Hálózatok II
2006

5: Online Hozzáféréskontrol és Útvonalválasztás ATM Hálózatokban

Asynchronous Transfer Mode ATM

- Videokonferencia, video-on-demand:
- Hogyan lehet szélessávú kommunikációs hálózatban (pl. ADSL) ilyen applikációknál garantálni egy megadott Quality of Service-t (QoS)?
- Asynchron Transfer Mode (ATM): Az adatok egyenlő hosszúságú cellákban (53 Byte) asynchron módon kerülnek átvitelre.
- Különböző adatfolyamok cellái meg tudják osztani egy link kapacitását, a cellák a linken egymás után kerülnek átvitelre.
- Aszinkron itt azt jelenti, hogy egy adatfolyam cellái nem szükségszerűen periódikusan kapnak hozzáférést a linkhez.

ATM-Hálózatok – Sávszélesség-foglalás, CAC

- Egy kommunikációs kapcsolat felépítésénél egy adatfolyamhoz a hívó és a fogadó között egy útvonalon minden linken a megfelelő sávszélesség lefoglalásra kerül, ami az adatfolyamnak garantáltan rendelkezésére áll.
- Az adatfolyamok sávszélességigényének összege egy linken nem lépheti túl a link kapacitását.
- ATM-hálózatokon egy hozzáféréskontrol (ang. Connection Admission Control CAC) mechanizmus szükséges, azaz a kapcsolatkéréseket, amelyek sávszélességigénye nem garantálható, el kell utasítani.

QoS-osztályok a Gyakorlatban

Megjegyzés: A gyakorlatban a szituáció összetettebb: ATM-hálózatok különböző QoS-osztályokat támogatnak, pl.

- CBR (constant bit rate): a megfelelő bit-rátát garantálni kell.
- rt-VBR(real time variable bit rate) interaktív kommunikáció esetén video- vagy audioátvitelnél az adatokat nem szükséges mindig ugynazzal a sebességgel küldeni, de garantálni kell
  - a maximális késést egy cella küldése és fogadása között,
  - a maximális késést egy adatfolyam két egymást követő cellája között és
  - az átlagos bit-rátát.
- nrt-VBR (non real time variable bit rate) pl. video- vagy audio-streaming esetén csak
  - a maximális késés két egymást követő cella között és
  - az átlagos bit-ráta releváns.
- UBR (undefined bit rate): pl. file-átvitel (ftp) esetén. Csak az a fontos, hogy megérkezzenek az adatok…

CAC és Útvonalválasztás ATM-Hálózatokban

- Az ATM-hálózatokat egy absztrakt szinten vizsgáljuk:
  a sávszélesség-foglalás problémát úgy tekintjük, mintha csak CBR osztály létezne.

A Probléma:

- Minden kapcsolatkérés specifikálja a küldőt, a fogadót és a sávszélességigényt.
- A hálózatnak minden kapcsolatkérésnél el kell dönteni, hogy a kérést elfogadja-e vagy elutasítja (CAC).
- Minden elfogadott híváshoz hozzá kell rendelni a küldő és a fogadó között egy útvonalat, amelyet az adatfolyam használ (routing).
- A döntéseket a jövőbeli kapcsolatkérések ismerete nélkül kell meghozni.
  Ezt a szcenáriót online-nak nevezzük. (Ellentéte: u.n. offline-szenarióban az algoritmus ismeri az input adatokat a kezdettől fogva.)
- Optimalizálási kritériumok: az elfogadott híváskérések
  - számát (vagy profitját) maximalizálni, vagy
  - sávszélességének összegének maximumát egy linken (congestion) minimalizálni,…

Online Algoritmusok Minőségének Értékelése

Hogyan lehet a megoldás minőségét értékelni egy online-szcenárióban? A jövőbeli kapcsolatkérések nem ismertek.

