



Szenzorhálózatok és alkalmazásaik

Útvonalválasztás (elosztott, hierarchikus routing)

Tartalom

- Útvonalválasztás
 - Routing tervezési kérdések
 - Hálózat és routing modellezése
 - **Útvonalválasztó protokollok**
 - Elosztott (flat)
 - Hierarchikus
 - Elhelyezkedés alapú (location based)

Elosztott (flat) routing

Elosztott (flat) routing (ism.)

- A hálózat csomópontjai **egyenrangú** szerepet töltenek be.
- A szenzorok **együttműködnek** a feladat ellátásában.
- A node-ok nagy száma miatt nem mindig lehetséges önálló ID hozzárendelés. => **adatközpontú** útvonalválasztás

Adatközpontú útvonalválasztás:

- A BS lekérdezéseket küld bizonyos területek felé, és várja a választ az adott területen elhelyezkedő szenzoroktól.
- **Attribútum alapú címzés** szükséges a várt adatok típusának meghatározásához.

Elosztott routing algoritmusok

- SPIN – Sensor Protocols for Information via Negotiation
- Irányított diffúzió (directed diffusion)
- Pletyka (rumor) routing

SPIN

- SPIN – Sensor Protocols for Information via Negotiation
- **Minden adatot szétterjeszt** az összes node számára, így a keresett információ bármelyik csomóponttól azonnal megkapható.
- Egy protokollcsalád, amely
 - Képes adaptálódni a node-ok erőforrásaihoz.
 - **Előzetes adat-egyeztetésen** alapul.
 - Idővezérelt.
- Alapötlet: A közeli szenzorok nagyon hasonló (vagy azonos) adatokkal rendelkeznek, így elég csak azokat az adatokat terjeszteni, ami még nincs meg a szomszédoknál.

SPIN (folyt.)

Meta-adat (meta-data):

- Az adathoz hozzárendelődik egy **leíró** (meta-data), amely pontosan leírja az adatot.
- A tényleges adatcsere előtt a meta-adatok alapján egyeznek meg a node-ok a kommunikáció szükségességéről, így a redundáns adatátvitel elkerülhető.
- A meta-adat szemantikája alkalmazásfüggő, a SPIN-ben nem specifikált.
 - PI: Egy szenzor egyedi ID-je használható meta-adatként, amely megadja, hogy a szenzor milyen területet figyel meg.

SPIN (folyt.)

- A SPIN protokoll 3 állapotú:
 - (1) adathirdetés: **ADV** / (2) adatkérés: **REQ** / (3) adatküldés: **DATA**
- A protokoll működése:
 - Ha egy szenzor új adat birtokába kerül, meghirdeti annak meta-adatát egy **ADV** üzenetszórással.
 - Ha egy szomszédja érdeklődik az adat iránt, egy **REQ** üzenettel jelzi azt, amit a forrás elküld egy **DATA** üzenetben.
 - Ha a szomszéd megkapta a (számára új) adatot, elkezd hirdetni azt a szomszédainak. (GOTO 1)
- Az üzenetküldések eredményeképp egy idő után az adat az egész hálózatban szétterjed.

SPIN (folyt.)

- A SPIN protokollcsalád variánsai:
 - **SPIN-1:** Id. fent
 - **SPIN-2:** Az 1-es verzió kiegészítve egy küszöbértéken alapuló erőforrástudatos tárgyalási fázissal: Ha túl kevés az energia, nem vesz részt minden üzenettovábbításban.
 - **SPIN-BC:** Üzenetszórásos csatornákra optimalizált
 - **SPIN-PP:** Pont-pont (hop-by-hop) kommunikációra optimalizált
 - **SPIN-EC:** A SPIN-PP egy variánsa, energia-heurisztika hozzáadásával.
 - **SPIN-RL:** Sokat hibázó („lossy”) csatornára optimalizált.

SPIN (folyt.)

Előnyök:

- A klasszikus elárasztással szemben az átvitt adatmennyiség kisebb a meta-adatok egyeztetésének köszönhetően. (nincs „adatrobbanás”)
- Az energia-adaptivitás energiahatékony működést eredményez.
- Jól alkalmazható pl. akkor, ha a szenzorok mozoghatnak.
 - Nincs szükség csak a közvetlen szomszédok ismeretére.

Hátrányok:

- Az adatok nem biztos, hogy eljutnak azokhoz a node-okhoz, amelyek érdekesnek találnák azt.
 - Pl. Ha a közvetlen szomszédok nem érdekeltek, akkor nem fogják elkérni az új adatokat.

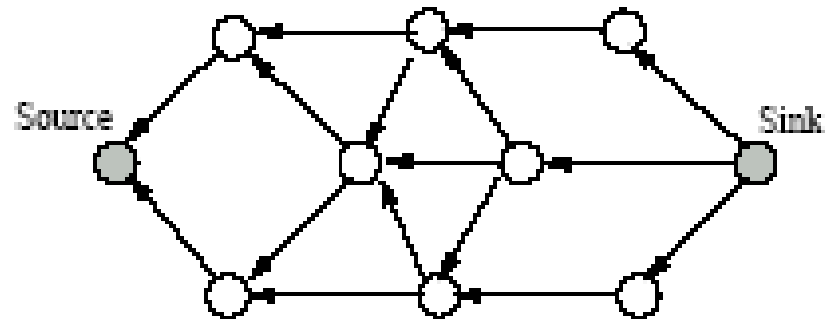
Irányított diffúzió

- DD - Directed Diffusion
- **Adatcentrikus (DC)** és alkalmazástudatos protokoll.
 - A szenzorokban keletkező adatokat egy **attribútum-érték párossal** írja le.
 - Több forrástól továbbít adatokat **egyetlen célállomás** (BS - bázisállomás) felé.
 - A folyamatok összefogása lehetőséget ad az **aggregálásra!**
- Alapötlet: A különböző forrásoktól származó adatokat a hálózaton belül aggregáljuk.

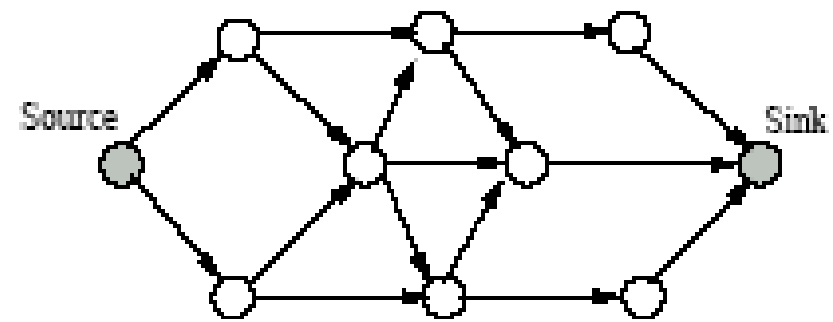
Irányított diffúzió

- A kommunikáció menete:
 - Ha a BS-nek információra van szüksége, egy kérést küld a hálózatba üzenetszórással.
 - kérés = egy hálózat által végrehajtandó feladat
 - A kérés végig „diffundál” a hálózaton lépésről lépésre, minden állomás továbbítja azt szomszédainak (hop-by-hop).
 - A node-ok gradienseket állítanak fel:
 - gradiens = attribútum-érték pár + irány
 - Pl. minden állomás arra állítja a gradiensét, ahonnan a kérés érkezett hozzá.
 - A gradiens nagysága a szomszédoktól függően más-más lehet.
 - Adatküldés visszafelé a legnagyobb gradiens irányába.
 - A nyelő felé haladva az adatok aggregálhatók.

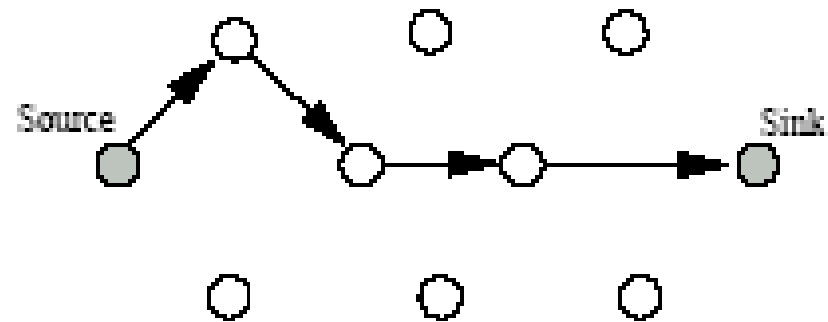
Irányított diffúzió



(a) Propagate Interest



(b) Set up Gradients



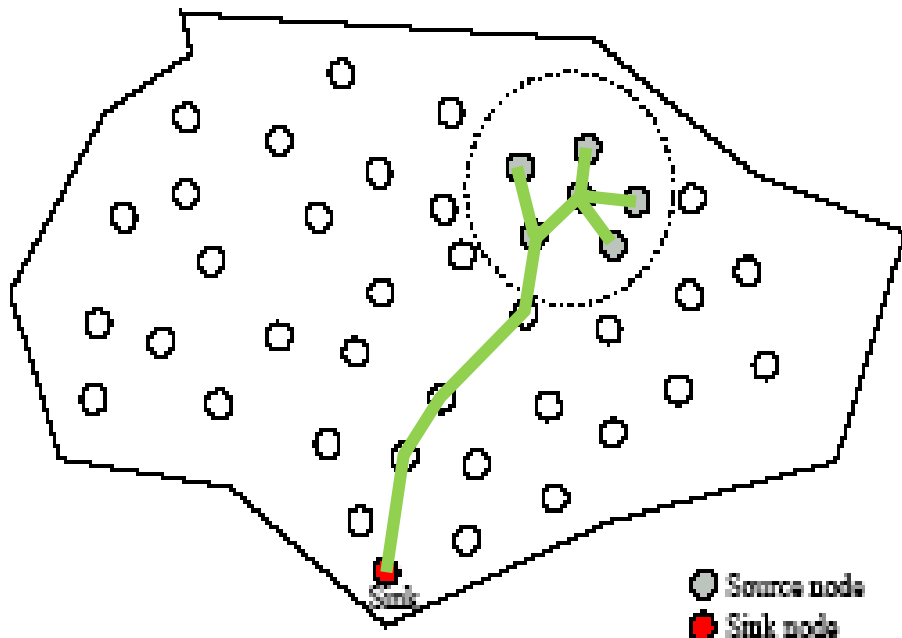
(c) Send data and path Reinforcement

3. ábra. Példa kérés terjesztésre szenzorhálózatban.

