

Szenzorhálózatok

Szenzor MAC (2011.10.21)

Vidács Attila
Távközlési és Médiainformatikai Tanszék
I.B.228, T:19-25, vidacs@tmit.bme.hu

Közeghozzáférési (MAC) technikák

Vezetéknélküli MAC technikák

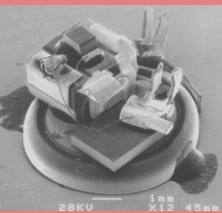
- ALOHA
- CSMA – Vivőérzékeléses többszörös hozzáférés
- Lekérdezés (Polling)

Szenzorhálózati megoldások

- WINS
- PicoRadio

MD (Mediation Device) protokoll

S-MAC



WINS megoldás

□ **Lekérdezéses** megoldás **relatív TDMA** rendszerben.

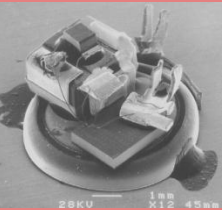
- Multi-hop rendszer.
- Nincs globális „beacon”.
- A TDMA keret aszinkron a node-ok között.
- A node-ok páronként megegyeznek egy mindkettőjük által elfogadható időrésben.

□ **Hátrányok:**

- Egy sűrű hálózatban sok aktív, véletlenszerűen allokált időrés lesz a TDMA keretben.

□ **Előnyök:**

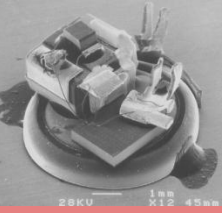
- Könnyen implementálható.
- Alkalmas valós idejű forgalom átvitelére a kontrollált késleltetés-ingadozás miatt.



PicoRadio

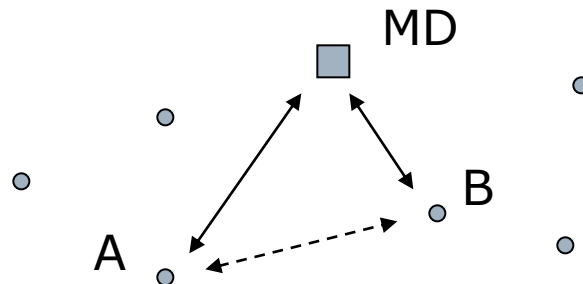
- Javaslat egy többcsatornás (~ 30), **kódosztásos** többszörös hozzáférésre (**CDMA**).
 - Minden node és a szomszédai különböző kódokat kapnak.
 - Ortogonalis CDMA kódok esetében nincs csomagütközés.
 - A node-ok aszinkron működésűek, így nincs lehetőség „alvásra”, mindig figyelni kell a csatornát.

- Ultra-low-power „wake-up radio”
 - $1 \mu\text{W}$ aktív energiafelhasználású
 - Egy egyszerű RF erősítő + szűrő + detektor
 - Figyeli a csatornát, és felébreszti a node-ot ha vesz egy „wake-up” beacon-t.



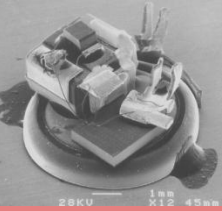
MD – Mediation Device protokoll

- Egy node a hálózat üzemideje alatt az idő 99.9%-ában „alszik” → A rövid ideg ébren lévő node-ok felfedezése és szinkronizálása a hálózatban nem triviális feladat!
- Lehetséges megoldás: **közvetítő állomás** (MD - mediation device)
 - Az MD közvetít két hálózati csomópont között.
 - Képes kontroll üzenetek rögzítésére és továbbadására.
 - A csatornát mindig figyeli, van elég energiája.



