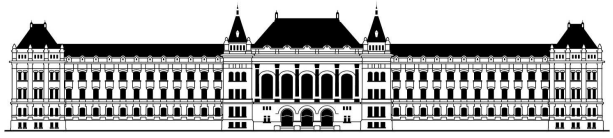


BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
HÍRADÁSTECHNIKAI TANSZÉK

ÖNJAVÍTÓ AGGREGÁLÁS ÉS AGGREGÁTOR NODE VÁLASZTÁS SZENZORHÁLÓZATOKBAN

Tézisfüzet

Schaffer Péter

Konzulens:
Buttyán Levente, Ph.D.

Budapest
2009

1. Bevezetés

A szenzorhálózatok elosztott rendszerek, amelyek több száz vagy akár több ezer apró, olcsó, kis teljesítményű szenzor node-ból (másnéven szenzor csomópontból) és néhány nagy teljesítményű bázisállomásból állnak. A szenzorok jellemzően valamilyen fizikai jelenséget mérnek és a méréseiket a bázisállomásnak küldik vezetéknélküli közegen. A bázisállomás adatfeldolgozást végez és hozzáférést nyújt az eredményekhez más hálózatok számára (pl. az Internet számára). A szenzorok csak rövid hatótávolságú adatközlésre alkalmasak, ezért várhatóan a szenzorok többugrásos hálózatokat fognak alkotni, amelyben a node-ok egymás üzeneteit is továbbítják a bázisállomás felé és hasonlóképpen visszafelé is. A szenzorok által küldött üzenetek számának csökkentése érdekében hálózaton belüli adatfeldolgozást (in-network processing) alkalmazhatunk, amikoris néhány szenzor node aggregálási feladatokat is ellát. Az aggregátor csomópont összegyűjti a környező szenzorok mérési eredményeit és helyileg feldolgozza azokat, majd egyetlen aggregált üzenetet továbbít a bázisállomás felé.

Tekintve, hogy a szenzorhálózatok igen fontos feladatokat is elláthatnak (pl. biztonságtechnikai és katonai alkalmazások esetén), joggal feltételezhetjük, hogy ezeket a hálózatokat számos különböző támadás érheti. Még ha feltételezzük is, hogy a támadó lehetőségei korlátozottak, vagy hogy a szenzorok behatás-ellenállóak (tamper resistant), a támadó képes *bemeneten alapuló támadás*t véghezvinni, vagyis képes a szenzor által mért fizikai jellemzőket befolyásolni, ami által képes a mért értékeket ”meghamisítani” és a bázisállomás által számolt aggregátumot eltorzítani. Ez a meghamisítás a szenzorok környezetének a fizikai paramétereinek a megváltoztatásával érhető el. Ezt a támadást nem lehet kriptográfiai úton detektálni vagy kiküszöbölni. Sőt, ez a fajta támadás viszonylag egyszerűen kivitelezhető. Először is, a támadó könnyedén megközelítheti a szenzorokat, tekintve, hogy azok általában felügyelet nélkül működnek. Másodszor, ezen szenzorok méréseinek kompromittálása nem igényel speciális eszközöket, hanem általában mindennapi eszközök használhatóak hatékonyan (pl. egy öngyújtó, egy zseblámpa, vagy egy pohár víz elegendő lehet hőmérséklet-, fényerősség-, ill. páratartalom mérő szenzorok mérési eredményeinek kompromittálásához). Sajnálatos módon a legtöbb aggregáló függvény érzékeny akár egyetlen hibás mérési eredményre is, vagyis akár egyetlen szenzor mérési eredményének megfelelő eltorzításával tetszőlegesen módosíthatjuk az aggregálás végeredményét. Az alkalmazás fontosságától függően ennek akár fatális következményei is lehetnek.

Létezik már a statisztikának egy ága, amelyik a minta nagy részétől eltérő elemekkel foglalkozik; ezt az ágat *robusztus statisztiká*nak hívják. Mindazonáltal, a robusztus statisztika eszközei pontatlanok lehetnek szenzorhálózati alkalmazások esetén. Általános értelemben a robusztus és ellenálló (resistant) módszerek csak bizonyos struktúrájú eltéréseket képesek detektálni, és ezen detektálási képességük gyorsan csökken a minta méretének növekedésével [Oli05]. A mi esetünkben a támadó bármilyen eltérést előidézhet, így az olyan megoldások, amelyek csak bizonyos struktúrájú eltéréseket képesek detektálni esetleg nem alkalmasak. Továbbá, a legtöbb robusztus és ellenálló módszer számítási komplexitása nagy, ami alkalmatlanná teszi ezeket kis energiafogyasztású, egyszerű szenzor node-okon való alkalmazásra. Végül, az ellenálló becslési eljárások felülmúlhatják a klasszikus becslési eljárásokat amennyiben vannak a mintából kilógó elemek, de sokkal rosszabbak azoknál, ha nincsenek ilyenek [Oli05]. Ezek a problémák motiválták a kutatásomat, amely során olyan új módszerek kifejlesztésén dolgoztam, melyek képesek egy eltökélt támadó ellen védekezni.

A kapcsolódó irodalomban az olyan módszereket, amelyek a bemenetre irányuló támadás ellen védekeznek a szenzorhálózatok témakörében önjavító aggregálási sémáknak nevezik (lsd. például [Wag04]). A robusztus statisztikák hátrányainak kiküszöbölésére az 1. tézisben egy kétlépéses önjavító aggregálási megoldás alkalmazását javasolom, amelyik az első lépésben megvizsgálja a mintát és statisztikai hipotézisvizsgálat segítségével támadás jeleit keresi abban, majd ezután, ha nem volt támadásra utaló jel, akkor elvégzi az aggregálást a szokásos módon. Máskülönben, ha támadást fedezett fel, akkor egy speciális műveletet hajt a támadás aggregátumra gyakorolt hatásának enyhítésére.

Ezután, a 2. tézisben egy, a RANSAC (RANdom SAmple Consensus) elvre épülő önjavító aggregá-

lási technikát javasolok, amit RANBAR-nak nevezek. A RANSAC elv egy tanácsot ad arra vonatkozóan, hogy hogyan kell a mérési eredmények egy modelljét létrehozni, ha sok közöttük a kompromittált elem. Az elv legérdekesebb tulajdonsága az, hogy a klasszikus statisztikai hozzáállással ellentétben, amelyik próbál minél több elemet bevonni a modellalkotásba, a RANSAC igyekszik a lehető legkevesebb elemből felépíteni ezt a modellt.

A szenzorhálózatok önjavító aggregálási megoldásai mellett a 3. tézisben az aggregátor node választás problémájával foglalkozom. Mivel az aggregátor csomópontok a többi node-nál több feladatot látnak el (ezeknek a node-oknak kell összegyűjteniük és aggregálniuk a mérési eredményeket), több energiát is használnak fel. Ez az oka, hogy kívánatos az aggregátor szerepét időről időre más node-nak átadni, ezáltal elosztva a terhelést a szenzor node-ok között.

A 3. tézisben egy aggregátor kiválasztási protokollt javasolok vezetéknélküli szenzorhálózatokhoz, amely a PANEL nevet viseli és a szenzorok geográfiai elhelyezkedése alapján választja ki, hogy melyik legyen az aggregátor. A PANEL gondoskodik a terhelés elosztásáról, ugyanis minden szenzort nagyjából egyenletesen gyakran választ aggregátornak. Sőt, a PANEL egy fontos tervezési kritériuma volt, hogy az aggregátor kiválasztását manipulálhatatlan módon tegye, vagyis, hogy biztosítsa, hogy egyik node se lehessen gyakrabban aggregátor, mint a többiek. A manipulálhatatlanság mellett a PANEL magas szintű biztonságot is biztosít, ugyanis sikerrel tud védekezni számos támadás ellen amelyek az aggregátum eltorzítását, ill. az aggregátor választási folyamat megzavarását célozzák.

2. Kutatási célkitűzések

Az önjavító aggregálási sémák megpróbálják minimalizálni a környezetet megváltoztató támadó hatását az aggregálási függvény kimenetére. Az én célom az önjavító aggregálási sémák tervezési alapelveinek megértése, és ezen ismeretek alapján egy olyan új önjavító aggregálási algoritmus létrehozása, amelyik alkalmazható szenzorhálózatokban és ugyanakkor felülmúlja az eddig javasolt önjavító aggregálási sémákat a támadó által okozható torzítás tekintetében.

Az aggregátor csomópontok megpróbálják összegyűjteni a mérési adatokat a többi szenzortól és aggregálják azokat annak érdekében, hogy csökkentsék a bázisállomásnak küldendő adat mennyiségét. Mivel az aggregátorok több feladatot látnak el, mint a többi szenzor, így több energiát is használnak fel. Az én célom ezen a tématerületen az egyenetlen energiafelhasználás problémájának megoldása. Ehhez szeretnék egy olyan új aggregátor választási protokollt javasolni, amelyik egyenletesen osztja el a node-ok között a hálózatra eső terhelést, valamint hatékonyabb ebből a szempontból, mint a létező megoldások.

3. Kutatási módszerek

A szenzorhálózati önjavító aggregációval kapcsolatos eredményeim jelentősen támaszkodnak a valószínűségelméletre és a statisztikára. A szenzorok méréseit mindig valószínűségi változóknak tekintem, akár függetleneknek, akár korreláltaknak. Ez az absztrakció teszi lehetővé számomra, hogy használjam a statisztikai jelöléseit és tudásanyagát. Az eredményeim elsősorban analítikusak, de néhány esetben a probléma bonyolultsága megkövetelte a szimulációs eszközök és az empirikus vizsgálat alkalmazását. Utóbbi esetekben, illetve a numerikus analízis használata esetén a Maple, a Mathematica és a Matlab programokat használtam.

A szenzorhálózati aggregátor választással kapcsolatos eredményeim és következtetéseim átfogó szimulációkon alapszanak. Minden alkalommal tipikus hálózati topológiákat feltételezve szimulációs vizsgálatokat végeztem a javasolt protokoll energiafogyasztási és kluszterezési képességeinek feltárása érdekében. Az elvégzett szimulációk általában összehasonlító jellegűek, vagyis a javasolt megoldást egy másik, jól ismert algoritmushoz hasonlítom. Ezen szimulációkhoz a Matlab, a TOSSIM és a Power-TOSSIM programokat használtam.

4. Új Eredmények

4.1. A Mintafelező Eljárás Önjavító Aggregáláshoz

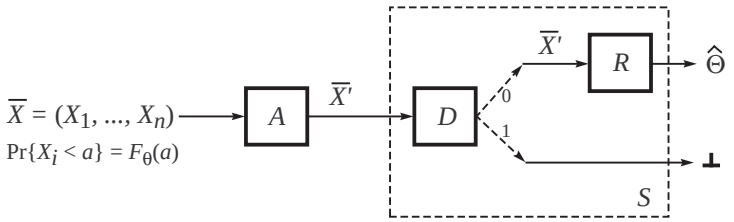
A szenzorhálózati alkalmazási lehetőségek egy potenciális problémája, hogy a szenzorok mérési eredményei kompromittálódhatnak mielőtt elérik a bázisállomást vagy az aggregátor node-ot. Egy támadó ezt például a környezeti paraméterek néhány szenzor körül történő megváltoztatásával érheti el, így kompromittálhatja a méréseiket. Ez a támadás nem detektálható és nem küszöbölhető ki kriptográfia segítségével.

1. TÉZISCSOPORT: *Javasolok egy kétmintás homogenitásvizsgálati eljárást, a mintafelezési eljárást, a környezetet megváltoztató támadás elleni védekezés céljából. Bemutatom két felhasználási lehetőségét ennek az általános elvnek; egyrészt független, másrészt korrelált szenzor mérések esetére. [J1] [C1] [C4] [P2]*

A mintafelező eljárás konzisztenciavizsgálatot hajt végre a mintán annak elfelezésével és a két mintafél statisztikai értelemben vett összehasonlításával (másszóval mintahasítást (data splitting) végez). A továbbiakban ezen általános elv két kissé különböző alkalmazási módját mutatom be. Az első esetben azt feltételezem, hogy a kapott minta statisztikailag független elemekből áll, majd alkalmazom a mintafelezés módszerét korrelált elemekre is. Mindkét esetben elkülönítem a támadásdetekciót az aggregálástól. A minta felezése és ellenőrzése a támadásdetekciós lépés része, az aggregálás (vagyis a kívánt statisztikai függvény kiszámolása a mintán) pedig csak azután történik meg, ha a támadásdetekciós lépés sértetlennek találta a mintát.

1.1. TÉZIS: *Javasolok egy új szenzorhálózati önjavító aggregálási modellt független szenzormméréseket feltételezve. A modellben az aggregátor megvizsgálja a mérési eredményeket és megpróbálja detektálni a váratlan mintaelemeket mielőtt elvégezné az aggregálást. A támadó célja ebben a modellben a szenzorok mérési eredményeinek elrontása úgy, hogy az aggregáló függvény kimenetén jelentkező torzítás maximális legyen, de ugyanakkor a támadás észrevétlen maradjon. [C4]*

A szenzorok méréseit függetlennek feltételező esetre javasolt, támadásdetekcióval kiegészített aggregálási modellemet az 1. ábra szemlélteti. Feltételezem, hogy n szenzor végez mérést és küldi el a mérések eredményét a bázisállomásnak. A bázisállomás aggregálja a kapott adatokat; ezen aggregálás célja az ismeretlen θ paraméter becslése. Az i. szenzor mérési eredményet az X_i valószínűségi változóval jelölöm, amelynek eloszlása θ-tól függ. Például θ lehet az átlaghőmérséklet és X_i-k eloszlása lehet $\mathcal{N}(\theta, 1)$, a Gauss-eloszlás θ várható értékkel és 1 szórásnégyzettel. Feltételezem, hogy az X_i $(i = 1, 2, \ldots, n)$ valószínűségi változók független azonos eloszlásúak. $\bar{X} = (X_1, X_2, \ldots, X_n)$ az a vektor, ami a szenzorok méréseit tartalmazza.



1. ábra. Támadásdetekcióval kiegészített aggregálási modell

A támadó képes módosítani a szenzorok mérési eredményeit mielőtt azok bekerülnek az aggregáló

függvénybe. Ezt A-val modelleztem, amelynek bemenete az eredeti mérési eredmények vektora ($\bar{X}$), míg kimenete a módosított vektor ($\bar{X}'$).

S működése formálisan a következő:

$$S(\bar{X}') = \begin{cases} R(\bar{X}') = \hat{\Theta} & \text{ha } D(\bar{X}') = 0 \\ \perp & \text{ha } D(\bar{X}') = 1 \end{cases} \tag{1}$$

ahol $\perp$ egy speciális szimbólum ami azt jelenti, hogy a támadás detektálva lett.

