

# TDMA ütemezés megvalósítása hibatűrő egyirányú kör-topológiájú hálózatokban

Vakulya Gergely                      Simon Gyula

Pannon Egyetem  
Rendszer- és Számítástudományi Tanszék  
Veszprém

e-mail: {vakulya,simon}@dcs.uni-pannon.hu

## Kivonat

Egyirányú kör-topológiájú hálózatok pl. alacsony energiafogyasztású felügyeleti rendszerekben használatosak, ahol az (általában státusz- és riasztási) információnak a hálózat minden eleméhez el kell jutnia, még egyes eszközök meghibásodása esetén is. A javasolt rendszer alacsony fogyasztás mellett garantálja az üzenetek határidőre történő kézbesítését, még egyszeres csomóponti- és összeköttetési hibák jelenlétében is.

## 1. Bevezetés

Az adattovábbítás egy meghatározó szolgáltatás vezeték nélküli szenzorhálózatokban. Az egyszerűbb forgalomirányítási algoritmusok jellemzően elárasztást használnak, míg az összetettebbek például feszítőfán, vagy gradiens módszeren alapulnak. Ezek az algoritmusok az eszközök rádióit folyamatosan bekapcsolt állapotban tartják, CSMA közeghozzáférést alkalmazva. Az algoritmusok energiahatékonysága low-power listening technikával javítható [1].

Egy másik megközelítés a hierarchikus hálózati struktúra, koordinátor, router és végesekő típusú csomópontokkal. A végesekőök számos megvalósításban alhatnak, de a koordinátor és router csomópontoknak folyamatosan ébren kell lenniük.

Abban az esetben, ha a valós idejű adattovábbítás és az energiahatékonyság egyszerre lép fel mint követelmény, az időosztásos multiplexelés (Time Division Multiple Access) a kézenfekvő megoldás. A TDMA egy széleskörűen alkalmazott közeghozzáférési (MAC) protokoll, amit szenzorhálózatokban is elterjedten használnak. A TDMA előre ütemezett időréseket használ a kommunikációhoz, így ütközésmentes és -- mivel a rádiót már nem kell folyamatosan bekapcsolva tartani -- egyúttal energiahatékony kommunikációt is biztosít. Ezen kívül a TDMA adattovábbítási ideje is determinisztikus. A TDMA hátránya, hogy kihasználatlan időrések is előfordulhatnak, így nem használható ki a teljes rendelkezésre álló sáv szélesség, illetve hogy az egész hálózatra kiterjedő időszinkronizáció szükséges a módszer működéséhez [2, 3]. A TDMA-alapú megoldások népszerűek a szenzorhálózatos alkalmazásokban, ahol a korlátos készletelés és az energiahatékonyság kulcsfontosságú paraméterek [4, 5].

Ebben a cikkben egy TDMA alapú forgalomirányítási algoritmus kerül bemutatásra, ahol minden hálózati csomópont az ideje nagy részét alvással töltheti, így biztosítva a magas energiahatékonyságot. A közölt megoldás garantált adattovábbítási időket nyújt. Emellett az algoritmus egy, vagy tetszőleges számú, nem egymást követő csomópont meghibásodása esetén is biztosítja a rendszerszolgáltatásokat.

## 2. A javasolt rendszer

A rendszerrel szemben támasztott fő követelmények a következők: A hibatűrés biztosításához szükséges, hogy elkerüljük a speciális, központi csomópontokat, ezért a teljes rendszernek decentralizálnak kell lennie. Emellett a rendszernek legalább egy eszköz kiesését el kell viselnie és az általa nyújtott szolgáltatásokat ebben az esetben is biztosítania kell. A rendszerben kialakuló késleltetésnek determinisztikusnak kell lennie: az üzeneteknek egy meghatározott, korlátos időn belül célba kell érniük. A rendszernek több száz csomópontig skálázhatónak kell lennie. A rendszer üzemidejének a lehető legtöbbnek kell lennie elemes táplálás esetén.

