ŠIAULIŲ UNIVERSITETAS
TECHNOLOGIJOS FAKULTETAS
ELEKTRONIKOS KATEDRA

Kęstutis Kvietkauskas

BEVIELIO TINKLO KOLIZIJOS
Magistro darbas

Vadovas
doc. dr. G.Daunys

ŠIAULIAI, 2008

ŠIAULIŲ UNIVERSITETAS
TECHNOLOGIJOS FAKULTETAS
ELEKTRONIKOS KATEDRA

TVIRTINU
Katedros vedėjas
doc. dr. G.Daunys
2008 06

BEVIELIO TINKLO KOLIZIJOS
Magistro darbas

Konsultantas

Vadovas
doc. dr. G.Daunys
2008 06

Atliko
RM-6 gr. stud.
K. Kvietkauskas
2008 06

Recenzentas
ŠU Technologijos fakulteto Elektronikos katedros lektorius
dr. N. Ramanauskas
2008 06

ŠIAULIAI, 2008

SUMMARY

Kvietkauskas K. Collisions in Wireless Networks: Master thesis of technology of signals/research advisor Assoc. doc. dr. G. Daunys; Šiauliai University, Technological Faculty, Electronics Department. – Šiauliai, 2008. – 58p.

By developing the technologies, approximately eight ears ago a new way of network where found and patented – wireless network. After seeing the advantages of these networks, they started fast development. As wireless network started to be more and more popular a problem of collisions in it started to give more concern.

In order to know more about collisions and the way of collision appearance, the task of thesis was to – inspect the situations of collision, bring out the consequences, possible avoidance of it. Seeing this goal most of popular wireless networks where investigated. Reasons of collisions, experimentally explored situations for collisions, possibilities of resolving it where reviewed. After doing all this work, it can be concluded:

To lower possible collision rate, ensure better throughput, make longer life time to mobile station (save energy) spreading of users (make smaller networks, with lower quantity of mobile users) might be a way.

For example: in case of collision in analyzed WLAN central topology, we would need to add extra access points. That‘s would lower quantity of users per AP.

Bluetooth „ad hoc“ topology is different in structure so this proposal would be valid just when minimum three active devices are in a range.

TURINYS

PRIEDAI

LENTELĖS

PAVEIKSLĖLIAI

ĮVADAS

Technologijoms vis tobulėjant ir greitėjant, prieš apytikriai aštuonerius metus buvo užpatentuotas naujas tinklo būdas – bevielis tinklas.

Bevielis tinklas leidžia naršyti po internetą, spausdinti dokumentus ir bendrai naudotis failais iš bet kurio kompiuterio ar kito prietaiso, nepriklausomai nuo to, ar jie stovi šalia, ar yra skirtinguose aukštuose.

Pastebėjus bevielio tinklo privalumus, jie buvo pradėti sparčiai vystyti. Duomenų perdavimas elektromagnetinių bangų pagalba ne tik žymiai sumažina kabelių panaudojimo kiekį, bet ir supaprastina aptarnavimo procedūras. Tačiau didžiausias privalumas yra vartotojų mobilume, bei tinklų produktyvume, padarant tinklus lanksčius.

Milžiniškas vartotojų susidomėjimas skatino ieškoti skirtingų tinklų tipologijų, technologijų, kurios atitiktų vartotojų reikalavimus, būtų praktiškos naudoti. Vis labiau plintant šioms technologijoms labai išryškėjo, bei tapo daug aktualesnė kolizijos problema.

Esant kolizijai, bevieliu tinklu keliaujantys duomenys sugadinami, gali būti prarandamas ryšys tarp prietaisų. Norint labiau suprasti kolizijas, jų susidarymo priežastis, buvo iškeltas darbo tikslas - išnagrinėti kolizines situacijas, pateikiant jų pasekmes, vengimo galimybes. Norint įgyvendinti šio darbo tikslą, reikia įvykdyti iškeltuosius uždavinius:

- išnagrinėti populiariausius bevielio tinklo atstovus;
- susipažinti su kolizijos priežastimis;
- apžvelgti kolizijų sprendimo būdus;
- Eksperimentiškai ištirti kolizinių situacijų susidarymą;
- pateikti išvadas.

1. BEVIELIAI TINKLAI

Bevieliai tinklai – alternatyvi, lanksti duomenų perdavimo sistema. Bevieliams vietiniams tinklams naudojama radijo dažnių (RF) technologija, leidžianti perduoti ir priimti duomenis per orą: taip sumažinamos lėšos, reikalingos vielinės įrangos įrengimui; palengvinamas tinklo reguliavimas; sumažiname išteklius, reikalingus susijungimams per vielinius tinklus.[1]

Bevielio ryšio pranašumai:

- greitas įrengimas ir įdiegimas;
- bevielio ryšio bangos gali praeiti ten, kur paprastas kabelis nepraeitų (kiaurai per sieną, lubas ir t.t);
- įrengti bevielį tinklą gali būti brangiau nei įprastą vielinį, tačiau jis ateityje atsipirks;
- bevielio ryšio tinklai gali būti pritaikomi įvairioms technologijoms;
- bevielį tinklą galima pritaikyti ir mažoms, ir didelėms įmonėms, kurių darbuotojams reikia susisiekti per atstumą;
- vartotojas gali pasirinkti skirtingas bevielio ryšio technologijas, tinkančias skirtingiems atvejams. [1]

1.1 ZigBee

Kaip ir Bluetooth, Zigbee (IEEE 802.15.4) dirba ISM 2.4 GHz dažniu (16 kanalų su 5 MHz tarpais). Šis standartas taip pat pateikia keletą versijų:

- europietišką 868 Mhz (vienas kanalas) dažniu;
- amerikietišką 915 Mhz (10 kanalų su 2MHz tarpais) dažniu.

