Budapesti Műszaki és Gazdaságtudományi Egyetem

Vezetéknélküli smart metering rendszer kidolgozása

Villamosmérnöki és Informatikai Kar
Híradástechnikai Tanszék

Milánkovich Ákos
Ill Gergely
Varga Norbert

Konzulensek:
Lendvai Károly
Dr. Szabó Sándor

2011.

Összefoglaló

Napjainkban a kutatások és fejlesztések egyik fő irányát az intelligens mérőrendszerek jelentik. Egyre több ország vezeti be a saját megoldását a feladatra, mert jelentős előnyökkel jár az automatizált leolvasás az eddigi gyakorlattal szemben. Hazánkban is egyre növekszik az ipari igény erre a megoldásra, mivel bevezetése Európai Uniós kötelezettségünk [1].

A TDK dolgozat célja egy olyan intelligens mérőrendszer megalkotása, amely energia-hatékony vezeték nélküli protokollt használ a szenzorhálózat elemei közötti kommunikációhoz.

A dolgozatban áttekintésre kerülnek a jelenlegi protokollok és mikrovezérlők, elemezzük azok előnyeit és hátrányait, majd egy saját protokollal és mikrokontrollerrel mutatunk új megoldást a problémára. Mindezt egy olyan vezeték nélküli technológiával valósítjuk meg, amely lehetővé teszi, hogy a kommunikációhoz felhasznált energiát olyan alacsony értéken tartsuk, hogy a szenzor évekig zavartalanul működhessen egy pár ceruzaelemmel. Egy szimulációs környezetet felhasználva verifikáljuk a kidolgozott protokoll helyes működését. Bemutatjuk az elvégzett mérések eredményeit, illetve egy matematikai módszert, amely segítségével a protokollok energiahatékonysága mérhető és összehasonlítható.

A rendszer lehetséges alkalmazási körei: gáz, áram, víz fogyasztásának mérése távolról skálázható módon. A megoldás működőképes falvakban, ritkán lakott területeken és városi társas- és panelházakban egyaránt.

Abstract

Nowadays one of the main focus areas in Research and Development is smart metering. More and more countries are introducing their own solutions for the task, as automated reading of data has tremendous advantages over the current methods. An industrial need is getting stronger for a solution also in Hungary, because of the European Union directives [1].

The goal of this paper is to build an intelligent metering system which uses energy efficient wireless protocol for communicating among the nodes of the sensor network.

This paper gives a brief introduction about the available protocols and microcontrollers, analyze their pros and cons, then provide an own protocol as a new solution for the problem. All this is implemented by a wireless technology, which enables as less power consumptions for communication, that the sensor is operable with a pair of batteries for years. Utilizing a silmulation framework the protocol's appropriate operation is verified. The paper presents our measurements and a mathematical formula, which calculates and makes us able to compare the energy efficiency of protocols.

Possible applications areas of the system: metering of gas, electricity, water consumption in a remote and scalable way. The solution is operable in villages, sparsely populated areas and also in blocks of flats in cities.

Tartalomjegyzék

1. fejezet

Bevezetés

A TDK dolgozatunk az alábbi felépítést követi: Ebben a fejezetben megmutatjuk, milyen motivációk miatt érdemes smart metering-gel foglalkozni, majd általános áttekintést nyújtunk a témáról. Definiáljuk az alapfogalmakat, amelyekre a későbbiekben építünk és megfogalmazzuk a céljainkat.

Az előzményekben bemutatjuk, hogy milyen jelenlegi protokoll és hardver megoldásokat találtunk, ezek milyen képességekkel rendelkeznek.

Ezután áttérünk a saját munkánkra, a kifejlesztett hardverre és a protokollok részletes összehasonlító elemzésére. Ismertetjük a kidolgozott energiahatékonysági képletet, az elvégzett méréseket. Bemutatjuk a saját fejlesztésű protokollunkat, és a rajta végzett szimulációk kimenetét.

Végül összefoglaljuk az elért eredményeket.

1.1. Motiváció

Napjainkban a vezeték nélküli hálózati technológiák virágkorukat élik. Egyre jobban elterjedtek az életünk minden területén, felváltva a vezetékes technológiákat, amelyekkel szemben számos előnyük fontosabbnak bizonyult a hátrányaiknál. A hagyományos mérőóra leolvasást is kezdik felváltani távolról leolvasható berendezésekre, illetve olyan eszközökre, amelyek képesek jelentést küldeni a fogyasztás mértékéről.

Az új eszközök egy része az elektromos hálózatot használja a kommunikáció lebonyolítására (Power Line Communication) azonban ez nem minden esetben optimális megoldás. A vezeték nélküli technológiák sok esetben megkönnyítik az eszközök telepíthetőségét, mivel teljesen mobilak. A köztük lévő kommunikációt azonban védeni kell a lehallgatás és manipuláció ellen, amely visszaélésekhez vezethet. További probléma lehet a távolság és a jel-zaj viszony váltakozása miatt az eszközök állandó láthatósága.

A megoldás egy hierarchikus mesh hálózat, amely adaptív módon képes megszervezni a csomópontok között az útvonalat, amelyen keresztül a mért adat egy központi elemhez jut.

1.2. Áttekintés

Számos országban létezik smart metering rendszer, általában egyéni megoldásokkal. A világ legnagyobb (27 milliós) smart metering rendszere Olaszországban üzemel. Az ottani megoldás az elektromos hálózat vezetékein keresztül kommunikál. Rendelkezik a kétirányú kommunikáció képességével, menedzsment funkciókkal, távoli szabályozással. A rendszert 2000 és 2005 között az Enel cég építette ki. Hátránya azonban, hogy az eszközök firmware-e nem frissíthető. További rendszerek működnek Japánban, az Egyesült Államokban, az Egyesült Királyságban, Kanadában, Ausztráliában, Hollandiában, illetve bevezetés alatt áll Franciaországban, Írországban és Máltán. [3]

A Berg Insight legújabb kutatási eredményei alapján a smart metering eszközöket használó háztartások száma 2015-re eléri Európában a 130 milliót, míg ez a szám 116,5 millió összesítve Ázsia, Ausztrália és Óceánia területén. [4]

A smart metering legnagyobb technológiai kihívása a kommunikáció. Minden egyes eszköznek megbízhatóan és biztonságosan el kell juttatni a mért eredményeket egy központi helyre. Tekintve, hogy a mérések helyszíne és típusa is nagyon változó, így rengeteg megoldás született, pl. mobilhálózaton keresztül, műholdakon, bérelt vagy nyilvános rádióvonalon, elektromos hálózati vezetékeken. A használt médiumon kívül a használt hálózat is fontos megkülönböztető szempont: fix vezeték nélküli, vagy mesh hálózat, vagy a kettő kombinációja is lehetséges. Jelenleg egyik megoldás sem alkalmas minden igény kielégítésére. A ritkán lakott területek és a városi környezet rendkívül különböző követelményeket támaszt. A továbbiakban a vezetéknélküli mesh hálózattal foglalkozunk, mert ez biztosítja a legnagyobb rugalmasságot. A megfelelő frekvencia kiválasztása kritikus, ugyanis a frekvencia teljes mértékben meghatározza a jel alábbi tulajdonságait:

- átviteli sebességget befolyásolja
- hatótávolsággal fordítottan arányos
- zajérzékenység a sávszélességgel egyenesen arányos
- kommunikációs hatékonyság a sávszélességgel fordítottan arányos
- áthatoló-képességgel összefügg

A paraméterek összehasonlításakor fontos szempont volt, hogy a kommunikáció megbízhatósága nagy, illetve az egyszerű mérési adatok miatt az átküldendő csomagok mérete pedig kicsi legyen. Minden szempontot figyelembe véve a 433 MHz-es frekvencia ideálisnak tűnik a probléma megoldására, hiszen nem egy túl zsúfolt frekvenciasávban tartózkodik (rádió, TV, műhold, Wifi, stb) és megfelelő áthatolóképességű és hatótávolságú egyszerre.

1.3. Célok

A cél egy olyan intelligens mérőrendszer megalkotása, amely energiatakarékos vezetéknélküli protokollt használ a szenzorhálózat elemei közötti kommunikáció során.

A kitűzött cél megvalósítása érdekében a feladatot az alábbi konkrét lépésekre bontottuk:

1. Az eddigi technológiák megismerése (protokollok és eszközök), részletesen elemezve a bennük rejlő lehetőségeket, előnyöket és hátrányokat.
2. Smart metering-re alkalmas frekvenciák összehasonlítása mérésekkel különböző környezetekben.
3. A legalkalmasabb frekvencián működő energia-hatékony mikrokontroller és rádiós modul tervezése.
4. Saját protokoll tervezése intelligens mérőrendszer feladatokra, amely a következő célokat teljesíti:
   a. energia-hatékonyság
   b. robusztusság
   c. skálázhatóság
   d. biztonságosság
5. A megtervezett protokoll szimulációs vizsgálata.

1.4. Definíciók

Mikrokontroller

A mikrokontroller egyetlen lapkára integrált, általában vezérlési feladatokra optimalizált számítógép. Költséghatékonyan képes ellátni egyszerű, kis számítási teljesítményt és operatív tárat igénylő műveleteket. A tervezés során törekednek arra, hogy minél kevesebb járulékos alkatrésszel lehessen megoldani a feladatok legszélesebb skáláját amellett, hogy az eszköz fogyasztását, méretét és költségét minimalizálják. Ezt az IC lábainak multiplex felhasználásával és beépített perifériákkal érik el. Programozásuk a logikai magas szintnél nagyobb égetőfeszültség alkalmazásával történik. A régi típusok egyszer voltak programozhatóak, de az új eszközök gyakorlatilag mindegyike Flash-ROM alapú programtárat tartalmaz, így akár egymillió beírás/törlés ciklust is elviselnek.

Smart Metering

A smart metering a mai hagyományos mérőórákat váltja fel, gyakorlatilag digitálissá tenné azokat. Az okos mérési eszközök segítségével a fogyasztó sokkal könnyebben tudná nyomon követni aktuális energia-felhasználást (víz, villany, gáz), ráadásul a szolgáltatók is folyamatos képet kaphatnának ügyfeleik fogyasztási szokásairól. A smart metering technológia kétirányú kommunikációt tesz lehetővé az eszközök és a központ között, valamint menedzsment funkciókkal is rendelkezik. [5]

Előnyök a szolgáltatónál:

- adatok pontossága növekszik
- adminisztratív költségek csökkentése
- veszteségek folyamatos figyelése
- több alkalom adódik a kiskereskedelmi szolgáltatás megújítására

Előnyök a fogyasztónál:

- kényelmes szolgáltatóváltás
- nem kell megfizetnie a mérőleolvasást és a számlázást
- figyelemmel kísérhető fogyasztás
- átlátható ár struktúra

Mesh hálózat

A mesh egy olyan vezetéknélküli hálózati topológia, ahol a csomópontok nem csak a saját adataikat továbbítják, hanem a többi csomópont számára továbbító feladatokat is ellátnak, így kollaborálva továbbítják az információt a hálózatban. Két alapvető tervezési módszer valósítja meg a kívánt működést: az elárasztás (flooding) és az útvonalválasztás (routing). Az elárasztás nagyon sok üzenet fölösleges küldésével az összes csomóponton átviszi az információt. Az útvonalválasztás egy kiszámított útvonal mentén eszközről eszközre ugrásonként (hop) továbbítja az adatokat, azonban feltételez egy önszerveződő-önkonfiguráló-helyreállító képességet.

2. fejezet

Előzmények

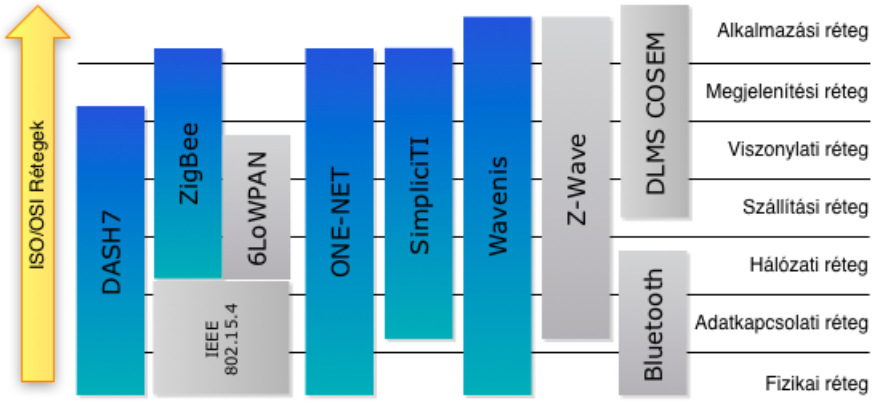
Ebben a fejezetben bemutatjuk, hogy milyen jelenlegi protokoll és hardver megoldásokat találtunk, ezek milyen képességekkel rendelkeznek.

2.1. Protokollok

A megoldások vizsgálta során elsődleges szempont volt, hogy az egyes protokollok open-source megoldások legyenek. Ennek eredményeként a jelenlegi megoldásokat feltérképezve a következő protokollokat találtuk számunkra legalkalmasabbnak intelligens mérőrendszerek kialakításra:

- DASH7
- Zigbee
- 6LoWPAN
- ONE-NET
- SimpliciTI
- Wavenis
- Z-Wave
- Bluetooth
- DLMS/COSEM

A vizsgált protokollok és szabványok közül a DASH7-es megoldással foglalkoztunk nagyobb részletességgel, mert ezt találtuk számunkra megfelelőnek és jó kiindulópontnak rendszerünk megtervezéséhez, de a többi is hasznos ismeretekhez juttatott bennünket. A 2.1 ábrán összefoglaljuk, hogy a kiválasztott protokollok és szabványok mely rétegeket implementálják az ISO/OSI hierarchiából.



2.1. ábra. A kiválasztott protokollok ISO/OSI rétegekbe sorolása [6]

2.1.1. DASH7

A DASH7 egy aktív RFID [7] szabványon (ISO/IEC 18000-7 [8]) alapuló vezeték-nélküli technológia. 2009 januárjában az USA Védelmi Minisztériuma egy 429 millió dolláros szerződést kötött DASH7 eszközök fejlesztésére a Savi Technology, Evigia Systems, és Identec Solutions hardvergyártókkal. 2009 márciusa óta a DASH7 Alliance [5], egy non-profit ipari konzorcium szorgalmazza a szabvány terjesztését. 2010 júliusában több mint 50 résztvevőjük volt 23 országból. Napjainkban az eredetileg katonai célokra fejlesztett technológiát kereskedelmi célokra is használni kezdik más eddigi vezeték nélküli technológiákkal (ZigBee, Bluetooth) szemben. A DASH7-et olyan hálózati alkalmazásokhoz használják, melyek alacsony energiafogyasztásúak és az adattovábbítás lassabb és szórványosabb, mint a telekommunikációs alkalmazásokban.

Felhasználási területek: katonai alkalmazások (főként vadászgépekben), szállító konténerek azonosítása és követése, szórakoztató elektronikai cikkek, helymeghatározás, személyazonosítás, orvosi alkalmazások, öntözőrendszerek vezérlése, smart metering, erózió, páratartalom és földrengés mérések, mobil hirdetések, épület automatizálás (intelligens otthon), jegykezelés, szociális hálók, logisztika stb.

Technikai összefoglaló

A DASH7 egyik főbb technikai jellemzője az alacsony energia-fogyasztás (30-60 μW), melynek segítségével egy eszköz elemének élettartama akár több évet is elérhet. A működési frekvenciának köszönhetően hatótávolsága LOS esetén garantáltan 1 kilométer, de elérheti akár a 2 kilométert is. A szabvány adatátviteli sebessége 27,8-200 kbit/s terjedhet, mely elfogadható nagyságú az alkalmazási területein. Mindemellett a késleltetés mozgó eszközök esetén 2,5-5 s közötti, de átlagosan 2 s. A rádiós kommunikációt (G)FSK moduláció segítségével oldja meg, az SNR alacsony értéken

tartásával. Az előbb felsoroltakat kis memóriaigény mellett képes megvalósítani (5 kB protokoll stack). A szabvány támogatja a multi-hop-os megoldást, mely lehetővé teszi a több eszközön keresztüli kommunikációt. A használt 433 MHz-es frekvencia az ISM (industrial, scientific and medical) sávból kerül ki, mely az egész világon szabadon használható. Ez a frekvencia teljesen alkalmas vezetéknélküli szenzorhálózatok kialakításához, mert a jelterjedési tulajdonságai megfelelőek (áthatol a vízen és a betonon is), illetve a jel kis teljesítménnyel is képes nagy távolságokat lefedni.