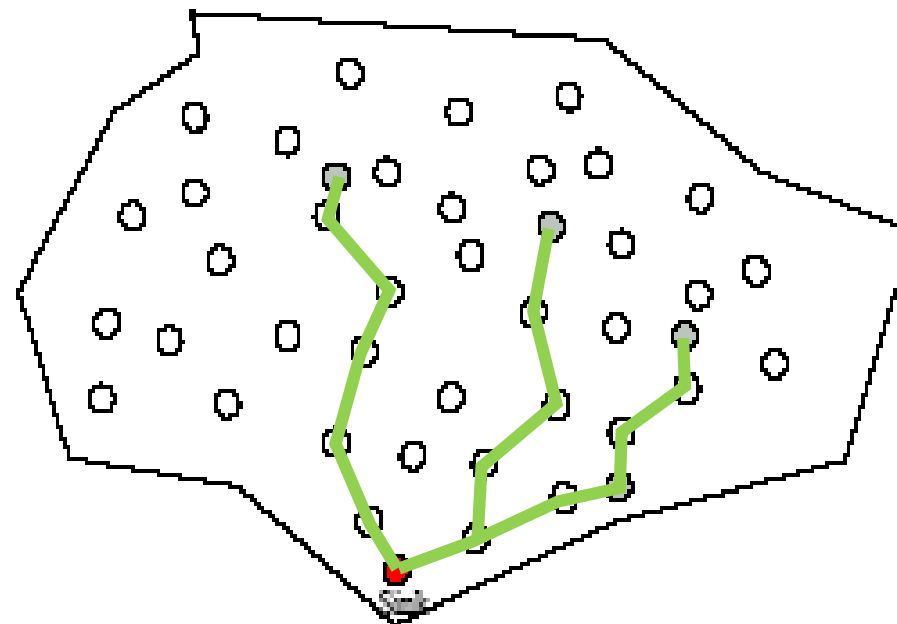
Irányított diffúzió

- A BS a kérést periódikusan ismételteti akkor is, amikor elkezd az információt begyűjteni.
 - Pl. ha nem mindig garantált az információ továbbítása.
- A cél egy olyan „hatékony” fa kiépítése, amely mentén az információ aggregálható.
- Az adat aggregáció hatékonysága függ:
 - a node-ok elhelyezkedésétől, számától;
 - a hálózati kommunikációs topológiától;
 - az eseménymodelltől.

Irányított diffúzió - példa



(a) Event Radius Model



(b) Random Source Model

□ Eseménymodellek:

- Esemény-sugár (ER – Event Radius)
- Véletlen források (RS – Random Sources)

□ Az ER modell esetén az aggregálás jobb hatásokkal végezhető el

Irányított diffúzió

Előnyök:

- Nincs szükség a teljes topológia ismeretére.
- Az adatforgalom csak igény szerinti. 😊

Hátrányok:

- Az adatforgalom csak igény szerinti. 😞
- A kérés elárasztásos módon jut el a címzetthez, még akkor is, ha az állomások csak nagyon kis csoportja érdekelt a kérés megválaszolásában.
- A kérések egyeztetése a meglévő adatokkal energiát von el.

Pletyka routing

- Pletyka (gossip) routing
- Az irányított diffúzió egy variánsa.
 - Kiküszöböli a kérések elárasztásos terjesztését.
- Alapötlet: Elárasztás helyett a kérést csak azokhoz a node-okhoz elég irányítani, amelyek az adott eseményt megfigyelték vagy tudnak róla.
- Azon alkalmazások számára hasznos, ahol...
 - az események száma kicsi, a kérések száma pedig nagy;
 - a geográfiai routing nem alkalmazható.

Pletyka routing

- Megolodás: Terjesszük az eseményt (is) ne (csak) a kérést.
 - Ha egy node egy eseményt érzékel, hozzáadja azt a lokális eseménytáblájához .
 - Létrehoz egy „hosszú életű” ügynök (agent) csomagot.
 - Az ügynökök a hálózatban utazva terjesztik az információt.
 - Egy ügynököt fogadó állomás megjegyzi azt.
 - Ha egy kérés érkezik, üzenetszórás helyett bármely node megválaszolhatja, ha ismeri az utat a forráshoz (azaz járt nála az ügynök).

Pletyka routing

- A node-ok csak egy útvonalat kezelnek a forrás és cél között, ellentétben az irányított diffúzióval.
- A kérések és ügynökök élettartama a protokoll paramétere (pl. TTL mezők)
- Az ügynökök útvonalának meghatározása nem egyszerű, és nagyban befolyásolja a teljesítményt.

Előny:

- Az üzenetszórás elkerülésével energia takarítható meg.

Hátrány:

- Csak akkor működik, ha az események száma kicsi.
 - Nagy eseményszám esetén az eseménytáblák karbantartása túl költséges, ha nincs elég érdeklődés az események iránt.

- LEACH, TEEN, APTEEN

Hierarchikus routing

Útvonalválasztási paradigmák

- Hálózati struktúra alapú protokollok
 - Elosztott (flat)
 - **Hierarchikus**
 - LEACH, TEEN, APTEEN
 - Elhelyezkedés alapú (location based)
 - egyéb...

Hierarchikus útvonalválasztás

- Hierarchikus = **klaszter alapú**
- Egy hierarchikus architektúrában...
 - a node-ok **nem** azonos szerepet töltenek be,
 - (Tipikusan) több erőforrással rendelkező node-ok foglalkoznak az információ feldolgozásával és továbbításával.
- Előnye: skálázhatóság(?), hatékony kommunikáció
- A hierarchikus routing kétrétegű:
 - Az első réteg feladata a **klaszterek formálása** és a **klaszter-vezérlő választása**.
 - A második rétegben van a tényleges útvonalválasztás.
- *A legtöbb esetben nem a routing az érdekes, hanem a klaszterek és vezérlők menedzsmentje!*

LEACH protokoll

- LEACH = Low Energy Adaptive Clustering Hierarchy
- Klaszter alapú protokoll, **elosztott klaszterképzéssel**
- A klaszter vezérlőket (clusterhead – CH) véletlenszerűen választja, és ezt a szerepet időről időre cseréli az energiamegosztás miatt.
- A **CH** a node-októl érkező adatokat **aggregálja és tömöríti**, majd **továbbítja** a BS-nek.
- TDMA/CDMA MAC a klaszteren belüli és klaszterek közötti ütközések csökkentésére.
- Az adatok gyűjtése centralizált és periódikus.

LEACH

- Feltételezések:
 - Minden node a BS rádiósugarában van.
 - Minden node elegendő energiával és számítási kapacitással rendelkezik a CH funkciók ellátásához.
 - Minden node képes CDMA és TDMA kommunikációra.
 - Minden node periódikusan küld adatot.
 - A szomszédos node-ok által küldött adatok erősen korreláltak.

LEACH

- A protokoll két fázisa:
 - **Inicializálás** (setup phase)
 - Klaszterek szervezése
 - Klasztervezérlők kijelölése
 - **Állandósult állapot** (steady phase)
 - Az adatforgalom vezérlése
- Bizonyos időnként periódikusan a hálózat visszatér az inicializáló állapotba.

Inicializálás

1. Minden node egy r véletlen számot sorsol magának. Ha r egy küszöbérték alatt van, a node klasztervezérlő (CH) lesz.
2. Minden CH hirdeti magát.
3. Minden nem-CH node a hirdetések alapján kiválasztja, hogy melyik CH-hoz akar tartozni (pl. vett jel erőssége alapján), majd értesíti erről a CH-t.
4. Minden CH egy TDMA ütemezést rendel a hozzá tartozó node-okhoz.

Állandósult állapot

1. A node-ok érzékelnek és továbbítják adataikat a CH-nak.
2. A CH az adatok begyűjtése után aggregálja azokat, majd CDMA kóddal továbbítja azt közvetlenül a BS-nek.

LEACH

Problémák:

- Nem nyilvánvaló, hogy a node-ok milyen valószínűséggel váljanak klasztervezérlővé.
 - Probléma lehet, hogy a CH-k eloszlása nem egyenletes, így egyes node-ok környezetében esetleg nem lesz CH.
- A dinamikus és periódikus klaszterképzés jelentős többlet-költséggel jár.
- A protokoll azzal számol, hogy minden körben a node-ok energiaszintje azonos.