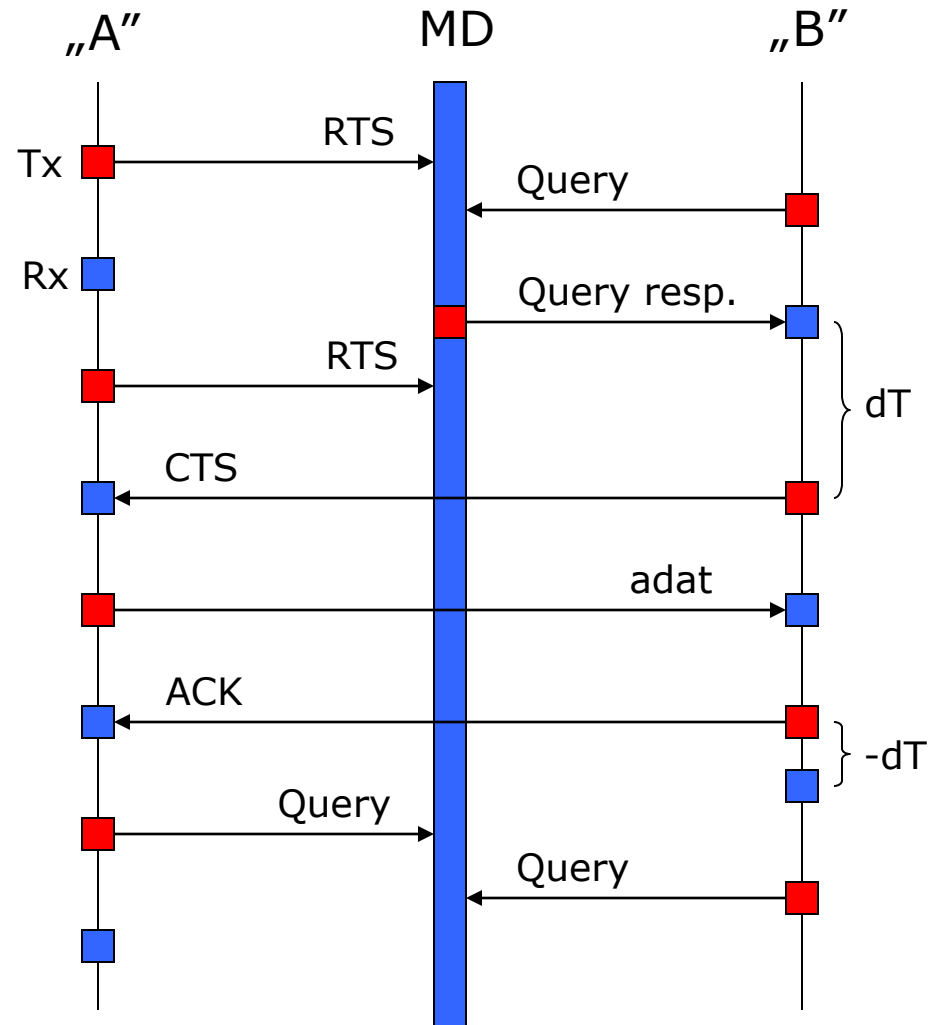
MD protokoll

- Normál üzemmódban minden node periódikusan (2 mp-enként) küld egy rövid ($< 1\text{ms}$) „beacon” csomagot az MD-nek, majd egy rövid ideig hallgatja a csatornát.
- Query-beacon: node ID, nincs küldeni való, szabad.
- Az MD veszi az összes node beacon-jét, miközben a node-ok nincsenek szinkronban egymással. (réseletlen ALOHA)
- Ha egy node küldeni akar, a query-beacon helyett RTS-beacon-okat küld periódikusan.
- RTS-beacon: node ID, cél ID



MD protokoll

1. „A” RTS beacon-okat küld MD-nek.
2. Az MD értesíti „B”-t , és elküldi „A” időszinkronját.
3. „B” szinkronizálódik „A”-ra, és a következő RTS csomag után küld egy CTS csomagot közvetlenül „A”-nak.
4. Egy ACK után kezdődik a kommunikáció „A” és „B” között.



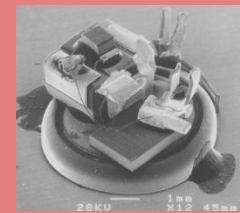
Elosztott MD protokoll

□ Az MD protokoll hátrányai:

- Minden node rádiótávolságon belül kell legyen.
- Az MD mindig be van kapcsolva, mindig figyeli a csatornát.
- A központosított rendszer nem robosztus.