A DASH7 a vezetékes session alapú technológia helyett a BLAST tervezési koncepció segítségével került megvalósításra:

- Bursty - börsztös

  Az adatátvitel hirtelen változó, azaz bizonyos ideig az adatforgalom kicsi, majd egy adott pillanattól kezdve pedig rövid időre hirtelen megugrik.

- Light - kis csomagméret

  A legtöbb alkalmazás csomagmérete 256 byte-ra korlátozott. Előfordulhat, hogy egy adat több csomagban megy át, de ezt általában elkerüljük.

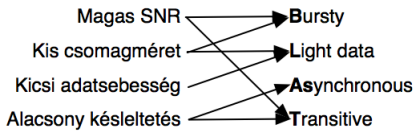
- ASynchronous - aszinkron

  A kommunikáció kérés-válasz alapú, így nincs szükség hand-shake algoritmusra vagy szinkronizáló eszközökre.

- Transitive - hordozható

  A DASH7 eszközök mobilak vagy hordozhatóak, feltöltés centrikusak a többi letöltés orientált vezetéknélküli technológiával ellentétben, illetve nincs szükség kialakított fix hálózati struktúrára (pl.: bázisállomás) sem.

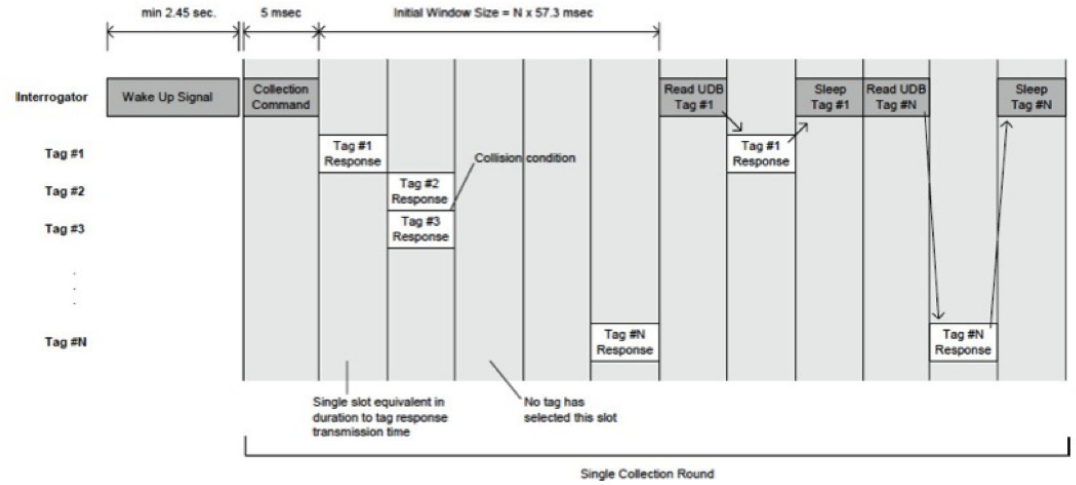
A 2.2 ábra a BLAST koncepció DASH7-beli megvalósítását szemlélteti.



2.2. ábra. BLAST

Működés

A DASH7 szabvány master-slave modellel definiálja az eszközök közötti kommunikációt. A kommunikációt Manchester-kódolással valósítja meg fizikai szinten. Az elküldött üzeneteket 16 bites CRC-vel látja el a hibák detektálására. A kommunikációt mindig a master, az “interrogator” kezdeményezi, polling módszerrel kérdezi le a “tag” adatait. A tagek adatainak begyűjtését az 2.3 ábra szemlélteti.



2.3. ábra. Tagek adatainak begyűjtése idődiagramon

Az interrogator egy 31,25 kHz-es jelet (Wake up signal) bocsát ki 2,35-4,8 másodpercig. Ez felébreszti a hatótávon belül alvó állapotban lévő tageket, amelyek készenléti állapotba kerülnek. Ezután egy broadcastolt Collect (begyűjtés fázis) üzenetet küld az interrogator, hogy megtudja mely eszközök érhetőek el (felderítés fázis). A Collection üzenet ablakokat, ezen belül pedig slotokat definiál, amik közül a tagek véletlenszerűen választanak egyet, amiben válaszolni fognak. Ha egy slotban több tag válaszol, akkor ütközés keletkezik (vagyis hibás lesz a CRC), ekkor az ablak lejártával az interrogator egy újabb Collection üzenetet küld, amiben újra lehetőségük lesz adni az előbb ütközött tageknek. Ahhoz, hogy a többi tag már ne válaszoljon erre az új begyűjtésre, az interrogator Sleep üzenetet küld a sikeresen kommunikáló tageknek point-to-point módon (azaz ezt az üzenetet mindig csak egy tag kaphatja meg).

Az eszközök jól definiált formátummal rendelkező üzenetekkel kommunikálnak egymással. A szabvány egyedi (point-to-point) és broadcast címzési módokat alkalmaz a kommunikáció megvalósítására. A point-to-point címzés egy 6 byte-os összetett cím (Tag Manufacturer ID + Tag Serial Number) segítségével történik. A broadcast címzési mód esetén minden hatótávolságon belüli eszköz megkapja az üzeneteket.

A DASH7 az elküldött adatok strukturálására Univerzális Adatblokkokat (UDB) használ, amelyek típus, hossz és max. 255 byte hosszú adat elemekből tevődnek össze. [9] [10] [11] [12]

OpenTag

Az OpenTag egy nyílt forráskódú DASH7 szoftver stack, amely C nyelven került implementálásra és különféle mikrokontrollereken futtatható. Emiatt az OpenTagnek nagyon kompaktnak kell lennie, azonban megfelelő konfiguráció mellett futtatható bármilyen POSIX környezetben. Érdemes megemlíteni, hogy az OpenTag

biztosítja a DASH7 összes funkcióját nem csak "tag" eszközökre. Az implementáció figyelmet fordít a biztonságos kommunikációra is, melyhez kriptográfiai primitívek támogatását nyújtja. Az OpenTag tervezése során kiemelkedő figyelmet fordítottak a hordozhatóságra, hogy minél több platformon futtatható legyen. Felépítése három lényegi komponensre bontható, annak érdekében, hogy szétválasztható legyen az alapkönyvtáraktól a platformfüggő kód és a felhasználói program kódja. [13] [14]

Értékelés

A következőkben összegyűjtöttük a DASH7 szabvány előnyös és hátrányos tulajdonságait. A legfőbb előnyök a következők voltak: az ISM frekvencia miatt könnyű a telepítés, elhelyezés és karbantartás. Szintén a frekvenciaválasztásnak köszönhető, hogy más népszerű technológiákkal, mint WLAN és Bluetooth nincs interferencia, a jel áthatol a falakon, betonon és vízen a 433 MHz-es frekvencia miatt, valamint nagy hatótávolságot (1,5 km) biztosít alacsony teljesítménnyel. Fontos szempont az előzőeken kívül, hogy az eszközök ára viszonylag alacsony $ 10 nagyságrendbe esik. A smart metering szempontjából fontos előny, hogy a szabvány feltöltés-orientált, képes a tag-to-tag kommunikációra, mely lehetőséget ad a vezetéknélküli "mesh" szenzor hálózatok leváltására, illetve hogy az alkalmazott parancs-válasz kommunikáció egyedi parancsokkal is bővíthető.

A szabvány hátrányai az alábbiak: a DASH7 eredetileg RFID szabvány, így közvetlenül nem használható smart meteringre, nincs lehetőség hierarchikus node szervezésre, a tisztán master- slave kommunikáció miatt. Interferencia léphet fel az ISM frekvencia miatt (pl.: autók központi zár vezérlését is zavarhatja), illetve a csatorna keskeny sávszélessége miatt interferencia érzékeny. Hiányosság, hogy nincsenek beépített biztonsági mechanizmusok, mint például titkosítás és hitelesítés. A nem túl magas adatátviteli sebesség miatt nem alkalmas nagy adatmennyiség átvitelére. Az open source implementáció jelenleg fejlesztési fázisban van, kevés fejlesztővel.

2.1.2. ONE-NET

A ONE-NET Open-source vezetéknélküli standard, amit alacsony költségű és fogyasztású hálózati alkalmazásokhoz dolgoztak ki (pl. biztonsági megoldások, szenzorhálózatok, otthon-automatizálás) hardver- és frekvencia függetlenül. Az információ kódolására FSK-t használ, a sávszélességet dinamikusan állíthatjuk a követelményeknek megfelelően. Támogatott topológiák:

- csillag: csökkenti a komplexitást és eszközköltséget, valamint egyszerűbb kulcsmenedzsmentet biztosít
- p2p: egy mester eszköz konfigurálja és autentikálja a tranzakciókat
- mesh: nagy távolságok lefedésére, routing támogatással

Támogatott tranzakciók:

- egyszerű
- blokk
- stream (adatfolyam)

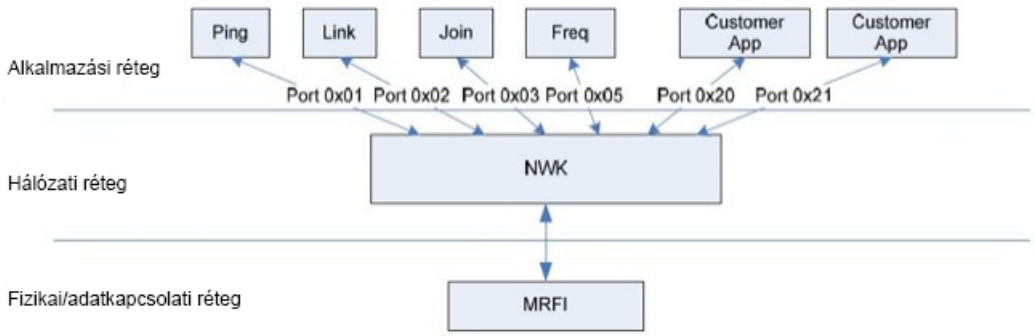
Rengeteg adattípust és üzenettípust előre definiál áram/gáz/hőmérséklet stb. mérésére, támogatja az üzenetek titkosítását (XTEA blokk-rejtjelező), véd a visszajátszásos támadástól és kulcs-menedzsmentet is biztosít. [15]

2.1.3. SimpliciTI

A Texas Instruments ingyenes hálózati protokoll stack-je, amely egyszerű API-t biztosít a rádiós eszközök közötti kommunikációra. E mögött a protokoll mögött áll ki a cég, ezzel garantáltan működnek az eszközei, példakódot és tutorialt mellékelnek a letölthető telepítőben [16]. A SimpliciTI [17] kifejezetten alkalmas smart metering technológiák megvalósítására. Tervezésekor nagyon fontos alapelv volt a hatékony energia-kihasználás (alacsony fogyasztás). Az eszközök a szerepeiknek megfelelően fogyasztanak, az idő nagy részét sleep módban töltve, ha lehet. Két egyszerű topológiát lehet megvalósítani az API-val: csillag és p2p. Az eszközök szerepei a következők lehetnek:

- End Device: a hálózat alap építő elemei, általában erre vannak kötve a szenzorok. A szigorúan p2p hálózat csak end device-okból áll
- Range Extender: ez az eszköz ismétlő funkciókat lát el, hogy nagyobb hatótávolságúvá tegye a hálózatot. Mindig bekapcsolt állapotban van, jelenleg 4 range extender lehet egy hálózatban.
- Acces Point: képes tárolni és továbbküldeni az üzeneteket az alvó eszközöknek, menedzsment, autentikációs, biztonsági feladatokat ellátni, továbbá rendelkezik az end device funkcionalitásával.

A SimpliciTI funkciói három rétegbe szerveződnek, amint azt a 2.4 ábra mutatja :

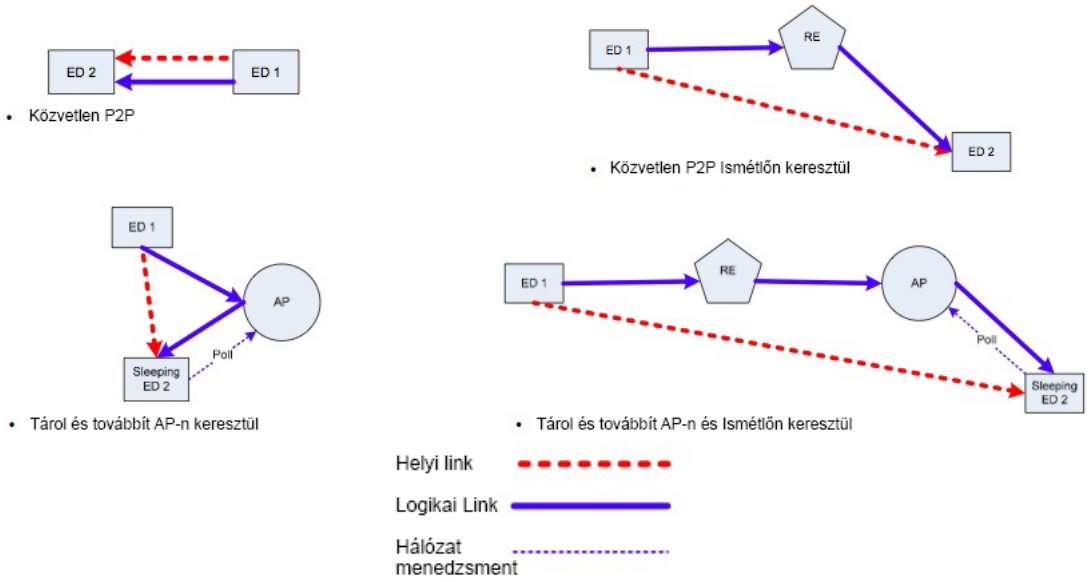


2.4. ábra. A SimpliciTI rétegei

Application: ez a legfelsőbb réteg, a fejlesztőnek csak ezt a réteget kell implementálnia. Ebben hozza létre a saját alkalmazását, felhasználva az API lehetőségeit, amely az alatta lévő rétegeket elfedi. (Megjegyzés: mivel nincs szállítási réteg, a megbízható átvitelt is itt kell implementálni.)

Network: a hálózati réteg feladata az Rx/Tx sorok menedzselése és a keretek célba juttatása. A cél mindig egy alkalmazás, amit egy porthoz rendelünk.

A SimpliciTI a címeket a következőképpen osztja ki: net address = hardwer address (4byte) + application port. Ez egy statikusan kiosztott hardver cím, és nincs címfeloldó mechanizmus. (0x00,0xFF broadcast számára lefoglalva) A támogatott kommunikációs topológiákat a 2.5 ábra mutatja :



2.5. ábra. A SimpliciTI kommunikációs topológiái [18]

2.1.4. Z-Wave

A Z-Wave egy vezetéknélküli kommunikációs protokoll alacsony fogyasztású eszközökhöz. A működési frekvenciája a 2,4 GHz-es és a 900 MHz-es ISM frekvenciasávból kerül ki. A fizikai szintű kommunikációhoz GFSK/BFSK modulációt használ és ezzel 40 kbit/s-os adatsebességet képes elérni. A rendszer source-routed mesh hálózati topológiában működik, azaz a küldő határozza meg a routing útvonalat, mindemellett master-slave kialakítást is használ. A protokoll kialakítása réteges szerkezeten alapul, melyek a következők:

- MAC
- Transfer (Singlecast, Multicast, Broadcast megvalósítás, CheckSum ellenőrzés, ACK)

- Routing Application

A Z-Wave valójában nem egy open-source megoldás, de létezik open-source implementációja, mely az Open-zwave, de ez még jelenleg nincs készen. [19]

2.1.5. Zigbee

A Zigbee egy rádiós adatátviteli szabvány, melyet neves elektronikai fejlesztő és gyártó cégek kezdeményezésével és tagságával létrehozott Zigbee Alliance szervezet alkotott meg. Tartalmát 2004-ben véglegesítették. A cél: kis adatátviteli sebességű és kis hatótávolságú, alacsony energiafelhasználású, kis komplexitású hálózati rendszerben és nem engedélyköteles frekvenciasávban üzemeltethető rádiók megalkotása volt. Olyan vezeték nélküli eszközök létrehozását teszi lehetővé, amelyekre a Bluetooth, a WiFi, a Wi-Max, a Wireless USB vagy egyéb szabványú rádiók nem adnak megfelelő megoldást. Elsődleges felhasználási területei: az ipariszenzor-hálózatok, az épületautomatizálás és a világítástechnika.