Lehetséges kiterjesztés: Egy node az energiaszintjétől teszi függővé, hogy milyen valószínűséggel deklarálja magát CH-nak.

LEACH esettanulmány

Inicializálás

1. lépés: A klasztervezérlők kiválasztása.

- **Cél:**
 - Minden node kb. azonos ideig legyen klasztervezérlő.
 - A vezérlők egyenletesen legyenek elosztva a hálózatban.
- **Megoldás:** Az i . node a t időpontban kezdődő $(r+1)$ -edik kör elején $P_i(t)$ valószínűséggel lesz klasztervezérlő.
- Egy körben a vezérlők átlagos száma k , azaz: $E[\#CH] = \sum_{i=1}^N P_i(t) = k$
ahol N a hálózatban lévő node-ok száma

Inicializálás (folyt)

- Azt szeretnénk, ha N/k kör alatt minden node egyszer lenne klasztervezérlő.

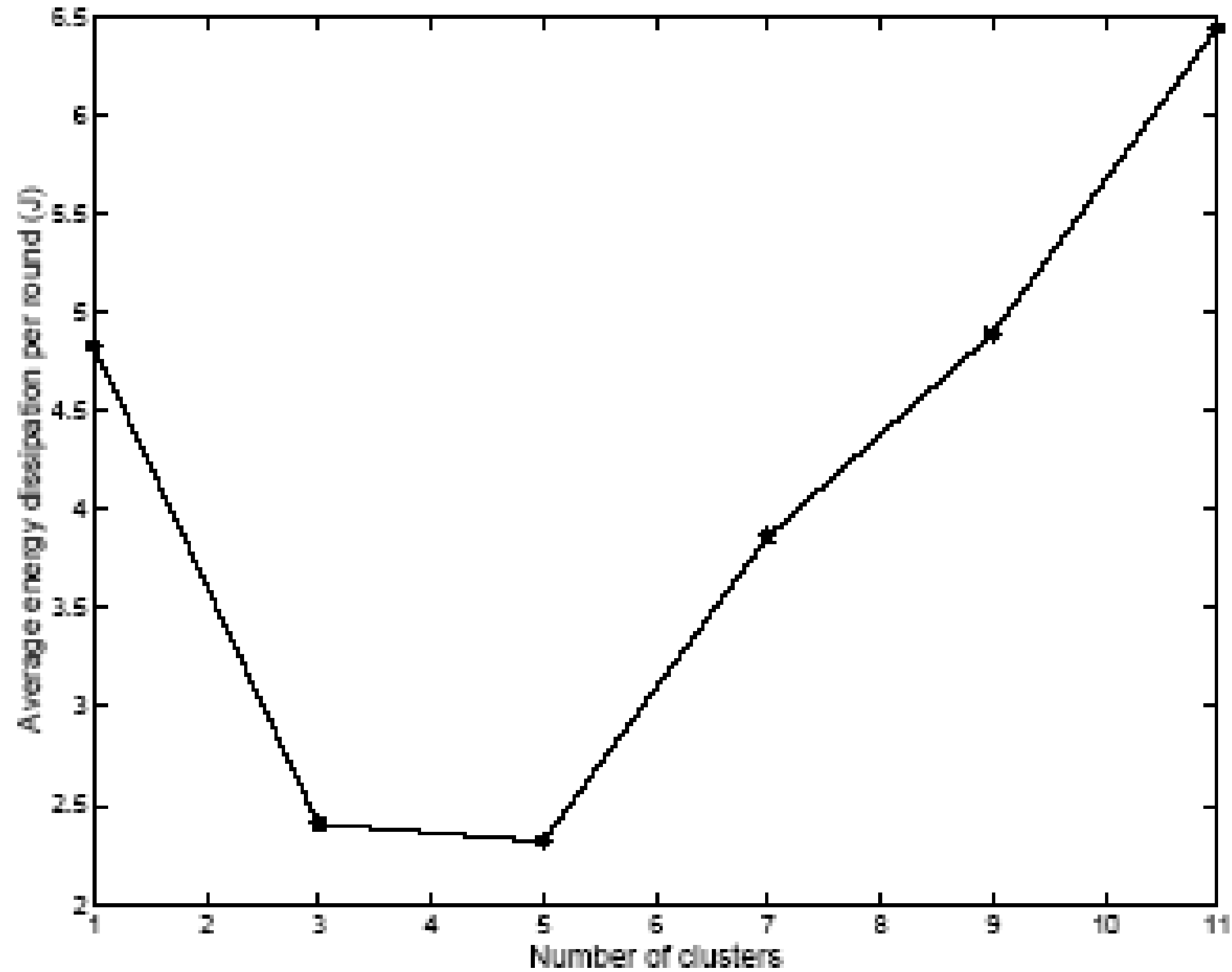
Ehhez legyen:

$$P_i(t) = \frac{k}{N - k \cdot (r \bmod (N / k))}$$

ha az i . node nem volt vezérlő az elmúlt $(r \bmod (N/k))$ körben.

- Az optimális k érték analitikusan meghatározható.
 - Túl kis k esetén messze lesznek a klasztervezérlők.
 - Túl nagy k esetén sok lesz a direkt kommunikáció a BS-sel.

Inicializálás (folyt)



Inicializálás (folyt)

- A P_i értékek számításának hátránya, hogy nem veszi figyelembe a node-ok (esetlegesen) eltérő ütemű lemerülését.
- Egy lehetséges adaptív alternatíva:

$$P_i(t) = \frac{E_i(t)}{\sum_{j=1}^N E_j(t)} k$$

ahol $E_i(t)$ az i . node aktuális energiaszintje.

- Előny: A node-ok élettartama kiegyenlítettebb lesz.
- Hátrány: Minden node-nak ismernie kell az egész hálózat aktuális energiatartalékát.
 - Ötlet: a hálózat összenergiája az adott klaszter energiaszintjéből becsülhető.

Inicializálás (folyt)

2. lépés: Klasztervezérlők (CH) hirdetése.

- Nem-perzisztens CSMA MAC protokoll használata.
- Minden CH egy ADV (advertisement) csomagot küld üzenetszórással az összes node-nak, mert:
 1. így - a CSMA miatt - nincs rejtett terminál probléma;
 2. biztosítja, hogy senki ne maradjon vezérlő nélkül véletlenül.
- Az ADV csomag tartalmazza a node ID-jét, valamint a „bejelentés” üzenetet.

Inicializálás (folyt)

3. lépés: Klasztervezérlő választás

- Minden nem CH node a hirdetések alapján kiválasztja, hogy melyik CH-hoz akar tartozni (pl. vett jel erőssége alapján).
- Egy csatlakozás-kérés (Join-REQ) üzenetet küld a választott vezérlőnek, nem-perzisztens CSMA-t használva.
 - Join-REQ tartalma: node ID; CH ID, üzenet típusa
- A Join-REQ üzenethez a node-ok nagy energiát használnak!
 - Így elkerülik a rejtett terminál problémáját.
 - Nincs RTS-CTS jelzésre szükség.

Inicializálás (folyt)

4. lépés: TDMA ütemezés kiosztása.

- Minden CH a hozzá tartozó klaszterben TDMA időszeleteket rendel a node-okhoz.
 - Előny: A node-ok a saját időzésükön kívül „sleep” módba kapcsolva energiát spórolnak meg.
- Az időzítés információkat tudatja mindenkivel.
- A hálózat átlép *állandósult állapotba* egy előre meghatározott fix ideig.