□ Megoldás: elosztott MD protokoll

- Az MD szerepét a node-ok elosztottan megosztva végzik.
- Minden node időről időre átkapcsol a normál és MD üzemmód között, a többiektől függetlenül, egy véletlen változótól vezérelve.
- Amikor egy node MD módba kapcsol, egy teljes beacon periódusideig aktív marad, és begyűjti szomszédai adatait (ID, időszinkron).
- A kommunikáció felépítésében az éppen MD módban működő állomás közvetít.



Elosztott MD protokoll

□ Előny:

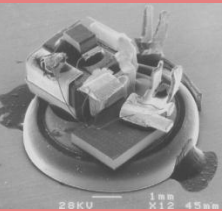
- Nincs kitüntetett MD node.

□ Hátrány:

- A késleltetés nem fix. (Adás előtt meg kell várni, hogy egy szomszéd MD módba váltson.)
- Ha egynél több node lép MD módba, a beacon csomagra mindketten válaszolnak, így ütközés lép fel.

□ Variációk:

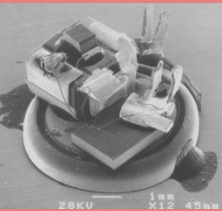
- Ha egy node MD módba lép és feltérképezi szomszédait, a ciklus végén bejelenti ezt. A további MD módban lévő node-ok hallják ezt, és...
 1. visszatérnek normál módba.
 2. továbbra is figyelik, hogy az első MD milyen beacon csomagokat nyugtáz. Lehetnek olyan állomások is, amelyeket az MD nem hall, csak ők maguk. Ebben az esetben MD-ként viselkednek a fennmaradó node-ok számára.



Megjegyzések

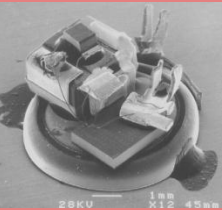
- Csatornahozzáférés az ISM sávban
 - Nem egyszerű különböző rendszerek együttélésénél!
 - A vivőérzékelés esetén nem elég a csatorna energiaszintjét mérni.
 - „Fairness” nem érvényesül, ha nem mindenki visszakozik ütközés esetén. (pl. mikrosütő?)

- Sok kérdés még megoldásra vár...



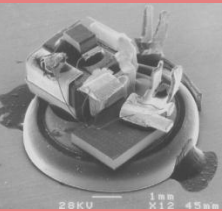
S-MAC

- Wei Ye, John Heidemann, Deborah Estrin, „**An Energy-Efficient MAC Protocol for Wireless Sensor Networks**“, *Proc. of Infocom 2002*, pp. 1567-1576, New York, USA, June 2002
- Wei Ye, John Heidemann, Deborah Estrin, „**Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks**“, *IEEE/ACM Transactions on Networking*, 12(3):493-506, June 2004



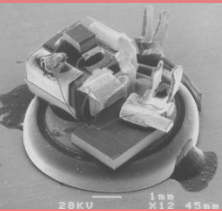
S-MAC

- S-MAC = Sensor Medium Access Control
- Cél: Szenzorhálózatokban alkalmazható MAC protokoll kifejlesztése, amely
 - energiatakarékos,
 - önszerveződő,
 - skálázható és adaptív
 - pl: A hálózat mérete, topológiája, nodesűrűsége változik.
- Megjegyzés: A node „fairness” és az alacsony késleltetés nem elsődleges követelmény.



S-MAC - feltevések

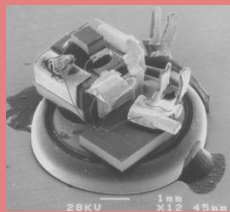
- A modellezés feltevései a szenzorhálózatról és alkalmazásokról:
 - Sok, apró, ad-hoc módon telepített csomópont.
 - Többugrásos (multi-hop) kommunikáció.
 - A kommunikáció főleg egyenrangú csomópont-párok (p2p) között zajlik. (Nem csak egyetlen bázisállomás felé.)
 - A hálózat ön-konfiguráló kell legyen.
 - Az egész hálózat egyetlen alkalmazás céljából jött létre.
 - Pl. Az Internet nem ilyen!
 - Az alkalmazás során előfordulnak hosszú tétlen időszakok.
 - Az alkalmazás bizonyos késleltetést elvisel.
 - Pl. nem valósidejű felügyeleti vagy megfigyelő rendszerek.