Toks ryšio tipas gali pasiekti 250 kbit/s.[2]

Duomenų persiuntimas Zigbee yra pagrįstas DSSS (Direct Sequence Spread Spectrum) schema. DSSS būdas gali skleisti mažesnius trukdžius, bet tokia sistema riboja vykdančiųjų paketų išsiuntimą, užtikrina dažnio pločio išnaudojimą. [2]

Šis standartas nukreipia potencialias silpnąsias vietas. Bluetooth tam tikrose aplinkose: tipiškai mažas gaišties laikas ir laikas duomenų perdavimui taikomosioms programoms. ZigBee taikomosios programos „RF“ fizinis sluoksnis geba pernešti tam tikrą dalį informacijos „overhead“ kai kurioms funkcijoms pagal 802.15.4 specifikacijas.[2]

Papildančios viena kitą technologijos.

Pagal Bluetooth ir ZigBee organizacijas šie standartai yra labiau papildantys vienas kitą negu konkuruojantys. [2]

ZigBee tinkle gali būti iki 4090 atskirų įrenginių. Palyginus su Bluetooth, jame gali būti 7 įrenginiai, plius pagrindinis įrenginys (master). ZigBee protokolas tinka industriniam ir vietiniam stebėjimui, bei sąsajų valdymui, kur ekstremaliai mažas aktyvumas. [2]

Galios išnaudojimas - tai pagrindinis išsiskyrima**s**. ZigBee suprojektuotas labai mažiems darbo apkrovų ciklams, labai ilgam sąsajų darbui, kur akumuliatoriaus gyvavimo laikas skaičiuojamas metais. [2]

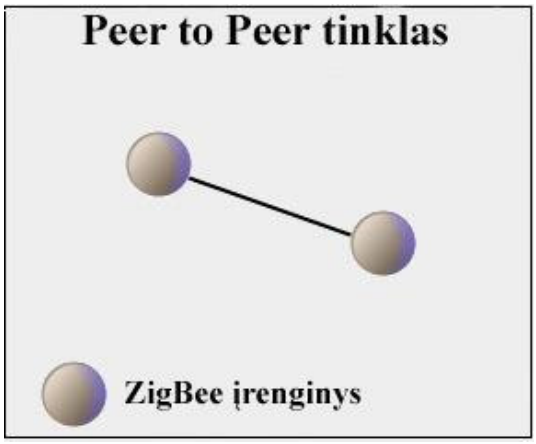
1.1.1 Zigbee įtaisai.

ZigBee tinkluose yra trijų rūšių įtaisai:

- ZigBee coordinator (ZC) koordinuojantis įtaisas: didžiausias galimybes turintis įtaisas. Koordinatorius nutiesia „kelius“, atliekant tinklo tipologiją, taip pat gali sujungti keletą tinklų. Kiekviename tinkle yra tik vienas koordinatorius. Tai yra įtaisas, nuo kurio pradedamas kurti tinklas. Šis tinklo mazgas gali saugoti informaciją, esančią aplink tinklą. Veikia kaip pasitikėjimo centras, atsakingas už saugumą.
- ZigBee Router (ZR) maršrutizatorius: šis tinklo mazgas naudojamas sąsajos paleidimui. Maršrutizatorius taip pat gali būti tarpine grandimi, praleidžiančia informaciją iš kitų tinklo įtaisų.
- ZigBee end device (ZED) galinis įtaisas: šis tinklo mazgas turi tik vieną galimybę - tai „kalbėti“ su pagrindiniu įtaisu (nesvarbu ar tai koordinuojantis įtaisas (coordinator) ar tai maršrutiatorius (rooter) . Šis tinklo mazgas negali perteikti duomenų iš kitų tinkle esančių mazgų. Toks tinklo įrenginio elgesio tipas jiems leidžia ilgą laiko tarpą būti ramybės būsenoje, taupant akumuliatoriaus gyvavimo laiką. ZED tinklo tipo įrenginys reikalauja mažiausio atminties kiekio.

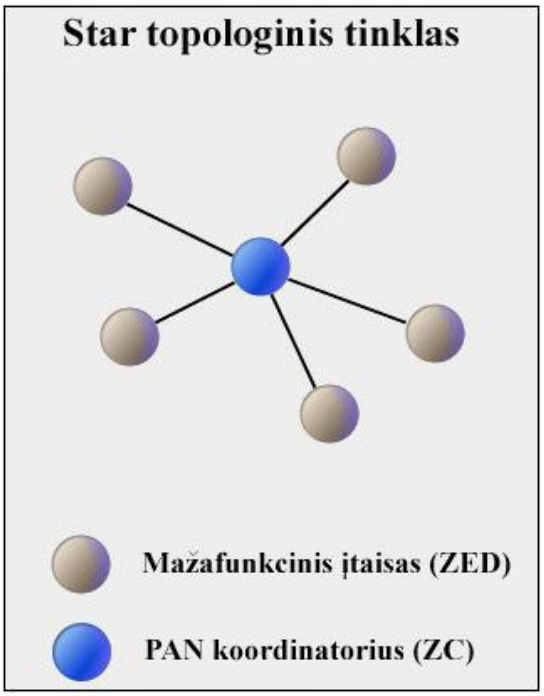
1.1.2 ZigBee tinklų tipai

ZigBee tinklai gali būti pertvarkomi taip, kad galėtų veikti daugeliu skirtingų būdų, atitinkančių sąsają ir aplinką. Palaikomos tinklų sudarymo technologijos: [3]



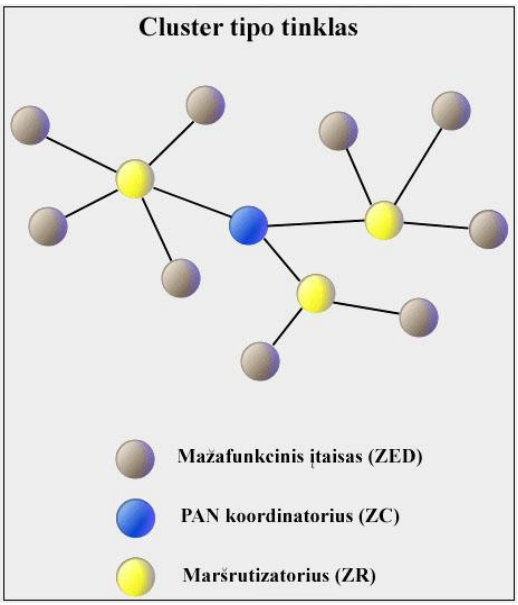
1.1 pav. *„Peer to peer (Ad-hoc)“ tinklo tipas.*

ZigBee tinklo įrenginiai (tinklo mazgai) sujungti tiesiogiai vienas su kitu.