Inicializálás - folyamatábra

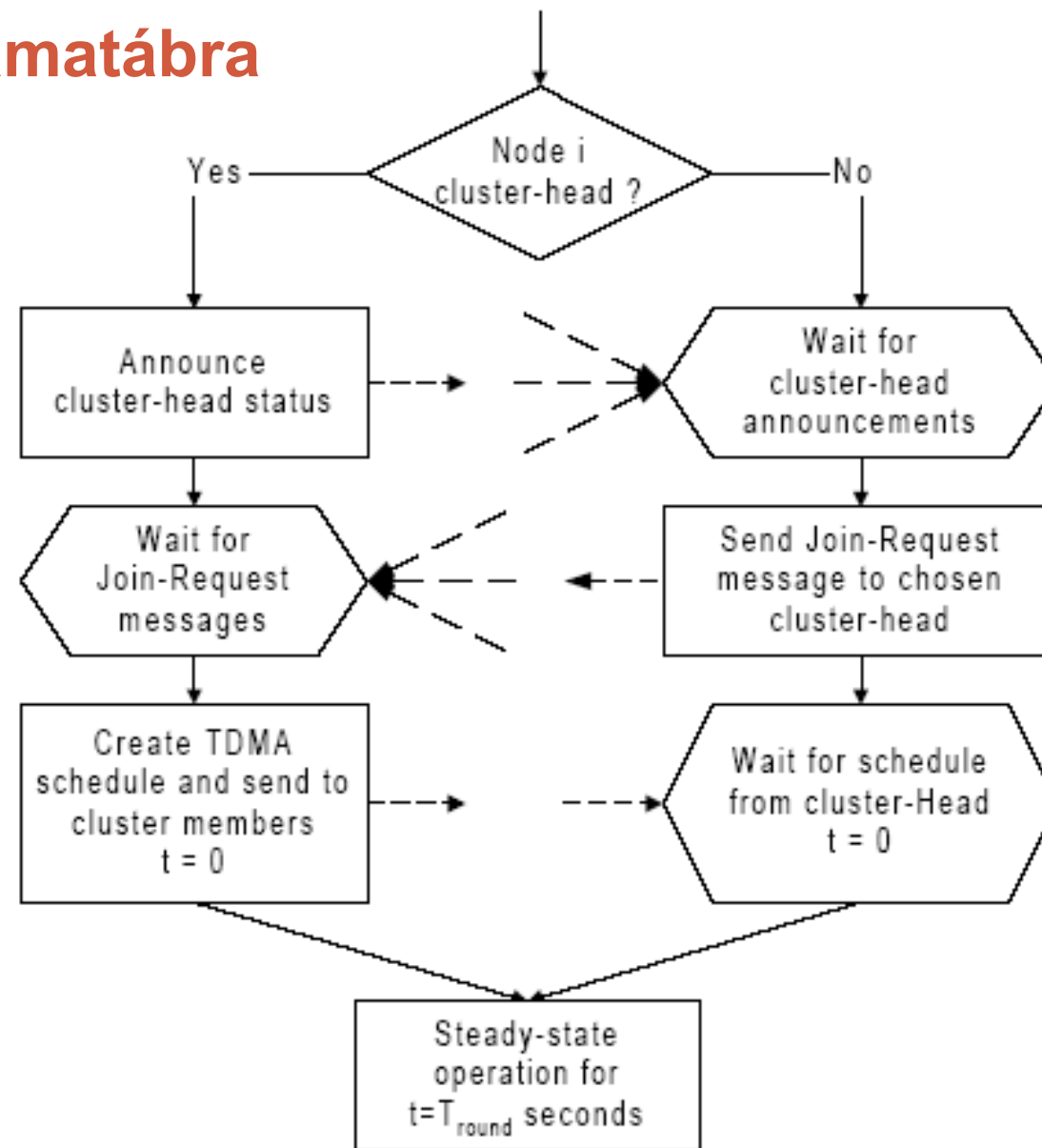
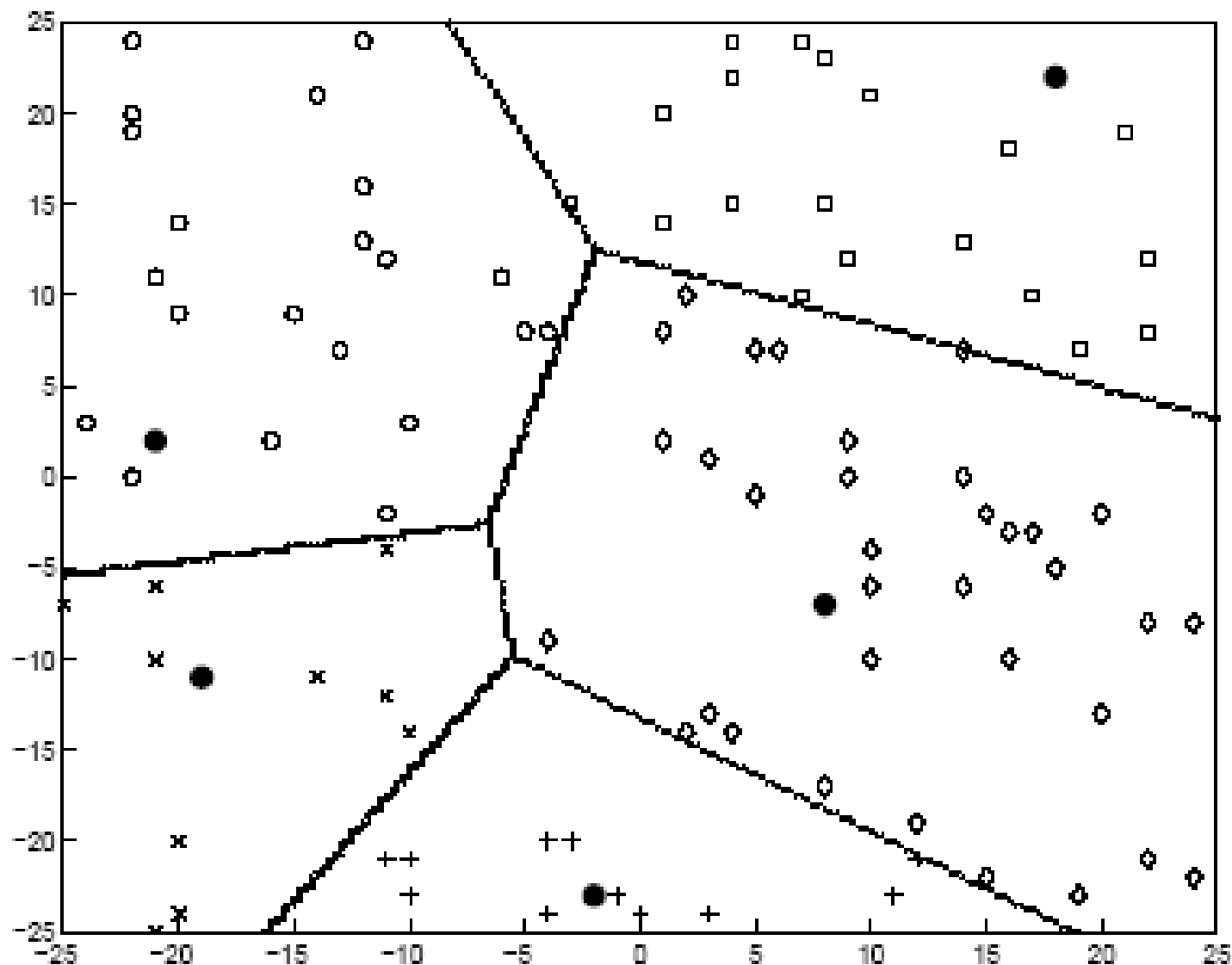


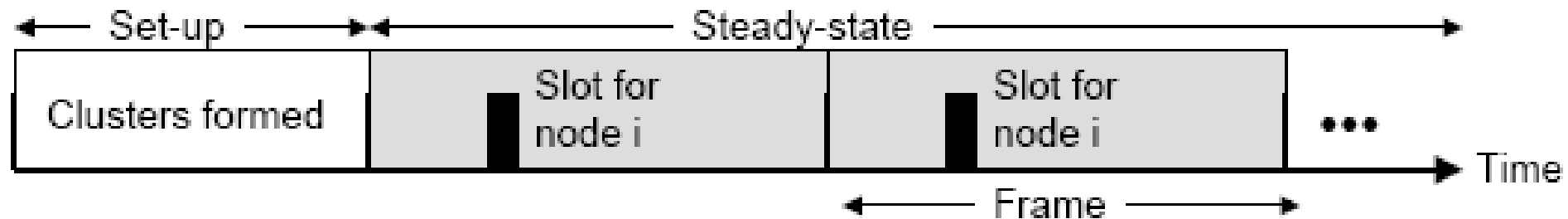
Figure 3-3: Flow-graph of the distributed cluster formation algorithm for LEACH.

Inicializálás – klaszterformálás példa



Állandósult állapot

- A kommunikáció keretekre bontva működik:

