

Útvonalválasztó protokollok vezeték nélküli szenzorhálózatokban

ÁCS GERGELY, BUTTYÁN LEVENTE

Budapesti Műszaki és Gazdaságtudományi Egyetem, Híradástechnikai Tanszék, CrySyS laboratórium
{acs, buttyan}@crysys.hu

Lektorált

Kulcsszavak: szenzorhálózatok, útvonalválasztás, protokollok, hálózati és működési modellek

A szenzorhálózatok változatos alkalmazásai különböző követelményeket támasztanak az útvonalválasztó protokollokkal szemben. A különböző követelményeknek köszönhetően igen sok javasolt protokoll található az irodalomban. Ebben a cikkben rendszerezük ezeket a útvonalválasztó protokollokat, és minden családból bemutatunk egy prominens képviselőt. A cikk újdonsága a rendszerezéshez használt szempontrendszer, mely a protokollok eddigieknél részletesebb taxonómiáját eredményezi.

1. Bevezetés

A szenzorhálózatokat egyrészt a mérést végző szenzorcsomópontok, másrészt az adatokat begyűjtő bázisállomások (báziscsomópontok) alkotják.

Míg a rendkívül nagy számú, és általában homogén szenzorcsomópontok tipikusan alacsony energiaellátottsággal és számítási kapacitással rendelkeznek, addig a kis számú bázisállomások erőforrása legtöbbször korlátlan. Az egyes szenzorcsomópontok egymással és a bázisállomásokkal egyaránt vezeték nélküli kapcsolatokon kommunikálnak, amelyeknek az energiaigénye jóval magasabb mint a csomópontok által végzett egyszerű számításoké, valamint egy üzenet elküldéséhez körülbelül kétszer annyi energia szükséges mind annak vételéhez.

Ezekből következik, hogy egy szenzorcsomópont által küldött üzenet útvonala egy bázis felé korántsem mellékes: a túl hosszú, illetve alacsony teleptöltöttségű csomópontokat tartalmazó útvonalak csökkentik a hálózat élettartamát, valamint a túl sok csomópontot tartalmazó útvonalak adott esetben növelhetik az üzenet késleltetését.

Sajnos ezen követelmények sokszor ellentmondanak egymásnak: ha mindig a legrövidebb útvonalon továbbít egy szenzorcsomópont a bázis felé, akkor az a köztes csomópontok rövid élettartamához, és bizonyos értelemben a hálózat élettartamának rövidüléséhez vezet, holott ez jelentené a legkisebb energiafogyasztást és késleltetést globális értelemben.

Összességében az útvonalválasztás célját maga az alkalmazás határozza meg; például valós idejű alkalmazásoknál a minimális késleltetés, míg statisztikai számításokat végző alkalmazásoknál a hosszú élettartam lehet elsődleges cél. A sokféle alkalmazásra így különböző útvonalválasztó protokollokat javasoltak [1], amelyek az útvonalválasztás céljában, valamint ezen célok elérésre használt technikákban különböznek, ahol a technikákat a hálózat által támasztott technológiai korlátok alakítják.

2. Osztályozási szempontrendszer

Az útvonalválasztó protokollok nagy száma a következő természetes kérdést vetheti fel egy alkalmazásfejlesztő számára: Melyik protokoll felel meg legjobban az alkalmazásom számára? E kérdés megválaszolásához az szükséges, hogy az összes útvonalválasztó protokollt egy olyan közös szempontrendszerbe helyezzük, ahol a protokollok összehasonlíthatóvá válnak.

Rendszerezésünk alapját egyrészt a hálózat technológiáját és a protokollok működését leíró rendszermodell, másrészt a protokoll által kitűzött cél alkotja. A rendszermodell tovább osztható a hálózati technológiát leíró hálózati modellre, valamint a működési tulajdonságokat leíró működési modellre.

2.1. A hálózati modell

2.1.1. Bázisállomás

A bázisállomások nagy számítási kapacitású és korlátlan energiaellátottsággal rendelkező hálózati eszközök. A következő tulajdonságok viszont már hálózatonként eltérőek lehetnek, és így az útvonalválasztó protokoll működését is befolyásolják.

- **Szám:** A bázisállomások száma lehet egy, illetve egynél több. A legtöbb gyakorlati esetben a tipikus számuk egy, de több bázisállomás növeli az adatbegyűjtés robusztusságát, valamint csökkenti annak késleltetési idejét. Az egy bázisállomást tartalmazó esetben (ha nincs egyéb igény a szenzorcsomópontok közötti explicit kommunikációra) a végcél minden üzenet számára ugyanaz, míg egynél több számú bázisállomás esetében, a cél lehet több, nem feltétlen minden üzenetre egyező bázisállomás.

- **Mobilitás:** Egy bázisállomás lehet fix (helyhezkött), korlátozott mértékben mobilis, illetve korlátlanul mobilis. Legtöbbször olyan hálózatokban mobilisak a bázisállomások, ahol a számuk kevés, mégis nagyobb robusz-

tusságra és kisebb késleltetésre van szükség az ugyanolyan számú de helyhez kötött bázisállomások esetéhez képest. A mobilitás miatt az útvonalválasztás által kialakított topológia időben nagy mértékben változhat, ami túlterheltséget jelenthet a hálózati rétegben.

- **Jelenlét:** Néhány alkalmazásnál a bázisállomások nincsenek folyamatosan jelen (például kikapcsolják őket karbantartási célból), míg más esetekben a jelenlét folyamatosan biztosított. Az időszakos leállásokat az útvonalválasztó protokollnak támogatnia kell, mivel a bázis jelenlétének hiánya nem feltétlen ebben az esetben nem hiba, és így az üzenetet nem kell eldobni vagy átírányítani, hanem azokat esetleg várakoztatni kell.