1.2 pav. *„Star configuration“ tinklo tipas.*

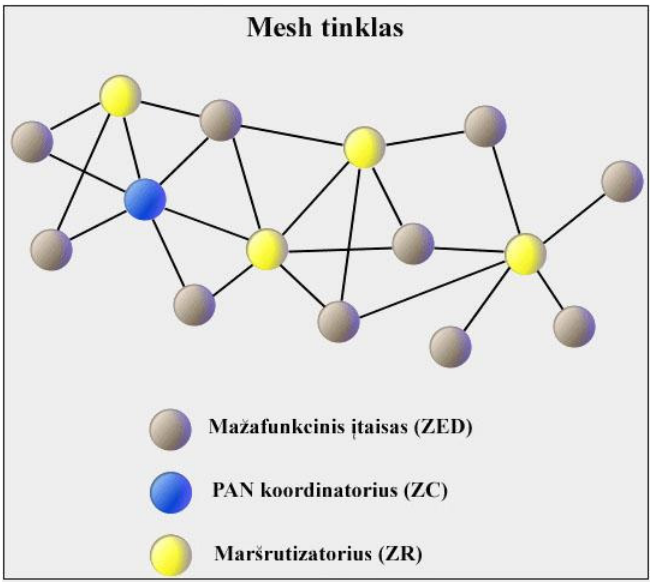
Tokiame tinkle naudojamas vienas PAN koordinatorius. Kiekvienas tinklo įrenginys (mazgas) jungiamas tiesiogiai prie koordinatoriaus (ZC) – visi duomenų apsikeitimai tarp ZED vyksta tarp ZC. [3]



1.3 pav. *„Cluster tree“ tinklo tipas*

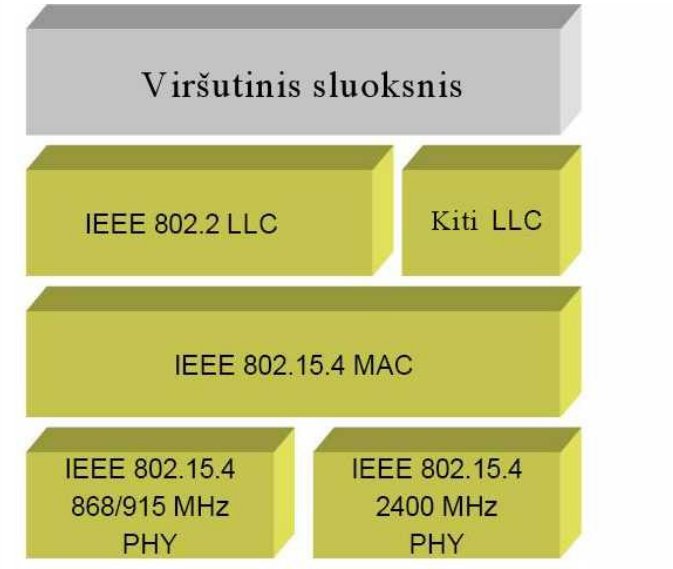
„Cluster“ tipo tinklas susideda iš kelių „Star“ tipo tinklų, sujungtų per centrinį tinklo mazgą. Jis turi galimybę tiesioginiam ryšiui su PAN ZC. [3]

Naudojant keletą maršrutizatorių (ZR) ir vieną ZC, tinklas yra suformuojamas kaip junginys susiderinusių kelių ir ZED įrenginių. ZED įrenginiai, perduodami informaciją, persiunčia duomenis iš ZED į ZED įrenginį naudojant patį efektyviausią kelią. Jeigu kuris nors ZR tampa nepasiekiamas, randami alternatyvūs keliai. [3]



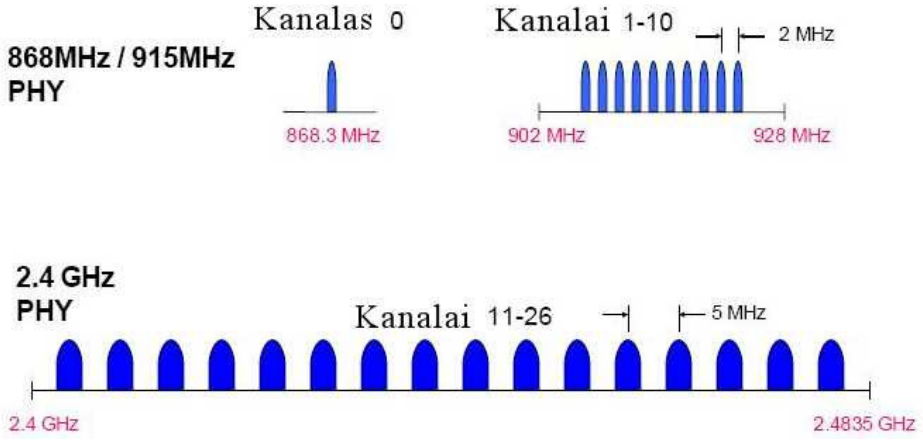
1.4 pav. *„Mesh“ tinklo tipas.*

1.1.3 802.15.4 architektūra



1.5 pav. 802.15.4 architektūra [4]

1.1.4 802.15.4 Darbiniai dažniai



1.6 pav. Darbiniai dažniai. [4]

1.1.5 802.15.4 PHY sluoksnis

Šis standartas pateikia dvi galimybes, pagrįstas dažniu. „Direct sequence spread spectrum“ (DSSS). [4]

1.1 lentelė

802.15.4 PHY sluoksnis

PHY (MHz)	Bangos dažnis (MHz)	Paplitimo parametrai		Duomenų parametrai		
		„Chip rate“ (kchip/s)	Moduliacija	Duomenų greitis (kb/s)	Simbolių dažnis (ksymbol/s)	Simboliai
868/915	868-868.6	300	BPSK	20	20	Dvejetainiai
	902-928	600	BPSK	40	40	Dvejetainiai
2450	2400-2483.5	2000	O-QPSK	250	62.5	16-ainiai

Paketų struktūra.