A Zigbee adatátviteli szabvány alapján az architektúra három fő rétegből áll, melyet az x. ábra szemléltet:

- a frekvenciasávokat, a csatornakiosztást, a modulációs sémát, az adatsebességet, vagyis a Media Access Layer-t (MAC) és a Physical layer-t (PHY) az IEEE 802.15.4 szabvány határozza meg,
- a Zigbee Alliance specifikálta a logikai hálózatot, a biztonsági és adatvédelmi eljárást és az alkalmazási profilt, melyek a firmware-stack-ben valósulnak meg. Minden mikrokontroller/RF chipkombináció saját Zigbee stacket igényel, melyet leggyakrabban a mikrokontroller vagy az RF-chip gyártója, esetenként a modulgyártók és pár független szoftver cég is kínál.
- Az alkalmazási réteget profilok definiálják, melyeknek két típusa lehet:
  - a Zigbee Alliance által elfogadott publikus profil, mely a különböző gyártók által készített eszközök közötti együttműködést hivatott garantálni. Ezen publikus profillal alkalmazott eszközök kaphatják meg a ZigBee Certified Product tanúsítványt,
  - minden egyéb esetben egyedi profilról van szó, ami zárt rendszerekben minden további nélkül használható, de más Zigbee-eszközökkel való együttműködése nem várható el, és Zigbee tanúsítványt sem kaphatnak.

A Zigbee rendszert a mesh hálózati topológia nyújtotta előnyök kihasználására tervezték. A hálózati elemeknek három típusa van:

- Coordinator: kitüntetett funkciójú, a hálózat létrehozásáért és fenntartásáért felel
- Router: elsődlegesen útvonal-irányítási és átjátszó (repeater) feladatokat old meg, de ezeken túl további feladatokat is elláthat

- End Node: csökkentett funkciójú eszköz, elsősorban rövid üzenetváltási feladatokra alkalmas. útvonal-irányításra nem képes, kizárólag végpont lehet.

A Zigbee rendszer előnyei

- Alacsony költség: A chipset szintű megoldás körülbelül $10-12 , míg a használatra kész, modul szintű megoldások 100 db-os tételre kalkulálva 20-25 $ körül elérhetők. Figyelembe véve a fejlesztési időt és költségeket, valamint a vizsgálati eljárás igényeit, a kész rádiómodulok használata hatékonyabbnak bizonyul, ha a gyártási sorozatok nem érik el a néhányszor tízezer példányos nagyságrendet.
- Hálózatkiterjedés: A routerek kettős, I/O-eszköz és repeater funkciója alapján felépülő mesh-rendszer a rádiók hatótávolságát megsokszorozza, akár egy város méreteit is elérheti.
- Az átviteli akadály leküzdése. Ha két hálózati pont között kommunikációs akadály van (például a rádióhullámokat elnyelő közeg vagy nem kívánt reflexió stb.), akkor a rendszer dinamikus routolás útján keres tiszta kapcsolatot a célállomáshoz. Mindez automatikusan történik.
- Alacsony energiafogyasztás: Az End Node-ok rövid üzenetváltások között alvó üzemmódba kapcsolhatnak, ekkor minimális az áramfelvételük. A rádiós eszközök telepes táplálása nemcsak az adat-, de a tápkábelek kiépítését is szükségtelenné teszi, ami jelentősen csökkenti a kivitelezési költségeket.
- Több forrású termék: A Zigbee-tanúsítvánnyal rendelkező rádiók kompatibilitása lehetőséget ad a felhasználónak, hogy több gyártó rádióját is alkalmazza egy rendszeren belül, vagy beszállítót válthat, megőrizve a termék kompatibilitását.

A rendszer korlátai és gyakorlati tulajdonságai

Az ad-hoc önszervező hálózatban egy 16 bites hálózati cím jelöli az egy rendszerbe tartozó egységeket. Maximum 65 535 eleme lehet egy hálózatnak, és mindegyiknek egyedi, 64 bites azonosítója, címe van. Az adatbiztonságot növeli a 128 bites titkosítás, az adatismétlés és visszaigazolás a protokoll része. A Routerek és a Coordinator egyben végpontok is lehetnek, tehát csatlakozhatnak eszközökhöz, szenzorokhoz. A Routerek és a Coordinator funkciójuk miatt nem állíthatók alacsony fogyasztású sleep módba. Az end node-ok egymás közt közvetlenül kapcsolatot fenntartani nem képesek, ez csak a routeren keresztül valósulhat meg. A hálózati topológia alapvetően mesh (hálós), de kialakulhat pont-pont kapcsolat, star- (csillag) vagy tree- (fa) struktúra. Erre csak közvetett ráhatásunk van, a fizikai elhelyezés és az egységek funkcióinak megszabásával. A 2,4 GHz-es, szabad rádiósávra vonatkozó korlátozások miatt két egység közti távolság (radio HOP) alkatrésztípustól és a fizikai környezettől függően tipikusan 50-150 méter lehet. (A Zigbee Alliance a 868 és a 915 MHz frekvencia körül is kijelöl egy-egy sávot.) 2,405 GHz-től 2,480 GHz-ig 5 MHz-enként

található a 16 csatorna. A rádiólink 250 kbit/s maximális sebességéből a nagy biztonsági szintet megvalósító protokoll miatt a hasznos adatsebesség hozzávetőleg a jelzett érték fele. [20] [21] [22] [23]

2.1.6. Wavenis

Wavenis több, mint egy vezeték nélküli technológia, ez egy elterjedőben lévő vezetéknélküli szabvány az ultra-alacsony fogyasztású, nagy hatótávolságú mesh hálózatokhoz. A technológia machine-to-machine kapcsolatot és alacsony tápellátású autonóm eszközök hálózatba szervezését támogatja. A Wavenis kiválóan alkalmas olyan valós helyzetekben, ahol a rossz terjedési viszonyokat kell kiküszöbölni (pl.: nehezen hozzáférhető helyek, jel csillapítás, alacsony energiájú átvitel stb.). A legfőbb jellemzője, hogy ez nem egy alkalmazás orientált protokoll, ami azt jelenti, hogy különböző magasabb rétegbeli szolgáltatások támogatására is alkalmas. Az alábbi kategóriájú alkalmazásokban használják főként:

- Automatikus mérőrendszerek menedzselése és távoli adat monitorozása
- Háztartási- és épületautomatizálás
- Ipari automatizálás, telemetria és követési alkalmazások
- RFID személyes címkézés és track& trace rendszerek
- Biztonság és riasztások

Működés

A Wavenis a nemzetközileg engedélyezett ISM sávokban a rádiós csatornára vonatkozó szabályokat szem előtt tartva az alábbi szabvány frekvenciákon kommunikál:

- 868 MHz (EU EN300-220) szigorú kitöltési szabályoknak megfelelően
- 915 MHz (US FCC15-247, 15-249)
- 433 MHz (Asia) kiegészítő frekvencián

A Wavenis adatsebessége programozható néhány kbit/s-től 100 kbit/s-ig. A legtöbb Wavenis alkalmazás alacsony adatsebességet használ (tipikusan 19.2-38.4 kbit/s). Az alacsony adatsebesség lehetővé teszi a szűk sávban érzékeny rádióvevők használatát, melyek így nagyobb hatótávolságot és kielégítő működést biztosítanak. A Wavenis maximalizálja az eszközök közötti rádiós linkek költségét, hogy kompenzálja a rossz terjedési tulajdonságokat, a jel elnyelődését és az erősítés hiányát, amely így olcsó megoldásokat eredményez. Többek között ez teszi lehetővé, hogy nehezen elérhető eszközökkel is képes ismétlő eszközök nélkül kommunikálni. A Wavenis hálózatban nincs megkötés az eszközök számára, néhány eszköztől több száz eszközből kiterjedő hálózat is kialakítható. Ugyanakkor az időkritikus alkalmazások megkövetelik a rögzített klaszterméretet (TDMA miatt a broadcast kérésre a visszajelzés adott idősávban történik).

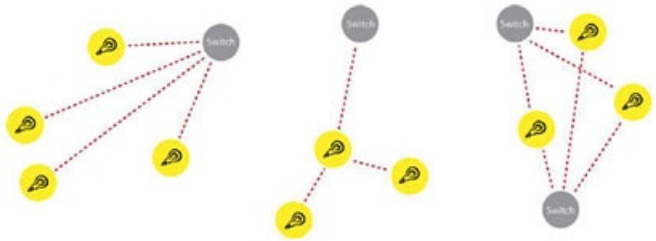
Teljesítmény

A Wavenis a lehető legnagyobb teljesítménnyel garantálja a hosszú eszközélettartamot, melyet a következő intelligens mechanizmusokkal ér el:

- Automatikus frekvenciaszabályozás
- Programozható kimeneti teljesítmény
- Automatikus érzékenység vezérlés
- Adaptív frekvenciaugratás

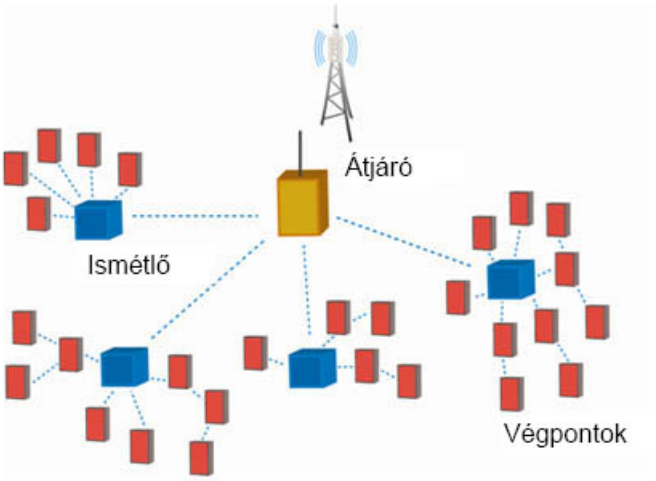
Hálózati architektúra

Hálózati architektúrája mindig az adott megoldásoktól függ, így például egy egyszerű világítás vezérlés (2.6 ábra) vagy pont-pont leolvasó csak két fajta vezetéknélküli eszközt igényel.



2.6. ábra. Pont-pont és pont-multipont kapcsolatok egy világítás szabályozó rendszerben

Egy fejlett mérési rendszer kialakításakor a hálózat egy gateway-t, több repeatert és több ezer végpontot is tartalmaz (2.7 ábra). Az ilyen típusú hálózatokban szinkron-öngyógyító automatikus konfigurációs technikák, valamint hálózati szinkronizáció, smart-payload szolgáltatások és akkumulátor élettartam optimalizálás és karbantartás is helyet kap.



2.7. ábra. Fejlettebb mesh infrastruktúra automatizált távoli megfigyelő rendszerekhez

A technológia a kommunikációhoz GFSK modulációt használ és hatótávolsága 1-4 km-ig terjedhet az adási teljesítménytől függően (1 km - 25 mW; 4 km - 500 mW). [24]

2.1.7. Bluetooth

A Bluetooth egy rövid távú vezeték nélküli kommunikációs szabvány, melyet rendszerint telekommunikációs eszközök használnak az egymás közti, valamint számítógépekkel, PDA-kal és más a szabványnak megfelelő kommunikációra képes eszközzel. Ennek a technológiának a segítségével a felhasználók szabadon cserélhetnek képeket, hanganyagokat, videókat egymás közt, de akár szinkronizálhatják a telefonkönyvüket is a számítógéppel, illetve minden digitális adatkapcsolaton elvégezhető műveletet végrehajthatnak, amire csak az adott készülék képes.

A Bluetooth kapcsolathoz mindössze egy olcsón előállítható adó-vevő chipre van szükség, melyet bármely eszközbe integrálni lehet. Az adóvevő a 2,45 GHz-es frekvenciatartományban működik. Az kommunikációs sávot 79 csatornára osztja fel, mindegyik 1 MHz széles. Minden eszköznek egy 48 bites egyéni azonosítója van, ami lehetővé teszi a két pont valamint a több eszköz közötti kapcsolatot is. Az adatcsatorna ebben a sávban másodpercenként 1600-szor változik véletlenszerűen, azaz szórt spektrumú frekvencia-ugratást használ. Egy hálózatban egy időben 1 „mester” eszközhöz legfeljebb 7 másik eszköz csatlakozhat. Az egymáshoz csatlakozott eszközök ún. personal-area network-öt (PAN), más szóval piconet-et hoznak létre, ami például az egy szobában lévő eszközök által alkotott hálózatot jelenti (vagy az autóban a mobiltelefon és a fejhallgató közötti kicsiny hálózatot).

A Bluetooth alacsony energiafogyasztása miatt különösen alkalmas hordozható eszközök számára. A Bluetooth technológiának az épületeken belüli falak sem jelentenek akadályt.

A kommunikációs távolság az adott eszköz hatóerejének megfelelően változhat. A harmadik kategóriába (Class 3) tartozó eszközöket használják a legtöbb eszközben, melyek 1 mW adóteljesítményűek és maximálisan 10 méteres hatótávolsággal rendelkeznek. A második kategória (Class 2) eszközei 2,5 mW adóteljesítményűek és maximálisan 20 méteres hatótávolságúak. Az első kategóriába tartozó eszközök 100 mW adóteljesítményéhez maximálisan 100 méteres hatótávolság társul. Az elméleti maximális adatsebesség, amit az 1.2-es verziójú adóvevőkkel el lehet érni 1 Mbit/s lehet, de az effektív sávszélesség 721 kbit/s. [25]

2.1.8. 6LoWPAN

A 6LoWPAN az Internet Engineering Task Force (IETF) 6LoWPAN munkacsoportja által fejlesztett szabvány és kezdetektől fogva kis vagy pico hálózatokra tervezték. Nem azért, mert az IP technológia túl költséges lenne, mert itt a költséges jelentése: a kódméret nagysága, a protokoll komplexitása, a szükséges konfigurációs infrastruktúra, és a fejléc vagy protokoll overhead. A 6LoWPAN implementációi könnyen elférnek egy 32K flash memórián (tipikusan kisebb helyet foglal, mint a Zigbee vagy más protokollok). A Zigbee mérete általában 32K és 90K között változik a források szerint.

A munkacsoport sikeresen kizárta a konfigurációs szerverek (DHCP és NAT) szükségességét az IPv6 Zero-Configuration és Neighbour Discovery mechanizmusaival. A protokoll IPv6-ra építésének további eredményeképpen a munkacsoport kidolgozott egy állapotmentes fejléc tömörítő eljárást, melynek segítségével az IPv6 csomagok mérete 4 byte-ra szorítható. Így nincs szükség az IPv6 40 byte-os fejlécére, ugyanakkor a hatalmas címtér megmaradt. (Az IPv6 128 bites címeket használ szemben a 32 bites IPv4 címekkel. Ebből adódóan az IPv6 esetén lehetőség van a Föld minden négyzetméterére akár 700 000 címet is kiosztani.) [26] [27]

2.1.9. DLMS/COSEM

A DLMS a Device Language Message Specification rövidítése és egy alkalmazási rétegbeli szabvány, ami általános fogalmakat határoz meg az objektum-alapú szolgáltatásokhoz, kommunikációs entitásokhoz és protokollokhoz. A COSEM a Companion Specification for Energy Metering rövidítése. Ez magában foglalja a mérés-specifikus objektumokon alapuló OBIS (Object Identification System) kódokat a DLMS-sel használva. Az xDLMS a DLMS egy kiterjesztése és azt írja le hogyan kapcsolódnak egymáshoz az attribútum és COSEM eljárás objektumok. A COSEM számos szabványos interfész osztályt definiál, ezek az úgynevezett objektumok mikor példányosították őket attribútumokat és metódusokat tartalmaznak a szükséges funkciókat leírva. Az attribútumok az adatokat írják le, a metódusok olvassák, írják módosítják az attribútumokat. Négy COSEM interfész csoport létezik: tárolással kapcsolatos, hozzáférés vezérlési, idő és ütemezéssel foglalkozó,

valamint kommunikációs. A szabványosított építőelemek kombinációjával modellezhetünk egy mérőeszközt hierarchikus struktúrában, így komplex mérőrendszereket építhetünk fel. Két kötelező objektum szabályozza eszközönként a hozzáférés és azonosítás vezérlését. Amikor egy mérőműszert leolvasunk, akkor az xDLMS szolgáltatás szerializálja az elkért objektumok paramétereit és APDU-kat (Application Protocol Data Unit) épít belőlük. A COSEM objektumok önleíróak és bővíthetőek az OBIS attribútumok használata miatt.

A DLMS/COSEM a kliens-szerver modellen alapul, ahol az adatgyűjtő rendszer, a kliens kéri az adatokat a szervertől, a mérőeszközökről. A kommunikációs protokoll stack, az úgynevezett profil, teljesen független az alkalmazás rétegtől, így a szerverek és kliensek függetlenül támogathatnak egy vagy több profilt változatos médiákon keresztül. A COSEM modell, amely az alkalmazás réteget írja le, változatlan marad. Az újabb verziók támogatni fogják a push műveletet (klienstől szerverig) és hatékonyabb adatcserét tömörítő eljárások segítségével. Számos más szabvány támogatja a DSLM/COSEM szabványt, mint pl.: M-Bus, IEC 62056-21, Zigbee és a dán DSMR rendszer.

