

# Objekta izsekošana Bezvadu Sensoru Tīklos: pārskats

Gundars Miežitis, Riga Technical University, Romans Taranovs, Riga Technical University

**Kopsavilkums.** Objektu izsekošana ir viens no Bezvadu Sensoru Tīklu pamata pielietojumiem, ko var izmantot, piemēram, ienaidnieku konstatēšanai vai dzīvnieku kustības izsekošanai. Objekta izsekošana ietver vietas noteikšanu– lokalizāciju– objektam un noteiktās vietas atjaunošanas– izsekošanas– paņēmienus. Raksta mērķis ir izpētīt populārākos izsekošanas paņēmienus un veikt to analīzi. Kā arī iegūt informāciju par iespējamo izsekošanas algoritmu pielietošanas situācijām. Izsekošanas paņēmieni izpēti laikā tiek noteiktas vairākas metrikas, ar kuru palīdzību būtu iespējams salīdzināt izsekošanas paņēmienus.

**Atslēgas vārdi:** Bezvadu Sensoru Tīkls, lokalizācija, izsekošana.

## I. IEVADS

Bezvadu sensoru tīkli (*Wireless Sensor Network* - WSN) ir rīks praktiski jebkuras fizikālas parādības mērīšanai. Šis tīkls sastāv no liela skaita sensoru mezglu ar ierobežotiem resursiem – enerģijas daudzumu, apstrādes jaudu un komunikācijas attālumu. WSN ir paredzēts plašam pielietojumu klāstam, kas atrodas grūti pieejamās, bīstamās vai plašās teritorijās, kā, piemēram, militārā novērošana, negadījumu atklāšana, dabas aizsardzības utt.

Taču viens no svarīgākajiem WSN pielietojumiem ir objektu izsekošana, piemēram, sensori nosaka ienaidnieku atrašanās vietu vai izseko dzīvnieku ieradumiem. Objekta atrašanās vietai ir liela nozīme kustīgu mezglu vai objektu izsekošanai militārajos vai civilajos pielietojumos. WSN galvenais mērķis ir atbrīvot no tīkla topoloģijas plānošanas un līdz ar to no mezglu izvietošanas iepriekš noteiktās atrašanās vietas.

Objekta izsekošana WSN notiek divos etapos: pirmajā tiek noteikta objekta atrašanās vieta tīklā – lokalizācija, bet otrajā šo atrašanās vietu paziņo tīkla noteci jeb datu saņēmējam (tas apkopo datus un, ja nepieciešam, nodod tos tālāk, piemēram, caur Internetu, tīkla administratoram. Šos abus etapus apvieno objekta izsekošanas algoritmi.

## II. LOKALIZĀCIJA

Par lokalizāciju Bezvadu Sensoru Tīklos sauc objekta pozīcijas noteikšanu izmantojot bezvadu sensorus. Lokalizācija dalās divās kategorijās: sīkgraudainā lokalizācijā un rupjgraudainā lokalizācijā [[7]].

### A. SĪKGRAUDAINĀ PIEEJA

Sīkgraudaini algoritmi parasti tiek balstīti uz kādu precīzu informāciju, piemēram, uz attālumu vai leņķi starp diviem mezgliem. Par minimālo pieņemot trīs bāzes mezglus var pielietot trilaterācijas, triangulācijas vai multilaterācijas (ja pieejami vairāk nekā trīs mezgli) pieejas, lai noteiktu objekta relatīvo atrašanās vietu attiecībā pret trim bāzes mezgliem.

Lai realizētu sīkgraudainu lokalizāciju sensoru mezglam nepieciešama speciāla aparatūra un bieži arī papildus resursi.

Kas noved pie paaugstinātas aparatūras izmaksas, augstu enerģijas patēriņu un lielākus fizikālos izmērus.

**Saņemtā Signāla Stiprums** (*Radio Signal Strength* - RSS) ir viena no visvienkāršākajām pieejām, ko izmanto attāluma noteikšanai starp diviem mezgliem. RSS nosaka, piemēram, kādai ziņojumu paketei, ko sūta otrs sensors. Tāpēc nav nepieciešama speciāla ziņojumu sūtīšana starp sensoriem, kas papildus noslogotu tīklu un novestu pie lielāka enerģijas patēriņa. Kā piemēru var minēt RADAR [[8]] sistēmu, kur izmanto signāla stiprumu, lai aptuveni noteiktu mobilu mezglu atrašanās vietu iekštelpās.

Izmantojot vienādojumu (1) un Saņemtā Signāla Stiprumu iespējams aprēķināt attālumu starp diviem sensoriem.

$$d = \alpha \sqrt{\frac{cP_{tx}}{P_{rcvd}}} \quad (1)$$

, kur  $P_{rcvd}$  – saņemtā signāla stiprumu,  $P_{tx}$  – pārraides jauda,  $\alpha$  – ceļa zuduma koeficients,  $c$  – izmantotā signāla ātrums,  $d$  – attālumu;

**Ierašanās Laiks** (*Time of Arrival* – ToA) ierasti lieto attāluma informācijas iegūšanai ar signāla izplatīšanās laika palīdzību un ir atkarīgs no laika atšķirības, lai aprēķinātu sensoru mezgla atrašanās vietu. ToA ir nepieciešama augsta takts izšķirtspēja, lai iegūtu precīzu atrašanās vietu un sinhronizācija starp mezgliem. ToA parasti izmanto signālus, kas nepārvietojas ātri, piemēram, ultraskaņu. Ultraskaņas signālu sūta raidītājs saņēmējam; un saņēmējs atbild arī ar ultraskaņas signālu. Pēc šī divvirzienu rokasspiediena, raidītājs var izsecināt attālumu ( $D$ ) no signāla turp un atpakaļ brauciena aizkaves, pēc vienādojuma (2).

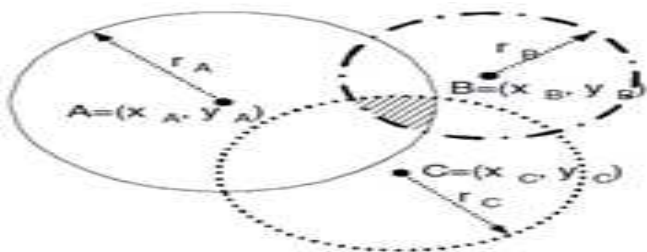
$$d = \frac{((T_3 - T_0) - (T_2 - T_1))V}{2} \quad (2)$$

, kur  $T_0$  – laiks, kad pirmais sensors raida ultraskaņas signālu,  $T_1$  – laiks, kad otrais sensors saņem ultraskaņas signālu,  $T_2$  – laiks, kad otrais sensors raida ultraskaņas signālu,  $T_3$  – laiks, kad pirmais sensors saņem atbildes signālu,  $V$  ir ultraskaņas signāla ātrums.