Feltételezem, hogy a támadó maximalizálni akarja az aggregáló függvény d torzítását, amit a következőképpen definiálok:

$$d = \mathbb{E}[\ |\theta - \hat{\Theta}|\ ] = \mathbb{E}[\ |\theta - R(A(\bar{X}))|\ ] \tag{2}$$

Továbbá feltételezem, hogy a támadó rejtve akar maradni, vagy pontosabban, hogy a támadó szeretné a támadás sikeres detektálásának valószínűségét egy p^* érték alatt tartani:

$$P\{D(\bar{X}') = 1\} = P\{D(A(\bar{X})) = 1\} \leq p^* \tag{3}$$

Feltételezem, hogy a támadó ismeri a D detekciós algoritmust (az abban használt a priori ismeretekkel együtt), valamint az R aggregáló függvényt is. A támadó egy paramétere azon $t < n$ szenzorok száma, amelyeket kompromittált. Ez azt jelenti, hogy $\bar{X}$ és $\bar{X}'$ pontosan t pozícióban különbözik.

Az én modellem újdonsága [Wag04]-hez képest az, hogy én alkalmazok egy támadásdetekciós lépést, és ha nincs támadás, akkor azokat az aggregálási függvényeket is fel tudom használni, amelyek amúgy nem tekinthetők önjavítónak egy olyan támadót feltételezve, aki képes a környezet paramétereit megváltoztatni a szenzorok körül.

1.2. TÉZIS: *Javasolok egy kétmintás támadásdetekciós algoritmust, ami a mintafelezés elvét alkalmazza és beleillik az 1.1. tézisben leírt modellbe egy specifikus támadó esetén. Ez a specifikus támadó képes a szenzorok egy a támadás előtt kiválasztott részhalmazának mérési eredményeit módosítani, ahol is a módosítás egy pozitív konstans hozzáadását jelenti minden mérési eredményhez. [C4]*

Egy olyan támadót feltételezek, aki képes megfigyelni és módosítani $t \ll n$, a támadás előtt kiválasztott szenzor mérési eredményét. A támadó úgy támad, hogy egy pozitív $m > 0$ konstansot ad minden kiválasztott szenzor mérési eredményéhez. Feltételezem, hogy a szenzorok X_i ($1 \leq i \leq n$) mérési eredményei független azonos eloszlásúak. Feltételezem, hogy semmi sem ismert ezen eloszlással kapcsolatban, csak az, hogy a szórásnégyzete 1.

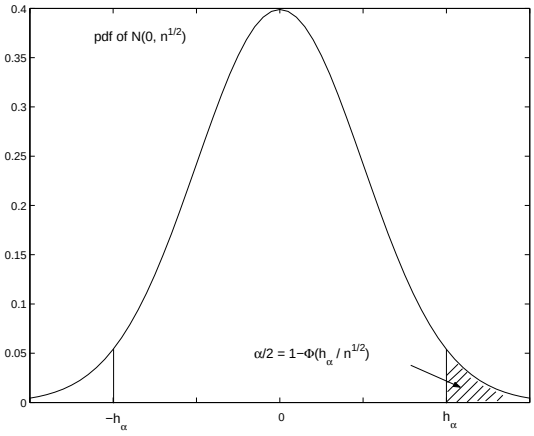
A támadásdetekció a következő algoritmus szerint működik. Először kiszámoljuk a $Z_1 = X'_1 + \ldots + X'_{n/2}$ és $Z_2 = X'_{n/2+1} + \ldots + X'_n$ értékeket, ahol az egyszerűség kedvéért n párosnak feltételezhető, majd kiszámoljuk a $W = Z_1 - Z_2$ értéket. A központi határeloszlás-tétel alapján ismert, hogy ha nincs támadás, akkor W eloszlása megközelítőleg $\mathcal{N}(0, \sqrt{n})$, a Gauss-eloszlás 0 várható értékkel és $\sqrt{n}$ szórásnégyzettel. Ezért gyanús, ha $|W|$ értéke távol van 0-tól. A támadásdetektáló algoritmus természetes módon használja a $h_\alpha > 0$ küszöbértéket:

$$D(\bar{X}') = \begin{cases} 1 & \text{ha } |W| > h_\alpha \\ 0 & \text{egyébként} \end{cases} \tag{4}$$

h_α értékét a detekciós algoritmus α paramétere határozza meg, ami a nem támadott esetben (H_0 hipotézis) előforduló hibás detekció valószínűsége:

$$P\{|W| > h_\alpha \mid H_0\} = 2 - 2 \cdot \Phi(h_\alpha / \sqrt{n}) = \alpha \tag{5}$$

A h_α és α közötti kapcsolatot a 2. ábra szemlélteti. Valójában ez az algoritmus szoros kapcsolatban áll a kétmintás U-próbával, mindazonáltal, amíg ez utóbbi megköveteli a normalitást, a mintafelezés ötlete általánosabbnak tekinthető ennél, hiszen bármilyen paraméteres eloszlás esetén alkalmazható (pl. aszimmetrikus eloszlások esetén).



2. ábra. h_α értékét a hibás detekció α valószínűsége határozza meg a támadásmentes esetben, ami a $\mathcal{N}(0, \sqrt{n})$ eloszlás farkának felel meg

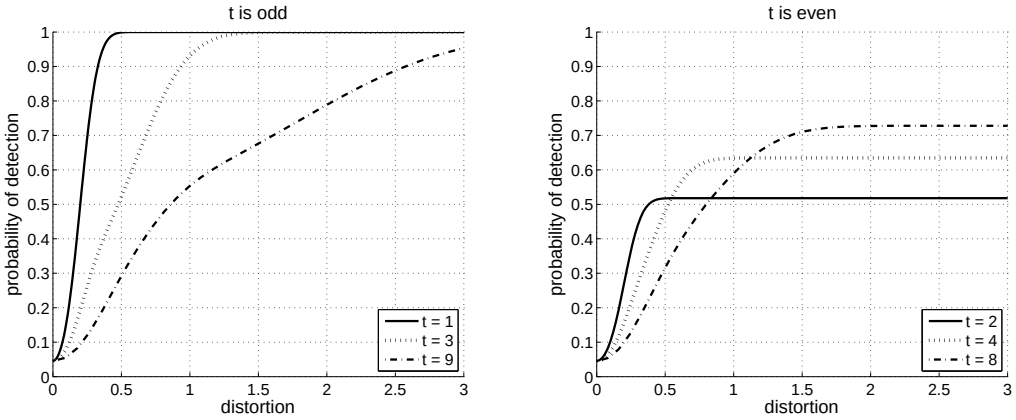
1.3. TÉZIS: *Analítikusan meghatározom az 1.2. tézisben javasolt algoritmus támadásdetekciós valószínűségét az 1.2. tézisben javasolt specifikus támadó esetén. [C4]*

A W valószínűségi változó $\mathbb{E}[W]$ várható értéke m többszöröse és a $[-tm, tm]$ intervallumba esik. Ha t_1 jelöli a kompromittált elemek számát az első mintafélben ($X'_1, \ldots, X'_{n/2}$), és t_2 jelöli a kompromittált elemek számát a második mintafélben ($X'_{n/2+1}, \ldots, X_n$), ahol is $t_1 + t_2 = t$, akkor

$$\begin{aligned} \mathbb{E}[W] &= \mathbb{E}[X'_1 + \ldots + X'_{n/2}] - \mathbb{E}[X'_{n/2+1} + \ldots + X'_n] \\ &= (\frac{n}{2} \cdot \theta + t_1 \cdot m) - (\frac{n}{2} \cdot \theta + t_2 \cdot m) \\ &= (t_1 - t_2) \cdot m \end{aligned} \tag{6}$$

Ez alapján a következőt írhatjuk fel a detekció valószínűségére a támadott esetben:

$$P\{D(\bar{X}') = 1 \mid H_1\} = \sum_{\ell=-t}^{t} P\{|W| > h_\alpha \mid \mathbb{E}[W] = \ell m\} \cdot P\{\mathbb{E}[W] = \ell m\} \tag{7}$$



3. ábra. A támadásdetekció valószínűsége a támadó által elért torzítás függvényében $n = 100$ és $\alpha \approx 0.05$ ($h_\alpha = 20$) értékekre.

A (7) egyenlet kombinatorika segítségével kiértékelhető a paraméterek adott értékeire. A 3. ábra ezen számolás eredményeit illusztrálja $n = 100$ és $\alpha \approx 0.05$ ($h_\alpha = 20$) értékekre. A baloldali ábra t páratlan, míg a jobboldali ábra t páros értékeit szemlélteti. A különböző görbék t különböző értékeihez tartoznak.

Jól látszik, hogy ha a támadó szeretné a támadása detektálásának valószínűségét egy adott p^* küszöbérték alatt tartani, akkor az általa elérhető torzítás erősen korlátos. Például, ha $p^* = 0.3$, akkor a torzítás nem lehet 0.5-nél nagyobb még akkor sem, ha a támadó 100 elemből 9-et kompromittált. p^* ugyanezen értékére a maximálisan elérhető torzítás 0.1-re csökken, ha a támadó csak 1 elemet kompromittál. Érdekesség, hogy az elérhető torzítás maximuma nem függ θ értékétől, ami azt jelenti, hogy a d/θ relatív torzítás nagyon kicsi lehet θ nagy értékeire.

A következőkben bemutatom a korrelált mintákkal kapcsolatos munkámat. A korreláció egy természetes jelenség mérési adatok esetében, ilymódon figyelembe kellene venni amikor szenzorhálózati mérési eredményekről beszélünk. Vizsgálataimhoz egy kézenfekvő szenzorhálózati adatmodellt alkalmaztam, amelyik képes a korreláció modellezésére, nevezetesen, a páronkénti korrelációs együtthatókat kezeli a számlások során.

1.4. TÉZIS: *Javasolom az Attack Detection Algorithm és az Enhanced Data Aggregation Algorithm eljárásokat önjavító aggregálási metódusként, amelyek felhasználják a mintaelemek közötti lineáris korrelációt. [J1] [C1] [P2]*

A támadás detektálására kifejlesztett metódus az 1. algoritmus.

Algorithm 1 $Det(x_1, x_2)$ Attack Detection Algorithm

1: Véletlenszerűen kiválasztunk egy elemet az $\{x_1, x_2\}$ mintából és a kiválasztott elemet x'-vel, a másik elemet x''-vel jelöljük
2: Kiszámoljuk x'' $(1-\alpha)$%-os konfidencia-intervallumát x' feltétel mellett a $p_{X''|X'}(\cdot|x')$ sűrűségfüggvény szerint
3: **if** x'' a konfidencia-intervallumon belül található **then**
4: $D = 0$ (* nincs detektált támadás*)
5: **else**
6: $D = 1$ (* támadás detektálva *)
7: **end if**

Ez a kézenfekvő megközelítés már ki is használja a korrelációt a $p_{X_1|X_2}(\cdot|\cdot)$ és $p_{X_2|X_1}(\cdot|\cdot)$, ismertnek feltételezett feltételes sűrűségfüggvények segítségével. (X_i jelöli az i. támadatlan mintaelemet.) $p_{X_1|X_2}(\cdot|\cdot)$ és $p_{X_2|X_1}(\cdot|\cdot)$ ismerete nem implikálja az egyes szenzorok mérési eredményei eloszlásának a priori ismeretét. Például egy adott $p_{X_1|X_2}(\cdot|\cdot)$ sűrűségfüggvény különböző együttes sűrűségfüggvényt ad X_2 különböző eloszlásai esetén, ami pedig X_1 különböző határeloszlásait eredményezi. Következésképpen nem feltételezek semmilyen a priori ismeretet a mérési eredmények várható értékével kapcsolatban.

Az 1.ãlgoritmus kimenete használható fel az aggregálás módjának megfelelő megválasztásához. Az én megközelítésem az Enhanced Data Aggregation Algorithmban (2. algoritmus) lett formalizálva, ahol az y kimenet a bemenet aggregáltja, míg y_{extr} a minimális torzítású kimenet, ha nem szűrjük ki a kilógó elemeket. y_{extr} általában extrapolációval számolható a korábbi nem támadott kimenetek segítségével.

Az Enhanced Data Aggregation Algorithm kimenete az aktuális kör aggregátuma. Az Attack Detection Algorithm és az Enhanced Data Aggregation Algorithm eljárások használatával jelentősen csökkenthető az aggregátum torzítása összehasonlítva azzal az esettel, amikor az aggregálás előzetes vizsgálat nélkül történik.

A fenti algoritmusok tetszőleges méretű minták esetén is alkalmazhatóak, miután a mintát két részre osztottuk és mindkét részt egy-egy elemmé tömörítettük. Egy kis módosítás szükséges ilyenkor az 1. algoritmusnál, nevezetesen ilyenkor az átlagok feltételes sűrűségfüggvényeit kell alkalmaznunk, hiszen

Algorithm 2 Enhanced Data Aggregation Algorithm

1: Vegyük mindkét mérési eredményt és alkalmazzuk a $Det(x_1, x_2)$ Attack Detection Algorithm eljárást
2: **if** $Det(x_1, x_2)$ támadást jelez **then**
3: Kimenet = y_{extr}
4: **else**
5: Kimenet = y
6: **end if**

$p_{X_1|X_2}(\cdot|\cdot)$ és $p_{X_2|X_1}(\cdot|\cdot)$ helyett $p_{\overline{X}_1|\overline{X}_2}(\cdot|\cdot)$ és $p_{\overline{X}_2|\overline{X}_1}(\cdot|\cdot)$ sűrűségfüggvényekre van szükségünk a kapcsolódó konfidencia-intervallum meghatározásához.