- **Lefedtettség:** A bázisállomás nagy erőforráskészlete lehetővé teszi egy erős antenna alkalmazását is, amivel esetleg az egész hálózat lefedhető, és így minden csomópontot közvetlenül elérhet a bázisállomás (ha nincs árnyékoló akadály). Ekkor nincsen szükség a bázis és a szenzorcsomópontok közötti útvonalválasztásra. Megjegyezzük azonban, hogy a legtöbb gyakorlati esetben a lefedettség nem teljes.

2.1.2. Szenzorcsomópont

Szenzor alatt a továbbiakban a vezérlő áramkörök és rádiós egység, az érzékelő egység, és a szükséges telep együttesét értjük. A szenzorok kis számítási kapacitású és korlátozott energiaellátottsággal rendelkező, legtöbbször helyhez kötött hálózati eszközök. A következő tulajdonságok viszont már hálózatonként eltérőek lehetnek, és így az útvonalválasztó protokoll működését is befolyásolják.

- **Elhelyezés:** A szenzorok elhelyezése lehet véletlenszerű, vagy determinisztikus. Sok esetben az útvonalválasztó protokollnak alkalmazkodnia kell a kötött hálózati szerkezethez (például út mentén szabályosan elhelyezett szenzorok), más esetben (helikopterről kiszórt szenzorok) pedig alapozhat arra a feltevésre, hogy a szenzorok egyenletesen véletlenül vannak elosztva.

- **Átviteli energia:** Az átviteli energia szintje lehet állítható, vagy pedig konstans, vagyis a szenzor minden üzenetet ugyanazon az energiaszinten sugároz. Ez elsősorban a szomszédos csomópontok energiában mért távolságának a meghatározásában játszik szerepet, hiszen sok protokoll ezt használja a szomszédokhoz rendelt költségek megállapításához.

- **Lefedtettség:** Szükséges esetben, ha egy szenzor nagy energiával ad, bármely más csomópontot közvetlenül elérhet a hálózatban. Ennek kihasználása azonban általában csak kis méretű hálózatokban lehet hatékony a nagy energiaigény és a megnövekedett interferencia miatt.

- **Címzés:** Az útvonalválasztás feladata a szenzorhálózatokban a bázistól (vagy ritkán más szenzortól) ér-

kező kérések eljuttatása ahhoz a csomópont(ok)hoz, amely rendelkezik a kért adattal, valamint ezen adat visszajuttatása a bázishoz. Ennek megfelelően beszélhetünk a kérés, illetve a válasz kezelése során alkalmazott címzésről:

- **Kérés:** A kérés címzése történhet a kért adat alapján (Mi az átlaghőmérséklet?), vagy a kért adat vagy szenzor pozíciója alapján (Mi a hőmérséklet az (x,y) helyen?).
- **Válasz:** A válasz címzése történhet a bázis vagy szenzor pozíciója alapján (Válasz az (x,y) helyen levő bázishoz), vagy a választ váró csomópont lokális, vagy globális azonosítója alapján. Utóbbi esetben, a kérés vétele során minden szenzor megjegyzi a kérést küldő szomszéd azonosítóját, és ennek a szomszédnak továbbítja a választ visszafelé.

- **A MAC réteg által nyújtott szolgáltatás:** A MAC réteg gondoskodhat a szomszédos csomópontok felderítéséről (ahol a szomszéd definíciója változó), illetve ezen felül, támogathatja a szomszédok költségeinek megállapítását (ahol a költség definíciója szintén protokollonként eltérő). Néhány útvonalválasztó protokoll integrálva van a MAC réteggel a gyorsabb és energiatakarékos működés végett (cross layer design). Viszont a legtöbb esetben a MAC réteg nem felel ezen feladatokért, és ezért magának az útvonalválasztó protokollnak a feladata a szomszédok és azok költségeinek megállapítása.

2.2. A működési modell

Az útvonalválasztó protokollok a következő ortogonális működési tulajdonságokkal jellemezhetőek:

- **Kommunikációs minta:** Az útvonalválasztó protokoll támogathatja a szenzorok közötti, a bázis és szenzorok közötti, valamint szenzorok és bázis közötti kommunikációt.

- **Szenzor–szenzor kommunikáció:**

Ezt a kommunikációs mintát elsősorban az ad-hoc hálózatok számára javasolt protokollok támogatják, szenzorhálózatokban ilyen kommunikációra ritkán van szükség.

- **Bázis–szenzor kommunikáció:**

A bázis felől érkező kérések irányítása során szükséges ennek a kommunikációs mintának a támogatása, mely azt a képességet jelenti, hogy a bázis közvetlenül vagy közvetve bármely szenzornak tud üzenetet küldeni.

- **Szenzor–bázis kommunikáció:**

A válaszok irányítása során szükséges ennek a mintának a támogatása, mely azt a képességet jelenti, hogy minden szenzor képes közvetlenül vagy közvetve bármely bázisnak üzenetet küldeni. Minden egyes minta esetén a kommunikáció típusa lehet unicast (egy-egy), multicast (egy-több), reverse-multicast (több-egy), illetve anycast (egy-

bármely). Anycast kommunikációra például akkor lehet szükség amikor a bázis adatot kérdez a hálózattól, és nem lényeges, hogy pontosan melyik szenzor válaszol a kérésre, hanem bárki válaszolhat, aki rendelkezik a kért adattal.

- *Hierarchia:* A hierarchikus protokollok esetében az egyes szenzorok (logikai) hierarchiaszinteken helyezkednek el. A szenzorok az alacsonyabb szinten levő szenzoroktól fogadnak üzeneteket, ezeket aggregálják saját adataikkal, és az aggregátumot továbbítják a magasabb hierarchiaszinten levő szenzoroknak. A hierarchia tetején a bázis található. A hierarchia kialakítása lehet statikus vagy dinamikus. Utóbbi esetben, a szenzorok dinamikusan választanak aggregátorokat, és ennek az aggregátor csomópontnak küldenek minden üzenetet. Az aggregátorok további aggregátorokat választanak, és így tovább. A hierarchia kialakításának célja a hálózat élettartamának növelése. A nem-hierarchikus protokollok esetén az egyes csomópontok bármely csomóponttól fogadnak üzenetet aggregálásra, így minden csomópont viselkedhet aggregátorként.