S-MAC: Az energiapazarlás okai

Az energiafogyasztás szempontjából kritikus kérdések:

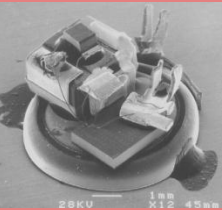
- Ütközések.
 - Ha egy elküldött csomag megsérül, azt újra kell adni.
 - (A késleltetést is növeli.)
- Áthallás.
 - Másnak címzett csomagok vétele.
- Járulékos kontroll csomagok.
 - Jelzésinformációk adása és vétele nem „hasznos” adatátvitel.
- Tétlen figyelés.
 - Várakozás esetleges adatok fogadására, csatornafigyelés.



S-MAC építőkövek

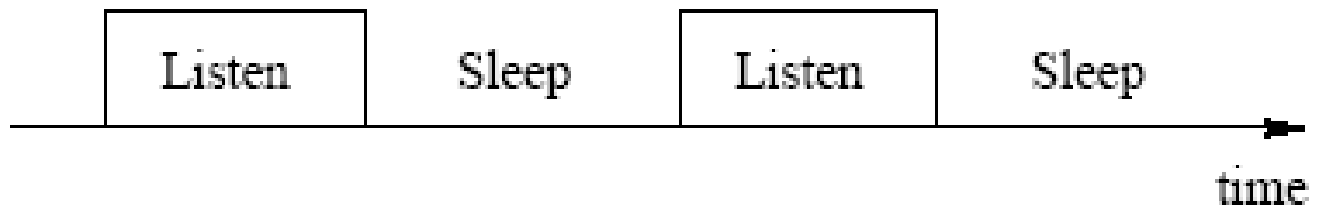
A 4 fő építőelem:

- Periódikus figyelés és alvás.
- Ütközés elkerülése.
- Áthallás elkerülése.
- Üzenet továbbadás.
 - Hosszú üzenetek hatékony továbbítására.



Periódikus figyelés és alvás

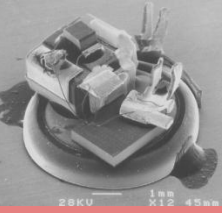
- Ha nem érzékelnek eseményt, a node-ok tétlenek.
- Mivel a nyugalmas időszakokban a forgalom kicsi, nem szükséges az összes szenzornak aktívnak maradnia.
- Az S-MAC megengedi, hogy a node-ok periódikusan alvó módba kapcsoljanak.



- Alvó (sleep) módban az eszköz kikapcsolja a rádióját, és egy időzítőt (wake-up timer) állít be a felébredéséhez.

Periódikus figyelés és alvás (2)

- Egy teljes aktív-inaktív ciklus a keret.
- Az aktív (figyelő) időtartam fix
 - A fizikai réteg és MAC réteg paramétereit határozzák meg (pl. rádiós sáv szélesség, ablakméret)
- Az alvó periódus hosszával állítható be a keret hossza, valamint az ún. „duty-cycle” érték (=aktív/inaktív arány).
 - Az aktív és alvó időtartamok a továbbiakban azonosak minden csomópont esetében (az egyszerűség kedvéért).
- Minden node megválaszthatja az aktív/inaktív ütemezését.



Szinkronizáció

- Célszerű, ha szomszédos node-ok szinkronizáltak, azaz ugyanakkor alszanak ill. figyelnek.
(= virtuális klaszter)

- *Ez persze nem mindig lehetséges multi-hop esetben!*

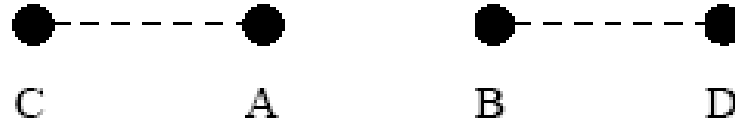
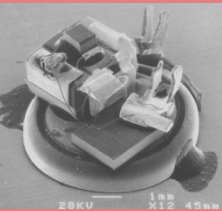


Fig. 2. Neighboring nodes A and B have different schedules. They synchronize with nodes C and D respectively.

- Szomszédos node-ok között időnként szinkronizációra van szükség az órajelek elcsúszása miatt.
- A node-ok saját ütemezésüket üzenetszórással (SYNC) juttatják el közvetlen szomszédaiknak.