- Legyen I egy maximalizálási- (minimalizálási-) probléma P egy instanciája.
- Legyen A egy online-algoritmus a P problémához és
- $M(A,I)$ az A algoritmus által kiszámított célfüggvényérték I-hez.
- Legyen OPT_I az optimális célfüggvényérték az offline-szcenárióban (amit egy algoritmus a futási idő korlátozása nélkül kiszámíthatna, ha az algoritmus I-t teljesegészében előre ismerné.)
- $\max_I \{ OPT_I / M(A,I) \}$ értékét az A algoritmus kompetitív rátájának (competitive ratio) nevezzük (minimalizálási problémáknál a kompetetív rátát mint $\max_I \{ M(A,I) / OPT_I \}$ definiáljuk).

A hozzáférés kontrollnál az elfogadott hívások profitjának maximalizálását vizsgáljuk (a hálózat üzemeltető profitjának maximalizálása).

Példa Online-Algoritmusra: Síkölcsönzés Probléma

- Egy kezdő síelő problémája:
  - Egy sífelszerelés ára: 30.000 Ft
  - Kölcsönzés ára: 3.000 Ft / nap
- Rossz stratégiák:
  - Azonnal vásárolni
    - Költség: 30.000 Ft
    - Az első nap után elolvad a hó, nem tetszik, vagy baleset történik…
    - A legjobb (offline) stratégia: egy nap kölcsönzés: 3.000 Ft
    - Kompetitív ráta: 30.000 Ft / 3.000 Ft = 10
  - Minden nap kölcsönözni
    - Kompetitív ráta (ha tetszik... Opt. stratégia: azonnal megvenni) $\rightarrow \infty$
- Optimális stratégia:
  - 10 napig kölcsönözni, 11. nap vásárlás
  - Optimalis: kompetetív ráta minden nap ≤ 2

CAC és Routing ATM-Hálózatokban [Awerbuch, Azar, Plotkin 93]

A problema:

- A hálózat a $G=(V,E)$ gráf által adott, $|V|=n$, $|E|=m$. Minden $e \in E$ élnek van egy kapacitása $u(e)$.
- A hálózathoz egymás után k kapcsolatkérés $\beta_1,\beta_2,\ldots,\beta_k$ érkezik. $\beta = (\beta_1,\beta_2,\ldots,\beta_k)$ a k kapcsolatkérés sorozata.
- Minden β_i kapcsolatkérés a $\beta_i = (s_i,t_i,b_i,\rho_i)$ négyes által adott, ahol
  - $s_i \in V$ a küldő,
  - $t_i \in V$ a fogadó,
  - b_i a sávszélesség-igény (átviteli ráta) és
  - ρ_i a hálózat üzemeltető profitja, ha β_i elfogadásra kerül. Feltesszük, hogy $\rho_i = n \cdot b_i$, azaz a profit egyenesen arányos a sávszélességhez. (A szorzó n egy normalizálás, amely később hasznos lesz).
- Az egyszerűség kedvéért feltesszük, hogy a kapcsolat minden elfogadott kapcsolatkérés végtelen ideig (a k-adik kérés beérkezése utánig) megmarad.

CAC és Routing ATM-Hálózatokban

<u>Megjegyzés</u>: A valóságot reálisabban leíró modelleket hasonló módszerekkel kezelhetünk és analizálhatunk. A modell kiterjesztéséhez azokra az esetekre, amikor

- a kapcsolatok legfeljebb $\leq T$ ideig tartanak, vagy
- a sávszélesség a kapcsolat fennállásának ideje alatt változik, vagy
- a profit legfeljebb egy F szorzóval tér el $n \cdot b_i$ -től,

lásd [Awerbuch, Azar, Plotkin 93] és [Plotkin 95].

Az alapul szolgáló technika ugyanaz: Routing a legrövidebb utakon egy költségfüggvény szerint az éleken, amely exponenciális az él kapacitásának a már lefoglalt arányában.