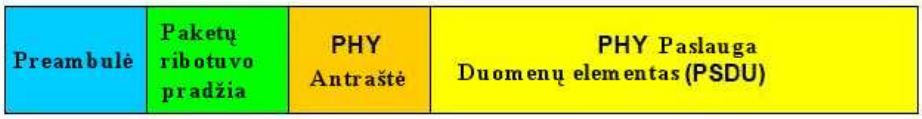
PHY paketų laukeliai:

Preambulė (32 bitai) – sinchronizacija;

Paketų pradžios ribotuvas (8 bitai);

PHY „Header“ (8 birai) – PSDU ilgis;

PSDU (nuo 0 iki 1016 bitų) – duomenų laukas. [4]



1.7 pav. Paketų struktūra.

1.1.6 802.15.4 MAC sluoksnis

Pagrindinė rėmo „frame“ struktūra.

1.8 pav. MAC sluoksnis.

MAC rėmai yra keturių tipų:

1. Duomenų freimas;

2. Pažymėtasis freimas;
3. Vykdomasis freimas.

MAC valdymo freimas. [4]

1.2 Bluetooth

Bluetooth turi visiškai skirtingą sujungimo būdą tarp elektroninių prietaisų trumpuose atstumuose. Tai (Industrial-Scientific-Medical) (ISM) 2,4 GHz technologija.[5]

1.2.1 „Bluetooth“ ir „Infrared“ skirtumai.

Namų elektronikos prietaisai, tokie kaip televizorius ar video grotuvas, komunikuoja su nuotolinio valdymo pulteliu, pasitelkiant šviesos spindulius infraraudonajame spektre. Tokia sistema yra mažo patikimumo.[5]

„Infrared“ trūkumas:

- Siuntėjas ir gavėjas turi būti vienas prieš kitą.

Tai (one-to-one) tipo ryšys. Toks prietaisas negali siųsti signalą į skirtingus prietaisus tuo pačiu metu, todėl, kad prietaiso matymo atstumas yra ribotas, trikdžiai nėra dažni. [5]

1.2.2 „Bluetooth“ ir 802. 11b skirtumai.

„Bluetooth“ ir IEEE 802.11b yra bevielio ryšio atstovai, bei abu dirba 2.4GHz dažniu. „Bluetooth“ - tai nėra technologija pakeisianti 802.11b (bevielio LAN technologiją). Jie suprojektuoti atlikti skirtingus uždavinius.

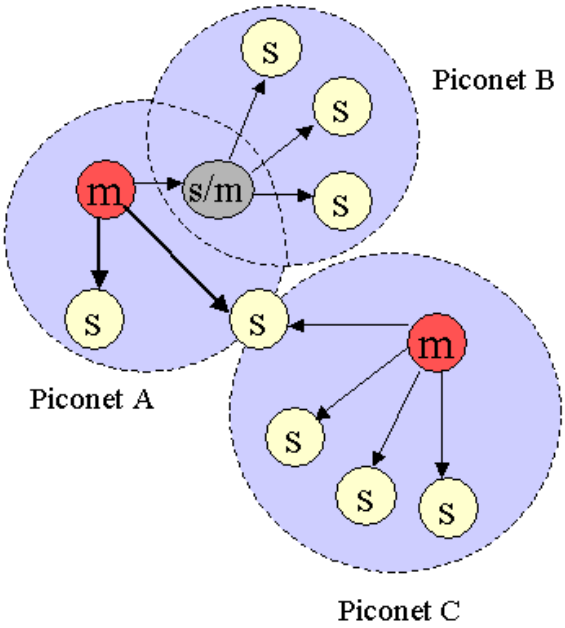
IEEE 802.11b protokolas skirtas sujungti santykinai didelius prietaisus su didele galia bei greičiu (tai gali būti stalinis ar nešiojamas kompiuteris). Esant atstumui, apytiksliai 100 m, prietaisai komunikuoja iki 11 Mbit/sek greičiu. „Bluetooth“ yra sukurtas mažesnių prietaisų bei galių tokių kaip PDA mobiliojo ryšio telefonų ir periferinių gaminių sujungimui, periferinių gaminių su greičiu mažesniu nei 1 Mbit/s ir mažesniu ryšio atstumu (apytiksliai 10 m) sujungimui, kas leidžia sumažinti galios reikalavimą.[5]

Pastabos:

Pagrindinis skirtumas tarp šių technologijų yra tas, kad IEEE 802.11b sujungia prietaisus į LAN (Local area network), o „Bluetooth“ sujungia prietaisus į PAN (personal area network).

IEEE 802.11b buvo sukurtas balso komunikacijai, o „Bluetooth“ palaiko abu - balso ir duomenų komunikavimą.[5]

1.2.3 Bluetooth tinklo tipologija.



1.9 pav. Buetooth tinklo tipologija.

Grupė įjungtų Bluetooth prietaisų vadinami „piconet“. „Piconet“ susideda iš pagrindinio (Master) prietaiso, bei ne daugiau kaip septynių jam priskirtų prietaisų (Slave). Pagrindinis prietaisas, bei jam priskirtasis naudoja „point-to-point“ sistemą, jei yra daugiau priskirtų prietaisų, tai naudojama „point-to-multipoint“ sistema. Pagrindinis prietaisas (Master) inicijuoja ryšio užmezgimą. Prietaisas esantis viename „piconet“ gali bendrauti su kitu prietaisu, esančiu kitame „piconet“, suformuojant „scatternet“. Pastebėkite, kad pagrindinis įrenginys (Master) viename „piconet“ gali būti priskirtuoju (Slave) kitame „piconet“.[5]

M - pagrindinis prietaisas (Master)

S – priskirtasis prietaisas (Slave)

Norint palaikyti „full-duplex“ ryšį, Bluetooth naudoja laiko išskirstymo multipleksavimo (TDM – time division multiplexing) schemą, kurioje pagrindinis prietaisas (Master) persiuntimo metu visada naudoja lyginį prieigos (slot) skaičių. Priskirtieji prietaisai naudoja nelyginį prieigos (slot) skaičių.[5]