Szinkronizáció (folyt.)

- Bármely két node kommunikálhat egymással akkor is, ha ütemezésük különböző:
 - Pl. ha „A” kommunikálni akar „B”-vel, megvárja, míg „B” aktív állapotba kerül.
 - Ha egyszerre többen is kommunikálni akarnak „B”-vel, versenyezniük kell a csatornáért.
- Versengés a csatornáért:
 - Az eljárás azonos az IEEE 802.11-ben használt RTS-CTS megoldással.
 - Az a node kapja a CTS csomagot „B”-től, aki elsőként küldte az RTS csomagját.
- Az adatkapcsolat létrejötte után, annak végéig a node-ok nem követik az addigi ütemezésüket. (Azaz nem mennek aludni.)



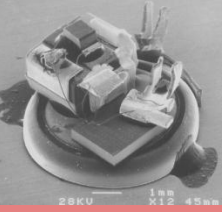
Szinkronizáció (folyt.)

Előny:


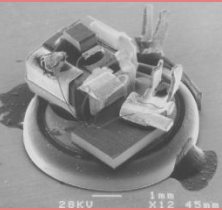

- Peer-to-peer topológia.
- Nincs szükség klaszterképzésre és koordinációra a klasztervezérlők között.

Hátrány:

- A node-ok periódikus elalvása miatt a késleltetés nagy is lehet.
- Multi-hop kommunikációnál a késleltetés arányos a node-ok számával.



Ütemezés választása és karbantartása

- 
- 
- 
- Minden node karban tart egy ütemezési táblázatot a szomszédairól.
 - Az ütemezés és az ütemezési tábla felállításához minden node a következő algoritmust használja:
 1. Egy bizonyos ideig figyel a csatornát. Ha nem hall ütemezési információt egyetlen állomástól sem, véletlenszerűen választ egy időt az alváshoz, és közli ezt egy SYNC üzenetben a szomszédaival: SYNC üzenet: „ t másodperc múlva alszom” (Ebben az esetben az állomás lesz a „szinkronizáló”.)
 2. Ha a csatorna figyelése közben meghallja egy szomszédja ütemezését, akkor:
 - ugyanazt az értéket állítja be magának is. („követő” állomás)
 - **Véletlen t_d ideig vár**, majd elküldi az ütemezését a szomszédainak.
SYNC üzenet: „ $t-t_d$ másodperc múlva alszom”

Ütemezés választása és karbantartása

3. Ha egy állomás vesz egy sajátjától eltérő ütemezésű SYNC csomagot, akkor... ?
 1. ha az az egyetlen szomszédja, akkor átveszi azt és törli a sajátját.
 2. ha több szomszédja is van, akkor átveszi azt is, azaz felébreszti magát akkor is.



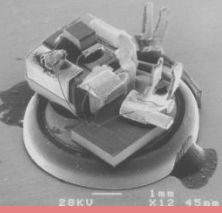
Fig. 2. Neighboring nodes A and B have different schedules. They synchronize with nodes C and D respectively.

- Hátrány: A virtuális klaszter határ csomópontjai (A és B) kevesebbet alszanak.

Ütemezés választása és karbantartása

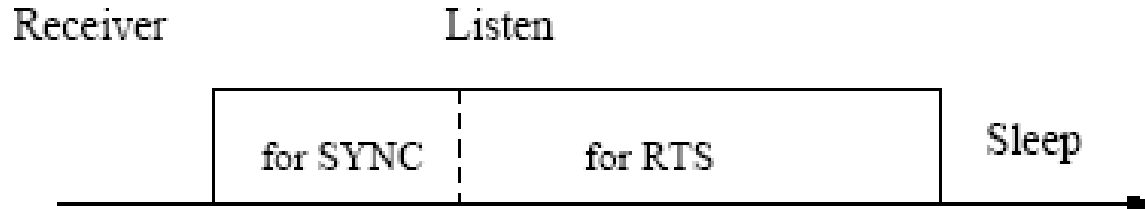
- Az órajelek elcsúszása miatt az állomásoknak időről időre frissíteniük kell az időzítési információikat. (pl. percenként)
 - SYNC: node ID + hátralévő idő a következő alvásig

- Egy állomás akkor is küld időnként SYNC csomagot, ha nincsenek követői. (?)
 - Ez lehetővé teszi az esetleg újonnan érkező állomások szinkronizálódását is.