CAC és Routing ATM-Hálózatokban

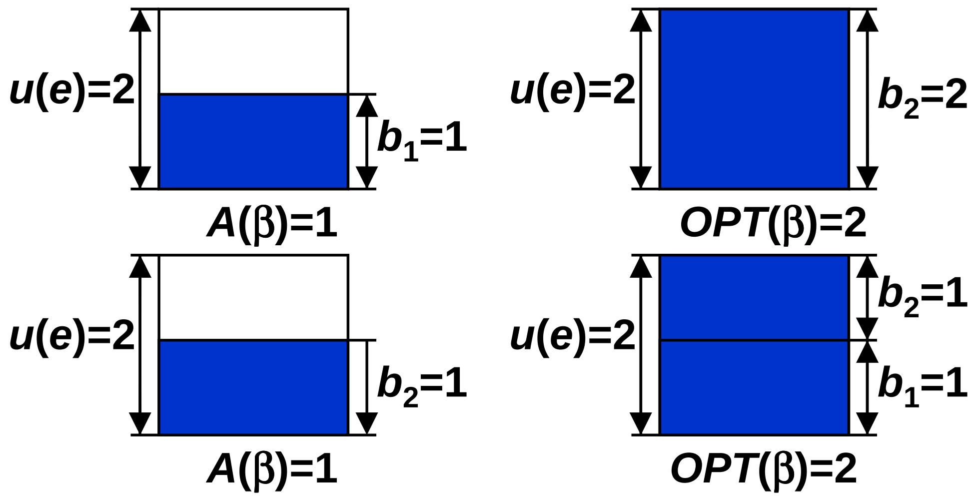
- Egy *A* algoritmus az online-CAC problémához a $\beta_1,\beta_2,\ldots,\beta_k$ kapcsolatkéréseket egymás után dolgozza fel ebben a sorrendben.
- Minden β_i kapcsolatkéréshez *A*-nak el kell dönteni a későbbi kapcsolatkérések ismerete nélkül, hogy β_i -t elfogadja, vagy elutasítja.
- Ha β_i -t elfogadja, akkor meg kell határozni egy P_i útat s_i -től t_i -hez *G*-ben, amin β_i -hez a sávszélesség lefoglalásta kerül (routing).
- Minden időpontban minden $e \in E$ élre érvényesnek kell lenni, hogy az elfogadott β_i kérések b_i sávszélességeinek az összege, melyek az *e* élet használják, legfeljeb $u(e)$.
- Elfogadott kapcsolatkéréseket később nem szabad megszakítani.

CAC és Routing ATM-Hálózatokban

- Legyen $A(\beta)$ az összes elfogadott β_i kapcsolatkérés ρ_i profitjának összege, amit az *A* online-algoritmus β inputra elfogad.
- Legyen $OPT(\beta)$ az összes elfogadott β_i kapcsolatkérés ρ_i profitjának összege egy optimális megoldásban β inputra.
- Mivel egy online-algoritmus a jövőbeli kapcsolatkéréseket nem ismeri, nem találhat optimális megoldást.

<u>Példa.</u>: Álljon a hálózat két csomópontbol és egyetlenegy linkből G=({*u*,*v*},{*e*={*u*,*v*}}), amely kapacitása $u(e)$=2. Tekintsünk két kapcsolatkérést $\beta=(\beta_1,\beta_2)$. Legyen b_1=1.

- 1. eset: Ha az online-algoritmus β_1-t elfogadja, akkor legyen b_1=2. Ekkor $A(\beta)$ =1 és $OPT(\beta)$ = 2.
- 2. eset: Ha az online-algoritmus β_1-t nem fogadja el, akkor legyen b_1=1. Ekkor $A(\beta)$ =1 és $OPT(\beta)$ = 2.



CAC és Routing ATM-Hálózatokban

- Azt mondjuk, hogy az *A* online-algoritmus kompetitív rátája $c \geq 1$, ha minden β inputra teljesül, hogy $OPT(\beta) / A(\beta) \leq c$.
- Egy ilyen algoritmust *c*-kompetitív-nek nevezünk.
- A profit, amit egy *c*-kompetitív online-algorithmus a CAC és routing problémához garantál, legfeljebb *c*-szer kisebb, mint az optimális profit.