1.2.4 Tinklo ryšio sudarymas

Kai prietaisas nėra prijungtas prie „piconet“, jis yra „standby“ rėžime. Šiame rėžime prietaisas klausosi eterio kas 1.28 sekundes, šokinėdamas per 32 dažnius. Kai vienas prietaisas nori prisijungti prie kito, jis išsiunčia šešiolika identiškų kintamųjų į šešiolika dažnio juostų. Jei priskirtasis (Slave) prietaisas neatsako į užklausas, tai jungimąsi inicijuojantis prietaisas pakartoja identifikacijos kodo siuntimą į likusias šešiolika dažnio juostų. Jei jungimąsi inicijuojantis prietaisas nežino priskirtojo (Slave) prietaiso adreso, tai prieš tai jis turi išsiųsti įterptą užklausą, kuri reikalauja papildomo priskirtojo (Slave) prietaiso atsakymo. Jei priskirtasis (Slave) prietaisas atsako į šią užklausą, tada jungimąsi inicijavęs prietaisas gali pradėti balso ar informacijos persiuntimą.[5]

Kad lengviau suprasti kaip „Bluetooth“ įtaisai sukuria tarpusavio ryšį, galime įsivaizduoti, kad Kęstas nori prisijungti prie savo elektroninio pašto. Kai jis paspaudžia elektroninio pašto ikoną, pagal nutylėjimą įvykdomi šie veiksmai:

1. Identifikavimas: naujoje aplinkoje prietaisas automatiškai užmezga ryšį su prieigos tašku (AP). Visi šalia esantys AP atsako į užklausą, pateikia savo adresą. Prietaisas išsirenka vieną iš jų.
2. Sinchronizavimas: su AP įvykdoma sinchronizavimo procedūra.
3. Ryšio sudarymas: LMP (Link manager protocol) protokolas sudaro ryšį su AP.
4. Servisų paieška: LMP naudoja SDP (Service discovery protocol) protokolą, norėdamas išsiaiškinti galimus AP servisus. Iš AP sužinome ar elektroninio pašto servisas yra leidžiamas.
5. L2CAP kanalo sukūrimas: norint sukurti kanalą su AP LPM, naudoja informaciją gautą iš SDP (Service Discover Protocol). Vartotojo sąsaja gali naudoti šį kanalą tiesiogiai arba naudoti protokolą panašų į RFCOMM (Radio frequency communications protocol), kuris gali būti paleistas per L2CAP.
6. RFCOMM kanalo sukūrimas: priklausomai nuo poreikių RFCOMM kanalas (ar kitas kanalas) yra sukuriamas per L2CAP kanalą. Sukūrimas RFCOMM kanalo leidžia inicijuoti „Bluetooth“ prisijungimą, nepadarant jokių pakeitimų.
7. Identifikavimas: tai vienintelis punktas reikalaujantis vartotojo įsikišimo. Jei AP reikalauja identifikacijos, tai bus atsiunčiama identifikacijos užklausa su prašymu įvesti slaptažodį.
8. Prisijungimas: jei prietaisas naudoja PPP (point-to-point) protokolą per RFCOMM, tada sugeneruojamas prisijungimas ir John leidžiama prisijungti.
9. Duomenų siuntimas bei gavimas: atlikus prieš tai buvusias operacijas tarp AP bei vartotojo duomenų persiuntimo, pradedami naudoti standartiniai tinklo protokolai kaip TCP/IP.[5]

1.2.5 Bluetooth ypatumai

„Bluetooth“ naujovėms yra užtikrinamas suderinamumas su skirtingų gamintojų produktais.

„Generic Access“ reglamentuoja jungimosi procedūrų, prietaisų aptikimo, bei ryšio užmezgimo procedūras, apibrėžia šių procedūrų, surištų su saugumu bei dažniausiai naudojamų formatų reikalavimais ir parametrais. Šios nuostatos turi laikytis visi „Bluetooth“ prietaisai.

Veikiančių servisų (service) sąsaja reglamentuoja numatomas naujoves ir procedūras, kurios bus viename „Bluetooth“ įrenginyje; suderinamumą su kito įrenginio „Bluetooth“ įranga.

„Serial port profile“ nustato reikalavimus „Bluetooth įrenginiui, kuris turi sudaryti ryšį bei pamėgdžioti kabelius, naudojant RFCOMM protokolą.

„LAN synchronization profile“ numato kaip „Bluetooth“ prietaisai prisiderina prie LAN, PPP naudojimo. Parodo kaip naudojant PPP mechanizmą galima suformuoti tinklą, susidedantį iš „Bluetooth“ įrenginių.

„Synchrinization Profile“ nurodo sąsajos reikalavimus „Bluetooth“ įrenginiams, kuriems reikia susinchronizuoti duomenų srautą dviejuose ar daugiau įrenginių.[5]

1.2.6 Bluetooth saugumas

Bluetooth, 2.4GHz spektrą dalinasi kartu su mikrobangų krosnelėmis bei 802.11 tinklų prietaisais. Sumažinti abipusius trukdžius, Bluetooth naudoja dažnio šokinėjimo, paskleidžiant spektrą, technologiją, kuri padaliną spektrą į 79 kanalus ir perkeičia komunikacijas 1600 kartų per sekundę. Esant sugadintam paketui, jis persiunčiamas kitu kanalu. Ši persiuntimo schema tvirtai komunikuoja. 128-bit kodavimas užtikriną saugumą. Limituotas Bluetooth prietaisų veikimo laukas taip pat padidina saugumą, kadangi įsilaužimas galimas tik esant tam tikru atstumu. [6]

1.3 IrDA

IrDA yra nebrangi ir plačiai naudotina trumpo atstumo bevielio ryšio technologija. Maksimalus nuotolis tarp dviejų IrDA komunikuojančių prietaisų turėtų būti apytiksliai 2m. Šis ryšio tipas veikia tik tiesioginio matymo zonoje. Ryšys sudaromas tarp dviejų prietaisų. Duomenų siuntimo greitis gali pasiekti iki 16Mbps. Vidutinis siuntimo greitis apytiksliai 1- 4Mbps.[7]

IrDA reikalavimai yra keletą lygių. Pirmasis IrDA reikalavimas - IrPHY (Infrared Physical Layer Specification), kuris yra būdingas visiem IrDA prietaisam. IrDA veikimo technologija apibrėžia:

- kampų limitus;
- greitį;
- nuotolį;
- moduliaciją.