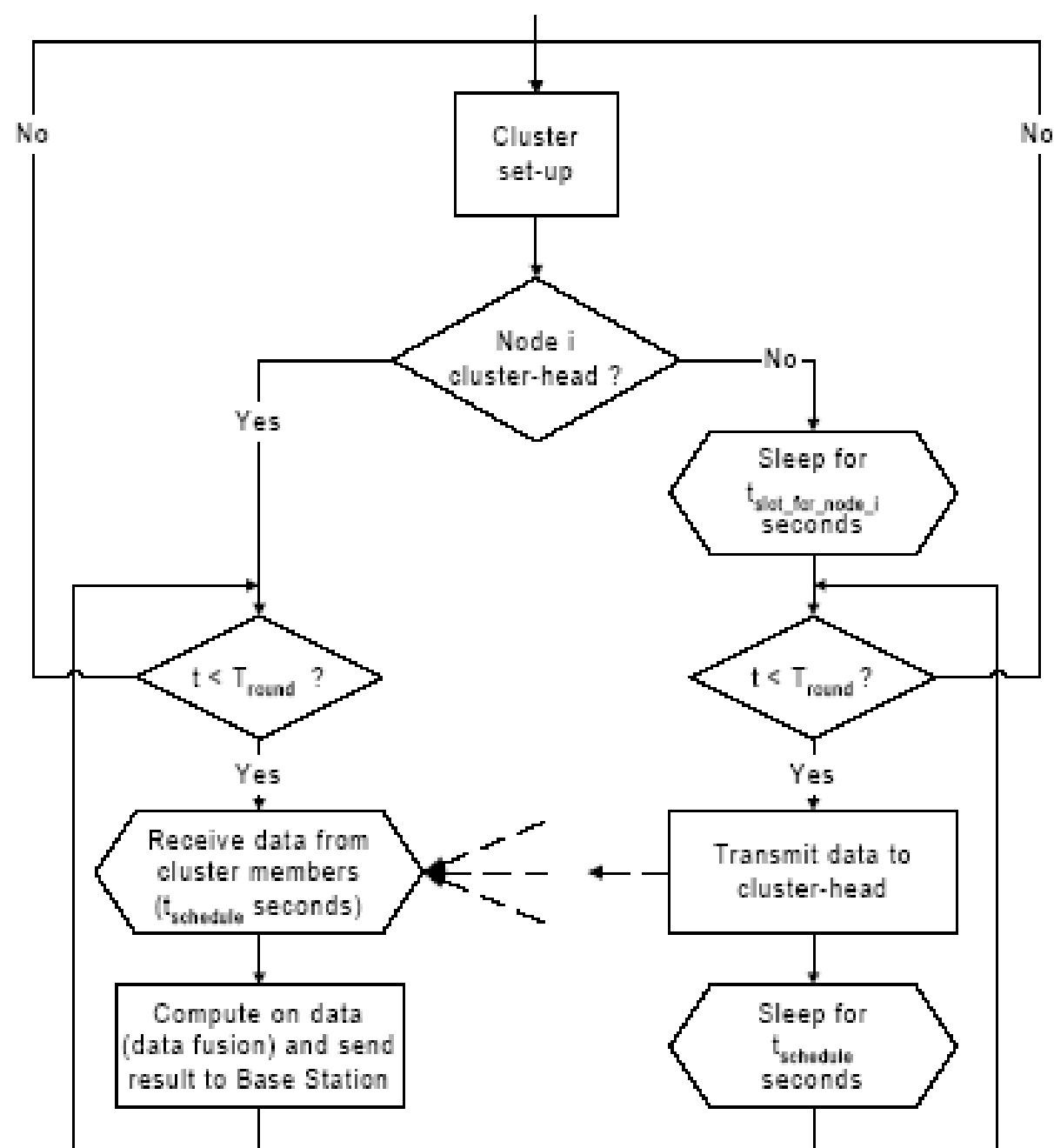


- A node-ok keretenként max egyszer küldhetnek adatot a CH-nak a számukra allokált fix időrésben.
- Egy keret hossza a klaszterben levő node-ok számával egyenesen arányos.
 - Átlagosan k node van egy klaszterben, de az aktuális érték ettől nagyon is eltérhet!

Állandósult állapot (folyt)

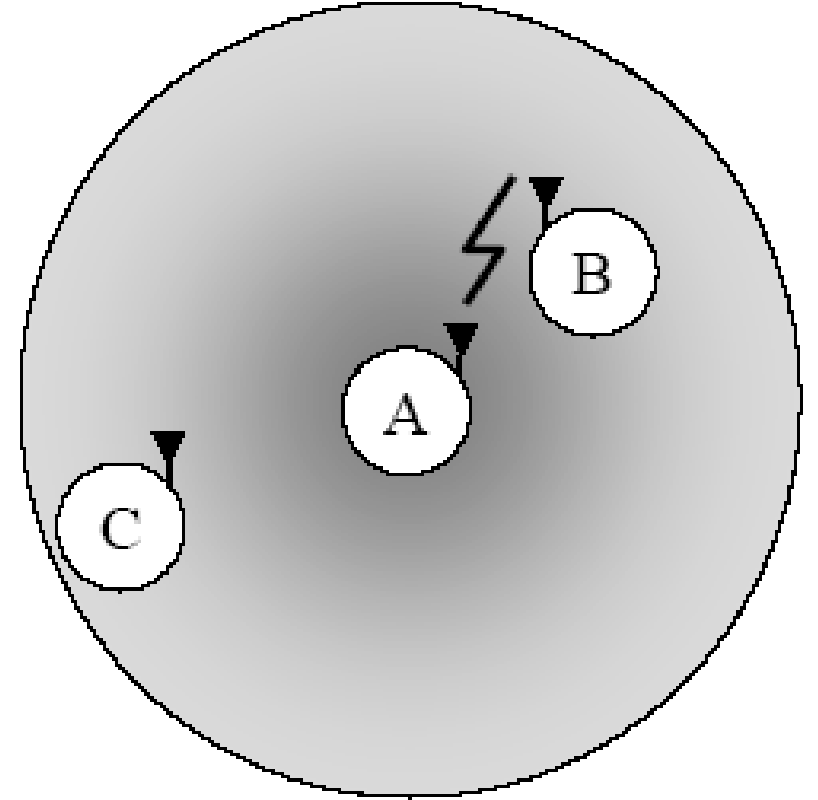
- A nem-CH szenzorok
 - teljesítményszabályozást használnak a CH-tól vett jel erőssége alapján;
 - két allokált időrés között kikapcsolják a rádiójukat.
- A CH node
 - folyamatosan bekapcsolva tartja a rádióját;
 - keretenként összegzi és előfeldolgozza a begyűjtött adatokat;
 - továbbítja az információt a BS-nek. (Nagy energia szükséges!)

Állandósult állapot



Állandósult állapot

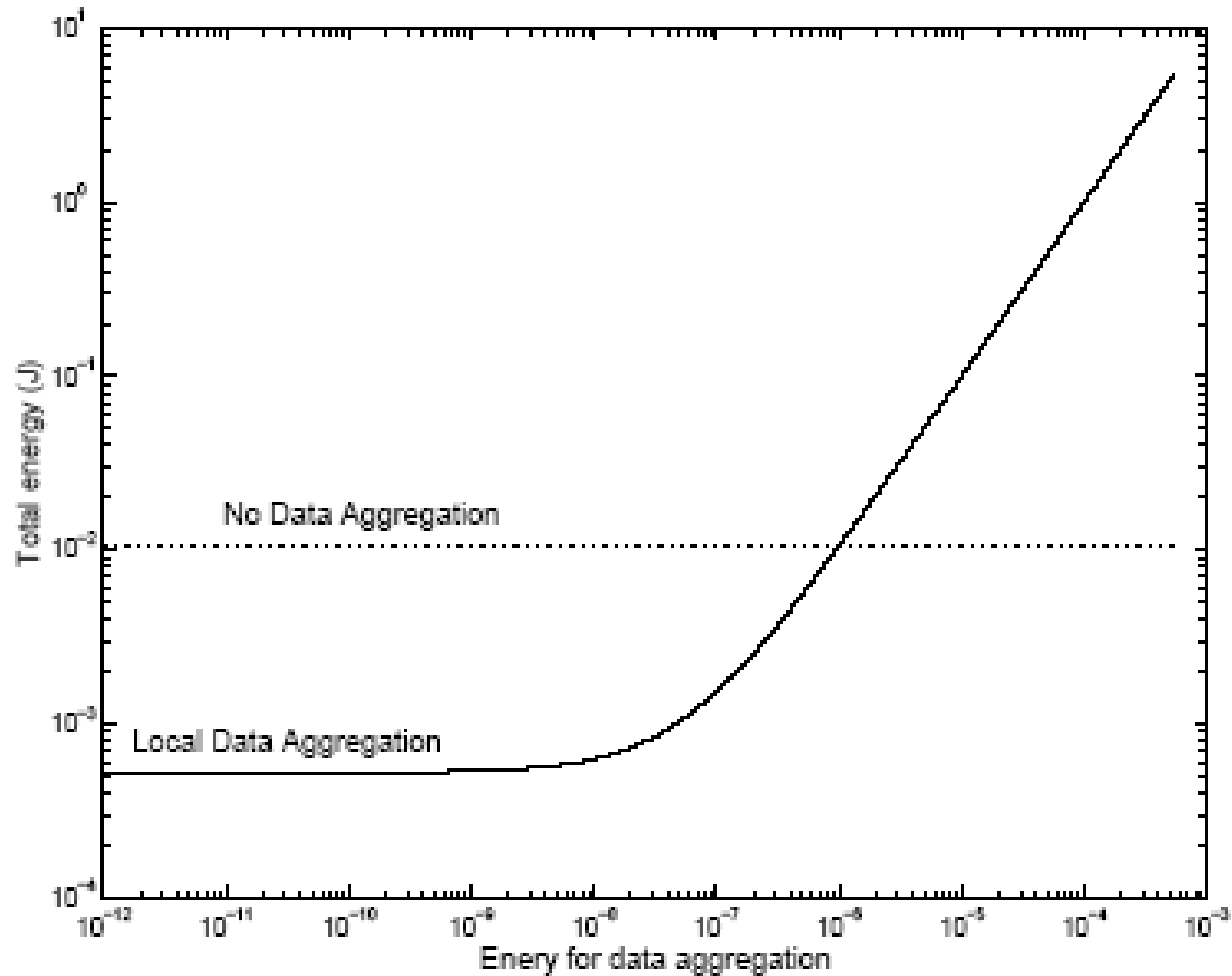
- Probléma: Egy klaszteren belüli kommunikáció hatással van a környező klaszterek rádiós adásaira is.
- Megoldás: DS-SS (direkt-szekvenciális szórt spektrum)
 - Klaszterenként különböző kódok.
 - Klaszteren belül azonos kód.
- CDMA \neq DS-SS + TDMA
 - A CH-ban csak egyetlen kódra kell figyelni!
- Az összes CH azonos kódot használva, CSMA hozzáféréssel küldi adatait a BS-nek.