- *Kézbesítési módszer:* A legtöbb protokoll egyetlen útvonalat választ a bázis felé, s ezen minden üzenet egyetlen példányát továbbítja (egy/egy). Néhány protokoll azonban a robusztusság érdekében több útvonalat is választ, s vagy minden üzenetet ezen útvonalak egyikén továbbít (több/egy), vagy minden üzenetet minden útvonalon továbbít (több/több). Előbbi esetben, a továbbításra használt útvonal kiválasztása lehet determinisztikus vagy véletlenszerűen.

- *Számítás:* Az egyes szenzorok vagy maguk határozzák meg lokálisan a következő csomópontot a bázis felé (decentralizált), vagy pedig minden csomópont elküldi a szomszédossági listáját a bázisnak, amely globálisan meghatározza minden egyes szenzornak a következő csomópontot a bázis felé (centralizált). Az utóbbi optimális megoldást nyújt, viszont ehhez nagy mértékű hálózati kommunikáció szükséges, amely csak kevés számú csomóponttal rendelkező és fix topológiájú hálózatoknál használható hatékonyan. A szenzorok csak a szomszédokkal tartják a kapcsolatot, ahol a szomszéd definíciója változó lehet. Legegyszerűbb esetben, a szenzorok azt tekintik szomszédnak, akitől útvonalválasztó üzenetet kapnak. Más esetekben explicit HELLO üzenetek segítségével derítik fel a szomszédokat. Ekkor minden csomópont egy bizonyos energiaszinten küld egy HELLO üzenetet többesküldéssel (broadcast), és minden csomópont azt tekinti szomszédnak akinek HELLO üzenetét hallotta. Ez a mechanizmus lehet része az útvonalválasztó protokollnak, vagy pedig az alsóbb protokollrétegben (például MAC réteg) lehet megvalósítva.

- *Állapot:* Az egyes protokollok futtatáskor szükséges lehet minden csomópontnak valamilyen információt tárolni az aktuális állapotról (például a saját költség-

gét, ki a következő csomópont a bázis felé, annak mi a költsége stb.) Ezzel szemben néhány protokoll semmilyen, vagy csak elhanyagolható mennyiségű állapotinformációt tárol. Utóbbi esetben, minden csomópont az üzenetben elhelyezkedő információ, vagy minimális lokálisan tárolt állapotinformáció segítségével (például a szomszédos csomópontok pozíciói) képes meghatározni a következő csomópontot a bázis felé.

- *Következő szomszéd választása:* „Az összes protokoll közös tulajdonsága, hogy a csomópontok lokálisan, a saját információik alapján választják a következő csomópontot a továbbítandó csomag útján. A választás történhet: véletlenszerűen, az üzenet tartalma alapján, az üzenetben szereplő geometriai pozíció alapján, hierarchia szintek alapján, illetve néhány protokollban a csomópontok az üzeneteket többszórással továbbítják, és a szomszédok maguk döntenek a továbbításról.”

- *Riportolási modell:* A riportolási modell három féle lehet, ahol a csoportosítás alapját az ok adja, ami a forrásszenzorokat üzenetküldésre készíti:

- *Idővezérelt:* A szenzorok szabályos időközönként, vagy egy bizonyos időpontban válaszolnak. Az idővezérelt protokoll támogathat folytonos (periodikus) vagy időben egyszeri jelentést, komplex (összetett típusú) vagy egyszerű (atomi típusú) adatok jelentését, aggregálható vagy nem aggregálható adatok jelentését, illetve replikált (több szenzornál megtalálható) vagy egyedi (egyetlen szenzornál megtalálható) adat jelentését.

- *Kérésvezérelt:* A szenzorok a bázis kérésére válaszolnak. A kérésvezérelt protokoll támogathatja komplex vagy egyszerű adatok jelentését, aggregálható vagy nem aggregálható adatok jelentését, illetve replikált vagy egyedi adat jelentését.

- *Eseményvezérelt:* A szenzorok egy bizonyos esemény hatására küldenek üzenetet a bázis felé. Az eseményvezérelt protokoll támogathatja komplex vagy egyszerű adatok jelentését, aggregálható vagy nem aggregálható adatok jelentését, illetve replikált vagy egyedi adat jelentését.

Néhány protokoll több csoportba is tartozhat, ha többféle riportolási modellt is támogat.

2.3. A protokoll célja

Minden útvonalválasztó protokoll alapvető célja az üzenet eljuttatása a forrástól a célig. Ez történhet valós idejű követelményekkel együtt, amikor az üzenetnek egy bizonyos időn belül el kell érnie a célt, vagy pedig valós idejű követelmények nélkül. Utóbbi esetben a sikeresség egy mérőszáma lehet a sikeresen kézbesített üzenetek száma, míg az előbbi esetben a sikerességet az adott időn belül sikeresen kézbesített üzenetek száma méri.

A protokolloknak az előbbivel párhuzamos célja lehet a hálózat élettartamának maximalizálása, aminek a

mérőszáma már nem egyértelmű. Ha minden csomópont egyenrangú, akkor a hálózat élettartama lehet az első csomópont működésképtelenségéig eltelt idő, vagy akár az utolsó csomópont működésképtelenségéig eltelt idő. Ritka esetekben, amikor csomópontok nem egyenrangúak, akkor a magasabb prioritású csomópontokra érvényesek ezek a megállapítások. Az alkalmas metrika megválasztása alkalmazásfüggő.