### 2.1. A felderítési mód

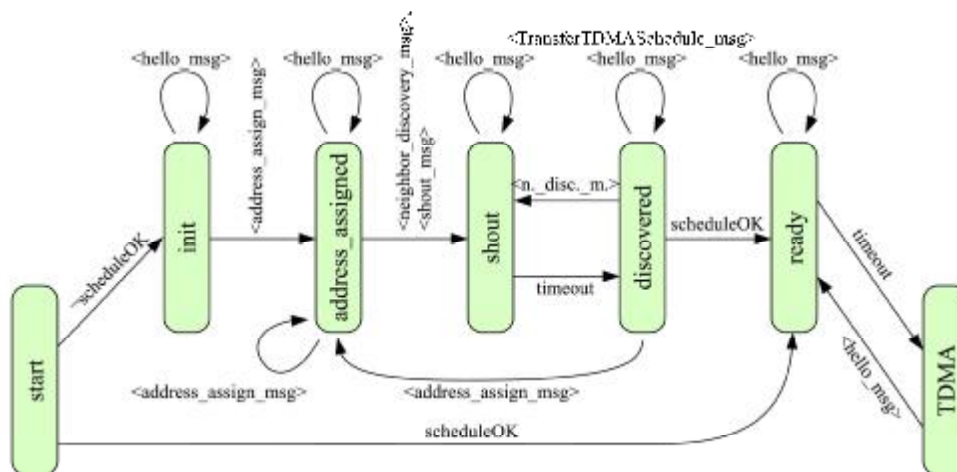
Induláskor minden szenzor csomópont megvizsgálja a nemfelejtő memóriáját. Ha itt egy valós TDMA ütemezést talál, akkor rövid várakozási idő után átlép TDMA módba. Egyéb esetben megkezdődik a hálózat-felderítési fázis, majd a TDMA ütemezés letöltődik a csomópontokba. Először a bázisállomás egy *hello\_msg* üzenetet küld a csomópontoknak, amire azok a saját, egyedi 64 bites hardver-címükkel válaszolnak. Ezután a bázisállomás a 64-bites címeket 16-bit-esekre cseréli *address\_assign\_msg* üzenetekkel. A csomópontok erre *address\_assign\_reply\_msg* üzenetekkel válaszolnak, ami nyugtaként szolgál, így a bázisállomás szükség esetén meg tudja ismételni az esetlegesen elveszett üzeneteket. Miután minden csomópont megkapta a rövid címét, a csomópontok feltérképezik egymás szomszédságát a *neighbor\_discovery\_msg* üzenet hatására. A csomópontok broadcast üzeneteket küldenek és összegyűjtik a szomszédaitól vett üzeneteket. Az érzékelt szomszédok listáját *neighbor\_reply\_msg* típusú üzenetek formájában küldik vissza a csomópontok a bázisállomásnak. Az üzenetek tartalmazzák, hogy melyik eszközt hányszor és milyen minőségben hallotta az adott csomópont. A szomszédlistákból a bázisállomás felépíti a hálózat kapcsolati gráfját és ez alapján generálja a TDMA ütemezést. Az ütemezés *TDMA\_schedule\_msg* típusú csomagok formájában töltődik le az eszközökre. A szenzorok minden vett csomagra egy-egy *TDMA\_schedule\_reply\_msg* üzenetet küldenek vissza a bázisállomásra nyugtaként, így az elveszett csomagok megismételhetők.

A rendszer viselkedését részletesen az 1. ábrán látható állapotgép írja le. Induláskor a rendszer a *start* állapotban van. Ekkor az eszköz bekapcsolja a rádióját és ellenőrzi a nemfelejtő memóriában tárolt konfigurációs adatokat – a mote ID-t és a TDMA ütemezést. Ha a konfiguráció megfelelő, a rendszer a *ready* állapotba megy át, egyébként pedig az *init* állapotba.

A *TDMA* állapotot kivéve minden állapotban a mote a *hello\_msg* üzenetre véletlenszerű késleltetés után az egyedi 64-bites címével válaszol, az állapota pedig változatlan marad. A *hello\_msg* a *TDMA* időzítőt is újraindítja, amennyiben az már futott.

Az *init* állapotban az eszközök a elfogadják az *address\_assign\_msg* üzenetet, amit a rövid (16-bites) cím beállítására használunk; minden további kommunikációhoz ezt a címet használjuk. Az üzenet az eszközt az *address\_assigned* állapotba viszi át.