CAC és Routing Algoritmus

Notáció:

- Legyen $\mu = 2n+2$ és $u_{\min} = \min_{e \in E} u(e)$.
- Szükségünk van arra a feltételre, hogy minden β_i-ra teljesül, hogy $b_i \leq u_{\min} / \log \mu$.
  Azaz, minden kapcsolat legfeljebb $(1 / \log \mu)$ részét foglalhatja le egy él kapacitásának. Ez a feltétel (legtöbbször) teljesül a gyakorlatban.
- Legyen $I(j)$ az online-algoritmus által elfogadott kapcsolatkérések indexeinek halmaza direkt β_j feldolgozása előtt.
- A normalizált terhelés $\lambda_e(j)$ az e élen közvetlenül β_j feldolgozása előtt $\lambda_e(j) = (\Sigma_{i \in I(j),\, e \in P} b_i) / u(e)$, az elfogadott β_i, $i < j$, kérések b_i sávszélességeinek összege, melyek P_i útvonala az e élet tartalmazza, osztva $u(e)$-vel.
- $I(k+1)$ jelöli az elfogadott kapcsolatkérések indexeinek halmazát $\lambda_e(k+1)$ pedig a normalizált terhelést az e élen a $\beta = (\beta_1,\beta_2,\ldots,\beta_k)$ sorozat feldolgozása után.

CAC és Routing Algoritmus

Az Algoritmus:

- Legyen β_j a következő feldolgozandó kapcsolatkérés.
  Definiáljuk minden *e* élhez $c_e(j) = u(e)\cdot(\mu^{\lambda_e(j)} - 1)$.
- Utaljuk minden *e* élhez a $w_j(e) = (b_j/u(e))\cdot c_e(j)$ költséget.
- Számítsunk ki egy legrövidebb *P* utat s_j -től t_j -hez a $w_j(e)$ élköltségek szerint. (pl. Dijkstra algoritmusával).
- Ha *P* költsége legfeljebb ρ_j, azaz $\Sigma_{e\in P} w_j(e) \le \rho_j$, akkor fogadjuk el β_j -t, egyébként utasítsuk el β_j-t.

Az alapötlet tehát, minden *e* élhez egy költséget definiálni, amely exponenciális a $\lambda_e(j)$ normalizált élterhelésben. Ekkor egy β_j kapcsolatkérést pontosan akkor elfogadni, ha a legrövidebb út a küldőtől a fogadóig ezen költség alapján nem drágább, mint β_j profitja ρ_j.

CAC és Routing Algoritmus Analízise

Megmutatjuk, hogy az online-algoritmus a CAC és Routing problémához $O(\log n)$-kompetitív. Az analízist három lemmára osztjuk.

Lemma 1: Legyen $0 \le j \le k$ és legyen $I(j+1)$ a kapcsolatkérések indexeinek halmaza, amiket az algoritmus $\beta_1,\beta_2,\ldots,\beta_j$ közül elfogad:

$$\mathbf{2\log}\mu \sum_{i\in I(j+1)} \rho_i \ge \sum_{e\in E} c_e(j+1).$$

Biz.: Indukció j szerint. j=0 esetén minden rendben: mindkét oldal 0.
Tegyük fel, hogy az egyenlőtlenség igaz minden j-től kisebb indexre.
Tekintsük a β_j feldolgozását.
Ha β_j-t elutasítjuk, akkor az egyenlőtlenség egyik oldala se változik, így érvényes marad.

CAC és Routing Algoritmus Analízise

Ha β_j -t elfogadjuk, akkor a bal oldal $(2\rho_j \log \mu)$ -vel, a jobb oldal pedig $\Sigma_{e\in P_j}\big(c_e(j+1) - c_e(j)\big)$ -vel lesz nagyobb. Elég megmutatni, hogy

$$\sum_{e\in P_j}\big(c_e(j+1)-c_e(j)\big)\le 2\rho_j \log\mu.$$

$c_e(j)$ definíciója szerint teljesül minden $e\in P_j$ -re:

$$\begin{aligned} c_e(j+1)-c_e(j) &= u(e)\cdot\left(\mu^{\lambda_e(j)+\frac{b_j}{u(e)}}-\mu^{\lambda_e(j)}\right)\\ &= u(e)\cdot\mu^{\lambda_e(j)}\cdot\left(\mu^{\frac{b_j}{u(e)}}-1\right)\\ &= u(e)\cdot\mu^{\lambda_e(j)}\cdot\left(2^{\frac{b_j}{u(e)}\log\mu}-1\right)\end{aligned}$$