**Ierašanās Laika Atšķirība** (*Time Difference of Arrival* – TDoA). Attālumu starp mezgliem var noteikt daudz precīzāk nosakot laika atšķirību starp lēni ceļojošiem signālu, piemēram, ultraskaņas vai akustisko viļņu, pārraidi un uztveršanu. Pārraidot gan radio, gan ultraskaņas/akustiskos viļņus vienlaicīgi, uztverošajam mezglam ir pārraides atskaites laiks pēc, kura var aprēķināt laika atšķirību. Trūkums ir papildus aparatūras nepieciešamība signālu apstrādei.

TABULA 1  
LOKALIZĀCIJAS PAŅĒMIENU KOPSAVILKUMS [[6]]

Lokalizācijas paņēmieni	Priekšrocības un trūkumi	Pielietojums
ToA	Nepieciešama augsta sinhronizācija starp mezgliem	Ierasti izmanto šūnu tīklos
TDoA	Nepieciešama augsta sinhronizācija starp nekustīgajiem mezgliem, bet nav nepieciešama sinhronizācija starp kustīgajiem un nekustīgajiem mezgliem tīklā	Ierasti Bezvadu Sensoru Tīklos
AoA	Nepieciešama papildus aparatūra (antenu masīvs), kas noved pie augstākām resursu izmaksām un mezgla fiziskajiem izmēriem	Piemerotāk izmantot nekustīgajos mezglos, nekā kustīgajos dēļ izmēra.
RSS	Nepieciešams precīzs signāla izplatīšanās modelis uzticamu rezultātu iegūšanai. Ir zemas izmaksas, jo izmantojot RX iespējams noteikt RSS. Kustīgā mezgla mobilitāte un kanālu piesārņotība var radīt kļūdas.	Tā kā RSS ir zema precizitāte, to izmanto lietojumos, kam nepieciešama aptuvena attāluma noteikšana.
Rakstu Salīdzināšana	Nepieciešams <i>off-line</i> apmācības posms, lai iegūtu datubāzi. Kā arī šī datu bāze var būt neuzticama, ja ar laiku mainās kanāls vai vide.	Visbiežāk izmanto Lokālajos bezvadu tīklos kopā ar RSS, kā datubāzes metriku. Apsvērs arī šūnu tīkliem.



1. att. Trilaterācija

$$d = ((T_3 - T_1) - (T_2 - T_0)) \frac{V_{RF} * V_{US}}{V_{RF} - V_{US}} \quad (3)$$

, kur  $V_{RF}$  un  $V_{US}$  ir radio signāla un ultraskaņas ceļošanas ātrums atbilstoši,  $T_0$  – laiks, kad pirmais sensors nosūta radiosignālu,  $T_1$  – laiks, kad otrais sensors uztver radio signālu,  $T_2$  – laiks, kad pirmais sensors nosūta ultraskaņas signālu,  $T_3$  – laiks, kad otrais sensors uztver ultraskaņas signālu. TDoA papildus kļūdu var izraisīt apstrādes laiks, saņēmējam ir arī jāzina precīzs  $T_2 - T_0$  laiks, lai noteiktu attālumu.

**Ierašanās Leņķis** (*Angle of Arrival – AoA*). Tā vietā, lai balstītos uz attālumu starp divām ierīcēm, atrašanās vietu var noteikt arī pēc ziņojuma ierašanās leņķa. Paņēmieni, lai iegūtu ierašanās leņķi, ietver RF vai ultraskaņas uztvērēju masīvu izmantošanu vai izmantojot rotējošu virziena bāku.

Kad iegūti attālumi starp vairākiem mezgliem visvienkāršākais paņēmiens, ka iegūt objekta atrašanās vietu ir izmantot **trilaterāciju** (Sk. Att.1). Vai arī alternatīvi paņēmieni, ja iespējams izmantot vairāk kā trīs mezglus ir izmantot **multilaterāciju**, vai arī, ja izmanto AoA, pielietot **triangulāciju**.

Objekta atrašanās vietu izmantojot trilaterāciju iegūst atrisinot vienādojumu (4):

$$\sigma_{x,y} = \left| \sqrt{(x - x_A)^2 + (y - y_A)^2} - r_A \right| + \left| \sqrt{(x - x_B)^2 + (y - y_B)^2} - r_B \right| + \left| \sqrt{(x - x_C)^2 + (y - y_C)^2} - r_C \right| \quad (4)$$

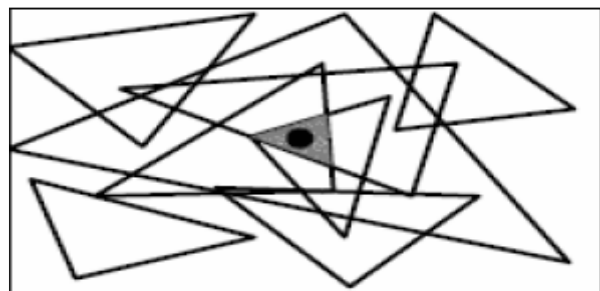
, kur  $x$  un  $y$  – objekta nosakāmās koordinātes,  $x_A, y_A$  – bāzes sensora A atrašanās vieta,  $x_B, y_B$  – bāzes sensora B atrašanās vieta,  $x_C, y_C$  – bāzes sensora C atrašanās vieta,  $r_A, r_B, r_C$  – attiecīgi attālums starp sensoriem A, B un C un objekta atrašanās vietu

#### B. RUPJGRAUDAINĀ PIEEJA

Pretstatā iepriekšējām metodēm, rupjgraudainās lokalizācijas pieejas pievēršas minimālas informācijas izmantošanai atrašanās vietas noteikšanā. Ierastas pieejas:

**Tuvums:** vienkāršākā no visām vietas noteikšanas metodēm ir tuvuma noteikšana – tas ir vienkāršs lēmums par to, kur atrodas mezgls radio uztveres zonā starp mezgliem. Lēmumu var pieņemt balstoties uz to vai datu paketes vispār tiek saņemtas vai balstoties uz kādu sliekšņa pieeju, kurai nepieciešama nepārtraukta savienojamība kādā laikā. Šīs pieejas labs piemērs ir RFID piespraudes izmantošana mezglu lokalizācijā.

**Centrs:** Katrs mezgls aprēķina tā atrašanās vietu aprēķinot visu mezglu, ko tas dzird centrus. Ja mezgli ir precīzi novietoti, iespējama neliela kļūda.



2. att. APIT paņēmieni

**Tuvuma Punkts Trijstūrī** (*Approximity Point in Triangle – APIT*). Šajā paņēmienā definē reģionus formējot trijstūru

kopu, kas ir nezināmā mezgla atrašanās vietas tuvumā. Nezināmo atrašanās vietu var noteikt, kā trijstūru pārklāšanās reģionus (Sk. Att. 2) [[9]].

### C. RAKSTU SALĪDZINĀŠANA

Cits paņēmieni atrašanās vietas noteikšanai ir raktu salīdzināšana (*Pattern Matching*). Tā vietā, lai noteiktu attālumu starp bāku un mezglu, šajā pieejā cenšas salīdzināt saņemto signālu ar jau iepriekš nomērtu vērtību, kas tiek glabāta datu bāzē. Tādējādi šo metodi pazīst arī kā *pirkstu nospiedumu (Finger Print)* pieeju. Pamata doma ir tāda, ka signāla stiprums saņemts konkrētā punktā nav nemainīgs. Tas parasti mainās, tāpēc labāk būtu modelēt Saņemta Signāla Stiprumu ar gadījuma vērtībām. Tas īpaši apstiprinās iekšējās.