Az eszközök *address\_assigned* állapotban négyféle üzenetet fogadnak. A *hello\_msg* működését már tárgyaltuk. Az előzőleg hozzárendelt rövid cím egy újabb *address\_assign\_msg*-vel felülírható, ebben az esetben az eszköz az aktuális állapotában marad.



1. ábra. A rendszer véges automata modellje

A *neighbor\_discovery\_msg* üzenet hatására az eszközök elkezdik felderíteni szomszédaikat és *shout* állapotba mennek át. A folyamat során minden eszköz megadott számú *shout\_msg* típusú broadcast üzenetet küld és feljegyzi az érzékelt szomszédok címét és a kapcsolat minőségét. Ha a *neighbor\_discovery\_msg* üzenet elveszik, egy vett *shout\_msg* üzenet is elindítja a felderítési folyamatot, *shout* állapotban viszont a *neighbor\_discovery\_msg* és a *shout\_msg* már nem indít újabb felderítést. A felderítés után az eszközök elküldik a bázisállomásnak az összegyűjtött szomszédsági információt és *discovered* állapotba kerülnek.

A *discovered* állapotban az *address\_assign\_msg* üzenettel egy új rövid címet rendelünk hozzá az eszközhöz, *address\_assigned* állapotba léptetve azt. A TDMA ütemezés *TDMA\_schedule\_msg* üzenetekkel tölthető le a mote-okra. Az ütemezés egyes bejegyzései egyenként beíródnak az eszköz nemfelejtő memóriájába. Ha az összes bejegyzés sikeresen letöltődött, a mote egy bizonyos idő elteltével *ready* állapotba megy át.

Mind az *address\_assign\_msg*, mind pedig a *TDMA\_schedule\_msg* típusú üzenetek nyugtázva továbbítódnak, így az elveszett üzenetek újraküldhetők.

A rendszer a *ready* állapotból egy bizonyos idő elteltével a *TDMA* állapotba megy át, hacsak nem tartjuk *hello* üzenetekkel a felderítési állapotok egyikében.

## 2.2. A TDMA mód

A közölt kommunikációs módszerhez léteznie kell egy Hamilton-körnek a kapcsolati gráfon, ezen kívül a hibatűrési biztosításához arra is szükség van, hogy minden csúcs a Hamilton-körön kettővel előtte és utána levő csomóponttal is tudjon kommunikálni.

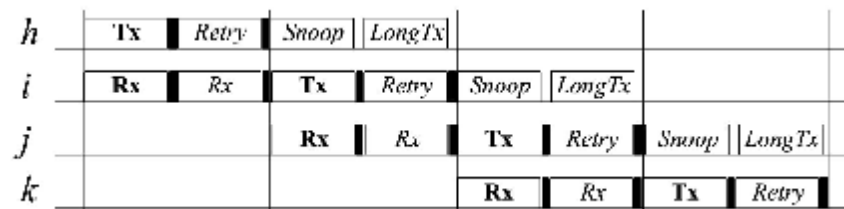
A TDMA alapú kommunikáció azonos periódusokból álló körkörös réselt kommunikációt használ. Az ütemezés a hálózatban szereplő összes csomópont eseményeinek időbeosztását megadja.

Az ütemezés vétel-adás-lehallgatás-áthidalás négyesekből áll, a következő módon. Jelöljön  $h$ ,  $i$ ,  $j$  és  $k$  négy egymást követő csomópontot a Hamilton-körön. A kommunikáció a következőképp alakul:

- Az  $i$  mote fogad egy csomagot a  $h$  mote-tól.
- Az  $i$  mote küld egy csomagot a  $j$  mote-nak.
- Közvetlenül a vétel után a  $j$  mote küld egy nyugtát az  $i$  mote-nak.