Adatküldés és szinkronizáció

- Hogy egy állomás SYNC csomagok mellett adatcsomagokat is fogadni tudjon, az aktív intervalluma két részre van osztva:



- Minden csomag adása előtt az állomás egy véletlen ideig figyeli a csatornát, hogy az szabad-e. (vivőérzékelés – carrier sense)

Adatküldés és szinkronizáció

- Egy állomás a vevő aktív idejében küldhet egy SYNC és/vagy egy adatcsomagot:

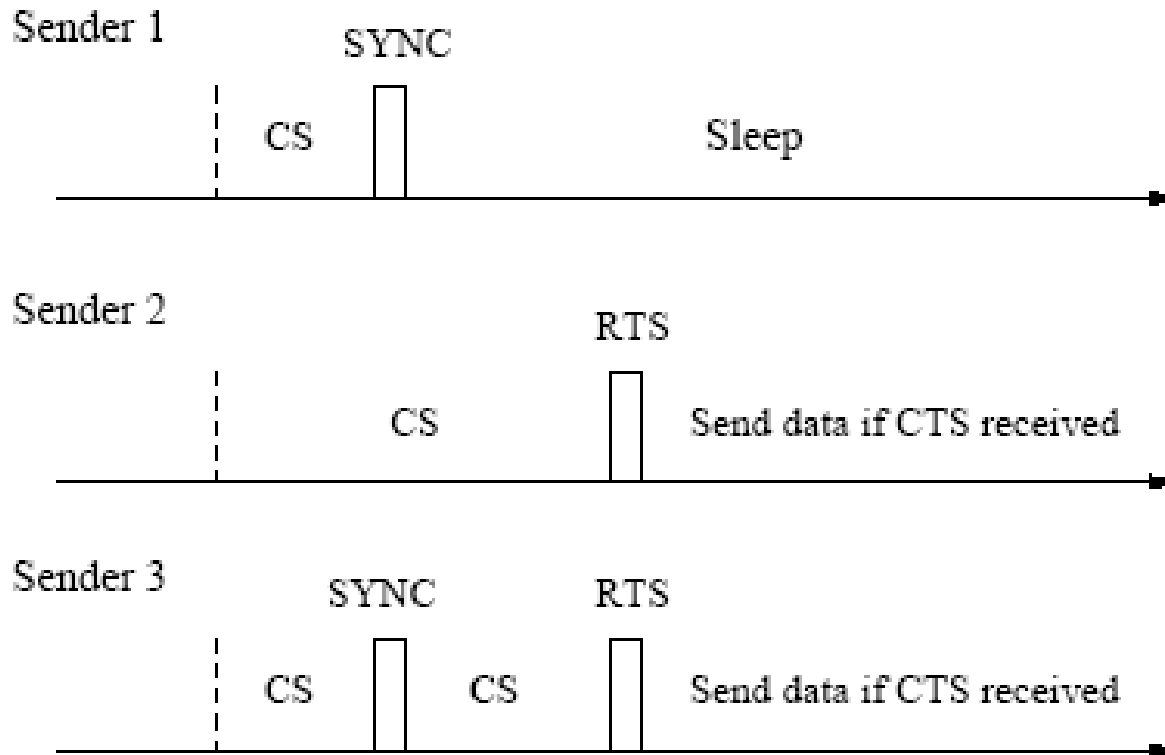


Fig. 3. Timing relationship between a receiver and different senders. CS stands for carrier sense.