Galvenā ideja ir salīdzināt saņemto signālu ar to, kas ir datubāzē un noteikt varbūtību, ka ierīce atrodas konkrētajā pozīcijā. Ierastam risinājumam ir divas fāzes:

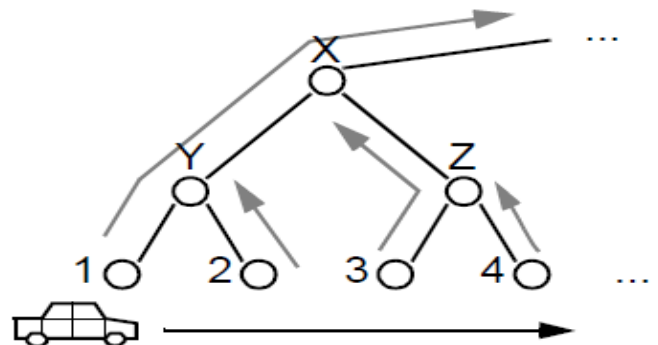
*Off\_line* fāze: Šī fāzes mērķis ir apkopot signālus no visām bāzes stacijām katrā treniņa atrašanās vietā. Vispirms tiek izlemts par treniņa atrašanās vietu skaitu. Tad ieraksta Saņemto Signāla Stiprumu (bāzes stacijai, kurai tas ir pārāk tālu, signāla stiprumu uzskata par vienādu ar nulli). Katram ierakstam datu bāze ir forma:  $(x,y,\{ss_1,ss_2,\dots,ss_n\})$ , kur  $(x,y)$  ir treniņa vietas koordinātes un  $ss_i, i=1..n$ , ir signāla stiprums, ko saņem no  $i$ -tās bāzes stacijas. Šos ierakstus glabā datu bāzē. Jāpiezīmē, ka augstākai precizitātei var saglabāt vairākus ierakstus vienai un tai pašai pozīcijai.

*Real-time* fāze: Ar labu apmācītu pozicionēšanas modeli var noteikt objekta atrašanās vietu apkopojot signāla stiprumu no visām bāzes stacijām. Pozicionēšanas modelis var noteikt vairākas atrašanās vietas, katru ar noteiktu varbūtību. Kaut arī parasti izvada pozīciju ar vislielāko varbūtību.

## III. POZĪCIJAS IZSEKOŠANA

Ierasti izsekošanas paņēmieni izmanto centralizētu pieeju. Palielinoties mezglu skaitam tīklā palielinās arī ziņojumu skaits, kuras nodod notecei aizņemot arvien lielāku joslas platumu. Tādējādi šī pieeja nav kļūdu noturīga, jo ir viens iespējamo kļūdu punkts, turklāt tie ir slikti mērogojami. Turklāt tradicionālajos paņēmienos jušanas uzdevumus veic viens mezgls vienā laika momentā, kas noved pie zemākas precizitātes un lielākas aprēķinu noslodzes šim mezglam. WSN katram mezglam ir ierobežots enerģijas avots un līdz ar to tradicionāli izsekošanas paņēmieni, kas balstīti uz sarežģītu signālu apstrādi nav piemēroti WSN [[10]].

Reālos WSN mezglu skaits ir liels un mezgliem ir jārealizē vairāk-lēcienu komunikācija, lai sasniegtu gala lietotāju, kā arī komunicētu savā starpā. Bet daudzos lietojumos, īpaši militārajos un katastrofu atklāšanas, nepieciešama reāla laika (vai tuvu tam) datu pārraide lietotājam. Taču tajā pašā laikā jāievēro tas, ka sensoriem ir ierobežoti resursi, no kuriem vissvarīgākais ir enerģija. Tādējādi objekta pozīcijas izsekošanas algoritmiem jācenšas minimizēt ziņojumu aizkavi un samazināt enerģijas patēriņu.



3. att. STUN piemērs

Pozīcijas izsekošanas algoritmi var sagrupēt trīs lielās algoritmu klasēs [[5]]:

- Koku balstīti algoritmi
- Klasteru balstīti algoritmi
- Paredzēšanas balstīti algoritmi

### A. KOKU BALSTĪTI ALGORITMI

Šie paņēmieni organizē tīklu hierarhiskā kokā. Alternatīvi, sensoru tīklu var atspoguļot kā grafu, kurā sensoru mezgli ir virsotnes un saites ir saites starp mezgliem. Koks var tikt veidots statistiski (viens, koks visa tīkla darbības laikā) vai dinamiski (mainoties objekta pozīcijai vai parādoties jauniem mezgliem vai beidzot darboties kādam mezglam, koks tiek veidots no jauna).

## STUN

STUN (Mērogojama Izsekošana Izmantojot Tīklā savienotus Sensorus – *Scalable Tracking Using Networked sensor*) balstās uz novērtējumu, ka objekti ko vēlas izsekot pārvietojas paredzami, proti, eksistē dabīgi ierobežojumi, piemēram, to ātrums, un tādējādi maksimālais attālums, ko tie var veikt noteiktā laikā ir ierobežots. Pieeja izmanto hierarhiju, lai pierakstītu informāciju par objekta klātbūtni; tā kā kustības attālums ir ierobežots, tad informācijas atjaunošana visbiežāk jāveic mezgliem hierarhijas apakšā. Kā rezultātā sistēma var izsekot vairākus objektus vienlaicīgi. Izmantotajā hierarhijas kokā par sakni pieņem *vaicājuma punktu* – noteci. Koka lapas ir sensori un pārējās virsotnes mezgli ir komunikācijas mezgli – starpmezgli. Informāciju par konstatētajiem objektiem glabā starpmezglos; konkrēti katrs starpmezgls glabā konstatēto objektu kopu, ko, sadarbojoties ar hierarhijā zemāk esošiem mezgliem, ir konstatējuši. Lai šī kopa būtu svaiga lapas mezgli sūta konstatācijas ziņojumus saknes virzienā. Bet ziņojumam obligāti nav jāsasniedz sakne; starpmezgls nodos ziņojumu tālāk tikai tad, ja ir mainīta tā konstatēto objektu kopa. Pretējā gadījuma ziņojums tiek iznīcināts. Tādā veidā tiek samazināts lieks enerģijas patēriņš, kas rodas lieku ziņojumu sūtīšanas dēļ.

Att.3 redzams piemērs, kurā katra koka lapa konstatē braucošas mašīnas ierašanos. Ziņojums no sensora 1 iedarbina Y un tā priekštečus pievienot mašīnu to konstatēto objektu kopai. Sekojošie ziņojumi no citiem sensoriem nenoved pie X konstatēto objektu kopas izmaiņšanos un tādējādi X nepadod ziņojumu tālāk.

