

# Location Area tervezés következő generációs mobil hálózatokban

---

## 1. Miért van szükség Location Area-k kialakítására?

A következő generációs (3G illetve 4G) mobil hálózatok egyik legfontosabb szolgáltatás minőségi paramétere (Quality of Service, továbbiakban QoS) a késleltetés illetve a késleltetés ingadozás. Mivel ezekben a mobil hálózatokban lecsökken a cellák mérete, ezért gyakrabban kerül sor a mobil terminálok cellaváltására (handover). Minden cellaváltás járulékos jelzések generálását jelenti (mivel értesíteni kell a felsőbb hierarchia szinteket és ügynököket, hogy a mobil másik cellában tartózkodik), így ez a megnövekedett jelzésforgalom kihatással lesz a késleltetés ingadozására is, ami komoly fennakadásokat okozhat időfüggő, valós idejű szolgáltatások esetén. A jelzés többlet forgalom az ún. regisztrációs folyamat eredménye, amikor a mobil terminál frissíti a tartózkodási helyének információját az otthoni ügynökénél.

A jelzések csökkentése érdekében ún. Location Area-kat lehet kialakítani, ami azt jelenti, hogy több cellát egy adminisztrációs egységbe vonunk össze és együtt kezeljük a jelzésforgalmukat. Ily módon, az egységen belüli cellaváltások rejtve maradnak a felső hierarchia szintek előtt. Ami azt jelenti, hogy csak akkor keletkezik többletjelzés, ha az egységek (Location Area-k) határát lépi át a mobil terminál.

## 2. Milyen két egymásnak ellentmondó szempont nehezíti a Location Area optimalizálást?

Location Area (LA) méretéből addódik ez a probléma:

- A méret csökkentésének és növelésének is megvan a maga előnye. Ugyanis ha több cellát vonunk össze egy LA-ba, akkor a LA-k közötti váltások száma csökkeni fog, így a hely információt hordozó (location update) üzenetek száma is kisebb lesz a szokásosnál.
- **Túl sok:** Ha túl sok cella tartozik egy LA-ba, a hálózatba bejövő hívás sok paging üzenetet (a paging eljárás arra szolgál hogy megtaláljuk azt a cellát, amelyben az a mobil terminál tartózkodik, amelyiknek épp hívása érkezett) a fog generálni, ugyanis minden cellába ilyen üzenetet kell kiküldeni hogy megtaláljuk vajon melyikben tartózkodik az adott mobil.
- **Kevés:** Ha kevesebb cellát vonunk össze egy ilyen egységbe, akkor nem kell annyi paging üzenetet kiküldeni (így kevésbé terheljük a csatornákat és a feldolgozási idő is lecsökken), de viszont ilyenkor meg növekszik az LA váltások száma (mivel kisebbek).

## 3. Miért van szükség pagingre? Vezesse le a paging költségfüggvényt egy Location Area-ra!

**Paging:** a paging eljárás arra szolgál, hogy megtaláljuk azt a cellát, amelyben az a mobil terminál tartózkodik, amelyiknek épp hívása érkezett.

Amikor hívás érkezik a mobil hálózatba, a mobil kapcsoló központ minden hozzá tartozó bázisállomáshoz kiküldi a paging üzenetet, hogy megtalálja a mobil terminál tartózkodási helyét. Így az adott LA-ban minden cellába beérkeznek a paging üzenetek, annak ellenére, hogy csak egyikük fog

felelni, ahol éppen a mobil pillanatnyilag van. Így definiálhatjuk a paging költségfüggvényt az  $i$ -ik LA-ra, amivel leírható az a sávszélesség, ami a paging folyamat során lefoglalódik az adott időintervallumra:

$$C_{p_i} = \sum_{i=1}^K N \cdot \lambda_i \cdot B_p$$

$N$  – a cellák száma a  $j$ -ik LA-ban

$\lambda_i$  – a bejövő hívások száma az adott  $i$ -ik mobil terminálon

$B_p$  – a paging üzenet sávszélessége

$K$  – a mobil terminálok száma az adott  $j$ -ik LA-ban

#### 4. Írja le a Location update költségfüggvényt egy Location Area-ra.

A location update költségfüggvény a LA váltások számával függ össze, ilyenkor járulékos elyinformáció frissítő jelzésforgalom keletkezik.