Antrasis ir trečiasis privalomi reikalavimai yra Infrared Link Access Protocol (IrLAP), bei Infrared Link Management Protocol (IrLMP).

IrLAP pažymi sujungimo galimybę su kitu IrDA prietaisu. IrLMP parodo kaip galima prieiti prie paslaugų tiekėjo sąrašo, taip pat kaip randami skirtingų duomenų kanalai.[7]

Yra ir kitų neprivalomų protokolų, tokių kaip:

- IrFM – kuris leidžia naudoti PDA, bei mobiliojo ryšio telefonus kaip bevieles duomenų saugyklas.
- IrLAN – parodantis, kaip galima prisijungti prie egzistuojančio vietinio tinklo.[7]

1.4 GPRS

GPRS (General packet radio service) - tai bevielė perdavimo sistema, kuri užtikrina duomenų perdavimo greičius nuo 56Kbps iki 114Kbps, bei nenutrūkstamą ryšį internetu mobilių telefonų ir personalinių kompiuterių vartotojams. Spartesnis duomenų persiuntimas, leidžia dalyvauti video konferencijose, dirbti su multimedija, bei tinklalapiais. GPRS yra pagrįstas GSM (Global system for mobile) komunikavimo tipu ir atlieka tokias paslaugas, kaip „circuit-switced“ - mobilaus ryšio sujungimus, bei SMS siuntimus (short massage service).[8]

GPRS naudojimo populiarumą nulemia tai, kad prisijungti galima esant, bet kurioje vietovėje, taip pat naudotis interneto naršymo, elektroninio pašto tikrinimo, bei video failų peržiūrėjimo galimybėmis. Operatoriams GPRS įsisavinimas yra daug laiko neužimantis, bei gerai atsiperkantis. Tai yra didelis žingsnis link 3GSM (arba plačiajuosčio CDMA) tinklų ir paslaugų.[9]

GPRS taip pat papildo Bluetooth standartą, keičiantį vielinius sujungimus į bevielius. Papildant IP (internet protocol), GPRS palaiko X.25 protokolą, kuris dažniausiai yra naudojamas Europoje. GPRS - milžiniškas žingsnis link EDGE (Enhanted data GSM enviroment), bei UMTS (Universal Mobile telephone service).[8]

1.5 EDGE

Tolimesnis GSM tinklų tobulinimas privedė prie EDGE technologijos vystimosi. EDGE duomenų talpumas yra tirs kartus didesnis už GPRS. Naudojant EDGE mobilųjį ryšį, operatoriai gali sutalpinti tris kartus daugiau abonentų negu GPRS. EDGE naudoja TDMA (Time Division Multiple Access) struktūrą. Logikan kanalai ir 200kHz nešantysis dažnis išlikęs toks pat kaip ir dabartinių GSM tinklų. Tai leidžia šią technologiją lengvai pritaikyti esamuose GSM tinkluose. Daugumoje egzistuojančių GSM/GPRS tinklų EDGE įdiegimas yra tiesiog programinės įrangos atnaujinimas.[10]

EDGE suteikia daugiau galimybių:

- video bei muzikos klipų parsisiuntimas;
- pranešimų atlikimas multimedijos pagalba;

- naršymas internete;
- elektroninio pašto tikrinimas.[10]

Esant labai mažai EDGE įdiegimo kainai į GSM tinklus, daugumoje naujai statomų GSM tinklų, iškart įdiegta EDGE technologija. Pasaulio vartotojų asociacija (GSA) pranešė, jog 2006 balandžio mėnesį buvo 139 GSM/EDGE tinklai 78 šalyse, iš 192 EDGE tinklų 102 šalyse. GSA pastebi, jog daug šios technologijos yra pritaikoma prakybos centruose, kur prietaisus palaiko EDGE, 3GSM/W-CDMA. Mažesnei daliai prietaisų taikoma EDGE, W-CDMA/HSDPA.[10]

1.6 WLAN

1.6.1 WLAN tipologijos

Centralizuotieji WLAN

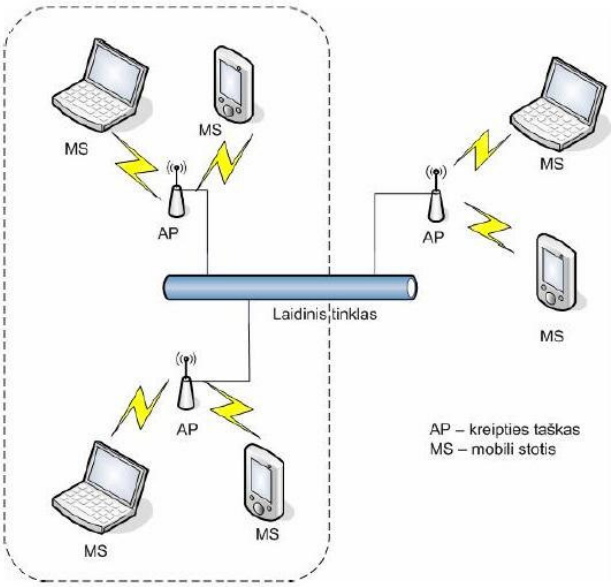
Centralizuotuose tinkluose yra dviejų tipų įrenginiai:

- prieigos taškai (AP);

- mobiliosios stotys MS.

Visi duomenų apsikeitimai tarp mobiliųjų stočių yra vykdomi tik per prieigos taškus. Prieigos taškas vykdo ir tinklo administratoriaus funkcijas. Centralizuoto WLAN tinklo tipologija yra pateikiama 1.10 pav. Kaip matyti centralizuotųjų WLAN struktūra gimininga tradicinių vietinių telefono tinklų struktūrai. AP dalinai vykdo ir komutacinės stoties funkcijas. [11]

Pagal tipologiją WLAN tinklai skirstomi į centralizuotus ir paskirstytus.