Szenzor adat-aggregáció

- Az adatok feldolgozása elvégezhető:
 - A BS-ben minden adat begyűjtése után;
 - A CH-kban előfeldolgozva a klaszteren belüli adatokat.
- Az adatfeldolgozás a CH-ban megéri, ha a feldolgozás költsége kisebb, mint a megtakarított átviteli költség.
- Paraméterek:
 - 1 bit feldolgozása: E_{DA} ; 1bit továbbítása: E_{TX} ; tömörítési arány: $L:1$
- Megéri az előfeldolgozás, ha: $L * E_{DA} + E_{TX} < L * E_{TX}$
- A CH csak akkor tud hatékonyan tömöríteni, ha a szenzorok által mért adatok erősen korreláltak.

Szenzor adat-aggregáció

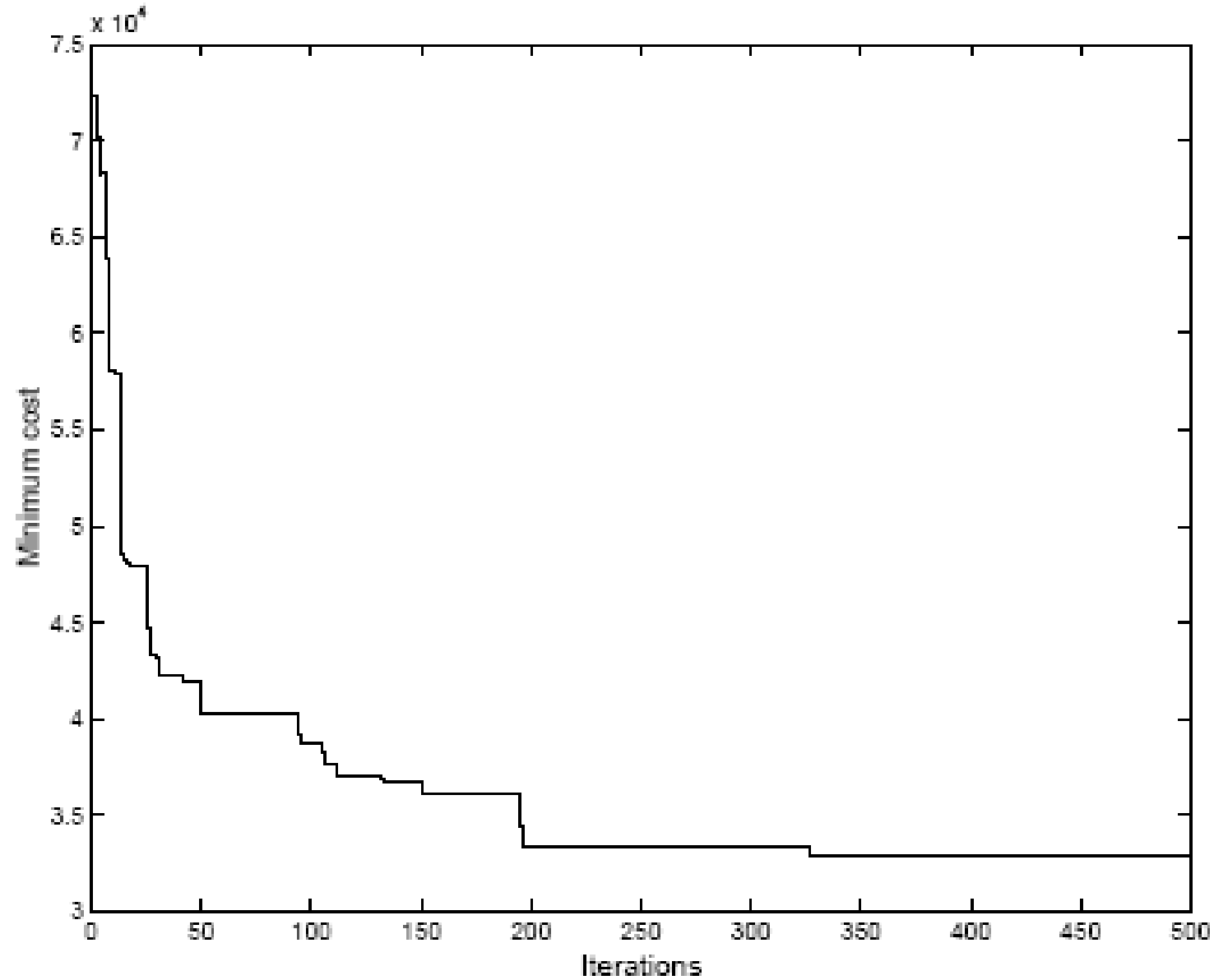


LEACH-C

- A LEACH protokoll problémája: Nincs semmi garancia a klaszterek számára és elhelyezkedésére.
 - „Statisztikailag” több körön keresztül azért „kiátlagolódik”.
- Megoldás: LEACH-Centralized
 - Az inicializáló fázist egy **központi vezérlő** végzi.
 - Az állandósult állapot nem változik.
- LEACH-C setup fázis:
 - Minden node elküldi helyzetét(!) és energiáját.
 - A BS optimalizációs algoritmussal meghatározza a CH-kat. („NP-nehéz” probléma!)

megoldás: pl. szimulált lehűtéssel

LEACH-C klaszterformálás-optimalizálás



LEACH-C

Előny:

- Optimális klaszterkiosztás (energia, BW, ...)

Hátrányok:

- Minden node körönként a BS-sel kommunikál.
- Mindenkinek ismernie kell a koordinátáit.

Teljesítményanalízis

- Vizsgált routing megoldások:
 - Közvetlen kommunikáció („**Direct**”)
 - Minimális energiájú multi-hop kommunikáció („**MTE**”)
 - **LEACH**

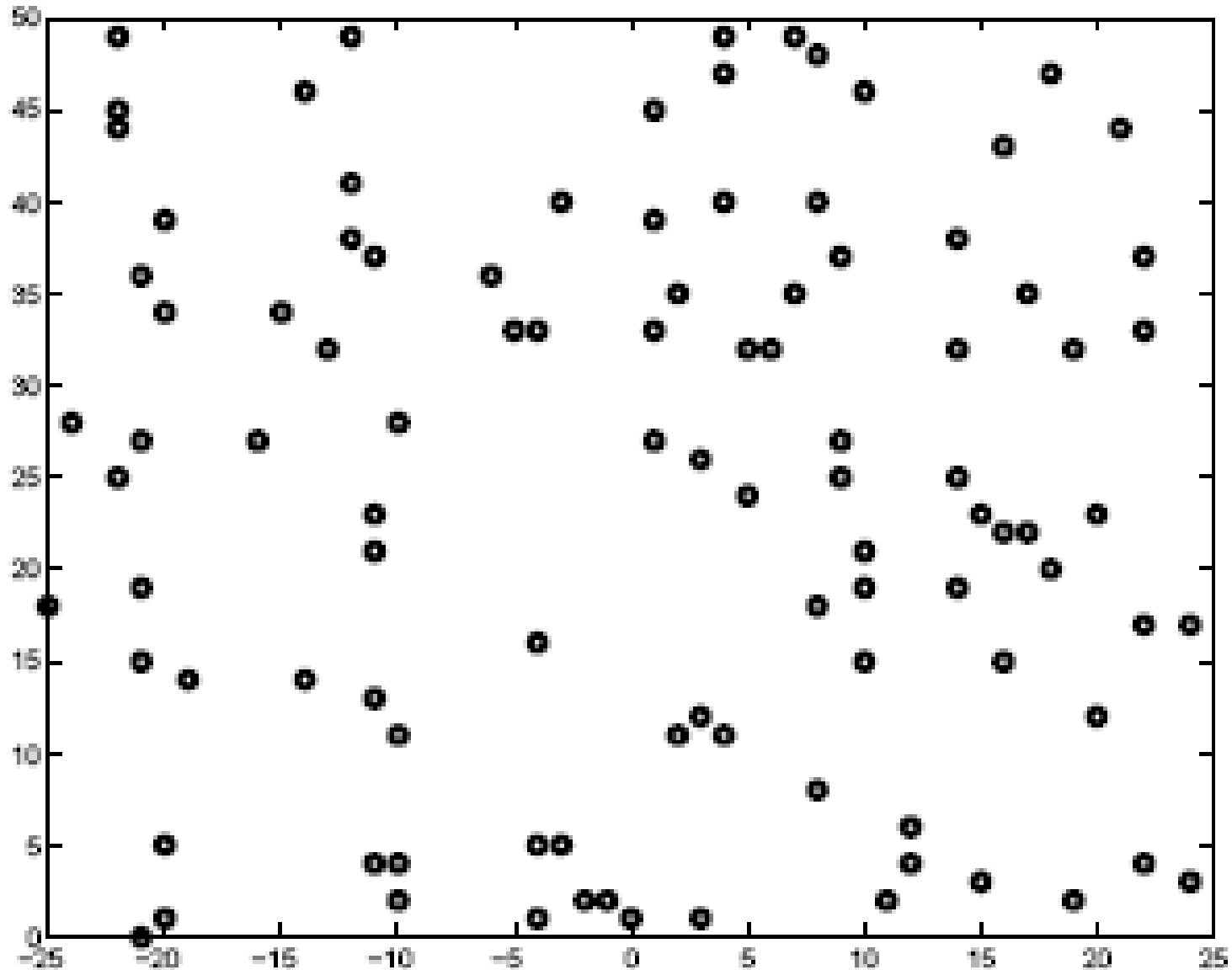
Közvetlen kommunikáció (Direct):

- Minden node közvetlenül a BS-nek ad.