- Ha az  $i$  mote megkapja a nyugtát, tudomásul veszi, hogy az üzenet megérkezett a  $j$  mote-hoz és a rádióját kikapcsolja. A nyugta hiányában megpróbálja még egyszer elküldeni az üzenetet, amennyiben az újraküldés engedélyezett. Ha az  $i$  mote semelyik nyugtát sem kapja meg, feltételezhető, hogy a  $j$  mote meghibásodott. Ebben az esetben az  $i$  mote lehallgatja  $j$  következő adását.
  - Az újraküldés csak akkor engedélyezett, ha az előző vétel sikeres volt, tehát ha a mote friss adatokkal rendelkezik. Ellenkező esetben az újraküldési időrés áthidalásra használható.
  - Minden csomagban van egy jelzőbit (*last\_OK*, vagy LOK), ami azt jelzi, hogy csomagot küldő mote utolsó vétele sikeres volt-e.
- A következő adási időrésben a  $j$  mote küld egy csomagot a  $k$  mote-nak.
- Ugyanabban az időrésben, amikor a  $k$  mote veszi a  $j$  mote üzenetét, az  $i$  mote is lehallgathatja azt, amennyiben nem vette egyik üzenetének nyugtáját sem. A következő három eset lehetséges:
  - Az  $i$  mote egy olyan csomagot hallgat le, ahol az LOK jelzőbit be van állítva. Ez azt jelenti, hogy a  $j$  csomópont sikeresen vette az  $i$  csomópont adását, így az  $i$  mote-nak nincs további teendője: kikapcsolhatja rádióját.
  - Az  $i$  mote egy olyan csomagot hallgat le, ahol az LOK jelzőbit hamis értékű. Ezt azt jelenti, hogy a  $j$  csomópont működik, de nem vette az  $i$  üzenetét. Ekkor  $i$  a  $j$  egyébként kihasználatlan újradási időrésében küldhet egy csomagot a  $k$  csomópontnak.
- Az  $i$  mote egyáltalán nem veszi  $j$  adását. Ez azt jelenti, hogy a  $j$  csomópont valószínűleg nem működik. Ekkor  $i$   $j$  üres újrapróbálkozási időszeletében küldhet egy csomagot  $k$ -nak, átugorva  $j$ -t.

Az utóbbi két esetet *áthidaló adásnak* nevezzük.



2. ábra. A TDMA kommunikáció idődiagrammja. A Tx a normál adást, az Rx a normál vételt, a Retry pedig az adás ismétlését jelöli. A vastagon szedett események a normál működésre vonatkoznak, a dőlt betűsek pedig az opcionális kommunikációt jelölik, amik hibajavításkor használhatósak.

A folyamat a 2. ábrán is látható. Az adást piros, a vételt zöld színű téglalap jelöli. A sötét téglalapok a nyugtázó üzenetek. A dőlt betűs téglalapok azokat a kommunikációs tevékenységeket jelölik, amik csak hibajavításkor lépnek működésbe (újraküldés és áthidalás). Megjegyzendő, hogy a mote-ok csak a saját adási és vételi időrésükben vannak ébren, a többi megjelölt időben csak egy eszköz, vagy kommunikációs link meghibásodása esetén.

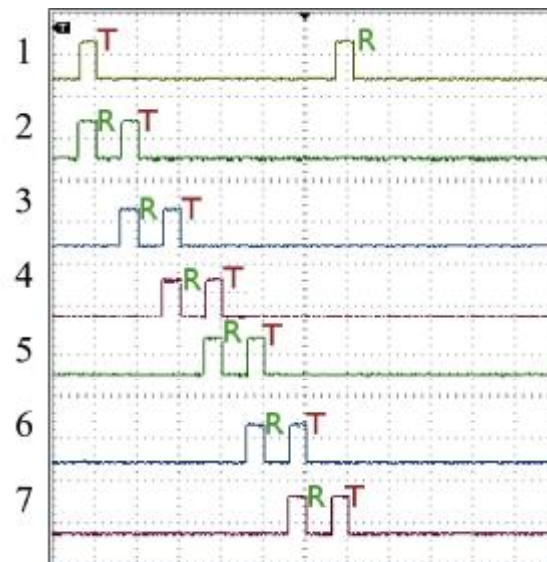
### 2.3. Időszinkronizáció

A TDMA alkalmazásához pontos időszinkronizáció szükséges. Kommunikáció közben minden egyes adás-vétel párnál szinkronizáció is történik a szomszédok között. A legtöbb alkalmazásnál a kommunikáció gyakorisága elegendő ahhoz, hogy az órák driftjét ne kelljen kompenzálni. A mote-ok lokális órája a 62,5kHz-es 32-bites hardverórát használja. A globális

időt minden mote a hardveróra és egy saját ofset összegeként számítja ki. A küldött és fogadott üzeneteket az FTSP-nél [6] is alkalmazott módszer szerint időbélyegezik a mote-ok, így a vevő mote meg tudja mérni a saját és az adó globális ideje közti különbséget, ami alapján saját óráját kompenzálhatja.

