük az eredeti célt



- Pontosan ugyanez a megfigyelés érvényes a jobb-élekre

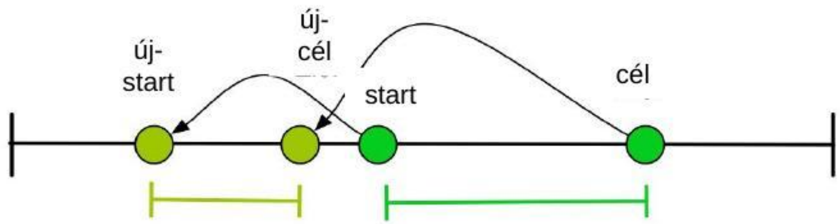


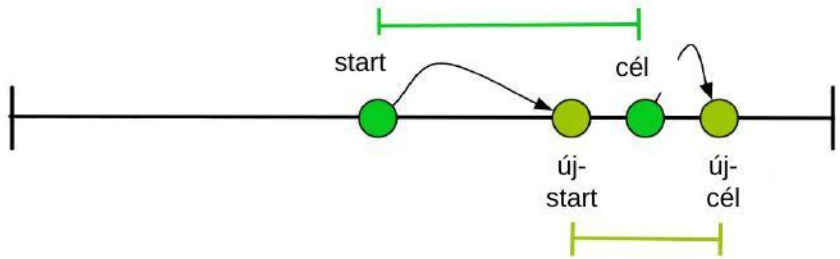
Lemma

A keresés O(log n) lépést és üzenetet igényel

Random-routing

- **random-routing**(start,cél)
  - **if** start és cél szomszédos **then**
    - küldjük az üzenetet a start-tól a cél-nak
  - **else**
    - 50% valószínűséggel
      - új-start ← bal-mutató(start)
      - új-cél ← bal-mutató(cél)
    - egyébként (50%valsz.)
      - új-start ← jobb-mutató(start)
      - új-cél ← jobb-mutató(cél)
    - küldjük az üzenetet a start-tól az új start-nak
    - **random-routing**(új-start,új-cél)
    - Küldjük az üzenetet az új-céltól a cél-nak





Maximális terhelés az éleken keresésnél

- A bal- és jobb-éleket tetszőlegesen felváltva is használhatjuk (ekkor ezt a sorozatot az éleken visszafelé fordított sorrendben kell használni).

- Meg lehet mutatni, hogy ekkor
- A maximális terhelés az éleken O(log n),
  - azaz minden Peer legfeljebb O(log n) csomagot továbbít, ha minden Peer pontosan egy lekérdezést kezdeményez

Új Peer befűzése a Distance-Halving hálózatba

- Hajtsunk végre többszörös kiválasztást,
  - azaz válasszunk c log n véletlen pozíciót $y_1, \ldots, y_{clogn}$
  - minden y_i pozícióhoz tekintsük az intervallumot, ami y_i-t tartalmazza
  - ezen intervallumok közül válasszuk a legnagyobbat
  - felezzük meg ezt az intervallumot
- Aktualizáljuk a szomszédokhoz vezető éleket
- Aktualizáljuk a bal- és jobb-éleket
  - A szomszédok bal- és jobb-éleinek felhasználásával

Tétel 1

Új Peer befűzéséhez a Distance-Halving hálózatba $O(\log^2 n)$ idő és üzenet szükséges

Új Peer befűzése a Distance-Halving hálózatba

Lemma 1

Ha $n=2^k$ Peer a többszörös kiválasztás elvével kerül befűzésre a Distance-Halving hálózatba, akkor nagy valószínűséggel csak 1/(2n), 1/n és 2/n hosszú intervallumok maradnak

Biz:

Mivel az intervallumokat mindig középen felezzük, minden intervallum hossza 2-hatvány.
Elég megmutatni, hogy nincs rövidebb intervallum, mint 1/(2n) és nincs hosszabb, mint 2/n. Ehhez:

Lemma 2

Legyen a leghosszabb intervallum hossza g/n (g értéke függhet n-től).
Akkor 2n/g Peer befűzése után, nagy valószínűséggel, minden intervallum kisebb lesz mint g/(2n)

Új Peer befűzése a Distance-Halving hálózatba

Lemma 2

Legyen a leghosszabb intervallum hossza g/n (g értéke függhet n-től). Akkor 2n/g Peer befűzése után, nagy valószínűséggel, minden intervallum kisebb lesz mint g/(2n).

Biz.:

Tekintsünk egy g/n hosszú intervallumot.
Minden befűzésnél c log n pozíciót tekintünk és 2n/g Peert fűzünk be.
Így a találatok számának X ebben az intervallumban a várható értéke:

$$E[X] = \frac{g}{n} \cdot \frac{2n}{g} \cdot c \log n = 2c \log n.$$

A Chernoff korlátból:

$$\Pr[X \le (1-\varepsilon) E[X]] \le n^{-\varepsilon^2 c}.$$

Ha $\varepsilon^2 c$>=2, akkor minden ilyen hosszú intervallumot eltalálunk legalább (1-ε)2c log n –szer. Itt figyelembe kell venni azt is, hogyha egy intervallumot eltalálunk, akkor a Peer többi (c log n) -1 választása nem okoz intervallum felezést. 2(1-e)>=1 esetén érvényes, hogy minden >=g/n hosszú intervallum nagy valószínűséggel felosztásra kerül. □

Új Peer befűzése a Distance-Halving hálózatba

Biz. (Lemma 1):

- Lemma 2 alkalmazásával g = n/2, n/4,…,4 értékekre egymás után, minden kör után nagy valószínűséggel kizárható, hogy egy g/n hosszú intervallum megmarad.
- A körökben belépő Peer-ek száma: 4+8+…+n/2 < n.
- Az utolsó kör után minden intervallum hossza <= 2/n.
- Mivel csak O(log n) eseménynek kell fellépni nagy valószínűséggel, az állítás még mindig nagy valószínűséggel érvényes.

- A legfeljebb 1/(2n) hosszú intervallumok összhossza legfeljebb n/2
- Ezért annak a valószínűsége, hogy c log n kísérlet során csak ilyen intervallumokat választunk (és tovább osztunk) <= $2^{-c \log n} = n^{-c}$. □

Distance-Halving

- fokszám O(1)
- átmérő O(log n)
- terhelés kiegyensúlyozás
- keresés O(log n)
- befűzés $O(\log^2 n)$

- A folytonosról diszkrét gráfra való átmenet
  Itt lett először formalizálva

- A többszörös kiválasztás elve felhasználható pl. a Chord hálózatban is
  - be- és ki-fok O(log n)
  - Továbbra is $O(\log^2 n)$ idő és üzenet szükséges a befűzéshez

Irodalom

M. Naor, U. Wieder: **Novel architecture for P2P applications: the continuous-discrete approach**. *Proc. 15th ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, 50-59, 2003.