Galvenais iemesls konstatēto objektu kopas uzturēšanai ir atļaut efektīvu vaicāšanu. Vaicājumu maršrutē no hierarhijas

saknes pa vienu ceļu uz sensoru, kas ziņoja par pamanīto objektu. Bez konstatēto objektu kopas vaicājums būtu jānosūta visām lapām [[2]].

**Priekšrocības:** Var veikt vairāku objektu izsekošanu vienlaicīgi. Izmanto energoefektīvu objekta pozīcijas atjaunošanas paņēmieni.

**Trūkumi:** Koks tiek veidots statistiski, tādējādi ir nenoturīgs pret kļūdām. Izveidotais koks neatspoguļo sensoru izvietojumu dabā.

## DAB

DAB (Notece un Balasts – *Drain and Balance*) ir metode vēlamu ziņojumu atzarošanās hierarhijas koku izveidei, t.i., kokus, kuriem ir zemas komunikācijas izmaksas un vaicājumu aizkave.

DAB koku veido no apakšas uz augšu, t.i., no lapām uz saknes vairākos DAB soļos. Katrā DAB solī sensoru apakškopa tiek apvienota sabalansētos apakškokos, neatkarīgi no notikumu biežuma. Metodes efektivitāte ir rūpīgi izvēloties mezglus, ko sapludināt katrā no soļiem izmantojot notikumu biežuma informāciju. Sensorus sagrupē izmantojot vienu vai vairākus notikumu biežumu sliekšņus – *nosusināšanas* sliekšnis, turklāt, vispirms sapludina apakškopas ar biežāk konstatētajiem notikumiem.

Piemērs: 3 DAB soļu koka konstruēšana 1D sensoru grafam, kur svarus atspoguļo diagramma un nosusināšanas sliekšņi – 6,3,0 ( $h_1, h_2, h_3$ ) – Att. 4-a. Secīgi Att. 4- b, c, d norāda koka izveidi 1., 2. un 3. DAB solī. Att.4-d redzams beigu ziņojumu izplatīšanas hierarhijas koks.

Kā redzams sensoru ar augstākām izmaksām apstrāde ātrāk, ļauj novietot to savienojošos mezglus tuvāk lapām hierarhijā. Turpretī, sensori ar mazākajām izmaksām tiek apstrādāti pēdējie, tādējādi to savienojšie mezgli ir izkaisīti starp lieliem reģioniem iepriekšējos soļos izveidotajos apakškokos.

Taču soļu skaita un nosusināšanas sliekšņa izvēle var ievērojami ietekmēt izveidotā koka kvalitāti. Viena galējība, ja koku konstruē vienā solī – nevar izšķirt augstas noslodzes un zemas noslodzes reģionus un tādējādi izveidotais koks neņem vērā noslodzes informāciju. Šāds koks būs balansēts un būs ar mazu augstumu, jo nav tukšumu starp sensoriem kokā. Tāda koka komunikācijas izmaksas būs augstas.

Bet otra galējība ir, ja koku konstruē ar pārmērīgi lielu soļu skaitu. Ir tieksme būt diezgan ierobežotai sensoru izvēlei, ko sapludināt katrā solī, un tādējādi ir ļoti maz iespējas izcelt koka sabalansētību. Šajā gadījumā beigu koka formu nosaka neatklāto sensoru atrašanās vieta katrā solī. Un šādi koki būs ierasti ir ar garām saitēm un daudz slāņiem.

DAB metode var nepalīdzēt tādiem trafika rakstiem, kas uzrāda mazu lokalitāti. Piemēram, ļoti regulāriem trafika rakstiem – nemainīgiem vai strauji pieaugoši kustības frekvencei 1D reģionā [[2]].

**Priekšrocība:** Izveidotais koks vienmērīgi sabalansē sensorus ar atšķirīgu noslodzi izveidotajā hierarhijā.

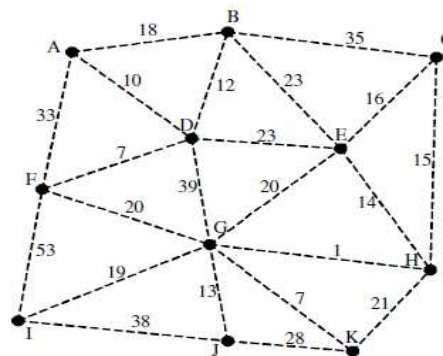
**Trūkumi:** Regulāru trafiku gadījumos DAB nevar izveidot labāko ziņojumu nodošanas koku. Koks neatspoguļo sensoru izvietojumu dabā.

## DAT un Z-DAT

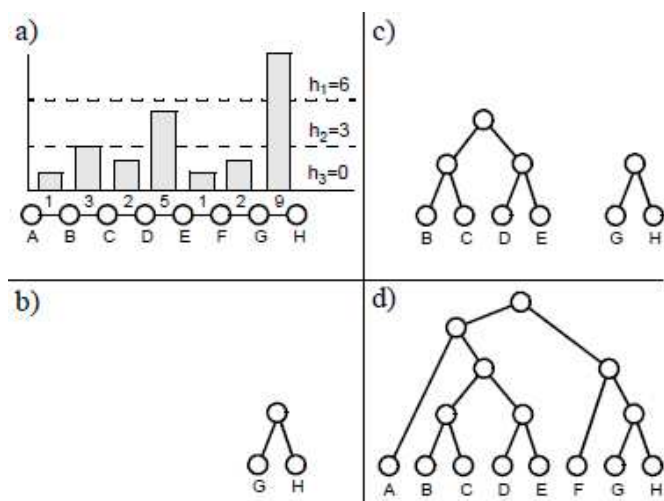
DAT (Novirzes-Izvairīšanās koks – *Deviation-Avoidance Tree*) var uzskatīt par DAB paplašinājumu, objekta atrašanās vietas atjaunošanas ziņā. Šajā metodē tiek ņemta vērā tīkla fizikālā topoloģija (Sk. Att. 5) un tiek izmantota divu stadiju pieeja. Pirmajā stadijā ievēro vairākus principus, proti, nobīdes-izvairīšanās un augstākais-svars-pirmais, lai izveidotu objekta izsekošanas koku (Sk. Att. 6), kas samazinātu atrašanās vieta atjaunošanas komunikācijas izmaksas. Z-DAT (Reģiona-balstīts DAT – *Zone-based-DAT*) pieeja jušanas zonas cenšas sadalīt kvadrāta-veida zonās un rekursīvā veidā kombinē kokā.  $w_t$  – saites svars starp diviem kokā blakus esošajiem mezgliem [[4]].

**Priekšrocības:** Koks izmanto sensoru reālo izvietojumu dabā, tādējādi samazinot kopējās komunikācijas izmaksas.

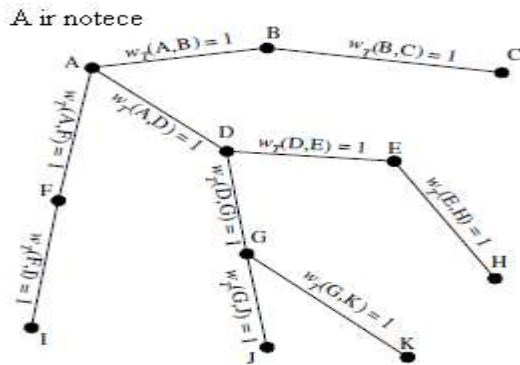
**Trūkums:** Noslogo mezglus, kas tuvāki notecei.