3. Protokollok

Amint látható, a protokollok csoportosítása többféleképpen lehetséges. A táblázatokban felsoroltuk a létező jelentősebb útvonalválasztó protokollokat, ahol az 1. táblázat tartalmazza az egyes protokollok rendszerezését a hálózati modell és a kitűzött cél tekintetében, a 2. táblázat pedig ugyanezen protokollok működési modell szerinti osztályozását.

A modellek szerint megegyező protokollokat egyként kezeltük, összefoglaló nevük végére csillagot tettünk. Az egyes cellák tartalma a fentebb leírtak szerint értelmezhető, az egyetlen csillagot tartalmazó cella jelentése, hogy a protokoll az adott tulajdonság minden értékével rendelkezik. A következőkben a protokollokat aszerint osztjuk családokra, hogy a csomópontok hogyan választja következő csomópontot a továbbítandó csomag útján, és minden így nyert családból bemutatunk egy reprezentáns protokollt, amelyeket részletesebben is ismertetünk.

3.1. Következő csomópont választása az üzenet tartalma alapján

Ezek a protokollok csupán a kérésben szereplő kért adat alapján döntenek arról, hogy melyik csomópontnak továbbítsák a kérést. Ez a modell illeszkedik legjobban a szenzorhálózatokhoz, hiszen a bázis többnyire nem egy konkrét szenzort akar lekérdezni, hanem egy bizonyos adat iránt érdeklődik. Ezen protokollok, paradigmák közül a Directed Diffusion [2] protokollt ismertetjük röviden. A Directed Diffusion több más protokoll alapjául szolgált (Energy Aware Routing, GBR stb.), így az itt leírtak részben azokra is érvényesek.

A bázis kezdetben elárasztja kérésével az egész hálózatot, amely tartalmazza a kért adatot leíró attribútumérték párokat. A kérés vételekor a csomópontok gradienseket állítanak be a kérést küldő csomópontra. A forrás így az üzeneteket ezen gradiensek mentén továbbítja a bázis felé, egy üzenetet akár több gradiens (több szomszédnak) küldve. A paradigma jól alkalmazható nyomkövetési alkalmazásoknál, és csak a közvetlen szomszédok egyértelmű címzését követeli meg. A gradiens minden csomópontnál definiálja a keresett adatot, amire illik a keresett attribútumérték pár, és a következő csomópontot, akinek az adatot majd tovább kell küldeni a bázis felé. Minden gradienshez egy súly van rendelve, amely súly egyenesen arányos a rajta küldött adat mennyiségével. Kezdetben a közbelső

csomópontok több szomszédtól is kaphatnak egyező kérést, így ennek megfelelően több gradienst is beállítanak a bázis felé. Idővel a legjobb minőségű út mentén a bázis növeli a gradiensek súlyát, amivel ezen úton növeli a forgalmat, míg másokon csökkenti. A közbelső csomópontok aggregálják a küldött adatot, és az aggregátumot továbbítják a megfelelő gradiensek mentén a gradiens súlyával arányos sebességgel. A bázis periodikusan újraküldi a kérést, amivel életben tartja az empirikusan megfelelő minőségű útvonalakat. Az egyes csomópontok különböző cache technikákkal növelik a protokoll robusztusságát és teljesítményét. A protokoll hátránya, hogy a kezdeti elárasztás költséges, és az empirikusan megfelelő minőségű utak kiválasztásáig a hálózat energiafogyasztása magas.

3.2. Véletlenszerűen választott következő csomópont

A véletlenszerűen választott következő csomópont mögötti szándék elsősorban a robusztusság és a egyenlő terheléelosztás elérése, illetve megközelítése. Ezek a protokollok nagyban alapoznak arra a tényre, hogy a szenzorok homogén, egyenletesen elhelyezett eszközök. Ezek közül az Energy Aware Routing [5] protokollt ismertetjük.

Az Energy Aware Routing protokoll fő célja a hálózat élettartamának növelése az egyenlő terheléelosztás megközelítésével. A protokoll futása során a cél csomópont (bázis) kezdeményezi a kommunikációt, amikor arra igény van. A szenzorok igyekeznek mindig különböző szomszédot választani a továbbításra. A választás véletlenszerű, ahol egy adott szomszéd választásának a valószínűsége fordítottan arányos annak költségével. A költség függ az adott szomszéd aktuális maradék energiaszintjétől, illetve a hozzá történő üzenet továbbításához szükséges energiától. A protokoll a szomszédok listáját, és az elérésükhöz szükséges energiát a MAC rétegtől kapja.

A protokoll három fázisból áll: inicializálás, üzenet továbbítás, útvonal karbantartása.

I. Inicializálás

Korlátozott elárasztás segítségével minden csomópont felépíti az útvonalválasztó tábláját, vagyis minden csomópont megállapítja a lehetséges útvonalakat a cél felé, és azok költségét.

1. A cél elárasztja a hálózatot a forrás irányába egy kérés üzenettel, aminek a költségmezejét nullára állítja: $Cost = 0$
2. Minden közbelső csomópont csak azon szomszédjainak továbbítja a kérést, amelyek a forráshoz közelebb, a céltól viszont távolabb helyezkednek el. Formálisan N_i csak akkor küldi el a kérést N_j szomszédjának, ha a következő egyenlőtlenségek teljesülnek: $d(N_i, N_S) \geq d(N_j, N_S)$, $d(N_i, N_D) \leq d(N_j, N_D)$, ahol $d(N_i, N_j)$ az N_i és N_j csomópontok távolságát jelenti, és N_S a forrás, N_D pedig a célcsoomópont azonosítója.