CAC és Routing Algoritmus Analízise

A feltétel miatt, hogy $b_i \le u_{\min} / \log \mu$, minden $1 \le i \le k$ esetén teljesül
$(b_j / u_{\min}) \cdot \log \mu \le 1$.
Mivel $0 \le x \le 1$, érvényes, hogy $2^x - 1 \le x$, következik, hogy

$$
\begin{aligned}
c_e(j+1) - c_e(j) &= u(e) \cdot \mu^{\lambda_e(j)} \left(2^{\frac{b_j}{u(e)} \log \mu} - 1 \right) \\
&\le u(e) \cdot \mu^{\lambda_e(j)} \cdot \frac{b_j}{u(e)} \log \mu \\
&= b_j \cdot \mu^{\lambda_e(j)} \cdot \log \mu \\
&= b_j \cdot \left(\frac{c_e(j)}{u(e)} + 1 \right) \cdot \log \mu \\
&= \log \mu \cdot \left(\frac{b_j}{u(e)} c_e(j) + b_j \right) \\
&= \log \mu \cdot \left(w_j(e) + b_j \right)
\end{aligned}
$$

CAC és Routing Algoritmus Analízise

Összeadva minden $e \in P_j$ -re és felhasználva, hogy $\Sigma_{e \in P_j} w_j(e) \leq \rho_j$ (ez teljesül, mert β_j -t elfogadtuk) azt kapjuk, hogy:

$$\begin{aligned}
\sum_{e \in P_j} \left(c_e(j+1) - c_e(j) \right) &\leq \sum_{e \in P_j} \log \mu \left(w_e(j) - b_j \right) \\
&\leq \log \mu \left(\rho_j + |P_j| \cdot b_j \right) \\
&\leq \log \mu \left(\rho_j + n \cdot b_j \right) \\
&= \log \mu \left(\rho_j + \rho_j \right) \\
&= 2\rho_j \log \mu. \qquad \square
\end{aligned}$$

CAC és Routing Algoritmus Analízise

Lemma 2: Legyen Q azon kapcsolatkérések indexeinek halmaza, amelyeket az optimális megoldás A^* elfogad, de az online-algoritmus nem. Legyen $h = \max Q$. Ekkor

$$\sum_{j \in Q} \rho_j < \sum_{e \in E} c_e(h).$$

Biz.: Jelöljük $j \in Q$ -hoz P_j^* -vel az utat, amit az optimális megoldás a β_j kapcsolatkéréshez hozzárendelt.
Mivel β_j -t az online-algoritmus elutasította, és mivel $c_e(j)$ értéke j -vel monoton nő, teljesül

$$\rho_j < \sum_{e \in P_j^*} w_j(e) = \sum_{e \in P_j^*} \frac{b_j}{u(e)} c_e(j) \leq \sum_{e \in P_j^*} \frac{b_j}{u(e)} c_e(h).$$

CAC és Routing Algoritmus Analízise

Összeadva minden $j \in Q$ -ra azt kapjuk, hogy:

$$\sum_{j \in Q} \rho_j < \sum_{j \in Q} \sum_{e \in P_j^*} \frac{b_j}{u(e)} c_e(h)$$

$$= \sum_{e \in E} \sum_{j \in Q: e \in P_j^*} \frac{b_j}{u(e)} c_e(h)$$

$$= \sum_{e \in E} c_e(h) \sum_{j \in Q: e \in P_j^*} \frac{b_j}{u(e)}$$

$$= \sum_{e \in E} c_e(h).$$

Az utolsó egyenlőtlenség ázért teljesül, mert az optimális megoldás a kapacitás-korlátot be kell hogy tartsa, és így $\sum_{j \in Q:\, e \in P_j^*} (b_j/u(e)) \leq 1$ teljesül. □