5. att. DAT tīkls ar svariem



4. att. DAB koka veidošanas piemērs



6. att. Izveidotais DAT koks

### B. KLASTERU BALSTĪTI ALGORITMI

Šie paņēmieni WSN sadala klasteros, kas sastāv no klasteru galvām (KG) un kaimiņu sensoriem. Ierasti, klasterus veido statiski, tīkla izvietojanas laikā un klastera atribūti – klastera izmērs, pārklājamā teritorija, klastera locekļu skaits – ir statiski. Bet šādai pieejai var būt vairāki trūkumi:

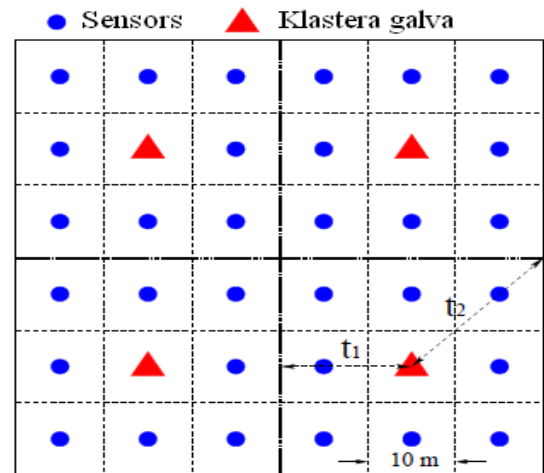
- klasteru statistiskums nenodrošina robustumu. Piemēram, ja KG beidz darboties, visi klastera locekļi kļūst nederīgi turpmākam darbam;
- neļauj sensoriem dažādos klasteros sadarboties un dalīties ar informāciju;
- nevar pielāgoties augstas dinamiskas scenārijiem, kad sensoram augstas aktivitātes reģionā liek palikt nomodā;

Bet alternatīva ir veidot klasterus dinamiski, kad klastera veidošanu, piemēram, izraisa objekta ierašanās. Sensors ar pietiekamu baterijas jaudu un aprēķinu jaudu konstatē objektu, tas kļūst par KG. Tā kā var būt vairāki šādi augstākas jaudas mezgli, kas konstatē notikumu nepieciešams mehānisms, kas atstāj tikai vienu KG. Sensori, kas atrodas KG tuvumā tiek „aicināti” kļūt par klastera locekļiem. Šādā veidā klasteris tiek veidoti tikai biežu notikumu zonās. Turklāt sensori nepieder tikai vienam klasterim dažādos laika posmos [[1]].

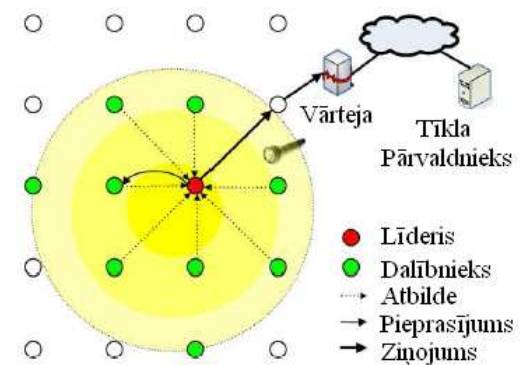
### DCA

DCA (Dinamiskas Klasterizācijas Algoritms – *Dynamic Clustering Algorithm*) ir decentralizēts, viegla svara, dinamiskas klasterizācijas algoritms viena objekta izsekošanai. Algoritms pievēršas akustiskai mērķa izsekošanai. Sensori akustiskajā izsekošanas sistēmā veic

divu veidu uzdevumus: jūt signāla enerģijas līmeņus un analizē, klasificē skaņu un veic datu sapludināšanu. Tā kā otrajam uzdevumam nepieciešama augstāka veikspēja tad DCA izmanto statistisku mugurkaulu, kas veidots no reti izvietotiem augstas-veikspējas sensoriem, kas pieņem KG lomu, un dažādos blīvumos izvietoti zemā-gala sensori, kuru funkcija ir nodrošināt jušanas informāciju, kad KG to pieprasa. Klasteri tiek veidoti, kad KG tiek aktivizēta pēc pieprasījuma, kad konstatētais akustiskā signāla stiprums pārsniedz kādu sliekšni. Tad aktīvais KG nosūta lūguma ziņojumu apkārtējiem sensoriem pievienoties klasterim un nodrošināt to ar jušanas informāciju. Akustiskajā izsekošanā enerģijas-balstītu lokalizācijas mehānismu. DCA izmanto Att. 7 norādīto sensoru izvietojumu.



7. att. DCA izmantotā klasteru organizācija



8. att. DELTA novērotai objekts un ziņojumu nodošana

DCA izmanto Voronoi diagrammas, lai realizētu varbūtības balstītu līdera pieteikšanās procedūru un sensoru atbildes metodi. Proti, visi sensori uzstāda savu relatīvo atrašanās vietu attiecībā pret kaimiņu, tīkla izvietojanas laikā. Tad ar Voronoi diagrammu palīdzību KG (vai sensors) var aprēķināt un sakārtot tabulā varbūtību, ka konkrētais attālums starp mērķi un objektu, KG ir vistuvāk mērķim. Šo informāciju izmanto, lai uzstādītu atpakaļ skaitošo taimeri, ko KG izmanto, lai paziņotu savu vēlmi aktivizēties. Ja neviena cita KG nepiesakās aktivizēties, tad KG kļūst aktīva, pretējā gadījumā taimeri atvieno. [[1]]

**Priekšrocības:** Klasteri veido tikai objekta apkārtnē, tādējādi tiek nodrošināta energoefektivitāte.

**Trūkumi:** var izsekot tikai viena objekta pozīcijai;

### DELTA

DELTA (Dalīts Notikumu Lokalizācijas un Izsekošanas Algoritms – *Distributed Event Localization and Tracking Algorithm*). DELTA ir dalīts objekta izsekošanas algoritms, kas izvairās no augstas datu noslodzes uz bāzes staciju.

Turklāt, DELTA izmanto sensoru gaismas mērījumu KG izvēlēšanā un lokalizācijā. DELTA uztur mezglu grupas, kas ir dinamiski izveidotas ap objektiem, tik līdz šie objekti parādās (Sk. Att. 8). Mērījumu balstīts ievēlēšanas algoritms nosaka grupas līderi, kas ir atbildīgs par grupas uzturēšanu, datu apkopošanu un apstrādei, kā arī ziņošanai bāzes mezglam. Bāzes stacija ir savienota ar Internetu un var iedarbināt signalizāciju, kad tiek novērota neparasta situācija.

DELTA mezgliem nepieciešams zināt atrašanās vietu, ko var nodrošināt ar GPS vai kādu citu lokalizācijas paņēmieni [[11]].