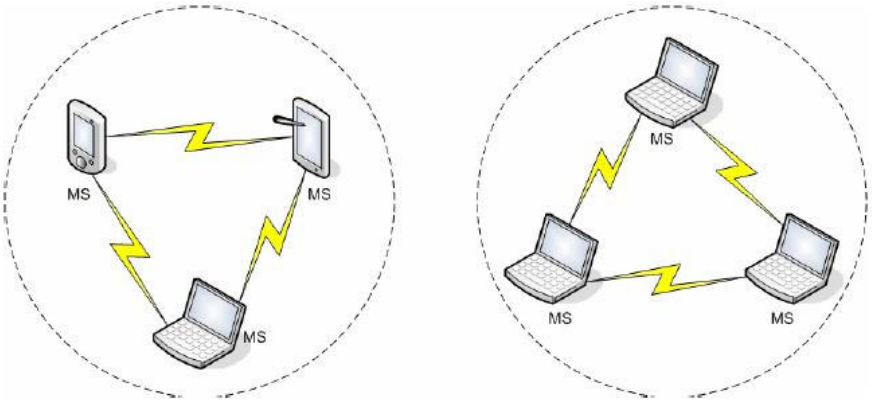


1.10 pav. Centralizuoto WLAN tinklo tipologija.

Paskirstytieji WLAN. Paskirstytieji tinklai dažnai yra vadinami „ad hoc“ tinklais. Šiame darbe paskirstytieji WLAN taip pat vadinami „ad hoc“ tinklais. Tipinė „ad hoc“ tinklo tipologija pateikiama 1.11 pav. Bazinių stočių ar kreipties taškų tokiame tinkle nėra. Esminis „ad hoc“ tinklo skirtumas nuo centralizuoto tinklo yra tas, kad čia stotys ne tik siunčia savo ir priima joms skirtus duomenis, bet ir retransliuoja kitų stočių signalus, kai tos stotys tiesiogiai negali sąveikauti.

„Ad hoc“ tipologijos tinklams būdingi privalumai:

- aukštas gyvybingumas;
- patikimumas. [11]



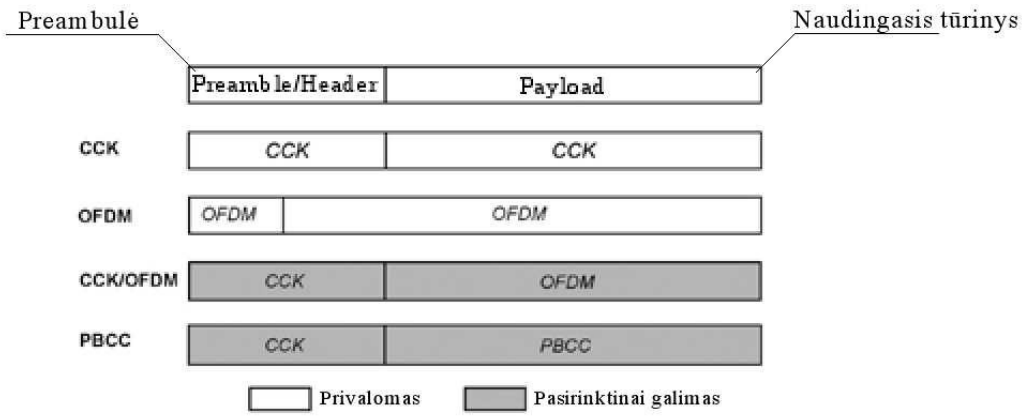
1.11 pav. „Ad hoc“ WLAN tinklo tipologija.

1.6.2 Paketų struktūra

Kiekvienas išsiųstas paketas susideda iš dviejų dalių:

1. Preamble/Header;
2. Playload.

„Preamble/Header“ perspėja apie prasidėjusį duomenų persiuntimą. „Header“ iškart „seka“ „Preamble“ ir pateikia keletą svarbių dalių informacijos, įskaitant trukmę (ms). Kitos radijo stotys, esant šiam periodui, nepradės siuntimo, taip išvengia tinkle kolizijos. Paprastai „Preambule/Header“ bei naudingajai apkrovai (Payload) yra naudojama ta pati moduliacija. Tačiau gali būti išimčių: pavyzdžiui - 802.11 standarte.[12]

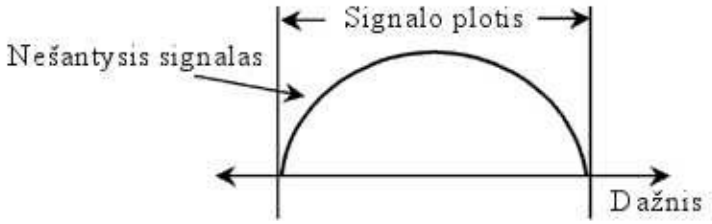


1.12 pav. Skirtingi IEEE 802.11g „draft“ standarto išskirstymas į paketų formatus.

1.6.3 Kodavimai

CCK

Papildomas kodavimas CCK yra naudojamas esamose Wi-Fi (IEEE 802.11b) sistemose. Iš 1.13 pav. matome, kad „Preamble/Header“ bei „Payload“ naudoja tą pačią moduliaciją CCK. Duomenys yra perduodami moduliuojant tuo pačiu radijo dažniu.[12]

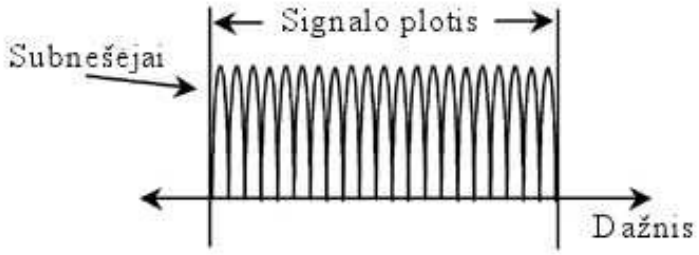


1.13 pav. CCK, „single-carrier“ moduliacijos formatas.