1.5. TÉZIS: *Analítikusan meghatározom az Attack Detection Algorithm másodfajú hibavalószínűségét feltételezve, hogy a támadó az elemeket egy eltolás hozzáadásával tudja támadni, aholis az eltolás értékei független azonos eloszlásúak tetszőleges paraméterű normális eloszlás szerint. Továbbá a másodfajú hibavalószínűség felhasználásával analítikusan meghatározom a támadó által az Enhanced Data Aggregation Algorithm kimenetén elért torzítás mértékét. [J1] [C1] [P2]*

A β másodfajú hiba a részvalószínűségek alapján a következőképpen határozható meg:

$$\beta = \sum_{j=0}^{t} P(t_1 = j)\beta^{(j,t-j)} \tag{8}$$

ahol

$$P(t_1 = j) = \frac{\binom{t}{j}\binom{n-t}{\frac{n}{2}-j}}{\binom{n}{\frac{n}{2}}} \tag{9}$$

a hipergeometrikus eloszlás n, t és $\frac{n}{2}$ paraméterekkel. A $\beta^{(j,t-j)}$ részvalószínűségek definíciója

$$\beta^{(t_1,t_2)} = \frac{1}{2}(\beta^{(1)} + \beta^{(2)}) \tag{10}$$

ahol (t_1, t_2) felső index jelöli azt, hogy a minta első fele t_1, míg a minta második fele t_2 kompromittált elemet tartalmaz ($t = t_1 + t_2$). A $\beta^{(t_1,t_2)}$ a két különböző feltételválasztásnak megfelelő részvalószínűségek átlaga (lsd. 1. algoritmus). Ezek a részvalószínűségek a következőképpen határozhatóak meg:

$$\beta^{(1)} = \int_{-\infty}^{\infty}\int_{b_1(\overline{x}_{h,1})}^{b_2(\overline{x}_{h,1})} p_{\overline{X}_{h,2},\overline{X}_{h,1}}(u, v)\mathrm{d}u\mathrm{d}v \tag{11}$$

$$\beta^{(2)} = \int_{-\infty}^{\infty}\int_{b_1(\overline{x}_{h,2})}^{b_2(\overline{x}_{h,2})} p_{\overline{X}_{h,1},\overline{X}_{h,2}}(u, v)\mathrm{d}u\mathrm{d}v \tag{12}$$

ahol a h alsó index utal arra, hogy az elemek kompromittáltak lehetnek.
Az Enhanced Data Aggregation Algorithm kimenetén jelentkező torzítás a következőképpen fejezhető ki:

$$\begin{aligned} d(Y|A=1) &= E\big[|Y-\hat{Y}|^2\big|A=1\big] = \\ &= E\big[|Y-\hat{Y}|^2\big|A=1, D=1\big]\cdot(1-\beta) + E\big[|Y-\hat{Y}|^2\big|A=1, D=0\big]\cdot\beta \\ &= E|Y_{extr}-\hat{Y}|^2\cdot(1-\beta) + \frac{1}{n^2}\big(\tilde{\mu}^2+\tilde{\sigma}^2\big)\cdot\beta \end{aligned} \tag{13}$$

ahol $A = 1$ jelentése, hogy támadás történt. Feltételezve, hogy $E|Y_{extr} - \hat{Y}|^2$ értéke megközelítőleg 0, a torzítás felírható

$$d(Y|A = 1) \cong \frac{1}{n^2}\left(\tilde{\mu}^2 + \tilde{\sigma}^2\right) \cdot \beta \tag{14}$$

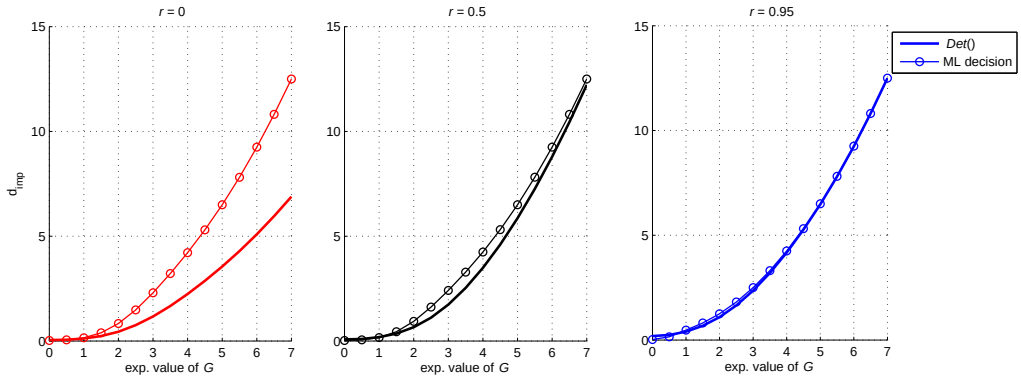
alakban.

1.6. TÉZIS: *Összehasonlítom az 1.4. tézisben javasolt Attack Detection Algorithm torzítását a Maximum Likelihood döntés által okozott torzítással abban az esetben, amikor a minta kételemű és eloszlása standard normális. Eredményeim tanulsága szerint a két algoritmus torzításának különbsége a korreláció növekedésével csökken. Ha a korreláció értéke megközelítőleg 1, a két algoritmus majdnem azonos torzítás elérésére képes. [J1]*

A Maximum Likelihood döntés nem alkalmazható az én adat- és támadómodellem esetén további feltételezések nélkül, mindazonáltal annak a döntéselméletben való fontossága arra ösztönzött, hogy mégis összehasonlítsam a hatékonyságát az 1. algoritmus hatékonyságával egy jelentősen korlátozott modellben. A korlátozás a következő: feltételezem, hogy a támadó eloszlása *előzetesen* ismert. Hangsúlyozom, hogy ez a feltételezés szükséges a Maximum Likelihood döntés működéséhez, viszont nem összekeverendő a normalitási feltétellel, amelyet csak a javasolt eljárás analízisének végrehajthatósága miatt tettem; az Attack Detection Algorithmnek nem szükséges ismernie a támadó eloszlását, míg a Maximum Likelihood döntésnek szükséges. Az egyszerűség kedvéért feltételezem, hogy a támadó eloszlása ismert paraméterű normális eloszlás.

Hogy megfigyelhessük a Maximum Likelihood döntés torzításra gyakorolt hatását, beleraktam azt az Enhanced Data Aggregation Algorithmba a $Det(\cdot,\cdot)$ helyére. A 4. ábra mutatja gaussi adatmodell esetén az Attack Detection Algorithm és a Maximum Likelihood döntés összehasonlításának eredményét, külön-külön az Enhanced Data Aggregation Algorithm részeiként. A paraméterek értékei $\mu = 0$, $\sigma = 1$, $\tilde{\sigma} = 1$, továbbá d_{imp} definíciója

$$\begin{aligned} d_{imp} &= d(Y|A = 1, D = 0) - d(Y|A = 1) \\ &= \frac{1}{4}\left(\tilde{\mu}^2 + \tilde{\sigma}^2\right) - \left[E|Y_{extr} - \hat{Y}|^2 \cdot (1 - \beta) + \frac{1}{4}\left(\tilde{\mu}^2 + \tilde{\sigma}^2\right)\beta\right] \\ &\cong \frac{1}{4}\left(\tilde{\mu}^2 + \tilde{\sigma}^2\right) \cdot (1 - \beta) \end{aligned} \tag{15}$$



4. ábra. A Maximum Likelihood döntés és az Attack Detection Algorithm összehasonlítása

Ahogy a 4. ábra mutatja, a torzítás javulása (d_{imp}) a Maximum Likelihood döntés esetén nagyobb, mint az Attack Detection Algorithm esetén, ha a korreláció értéke kicsi, viszont ez a különbség nagyon

kicsivé válik nagyobb korreláció esetén. A különbség abból a tényből adódik, hogy a Maximum Likelihood döntés kiaknázza az ismeretét a támadó eltolásának eloszlásával kapcsolatban. Ezért ebben az összehasonlításban, amikoris ez utóbbi eloszlás ismertnek volt feltételezve a Maximum Likelihood döntés számára, akkor ez utóbbi jobban teljesített az Attack Detection Algorithmnél. Mindazonáltal nagyobb korreláció esetén az Attack Detection Algorithm hasonlóan teljesít, mint a Maximum Likelihood döntés, anélkül, hogy támaszkodna erre az extra tudásra.

1.7. TÉZIS: *Az Attack Detection Algorithm a $p_{X_1|X_2}$ és $p_{X_2|X_1}$ feltételes sűrűségfüggvények ismeretét feltételezi, ahol X_1 és X_2 jelölik a kételemű mintában a szenzorok méréseit. Modellezem és analizálom ezen feltételes eloszlások pontatlan ismeretét és megmutatom, hogy nem eredményez jelentős eltérést a torzításban azzal az esettel összehasonlítva, amikor a feltételes sűrűségfüggvények pontosan ismertek, feltéve, hogy a pontatlanság mérsékelt. [J1]*

Feltételezzük most, hogy az Attack Detection Algorithm csak $\hat{p}_{X_1|X_2}(x|y) = p_{X_1|X_2}(x|y) + \Delta(x|y)$ és a hasonló $\hat{p}_{X_2|X_1}(\cdot|\cdot)$ sűrűségfüggvényeket ismeri, ahol $\int_{-\infty}^{\infty} |\Delta(x|y)|\mathrm{d}x < \delta$ minden adott y esetén. Mivel mind $p_{X_1|X_2}(\cdot|\cdot)$, mind $\hat{p}_{X_1|X_2}(\cdot|\cdot)$ sűrűségfüggvények, $\int_{-\infty}^{\infty} \Delta(x|y)\mathrm{d}x = 0$ bármilyen y értékre. A pontatlan tudás szélesebb konfidencia-intervallumot eredményez az 1. algoritmusnál $\hat{b}_1(\cdot)$ és $\hat{b}_2(\cdot)$ alsó és felső határokkal.

Tekintve, hogy $\int_{-\infty}^{\infty} \Delta(x|y)\mathrm{d}x = 0$, Δ-nak vannak pozitív és negatív területei is. Továbbá a pozitív területek integrálja megegyezik a negatív területek abszolútértékének integráljával. A legrosszabb eset (vagyis amikor $|\hat{b}_i(\cdot) - b_i(\cdot)|$ a legnagyobb) akkor következik be, ha az összes pozitív terület $\hat{b}_1(\cdot)$ előtt vagy $\hat{b}_2(\cdot)$ után helyezkedik el, miközben az összes negatív terület $\hat{b}_1(\cdot)$ és $\hat{b}_2(\cdot)$ között van. Egyenletesen gyengítve a konfidencia-intervallum mindkét oldalát azt jelenti, hogy azonos "súlyt" helyezünk $\hat{b}_1(\cdot)$ alá és $\hat{b}_2(\cdot)$ fölé. Ez az

$$\int_{-\infty}^{b_1(z)} p_{X_2|X_1}(u|z)\mathrm{d}u = \frac{\alpha}{2} \tag{16}$$

$$\int_{b_2(z)}^{\infty} p_{X_2|X_1}(u|z)\mathrm{d}u = \frac{\alpha}{2} \tag{17}$$

$$\int_{-\infty}^{b_1(x_2)} p_{X_1|X_2}(u|x_2)\mathrm{d}u = \frac{\alpha}{2} \tag{18}$$

$$\int_{b_2(x_2)}^{\infty} p_{X_1|X_2}(u|x_2)\mathrm{d}u = \frac{\alpha}{2} \tag{19}$$

egyenletek helyett a következő egyenleteket implikálná:

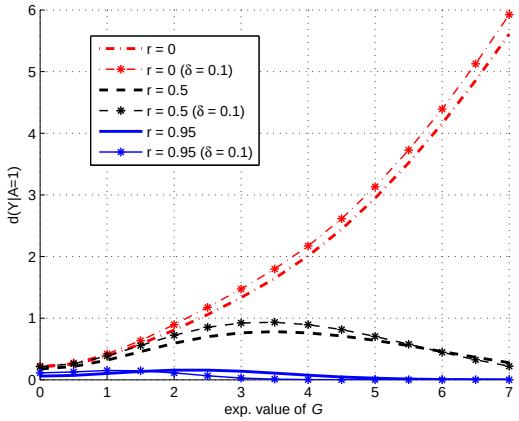
$$\int_{-\infty}^{\hat{b}_1(z)} p_{X_2|X_1}(u|z)\mathrm{d}u = \frac{\alpha}{2} - \frac{\delta}{4} \tag{20}$$

$$\int_{\hat{b}_2(z)}^{\infty} p_{X_2|X_1}(u|z)\mathrm{d}u = \frac{\alpha}{2} - \frac{\delta}{4} \tag{21}$$

$$\int_{-\infty}^{\hat{b}_1(x_2)} p_{X_1|X_2}(u|x_2)\mathrm{d}u = \frac{\alpha}{2} - \frac{\delta}{4} \tag{22}$$

$$\int_{\hat{b}_2(x_2)}^{\infty} p_{X_1|X_2}(u|x_2)\mathrm{d}u = \frac{\alpha}{2} - \frac{\delta}{4} \tag{23}$$

ahol α az elsőfajú hibát jelöli. Ezen formulák felhasználásával kiszámolhatóak az új konfidencia-intervallum $\hat{b}_1(\cdot)$ és $\hat{b}_2(\cdot)$ határai, amikkel aztán kiértékelhető a feltételes sűrűségfüggvények pontatlan ismeretének hatása a torzítás vonatkozásában. (Megyjegyzem, hogy a (20)-(23) képletek implicite felülről korlátozzák δ értékét 2α-val.)



5. ábra. A feltételes sűrűségfüggvények pontatlan ismeretének hatása a torzításra

Az 5. ábra ezen kiértékelés eredményét mutatják $\delta = 0.1$ és $n = 2$ esetén. Ahogy az várható is volt, a feltételes sűrűségfüggvények pontatlan ismerete általában gyengébb támadásdetekciós képességeket eredményez, mindazonáltal, ezek a számolások a legrosszabb esetet (egy speciális Δ konstrukciót) szemléltetik. Az ábra érdekes üzenete az, hogy a konfidencia-intervallum határainak eltolása nem szükségszerűen végződik nagyobb torzításban, ha a minta korrelált.

1.8. TÉZIS: *Az Attack Detection Algorithm arra a feltevésre épít, hogy a mintaelemek közötti korrelációs együttható konstans. Formálisan analizálom a távolságfüggő korreláció esetét Power Exponential korrelációs modellt [BOS01] feltételezve és megmutatom, hogy nem eredményez számottevő különbséget a torzítás tekintetében azzal az esettel összehasonlítva, amikor a korreláció konstans és az értéke 0.95. [J1]*

Mostanáig azt feltételeztem, hogy az r korrelációs együttható értéke azonos minden mérési eredmény párra. A valóságban minden mérési eredmény párnak specifikus korrelációs együtthatója van, amelyik a mérést végző node-ok távolságától és a környezetük néhány fizikai jellemzőjétől függ. A legszélesebb körben alkalmazott korrelációs modell a térbeli statisztikai irodalomban a Power Exponential modell [GGG07, WT93], melyet számos felhasználása mellett [Stu01, VA06, VAA04, AVA04, Rap01, BKK06] én is alkalmaztam.

A nem konstans korreláció hatása analízisének eredményei nagyon érdekesek. A *torzítás javulásá*ra kapott eredmények nagyon hasonlóak a konstans korrelációt feltételező esethez $r = 0.95$ mellett. A torzítás javulása az (15) egyenletben lett definiálva (lsd. 1.6. tézis).

Az 1. táblázat mutatja a numerikus értékek összehasonlítását $t = 2$ esetben.