CAC és Routing Algoritmus Analízise

Az optimális megoldás profitja nem lehet több, mint a kapcsolatkérések profitjának összege, melyek indexe *Q*-ban van, plussz a kapcsolatkérések profitja, melyek indexe $I(k+1)$-ben van (az online-algoritmus *A* által elfogadott indexek). Lemma 1 és Lemma 2 miatt $OPT(\beta) \leq (2 \log \mu + 1) \cdot A(\beta)$:

$$
\begin{aligned}
OPT(\beta) &\leq \sum_{i \in Q} \rho_i + \sum_{i \in I(k+1)} \rho_i \\
&< \sum_{e \in E} c_e(h) + \sum_{i \in I(k+1)} \rho_i \\
&\leq \sum_{e \in E} c_e(k+1) + \sum_{i \in I(k+1)} \rho_i \\
&\leq 2 \log \mu \sum_{i \in I(k+1)} \rho_i + \sum_{i \in I(k+1)} \rho_i \\
&\leq (2 \log \mu + 1) \sum_{i \in I(k+1)} \rho_i \\
&= (2 \log \mu + 1) \cdot A(\beta).
\end{aligned}
$$

CAC és Routing Algoritmus Analízise

Meg kell még mutatni, hogy az online algoritmus által kiszámolt megoldás betartja a kapacitás-korlátokat.

Lemma 3: Legyen $I(k+1)$ a kapcsolatkérések indexeinek halmaza, amiket az online-algoritmus elfogad. Minden $e \in E$ élre teljesül:

$$\sum_{j \in I(k+1): e \in P_j} b_j \leq u(e).$$

Biz.: Ellentmondás által. Tegyük fel, hogy az online-algoritmus β_j elfogadása után sérti meg először a kapacitás-korlátot. Legyen e az az él, ahol ez történik. Ekkor teljesülni kell, hogy

$$\lambda_e(j) > 1 - \frac{b_j}{u(e)}$$

mivel az e él P_j –ben kell hogy legyen és β_j elfogadása után $\lambda_e(j+1) = \lambda_e(j) + b_j / u(e) > 1$.

CAC és Routing Algoritmus Analízise

A definíció $c_e(j) = u(e) \cdot (\mu^{\lambda_e(j)} - 1)$ miatt és a feltétel alapján, miszerint $b_i \leq u_{\min} / \log \mu$ minden $1 \leq i \leq k$, azt kapjuk, hogy:

$$\frac{c_e(j)}{u(e)} = \mu^{\lambda_j(e)} - 1 > \mu^{1 - \frac{b_j}{u(e)}} - 1 \geq \mu^{1 - \frac{1}{\log \mu}} - 1 = \frac{\mu}{2} - 1 = \frac{2n + 2 - 2}{2} = n.$$

Tehát az teljesül, hogy:

$$\sum_{e' \in P_j} w_j(e') \geq w_j(e) = \frac{b_j}{u(e)} c_e(j) > b_j n = \rho_j.$$

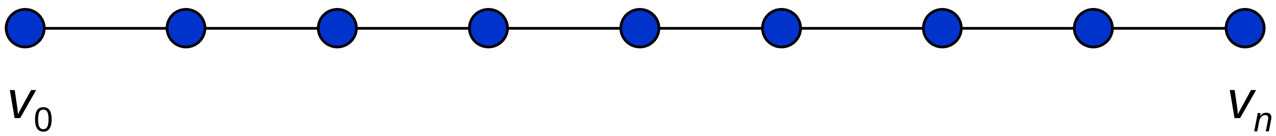
Ez ellentmond annak, hogy β_j -t elfogadta és ahhoz a P_j utat rendelte az online-algoritmus, mivel akkor a költség legfeljebb $\rho_j = nb_j$ lehet.