**Priekšrocība:** Pārlietu nenoslogo bāzes staciju.

**Trūkumi:** Var izsekot objektus, kas pārvietojas ar nemainīgu ātrumu.

### C. PAREDZĒŠANAS BALSTĪTI ALGORITMI

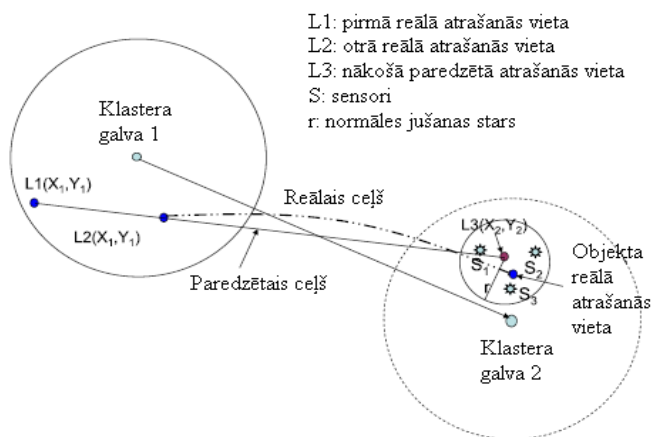
Šīs metodes tiek veidotas balstoties uz koku vai klasteru metodēm, kurām pievieno paredzēšanas modeļus. Modeļi paļaujas uz heuristikām, kas balstīta uz kādu no sekojošajiem pieņēmumiem: 1) kustībā esošais objekts saglabās pašreizējo kustības virzienu un ātrumu vēl dažas sekundes; 2) objekta ātrumu un virzienu var secināt no objekta kustības vēstures; 3) dažādos posmos var piešķirt dažādus svarus balstoties uz vēsturi.

### DPT

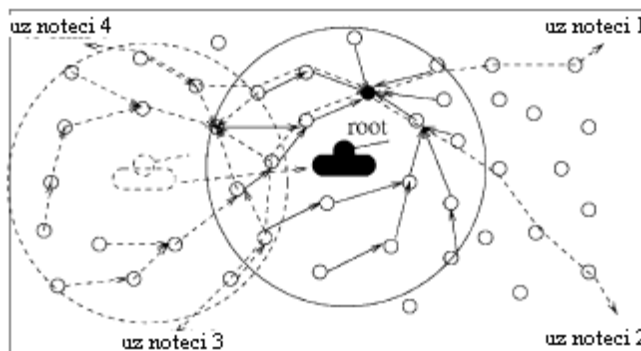
DPT (*Distributed Predictive Tracking algorithm*) ir dalīts (nav nepieciešams centralizēts datu kontroles punkts – nodrošina robustumu), mērogojams paredzēšanā balstīts algoritms, kas precīzi seko mobilam objektam izmantojot WSN. Šis algoritms izmanto klasteru balstītu arhitektūru, lai būtu mērogojams un robusts. Konstatējot objektu protokols nodrošina dalītu mehānismu, kas vietēji nodrošina optimālo sensoru kopu, kas piemēroti uzdevuma veikšanai. Un tikai pēc tam šie mezgli tiek iedarbināti minimizējot patērēto enerģiju izsekošanai.

Papildus tam DPT izmanto paredzēšanas mehānismu, lai brīdinātu klasteru galvas par objekta tuvošanos. Balstoties uz šiem paredzējumiem klasteru galvas aktivizē vispiemērotākos mezglus uzdevuma veikšanai tieši pirms objekta ierašanās. Tādējādi tiek samazināta varbūtība, ka tiks pazaudēts objekts laikā, kad vairums mezglu ir miega stāvoklī. Taču protokols nodrošina arī efektīvus kļūdu labošanas paņēmienus, kas var atjaunot pazaudēta objekta atrašanās vietu (kas radušies gan paredzēšanas, gan mezglu atteikumu dēļ) ar minimālu enerģijas patēriņu.

Enerģijas patēriņš un arī atvieglojot aparatūras un programmatūras prasības tiek samazināts pateicoties vienkāršākiem aprēķiniem un komunikācijai starp sensoriem vai klasteriem.



9. att. DPT paredzēšanas piemērs



10. att. DCTC kokā veidošanas piemērs pārvietojoties objektam

DPT izšķir robežas sensorus, sensorus, kas atrodas nelielā attālumā no robežas un tie, kas nav robežas sensoru. Kamēr robežas sensoriem jādarbojas visu laiku, lai varētu konstatēt objekta ierašanos, ne-robežas mezgliem sensoru atrodas miega stāvoklī līdz brīdim, kad to aktivizē tā klastera galva. Tik līdz kā konstatē objektu, tiek izpildīta uzdevumu secība: „jušana-paredzēšana-komunikācija-jušana” dalītā veidā uz sensoriem, kas atrodas gar objekta ceļu.

Katru objektu identificē ar Mērķa Deskriptoru (MD), kas sastāv no: 1)Mērķa piederības; 2)Mērķa pašreizējās atrašanās vietas; 3)Mērķa nākošo paredzēto atrašanās vietu; 4)Laika zīmogu;

Pielietota tiek lineārā paredzēšana, kas izmanto tikai divas iepriekšējās atrašanās vietas, lai lineāri paredzētu trešo atrašanās vietu (Sk. Att. 9).

Kļūda tiek konstatēta, ja nākošā paredzētā klastera galva nav apstiprinājis objekta ierašanos klasterī. Enerģijas taupīšana tiek realizēta tā, ka sensoru lielāko daļu laika ir miega režīmā un tos aktivizē tikai klastera galva. Bet, ja nav neviena paredzējuma tad lielai daļai sensoru ir jāpaliek nomodā [[5]].

**Priekšrocības:** Efektīvi kļūdu novēršanas paņēmieni ar minimālu enerģijas patēriņu. Vienkāršs un efektīvs paredzēšanas modelis – lineārā paredzēšana. Nodrošina mērogojamību un robustumu.

**Trūkumi:** Paredzēšanas dēļ var rasties kļūdas, kas rada lieku enerģijas patēriņu. Ja nav paredzējumu rodas lieks enerģijas patēriņš, jo sensoriem jāpaliek nomodā un jāgaida objekts.

### DPR

DPR (Duālas Paredzēšanas-balstīts Ziņošanas mehānisms – *Dual Prediction-based Reporting mechanism*) algoritmā gan sensoru mezgli, gan bāzes stacijas veic identiskus paredzējumus par objekta kustību balstoties uz tā pārvietošanās vēsturi. Novērojot objektu, sensoru mezgli var pateikt vai paredzējumi ir vai nav bijuši pareizi un atspoguļo reālo objekta atrašanās vietu. Ja paredzējums sakrīt ar reālo objekta atrašanās vietu nav nepieciešams sūtīt atjauninājumus bāzes stacijai, kas arī ir pareizi paredzējusi objekta pārvietošanos. Pretējā gadījumā sensoru mezgli ir atbildīgi par neprecīzās paredzēšanas labošanu bāzes stacijā [[1]].