Rendszer modell													
No.	Protokoll	Bázisállomás						Szenzor				Cél	
		Szám	Mobilitás	Jelenlét	Lefedtettség	Lehelyezés	Átv. energia	Lefedtettség	Címzés		MAC felület		
									Kérés	Válasz			
1	Rumor Routing	Egy	Fix	*	Részleges	Véletlen	*	Részleges	Adat	ID	ID	*	Nem valós idejű
2	MCFA Energy Aware Routing	Egy	Fix	*	Részleges	*	*	Részleges	X	X	X	*	Nem valós idejű
3	Directed Diffusion	Több	Fix	*	Részleges	*	Allítható	Részleges	Adat	ID	ID	Kell	Nem valós idejű, élettartam
4	GBR	Több	Korlátozott	Folyamatos	Részleges	*	*	Részleges	Adat	ID	ID	*	Nem valós idejű
5	LEACH*	Több	Korlátozott	Folyamatos	Részleges	*	*	Részleges	Adat	ID	ID	*	Nem valós idejű, élettartam
6	LEACH*	Egy	Fix	Folyamatos	Részleges	Véletlen	*	Részleges	Adat	ID	ID	*	Nem valós idejű, élettartam
7	TEEN	Egy	Fix	Folyamatos	Részleges	Véletlen	*	Részleges	Adat	ID	ID	*	Nem valós idejű, élettartam
8	APTEEN	Egy	Fix	Folyamatos	Részleges	Véletlen	*	Részleges	Adat	ID	ID	*	Valós idejű, élettartam
9	PEGASIS	Egy	Fix	Folyamatos	Részleges	Véletlen	*	Részleges	Adat	ID	ID	*	Nem valós idejű, élettartam
10	ACQUIRE	Több	Korlátozott	Folyamatos	Részleges	*	*	Részleges	X	ID/Pozíció	ID/Pozíció	*	Nem valós idejű, élettartam
11	IDSQ/CADR	Több	Fix	Folyamatos	Részleges	*	*	Részleges	Adat	ID	ID	*	Nem valós idejű
12	Greedy Face Routing*	Több	Mobilis	*	Részleges	*	*	Részleges	Adat	ID/Pozíció	ID/Pozíció	*	Nem valós idejű, élettartam
13	DIR/GEDIR/MFR*	Több	Korlátozott	*	Részleges	*	*	Részleges	Adat/Pozíció	ID/Pozíció	ID/Pozíció	*	Nem valós idejű, élettartam
14	GEAR	Több	Korlátozott	*	Részleges	*	*	Részleges	Adat/Pozíció	ID/Pozíció	ID/Pozíció	*	Nem valós idejű, élettartam
15	MECN	Egy	Fix	*	Részleges	*	Allítható	Részleges	Adat	ID	ID	*	Nem valós idejű, élettartam
16	TTDD	Több	Mobilis	*	Részleges	*	*	Részleges	Adat	ID	ID	*	Nem valós idejű, élettartam
17	SAR (DAM)	Több	Korlátozott	*	Részleges	*	*	Részleges	*	ID	ID	*	Nem valós idejű, élettartam
18	HPAR	Egy	Fix	*	Részleges	*	Allítható	Részleges	Adat	ID	ID	*	Nem valós idejű, élettartam
19	SPEED	Több	Fix	Folyamatos	Részleges	*	*	Részleges	Adat/Pozíció	ID/Pozíció	ID/Pozíció	*	Valós idejű, élettartam

1. táblázat Hálózati modell és célkitűzés

Rendszer modell												
No.	Protokoll	Köv. szom.	Kommunikációs minta			Hierarchia	Kézbesítés	Számítás	Állapot	Riportolási modell		
			Sz-Sz	Sz-B	B-Sz					Idővezérelt	Kérésvezérelt	Eseményvezérelt
1	Rumor Routing	Véletlen	X	Rev. M.	Anycast	Nincs	Egy / Egy	Decentr.	van	X	Nem-agg.	X
2	MCFA	Összes	X	Rev. M.	X	Nincs	Egy / Egy	Decentr.	nincs	*	X	*
3	Energy Aware Routing	Véletlen	X	Rev. M.	Anycast	Nincs	Több / Egy	Decentr.	van	*	*	X
4	Directed Diffusion	Adat	X	Rev. M.	Anycast	Nincs	Több / Több	Decentr.	van	*	*	X
5	GBR	Adat	X	Rev. M.	Anycast	Nincs	Több / Több	Decentr.	van	*	*	X
6	LEACH*	Hierarchia	X	Rev. M.	Anycast	Van	Egy / Egy	*	van	*	*	X
7	TEEN	Hierarchia	X	Rev. M.	Anycast	Van	Egy / Egy	*	van	X	*	*
8	APTEEN	Hierarchia	X	Rev. M.	Anycast	Van	Egy / Egy	*	van	*	*	*
9	PEGASIS	Hierarchia	X	Unicast	Anycast	Van	Egy / Egy	*	van	*	*	X
10	ACQUIRE	Véletlen	X	Rev. M.	Anycast	Nincs	Egy / Egy	Decentr.	van	X	Nem-agg.	X
11	IDSQ/CADR	Pozíció	Unicast	Rev. M.	Anycast	*	Egy / Egy	Decentr.	van	X	*	X
12	Greedy Face Routing*	Pozíció		Unicast		Nincs	Egy / Egy	Decentr.	nincs	Nem-agg.	Nem-agg.	Nem-agg.
13	DIR/GEDIR/MFR*											
14	GEAR	Pozíció	Unicast	Rev. M., Uni	Anycast	Nincs	Egy / Egy	Decentr.	van	Nem-agg.	Nem-agg.	Nem-agg.
15	MECN	Pozíció	X	Rev. M., Uni	Anycast	Nincs	Egy / Egy	Decentr.	van	*	X	*
16	TTDD	Hierarchia	X	Rev. M.	Anycast	Van	Egy / Egy	Decentr.	van	Nem-agg.	Nem-agg.	X
17	SAR (DAM)	Hierarchia	X	Rev. M.	Anycast	Van	Egy / Egy	Decentr.	van	X	*	*
18	HPAR	Hierarchia	X	Rev. M.	*	Van	Egy / Egy	Centralizált	van	*	*	*
19	SPEED	Pozíció	Uni, Any, Mult	Uni, Any, Rev. M.	Uni, Any, Mult	Nincs	Egy / Egy	Decentr.	nincs	Nem-agg.	Nem-agg.	Nem-agg.