CAC és Routing Algoritmus Analízise

Tétel 1: Az online-algoritmus a CAC és Routing problémára nem sérti meg soha az élek kapacitását és a kompetetív rátája $O(\log n)$, azaz az online algoritmus minden β kapcsolatkérés-sorozatra olyan $A(\beta)$ profitot garantál, amely legfeljebb $2\log(\mu) = 2\log(2n+2) = O(\log n)$ -szor kisebb, mint egy optimális (offline) megoldás profitja. □

Alsó Korlát a Kompetitív Rátára

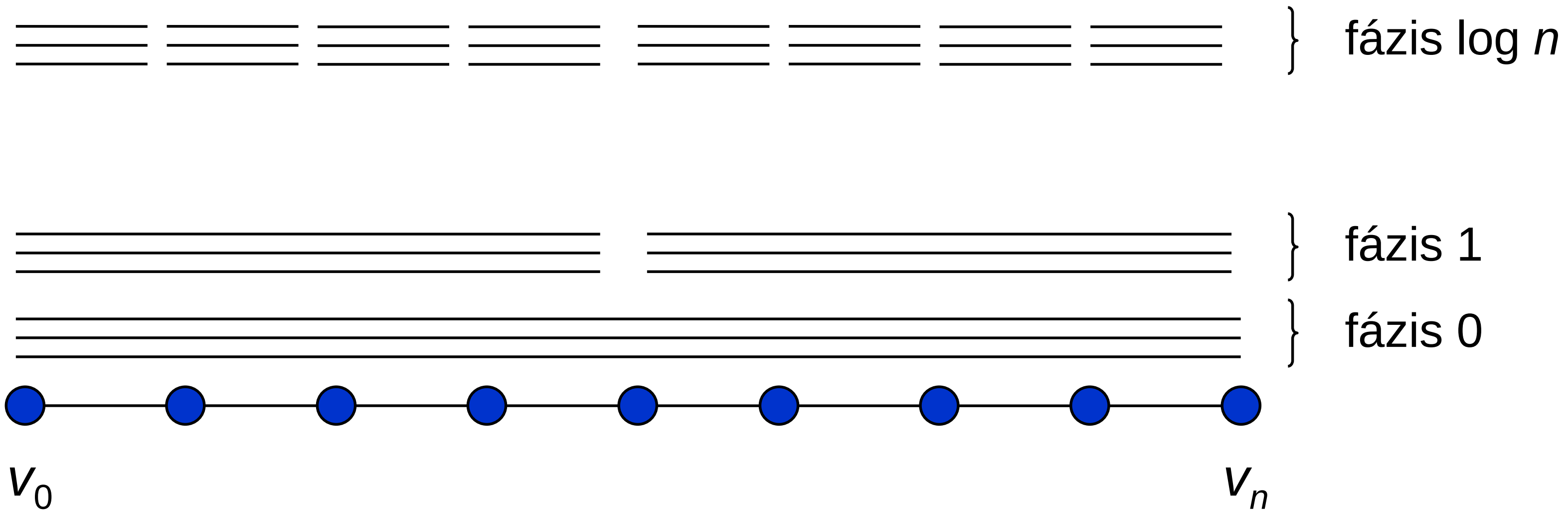
Legyen $G(n)$ egy gráf, amely n élből álló láncból áll (n+1 csomópont). Jelölje $V=\{v_0,\ldots,v_n\}$ a csomópontokat és legyen $n = 2^k$, k2 **N.** Legyen minden él kapacitása 1.



<u>Tétel 2</u>: $G(n)$-ben a CAC és Routing problémára minden online-algoritmus kompetetív rátája $\Omega(\log n)$.

Alsó Korlát a Kompetitív Rátára

<u>Biz.</u>: Tekintsünk egy β kapcsolatkérés-sorozatot, amely log n + 1 fázisból áll.
Minden fázis i, $0 \cdot i \cdot \log n$, 2^i kérés-csoportot tartalmaz, $0 \cdot j \cdot 2^i - 1$.
Minden kérés az i. fázisban, j. csoportban $v_{jn/2^i}$-től $v_{(j+1)n/2^i}$-hez irányul.
Minden i,j-hez $1/\alpha$ azonos kérés tartozik, mindegyik α sávszélességet kér és α profitot hoz ($\alpha \cdot 1 / \log n$ fix).