Viselastīgākais lokalizācijas modelis ir režģis (viens no bezattāluma objekta pozīcijas noteikšanas paņēmieniem). Kopējais režģu skaits nosaka granularitāti

**Priekšrocības.** Ja sensoru mezgli un bāzes stacija precīzi paredz visas objekta nākotnes atrašanās vietas, nav

nepieciešama komunikācija, tādējādi tiek nodrošināta augstāka energoefektivitāte.

**Trūkumi.** Algoritmam ir jāpaļaujas uz paredzēšanas mehānisma precizitāti.

## DCTC

DCTC (Dinamiska Konvoja Koka-balstīta sadarbšanās – *Dynamic Convoy Tree-based Collaboration*) izmanto mobilu objektu izsekošanai un apkārtējās vides uzraudzīšanai. DCTC balstās uz koka struktūru, ko sauc par *konvoja koku*, kas ietver sensoru mezglus ap izsekojamo objektu un koku dinamiski maina pievienojot un atvienojot mezglus kokam, pārvietojoties objektam. Att. 10 attēlo kā tiek izmantots konvoja koks, lai izsekotu objektu. Kā objekts ierodas sensoru laukā tie sensori, kas var konstatēt objektu sadarbības savā starpā, lai izveidotu sakni un sākotnējo konvoja koku. Paļaujoties uz konvoja koku sakne savāc informāciju no sensoru mezgliem un apstrādā to, lai iegūtu precīzāku informāciju par objektu. Objektam pārvietojoties dažī mezgli kokā kļūst pārāk tāli, lai veiktu izsekošanu un tie atvienoti no koka. Tā kā lielākā daļa mezglu paliek miega stāvoklī tad saknei vajadzētu paredzēt objekta kustības virzienu un iedarbināt konkrētos sensoru mezglus. Kamēr konvoja koks tiek mainīts arī sakni ir nepieciešams mainīt, lai minimizētu komunikācijas izmaksas [[3]].

**Priekšrocības:** Koks izmanto sensoru reālo izvietojumu dabā. Izveidotais koks nav statisks.

**Trūkumi:** Mezgli, kas atrodas tuvāk notecēm tiek vairāk noslogoti, jo tiem ir jānodod vairāk ziņojumu. Papildus ziņojumi, lai pamodinātu sensorus, kuriem tuvojas objekts.

## IV. IZSEKOŠANAS ALGORITMU NOVĒRTĒŠANA

Izsekošanas algoritmu izpētes laikā izcēlās vairāki algoritmu raksturojošie lielumi ar, kuriem varētu veikt analītisku algoritmu salīdzināšanu. Šos raksturlielumus literatūrā sauc par metrikām [[10]].

Taču izmantotās metrikas objektu izsekošanas algoritmu analīzē atšķiras atkarībā no algoritma uzdevumiem. Tādām metrikām kā klasteru veidošanai, izsekošanas precizitātei un ceļa atjaunošanas ātruma, izsekojamo objektu skaitam ir vistiešākā ietekme attiecībā pret enerģijas patēriņu bezvadu sensoru tīklos. Ja ir augsta komunikācija starp mezgliem klastera veidošanas laikā, tiek patērēta papildus enerģija. Augsta izsekošanas precizitāte (arī reāla laika izsekošana) novedīs pie papildus enerģijas patēriņa. Līdzīgi arī ar ceļa atjaunošanas ātrumu palielinās enerģijas patēriņu. Tādējādi var secināt, ka jebkura cita rādītāja uzlabošana novedīs pie lielāka enerģijas patēriņa un tāpēc arī energoefektivitāte var tikt uzskatīta par svarīgāko no metrikām.

Citas metrikas objektu izsekošanas algoritmos var būt kļūdu biežums, objekta virzienu maiņu skaits, zaudējumu laiks, objekta mainīgais ātrums, atlikusī sensoru mezglu enerģija, attālums starp mezglu un objektu, tīkla mūža ilgums, kopējais tīkla enerģijas patēriņš, kopējais ziņojumu virstēriņš, laiks pirms pirmais mezgls beidz darboties un attālumu mērījumu biežums.

Bet kopējā enerģijas patēriņa analīze objektu izsekošanās ir vissvarīgākā metrika izsekošanā izmantoto protokolu klāstā bezvadu sensoru tīklos, jo mezgli ir ierobežoti ar pieejamo enerģijas daudzumu. Enerģijas patēriņu var sadalīt trīs

galvenajās komponentēs: radio komunikācija, jušana un datu apstrāde. Tomēr, bieži ir tā, ka vairumā pētījumu kopējais enerģijas patēriņš neatbilst reālajam enerģijas patēriņam, jo netiek ņemti vērā visi trīs faktori.

## V. SECINĀJUMI

Raksta galvenais mērķis ir aplūkota vienu no Bezvadu Sensoru Tīklu pielietojumiem – objekta izsekošanu, ko var sadalīt divos etapos. Pirmais ir objekta atrašanās vietas noteikšana – lokalizācija un otrs ir objekta pozīcijas izsekošana. Rakstā aplūkoti gan izplatītākie lokalizācijas paņēmieni, gan objekta izsekošanas paņēmieni uzsvāru pievēršot izsekošanas algoritmiem.

Vissvarīgākais paņēmieni izsekošanas algoritmu novērtēšanai ir energoefektivitāte, jo, piemēram, ja grib uzlabot izsekošanas precizitāti vai ātrumu tas tieši ietekmē enerģijas patēriņu uz sensoru mezgliem un, kā jau tika minēts, sensoru mezgliem pieejamā enerģija ir ierobežota. [[12]] tiek piedāvāti kritēriji energoefektīvai WSN izveidošanai, ko arī varētu ņemt vērā izstrādājot energoefektīvu izsekošanas algoritmu.

Koku balstīti algoritmi veicina efektīvu datu apkopošanu un datu agregāciju. Bet, ja koks ir izveidots statistiski, tad algoritms ir nenoturīgs pret kļūdām un izveidotais koks nav mērogojams vai arī to realizēt ir ļoti grūti. Tādējādi koku balstīti algoritmi ir piemēroti pielietojumiem, kur energoefektivitāte ir svarīga, bet svarīgāk ir precīzi izsekot objekta kustību ar augstāku objekta ceļa atjaunošanas ātrumu un, kur mezglu atceses gadījumos mezglus ir iespējam vieglāk nomainīt.

Klasteru balstīti algoritmi izmanto vairākus mezglus, lai iegūtu lielāku precizitāti. Kā arī samazina ziņojumu vairākkārtēju nosūtīšanu un nodrošina enerģijas taupīšanu. Tie nodrošina arī mērogojamību, robustumu un labāk izmanto joslas platumu. Ja klastera galvu izvēlas lokālā tīkla apstrādes veidā samazinās pārsūtīto ziņojumu skaits uz bāzes mezglu, kā arī nodrošina lielāku drošību. Līdz ar to klasteru balstīti algoritmi piemēroti pielietojumiem, kam energoefektivitāte ir svarīga un tai pašā laikā ir svarīga precīza objekta atrašanās informācija, kā arī gadījumos, kad izvietotos mezglus ir grūti apkalpot vai nomainīt.