MTE

- Minden node a legközelebbi szomszédjának továbbítja a csomagot a BS-felé vezető út mentén.

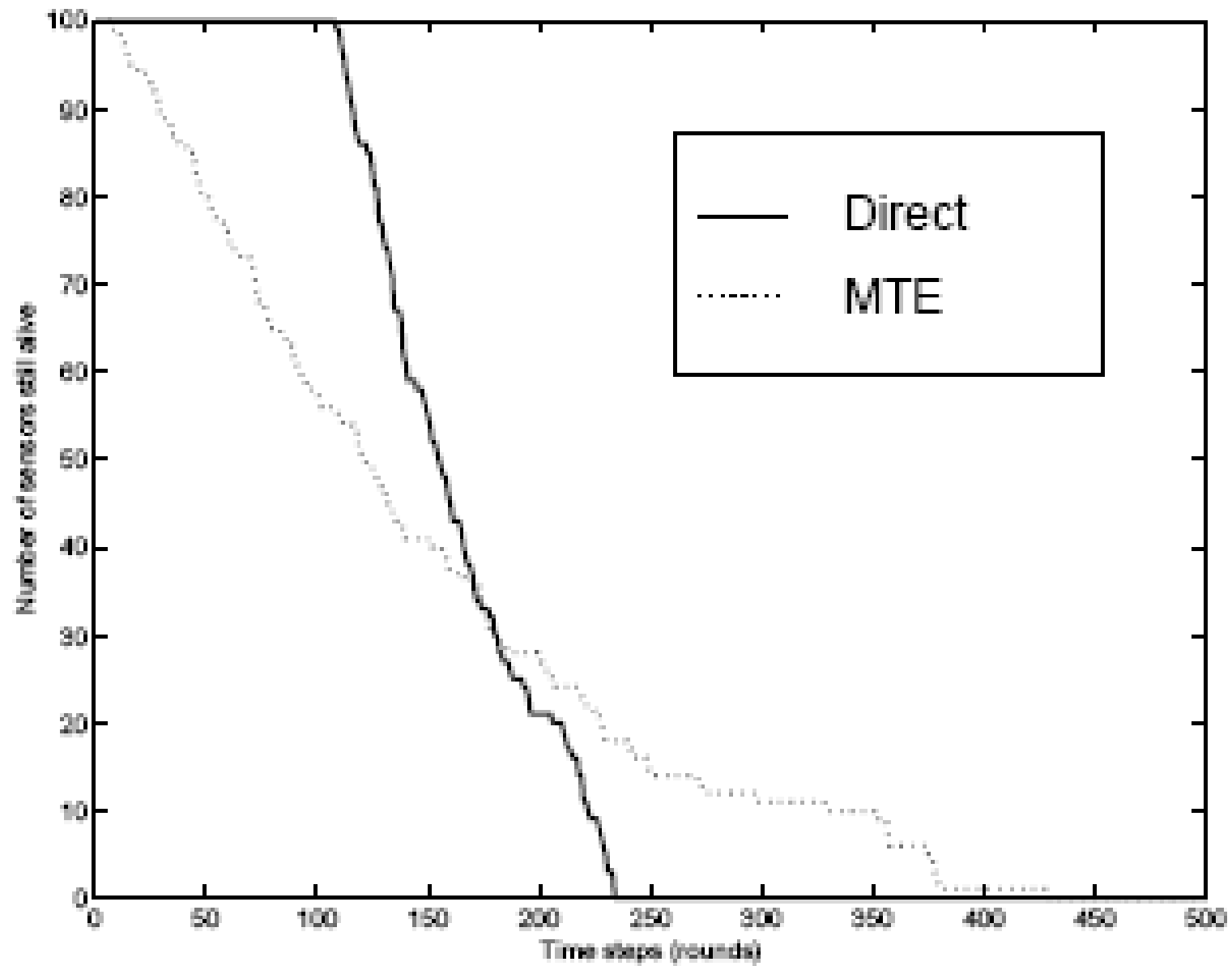
Teljesítményanalízis



vizsgált topológia

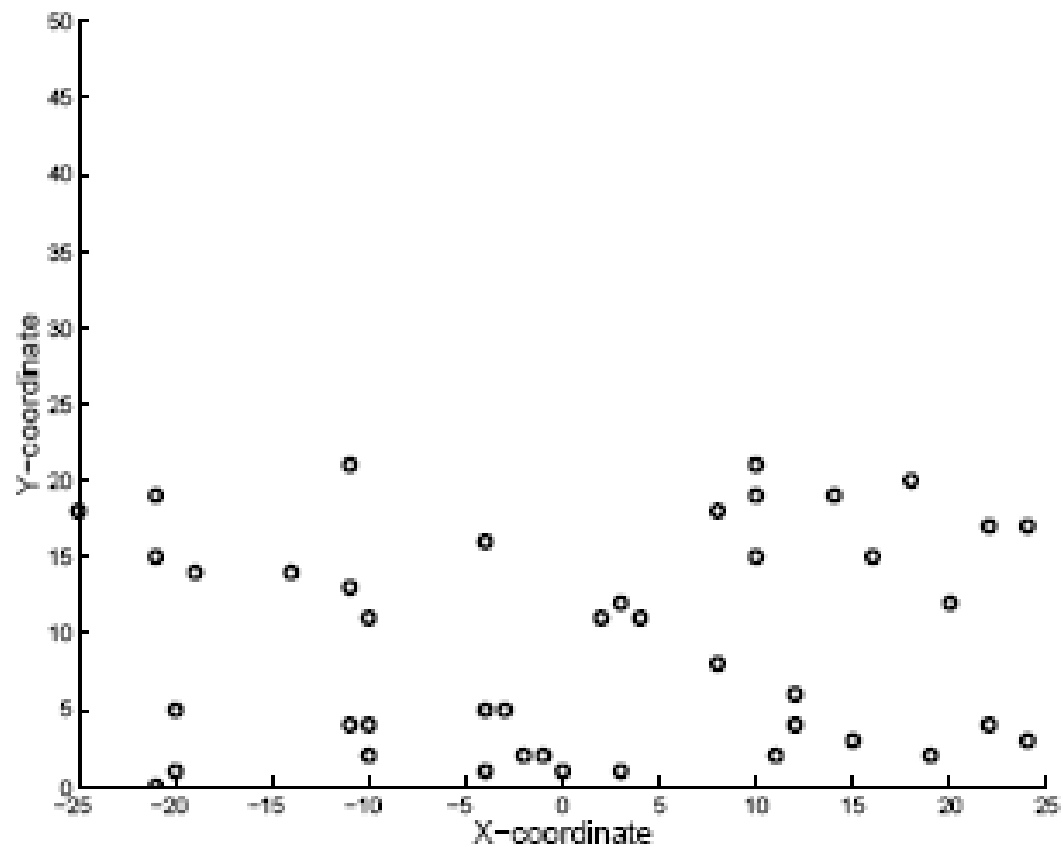
BS koordinátái:
($x=0$, $y=-100$)

Hálózat élettartama (Direct, MTE)

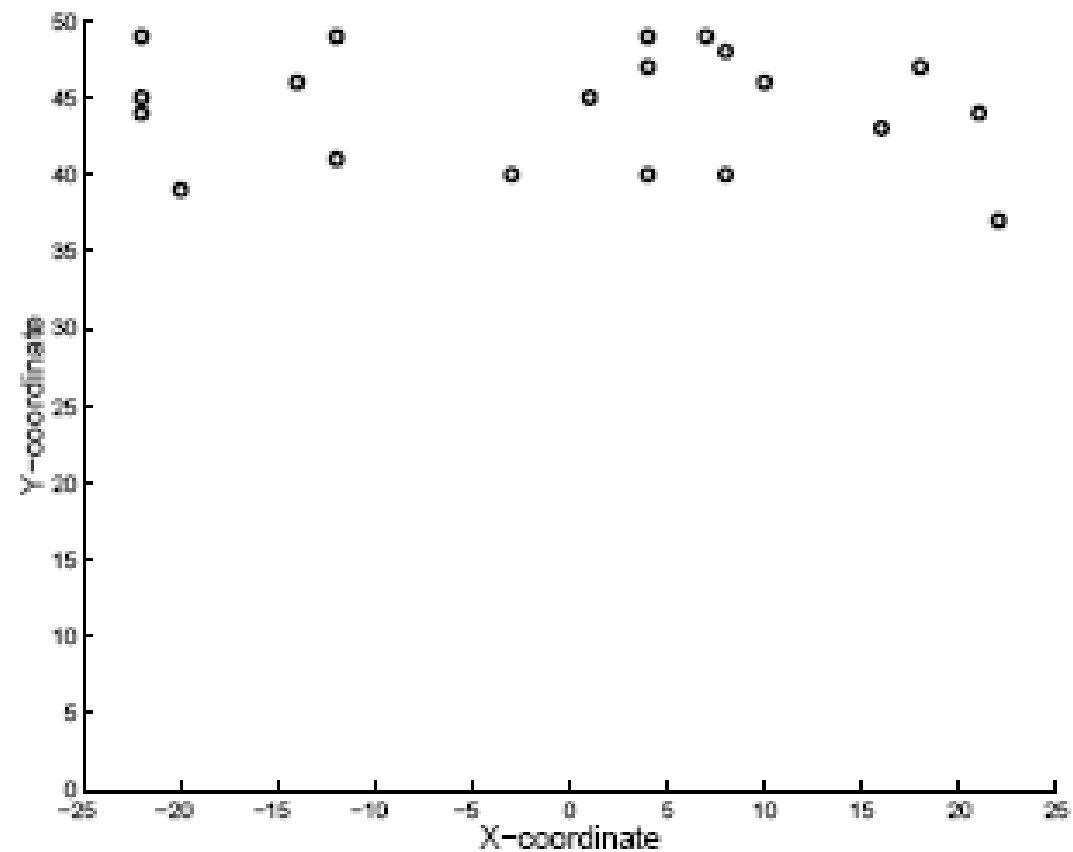


Hálózat élettartama (Direct, MTE)