2. táblázat Működési modell

3. A kérés vétele során N_j kiszámítja az N_j -n keresztül haladó útvonal költségét a következőképpen:

$$C_{N_j, N_i} = Cost + Metric(N_j, N_i),$$

ahol $Metric(N_j, N_i)$ az N_j és N_i közötti metrikát jelenti (lásd később).

4. A magas költségű útvonalakat N_j nem veszi figyelembe és eldobja a megfelelő kérést, csak az alacsony költségű csomópontokat adja hozzá az útvonalválasztó táblákhoz:

$$FT_j = \{i \mid C_{N_j, N_i} \leq \alpha(\min_k C_{N_j, N_k})\}$$

5. Minden N_j csomópont egy valószínűséget rendel minden szomszédjához az FT_j táblában:

$$P_{N_j, N_i} = \frac{1/C_{N_j, N_i}}{\sum_{k \in FT_j} 1/C_{N_j, N_k}}$$

6. N_j kiszámolja a célhoz vezető utak átlagos költségét (amelyek az FT_j útvonalválasztó táblában szerepelnek):

$$Cost(N_j) = \sum_{i \in FT_j} P_{N_j, N_i} \cdot C_{N_j, N_i}$$

7. Ez az átlagos költség lesz a továbbítandó kérés új költségértéke: $Cost = Cost(N_j)$, majd az új kérést N_j továbbítja a forrás fele a 2. lépés szerint.

II. Üzenet továbbítás

1. A forrás elküldi az üzenetet a táblájában szereplő egy szomszédjának, ahol a szomszédot a hozzá rendelt valószínűséggel választja.
2. Minden közbenső csomópont hasonlóképpen választja ki a következő csomópontot a cél felé, vagyis választ egy szomszédot a táblájából a hozzá rendelt valószínűséggel.
3. Ez folytatódik addig, ameddig az üzenet eléri a célt (bázist).

III. Útvonal karbantartása

Az útvonal karbantartása minimális; nem túl gyakran a cél (bázis) elárasztással indirekt frissíti a szenzorok útvonalválasztó tábláját.

A metrikát, amit az 1.3 pontban használunk a következőképpen számoljuk ki: $C_{i,j} = e_{i,j}^\alpha R_i^\beta$, ahol $C_{i,j}$ az N_i és N_j csomópontok közötti költségmetrika, $e_{i,j}$ az egy üzenet átviteléhez szükséges energia N_i -ből N_j -be, R_i pedig N_i csomópont maradék energiája (az aktuális teleptöltöttség) normalizálva a csomópont indulási energiájával (kezdeti teleptöltöttség). Az α és β változtatható paraméterek, attól függően, hogy a hálózat élettartamát vagy a globális energiafogyasztást akarjuk minimalizálni.

A protokoll a Directed Diffusion paradigmához képest átlagosan 21,5%-kal több energiát spórol, és 44%-kal növeli a hálózat élettartamát (ha a hálózat élettartamát az első szenzorcsomópont kimerüléséig eltelt idővel számítjuk). Hátránya az egyes csomópontok esetleges komplikált címzése, illetve az inicializálási fázis megnövekedett kommunikációs költsége a Directed Diffusion protokollhoz képest.

3.3. Következő csomópont választása geometriai információ alapján

Az útvonalválasztó protokollok ezen családja a szomszédok ismert geometriai pozíciói és a cél pozíciója alapján döntenek a következő csomópontról a cél felé, ami lehet egy régió vagy egy konkrét csomópont. Ezzel elkerülük az elárasztás okozta többletterhelést, viszont többletköltséget jelenthet a pozíciók megállapítása, ami történhet statikusan, előreprogramozott módon, vagy valamilyen pozíció-meghatározó rendszer segítségével (pl. GPS). Többnyire mind a kérés, mind pedig a válasz útvonalirányítására ugyanazt a technikát használják. A legtöbb decentralizált pozíció alapú protokoll közös problémája a lokális minimum feloldása, amikor minden szomszédos csomópont távolabb helyezkedik el a céltől mint maga a továbbító csomópont. Ezen probléma feloldására különböző technikákat használnak az egyes protokollok. Itt most a GEAR (Geographical and Energy Aware Routing) [6] protokollt ismertetjük vázlatosan.

A GEAR esetében a csomópontok csak bizonyos szomszédoknak továbbítják a kérést, így több energiát spórolnak meg mint a Directed Diffusion. A válasz továbbítása történhet vagy a kéréshez hasonló módon, vagy pedig a Directed Diffusion esetében leírt módon. Minden csomópont a célcsomópontra (forrás) nyilvántart egy becsült és egy tanult költségértéket. Az utóbbi felelős a lyukak megkerüléséért és a becsült költségérték finomításának tekinthető. Ha az útvonalban nincsenek lyukak (minden csomópontnak van a célhoz közelebb eső szomszédja), akkor a két költségérték megegyezik. A protokoll két fázisból áll: üzenet továbbítás a célterület felé, valamint üzenet továbbítás a célterületen belül.