Paredzēšanas balstīti algoritmi var minimizēt sekošanā izmantojamo mezglu skaitu. Bet šajos algoritmos var rasties neliels apjoms kļūdu un kā arī rodas latentums. Līdz ar to paredzēšanas balstīti algoritmi piemēroti pielietojumiem, kam vissvarīgākā ir energoefektivitāte, bet, kas var pieļaut zināmu neprecizitāti objekta pozīcijā vai nelielu aizkavi objekta pozīcijas noteikšanai un, līdzīgi kā klasteru algoritmos, var izmantot grūti pieejamās mezglu izvietojuma vietas.

Taču apskats nav pilnīga, jo algoritmu aprakstos tiek prezentētas atšķirīgas tīklu situācijas – vide, mezglu izvietojums, utt.. Lai algoritmus varētu precīzāk salīdzināt būt nepieciešams radīt vienādus apstākļus visiem algoritmiem un izvēlēties svarīgākās metrikas to salīdzināšanai. Neskatoties uz to izdevās izvīzīt ieteicamās algoritmu klašu pielietojumu situācijas.

## IZMANTOTĀ LITERATŪRA

- [1] Y.Xu, J.Winter un W-C. Lee, „Dual Prediction-based Reporting for Object Tracking Sensor Networks,” Proceeding of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004.

- [2] H.T.Kung, un D.Vlah, „Efficient Location Tracking Using Sensor Networks”, Proceeding of 2003 IEEE Wireless Communication and Networking Conference, 2003.
- [3] W.Zhang un G.Cao, „DCTC: Dynamic Convoy Tree-Based Collaboration for Target Tracking in Sensor Networks”, IEEE Transactions on Wireless communications, Vol.3, No5, Septembris 2004
- [4] C-Y. Lin, W-C Peng, un Y-C. Tseng, „Efficient In-Network Moving Object Tracking in Wireless Sensor Networks”, IEEE Transactions on Mobile Computing, Vol.5, Issue, p 1044-1056, 2006
- [5] C-T. Lee, „Object Tracking in Wireless Sensor Networks”
- [6] I.Gutvenc, „A Survey on TOA Based Wireless Localization and NLOS Mitigation Techniques”, IEEE Communication Survey & Tutorials, Vol. 11, No. 3, Trešasi ceturksnis 2009
- [7] T.A. Alhmied un S-H. Yang, „A Survey: Localisation and Tracking Mobile Targets Trough Wireless Sensor Networ”.
- [8] P. Bahl and V. Padmanabhan, “RADAR: An In-Building RF-Based User Location and Tracking System,” Proceedings, vol. 2, pp. 775-784, 2000.
- [9] Tian He, Chengdu Huang, Brian M. Blum, John A. Stankovic, Tarek Abdelzaher. Range-free Localization Schemes for Large Scale Sensor Networks. MobiCom 2003.
- [10] S. Bhati un J. Xu, „Survey of Target Tracking Protocols using Wireless Sensor Network, Fifth International Conference on Wireless and Mobile Communications, 2009
- [11] M. Walchli, P. Skoczylas, M. Meer un T.Braun, „Distributed event Localization and tracking with wireless sensors”.
- [12] Q. Wang, M.Hempstead un W. Yang, „A Realistic Power Consumption Model for Wireless Sensor Network Devices”.

**Gundars Miežitis** was born in year 1987. Bachelor degree in computer science (Riga Technical university, 2010). Now masters' student in Riga Technical university, Institute of computer control, automation and computer technics, Department of computer networks and system technology. Email: gundars\_m@inbox.lv

**Romans Taranovs**. Master degree in computer science. Now phd student in Riga Technical university, Institute of computer control, automation and computer technics, Department of computer networks and system technology. Email: romans.taranovs@rtu.lv

#### **Gundars Miežitis, Romans Taranovs. Object tracking in Wireless Sensor Network: survey**

In this paper is overlooked one of Wireless Sensor Network applications - object tracking, which consists of localization, and object tracking algorithms. Localization is used to determine objects location in 2D or 3D space trough use of trilateration – where are needed three base nodes, with known location and distances from each node to object, triangulation – where are needed three base nodes and incoming signals angle from object, or multilateration – where are needed more than three base nodes with known location and distances from object. Object tracking algorithms task is to track objects location, and sends location information to the end user of the network. Depending on the algorithm it is plausible to track location energo-efficiently – when some of the nodes are in sleep mode, but other are tracking object, or non-energo-efficiently – where all nodes are tracking object whether they can see object or not. Tracking algorithms were classified into three classes - tree, cluster and prediction – based algorithms. Main goal was to look at the most common algorithms in each class to be able to deduce in which situations to use the algorithms from appropriate class, and, furthermore, suggest criteria that could be taken into account for algorithm comparison. In paper were used researches of recent year researches on wireless sensor networks. Work is concluded with main metrics which could be used to better compare algorithms, but in order to do that it is necessary to ensure a uniform start position, e.g. an equal number of nodes, the same environment in which WSN nodes operate, energy sources, etc. Acquired results could be used to develop new tracking algorithm or improve an existing algorithm that would not have the reviewed algorithms negative aspects.

#### **Гундарс Миезитис, Роман Таранов. Слежения за объектами в беспроводных сенсорных сетях: обзор**

В настоящей работе рассмотрено применение беспроводных сенсорных сетевых –при слежения за объектом. Локализация используется для определения место нахождениями объектов в 2D или 3D просранстве при использования трилатерации - где необходимы три базовые узла, с известном место нахождениями и расстояниями от каждого узла до объекта, триангуляция - где необходимы три базовые узла и угл входящих сигналов от объекта, или мультилатерация - где необходимо боле трех базовых узлов с известными место нахождениями и расстояниями от объекта. Задача алгоритмам объекто отслеживании, отслеживать местоположение объектов, и отправляет информацию о местоположении конечным пользователем сети. В зависимости от алгоритма это правдоподобно отслеживать местоположение энерго-эффективно - когда некоторые узлы находятся в спящем режиме, но другие отслеживают объекта, или не-энерго-эффективно – когда все узлы отслеживают объекта, могут ли они увидеть объект или нет. Алгоритмы слежения были разделены на три класса - алгоритмы на основе дерева, кластеров и прогнозирования. Основная идея было рассмотреть наиболее распространенные алгоритмы в каждом классе, чтобы определить в каких ситуациях наиболее удобно использовать алгоритм каждого класса и предложить критерии, которых можно использовать для сравнения алгоритмов. В работе использованы исследования последних лет в области беспроводных сенсорных сетей. Для того, чтобы лучше сравнить алгоритмы, необходимо обеспечить единые исходные положения, то есть одинаковое количество узлов, среду, в которой работают, источники энергии, и так далее. Полученные результаты могут быть использованы для разработки новых алгоритмов отслеживания или улучшить существующий алгоритм, которые бы не недоучестили негативных аспектах рассмотренных алгоритмов.