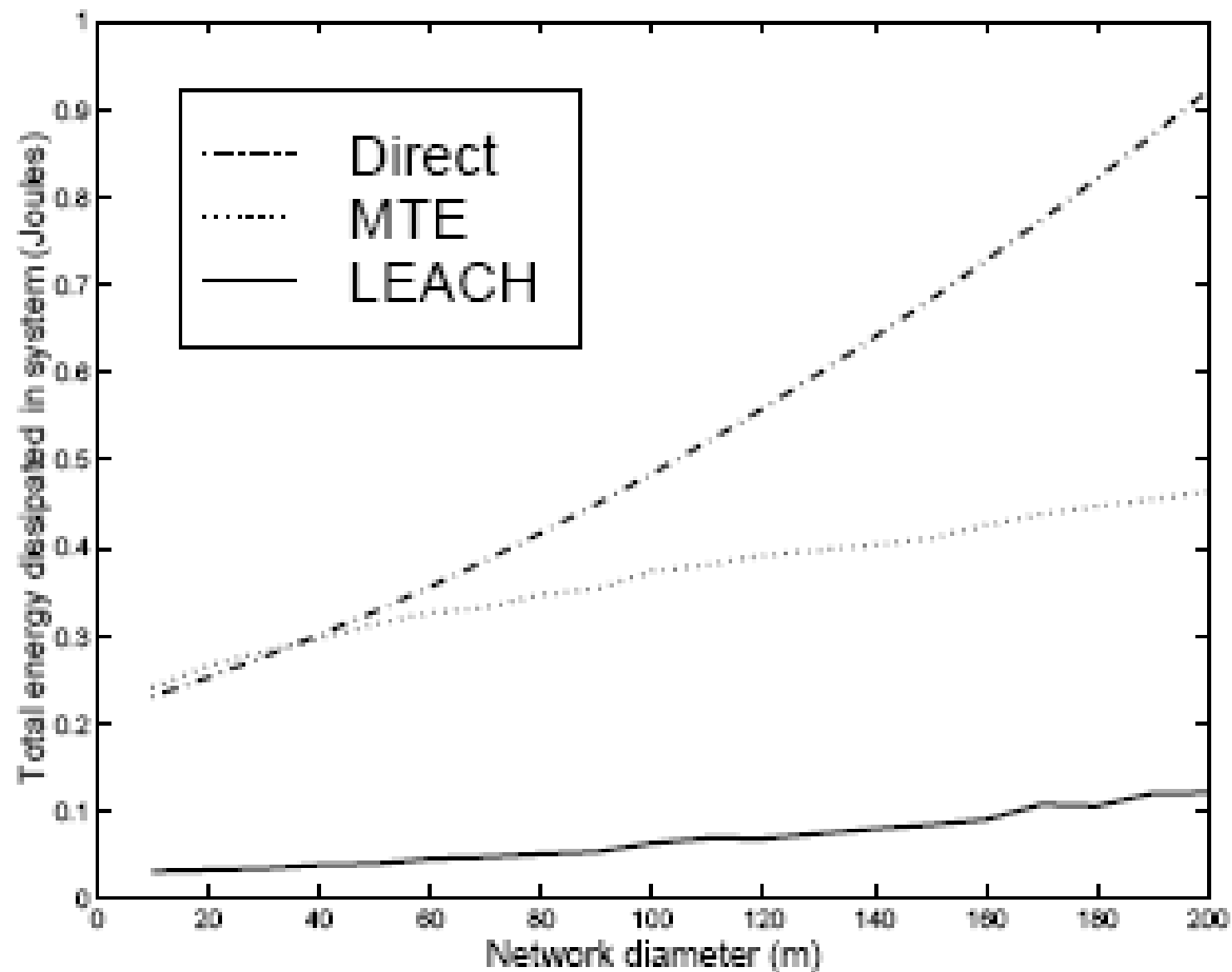
Direct:



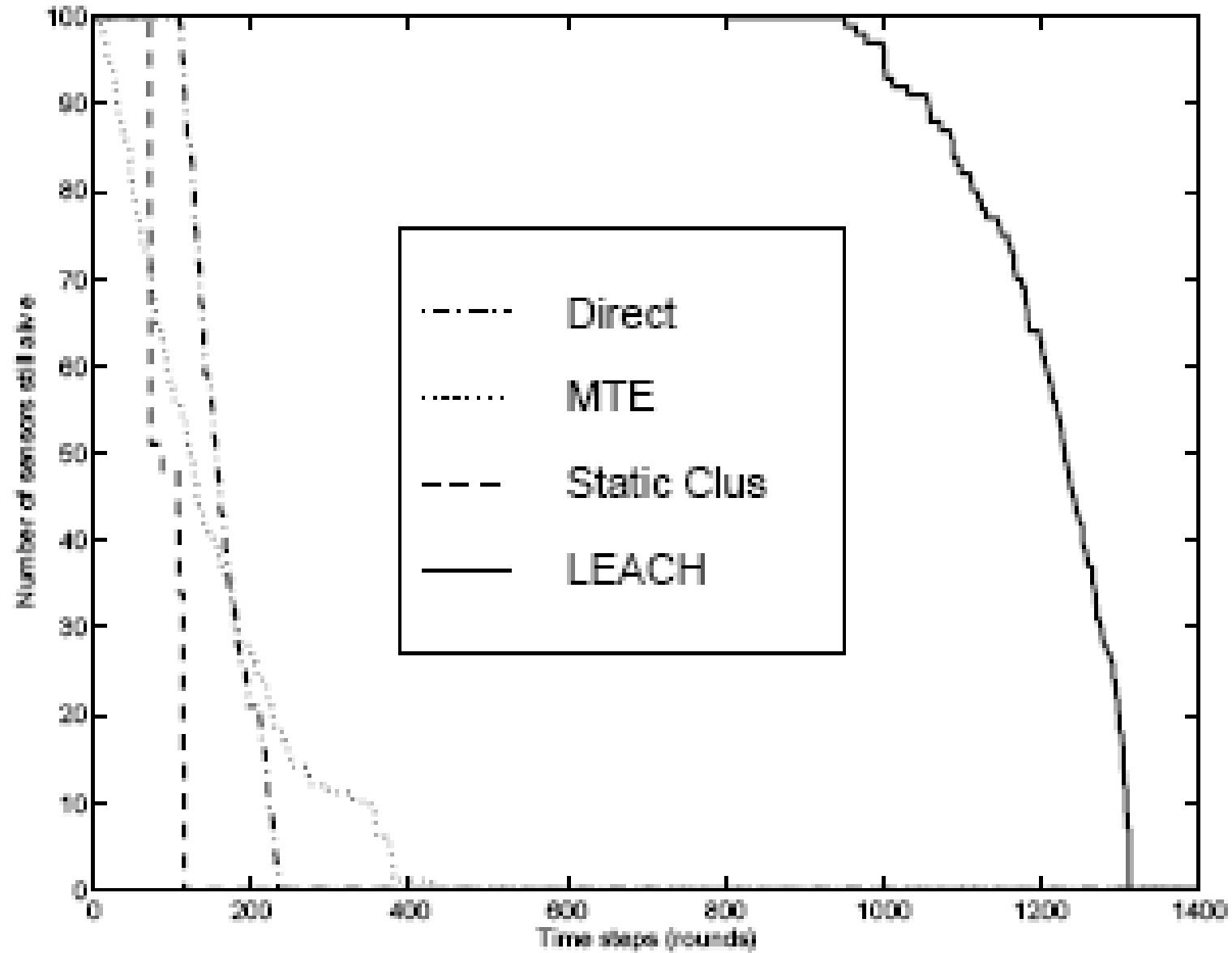
MTE:



Energiafelhasználás



Hálózat élettartama



Hálózat élettartama (2)

Table 2. Lifetimes using different amounts of initial energy for the sensors.

Energy (J/node)	Protocol	Round first node dies	Round last node dies
0.25	Direct	55	117
	MTE	5	221
	Static Clustering	41	67
	LEACH	394	665
0.5	Direct	109	234
	MTE	8	429
	Static Clustering	80	110
	LEACH	932	1312
1	Direct	217	468
	MTE	15	843
	Static Clustering	106	240
	LEACH	1848	2608

LEACH összefoglalás

- LEACH = Low-Energy Adaptive Clustering Hierarchy

A LEACH protokoll főbb jellemzői:

- Önszerveződő
- Adaptív
- Klaszter alapú, elosztott koordinációval
- Klasztervezérlők véletlenszerű cserélgetése
- Skálázható, robosztus
- Adattömörítés a globális kommunikáció csökkentésére.