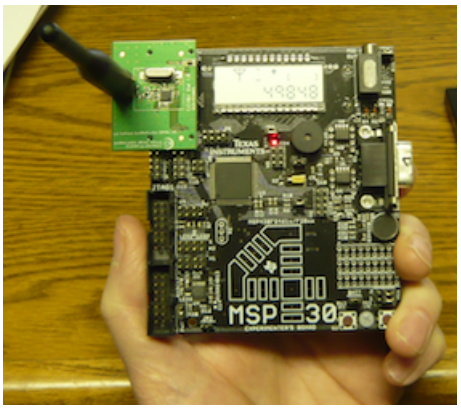
A harmadik fejezet Elemzés pontjában található egy összefoglaló táblázat a vizsgált protokollokról, melyben különböző paramétereket és jellemzőket gyűjtöttünk össze. Ez a táblázat megkönnyíti az egyes protokollok vizsgálatát és kiválasztását az adott alkalmazási igényekhez.

2.2. Eszközök

A megoldásunk megvalósításához valamilyen vezetéknélküli szenzor eszközre is szükségünk volt. Ennek érdekében több alkalmasnak vélt, piacon megtalálható berendezést is megvizsgáltunk. A kutatás során egy számunkra fontos elemekből álló követelményrendszert is felállítottunk, mely a következő:

- minél alacsonyabb energiafogyasztás
- több állapot támogatása
- 433 MHz-es rádiós modul
- könnyű programozhatóság (lehetőleg C támogatás)
- olcsó
- bővíthető, testreszabható

2.2.1. TI MSP430 - CC430F6137



2.8. ábra. Texas Instruments MSP430

Képességek:

- C nyelven programozható: Code Composer Studio / Eclipse Plugin + GRACE GUI
- Ultra alacsony fogyasztású (ULP)
- 16-bites RISC processzor, Neumann architektúra
- számos perifériával bővíthető
- flexibilis oszcillátor rendszer
- instant ébredés
- operációs rendszerek: TinyOS, CMX, FreeRTOS, EmbOS, QP, Salvo

Beépített perifériák:

- Analóg/digitális konverter (ADC)
- konfigurálható időzítők
- Watchdog (szoftverhiba esetén újraindul)
- 128 bites AES kódoló
- DMA vezérlő
- BOR (Brown-out reset)
- Debugger + JTAG csatlakozás
- beépített szorzó
- fogyasztás szabályzók (PMM,SVS)
- RF frontend

- I/O csatornák
- USB, USART, USCI interfészek
- LCD csatlakozás
- Scan Interface, Metering interfészek

A Texas Instruments MSP430-as modulja (2.8 ábra) nagyon népszerű, rengeteg támogatás és felhasználói bázis áll mögötte. A Texas a saját eszközeihez a SimpliciTI protokollt javasolja, de az OpenTag is kompatibilis vele. A SimpliciTI tesztek során azonban nehézségek léptek fel, mint például az eszközök címzésének helytelen működése. [28]

2.2.2. Berkley-MICA

- CPU 8-bit, 4 MHz
- 8 kB utasítás flash
- 512 bytes RAM
- 512 bytes EEPROM
- 916 MHz-es rádiós modul
- 10 kbit/s
- OS TinyOS (3.5 kB)

2.2.3. MicaZ [2]

- 2.4 GHz
- IEEE 802.15.4
- 250 kbit/s
- Router képességek
- Rengeteg kiegészítőt támogat
- Atmel ATmega128L alapú (128kB flash, 4kB RAM)
- Épületen belül 20-30 m
- Épületen kívül 75-100 m
- AES-128
- MoteWorks platform (TinyOS)

2.2.4. Telos

- 250 kbit/s
- 2.4 GHz IEEE 802.15.4 rádiós modul
- 8 MHz Texas Instruments MSP430 mikrovezérlő (10kB RAM, 48kB Flash)
- Integrált ADC, DAC, túlfeszültség védelem és DMA vezérlő
- Lapkára integrált antenna 50 m-es beltéri / 125 m-es kültéri hatótávolsággal
- gyors felébredés sleep módból (< 6 μs)
- hardveres link-layer kódolás és autentikáció
- USB-n keresztül programozható
- TinyOS támogatás : mesh hálózat és kommunikáció implementálás

A vizsgált eszközök közül valamennyi kielégítette valamelyik igényünket, de nem mindet. Emiatt új szenzor eszközt kell terveznünk, melynek részletes leírása a 3. fejezetben található a Saját fejlesztésű hardver eszköz cím alatt.

3. fejezet

Elvégzett munka

Ebben a fejezetben bemutatjuk a saját fejlesztésű hardver eszközünket, a protokollok részletes elemzését, a kidolgozott energiahatékonysági képletet, az elvégzett méréseket és részletesen leírjuk a megtervezett protokollt. Az előző fejezetben ismertetett korábbi megoldásokra építve, azok előnyeit kihasználva oldottuk meg a feladatokat.

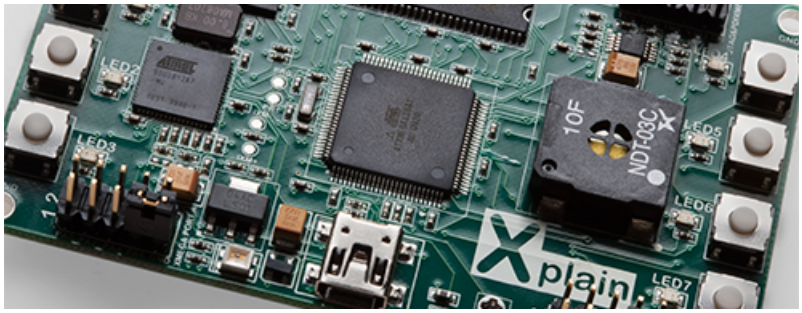
3.1. Saját fejlesztésű hardver eszköz

A vizsgált eszközök közül mindegyik kielégítette valamelyik vagy akár több igényünket is, de valójában egyik sem teljesítette az összeset. Ennek következtében arra jutottunk, hogy új szenzor eszközt kell terveznünk. A tervezés során figyelembe vettük az egyes eszközök előnyös tulajdonságit és ezek alapján próbáltuk megvalósítani a saját megoldásunkat. Végül az Atmel AVR ATMega128 (3.1 ábra) mikrokontrollerének és a Texas Instruments CC1110-es rádiós moduljának egybeintegrálása mellett döntöttünk. Ez az eszközegyüttes teljesítette elvárásainkat.

Atmel AVR ATMega128 tulajdonságai

- Áramfelvétel a különböző állapotokban:
  - alvó: 1,5 uA
  - készenléti: 5 uA
  - aktív: 30 mA
- 1,6 V-on üzemel 32 MHz-en
- Eseménykezelő rendszer és DMA vezérlő
- DES és AES kriptográfiai motor
- C/C++ programozás: AVR Studio
- Beépített perifériák:

- gyors 12-bites ADC/DAC
- konfigurálható időzítők
- Real Time Clock
- Watchdog, BOR (újraindulás szoftverhiba esetén)
- 128 bites AES kódoló
- DMA vezérlő
- programozható interruptok
- On-Chip Debug
- I/O csatornák
- USART, IRCOM, SPI, TWI interfészek



3.1. ábra. Atmel AVR XMEGA

Az Atmel cég AVR moduljai az ismert és elismert mikrokontroller-család részét képezik. [29] A képességei gyakorlatilag teljesen megegyeznek a TI MSP430 képességeivel, de nem annyira gyártófüggő, így rugalmasabban használható.

TI CC1100

- Nagy érzékenység (–111 dBm 1,2 kBaud-on, 868 MHz-en, 1
- Alacsony áramfelvétel (14,4 mA RX módban, 1,2 kBaud-on, 868 MHz-en)
- Programozható kimeneti teljesítmény, akár +10 dBm minden támogatott frekvencián
- Kiváló vevő szelektálás és teljesítményblokkolás
- Programozható adatsebesség 1,2-tól 500 kBaud-ig
- Frekvenciasávok: 300-348 MHz, 400-464 MHz és 800-928 MHz
- 2-FSK, GFSK, és MSK támogatás, valamint OOK és rugalmas ASK kialakítás
- Csomagkezelés támogatása: szinkronszó detektálás, cím ellenőrzés, állítható csomaghossz, automatikus CRC kezelés

- Hatékony SPI interfész: minden regiszter programozható "börszt" átvitellel
- Digitális RSSI kimenet
- Programozható Carrier Sense (CS) indikátor
- Programozható Preamble Quality Indicator (PQI) védelmet nyújt a zajok ellen
- Támogatja az automatikus Clear Channel Assessment-et (CCA) (küldés előtt belehallgat a csatornába)
- Hatótávolság: nyílt terepen: 1km garantált (elméleti max. 5 km); városban: 500m

A megtervezett hardver teljesíti a kitűzött harmadik célunkat, melyben a legalkalmasabb frekvencián működő energia-hatékony mikrokontroller és rádiós modul tervezését tűztük ki.

3.2. Elemzés

Ebben a szakaszban egy egységes kategorizálási rendszer segítségével mutatjuk be az általunk kiválasztott megoldások jellemzőit, ezzel teljesítve az elsőként megfogalmazott célt.

Az összehasonlítási jellemzők a következők: *Frekvenciasáv* [MHz]: A rádióhullámok vízben, levegőben, városi környezetben különböző veszteséggel és visszaverődéssel terjednek a frekvenciájuktól függően. Például a DASH-7 által használt 433 MHz ISM-sáv áthatol a betonon és vízen, valamint ingyenesen használható számos országban viszonylag kis zavarással. Tapasztalatok alapján a 433 MHz-es sáv jobb terjedési tulajdonságokkal és nagyobb hátótávolsággal rendelkezik kisebb adatsebesség árán, mint a 868 MHz-es sáv. A vizsgált megoldások mindegyike valamelyik ISM sávban működik.

Hatótávolság [m] azt jellemző mennyiség, hogy egy adó berendezés milyen messzire képes megbízhatóan adatokat adni és egy vevő berendezés milyen messziről képes megbízhatóan adatokat venni. A DASH-7 által használt 433 MHz-es frekvencia hatótávolsága több mint 2 km szabad téren és kb. 3 emeletnyi beltéren szabványos antennával és tápegységgel az engedélyezett adási teljesítmény mellett (10mW az EU-ban).

Teljesítmény felvétel [mW] azt jellemzi, hogy T [s] idő alatt mennyi energiát használ fel az adott megoldás. Az energiát elemek szolgáltatják és ezeknek évekig működniük kell anélkül, hogy cserére szorulnának.

Átviteli sebesség [kbps]: megadja, hogy a fizikai réteg hány bitnyi információt képes átvinni egy másodperc alatt. Ez azért fontos, mert gyorsabb átvitel esetén kevesebb ideig lesz ébren az eszköz, valamint a rádiós modul energiafogyasztása is ettől függ.

Overhead [byte] megmutatja egy csomag fejléc és CRC byte-jaink számát a PHY rétegben. Ez azért jelentős mennyiség, mert a többlet információ átviteléhez több energia is szükséges.

Payload/csomag arány azt mutatja meg, hogy a milyen arány áll fenn a hasznos teher és a csomag mérete között, ahol a csomag mérete az overheadből és a hasznos teherből tevődik össze.

Skálázhatóság azt a maximális elem (node) vagy hop számot adja meg, amelyet a protokoll még kezelni képes. A megoldásnak működőképesnek kell lennie falvakban, ritkán lakott területeken és városi lakótelepeken is.

Technológiák használata PHY és MAC ISO/OSI rétegekben: például moduláció: (G)FSK, ütközés-elkerülés: CSMA-CA stb.

Biztonság: kriptográfiai támogatás, mint például titkosítás, hibajavító kódolás, CRC stb. Ez egy fontos jellemző, mert ennek használatával megelőzhetőek a visszaélések, a csalások és az információk kiszivárgása. Ezeket a technológiákat autentikáció és autorizáció során alkalmazzák, valamint man-in-the-middle és visszajátszásos támadások esélyének csökkentésére használják.

Ébredési idő (Latency) azt az időtartamot adja meg, amíg az eszköz alvó állapotból aktív (vagy standBy) állapotba kerül, azaz felébred.

Az 3.2 táblázat a vizsgált rendszerek összehasonlítását mutatja a fenti jellemzők alapján. Az utolsó sorban található Λ paraméter a következő szakaszban kerül definiálásra.

Jellemző	DASH7	ZigBee	ONE-NET	SimpliciTI	Wavenis
Nemzetközi szabvány	ISO/IEC 18000-7	IEEE 802.15.4	-	-	-
Frekvencia sáv [MHz]	433.92	868/915 /2400 (Pro)	US: 915 EU: 866.5	480/868/ 915/955/ 2400	868 (EU) 915 (US) 433 (kiegészítés) 2400 (opcionális)
Beltéri hatótávolság [m]	20-50	10-70	60-100	10-60	200
Kültéri hatótávolság [m]	1500	1500	500	1739.5 433 MHz-en 825.1 915 MHz-en 314.5 2.45 GHz-en	> 1km 25mW és LOS; > 4km 500mW és LOS
Átviteli sebesség [kbps]	200	20/40/250	38.4-230.4	up to 250	4.8-100 9.6 433/868 MHz-en 19.2 915 MHz-en
Overhead byte-ok a PHY-ben	20	32	15	14	7
Payload/csomag a PHY-ben [%]	92	76	73	78	97
Ébredési idő [ms]	2500-5000 sleep módban	15 sleep módban	2000 sleep módban tip. 50	eszköz függő	12.8-12800 (tip. 12800)
Skálázhatóság	15 hop	254 node	3 hop	2-~30 node (4 hop)	200 node
Biztonság	AES, CRC	AES, CRC	XTEA, CRC	XTEA, CRC	AES, RSA, 3DES, FEC BCH-val
PHY/MAC rétegben használt technológiák	GFSK, CSMA-CA	DSSS, BPSK, CSMA-CA	2-FSK, CSMA	(G)FSK, OOK/ASK	FHSS, GFSK, SCP (Scheduled Channel Polling) SMAC-hez hasonló
Lambda paraméter [bit/J]	5690	2501	1003	1229	5460

3.2. ábra. A vizsgált protokollok összehasonlítása [30]

3.3. Energia-hatékonysági képlet

Az ebben a szakaszban bemutatásra kerülő számítási módszer a smart metering protokollok energia-hatékonyság szerinti osztályozását teszi lehetővé. A képletekben az eszközök következő állapotait különböztetjük meg:

- alvó mód
- felhasználói adatokat küldő mód
- felhasználói adatokat feldolgozó (rádiós modul kikapcsolva) mód

- menedzsment információt küldő mód
- menedzsment információt feldolgozó (rádiós modul kikapcsolva) mód.

Ezeket a módokat fontos megkülönböztetni, mert az eszközök különböző módokban különböző energiaszinteken működnek. Ebben a megközelítésben csak két módban történik a felhasználói adatok kezelése, a menedzsment adatok overheadnek minősülnek.