1. táblázat. Az $r = 0.95$ görbe numerikus értékei a d_{imp} értékekkel összehasonlítva nem konstans korreláció esetén

d_{imp} $r = 0.95$ esetén	d_{imp} r_{ij} esetén
0.0046	0.0048
0.0115	0.0122
0.0342	0.0345
0.0700	0.0716
0.1192	0.1199
0.1823	0.1852
0.2595	0.2741
0.3508	0.3587

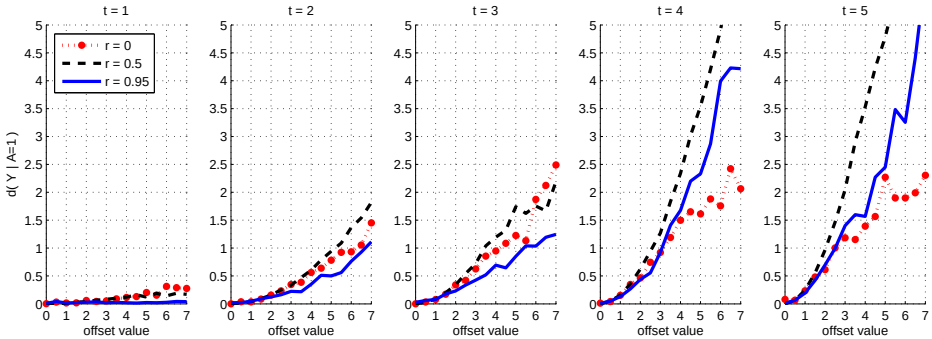
A két esetre kapott d_{imp} értékek között kis különbség világosan mutatja, hogy egyrészt a páronkénti korreláció modellezhető fix korrelációs együtthatóval hosszútávon. Ez pedig megerősíti a korábbi eredményeimet: annak ellenére, hogy egy olyan egyszerűsített sémát alkalmaztam, amelyben a korrelációs együtthatót konstansnak feltételeztem (a 0.95 és a 0.5 leíró értékekkel, valamint a 0 értékkel a független esethez), az eredményeim relevánsak akkor is, ha a realisztikusabb távolságfüggő korrelációs együttható sémáját feltételezzük.

1.9. TÉZIS: *A támadót eddig úgy modelleztem, mint aki eltolást ad néhány szenzor mérési eredményéhez, ahol az eltolás egy független azonos eloszlású valószínűségi változó. Szimulációval megvizsgáltam egy kifinomultabb támadó esetét, amikoris a támadó tetszőlegesen módosíthatja az általa megfigyelt mintaelemeket. Megmutatom, hogy a kifinomult támadó esetén kapott torzítás értékek szoros kapcsolatban állnak az 1.5. tézisben ismertetett analítikus eredményekkel. [J1]*

Feltételezem, hogy a támadó ismeri az Enhanced Data Aggregation Algorithm működését, és egyben a $Det(\cdot,\cdot)$ Attack Detection Algorithm működését is. Továbbá a támadó ismeri a bázisállomás által egy adott lekérdezés által gyűjtött minta méretét, meg tud figyelni néhány mintaelemet, és tetszőlegesen meg tudja változtatni azokat. Viszont a támadó nem ismeri a felezés pontos menetét, amit az n elemből két elemre csökkentésnél használunk.

A kifinomult támadó képes megválasztani a hosszútávon legjobb támadást, miután megbecsüli a minta meg nem figyelt (ismeretlen) elemeit. Ez a következőképpen tehető meg. Először is a támadó analizálja a megfigyelt mintát és ad egy becslést a maradék elemekre (ezt meg tudja tenni, hiszen ismeri a teljes minta méretét). A becslés bármilyen típusú lehet, az alábbi szimulációkban minden ismeretlen elemet az ismert elemek átlagával helyettesítettem. Ezután a támadó képes megvizsgálni az összes lehetséges felezést és ki tudja számolni ezekre a torzítás értékét az eltolás minden lehetséges értékére, hiszen azt a támadó befolyásolja. Megjegyzem, hogy a támadónak nem kell szükségszerűen megtámadnia az összes megfigyelt elemet, hanem képes a megtámadott elemek számát az $[1, t]$ intervallumban megválasztani, ahol t a megfigyelt elemek száma ebben az esetben. A támadó azokat az elemeket választja ki kompromittálásra, amelyek megváltoztatása a legnagyobb torzítást eredményezi átlagosan.

Mivel a támadó nem ismeri a mintafelezés menetét, az átlagosan legnagyobb torzítást az összes lehetséges felezésre kapott torzítások átlagaként tudja kiszámolni (minden felezés azonosan $\frac{1}{2^t}$ valószínűséggel fordul el a $Det(\cdot,\cdot)$ algoritmusban), majd ennek a vektornak a maximumát tekinti.



6. ábra. A kifinomult támadó által okozott torzítás az r korrelációs együttható különböző értékeire

A 6. ábra első három részábráján (30%-nyi kompromittált node-ig, ha $n = 10$) az erősen korrelált mérési eredmények kisebb torzítást eredményeznek, mint a független mérési eredmények. Az utolsó két részábra viszont azt mutatja, hogy egy kifinomult támadó hatása, aki nagyszámú szenzor mérési eredményét képes kompromittálni, jobban kiküszöbölhető, ha a szenzorok mérési eredményei függetlenek. Mindazonáltal a kis korreláció (mint pl. $r = 0.5$) általában gyengítik a javasolt séma képességeit. Egy valósághű támadási forgatókönyv esetén (azaz amikor a támadó csak kisszámú szenzor mérési ered-

ményét képes megváltoztatni) az Enhanced Data Aggregation Algorithm torzítása 2.5σ értékig nőhet kevéssé korrelált és független minták esetében, míg nagyobb korreláció esetén jellemzően 1.2σ alatt marad $\alpha = 0.1$ feltétel mellett.

A kifinomult támadó által okozott torzítás eredmények úgy foglalhatóak össze, hogy azok szoros kapcsolatban állnak az 1.5. tézisben bemutatott analítikus eredményekkel a megfelelő görbék formáját és helyzetét illetően, bár a kifinomult támadó nagyobb torzítás elérésére képes, mint a korábban tárgyalt egyszerűsített támadó.

4.2. RANBAR: RANSAC-alapú Önjavító Aggregálás Szenzorhálózatokban

A környezetet megváltoztató támadás hatásainak kiküszöbölése javasolok egy új önjavító aggregálási eljárást, ami mintaszűrésre épül. Az eljárást RANBAR-nak (RANSAC-based Aggregation) hívják és a RANSAC (Random Sample Consensus [FB81]) elv az alapja, ami jól ismert pl. a számítógépes látás irodalmában [LPT00]. A RANSAC elv egy tanácsot ad arra vonatkozóan, hogy hogyan alkossuk modellt, ha sok a mintában a kompromittált elem. Mindazonáltal ez az elv nem határoz meg egy algoritmust, továbbá egy becslést igényel a támadott elemek számára vonatkozóan, ami általában nem ismert.

2. TÉZISCSOPORT: *Bemutatok és megvizsgálok egy új szenzorhálózati önjavító aggregálási technikát, amit RANBAR-nak hívnak és a RANSAC elvre épül. [B1] [C3] [N1]*

A RANSAC egy elvet definiál az adathalmazzal nem konzisztens elemek kiszűrésére, másszóval kísérleti adatokra történő modellillesztésre. A RANSAC alapelve éppen ellenkezője a hagyományos illesztési technikáknak: ahelyett, hogy a lehető legtöbb elemből felépítünk egy kezdeti modellt, majd megpróbáljuk kiszűrni a modellel nem konzisztens elemeket, a RANSAC a minimális számú elemből épít egy lehetséges modellt és megpróbálja kiegészíteni a kezdeti adathalmazt a konzisztens elemekkel.

2.1. TÉZIS: *Bemutatok egy új önjavító aggregálási algoritmust, a RANBAR-t, amelyik a RANSAC elvet követi és független azonos eloszlású mérési eredményeket feltételez. Továbbá empirikus analízissel meghatározom a RANBAR legjobb kompromisszumos paraméterértékeit azon feltételezés mellett, hogy a szenzorok mérései ismeretlen paraméterű normális eloszlásúak. [B1] [C3] [N1]*

A RANBAR algoritmus működése a következő (lsd. 3. algoritmus).

Algorithm 3 RANBAR Pseudo-Algorithm

1: **while** $Próbálkozások\ száma \leq\ Maximális\ próbálkozás$ **do**
2: Válasszunk ki véletlenszerűen s adatelemet (S)
3: Alkossuk meg az M modellt S alapján
4: Válasszuk ki az összes adatelemet, amelyek egy hibahatáron belül vannak M-hez képest (S^*)
5: **if** #$(S^*) > küszöbérték$ **then**
6: Alkossuk meg az M^* modellt S^* alapján
return
7: **end if**
8: **end while**
9: Számítsuk ki M^*-ot a legnagyobb S^* alapján vagy fejezzük be sikertelenül

A bázisállomás megkapja a támadó által már kompromittált mintát. Ez a minta a RANBAR algoritmus bemenete. Először egy minimális méretű S halmaz kerül kiválasztásra és ez alapján hozzuk létre az előzetes modellt. Az S halmaza mérete s, és az M modell a normális eloszlás $p(x)$ sűrűségfüggvénye $\hat{\theta} = \frac{1}{s}\sum_{i=1}^{s} S_i$ empirikus várható értékkel és $\hat{\sigma}^2 = \frac{1}{s-1}\sum_{i=1}^{s}(S_i - \hat{\theta})^2$ empirikus szórásnégyzettel, ahol S_i jelöli az S halmaz i. elemét.

Algorithm 4 RANBAR konzisztenciavizsgálat

1: **repeat**
2: újraszámoljuk a elemek hisztogramját
3: kiszámoljuk az M model $p(x)$ sűrűségfüggvénye és a minta $h(x)$ hisztogramja közötti távolságot, ahol a távolságot a
$d = \int |h(x) - p(x)|_+ \mathrm{d}x$ egyenlettel definiáljuk, ahol $|x|_+ = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$
4: eldobunk egy elemet a hisztogram azon tartományából, amelyik a legnagyob $|h(x) - p(x)|_+$ értékhez tartozik
5: **until** $d > \epsilon$

A 4. sorban a konzisztenciavizsgálatot a 4. algoritmus végzi.

A 4. algoritmus futása után megmaradó mintaelemek alkotják az S konszenzus halmazát, vagyis S^*-ot. Ha S^* mérete kisebb, mint a q megkívánt méret, akkor az algoritmus újraindul az első lépéstől kezdve, egyébként S^* az aggregátor felé továbbítódik. Az újrapróbálkozások maximális számára van egy f felső korlát. Ha f-nél több iteráció lenne, akkor az algoritmus sikertelenül végződik.

A RANBAR algoritmusnak négy paramétere van, amik eddig nem lettek definiálva. Ezek közül kettőt empirikus úton határoztam meg, mivel az algoritmus bonyolultsága és probabilisztikus természete meggátolta a formális analízist.

A kezdeti halmaz s méretének olyan kicsinek kell lennie, amennyire csak lehetséges a RANSAC elvnek megfelelően. A RANBAR algoritmus esetén a Gauss-eloszlás elméleti hisztogramját kell létrehoznunk. A Gauss-eloszlásnak két paramétere van, a θ várható érték és a σ szórás. A várható értékre már egyetlen elemből is adható egy durva becslés. A szórás becsléséhez legalább két elemre van szükségünk. Ez motiválta az

$$s = 2 \tag{24}$$

választást.

A konszenzus halmaz megkívánt q mérete az algoritmus legfontosabb paramétere. Mindazonáltal a RANSAC elv nem ad tanácsot az értékének korrekt megválasztására. Ha q kicsi, akkor az algoritmus nagyobb eséllyel fut le sikeresen, de az aggregátum a végén nagy mennyiségű támadott mintaelemet tartalmazhat. Ha q túl nagy, akkor az algoritmus nem tud lefutni a konszenzus halmaz méretével kapcsolatos túlzott elvárás miatt. Általában nincs információnk a kompromittált node-ok részarányára vonatkozóan, de megköveteljük, hogy az algoritmus még extrém körülmények között is működjön, azaz amikor csak a minta fele támadatlan. Ezért választottam a

$$q = \frac{n}{2} \tag{25}$$

értéket, ahol n a szenzorok száma.

f értékét empirikus vizsgálattal határoztam meg. Számos f értékre teszteltem az algoritmust és arra jutottam, hogy f választása nem befolyásolja jelentősen a végső aggregátum torzítását, de van egy kompromisszum f értéke és a megfelelő S^* konszenzus halmaz megtalálásának valószínűsége között. Ha f kicsi, akkor nagy a valószínűsége, hogy az algoritmus nem fog megfelelő modellt találni. f növelésével ez a valószínűség csökken, viszont a futási idő nő. Empirikus vizsgálataim szerint az

$$f = 15 \tag{26}$$

választás megfelelőnek tűnik.

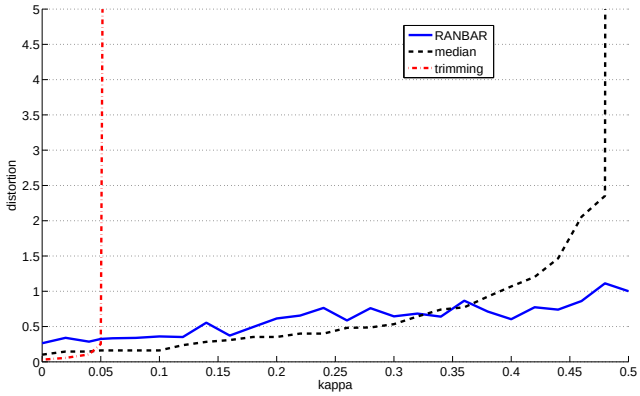
Az ϵ hibatolerancia az algoritmus megállási feltételeként lett definiálva. Az algoritmus ismétlődő fázisa akkor fejeződik be, ha d értéke kisebb lesz, mint ϵ. Ezért ϵ egyfajta pontossági követelménynek is tekinthető. Ha ϵ túl nagy, akkor az algoritmus kimenete messze lehet a támadatlan minta valódi várható értékétől. Ha ϵ túl kicsi, akkor az algoritmus esetleg nem tudja $h(x)$-et $p(x)$-hez hasonlóra formálni,

és emiatt sikertelenül fut le. ϵ megfelelő értékét úgy határozhatjuk meg, hogy teszteljük az értékeit a támadatlan esetben és különböző részarányú támadott elemek esetén is. A tesztesetek során minden szóbajövő ϵ értéket megvizsgáltam néhány tipikus támadási erősség (κ) esetén, és azokat az ϵ értékeket részesítettem előnyben, amelyekre mint a torzítás átlaga, mint a szórásnégyzete kicsi volt. A κ összes értékével kompatibilis ϵ érték az

$$\epsilon = 0.3 \tag{27}$$

2.2. TÉZIS: *Szimulációval megmutatom, hogy a 2.1. tézisben leírt algoritmus letörési pontja (breakdown point) 0.5. Továbbá megmutatom, hogy amennyiben a kompromittált szenzor mérési eredmények részaránya a letörési ponthoz közeli, úgy a 2.1. tézisben leírt algoritmus kisebb torzítást eredményez, mint a medián. Ezen tézis összes szimulációja azon feltételezések mellett készült, hogy a szenzorok mérési eredményei független azonos normális eloszlásúak, és a támadó stratégiája az, hogy minden kompromittált elemet egy tetszőleges közös értékre állít. [B1] [C3] [N1]*

A 7. ábrán összehasonlítottam a Wagner [Wag04] által javasolt önjavító aggregálási mechanizmusokat (a trimminget és a mediánt) a RANBAR-ral a fent említett támadó esetén. A vízszintes tengelyen a különböző támadási erősségek szerepelnek, a függőleges tengelyen pedig a torzítás. A pontvonal azt mutatja, hogy hogyan szerepel ebben az összehasonlításban az 5%-os trimming. Az 5%-os trimming letörési pontja 0.05. Természetesen a trimming levágási szintje felemelhető akár 50%-ra is, de ezzel csökken a módszer pontossága. Emiatt szükség lenne a kompromittált mintaelemek részarányának pontos előrejelzésére, de ez információ általában nem ismert.