I. Üzenet továbbítás a célterület felé

Egy közbenső N csomópont a célterülethez közelebb eső szomszédokból választja ki azt a következő csomópontot a cél felé, amelynek tanult költsége a legkisebb. Ha ilyen nem létezik, akkor azt a szomszédot választja az összes közül, amelynek becsült költsége a legkisebb. Kezdetben a tanult költség a célterületre megegyezik a becsült költséggel, ahol a becsült költség formulája a következő:

$$c(N_i, R) = \alpha d(N_i, R) + (1 - \alpha)e(N_i),$$

ahol $d(N_i, R)$ a szomszédos N_i csomópont távolsága a célterület középpontjától, $e(N_i)$ az N_i csomópont normalizált maradék energiája, α pedig egy állítható paraméter. Minden csomópont kiszámítja saját becsült költségét, majd elküldi minden szomszédjának, így minden csomópont értesül minden szomszédjának a becsült költségéről. Kezdetben a tanult költség minden N_i csomópontra egy adott célterületre $h(N_i, R) = c(N_i, R)$. Mivel egy N csomópont elküldte az üzenetet a minimális tanult költségű szomszédjának aminek az azonosítója N_{min} , frissíti a saját tanult költségét a következő módon:

$$h(N, R) = h(N_{min}, R) + C(N, N_{min}),$$

ahol $C(N, N_{min})$ az üzenet továbbításának költsége N -ből N_{min} -be (lehet a küldéshez használt energia mennyisége, a távolság, a normalizált maradék energia,

illetve ezek kombinációi). Így, ha az utat n csomópont alkotja, akkor az útvonal tanult költsége n lépésben konvergál az útvonal valódi költségéhez. A csomópontok a tanult költségük értékét bizonyos időközönként elküldik az összes szomszédjuknak. Így a tanult költségek használata a megfelelő frissítési technikával lehetőség adnak a lyukak kikerülésére.

A GEAR nem csupán hosszabb élettartamot biztosít a szenzorhálózatoknak mint más pozíció alapú útvonaválasztó protokollok, de a kézbesített csomagok száma akár 80%-kal több lehet, mint más pozíció alapú protokolloknál.

II. Üzenetovábbítás a célterületen belül

Miután az üzenet elérte a célterületet, az ottani csomópontok korlátozott vagy pedig rekurzív elárasztásos technikát használnak az üzenet elterjesztésére a területen belül. Ha a csomópontok ritkán helyezkednek el, akkor a korlátozott elárasztás, míg sűrűn elhelyezett csomópontok esetén a rekurzív elárasztásos technika javasolt. Az utóbbi esetben a területet négy, körülbelül egyenlő alterületre osztjuk, és minden alterületre továbbítjuk az üzenet egy másolatát. Ez a felosztásos folyamat rekurzív módon addig folytatódik, amíg csak egy csomópont marad egy alterületen belül.

3.4. Hierarchia szint alapján választott következő csomópont

A hierarchikus protokollok esetében minden csomópont a hierarchiában fentebb elhelyezkedő csomópontnak (aggregátor) küldi az üzenetet, ahol a hierarchia csúcsán a bázis helyezkedik el. Az egyes csomópontok a bejövő üzeneteket aggregálás után küldik tovább, amivel jelentősen csökkenthetik az adatforgalmat, így energiát spórolnak meg, aminek eredménye a hálózat megnövekedett élettartama. A hierarchikus protokollok másik nagy előnye, hogy jól skálázhatóak.

Az egy aggregátornak továbbító csomópontok halmazát klaszternek, míg az aggregátort klasztervezetőnek is nevezik. Klasztervezetőknek mindig olyan csomópontokat választanak (statikusan vagy dinamikusan), amelyek nagyobb erőforrással (általában nagyobb teleptöltöttséggel) rendelkeznek, mivel a többi csomópont-hoz képest nagyobb forgalmat bonyolítanak le és több számítás végeznek. A hierarchikus útvonalválasztó protokollok alappillére az klaszterek kialakítása és a klasztervezető megválasztása, majd erre épülve az útvonalválasztás megvalósítása a klasztervezetők felé. A következőkben a *LEACH (Low Energy Adaptive Clustering Hierarchy)* [3] protokollt vázoljuk fel.

A LEACH protokollban a csomópontok dinamikusan, elosztott módon alakítják ki a klasztereket. A LEACH futásakor minden klasztervezető véletlenszerűen választódik meg, és ez a szerep dinamikusan rotálódik az egyenletes terheléselosztás végett. Minden klasztervezető a klaszterből érkező üzeneteket aggregálja, majd az aggregátumot közvetlenül elküldi a bázisnak. Így a LEACH feltételezi, hogy minden csomópont ké-

pes elérni a bázisállomást. A LEACH TDMA/CDMA csatornahozzáférési modellt követ a klaszteren belüli, illetve klaszterek közötti ütközések feloldására. Szimulációk szerint a csomópontok mintegy 5%-ának kell csak klasztervezetőnek lennie egy adott pillanatban a hálózat optimális energiafogyasztásához. A LEACH működése két fázisból áll, amelyek periodikusan ismétlődnek, a két fázis együtt alkot egy menetet.

I. Felépülési fázis

A felépülési fázis során létrejönnek a klaszterek és megválasztódnak a klasztervezetők. A csomópontok egy előre definiált p (pl. 0,05) hányada választja meg magát minden menet elején klasztervezetőnek a következőképpen. Egy n szenzor választ magának egy véletlen számot 0 és 1 között, és ha ez szám kisebb mint egy $T(n)$ küszöbérték, akkor a szenzor klasztervezetőnek deklarálja magát, ahol

$$T(n) = \frac{P}{1 - p \cdot (r \bmod 1 / p)} \quad \text{ha } n \in G.$$

Itt r jelöli az aktuális menetszámot, G pedig azon csomópontok halmaza, amelyek még nem voltak klasztervezetők az elmúlt $(1/p)$ menet során. Ezek után a magukat vezetőnek deklaráló csomópontok értesítik a körülöttük levő csomópontokat a választás eredményéről egy bizonyos energiaszinten küldött üzenetben. Minden csomópont azt a csomópontot választja saját klasztervezetőjének, akitől a legnagyobb jelerősséggel kapta az üzenetet, majd erről informálja a választott vezetőt is. Ezután a vezető létrehozza a megfelelő TDMA ütemezést, ahol minden hozzá tartozó csomópontnak lefoglal egy időrészt, majd az ütemezési információt elküldi a klasztertagoknak.