Egy érdekes és fontos probléma lenne annak eldöntése, hogy az adatok feldolgozása az eszközökben történjen és ritkábban kerüljenek továbbításra, vagy minden információ minden esetben elküldésre kerüljön. Az alábbi képlet választ ad a kérdésre. A képletek megalkotása során viszonylag könnyen mérhető vagy adatlapokból elérhető adatok használatára törekedtünk. Az (1) képlet a felhasználói adatcsomagok és az összes küldött adatcsomag arányát mutatja meg T időintervallumra nézve. A T egy mért időintervallum, melyet az összes számításunk során felhasználunk.

$$\alpha = \frac{adat\ csomagok\ száma\ T\ idõ\ alatt}{összes\ csomag\ száma\ T\ idõ\ alatt} \quad (1)$$

A (2) kifejezés azt adja meg, hogy mennyi időre volt szükség az adatcsomagok elküldéséhez T idő alatt.

$$t_{adat\ küldés} = \alpha \cdot \frac{|adat\ csomag|}{bitsebesség}\ [\mathrm{s}] \quad (2)$$

, ahol | x | az x változó bitekben mért hosszát jelenti. A következő formulával a menedzsment csomagok küldésére fordított időt tudjuk kiszámítani T alatt.

$$t_{menedzsment\ küldés} = (1 - \alpha) \cdot \frac{|menedzsment\ csomag|}{bitsebesség}\ [\mathrm{s}] \quad (3)$$

A következőkben azt definiáljuk, hogy mennyi időt töltött az eszköz nem alvó (standby és aktív) állapotban.

$$t = t_{adat\ küldés} + t_{adat\ feldolgozás} + t_{menedzsment\ küldés} + t_{menedzsment\ feldolgozás}\ [\mathrm{s}] \quad (4)$$

, ahol $t_{adat\ feldolgozás}$ az adatcsomagok, míg a $t_{menedzsment\ feldolgozás}$ a menedzsment csomagok feldolgozásához szükséges időt jelöli. $E_{alvó}$ azt, mutatja meg, hogy az eszköz mennyi energiát használ fel alvó (sleep) módban.

$$E_{alvó} = (T - t) \cdot P_{alvó}\ [\mathrm{J}] \quad (5)$$

, ahol $P_{alvó}$ az alvó eszköz teljesítménye. (T-t) azaz az időtartam, amennyit az eszköz alvó állapotban töltött T-ből. (6) képlet segítségével a menedzsment csomagok küldésére és feldolgozására felhasznált energiát tudjuk kiszámítani.

$$E_{menedzsment} = t_{menedzsment\ küldés} \cdot P_{küldés} + t_{menedzsment\ feldolgozás} \cdot P_{aktív}\ [\mathrm{J}] \quad (6)$$

, ahol $P_{küldśs}$ az átvitelhez szükséges teljesítményt (aktív rádiós modullal), míg a $P_{aktív}$ az eszköz aktív módban való teljesítményét jelöli. A (7) formula az (5)-höz hasonló annyi különbséggel, hogy itt az adatcsomagok küldéséhez és feldolgozásához szükséges energiát számítjuk ki.

$$E_{adat} = t_{adat\,küldés} \cdot P_{küldés} + t_{adat\,feldolgozás} \cdot P_{aktív}\ [\mathrm{J}]\ (7)$$

, ahol a teljesítmény értékek az előzőekhez hasonlóak, az időértékek pedig adatcsomagokra vonatkozó címkézést kaptak. Az előbbiek kiszámításával meghatározható, hogy T időintervallum alatt mekkora volt a kiválasztott protokoll által elhasznált teljes energia mennyisége

$$E_{összes} = E_{alvó} + E_{menedzsment} + E_{adat}\ [\mathrm{J}]\ (8)$$

Ha ismertnek tekintjük az adatcsomagokban lévő payload bitek számát, akkor kiszámíthatjuk, hogy az általunk választott protokoll hány felhasználói adatbitet képes átvinni egyetlen egységnyi energiával.

A képletek eredményeként egy protokoll- és eszköz-specifikus mérőszámot kapunk, ami alkalmas a különböző megoldások hatékonysági alapon történő összehasonlítására. Az egyes eszközök fogyasztása egyedi a különböző működési módoknak megfelelően, valamint az egyes módokban eltöltött idők is a használt protokolltól függenek, így a formula ötvözi a hardver és a szoftver hatékonyságát. Végeredményként hasznos bit / Joule arányt kapunk, mely könnyedén használható további számítások, mint például akkumulátor élettartam és átvitt adatmennyiség meghatározása során is. Az eredményként kapott mérőszám egy eszközt és egy protokollt magában foglaló teljes rendszert jellemez.

$$\Lambda = \frac{|adat\ payload|}{E_{összes}}\ \left[\frac{bit}{J}\right]\ (9)$$

A számításoknál használt feltételezések:

- Az eszköz az idő 1%-ban ébren, míg 99%-ában alvó állapotban van
- A csomagok 20%-a adat és 80%-a pedig menedzsment információt hordoz, így α= 0,2.
- T egy órának választott időintervallum, amiből t = 0,01·3600 s = 36 s

3.4. Számítási példa

A következőkben egy példán keresztül mutatjuk be részletesen a számítások menetét. A kalkulációkhoz a Zigbee protokoll és a MicaZ hardver adatait használtuk fel. Az adatlapok alapján a teljesítmény értékek a következők:

$$P_{alvó} = 30\ \mu W$$
$$P_{aktív} = 6\ mW$$
$$P_{küldés} = 45\ mW$$
$$P_{küldés} = 45\ mW$$

A Zigbee specifikáció alapján:

$$|\ adat\ csomag\ | = |\ menedzsment\ csomag\ | = 133\ byte = 1064\ bit$$
$$|\ adat\ payload\ | = 101\ byte = 808\ bit$$
$$bitsebesség = 250\ kbps$$

A [(2)-(9)] formulák ugyanebben a sorrendben kiszámolva:

$$t_{adat\ k\tilde{u}ld\acute{e}s} = 0.2 \cdot \frac{1064\ bit}{250\ kbps} = 0.85\ ms$$
$$t_{menedzsment\ k\tilde{u}ld\acute{e}s} = (1 - 0.2) \cdot \frac{1064\ bit}{250\ kbps} = 3.4\ ms$$
$$t_{adat\ feldolgoz\acute{a}s} = 7.2\ s$$

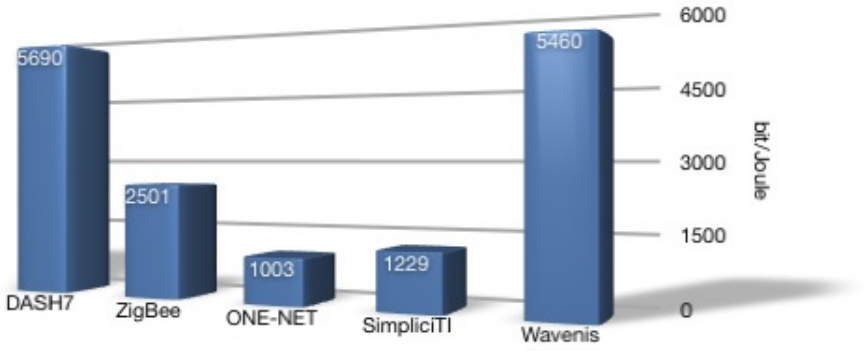
,mert a csomagok t idő 20%-ban hordoznak hasznos információt.

$$t_{menedzsment\ feldolgoz\acute{a}s} = 28.8\ s$$

,mert a csomagok a t idő 80%-ban menedzsment információt hordoznak.

$$E_{alv\acute{o}} = (3600\ s - 36\ s) \cdot 30\ \mu W = 0.107\ J$$
$$E_{menedzsment} = 3.4\ ms \cdot 45\ mW + 28.8\ s \cdot 6\ mW = 0.173\ J$$
$$E_{adat} = 0.85\ ms \cdot 45\ mW + 7.2\ s \cdot 6\ mW = 0.043\ J$$
$$E_{\ddot{o}sszes} = 0.107\ J + 0.173\ J + 0.043\ J = 0.323\ J$$
$$\Lambda = \frac{808\ bit}{0.323\ J} = 2501.55\ \frac{bit}{J}$$

A 3.3 ábra az egyes protokollokra számított Λ értékeket mutatja hasonló feltételek és MicaZ hardver eszköz használata mellett.



3.3. ábra. A vizsgált protokollok összehasonlítása Λ paraméter alapján

Mint ahogy a 3.3 ábrán is látható, az egyes megoldások között lényeges különbségek fedezhetők fel. A grafikonon látható magasabb értékek energiahatékonysági szempontból jobb megoldásokat jelentenek. Megfigyelhető, hogy a DASH7 és a Wavenis protokollok képesek a legtöbb adat átvitelére egyetlen Joule energiával, valamint az is, hogy a Λ paraméter a 3.2 táblázatban is megtalálható payload / csomag aránnyal áll leginkább korrelációban.

3.5. Mérések

Egy másik fontos tényező az energia-hatékonyság mellett - mely az előző fejezetben került bemutatásra - a különböző protokollok frekvenciától függő hatótávolsága, ami

azt jelenti, hogy milyen messze vagyunk képesek hasznos információt tartalmazó biteket hibamentesen eljuttatni.

A 2,4 GHz-es ISM frekvenciasáv zsúfolt, így a zavarás és az interferencia is jelentős mértékű. Sőt mi több a hullámok ezen a frekvencián nem képesek sem a vízen, sem a betonon való áthaladásra, ami hátrányos lehet mérőórák elhelyezkedését illetően. Ezzel szemben a 433 MHz-es és 868 MHz-es sávok megfelelő terjedési tulajdonságokkal rendelkeznek smart metering alkalmazások megvalósítására. Ezeket figyelembe véve méréseket végeztünk 433,92 MHz és 868 MHz frekvenciákon 10 mW-os adóteljesítménnyel különböző környezetekben:

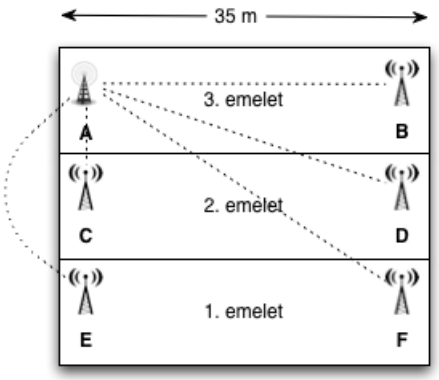
- épületen belül,
- épületből szabad térre
- egy repülőtéren (LOS-nak megfelelő környezetben).

A mérések során a vételi jelszinteket és a távolságokat jegyzetük fel.

Az első mérésünket beltéren végeztük. A mért eredmények a 3.4 ábrán és a 3.1 táblázatban láthatók, melyen megfigyelhető, hogy a szenzor eszközzel a harmadik emeletet elhagyva a kapcsolat megszakadt. Ebből látható, hogy az épület vasbeton szerkezete nagy mértékben képes elnyelni a rádiójeleket különösen 868 MHz-es frekvencián.

Helyzetjelző	433 MHz	866 MHz
A	-14 dBm	-25 dBm
B	-68 dBm	-68 dBm
C	-42 dBm	-61 dBm
D	-86 dBm	-95 dBm
E	-70 dBm	-87 dBm
F	nincs jel	nincs jel

3.1. táblázat. Beltéri mérések

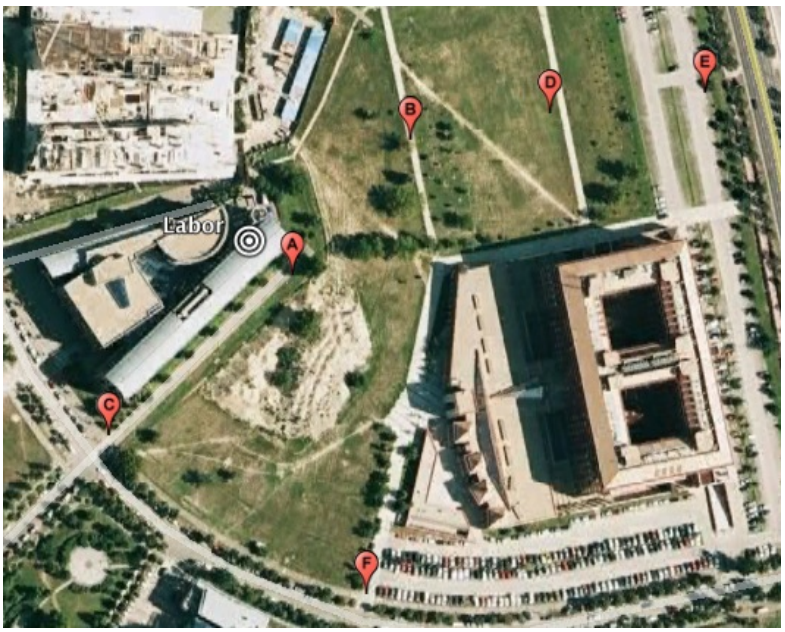


3.4. ábra. Beltéri mérések

A második mérés beltérről kültérre történt. A 3.5 ábra C pontjánál egy törés figyelhető meg a jel erősségében, mivel ekkor az épület túlsó oldalán került sor a mérésre, így át kellett haladnia a jelnek a vasbeton szerkezeten is. A mért értékeket a 3.2 táblázat tartalmazza.

Helyzetjelző	Laboratóriumtól való távolság	433 MHz	866 MHz
A	15 m	-78 dBm	-90 dBm
B	90 m	-80 dBm	-92 dBm
C	100 m	-90 dBm	nincs jel
D	150 m	-87 dBm	-95 dBm
E	210 m	-84 dBm	-94 dBm
F	250 m	-87 dBm	-94 dBm

3.2. táblázat. Beltérről szabadba történő mérések



3.5. ábra. Beltérről szabadba történő mérések

A harmadik mérésen a LOS eset vizsgálatára egy repülőtéren került sor. Az 3.6 ábrán egy térkép látható a helyszínről. A 3.3 táblázat a mért értékek mellett a 433 MHz-es, 868 MHz-es és 2,4 GHz-es LOS számított értékeket is tartalmazza. A számított értékek egy tiszta csatorna ideális jel-zaj viszonyának becslését adják. A mért értékek alacsonyabbak a számítottaknál, mert jelentős volt a mérés során a repülőtéren lévő háttérzaj. A számított értékeket asz alábbi képlettel határoztuk meg:

d - távolság [m]
f - frekvencia [MHz]
P_{tx}- adóteljesítmény [dBm]

G_{tx} - adó antenna nyereség [dBi]
L_{tx} - adó oldali egyéb veszteségek [dB]
L_m - terjedési veszteségek [dB]
G_{rx} - vevő antenna nyereség [dBi]
L_{rx}- vevő oldali egyéb veszteségek [dB]
L_{fs} - Line of Sight veszteség [dB]
P_{rx} - vételi teljesítmény [dB]
$L_{fs} = 20 \cdot log_{10}d + 20 \cdot log_{10}f - 27.55$
$P_{rx} = P_{tx} + G_{tx} - L_{tx} - Lfs - Lm + G_{rx} - L_{rx}$

Helyzetjelző	Távolság a bázistól	433 MHz mért	433 MHz számított	868 MHz számított	2.4 GHz számított
A	173 m	-70 dBm	-58.65 dBm	-64.68 dBm	-73.51 dBm
B	335 m	-79 dBm	-64.39 dBm	-70.42 dBm	-79.25 dBm
C	564 m	-85 dBm	-68.92 dBm	-74.95 dBm	-83.78 dBm
D	774 m	-95 dBm	-71.67 dBm	-77.69 dBm	-86.53 dBm
E	1045 m	-100 dBm	-74.28 dBm	-80.3 dBm	-89.14 dBm

3.3. táblázat. Szabadtéri mérések

3.6. ábra. Szabadtéri mérések

A negyedik méréssorozatban a 433 MHz-es rádiós jelek viselkedését vizsgáltuk vidéki környezetben. Ezek a mérések a tényleges környezetben kerültek elvégzésre, mert az eszközöket a vízóraaknába, valamint a villanyórába helyeztük el. A mérési pontok a 3.7 ábra láthatók. Mindkét mérési esetben a mért értékeket a 3.4 táblázat foglalja össze. A vízóraakna jelentős mértékben árnyékolja a rádiós jelet, így távolabbi pontokban már nem mértünk jelet. A jel erőssége gyorsabban csökkent, mint a második mérés esetén, ahol beltérből kültérre történő sugárzás teljesítményét vizsgáltuk.

Helyzetjező	Mérőórától való távolság	Vízóraakna esetén	Villanyóra esetén
A	192 m	-101 dBm	-85 dBm
B	149 m	-99 dBm	-83 dBm
C	77 m	-95 dBm	-73 dBm
D	19 m	-75 dBm	-54 dBm
E	79 m	-93 dBm	-78 dBm
F	107 m	-100 dBm	-91 dBm
G	195 m	nincs jel	-94 dBm
H	266 m	nincs jel	-98 dBm
I	305 m	nincs jel	-97 dBm
J	354 m	nincs jel	nincs jel
K	336 m	nincs jel	-100 dBm
L	289 m	nincs jel	-96 dBm
M	316 m	nincs jel	-96 dBm

3.4. táblázat. Vidéki környezetben történő mérések

3.7. ábra. Vidéki környezetben történő mérések

A mérések bizonyítják, hogy a 433 MHz-es jel jobb terjedési tulajdonságokkal rendelkezik, mint a 868 MHz-es, így alkalmasabb smart metering alkalmazások megvalósítására. A mérések elvégzésével teljesítettük a 2. pontban kitűzött céljainkat.