Alsó Korlát a Kompetitív Rátára

- Legyen x_i a profit, amit az online-algoritmus az *i*. fázisban gyüjt.
- Egy egségnyi profitot az *i*. fázisban úgy lehet csak szerezni, ha lefoglalunk $n / 2^i$ egységnyi sávszélességet a kapacitásokból.
- Mivel a rendelkezésre álló kapacitások teljes összege *n*, azt kapjuk, hogy

$$\sum_{i=0}^{\log n} x_i \cdot n/2^i \;\leq\; n$$

- Legyen $S_j = 2^{-j} \sum_{0 \leq i \leq j} x_i.$
- Ekkor

$$\sum_{0 \leq j \leq \log n} S_j \;=\; \sum_{0 \leq i \leq j \leq \log n} 2^{-j} x_i \;\leq\; \sum_{0 \leq i \leq \log n} 2 \cdot 2^{-i} x_i \;=\; 2.$$

- Így létezik egy fázis *k*, melyre $S_k \cdot 2 / \log n$.

Alsó Korlát a Kompetitív Rátára

- Tekintsük a kapcsolatkérés-sorozatot, ami β-nak pontosan az első *k* fázisát tartalmazza.
- Az online-algoritmus profitja a *k*. fázis végén:

$$\sum_{i=0}^{k} x_i \;=\; 2^k S_k \;\leq\; 2^k \cdot (2/\log n).$$

- Az optimális offline-algoritmus elutasít mindent az első *k*-1 fázisban és elfogadja a *k*. fázis összes kérését. Ekkor a profit: 2^k. □

Megjegyzés

- Eddig nem tettünk fel semmit a kapcsolatkérések érkezéséről. Az online algoritmus O(log n) kompetetív rátát garantál a legrosszabb esetre is.

- Speciális eset (a telekommunikációban gyakran használt modell):
  - a kapcsolatkérések érkezési ideje Poisson eloszlású és
  - a kapcsolatok tartási ideje exponenciális eloszlású.
- Ekkor az online algoritmus módosított változatával garantálható egy $R^*+\varepsilon$ *várható* elutasítási ráta, ahol
  - R^* a várható elutasítási rátája az optimális offline algoritmusnak és
  - $\epsilon = O(\sqrt{b_{max} \log n / u_{min}})$
    - b_{max}: maximális sávszélesség amit egy kapcsolat kér
    - u_{min}: minimális ékapacitás
  - (lásd [Kamath et al. 96])

Megjegyzés

- Eddig feltettük, hogy a kérés elfogadása vagy elutasításához minden él terheléséről minden időpillanatban pontos információ áll rendelkezésre a döntéshozáshoz
- [Goel et al. 01] leír egy módszert, ahol k döntéshozó csomópont van a hálózatban
  - minden kérést a k döntéshozó csomópont egyike fogad és dolgoz fel.
  - a kérést feldolgozó csomópont a saját hálózatképe alapján beengedi vagy elutasítja a kérést
  - a döntés meghozása után a kéréshez rendelt útvonalról összegyüjti az akuális terhelés információt és aktualizálja a saját adatbázisát
- [Räcke, Rosen 05] bemutat egy teljesen elosztott módszert: minden csomópontnak csak lokális információ áll rendelkezésre

CAC és Routing

Irodalom:

[Awerbuch, Azar, Plotkin 93]: B. Awrbuch, Y. Azar, and S. Plotkin: *Throughput-Competetive On-Line Routing*. In Proc. 34th FOCS, pages 32-40, 1993.

[Plotkin 95]: S. Plotkin: *Competetive Routing of Virtual Circuits in ATM Networks*. IEEE Journal of Selected Areas in Communications, Vol. 13, pages 1128-1136, 1995.

Továbbvezető irodalom:

[Kamath et al. 96]: A. Kamath, O. Palmon, S. Plotkin: *Routing and Admission Control in General Topology Networks with Poisson Arrivals*. In Proc. 7th SODA, pages 269-278, 1996.

[Goel et al. 01]: A. Goel, A. Meyerson, and S. Plotkin: *Distributed Admission Control, Scheduling, and Routing with Stale Information*. In Proc. 12th SODA, pages 611-619, 2001.

[Räcke, Rosén 05]: H. Räcke, A. Rosén: *Distributed online call control on general networks*. In Proc. 16th SODA, pages 791-800, 2005.