7. ábra. A RANBAR, a medián és a trimming összehasonlítása

A medián eredményeit a különböző κ értékekre a pontozott vonal szemlélteti. A 7. ábra tanulsága az, hogy a medián és a RANBAR hasonlóan teljesítenek a torzítás vonatkozásában $\kappa < \frac{1}{3}$ esetén, viszont nagyobb κ értékekre a medián teljesítménye gyorsan romlik, míg a RANBAR eredményei továbbra is megközelítik az eredeti minta igazi átlagát. Ennek az a magyarázata, hogy a medián esetén egy kompromittált elem a helyes értékről annak szomszédjára változtatja a kimenet értékét a mérési eredmények rendezett sorában. Több kompromittált elem esetén az eredmény annyi indexszel lehet arrébb az igazi mediántól, amennyi kompromittált elemet a minta tartalmaz. Egy normális eloszlású mintát feltételezve, amelynél az elemek többsége az igazi medián közelében helyezkedik el, a támadó kénytelen az elemek egyharmadát kompromittálni ahhoz, hogy egy kis torzítást el tudjon érni, de ezen felül már minden további módosított elem jelentősen képes a medián értékét eltolni. Ezzel ellentétben a RANBAR eredményei még κ nagy értékeire sem különböznek a valódi átlagtól. Emlékezzünk vissza, hogy például $\kappa = \frac{1}{3}$ nem azt jelenti, hogy a támadó vezérel mindent a hálózat $\frac{1}{3}$ részében (pl. esetleg nem tudja a

kommunikációs protokollokat megzavarni), de képes arra, hogy a szenzorok $\frac{1}{3}$ részének mérési eredményét megváltoztassa. (Megjegyzem, hogy a 7. ábra nem mutat letörést a RANBAR esetében, hiszen a megfelelő görbe egyszerűen csak véget ér $\kappa = 0.5$ értéknél. Ennek az az oka, hogy a torzítás a RANBAR esetében korlátos egészen eddig a pontig, viszont ezen érték felett az algoritmus nem képes működni a konszenzus halmaz méretére vonatkozó nagy elvárás miatt.)

Amint látható, a RANBAR torzítása $\kappa < 0.5$ esetén mindig korlátos, másszóval a RANBAR letörési pontja 0.5 ezen specifikus támadó esetén. Továbbá a torzítás mindig σ alatt marad, ami azt jelenti, hogy a támadónak erősen korlátozottak a lehetőségei még akkor is, ha akár a szenzorok felét képes kompromittálni.

2.3. TÉZIS: *Analítikusan meghatározom a 2.1. tézisben leírt algoritmus elleni optimális támadást, ahol az optimalitás azt jelenti, hogy a támadó képes tetszőlegesen eltorzítani a 2.1. tézisben leírt algoritmus kimenetét minimális számú kompromittált elemmel. Továbbá analítikusan meghatározom a 2.1. tézisben leírt algoritmus letörési pontját ezen optimális támadás esetén. Ezen tézis levezetései azon feltételezés mellett készültek, hogy a szenzorok mérési eredményei független azonos normális eloszlásúak.*

Először bevezetek néhány fontos fogalmat:

1. Definíció. *Az eredeti eloszlás a támadatlan minta eloszlása.*

2. Definíció. *A céleloszlás az az eloszlás, amit a támadó szeretne, hogy a RANBAR M^*-ként elfogadjon.*

3. Definíció. *A támogató elem egy olyan mintaelem (akár támadott, akár sértetlen), amelyik, ha bekerül S^*-ba, illeszkedik egy adott eloszláshoz.*

Az optimális támadó úgy támad, hogy kitalál egy megfelelő céleloszlást, majd olyan értékűre módosítja a kompromittált elemeket, hogy amennyiben azok bekerülnek S-be, pont a céleloszlás paramétereit adják ki az M modell létrehozása során. Természetesen a támadó nem tudja S megválasztását befolyásolni, de van egy adott nem nulla valószínűsége annak az eseménynek, hogy a céleloszlás megfelelő elemei kerülnek kiválasztásra.

Tekintve, hogy a támadó minimális számú mérési eredményt akar csak megtámadni, fel kell használnia néhány elemet az eredeti mintából a céleloszlás támogató elemeként annak érdekében, hogy képes legyen a konzisztenciavizsgálat utáni minimális mintaméretre vonatkozó követelményt kielégíteni. Ez a minimális elemszám $\frac{n}{2}$, ahol n az eredeti minta mérete. Emiatt a támadónak úgy kell kialakítania a céleloszlását, hogy az átfedjen az eredeti eloszlással, különben az eredeti elemeket nem tudná beilleszteni a céleloszlás alá.

1. Lemma. *Ha a támadó úgy alakítja ki a céleloszlását, hogy annak legalább három tartománya átfed az eredeti eloszlással (figyelembe véve a konfidencia-intervallumnál alkalmazott levágást), akkor a támadó által elérhető torzítás felülről korlátos.*

2. Lemma. *Ha a támadó úgy alakítja ki a céleloszlását, hogy annak maximum egy tartománya fed át az eredeti eloszlással (figyelembe véve a konfidencia-intervallumnál alkalmazott levágást), akkor $\#(S^*) \geq q$ sosem teljesül, ha $\kappa < 0.25$.*

1. Következmény. *Ha a támadó $\kappa = 0.25$-nél kisebb részarányú elem kompromittálása mellett szeretné az optimális támadást végrehajtani, akkor úgy kell kialakítania a céleloszlását, hogy pontosan két tartománya fedjen át az eredeti eloszlással.*

3. Lemma. *A két, az eredeti eloszlással átfedő tartomány megválasztásakor a támadó akkor jár el a*

legjobban, ha a két legnagyobb tartományt választja (vagyis a két tartományt, amelyek legközelebb esnek a céleloszlás várható értékéhez).

4. Definíció. *Az x_{prior} a priori terület egy tartományban az elemek gyakorisága abban a tartományban a 4. algoritmus lefutása előtt, míg az x_{post} a posteriori terület egy tartományban az elemek gyakorisága abban a tartományban a 4. algoritmus lefutása után.*

A legnagyobb tartományok maximális a priori területe kiszámolható a 4. algoritmus normalizálási lépését is figyelembe véve az alábbi módon

$$x_{post} = \frac{nx_{prior} - cut\_num}{rem} \tag{28}$$

ahol cut_num azon elemek száma, amelyek el lettek dobva ebből a tartományból és rem a összes bent maradt elem száma. Ezen számolás eredménye, hogy $x_{prior} \leq 0.42$ mindkét legnagyobb tartományra. Mivel a támadónak képesnek kell lennie azon elemek kompromittálására, amelyek nem férnek bele ebbe a két utóbbi tartományba, a minimum részarány amit kompromittálnia kell $1 - 2 \cdot 0.42 = 0.16$. Ugyanakkor $\kappa = 0.16$ esetén a támadó már képes tetszőlegesen nagy torzítást elérni.

1. Tétel. *A RANBAR elleni optimális támadás feltétele $\kappa = 0.16$.*

Amint az az 1. tételből látszik, a RANBAR letörési pontja optimális támadás esetén 0.16, és függ a tartományok területétől és az ϵ hibatoleranciától.

2. Tétel. *A RANBAR letörési pontja optimális támadás esetén legalább $\frac{1}{2} - \frac{\epsilon}{2} - V$, ahol V a legnagyobb tartomány területe.*

Az algoritmus letörési pontja alapvető változtatások nélkül megnövelhető 0.5-ig, ami az elvi maximum. A letörési pont értéke megemelhető, ha lecsökkentjük a tartományok területeit (azaz több, mint 10 tartományt alkalmazunk) és/vagy lecsökkentjük ϵ értékét. Ha csak a tartományok területeit csökkentjük, akkor a letörési pont értéke 0.35-höz tart, míg ha párhuzamosan ϵ értékét is csökkentjük, akkor a letörési pont határértéke 0.5, hiszen

$$\lim_{\substack{V \to 0 \\ \epsilon \to 0}} \left(\frac{1}{2} - \frac{\epsilon}{2} - V \right) = 0.5 \tag{29}$$

Következésképpen a RANBAR még optimális támadás esetén is képes elérni a 0.5-ös letörési pontot.

4.3. PANEL: Pozíció-alapú Aggregátor Node Választás Szenzorhálózatokban

A szenzorhálózati önjavító aggregálás mellett az aggregátor node választás problémájával is foglalkozom szenzorhálózatokban. Tekintve, hogy a szenzor node-ok gyakran erőforrásaikban erősen korlátozottak, számos technikát javasoltak már a szenzorhálózatok hatékony működésének biztosítására. Az egyik ilyen technika az *aggregálás* vagy *hálózaton belüli feldolgozás (in-network processing)*. Az alapötlet az, hogy ahelyett, hogy továbbítanánk (szinkron alkalmazások esetén), vagy tárolnánk (aszinkron alkalmazások esetén) a szenzorok ”nyers” mérési eredményeit, inkább feldolgozzuk, egyesítjük és tömörítjük azokat egy kitüntetett szenzor node, az *aggregátor* segítségével.

Miközben az aggregátor csomópontok növelik a hálózat hatékonyságát, egyben több erőforrást is használnak fel, mint a többi node. Emiatt szükséges az aggregátor szerepkör időközönkénti átadása más node-nak, így ugyanis jobban eloszlik a terhelés a szenzorok között. Erre a célra aggregátor node választási protokollok használhatóak a szenzorhálózatban, amelyek lehetővé teszik az aggregátor szerepkör dinamikus átruházását.

3. TÉZISCSOPORT: *Bemutatok és analizálok egy új energiahatékony, pozíció-alapú aggregátor node választási protokollt, a PANEL-t, melyet vezetéknélküli szenzorhálózatokhoz fejlesztettem ki. [J2] [C2] [P1]*

A PANEL újszerűsége a többi aggregátor node választó algoritmushoz képest az, hogy támogatja az aszinkron szenzorhálózati alkalmazásokat, melyekben a szenzorok mérési eredményeit a bázisállomás csak késleltetetten kérdezi le. A PANEL kifejlesztésének egyik motivációja az volt, hogy támogassuk a megbízható és folytonos adattárolási alkalmazásokat, pl. a TinyPEDS-et [GWMA06]. A PANEL biztosítja a terheléselosztást, és támogatja az intra- és inter-klaszter csomagküldést lehetővé téve a szenzor–aggregátor, aggregátor–aggregátor, bázisállomás–aggregátor és az aggregátor–bázisállomás kommunikációt.

3.1. TÉZIS: *Bemutatok egy új pozíció-alapú aggregátor node választási protokollt, a PANEL-t. A PANEL gondoskodik a terheléselosztásról abban az értelemben, hogy minden node nagyjából azonos gyakorisággal válik aggregátorrá. Továbbá a PANEL többugrásos adattovábbítási utakat hoz létre a klasztertagok számára az aggregátor node felé. [J2] [C2] [P1]*

Az egyik alapfeltevés, amire a PANEL épít az, hogy a szenzorok statikusak és ismerik a saját geográfiai helyzetüket. A bázisállomásnak nem szükséges statikusnak lennie, lehet mobil is, és elég, ha szórványosan van csak jelen. A szenzorok egy behatárolt területen helyezkednek el, amit geográfiai klaszterekkel partícionálunk. A klaszterezés a hálózat alkalmazásba vétele előtt megtörténik és minden node-ra fel vannak töltve annak a klaszternek a geográfiai információi, amelyikbe a node beletartozik. Az egyszerűség kedvéért feltételezem, hogy az alkalmazási terület négyszög alakú és a klaszterek egyforma méretű négyzetek. A célom minden klaszterben egy aggregátor node kiválasztása. Feltételezem, hogy a hálózat sűrűsége elég nagy ahhoz, hogy az egy klaszterben lévő node-ok összeköttetésben legyenek, ha maximális energiájú átvitelt alkalmaznak. Végül feltételezem, hogy az idő szeletekre van osztva és a node-ok szinkronizáltak abban az értelemben, hogy minden node tudja, mikor kezdődik az új időszelet.

Minden időszelet kezdetén egy $\vec{R}_j$ referenciapont kerül kiszámításra minden j klaszterben minden egyes szenzor által, teljesen elosztott módon. Valójában a referenciapont számolása csak az időszelet számától függ, és minden node függetlenül és helyileg ki tudja számolni. Amint a referenciapont ki lett számítva, a node-ok a klaszterben megválasztják azt a node-ot aggregátornak az adott időszeletre, amelyik *legközelebb esik a referenciaponthoz*.

Az aggregátor node választási procedúrához kommunikáció szükséges a klaszteren belül. A PANEL előnyt kovácsol ebből a kommunikációból azáltal, hogy felhasználja azt az intra-klaszter útvonalak kiépítéséhez. Az aggregátor node választási procedúra végén a node-ok egyben megtanulják, hogy ki a következő ugrás az aktuális aggregátorhoz vezető útvonalon.

A PANEL tartalmaz egy pozíció-alapú útvonalválasztó protokollt is, amelyik az inter-klaszter kommunikáció során használatos. A pozíció-alapú útvonalválasztás egy távoli bázisállomástól vagy egy távoli aggregátor node-tól egy adott klaszter referenciapontja felé küldött csomag továbbítására használható. Amint egy ilyen üzenet belép a klaszterbe, a továbbiakban intra-klaszter útvonalválasztás segítségével továbbítódik az aggregátor felé. Az intra-klaszter útvonalválasztó táblákat a node-ok az aggregátor node választási procedúra közben építik fel.

A PANEL képes támogatni a megbízható és folytonos adattárolási alkalmazásokat, amilyen például a TinyPEDS [GWMA06]. A megbízhatóság azzal érhető el, hogy az aggregátor node-ok által aggregált adatokat átmásoljuk más aggregátor node-okra is (amiket backup aggregátoroknak nevezünk). A PANEL tudja támogatni ezt az átmásolandó üzenetek pozíció-alapú továbbításával a kiválasztott backup klaszter referenciapontja felé, majd intra-klaszter továbbítással az adott klaszter aggregátor node-jához történő eljuttatásig.