3.6. Protokoll

3.6.1. Koncepció

A rendszer megtervezése során figyelembe vettük a korábban megfogalmazott célokat és a lehetséges felhasználási területek igényeit. A protokollal szemben elvárás volt, hogy legyen energia-hatékony. Ezt úgy értük el, hogy az eszközök az idő legnagyobb részében alvó állapotban vannak, és amikor felébrednek, akkor nagyon rövid idő alatt elküldik az üzeneteiket, majd újra energiatakarékos üzemmódba kerülnek. Szintén ebbe a problémakörbe tartozik, hogy ne küldjenek sok üzenetet az eszközök egymás között, csak a minimális információ átvitelére van szükség, ezzel szintén energiát takarítunk meg. Egy másik követelmény, hogy a protokoll legyen skálázható. Alkalmasnak kell lennie nagy távolságok áthidalására, amikor a csomópontok ritkán vannak elhelyezve (pl. falvakban) és zsúfolt (pl. panelház) környezetben is. Szintén fontos szempont, hogy a protokoll legyen biztonságos. A kölcsönös hitelesítéssel meggyőződhetünk arról, hogy a kommunikáló felek valójában azok, akiknek mondják magukat. A visszajátszás és közbeékelődéses támadások elleni védelemről pedig a titkosítási algoritmusba épített időbélyeg és sorszám gondoskodik.

3.6.2. Felmerülő problémák és megoldásaik

A protokoll tervezése során számos probléma merült fel, melyekre megoldást kellett találnunk. Először is a rádiós csatorna kihívásaival találtuk magunkat szemben. Itt az egyes felhasználók csomagjainak ütközése (egy időben több csomag beérkezése ugyanahhoz a node-hoz) és a rejtett terminál problémaként ismert jelenség jelentettek problémát. A csomagok ütközését időosztásos kommunikációval és ALOHA-szerű eljárással oldottuk meg, tehát minden egyes felhasználó egy kiosztott időszeletben/résben kommunikálhat partnerével. Az egyes elküldött csomagok helyes vételét nyugtázással (Ack) jelezzük a küldő fél felé, aki ha nem kap ilyen megerősítést egy meghatározott időn belül akkor újraküldi ezt a csomagot.

A következő probléma mellyel szembe kellett néznünk az energia-hatékonyság kérdése. Arra a következtetésre jutottunk, hogy az eszközöknek az idő nagy részében alvó állapotban kell lenniük. Ennek elérése érdekében az egy node-hoz érkező csomagokat aggregált módon továbbítjuk, valamint mindig csak azok a node-ok kerülnek aktív állapotban, melyek éppen egymással kommunikálnak. Az előbb említett megoldások mellett a protokollban kialakított címzési eljárás is segíti az eszközök energia-szükségletének minimumon tartását.

A tervezés során kérdésként merült fel az is, hogy a protokoll működésének mennyire kell időkritikusnak lennie. Arra a következtetésre jutottunk, hogy a hálózatfelderítési fázisnak minél gyorsabban és hatékonyabban kell lefutnia. Ezzel ellentét-

ben a mért adatok küldése nem annyira időkritikus. Ebben az esetben inkább az a lényeges, hogy a mérési eredmények a jövőben valamikor beérkezzenek az azokat tároló adatbázisba. Ebből következően a mért adatok csomagjainak aggregálására is lehetőségünk nyílt.

Az eszközök címzése és a node-ok közötti routing eljárások képezték a tervezés következő problémáját. A címzést úgy kellett megvalósítani, hogy a hardver eszközök is képesek legyenek a megoldás kezelésére, valamint figyelembe kellett venni az energiahatékonysági követelményünket is. Mindemellett olyan hálózat-feltérképezési és routing eljárás kellett kidolgozni, mely viszonylag csekély üzenetváltással működik, az alacsony fogyasztás érdekében. A címzési megoldásunk segítségével hierarchikus hálózati topológia kialakítását is lehetővé tettük, mellyel kisebb egységekbe, klaszterekbe, szervezhetjük node-jainkat. Minden node rendelkezik egy saját azonosítóval, valamint a klaszterek is külön ID-val kerültek elválasztásra. Az útvonalválasztási eljárásunk megoldásaként egy táblázat alapú megoldást választottunk, mely broadcast üzenet küldések segítségével működik.

A kommunikációs csatornákon történő biztonságos átvitel is szerepet kapott a tervezés során. A protokollnak védelmet kell nyújtani különböző támadások ellen, mint például üzenetek visszajátszása, man-in-the-middle (MITM) stb. Ezen problémák megoldására kriptográfiai primitíveket használtunk, mint MAC (Message Authentication Code) digitális aláírás formájában a feladó node-ok azonosítására, AES titkosítás a csomagok rejtjelezéséhez stb. Ezek használata során figyelni kellett a hardveres erőforrások által nyújtotta lehetőségekre is.

A hasznos információk és menedzsment üzenetek küldéséhez nem állt rendelkezésünkre egy meghatározott formátum, így ezekhez különböző csomagformátumokat dolgoztunk ki. Ezek a formátumok egy alsóbb szintű hordozó és egy felsőbb szintű tartalom specifikus részből állnak. Ezzel jól szétválaszthatók lesznek az üzeneteink típusuktól függően.

3.6.3. Szükséges funkciók

A szenzorhálózat működéséhez, az adatok továbbításához, a rendszer konfigurálásához az alábbi funkciók szükségesek:

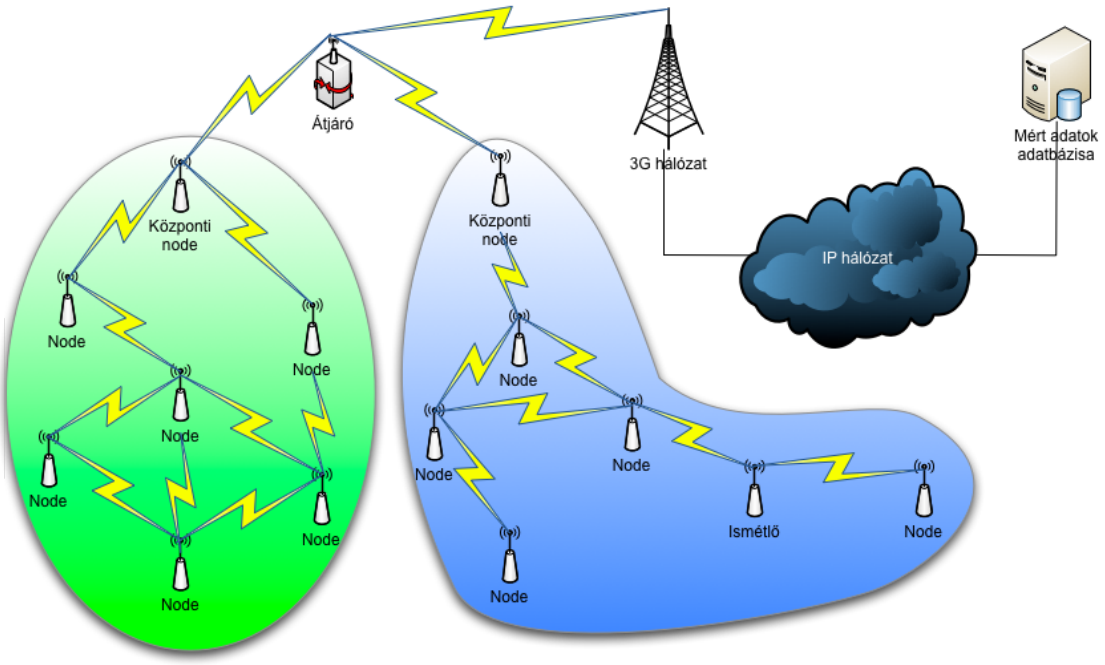
- Adatküldés
  - egyszeri lekérdezés és adatküldés
  - automatikus küldés időközönként vagy mért mennyiség alapján (pl.: óránként vagy 1 kW-os fogyasztásonként)
  - aggregált küldés (pl.: napi/heti/havi összegzés)
- Konfigurálás
  - hálózat, elérhető nodeok feltérképezése
  - sikertelen csomagküldés esetén ismétlés, ezek sikertelensége után új útvonal keresés
  - route táblák folyamatos frissítése a bejövő csomagok alapján

3.6.4. Architekturális elemek

A fenti funkciókat az alábbi architekturális elemekkel lehet robusztusan és skálázhatóan megvalósítani:

- Átjáró:
  - továbbítja a vett adatokat valamilyen hálózati kapcsolaton keresztül egy szerverre, tárolás és feldolgozás céljából
  - csak a központi node-ok kommunikálnak vele, az egyes klaszterekben lévő node-ok nem.
- Központi node:
  - a folyamat fő vezérlője menedzsment szinten
  - kisebb hálózat esetében, ahol nem kell több klaszter, a központi node tölti be a átjáró szerepét is
  - ha szükséges részt vesz a csomagok továbbításában is.
- Node:
  - csomag vétele esetén, ha szükséges továbbítja azt a cél node felé
  - gyűjti a mérési adatokat a rá csatolt szenzor(ok)ból
  - periodikusan felébred és elküldi a mért adatokat a központi node-nak
- Ismétlő
  - abban az esetben, ha a node-ok túl távol helyezkednek el egymástól, közöttük áthidaló, üzenet mediátor feladatot végzi
  - nincs hozzácsatolt szenzor

A teljes hálózat különálló alhálózatokból, klaszterekből épül fel. Mindegyik klaszter rendelkezik egy dedikált központi node-dal, amely ismeri és vezérli a klaszterbe tartozó összes node-ot. A 3.8 ábrán az architektúra elemei láthatók, ahol az egyes klaszterek különböző színű háttérrel vannak jelölve.



3.8. ábra. Tervezett topológia

3.6.5. Protokoll leírás

A következő szakaszban a tervezett protokoll sajátosságaira térünk ki. Bemutatjuk az egyes csomagok lehetséges típusait, a címzés, a hálózat-feltérképezés és a routing eljárás, valamint a biztonsági megoldás apróbb részleteit. A bemutatást a csomagtípusokkal és azok felépítésével kezdjük, hogy az azt követő részekben megfelelő módon tudjunk azok használatára hivatkozni.

Csomagfelépítés és típusok

A csomagok két részből állnak, melyek a hasznos tartalom (payload) és a fejléc mezők (header fields). A csomagok fejlécében lévő mezők elhelyezkedését a 3.9 ábra mutatja.

One-hop destination address	Cluster ID	One-hop source address	Real destination address	Real source address	Sequence number	Timestamp	Hop count	Type	Payload	MAC

3.9. ábra. Csomagfelépítés

A fejléc mezők a következők:

- One-hop destination address: több ugrásos átviteli út esetén a következő csomópont címe.

- Cluster ID: a klaszter azonosítója
- One-hop source address: a csomag aktuális küldőjének címe több ugrásos átvitel esetén
- Real destination address: a valós cél címe
- Real source address: a valós feladó címe
- Sequence number: a csomag sorszáma
- Timestamp: a csomag elküldésének időbélyege
- Hop count: jelzi hány node-on ment át az üzenet
- Type: a csomag tartalmának típusa
- MAC: csomaghitelesítő kód

A Type mező az alábbi csomagtípusokat definiálja:

- ack - pozitív nyugta
- nack - negatív nyugta
- discover - egy hop-os felderítés
- searchNodes - szomszédos node-ok keresését kezdeményezi
- foundNodes - talált szomszédos node-ok listája
- discoverRoute - alternatív útvonal keresése adott node-hoz
- ping - egy node elérhetőségének tesztelésére
- pong - válasz a pingre
- data - magasabb réteg adataihoz

Ha a csomag típusa data, akkor a payload a következőket tartalmazza:

- a csomagban szállított mért értékek darabszáma
- szenzor azonosító
- mért érték(ek) időbélyege
- mért érték(ek) byte-ban mért hossza
- mért érték(ek)

Címzés

A hálózatban lévő node-okat klaszterekbe csoportosíthatjuk. Egy klaszter kommunikációját és működését egy központi node koordinálja, melyből minden klaszterben kizárólag egy van. Egy klaszterben max. 253 node foglalhat helyet a központi node mellett. A központi node-ok címe, valamint a broadcast cím klaszterenként előre meghatározott. A node-ok címe egy node azonosítóból és egy klaszter azonosítóból

tevődik össze. Az implementálásnál használt rádiós IC hardveres megoldással képes minden csomag első byte-jára figyelni. Ebben a byte-ban a node-ok címét tároljuk, melynek energia-hatékonysági okai vannak. Abban az esetben, ha egy beérkező csomag nem az adott node-nak szól, akkor továbbra is alvó állapotban marad. Ezzel a módszerrel garantáljuk a 4(a) cél teljesülését. Minden node-ban egy beérkező csomag esetén a one-hop cél címre címegyezést vizsgálunk. Ha ez a cím egyezik a saját címmel, akkor megvizsgáljuk a valós cél címet is, ha nem, akkor nem foglalkozunk tovább a csomaggal. Abban az esetben, ha a valós cél cím is megegyezik a node címével, akkor a csomag az őt vevő node-nak szól, más esetben a valós cél címre kell továbbítani a csomagot.

Csomagok garantált átvitele

A node-ok közötti rádiós csatornán menő csomagok garantált átvitelére sorszámmal jelölt nyugtázást alkalmazunk. Az nyugtázandó csomagok a nyugtájuk megérkezéséig az adott node ackBuffer nevű tárolójában várakoznak. Ha bizonyos időn belül nem jött nyugta az elküldött csomagra, akkor periodikusan újra megismételjük a küldést. Ha az ismétlések után is sikertelen volt a küldés, akkor az egy ugrásnyira lévő node-ot elérhetetlennek vesszük, töröljük a route táblából, majd újra megpróbáljuk a küldést ezúttal egy új útvonalon. Ha már nincs több lehetőség új útvonal kialakítására, akkor a node egy discoverRoute broadcast csomagot küld ki, amivel megpróbálja kideríteni, hogy az adott cél node felé tudja-e valamelyik közvetlenül elérhető node továbbítani a csomagot. Ha nem érkezett válasz egyik node-tól sem, akkor törli az ackBuffer-ből a csomagot. Ellenkező esetben továbbítja valamelyik válaszoló node-nak. Vannak olyan csomagok, amiket nem kell nyugtázni, vagy más formában történik a nyugtázásuk. Ez igaz minden broadcast típusú csomagra. Például discover üzenet küldése esetén is vár a node ack-ot, de ez a nyugta nem azt fogja jelezni, hogy a vétel sikeresen megtörtént, hanem, hogy a node jelen van és elérhető. A másik probléma a csomagok garantált átvitele során az ütközések detektálása és elkerülése. Erre a protokoll egy Aloha-szerű megoldást használ. A node-ok (az átjárót is beleértve) ha adni akarnak, véletlenszerű ideig várakoznak, majd belehallgatnak a csatornába. Ha azt szabadnak érzékelik adásra kapcsolnak. Egy beérkező broadcast üzenetre minden node a saját időrésében válaszol, melyet saját címéből számít. Ennek segítségével elkerülhetőek az ütközések.

Biztonság

Minden klaszter minden node-jához eljuttatunk egy előre kiosztott mester kulcsot, ez lesz a klaszter közös titkos kulcsa, melyet kezdetben csak a központi node ismer. A közös kulcsból minden node előállít egy rá nézve egyedi kulcsot, ez lesz a node saját kulcsa. Ezt a saját kulcsot fogja használni a node a csomagjainak titkosításhoz. Minden elküldött csomagot el kell látni egy MAC értékkel, amit a fogadó fél leellenőriz, és abban az esetben, ha nem érvényes, akkor eldobja a csomagot. A visszajátszásos támadások elkerülése végett szükség van minden csomagban egy időbélyeg elhelyezésére is. Ha ez az időbélyeg meghatározott idejű differenciát mutat a

fogadó fél órájához képest, akkor eldobjuk az üzenetet. Az alkalmazási rétegben a mikrovezérlő hardveres AES titkosítását használjuk a felső szintű teljes adathalmaz titkosítására. A biztonsági mechanizmusok alkalmazásával teljesül a 4(d) célunk.

Routing

Minden node-nak van egy route táblája, amiben eltárolja, hogy egy adott node-ot milyen más node-okon keresztül és hány ugrás árán ér el. Egy csomag küldésekor a node végignézi a route tábláját, és a cél node-hoz tartozó legkisebb költséggel járó sorban lévő next-hop node-nak továbbítja a csomagot. Ha nincs bejegyzés a cél node felé, akkor a Csomagok garantált átvitele résznél taglalt discoverRoute csomag segítségével fog a node útvonalat keresni a valós cél felé. Az útvonalválasztás során one-hop routingot használunk, tehát egy csomag kiküldésekor a küldő nem határozza meg, hogy milyen útvonalon keresztül fog utazni a csomag a hálózatban, csupán a helyi kiértékelésnek megfelelően a legkisebb költséggel elérhető node-nak továbbítja azt. Minden csomag fejléce tartalmaz egy Hop count nevű mezőt, amit minden egyes továbbításkor minden node eggyel növel, majd ha ez az érték egy bizonyos felső korlátot átlépett, eldobja a csomagot. Ezzel elkerülhetjük a csomagok végtelen keringését a hálózatban. Amikor egy node vesz egy csomagot (akár nem neki szólót is), különféle következtetéseket von le a csomagban található információk alapján a hálózat felépítésére vonatkozóan. Ezen információk (pl.: one-hop forrás, valós forrás cím, foundNodes csomagban lévő node-ok stb.) alapján fogja frissíteni az útvonal tábláját.