II. Működési fázis

A működési fázis során a klaszterek tagjai a mért adatot a vezetőnek továbbítják, amely, miután minden adatot megkapott, az aggregátumot továbbküldi a bázisnak. A menet végén a hálózat átvált felépülési fázisba és új klaszterek alakulnak.

A működési fázis célszerűen jóval nagyobb mint a felépülési fázis a túlterheltség csökkentése végett. A protokoll hátránya, hogy nem alkalmazható nagy területen elhelyezett hálózatnál, hiszen minden csomópontnak el kell érnie a bázist közvetlenül, továbbá feltételezi, hogy minden csomópont folytonosan küld adatot a vezetőnek. Továbbá nem feltételen igaz az, hogy a vezetők a hálózatban egyenletesen helyezkednek el, így néhány csomópontnak lehetséges, hogy nem lesz vezetője. A protokoll feltételezi, hogy minden csomópont egyenlő kezdeti energiával rendelkezik. A legnagyobb gondot mindezek ellenére talán mégis a felépülési fázis okozta többletterhelés jelentheti.

3.5. Minden szomszédnak történő továbbítás

Ezen protokollok működése rendkívül egyszerű. Minden csomópont a hálózatban önmaga dönt arról, hogy a kapott üzenetet továbbítania kell-e vagy sem. Ha

igen, akkor minden szomszédjának elküldi az üzenetet (egyetlen többszörös üzenetben), ha nem, akkor pedig eldobja az üzenetet. Ezen protokollok képviselője az *MCFA (Minimal Cost Forwarding Algorithm)* [4] protokoll. Az MCFA protokoll nagy előnye, hogy semmilyen információt sem tárolnak a csomópontok a szomszédjaikról, csak a saját maguk költségét a bázishoz viszonyítva.

A protokoll két részből áll. Az első részben minden csomópont megállapítja a saját költségét a bázishoz képest, aminek érdekében a bázis elárasztja a hálózatot egy kezdetben $C = 0$ értékű költséget tartalmazó üzenettel. Kezdetben minden csomópont költsége végtelen. Minden N_i csomópont, amely megkapja ezt az üzenetet egy N_j csomóponttól, vár $\alpha \cdot C_{N_i, N_j}$ ideig (α választásánál figyelembe kell venni a csomópontok közötti kapcsolatok késleltetését, hibáit, a csomópontok késleltetését stb.), ahol C_{N_i, N_j} az N_i és N_j csomópontok közötti kapcsolat költsége (energia, késleltetés stb.), majd frissíti az üzenetben szereplő C értéket: $C = C + C_{N_i, N_j}$, beállítja a saját költségét erre az új C értékre, és továbbítja a frissített üzenetet. Bebizonyítható, hogy ideális esetben (amikor α elég nagy) minden csomópont csak egyetlen ilyen üzenetet küld tovább, ami a csomópont minimális költségét tartalmazza a bázistól.

A protokoll második részében minden csomópont képes már üzenetet küldeni a bázis felé a következő módon. A N küldő szenzor elhelyezi a saját C_N minimális költségét a küldendő üzenetbe, és többesküldéssel elküldi azt szomszédainak. Minden M csomópont, amely hallja az üzenetet, ellenőrzi, hogy $M - C_{N, M} = C_M$. Ha igen, akkor M a minimális költségű útvonalon van, ezért többesküldéssel továbbküldi az üzenetet. Egyébként M eldobja az üzenetet.

A protokoll hátránya, hogy minden olyan csomópont, amely vesz egy üzenetet, azt mindenképpen értelmezi is, ami többletterhelést jelent.

4. Összegzés

Jelen írásunkban a vezeték nélküli szenzorhálózatokban alkalmazott útvonalválasztó protokollokat tekintettük át röviden. A cikk első felében bemutattuk a létező útvonalválasztó protokolloknak egy új osztályozási szempontrendszerét. A protokollokat rendszereztük hálózati modelljük, működési modelljük és célkitűzésük szerint. Táblázatokba rendezve felsoroltuk a létező jelentősebb útvonalválasztó protokollokat hálózati modell és cél, valamint működési modell szerint. A cikk második felében részletesen bemutattuk néhány konkrét protokoll működését. A bemutatott protokollok egy-egy protokollcsalád képviselői voltak, ahol a családot az határozta meg, hogy a csomópontok hogyan választják a következő csomópontot a cél felé vezető úton.

Köszönetnyilvánítás

Az itt bemutatott munkát részben támogatta a HSN Laboratórium, a UbiSec&Sens EU projekt (www.ist-ubisecsens.org), és az OTKA (T046664).

Irodalom

- [1] J. N. Al-Karaki, A. E. Kamal:
Routing techniques in wireless sensor networks: a survey.
In IEEE Wireless Communications, Vol.11, 2004.
pp.6–28.
- [2] C. Intanagonwiwat, R. Govindan, D. Estrin:
Directed Diffusion: a scalable and robust communication paradigm for sensor networks.
In Proceedings of ACM MobiCom 2000, Boston, MA, pp.56–67.
- [3] W. Heinzelman, A. Chandrakasan, H. Balakrishnan:
Energy-Efficient Communication Protocol for Wireless Microsensor Networks.
In Proc. of the 33rd Hawaii International Conference on System Sciences (HICSS 2000), January 2000.
- [4] F. Ye, A. Chen, S. Liu, L. Zhang:
A scalable solution to minimum cost forwarding in large sensor networks.
In Proc. of the 10th International Conf. on Computer Communications and Networks (ICCCN 2001), pp.304–309.
- [5] C. Rahul, J. Rabaey:
Energy Aware Routing for Low Energy Ad Hoc Sensor Networks.
In IEEE Wireless Communications and Networking Conference (WCNC), March 17-21, 2002, Orlando, FL, Vol.1, pp.350–355.
- [6] Y. Yu, D. Estrin, R. Govindan:
Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks.
UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023, May 2001.