A referenciapont kiszámolása egy H álvéletlen függvény meghívását jelenti, amelyik leképzi e-t a $\vec{Q}$ relatív pozícióba a klaszteren belül. Formálisan $H(e) = \vec{Q}$, ahol $\vec{Q} \in (-\Delta d, d+\Delta d) \times (-\Delta d, d+\Delta d)$,

d a klaszter mérete, és $\Delta < 0.5$ egy paraméter amelyről a 3.2. tézisben lesz szó. Amint a referenciapont ki lett számolva, a node-ok az 5. algoritmus szerint indítják az aggregátor node választási eljárást.

Algorithm 5 A PANEL aggregátor node választási pszeudó-kódja

Bemenet:
az algoritmust futtató node id_{self} azonosítója és $\vec{P}_{self}$ pozíciója
az algoritmust futtató node klaszterének $\vec{O}_{self}$ és d paraméterei
a klaszter aktuális $\vec{R}_{self}$ referenciapontja és az időszelet e_{now} száma
az algoritmus T_{elec} futási ideje
Kimenet:
az aggregátor node id_{aggr} azonosítója és $\vec{P}_{aggr}$ pozíciója

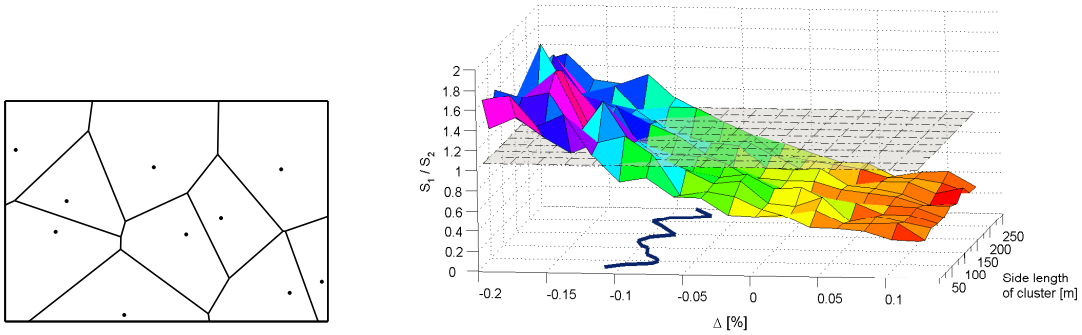
$id_{aggr} = id_{self}$;
$\vec{P}_{aggr} = \vec{P}_{self}$;
$t_0 = T_{elec}$ időzítő beállítása;
$t_1 = f(D(\vec{P}_{self}, \vec{R}_{self}))$ időzítő beállítása;
while t_0 időzítő aktív **do**
  várj amíg t_1 időzítő lejár vagy egy m hirdetmény érkezik;
  case t_1 időzítő lejár:
    sugározd az $[\mathsf{announcement} \mid e_{now} \mid id_{self} \mid \vec{P}_{self}]$ üzenetet maximális energiával;
  case $m = [\mathsf{announcement} \mid e \mid id \mid \vec{P}]$ hirdető üzenet érkezett:
    if az (e, id) pár már korábban is megjött **then** dobd el m-et;
    else if $e \neq e_{now}$ **or** $\vec{P} \notin square(\vec{O}_{self}, d)$ **then** dobd el m-et;
    else if $D(\vec{P}, \vec{R}_{self}) > D(\vec{P}_{aggr}, \vec{R}_{self})$ **then** dobd el m-et;
    else
      $id_{aggr} = id$;
      $\vec{P}_{aggr} = \vec{P}$;
      if t_1 időzítő még aktív **then** állítsd le t_1 időzítőt;
      sugározd m-et maximális energiával;
    end if
end while
kimenet id_{aggr}, $\vec{P}_{aggr}$

Egy előre meghatározott T_{elec} idő után az aggregátor node választási fázis befejeződik és minden node véglegesíti az aggregátor jelöltet az adott időszeletben érvényes aggregátorként.

3.2. TÉZIS: *A 3.1. tézisben bemutatott protokoll működése során a node-ok csak* megközelítőleg *azonos gyakorisággal válnak aggregátorrá. Javasolok egy empirikus megoldást az aggregátor node választás ezen irregularitásának csökkentésére. [J2] [C2]*

Elmagyarázom, hogy miért volt szükség a Δ paraméterre a referenciapont számolásánál a 3.1. tézisben, és megmutatom, hogy hogyan lehet az értékét meghatározni. Annak a valószínűsége, hogy egy adott node aggregátorrá váljon a PANEL-ban, a node Voronoi cellájának méretétől és azon terület méretétől függ, amin belül a referenciapontot megválasztjuk. Terheléselosztási okokból azt szeretném elérni, hogy minden közel azonos valószínűséggel váljon aggregátorrá, emiatt pedig azt szeretném, hogy a node-ok Voronoi cellájának mérete közel azonos legyen.

Tekintsük a 8(a). ábrát az egy klaszterben lévő node-ok Voronoi cellájának illusztrációjaként. Az ábrán megfigyelhető egy ú.n. ”szegélyhatás”, nevezeten azon node-ok, amelyek közelebb vannak a szegélyhez nagyobb Voronoi cellával rendelkeznek, mint azok, amelyek a klaszter közepéhez esnek közel.



(a) A node-ok Voronoi cellái egy klaszterben

(b) Δ értékének meghatározása szimulációval

8. ábra. Voronoi cella illusztráció és Δ megfelelő értékének meghatározása

Ezen jelenség oka egy dimenzióban egyszerűen megmagyarázható. Egy dimenzióban a Voronoi cellák intervallumok a számegyenesen, amelyek hossza a node-ok közötti távolság függvénye, ill. az első és utolsó intervallum esetén a szegélyek függvénye is. Mindazonáltal a node-ok Voronoi cellájának méretét *nem* az egyenletes lehelyezés determinálja, hanem az egyenletes eloszlás sorbarendezettje (order statistics). A szegélyhatás hatékonyan kiküszöbölhető azáltal, hogy szabályozzuk azt a területet, amin belül a referenciapont megválasztásra kerül, hiszen ezzel szabályozhatjuk a szegélyhez közel eső node-ok Voronoi cellájának méretét. A Δ paraméter ennek a beszabályozásnak a mértékét fejezi ki a d eredeti klaszterméret százalékában. Például $\Delta = -0.1$ azt jelenti, hogy a klaszter minden oldalán a szegélyt 10%-kal beljebb húzzuk.

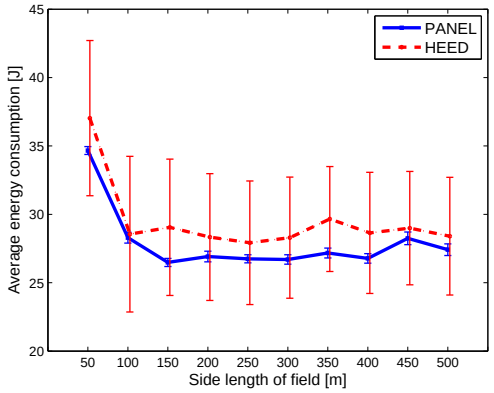
A Voronoi cellák mérete számolásának komplexitása miatt nem könnyű Δ megfelelő értékét analítikusan meghatározni. Ezért a megfelelő érték szimulációval történő meghatározását javasolom. A 8(b). ábra z tengelyén a zárt Voronoi cellák (vagyis azon cellák, amik közel esnek a klaszter középpontjához) átlagos méretének (S_1) és a nyitott Voronoi cellák (vagyis azon cellák, amik határosak a klaszter szegélyével) átlagos méretének (S_2) hányadosa szeperel a Δ paraméter és az oldalhosszúsággal adott klaszterméret függvényében. A $z = 1$ sík felel meg az optimumnak, amikoris a kétféle cellák átlagos mérete megegyezik. Ennek a síknak és a szimulációval kapott felületnek a metszetét levetítettem a $z = 0$ síkra. Ez a levetített görbe adja meg a Δ paraméter optimális értékét a különböző klaszterméretek esetén 10 node-ot feltételezve a klaszterban. Amint látható, az optimális érték általában -0.12 és -0.07 közé esik.

3.3 TÉZIS: *Szimuláció segítségével megvizsgálom a 3.1. tézisben javasolt protokoll energiahatékonyságát a HEED-del [YF04] összehasonlításban, ami egy jól ismert aggregátor node választási megoldás. Megmutatom, hogy a 3.1. tézisben javasolt protokoll energiahatékonyabb a HEED-nél a szenzorok sűrűségétől függetlenül. [J2]*

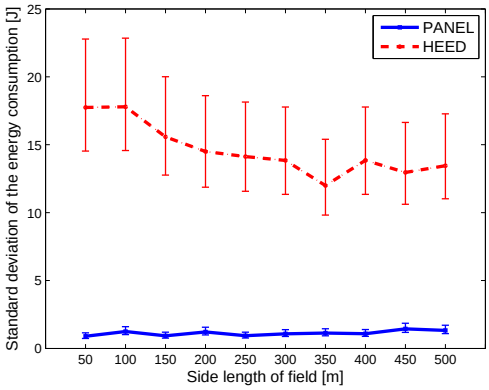
40 node-ot feltételeztem véletlenszerűen elhelyezve egy négyzet alakú területen 4 klaszterben, aholis a szenzorok sűrűsége a terület méretének segítségével szabályozható. A szimulációs forgatókönyveim nem csak a klasztervezérlő (vagy aggregátor node) kiválasztását tartalmazzák, hanem adatüzenetek küldését is. Ahelyett, hogy csak a klasztervezérlő választás energiaszükségletét mérném, a 9(a). ábrán a PANEL és a HEED teljes átlagos energiafogyasztását illusztrálom. A vízszintes tengely a terület méretének felel meg, a függőleges tengely pedig az átlagos energiafogyasztásnak. A folytonos vonal a PANEL értékeit mutatja, míg a szaggatott vonal a HEED-ét. A tüskék a megfelelő értékek 95%-os konfidencia-intervallumát mutatják.

Amint látható, a PANEL kevesebb energiát fogyaszt összességében, mint a HEED, függetlenül a node-ok sűrűségétől. Továbbá a 9(a). ábrán a tüskék azt mutatják, hogy a PANEL energiafogyasztása

pontosabban predikálható, mint a HEED energiafogyasztása, hiszen a PANEL energiafogyasztásának 95%-os konfidencia-intervallumai keskenyebbek, mint a HEED energiafogyasztásáé. Ez utóbbi tulajdonságot a 9(b). ábra is alátámasztja, amelyik szerint a HEED energiafogyasztásának szórása sokkal nagyobb, mint a PANEL-é. (A tüskék a 9(b). ábrán az energiafogyasztás szórásának 95%-os konfidencia-intervallumai.) Hangsúlyozom, hogy a körök száma (vagyis az adatüzenetek mennyisége) erősen befolyásolja az eredményeimet: a körök azok, ahol a PANEL energiahatékonyabb a HEED-nél, ezért a körök számának a jelenlegi 5 fölé emelése még jobb eredményt hozna a PANEL számára a HEED-del összehasonlítva.



(a) Összes átlagos energiafogyasztás a terület méretének függvényében



(b) Az összes átlagos energiafogyasztás szórása a terület méretének függvényében

9. ábra. A PANEL és a HEED összes energiafogyasztásának összehasonlítása

3.4. TÉZIS: *Biztonsági kiterjesztéseket javasolok a 3.1. tézisben bemutatott protokollhoz. Ezek a kiterjesztések segítenek enyhíteni vagy megelőzni egy támadó arra vonatkozó erőfeszítéseit, hogy eltorzítsa az aggregátumot vagy megzavarja az aggregátor node választási procedúrát. [J2] [C2]*

Egy támadónak számos lehetősége van a PANEL működésének megzavarására. Az alábbiakban részletesen tárgyalom ezeket a támadásokat és ellenintézkedéseket javasolok ellenük. Feltételezem, hogy a támadó képes megkaparintani és analizálni egy vagy több node-ot, így a támadó képes e node-okat kontrollálni és tudatában van a szenzorokon tárolt információknak (pl. titkos kulcsoknak, mérési eredményeknek, stb.). Az alábbi javaslataim formális biztonsági analízise nem témája a disszertációmnak, mivel ahhoz szükség lenne egy megfelelő formális modellre, aminek a kifejlesztése akár független kutatás témája is lehet. Mindazonáltal a szokásos hozzáállást követve az alábbiakban megvizsgálom az általános támadásokat.

Az aggregátum eltorzítása a környezet megváltoztatásával vagy a node megkaparintásával: A legkézenfekvőbb támadás a klasztervezérlőnél számolt aggregátum eltorzítását célozza meg. Ennek elérése érdekében a támadó (i) megváltoztathatja a támadott node körül a mért környezeti paraméterek értékeit, vagy (ii) megkaparinthatja a node-ot és tetszőlegesen megváltoztathatja a mért értékeit.

Ellenintézkedések: Mindkét említett támadás hatása kiküszöbölhető statisztikai mintaszűrés alkalmazásával a klasztervezérlőnél (amilyen pl. a RANBAR, lsd. 2. téziscsoport). (Megjegyzem, hogy a kriptográfia nem tud itt a segítségünkre lenni, mivel ezek a támadások kriptográfiai eszközökkel nem detektálhatóak.

Az aggregátum eltorzítása az üzenetek megváltoztatásával: Hogy a támadó elérje a célját (azaz, hogy eltorzítsa az aggregátumot), a támadó (iii) kényszerítheti a megkaparintott szenzort, hogy változtassa meg

a többi node-tól érkező, továbbítandó üzenetek adatmezőjének értékét, vagy (iv) küldhet hamis mérési eredményeket a klasztervezérlőnek más node nevében.

Ellenintézkedések: A megfelelő eszköz ezen támadások ellen a kriptográfiai integritásvédelem ((iii) esetében), és a hitelesítés ((iv) esetében). Ehhez a node-oknak szükségük van például egy nyilvános-titkos kulcspárra és alá kell írniuk az üzeneteiket a titkos kulccsal, továbbá mellékelniük kell a nyilvános kulcsot az üzenethez, miután az alá lett írva. (Megjegyzem, hogy a nyilvános kulcsú kriptográfia feltételezése szenzorhálózatokban nem túlzott [PLP06] szerint.)

A klasztervezérlő választás megzavarása: A támadó megzavarhatja a klasztervezérlő választás folyamatát azzal, hogy olyan node-ok nevében küld hirdető üzenetet, amelyek amúgy nem lennének klasztervezérlők a referenciapont helyzete alapján.

Ellenintézkedések: A node-ok, akik hallják ezt a hamis hirdető üzenetet, ellenőrizni tudják az azon lévő aláírás érvényességét és eldobhatják az érvénytelen aláírású üzeneteket.