Hálózat-feltérképezés

A hálózat feltérképezés tervezése során a következő feltevéseket és tervezési megfontolásokat vettük figyelembe:

- A központi node ismeri az egyes klaszterekben lévő node-ok számát. Ha új node kerül a klaszterbe, vagy onnan node-ot távolítunk el, akkor a központi egységet kell átkonfigurálni a megváltozott node-számra.
- Az egyes node-ok ismerik a központi egység címét.
- Csak a szomszédos node-ok akarnak egymással kommunikálni, a mért adatok a mérő node-tól a központi egységig utaznak, nincs szükség teljes csomagkapcsolt hálózat kialakítására (a lehetőség megvan, hogy akármelyik node tudjon akármelyik másikkal üzenetet cserélni, legrosszabb esetben a központi node-on keresztül).
- Törekedünk arra, hogy a discover fázis során egy klaszterben egyszerre csak egy node kezdeményezze üzenet küldését az ütközések elkerülése végett. Emiatt próbáljuk úgy alakítani a discover fázisunkat, hogy egyszerre mindig csak egy folyamatvezérlő node legyen, a többi addig várakozzon.
- Kicsi a csomagméret, ezért ehhez kell adaptálni a protokollunkat.

- A klaszterek különálló hálózatoknak tekinthetők, egy közös ponttal - a központi node-dal.
- A valós körülményekhez hasonlóan egy átlagos smart metering hálózatot úgy kell elképzelni, mint egy fa szerkezetű gráfot (amiben egy szinten belül is vannak kapcsolatok), csúcsában a központi node-dal. A fának nincs sok szintje, ezért a max. Hop count-ot alacsonyra kell állítani.
- A protokoll segítségével a node-ok adaptálódnak a link- és node-kiesésekhez, többször megpróbálnak elküldeni egy csomagot sikertelenség esetén, majd korrigálják a route táblájukat a szomszédos node-okat is megkérdezve a sikeres csomagtovábbítás érdekében.

Minden node, mely a broadcast discover csomagot vette, válaszol arra. Egy discover csomagot vevő node tudja, hogy e csomag küldője közvetlenül elérhető számára. Ennek hatására a vevő node frissíti a routing-tábláját és ack csomagot küld a feladónak. Az ack vételekor a discovert küldő node route táblája is frissül, így deríti fel a közvetlenül elérhető node-okat.

A feltérképezési folyamatot a központi node vezérli. Ő kezdi az első discovert, utána pedig egyenként szólítja fel az eddig megismert node-okat a searchNodes üzenet segítségével, hogy végezzék el ők is a discovert, majd küldjék vissza neki az általuk látható node-ok listáját egy foundNodes csomagban. A folyamat addig tart, amíg minden egyes, a központi node számára ismert node nem hajtott végre discovert. A folyamat végén a minden node tudni fogja, hogy merre kell továbbítani egy csomagot ahhoz, hogy az a helyi kiértékelésnek megfelelően a legrövidebb úton célba érjen.

A protokoll nem garantálja a csomagok globálisan legrövidebb úton történő utazását, hiszen egyik node sem látja a teljes hálózat felépítését, inkább a megváltozott körülményekhez való adaptálódáson, a garantált átvitelen és a minimális üzenetváltáson van a fő hangsúly.

A központi node a hálózat karbantartásának érdekében periodikusan pingelheti az egyes nodeokat. Ha valamelyiket többszöri ping után sem éri el, kezdeményezhet egy új discover fázist. Egy másik lehetséges mód egy node kiesésének detektálására, hogy ha bizonyos idő eltelte után nem vett a központi node csomagot az adott nodetól, kiesettnek tekinti és végrehajtja a szükséges lépéseket (feltételezve, hogy minden node periodikusan küldi a mért adatokat a központi nodenak, vagy más formában kapcsolatba lép vele).

A felderítés elindítása a ping üzenetektől függetlenül is kezdeményezhető, például naponta vagy pár óránként, a művelet költségének függvényében.

4. fejezet

Teszt és verifikáció

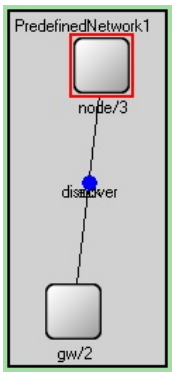
Ebben a fejezetben az előző fejezetben ismertetett és kidolgozott protokollt és mikrokontrollert teszteltük szimulációkkal és valós tesztekkel.

4.1. OMNet++

Az OMNet++ egy nyílt forráskódú diszkrét esemény szimulációs platform. Segítségével bármilyen hálózatot (legyen az akár vezetékes, vagy vezetéknélküli, on-chip, szenzor, ad-hoc vagy IP hálózat) létrehozhatunk és szimulációkat futtathatunk rajtuk. A program Eclipse alapú felhasználó felülettel rendelkezik, így a projektek kezelése és a programozás is kényelmes. A szimulálni kívánt modulokat C++ nyelven kell implementálni, majd egy hálózatleíró nyelvvel (NED) meg kell adni a modulok közötti kapcsolatokat. Az OMNet++ felhasználói felületet biztosít a szimulációk futtatására is, lehetővé téve az idő és az üzenet alapú léptetést, animálja a hálózaton folyó üzenetváltásokat és naplót készít. Nagy előnye, hogy multiplatform és népszerűsége miatt rengeteg kiegészítő készült hozzá.

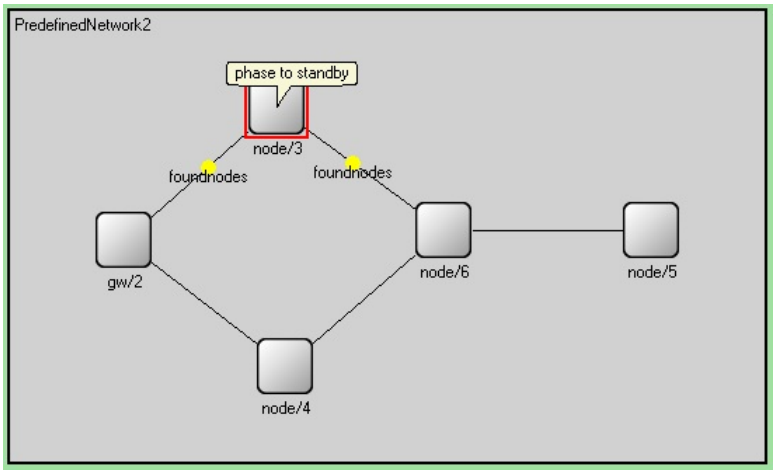
4.2. Teszt-topológiák bemutatása

A következő tesztek célja a protokoll verifikáció, azaz hogy a tervezett protokoll működése az elvárt eredménnyel megegyező és sikeresen megoldja a felmerülő problémákat. Ellenőrizzük a protokoll legfontosabb építőelemeit egyre komplexebb feladatok elé állítva a node-okat és a központi node-ot. Az első tesztkonfiguráció (4.1 ábra) a legegyszerűbb, egy központi node-ból és egy node-ból álló hálózat. Ezen a legkisebb modellen próbáltuk ki az alapvető funkciókat. Egy adat csomag küldését és a rá érkező ack üzenet vételét teszteli. Amennyiben nem érkezik ack (a véletlen csomagvesztés miatt ez adott valószínűséggel előfordulhat) a csomagot újra kell küldeni. A szimuláció eredményei szerint a protokoll ezt helyesen kezeli.



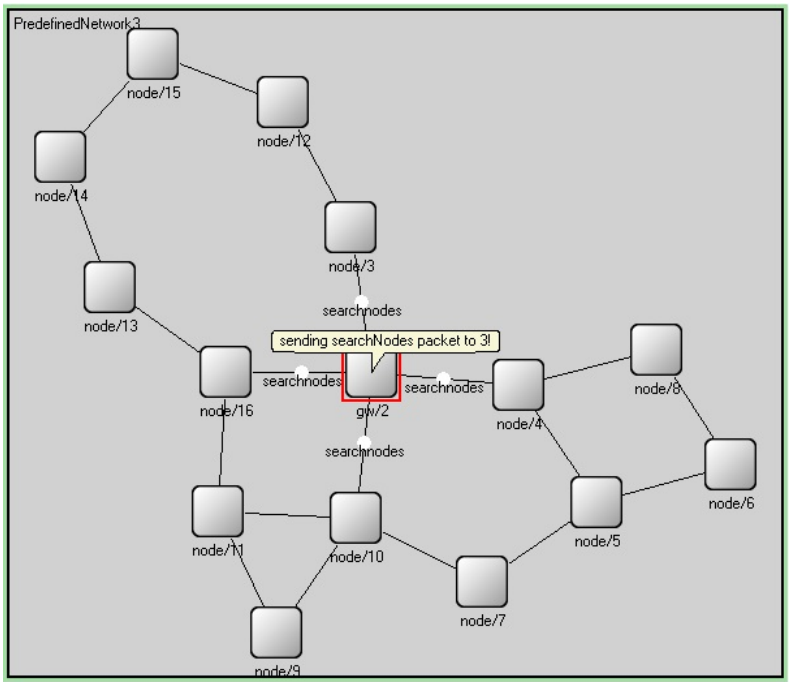
4.1. ábra. Első tesztkonfiguráció

A második teszthálózaton (4.2 ábra), amely egy központi node-ból és négy node-ból áll, már bonyolultabb elemeit tesztelhetjük a protokollnak: a felderítés (discovery) folyamatát és a routing tábla alakulását. A felderítés során minden node bekerül a központi node routing táblájába és a szomszédsági viszonyok is tisztázódnak, így a teszt sikeres.



4.2. ábra. Második tesztkonfiguráció

A harmadik szimulált hálózaton (4.3 ábra) az egyik node kiesésével a meghibásodás során fellépő helyreállító folyamatok helyességét ellenőrizzük. Miután lefutott a discover fázis, a gateway a legtávolabbi node felé egy ping üzenetet küld, azonban a hozzá vezető úton az egyik link meghibásodik. Miutána a protokoll ezt észlelte, helyesen alternatív útvonalat keres és az üzenet végül célba ér.

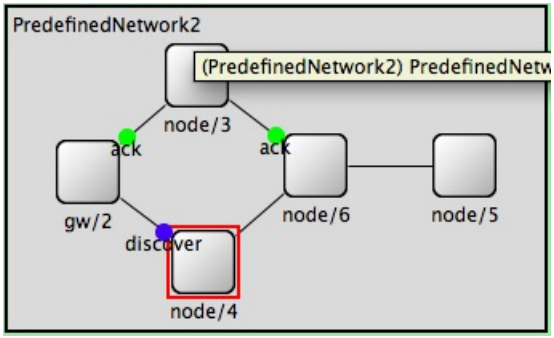


4.3. ábra. Harmadik tesztkonfiguráció

A negyedik teszthálózat pedig 200 node együttes szimulációjával a protokoll skálázhatóságát vizsgálja a node-ok között véletlenszerűen kiosztott összeköttetésekkel. A protokoll megnövekedett üzenetszámmal, de képes ekkora méretű hálózat kezelésére is.

4.3. Discover fázis szimulációs vizsgálata

Ebben a szakaszban a protokoll hálózat-felderítési fázisának szimulációs vizsgálatát mutatjuk be részletesen számos illusztrációs ábrával kiegészítve. A felderítési fázist a központi node kezdeményezi egy broadcast discover csomag küldésével, melyet az 4.4 ábra szemléltet.



4.4. ábra. A központi node discover üzenetének broadcast küldése, majd a node-ok nyugtát küldenek

```
gw/2: ------ changing phase to gateway/discover phase
gw/2: discover SEND - OHDst: 255 OHSrc: 2 RealDst: 255 RealSrc: 2
```

A fenti logban megfigyelhető, hogy a discover csomag one-hop és valós címe is a broadcast címnek megfelelő 255-ös értéket vette fel.

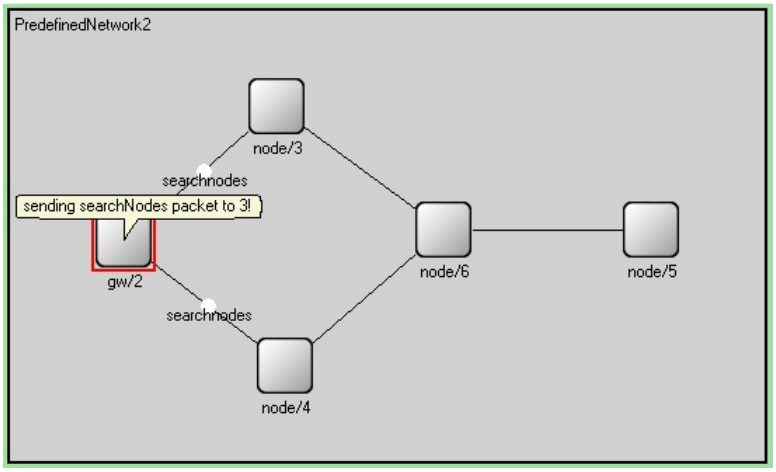
Minden node, mely vette ezt a csomagot egy ack nyugta üzenettel válaszol a feladónak, valamint a route táblájába bejegyzést készít. Az alábbi szimulációs eredmények a 3-as node discover üzenet megérkezésére történő route tábla frissítését szemléltetik.

```
node/3: discover ARRIVED - OHDst: 255 OHSrc: 2 RealDst: 255 RealSrc: 2
node/3: adding to routing table: (to: 2 via: 2 hops: 0)
node/3: routing table:
  - to: 3 via: 3 hops: 0
  - to: 2 via: 2 hops: 0
node/3: ack SEND - OHDst: 2 OHSrc: 3 RealDst: 2 RealSrc: 3
```

A központi node a beérkező ack-ok hatására frissíti route tábláját. Ezzel az egy ugrásra lévő csomópontok feltérképezésre kerültek a központi node szempontjából, melyet az alábbi naplóban is megfigyelhetünk.

```
gw/2: ack ARRIVED - OHDst: 2 OHSrc: 4 RealDst: 2 RealSrc: 4
gw/2: adding to routing table: (to: 4 via: 4 hops: 0)
gw/2: routing table:
  - to: 2 via: 2 hops: 0
  - to: 3 via: 3 hops: 0
  - to: 4 via: 4 hops: 0
```

Ezután a vezérlő node egyenként felszólítja az eddig közvetlen megismert csomópontokat, hogy derítsék fel a saját egy hop-os környezetüket. Ezt a searchNodes csomag küldésével érik el (4.5 ábra).

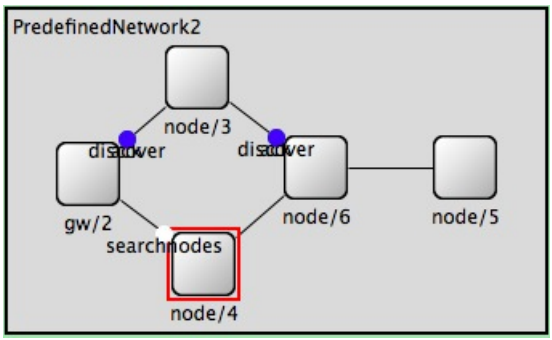


4.5. ábra. searchNodes csomagok küldése

Az ehhez tartozó naplóbejegyzések:

```
gw/2: ------ changing phase to gateway/sendingSearchNodes
gw/2: sendSearchNodesTo queue: 3 4 searchNodesHaveSentTo:
gw/2: searchnodes SEND - OHDst: 3 OHSrc: 2 RealDst: 3 RealSrc: 2
gw/2: ackBuffer:
  - searchnodes resendCount: 0 OHDst: 3 OHSrc: 2 RealDst: 3 RealSrc: 2
```

A felszólított node-ok szintén egy broadcast discover üzenettel végzik a felderítést, melyre nyugtá(ka)t várnak (4.6 ábra).