$$C_{lu_i} = B_{lu} \cdot \sum_{j=1}^B q_j$$

$B_{lu}$  – a location update üzenet átvitelére szükséges sávszélesség

$q_j$  – a  $j$ -ik cellahatár átlépési intenzitása

$B$  – a külső cellahatárok száma

#### 5. Mi a különbség az alap és a kiterjesztett alap algoritmus között?

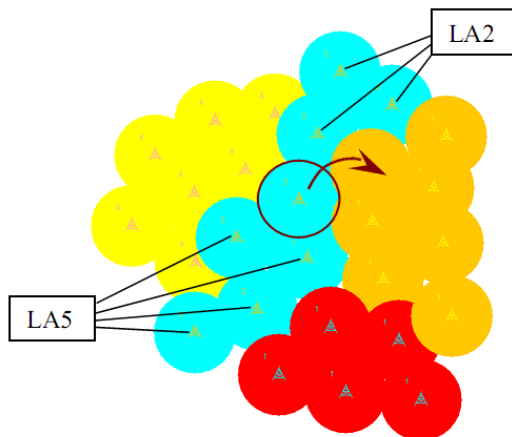
Az alap algoritmusban a kapott LA struktúra költsége függ attól, hogy a futtatás elején mekkora  $N$  (tetszőleges egész szám) paramétert adunk meg. A kiterjesztett alap algoritmus végigpróbálja, hogy a szóba jövő  $N$  értékek közül melyik érték adja a legkisebb költségű LA struktúrát, és végül ezzel az  $N$  értékkel futtatja az alap algoritmust.  $N$  értéke 1-től 100-ig terjedhet, ennél nagyobb  $N$  a jelenleg létező technológiák, és cellaméretek mellett nem jöhet szóba.

#### 6. Írja le az újrendező algoritmus működését!

Az újrendező algoritmus egy igen egyszerű szimulált lehűtéshez hasonlóan működő javító algoritmus.

Vegyünk két olyan cellát, melyek szomszédosak, de nem tartoznak egy LA-ba. Próbáljuk meg átrakni az egyik cellát a másik cella LA-jába. Ezután futtassuk a költség számító függvényeinket, és ha a kialakított új LA-struktúra kisebb költségű, mint az eredeti, akkor véglegesítsük az új struktúrát.

Előfordulhat azonban olyan eset, amikor az egyik LA-ból a másikba mozgatott cella eredeti LA-ja az áthelyezés miatt két, topológiailag különálló részre esik szét.



2. ábra LA-k szétesése az újrendezés során

Ilyen LA-áka tartalmazó struktúra elvileg megengedett, de a költséghatékonyság szempontjából mindenképpen elkerülendő. A szétesett LA két különálló része között nincs cellaváltás, így azzal hogy egy LA-ba tartoznak nem csökken a cellaváltás költsége. Ráadásul, ha bármelyik cellába hívás érkezik, a paging üzeneteket minden cellába ki kell küldenünk. A két különálló részt érdemes tehát két külön LA-ra bontani mivel így nagymértékben csökkenthetjük a paging költséget amellyel, hogy a cellaváltás költsége nem változik. Ez a lépés azért is fontos, mert az algoritmus így képes a kiindulási struktúrában levő LA-k számát növelni. Könnyen előfordulhat ugyanis, hogy a cella-rendszerben az optimális struktúra több LA-ból áll, mint az újrendező algoritmus bemeneti struktúrája, ekkor pedig szükség lehet a LA-k számának szaporítására.

A fenti lépéssorozatot folytassuk az összes lehetséges szomszédos cella-párra ugyanígy, ez lesz egy javítási kör. Mindaddig folytassuk a javítási körök futtatását, amíg egy teljes kör lefutása során sem tudunk javítani a struktúrán, hiszen ilyenkor biztosítva van, hogy a rákövetkező körben sem sikerül már javítanunk.

Előfordulhat olyan eset a globális optimum felé vezető úton, a LA struktúrák költségterében, hogy néhány lépésben rontani kellene a számított költségen. Az újrendező algoritmus azonban minden lépésben csak javítani tud a fennálló állapoton, ezért a lokális minimumokat nem tudja elkerülni, így azokban megáll. Ezen a ponton tehát eltér a szimulált lehűtés elvétől, ahol bizonyos valószínűséggel visszalépés is történhet.

## 7. Hogyan épül fel a LA tervező program?

A program két részből áll:

- A mobilitási szimulátor
  - Megadott bázisállomás-elrendezésre és úthálózatra szimulálja a mobil terminálok mozgását és a hozzájuk beérkező hívásokat.
  - A program szimulátorával kialakíthatunk egy tetszőleges szimulációs környezetet, melyben a mobil terminálok mozognak. A szimulációs környezet úthálózatból és cella-struktúrából áll össze, melyeket egy térképre lehet lefektetni.

## Location Area tervezés nextgen mobil hálózatokban – Újgenerációs hálózatok szakirány labor

- A LA tervező algoritmusokat megvalósító függvények
  - Futtathatjuk és összehasonlíthatjuk a LA tervező algoritmusokat.
  - Az egyes függvények az adott algoritmus által kialakított struktúrát, valamint annak költségértékeit adják vissza