A klasztervezérlő folyamat manipulálása klasztervezérlővé válás céljából: Egy másik tipikus támadás aggregátor node választási protokollok ellen az, hogy a támadó úgy manipulálja a protokoll futását, hogy az általa vezérelt node-ok gyakrabban váljanak aggregátorrá, mint ahogy kellene (lsd. például [SWAG07]). Így a támadó könnyebben tud információt gyűjteni a hálózatról, hiszen a node-ok a mérési eredményeiket az aggregátornak küldik. A PANEL esetében egy ilyen támadás úgy vihető véghez, hogy a megkaparintott node hamis pozíció információt küld a hirdető üzenetben az aggregátor node választási fázisban, amiben (i) a helyes azonosítója szerepel, de hamis pozíció információval, vagy (ii) hamis azonosító szerepel hamis pozíció információval. Továbbá, (iii) a támadó lehelyezhet új node-okat is tetszőleges helyekre.

Ellenintézkedések: A PANEL könnyen kiterjeszthető biztonsági eljárásokkal, amik még ezeket a csalásokat is kiküszöbölik. Először is, a bázisállomás használhat nyilvános kulcsú kriptográfiát és aláírhatja a node-ok azonosítóját a titkos kulcsával, és feltöltheti a megfelelő nyilvános kulcsot a szenzorokra azok lehelyezése előtt. Ezen aláírt azonosító segítségével a szenzorok, ha hirdető üzenetet kapnak, ellenőrizni tudják, hogy érvényes azonosítót tartalmaz-e a bázisállomás nyilvános kulcsa segítségével, de a bázisállomáson kívül senki más sem tud új azonosítókat létrehozni. Ennek megfelelően a hirdető üzeneteket ki kell egészíteni az aláírt azonosítóval, és amikor egy node megkap egy

$$\big[\mathsf{announcement} \mid epoch \mid id_{fake} \mid id_{sig_{BS}} \mid pos_{fake} \mid \ldots \\ [\mathsf{announcement} \mid epoch \mid id_{fake} \mid id_{sig_{BS}} \mid pos_{fake}]_{sig} \mid cert\big]$$

hirdető üzenetet, ellenőrizni tudja, hogy $K^P_{BS}(id_{sig_{BS}}) = id_{fake}$, ahol K^P_{BS} a bázisállomás nyilvános kulcsa, és ha nem, akkor eldobhatja az üzenetet. Ezzel kivédtük a (ii) és a (iii) támadásokat. Az (i) támadás úgy védhető ki, hogy engedélyezzük a node-oknak, hogy az útvonalválasztó táblájukban megjegyezzék azon node-ok pozíció információit, amelyektől már kaptak hirdető üzenetet. Ez az információ az időszelet elmúlása után is megőrizhető. Ennek segítségével a node-ok detektálhatják, ha egy megkaparintott vagy elrontott node megpróbálja a különböző időszeletekben különböző helyekre hazudni magát. Ha egy ilyen támadást detektálunk, akkor a detektáló node szétküldhet egy elárasztó figyelmeztető üzenetet, amely tartalmazza a csaló node azonosítóját és a csalás bizonyítékát, vagyis a két hirdető üzenetet azonos azonosítóval és különböző pozíció információval, mindkettőn a csaló node aláírásával.

3.5. TÉZIS: *A 3.1. tézisben leírt protokoll a helyes működéséhez feltételezi, hogy a klaszterben lévő node-ok összekötöttek. Javasolok egy kiterjesztést a 3.1. tézisben leírt protokollhoz azoknak a problémáknak a kiküszöbölésére, amik e feltétel nem teljesülése esetén bontakoznak ki. [J2] [C2]*

A PANEL egy szükségszerű feltételezése az, hogy az egy klaszteren belül lévő node-ok összekötött alhálózatot alkotnak. Ha ez a feltétel nem teljesül és az alhálózat szétszakad, akkor egyes node-ok

nem fogják hallani a a referenciaponthoz legközelebbi node hirdető üzeneteit, és más node-ot fognak aggregátornak választani.

1. Megoldás: Egy lehetséges megoldás annak a területnek a kiszélesítése az aktuális klaszter szegélyein túlra, amelyen belül a hirdető üzeneteket elárasztjuk. Például a hirdető üzenetekkel a szomszédos klasztereket is elárasztathatjuk. Ez megnövelné annak az esélyét, hogy az aktuális klaszterben minden node megkapja a hirdető üzenetet még akkor is, ha az alhálózat partícionált, hiszen ezek a partíciók összekötöttek lehetnek a szomszédos klasztereken keresztül. Ezen megoldás hátránya a node-ok megnövekedett energiafogyasztása.

2. Megoldás: Egy jóval energiahatékonyabb megoldás a következő. Az aggregátor node választás fázisában nem terjesztjük a területet, amiben a hirdető üzenetet elárasztjuk, és elfogadjuk, hogy több klasztervezérlő választódik ki. Ezután a következő protokoll alkalmazását javaslom:

$\mathbf{AN}_i \rightarrow \mathbf{BN}_k$: [backup | id_i | $cluster\_id_i$ | pos_i | $aggregate_i$]
$\mathbf{AN}_j \rightarrow \mathbf{BN}_k$: [backup | id_j | $cluster\_id_i$ | pos_j | $aggregate_j$]
$\mathbf{BN}_k$: detektálja, hogy #(backup messages) > 1 és kiszámolja $final\_aggregate$ értékét
$\mathbf{BN}_k \rightarrow \mathbf{pos}_i$: [correction | id_k | $final\_aggregate$]
$\mathbf{BN}_k \rightarrow \mathbf{pos}_j$: [correction | id_k | $final\_aggregate$]
$\mathbf{BN}_k \rightarrow \mathbf{BS}$: [notification_of_disconnectivity | id_k | $cluster\_id_i$] (opcionális)

ahol $\mathbf{AN}$, $\mathbf{BN}$, és $\mathbf{BS}$ az aggregátor node-ot, a backup node-ot, és a bázisállomást jelölik rendre, és $\mathbf{pos}$ a címzettnél azt mutatja, hogy az üzenetet pozíció-alapú üzenettovábbítással kell küldeni. A $final\_aggregate$ az az érték, amit az aggregátor node-oknak végül tárolniuk kell.

3. Megoldás: Ez utóbbi megoldás hatékony a kommunikációs többletterhelés tekintetében, de néha nem alkalmazható. Például az átlag esetén jól működik, hiszen két részminta átlagának az átlaga megegyezik a két részmintából álló minta átlagával (ha a két részminta azonos méretű), viszont a medián esetében nem működik. Ezért az olyan aggregálási függvények számára, amelyek a teljes mintát igénylik a helyes kimenet kiszámolásához, a következő metódus alkalmazását javaslom:

$\mathbf{AN}_i \rightarrow \mathbf{BN}_k$: [backup | id_i | $cluster\_id_i$ | pos_i | $aggregate_i$]
$\mathbf{AN}_j \rightarrow \mathbf{BN}_k$: [backup | id_j | $cluster\_id_i$ | pos_j | $aggregate_j$]
$\mathbf{BN}_k$: detektálja, hogy #(backup üzenetek) > 1 és kiválasztja a végsõ aggregátort
Ha az i node lett végsõ aggregátornak kiválasztva :
$\mathbf{BN}_k \rightarrow \mathbf{pos}_i$: [notification_of_disconnectivity | id_k | final_aggregator | id_j | pos_j]
$\mathbf{BN}_k \rightarrow \mathbf{pos}_j$: [notification_of_disconnectivity | id_k | not_final_aggregator | id_i | pos_i]
$\mathbf{BN}_k \rightarrow \mathbf{BS}$: [notification_of_disconnectivity | id_k | $cluster\_id_i$] (opcionális)
$\mathbf{AN}_j \rightarrow \mathbf{pos}_i$: [correction | id_j | $measurements_j$]
$\mathbf{AN}_i$: megkapja $measurements_j$ − t and kiszámolja *final_aggregate* értékét
$\mathbf{AN}_i \rightarrow \mathbf{pos}_j$: [correction | id_i | $final\_aggregate$]

Itt a $measurement_j$ a j. node *összes* mérési eredményét jelenti, amit aggregáltatni szeretne.

Ezen technika használatával még a partícionált alhálózatok is konzisztens képet fognak látni a klaszterükről, függetlenül az alkalmazott aggregálási függvénytől. Továbbá a lekérdezések helyes eredményt fognak visszaadni függetlenül a lekérdezett klasztervezérlőtől.

5. Az eredmények alkalmazása

Hogy bebizonyítsam a javasolt megoldásaim alkalmazhatóságát a kisteljesítményű szenzorhálózatokban az önjavító aggregálás és aggregátor node választás témakörében, mind a RANBAR-t, mind a PANEL-t implementáltam TinyOS 2-ben [Tin07], a szenzorok legszélesebb körben használt operációs rendszerében. Mindkét implementáció részét képezi a UbiSec&Sens EU FP6 STReP projekt [EU 08a] végső demonstrációinak. A projekt egyik specifikus alkalmazási területe a szőlőföldek monitorozása. A szőlőföldeken az elterjedten használt monitorozó berendezések a meteorológiai állomások. Ezek magas ára miatt általában csak korlátozott számú állomást alkalmaznak a földeken. Viszont ebben az esetben a mérési eredmények nem tükrözik a valós helyzetet, hiszen egy nagy terület különböző részein különböző mikroklimatikus viszonyok lehetnek. A vezetéknélküli szenzorhálózatok egy kiváló technológia, amely segítségével javíthatunk ezen a helyzeten. A UbiSec&Sens végső demonstrációjában mind a RANBAR, mind a PANEL felhasználásra került ebben a forgatókönyvben a Weingut Georg Naegele szőlőbirtokon [Wei08] 64 szenzorral 4 klaszterben a németországi Neustadtban.

A UbiSec&Sens másik specifikus alkalmazása az országút monitorozása. Az ötlet itt az, hogy telepítsünk szenzorokat az út szélére, amelyek információt gyűjtenek az időjárásról, a forgalomról és az útviszonyokról. Ezt az információt azután feldolgozzuk és felhasználhatjuk a sebességhatár dinamikus szabályozására, az optimális útvonal meghatározására, vagy rendellenes helyzetek detektálására (pl. balesetek, köd, hó) és így tovább. Ezt az eredményt aztán az autóknak elküldve azok figyelmeztetni tudják a vezetőt egy közelgő veszélyre vagy a helyi sebességhatárra. Sőt, egy baleset esetén törvényszéki vizsgálók lekérdezhetik a szenzorhálózatot, hogy információ birtokába jussanak a baleset pillanatában jellemző útviszonyokról. Tekintve, hogy a szenzorok mérési eredményei erősen korreláltak lehetnek egy ilyen kis geográfiai területen, az aggregálás és a hálózaton belüli feldolgozás hasznos lehet a hálózati forgalom csökkentése érdekében. Ezért a demonstációban mind a RANBAR, mind a PANEL szerepelt egy 20 node-os (és egy autóra szerelt node-os) hálózatban 4 klaszterben. A kültéri demonstrációra a németországi Heidelbergben, egy leszállópályán került sor.

Amint az a demonstrációkból látszik, a RANBAR és a PANEL implementációja, még ha nem is teljesen optimalizált, teljes mértékben használhatónak bizonyult mind TelosB [Cro05b], mind MicaZ [Cro05a] node-okon, és még e két különböző fajta node együttműködése is átlátszó volt. A RANBAR és a PANEL forráskódjai az [EU 08a] alatti webcímen megtalálhatóak.

A fenti példákon kívül még rengeteg lehetséges alkalmazási területe van a szenzorhálózatoknak. Nem célom az összes ilyen területe áttekintése, de az alábbiakban adok egy rövid betekintést a valószínűleg legizgalmasabb területekre, amelyekben az általam fejlesztett algoritmusok használhatóak.

A szenzorhálózatok egy ígéretes felhasználási területe az kritikus infrastruktúrák védelme (Critical Infrastructure Protection (CIP)). A CIP célja a sebezhető és összekötött infrastruktúrák védelmének biztosítása, amilyen az energiaellátó rendszer, a bankrendszer, a szükséghelyzeti szolgáltatások, a tömegközlekedési rendszer, vagy akár az Internet. Jó lenne, ha lenne egy autonóm, elosztott, öngyógyító, és egyszerűen telepíthető diagnosztikai rendszer a kritikus infrastruktúrák monitorozására, és az azok ellen irányuló esetleges támadások detektálására és kiküszöbölésére. A szenzorhálózatok felhasználhatóak ilyen célra, különösképpen az 1. és 2. téziscsoportban bemutatott technikák (azaz a CORA és a RANBAR) lehetnek különösképpen fontosak az ilyen alkalmazásokban, hiszen a diagnosztikai eredmények megbízhatósága fölöttébb kívánatos. Sőt, a 3. téziscsoportban javasolt megoldás (a PANEL) is alkalmazható a CIP céljaira adatgyűjtés céljára a biztonsági kiterjesztéseinek köszönhetően. Egy kapcsolódó kutatási projekt a WSAN4CIP EU FP7 STReP [EU 08b], amelyikben a PANEL a biztonságos aggregátor node választási protokollok kifejlesztésének kiindulópontjaként fog szolgálni.

A vadvilág monitorozása szintén elsődleges alkalmazási területe a szenzorhálózatoknak. A természet megfigyelése nagyszámú, hálózatba szervezett szenzorral olyan minőségű és olyan részletes adatgyűjtést tesz lehetővé akár hosszútávon is, amelyet más módszerrel nehéz, ha nem lehetetlen megoldani. A node-ok kommunikációs képessége nem csak az információs és a vezérlőjelek továbbítását teszi

lehetővé a hálózatban, de a node-ok kooperálhatnak is, hogy képesek legyenek komplexebb feladatok elvégzésére, amilyen például a statisztikai mintavételezés, az aggregálás, vagy a rendszer állapotának monitorozása. Ezekhez azonban mind önjavító aggregálási megoldásokra, mint hálózatmenedzsment protokollokra szükség van. Az általam javasolt önjavító aggregálási megoldások alkalmazhatóak itt az előbbi követelmény kielégítésére, míg a PANEL az utóbbi követelmény kielégítését teszi lehetővé. Sőt, mivel a PANEL megbízható a mérések távoli node-okon való tárolása miatt, azt még durva körülmények között is alkalmazni lehet, amilyen például egy sivatag vagy egy esőerdő.