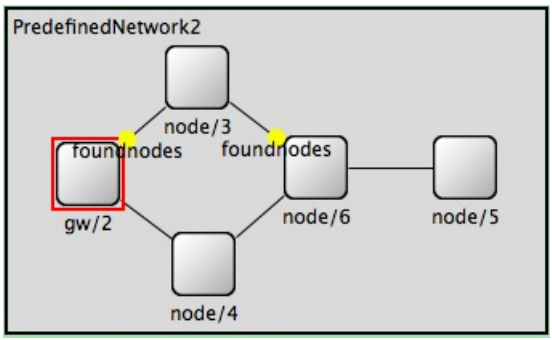


4.6. ábra. searchNodes csomag hatására történő discover broadcast

Az alábbi naplóbejegyzésekben is megfigyelhető a fenti folyamat lefutása:

```
node/3: searchnodes ARRIVED - OHDst: 3 OHSrc: 2 RealDst: 3 RealSrc: 2
node/3: routing table:
  - to: 3 via: 3 hops: 0
  - to: 2 via: 2 hops: 0
node/3: ack SEND - OHDst: 2 OHSrc: 3 RealDst: 2 RealSrc: 3
node/3: ------ changing phase to node/discover phase
node/3: discover SEND - OHDst: 255 OHSrc: 3 RealDst: 255 RealSrc: 3
```

A broadcast üzenetet küldő csomópont a bejövő nyugták küldőinek címeit összegyűjti és egy foundNodes csomagban eljuttatja a központi node-hoz (4.7 ábra). Mindemellett minden discover üzenetre vett ack csomag hatására az útvonaltáblák is frissülnek (lsd. alábbi napló részlet).



4.7. ábra. foundNodes csomag küldése a közoti node-hoz

```
node/3: ack ARRIVED - OHDst: 3 OHSrc: 6 RealDst: 3 RealSrc: 6
node/3: adding to routing table: (to: 6 via: 6 hops: 0)
node/3: routing table:
  - to: 3 via: 3 hops: 0
  - to: 2 via: 2 hops: 0
  - to: 6 via: 6 hops: 0
node/3: adding ack packet (OHSrcAddr 6) to foundNodesPacket.
node/3: foundnodes SEND - OHDst: 2 OHSrc: 3 RealDst: 2 RealSrc: 3
node/3: ackBuffer:
  - foundnodes resendCount: 0 OHDst: 2 OHSrc: 3 RealDst: 2 RealSrc: 3
node/3: ------ changing phase to standby phase
```

A foundNodes csomag beérkezésekor a központi node frissíti a route tábláját.

```
gw/2: foundnodes ARRIVED - OHDst: 2 OHSrc: 4 RealDst: 2 RealSrc: 4
gw/2: foundnodes packet content: 2 6
gw/2: adding to routing table: (to: 6 via: 4 hops: 1)
gw/2: routing table:
  - to: 2 via: 2 hops: 0
  - to: 3 via: 3 hops: 0
  - to: 4 via: 4 hops: 0
  - to: 6 via: 3 hops: 1
  - to: 6 via: 4 hops: 1
gw/2: ack SEND - OHDst: 4 OHSrc: 2 RealDst: 4 RealSrc: 2
gw/2: sendSearchNodesTo queue: 6 searchNodesHaveSentTo: 3 4
gw/2: searchnodes SEND - OHDst: 3 OHSrc: 2 RealDst: 6 RealSrc: 2
gw/2: ackBuffer:
  - searchnodes resendCount: 0 OHDst: 3 OHSrc: 2 RealDst: 6 RealSrc: 2
```

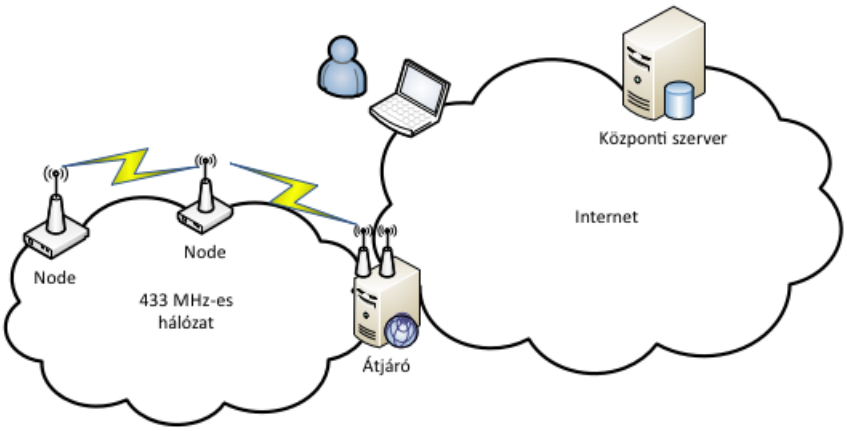
A folyamat mindaddig folytatódik, míg minden klaszterben lévő node-ot meg nem szólított a központi node. Ebben az esetben a hálózat discover fázisa lefutott és az egyes node-ok útvonal-táblái feltöltődtek. A központi node minden klaszterben található csomóponthoz ismer egy vagy több elérési lehetőséget, míg a többi node csak a hálózat topológiájának csak egy részét ismeri.

```
All routing tables' content:
gw/2: routing table:
  - to: 2 via: 2 hops: 0
  - to: 3 via: 3 hops: 0
  - to: 4 via: 4 hops: 0
  - to: 6 via: 3 hops: 1
  - to: 6 via: 4 hops: 1
  - to: 4 via: 3 hops: 2
  - to: 5 via: 3 hops: 2
  - to: 5 via: 4 hops: 2
  - to: 3 via: 4 hops: 2
```

```
node/3: routing table:
  - to: 3 via: 3 hops: 0
  - to: 2 via: 2 hops: 0
  - to: 6 via: 6 hops: 0
  - to: 4 via: 6 hops: 1
  - to: 5 via: 6 hops: 1
node/4: routing table:
  - to: 4 via: 4 hops: 0
  - to: 2 via: 2 hops: 0
  - to: 6 via: 6 hops: 0
node/5: routing table:
  - to: 5 via: 5 hops: 0
  - to: 6 via: 6 hops: 0
  - to: 2 via: 6 hops: 2
node/6: routing table:
  - to: 6 via: 6 hops: 0
  - to: 3 via: 3 hops: 0
  - to: 4 via: 4 hops: 0
  - to: 2 via: 3 hops: 1
  - to: 5 via: 5 hops: 0
```

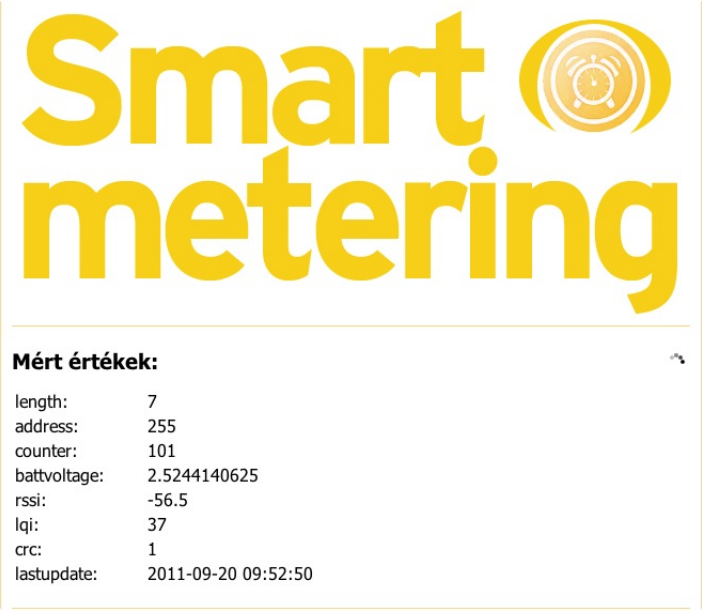
Az előző példán megmutattuk a discover fázis részletes működését. A szimuláció segítségével a teljes működési folyamatot megvizsgáltuk. A szimuláció alkalmas a fizikai valóság modellezésére, mivel lehetőségünk van terjedési késleltetés és véletlenszerű csomagvesztés beállítására is. Megvizsgáltuk, hogy node-ok valamint a közöttük lévő linkek kiesése esetén is stabil állapotba kerül a hálózat, miután minden node routing táblája frissült. Ha sikerül a helyreállítási folyamat, akkor pedig ismét konzekvens állapotba kerül a hálózat. Tesztjeink során nagyobb hálózatokon is verifikáltuk a protokoll helyes működését, mely eredményeként a tervezett protokoll jól skálázhatónak bizonyult, ezzel teljesítette a 4(b) és (c) célunkat.

A szimuláción túl egy valós példán is teszteltük a teljes rendszer működését. A tesztrendszer felépítését az 4.8 ábrán láthatjuk.



4.8. ábra. Tesztrendszer felépítése

A rendszerben 2 db node található, melyeken mérési adatokat generálunk, majd a mért adatokat a központi node felé továbbítjuk, amely jelenleg átjáróként is üzemel. Az átjáró 3G hálózaton továbbítja a vett adatokat egy szerverhez, amelyen webes felület segítségével nyomon követhetjük a méréseket. Az alábbi 4.9 ábra a kialakított webes felületet ábrázolja, melyen a mérési és fontosabb csomag adatok figyelhetők meg.



4.9. ábra. Mért adatok a webes felületen kijelezve

5. fejezet

Összefoglalás

Ebben a fejezetben összefoglaljuk a bevezetőben leírt célok teljesülését, bemutatjuk a jövőbeli kutatások irányát és a köszönetnyilvánítással zárunk.

5.1. Célok teljesülése

A cél egy olyan intelligens mérőrendszer megalkotása volt, amely energiatakarékos vezetéknélküli protokollt használ a szenzorhálózat elemei közötti kommunikáció során. A kitűzött cél megvalósítása érdekében a feladatot az alábbi konkrét lépésekre bontottuk:

1. Megismertük a jelenlegi technológiákat (protokollok és eszközök), részletesen elemeztük a bennük rejlő lehetőségeket, előnyöket és hátrányokat.
2. Összehasonlítottuk a smart metering-re alkalmas frekvenciákat különböző környezetekben végzett mérésekkel.
3. A legalkalmasabb frekvencián működő energia-hatékony mikrokontrollert és rádiós modult terveztünk.
4. Saját protokollt terveztünk intelligens mérőrendszer feladatokra, amely a következő célokat igazoltan teljesítette:
   a. energia-hatékonyság
   b. robusztusság
   c. skálázhatóság
   d. biztonságosság
5. A megtervezett protokollt szimulációs vizsgálatoknak vetettük alá.

A munkamegosztás a szerzők között az alábbi volt.

- Milánkovich Ákos 40%
- Ill Gergely 40%
- Varga Norbert 20%

5.2. Kitekintés

A tervezett protokoll és hálózati topológia nem csak smart metering alapú rendszerekben használható, hanem kisebb fajta feladatspecifikus módosításokkal tetszőleges vezetéknélküli szenzorhálózati feladatokra is. Így alkalmas lehet BAN (Body Area Network) kialakítására, vagy akár folyóvizek jellemzőinek méréséhez is.

A kidolgozott rendszer új menedzsment üzenetek bevezetését is támogatja, melyek segítségével egy központi állomásról küldhetünk különböző vezérlő csomagokat a hálózat monitorozásának, újrakonfigurálásának, illetve működésének megfigyelése érdekében. Mindemellett lehetőségünk van riasztási üzenetek definiálására, melyek a nem kívánt és a hálózat működése szempontjából fontos események jelzését segítik. Ebbe beletartozik az eszközök alacsony elem-feszültségszintjének jelzése, az irreleváns mért értékek jelzése például csőtörés esetén stb.

Lehetőségünk van a mért adatok begyűjtésén túl, azok elemzésére, statisztikák és kimutatások készítésére egy erre kialakított informatikai infrastruktúra támogatásával. Erre példa az általunk fentebb bemutatott egyszerű tesztrendszer is. A tervezett hardver eszköz alkalmas arra, hogy különböző mérőszenzorokkal egészítsük ki, így szélesítve felhasználhatósági körét.

5.3. Köszönetnyilvánítás

Köszönjük a konzulenseknek: Lendvai Károlynak és Dr. Szabó Sándornak értékes javaslataikat és visszajelzéseiket. Külön köszönjük:

- Marton Ákosnak, az elvégzett méréseknél és a saját hardver megtervezésénél a segítséget.
- Belső Zoltánnak, hogy rendelkezésünkre bocsátott tesztelési célokra MSP430 eszközöket.

Rövidítésjegyzék

AES Advanced Encryption Standard
API Application Programming Interface
BAN Body Area Network
CPU Central Processing Unit
CRC Cyclic Redundancy Code
CSMA-CA Carrier Sense Multiple Access with Collision Avoidance
DHCP Dynamic Host Configuration Protocol
(G)FSK (Gaussian) Frequency Shift Keying
IC Integrated Circuit
ISM Industrial Scientific Medical
ISO International Organization for Standardization
LOS Line of Sight
LSB Least Significant Bit MSB Most Significant Bit
MAC Message Authentication Code
MITM Man in the Middle
NAT Network Address Translation
OSI Open Systems Interconnection
PAN Personal Area Network
PHY Physical layer
POSIX Portable Operating System Interface for Unix
RAM Random Access Memory
RFID Radio Frequency IDentification
ROM Read-Only Memory
RSA Rivest, Shamir és Adleman (kriptográfiai titkosító algoritmusa)
SNR Signal Noise Ratio
TDMA Time Division Multiple Access
USB Universal Serial Bus
XTEA Extended Tiny Encryption Algorithm

Ábrák jegyzéke

Irodalomjegyzék

[1] E. S. M. I. Group, „Smart metering for europe," 2009.

[2] I. Crossbow Technology, „Micaz wireless measurement system datasheet," 2011. [Online, 2011. okt. 9.].

[3] Wikipedia, „Smart meter — wikipedia, the free encyclopedia," 2011. [Online, 2011. okt. 9.].

[4] B. Insight, „News archive," 2010. [Online, 2011. okt. 9.].

[5] S. I. Haddad Richárd, Dr. Morva György, „Smart metering," 2007. [Intelligens Energiarendszerek].

[6] G. D. Klaas De Craemer, „Analysis of state- of-the-art smart metering communication standards," 2010. [Leuven, YRS].

[7] I. S. Tóth Katalin, Schulcz Róbert, „Ütközésfeloldás rfid rendszerekben." [Híradástechnika folyóirat, 2007/4, 8. p 39-46.].

[8] I. S. Organisation, „Iso/iec 18000-7," 2009.

[9] J. Norair, „Introduction to dash7 technologies," 2009.

[10] D. Alliance, „Dash7 technical overview webinar," 2009. dec. 2.

[11] J. Norair, „Dash7 mode 2 wiki," 2011. [Online, 2011. okt. 9.].

[12] J. Norair, „Opentag: Office hours webinar," 2010. máj. 12. [Online, 2011. okt. 9.].

[13] J. P. Norair, „Opentag on sourceforge," 2011. [Online; accessed 8-May-2011].

[14] J. Norair, „Opentag wiki," 2011. [Online, 2011. okt. 9.].

[15] ONE-NET, „One-net specification," 2011.

[16] T. Instruments, „Simpliciti overview." [Online, 2011. okt. 9.].

[17] L. Friedman, „Simpliciti: Simple modular rf network specification," 2009. [Texas Instruments Inc.].

[18] T. Instruments, „Simpliciti," 2011. [Online; accessed 8-May-2011].

[19] Z.-W. Alliance, „About us," 2011. [Online, 2011. okt. 9.].

[20] Z. Alliance, „Zigbee specification," 2008.

[21] A. Tatsiopoulos, C. & Ktena, „A smart zigbee based wireless sensor meter system," 2009. 16th International Conference on Systems Signals and Image Processing 1-4,.

[22] T. Gábor, „Rádióhálózatok zigbee-adatátvitel alapján," 2008. [Magyar Elektronika 2008/1-2].

[23] R. R. L. Skrzypczak, D. Grimaldi, „Basic characteristics of zigbee and simpliciti modules to use in measurement systems," 2010. [in proceedings of the 14th WSEAS international conference on Systems: part of the 14th WSEAS CSCC multiconference - Volume II].

[24] Coronis, „Wavenis datasheet," 2011. [Online, 2011. okt. 9.].

[25] IEEE, „Specification of the bluetooth system," 1999. [Online, 2011. okt. 9.].

[26] G. Mulligan, „The 6lowpan architecture," 2007. [in proceedings of the 4th workshop on Embedded networked sensors ACM, New York].

[27] M. Nesterenko, „Ieee 802.15.4 presentation," 2011. [Online, 2011. okt. 9.].

[28] T. Instuments, „Msp430 hardware tools user's guide," 2010.

[29] Atmel, „Atmel avr xmega a4 microcontroller," 2010.

[30] L. K. S. S. I. S. Ill Gergely, Milánkovich Ákos, „Analysis of wireless smart metering solutions," 2011. [SoftCOM 2011 Workshop].