### 3. Eredmények

A hálózat működését egy 7 csomópontból álló hálózattal teszteltük. A hálózat elemei TinyOS [7] kompatibilis szenzor eszközök voltak, amik az 1-7 azonosítókat kapták. A rádiók ki- és bekapcsolt állapotát egy négycsatornás tároló oszcilloszkóppal mértük. A 3. ábra két szinkronizált felvételtől készült és a hálózat normál (hibamentes) működését mutatja. A maximális globális szinkronizációs hiba  $150\mu\text{s}$  volt, a szomszédok közötti hiba pedig legfeljebb  $16\mu\text{s}$ , ami a felhasznált óra egy időegységének felel meg. A hibatűrést csomópontok kikapcsolásával teszteltük. A 4. ábrán egy olyan eset látható, ahol a 3-as számú eszközt kikapcsoltuk. Két nem nyugtázott üzenet után a 2-es csomópont lehallgatja a 3-as csomópont adását és mivel nem vesz egyetlen üzenetet sem, ezért a 4-es csomópont felé egy áthidaló üzenetet küld a 3-as csomópont újrapróbálkozási időéréseben.



3. ábra. Az eszközök rádiójának ki- és bekapcsolt állapota hibamentes esetben.



4. ábra A rádiók aktivitása a 3-as számú eszköz kikapcsolásakor. A 2-es csomópont lehallgatja a 3-as adását és egy áthidaló adást tesz a 4-es csomópont felé a 3-as újrapróbálkozási időéréseben.

Az energiahatékonyságot szintén megmértük a tesztek során. A rádió kitöltési tényezője minden esetben 0,8%, processzor kitöltési tényezője pedig 0,2% alatti volt. Utóbbi természetesen erősen alkalmazásfüggő.

## 4. Összefoglalás

A dolgozatban egy TDMA alapú hibatűrő, energiahatékony és garantált válaszütemet biztosító forgalomirányítási módszert mutattunk be. A módszer működése és annak szoftver architektúrája részletesen bemutatásra került. A rendszer egy vezeték nélküli szenzorhálózatos tesztkörnyezetben implementálásra is került. A rendszer működési paramétereit részletesen megvizsgáltuk. A bemutatott rendszerben a rádió 0,8%-os kitöltési tényezővel működött, valamint a rendszer egy, illetve tetszőleges számú nem egymást követő eszköz kikapcsolása esetén is képes volt biztosítani az általa nyújtott szolgáltatásokat.

## Köszönetnyilvánítás

A kutatást részben a Magyar Kormány TÁMOP-4.2.2/B-10/1-2010-0025 projektje finanszírozta.

## Hivatkozások

[1] J. Polastre, J. Hill, and D. Culler, „Versatile low power media access for wireless sensor networks,” in Proceedings of the 2nd international conference on Embedded networked sensor systems. New York, NY, USA: ACM, 2004, pp. 95–107.

[2] S. C. Ergen and P. Varaiya, „TDMA scheduling algorithms for wireless sensor networks,” *Wireless Networks*, vol. 16, no. 4, pp. 985–997, 2010.

[3] K. Kredó, II and P. Mohapatra, „Medium access control in wireless sensor networks,” *Computer Networks*, vol. 51, pp. 961–994, 2007.

[4] L. Shi and A. O. Fapojuwo, „TDMA scheduling with optimized energy efficiency and minimum delay in clustered wireless sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 9, no. 7, pp. 927–940, 2010.

[5] N. S. M and A. S. Babu, „TDMA based energy conservation in wireless sensor networks,” *IJCA Proceedings on International Conference on VLSI, Communications and Instrumentation (ICVCI)*, no. 9, pp. 1–6, 2011, published by Foundation of Computer Science.

[6] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi, „The flooding time synchronization protocol,” in Proceedings of the 2nd international conference on Embedded networked sensor systems. New York, NY, USA: ACM, 2004, pp. 39–49.

[7] P. Levis and D. Gay, *TinyOS Programming*. New York, NY, USA: Cambridge University Press, 2009.