Ütközés elkerülés

- Az ütközések elkerülése a MAC protokoll egyik alapvető feladata.
- S-MAC eljárásai az ütközés elkerülésére:
 - Az IEEE 802.11 vivőérzékeléses eljárását alkalmazza a fizikai rétegben.
 - Az RTS-CTS mechanizmust alkalmazza a rejtett terminál problémájának elkerülésére.
 - Alkalmaz egy **virtuális vivőérzékeléses** eljárást is.
- Virtuális vivőérzékelés:
 - Ha egy állomás veszi egy nem neki címzett csomag fejrészét, akkor abból ki tudja olvasni a csomag hosszát.
 - Az adott időtartamig a csatornát foglaltnak tekinti anélkül, hogy közben megvizsgálná azt.
- Üzenetszórás (SYNC) esetén nincs RTS-CTS (?)
- Adatkommunikáció: RTS/CTS/DATA/ACK



Áthallás elkerülése

- Az IEEE 802.11 szerint minden állomás figyeli a szomszédai adásait, hogy hatékony virtuális vivőérzékelést végezhesen.
 - A sok „felesleges” csomag vétele energiapazarló, különösen nagy node-sűrűség és nagy forgalom esetében.
- S-MAC az áthallások csökkentésére egy állomás aludni megy, ha hall egy RTS vagy CTS csomagot.
 - Így nem kell feleslegesen vennie egy (esetlegesen) hosszú adatcsomagot és az ACK csomagot.
- Pl. „A” ad „B”-nek. Kérdés: *Kinek kell aludni?*

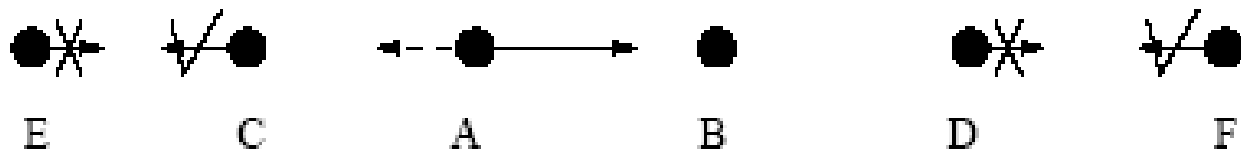


Fig. 4. Who should sleep when node A is transmitting to B?

- Válasz: Mind az adó, mind a vevő közvetlen szomszédjának. (Interferencia a vevőnél lép fel!)

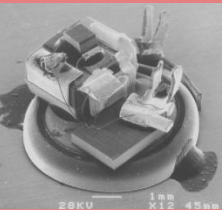
Üzenet továbbadás

- Kérdés: Hogyan lehet egy hosszú adatcsomagot továbbítani energiatakarékosan és lehetőleg kis késleltetéssel? (?)
- Két lehetséges megoldás:
 1. Egyben továbbítjuk.
Hátrány: Ha néhány bit meghibásodik, az egész csomagot újra kell küldeni.
 2. Szegmensekre bontva továbbítjuk.
Hátrány: Nagy „overhead” a kontrollcsomagok miatt. (Minden szegmenshez szükség van egy RTS/CTS/ACK csomagra.)
- Ötlet: S-MAC esetén az üzenetet kis szegmensekre bontjuk, majd a szegmenseket börstben küldjük el.
 - Az összes szegmenshez egyetlen RTS-CTS pár tartozik.
 - Az RTS csomag az egész börst küldéséhez szükséges időt jelzi.



S-MAC: üzenet továbbítás

- Minden szegmens továbbítása után az adó vár egy ACK csomagot a vevőtől. Ha nem kapja meg, újraküldi a szegmenst.
- Minden szegmens és ACK tartalmazza a teljes borszt továbbításához szükséges időt.
- A gyakori nyugták kiküszöbölik a rejtett terminál problémáját.
 - Ha egy állomás egy borszt továbbítása közben ébred fel, és nincs az adó rádiósugarán belül, akkor üresnek találja a csatornát és adásával interferenciát okoz.



S-MAC és 802.11 szegmentálása

- Megjegyzés: A 802.11 szintén támogatja a szegmentálást.
 - Az RTS és CTS csomagok csak az első szegmens idejére foglalja le a csatornát. A további foglalást a következő csomagok és nyugták tartalmazzák.
 - Ez nem teszi lehetővé az „alvást”.
- A 802.11 szegmentálása a fairness biztosítására hivatott.
 - Ha egy állomás nem kap nyugtát, abba kell hagynia az adást és újra versenyeznie kell a csatornáért.
- S-MAC esetében a fairness másodlagos kérdés.
- Határt kell azonban szabni az újraküldéseknek és a foglalási ablak állandó kitolásának.
 - Pl. A fogadó állomás működésképtelenné vált.