Egy kicsivel általánosabban azt mondhatjuk, hogy a szenzorhálózatok csak egy részét képezik a *mindent átható számítógépesítés elvé*nek (ubiquitous/pervasive computing paradigm). A mindent átható számítógépesítés olyan információfeldolgozást jelent, amelyik teljesen beépül a mindennapi tárgyainkba és aktivitásainkba. A legjobb példa erre valószínűleg az okos otthonok koncepciója. Egy okos otthonban a környezeti vezérlők (amilyen pl. a lámpa, a fűtés, a szellőztetés, stb.) kölcsönhatnak a személyes biometriai monitorainkkal, amiket a ruhánkban hordunk, és ezáltal a szobában a fényerősség, a hőmérséklet és a légkondícionálás a tulajdonos aktuális szükségleteihez igazodik. Egy másik gyakori forgatókönyv olyan hűtőszekrényeket tárgyal, amelyek tudják, hogy mit tárolnak bennük, és képesek a rendelkezésre álló ételekből menüt tervezni, vagy figyelmezteti a felhasználót a nem friss ill. a romlott élelmiszerekre. A mindent átható számítógépesítésnek információgyűjtő megoldásokra van szüksége, hogy hatékonyan mérhessék a döntések alapjául szolgáló paramétereket (pl. a biometriai paramétereket, vagy csak a tejesdobozok számát, stb.) Az én megoldásaim közül a PANEL alkalmazható ezen a területen (lsd. 3. tézis-csoport), tekintve, hogy képes a szenzorok adatait összegyűjteni és kimutatást készíteni azokból.

Az autóközi hálózatok (Vehicular Ad Hoc Network (VANET)) egy másik motiváló felhasználási területe a CORÁ-nak és a RANBAR-nak. A VANET a mobil ad hoc hálózatok egyik formája, amelyik a közeli gépjárművek közötti, valamint a gépjárművek és az útszéli rögzített eszközök közötti kommunikációt teremti meg. A VANET fő célja biztonság és komfort nyújtása az utazóknak. Minden jármű, amelyik fel van szerelve kommunikációs eszközzel egy node lesz az ad hoc hálózatban, és képes lesz fogadni és továbbítani mások üzeneteit a vezetéknélküli hálózatban. A VANET használatának egy példája, hogy a járművek figyelmeztető üzeneteket küldhetnek a kereszteződéseknél, hogy épp megközelítik azt, így a másik irányból érkező járművek vezetői előre tudhatnák, hogy figyelmesen kell megközelíteniük a kereszteződést. Az ilyen és hasonló alkalmazások minden bizonnyal csökkenteni fogják a közlekedési balesetek számát. Tekintve, hogy rengeteg autó van az utakon, az információ mennyisége, amit egy autó megkap nagy lehet, ezért valamilyen aggregálásra szükség van a VANET-ban. Viszont egy támadó komoly baleseteket okozhat hibás helyzetjelentések vagy meghamisított mérések beszúrásával. Az általam javasolt önjavító aggregálási technikák képesek lehetnek az ilyen támadásokat kiszűrni a kompromittált információ elvetésével. Másszóval a CORA és a RANBAR alkalmazhatóak lehetnek VANET-ekben is.

Mindemellett megjegyzem, hogy a disszertációmban bemutatott önjavító aggregálási megoldások nem alkalmazhatóak eseményvezérelt szenzorhálózatokban. Így az észlelési alkalmazások számára (amilyen pl. az erdőtűz észlelése, behatolás észlelése) nem ajánlott a bemutatott eljárások alkalmazása, mivel ezek az alkalmazások extrém értékek megjelenésére koncentrálnak, és a javasolt önjavító aggregálási eljárások egyszerűen kiszűrnék azokat.

Hivatkozások

[AVA04] I. F. Akyildiz, M. C. Vuran, and O. B. Akan. On exploiting spatial and temporal correlation in sensors networks. In *Proceedings of the 2nd Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, 2004.

[BKK06] G. Bravos, A. G. Kanatas, and A. Kalis. Lifetime evaluation and spatial correlation effects on wireless sensor networks. In *Proceedings of 15th IST Mobile & Wireless Communications Summit*, 2006.

[BOS01] J. O. Berger, V. De Oliveira, and B. Sansó. Objective Bayesian analysis of spatially correlated data. *Journal of the American Statistical Association*, 96(456):1361–1374, 2001.

[Cro05a] Crossbow Corporation. MicaZ Datasheet. `http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAZ_Datasheet.pdf`, 2005.

[Cro05b] Crossbow Corporation. TelosB Datasheet. `http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf`, 2005.

[EU 08a] EU FP6 STReP. UbiSec&Sens – Ubiquitous Sensing and Security in the European Homeland. `http://www.ist-ubisecsens.org/`, 2008.

[EU 08b] EU FP7 STReP. WSAN4CIP – Wireless Sensor Networks for the Protection of Critical Infrastructures. `http://www.eurescom.de/activities/EU_Projects/wsan4cip.asp`, 2008.

[FB81] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[GGG07] T. Gneiting, M. Genton, and P. Guttorp. *Geostatistical Space-Time Models, Stationarity, Separability, and Full Symmetry*. Chapman & Hall/CRC, Boca Raton, FL, USA, 2007.

[GWMA06] J. Girao, D. Westhoff, E. Mykletun, and T. Araki. TinyPEDS: Tiny persistent encrypted data storage in asynchronous wireless sensor networks. *Elsevier Ad Hoc Networks*, June 2006.

[LPT00] A. J. Lacey, N. Pinitkarn, and N. A. Thacker. An evaluation of the performance of RANSAC algorithms for stereo camera calibration. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2000.

[Oli05] David J. Olive. *Applied Robust Statistics*. `http://www.math.siu.edu/olive/ol-bookp.htm`, 2005.

[PLP06] K. Piotrowski, P. Langendoerfer, and S. Peter. How public key cryptography influences wireless sensor node lifetime. In *Proceedings of the 4th ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN)*, 2006.

[Rap01] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, Upper Saddle River, NJ, USA, 2001.

[Stu01] G. L. Stuber. *Principles of mobile communication (2nd ed.)*. Kluwer Academic Publishers, Norwell, MA, USA, 2001.

[SWAG07] M. Sirivianos, D. Westhoff, F. Armknecht, and J. Girao. Non-manipulable aggregator node election protocols for wireless sensor networks. In *Proceedings of the International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2007.

[Tin07] TinyOS 2. `http://www.tinyos.net/tinyos-2.x/doc/`, 2007.

[VA06] M. C. Vuran and I. F. Akyildiz. Spatial correlation-based collaborative medium access control in wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 14(2):316–329, 2006.

[VAA04] M. C. Vuran, O. B. Akan, and I. F. Akyildiz. Spatio-temporal correlation: theory and applications for wireless sensor networks. *Elsevier Computer Networks*, 45(3):245–259, 2004.

[Wag04] D. Wagner. Resilient aggregation in sensor networks. In *Proceedings of the 2nd ACM Workshop on Security of Ad hoc and Sensor Networks (SASN)*, 2004.

[Wei08] Weingut Georg Naegele Vineyard. `http://www.naegele-wein.de`, 2008.

[WT93] R. O. Weber and P. Talkner. Some remarks on spatial correlation function models. *Monthly Weather Review*, 121(9):2611–2617, 1993.

[YF04] O. Younis and S. Fahmy. Distributed clustering in ad hoc sensor networks: A hybrid, energy-efficient approach. In *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, March 2004.

Publikációk

Könyvfejezetek

[B1] Buttyán L., Schaffer P., és Vajda I.,
Resilient Aggregation: Statistical Approaches
N. P. Mahalik (szerk.): Sensor Network and Configuration, Springer, 2007.

Folyóiratcikkek

[J1] Buttyán L., Schaffer P., és Vajda I.,
CORA: Correlation-based Resilient Aggregation in Sensor Networks,
Publikálásra elfogadva, Elsevier Ad Hoc Networks, 2008. szeptember.

[J2] Buttyán L., Schaffer P.,
PANEL: Position-based Aggregator Node Election in Wireless Sensor Networks,
Beküldve az International Journal of Distributed Sensor Networks folyóirathoz, 2008. szeptember

[J3] Buttyán L., Holczer T., és Schaffer P.,
Kooperációra ösztönző mechanizmusok többugrásos vezetéknélküli hálózatokban,
Híradástechnika, vol. LIX, no. 3, March 2004, pp. 30–34.

Nemzetközi konferencia/workshop cikkek

[C1] Schaffer P., Vajda I.,
CORA: Correlation-based Resilient Aggregation in Sensor Networks,
10th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2007), Chania, Kréta, Görögország, 2007. október.

[C2] Buttyán L., Schaffer P.,
PANEL: Position-based Aggregator Node Election in Wireless Sensor Networks,
4th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), Pisa, Olaszország, 2007. október.

[C3] Buttyán L., Schaffer P., és Vajda I.,
RANBAR: RANSAC-Based Resilient Aggregation in Sensor Networks,
4th ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN), Alexandria, VA, USA, 2006. október.

[C4] Buttyán L., Schaffer P., és Vajda I.,
Resilient Aggregation with Attack Detection in Sensor Networks,
2nd IEEE Workshop on Sensor Networks and Systems for Pervasive Computing (PerSeNS), Pisa, Olaszország, 2006. március.

[C5] Buttyán L., Holczer T., és Schaffer P.,
Spontaneous Cooperation in Multi-domain Sensor Networks,
2nd European Workshop on Security and Privacy in Ad-hoc and Sensor Networks (ESAS), Springer LNCS 3813, Visegrád, Magyarország, 2005. július.

Nemzeti konferencia/workshop cikkek

[N1] Buttyán L., Schaffer P., és Vajda I.,
RANBAR: RANSAC-Based Resilient Aggregation in Sensor Networks,
HSN Workshop, Balatonkenese, Magyarország, 2006. május.

Poszterek

[P1] Ács G., Schaffer P., és Buttyán L.,
PANEL: Position-based Aggregator Node Election in Wireless Sensor Networks,
HSN Workshop, Balatonkenese, Magyarország, 2008. május.

[P2] Schaffer P., Buttyán L., és Vajda I.,
CORA: Correlation-based Resilient Aggregation in Sensor Networks,
HSN Workshop, Balatonkenese, Magyarország, 2007. május.

Szakdolgozatok

[T1] Schaffer P.,
Spontán kooperáció kialakulásának vizsgálata különböző fennhatóság alá tartozó szenzorhálózatok között – A közös bázisállomás esete,
M.Sc. diplomamunka (konzulens: Dr. Buttyán L.), BME, Budapest, Magyarország, 2005. június.

Egyéb

[O1] Holczer T., Schaffer P.,
Spontán kooperáció kialakulása különböző fennhatóság alá tartozó szenzorhálózatok között,
TDK dolgozat (konzulens: Dr. Buttyán L.), 3. helyezés, Budapest, Magyarország, 2004. november.

Hivatkozások a publikációimra

Az alábbiakban felsorolom a publikációim független hivatkozásait.

- L. Buttyán, P. Schaffer, and I. Vajda, **Resilient Aggregation with Attack Detection in Sensor Networks**, In Proceedings of the 2nd IEEE Workshop on Sensor Networks and Systems for Pervasive Computing (PerSeNS 2006), Pisa, Italy, March 2006.

  cikkre a következők hivatkoznak:

  - ◇ Y. J. Luo, X. Yang, and X. Zhang, **An Effective Resilient Data Aggregation Algorithm in Wireless Sensor Networks**, In Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCom), 2007.
  - ◇ S. Roy, S. Setia, and S. Jajodia, **Attack-Resilient Hierarchical Data Aggregation in Sensor Networks**, In Proceedings of the Fourth ACM Workshop on Security of Ad hoc and Sensor Networks (SASN), 2006.
  - ◇ C. Castelluccia, C. Soriente, **ABBA: A Balls and Bins Approach to Secure Aggregation in WSNs**, In Proceedings of the 6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops (WiOPT), 2008.

- L. Buttyán, P. Schaffer, and I. Vajda, **RANBAR: RANSAC-Based Resilient Aggregation in Sensor Networks**, In Proceedings of the 4th ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN), Alexandria, VA, USA, October 2006.

  cikkre a következők hivatkoznak:

  - ◇ S. Setia, S. Roy, and S. Jajodia, **Secure Data Aggregation in Wireless Sensor Networks**, Book chapter in J. Lopez, J. Zhou (eds.): Wireless Sensor Network Security, IOS Press, 2008.
  - ◇ J.-M. Bohli, A. Hessler, O. Ugus, and D. Westhoff, **A Secure and Resilient WSN Roadside Architecture for Intelligent Transport Systems**, In Proceedings of the First ACM Conference on Wireless Network Security (WiSec), 2008.
  - ◇ B. Sun, L. Osborne, Y. Xiao, and S. Guizani, **Intrusion Detection Techniques in Mobile Ad Hoc and Wireless Sensor Networks**, In IEEE Wireless Communications, 2007.
  - ◇ Y. J. Luo, X. Yang, and X. Zhang, **An Effective Resilient Data Aggregation Algorithm in Wireless Sensor Networks**, In Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCom), 2007.
  - ◇ R. Di Pietro, P. Michiardi, and R. Molva, **Confidentiality and Integrity for Data Aggregation in WSN Using Peer Monitoring**, Eurecom, Research Report, 2007.
  - ◇ E. D. Cristofaro, J.-M. Bohli, and D. Westhoff, **FAIR: Fuzzy-based Aggregation Providing In-network Resilience for Real-time Wireless Sensor Networks**, In Proceedings of the 2nd ACM Conference on Wireless Network Security (WiSec), 2009.

- L. Buttyán, P. Schaffer, **PANEL: Position-based Aggregator Node Election in Wireless Sensor Networks**, In Proceedings of the 4th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), Pisa, Italy, October 2007.

  cikkre a következők hivatkoznak:

- ◇ E. Meshkova, J. Riihijarvi, F. Oldewurtel, C. Jardak, and P. Mahonen, **Service-Oriented Design Methodology for Wireless Sensor Networks: A View through Case Studies**, In Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC), 2008.
- ◇ C. Jardak, E. Osipov, and P. Mahonen, **Distributed Information Storage and Collection for WSNs**, In Proceedings of the Fourth IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS), 2007.
- ◇ E. D. Cristofaro, J.-M. Bohli, and D. Westhoff, **FAIR: Fuzzy-based Aggregation Providing In-network Resilience for Real-time Wireless Sensor Networks**, In Proceedings of the 2nd ACM Conference on Wireless Network Security (WiSec), 2009.

- L. Buttyán, T. Holczer, and P. Schaffer, **Spontaneous Cooperation in Multi-domain Sensor Networks**, In Proceedings of the 2nd European Workshop on Security and Privacy in Ad hoc and Sensor Networks (ESAS), Springer LNCS 3813, Visegrád, Hungary, July 2005.

  cikkre a következők hivatkoznak:

  - ◇ A. G. Forte, H. Schulzrinne, **Cooperation Between Stations in Wireless Networks**, In Proceedings of the IEEE International Conference on Network Protocols (ICNP), 2007.

Az összes ismert független hivatkozások darabszáma 15. A fenti lista nem tartalmazza a projekt dokumentumokban szereplő független hivatkozásokat és az önhivatkozásokat.