OFDM (Orthogonal Frequency Division Multiplexing) tik dabar pradeda įsitvirtinti rinkoje su IEEE 802.11a prietaisais, veikiančiais 5Ghz dažniu.

OFDM naudojamas dėl to, kad sutrumpinti preambulę. OFDM preambulė tik 16ms ilgio. Palyginus su CCK 72ms, OFDM yra trumpesnė. Mažesnė preambulė sumažina tinklo apkrovimą. OFDM moduliacija yra naudojama tiek „Preambule/Header“, tiek „Payload“.

OFDM duomenys yra paskirstomi keliems šalia esantiems „pernešėjams“ 1.14 pav. Tai leidžia OFDM teikti labai patikimas operacijas (dėl atspindžių) net esant keletui signalo trikdžių, todėl tokia sistema gali palaikyti didesnį duomenų perdavimo kiekį. Iki 11Mbps, CCK yra geras pasirinkimas.[12]



1.14 pav.OFDM sistemos duomenų persiuntimas naudojant kelis „pernešėjus“ (multiple subscribers).

CCK/OFDM

CCK/OFDM tai hibridas CCK bei OFDM. CCK bei OFDM yra naudojami atskirai, nustatytose paketo dalyse. CCK moduliacija naudojama „Header/Preamble“ daliai, OFDM moduliacija naudojama „Playload“ daliai. CCK „Header“ įspėja visus Wi-Fi prietaisus, kada prasideda siuntimas. Informuojama: kiek sujungimas truks (ms). Tuomet galima siųsti „Playload“ dalį, naudojant OFDM, žymiai didesniu greičiu. Net ir tuo atveju, kai dėl kokių nors priežasčių Wi-Fi prietaisas negali priimti „Payload“ dalies yra išvengiama kolizija. Taip atsitinka todėl, kad „Preamble/Header“ dalis yra persiųsta naudojant CCK.[12]

PBCC

PBCC (Packet Binary Convolution Coding) yra „single carier“ sistema, bet ji skiriasi nuo CCK, nes turi skirtingą kodo struktūrą. Apie PBCC galima galvoti kaip apie hibridą bangų formos, nes ji naudojama CCK „Preamble/Header“ ir PBCC „Playload“ daliai. Maksimalus tokios struktūros pasiekiamas greitis - 33Mbps. [12]

1.7 Standartų palyginimas

1.2 lentelė

Standartų palyginimas

	ZigBee	**802.11 (Wi-Fi)**	**Bluetooth**	**UWB (Ultra Wide Band)**	**Wireless USB**	**IR Wireless**
Duomenų siuntimo sparta	20, 40, and 250 Kbits/s	11, 54, 108 Mbits/sec	1 Mbits/s	100-500 Mbits/s	62.5 Kbits/s	20-40 Kbits/s 115 Kbits/s 4 & 16 Mbits/s

1.2 lentelės tęsinys kitame puslapyje

1.2 lentelės tęsinys

	ZigBee	**802.11 (Wi-Fi)**	**Bluetooth**	**UWB (Ultra Wide Band)**	**Wireless USB**	**IR Wireless**
Atstumas	10-100 metrai	50-100 metrai	10 metrai	<10 metrai	10 metrai	<10 metrai (matymo zona)
Tinklo topologija	Ad-hoc, peer to peer, star, ar mesh	Point to hub	Ad-hoc, maži tinklai	Point to point	Point to point	Point to point
Dirbimo dažniai	868 MHz (Europa) 900-928 MHz (NA), 2.4 GHz (visas pasaulis)	2.4 and 5 GHz	2.4 GHz	3.1-10.6 GHz	2.4 GHz	800-900 nm
Kompleksavimas (prietaiso ir sąsajos susidūrimas)	Mažas	Didelis	Didelis	Vidutinis	Mažas	Mažas
Galios išnaudojimas (akumuliatoriaus gyvavimo laikas)	Labai mažas (pagrindini privalumas)	Didelis	Vidutinis	Mažas	Mažas	Mažas
Saugumas	128 AES plius sąsajos saugumas		64 ir 128 bitų kodavimas			
Kita informacija	Prietaisas gali prisijungti prie tinklo greičiau nei per 30 sec	Prietaiso prisijungimas reikalauja 3-5 sec.	Prietaiso prisijungimas reikalauja iki 10 sec.			

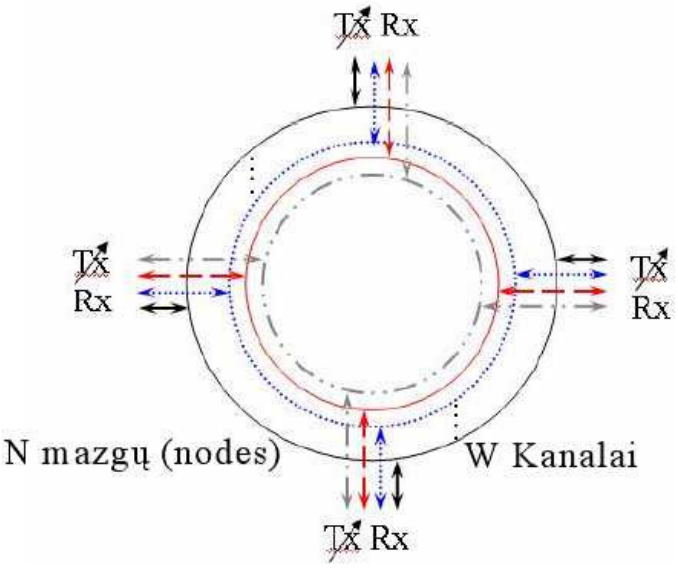
2. KOLIZINIŲ SITUACIJŲ SPRENDIMAS

2.1 CSMA-CP MAC protokolas All-Optical IP-over-DWDM MAN žiediniuose tinkluose

MAC (CSMA/CP) protokolas naudojamas „Metropolitan Area Network“ (MAN) „Access ring“ tinkluose. Šis protokolas suteikia kintamo dydžio IP paketus tiesiogiai į „all-Optical“ DWDM MAN žiedinius tinklus, daugiau išnaudoja tinklo pralaidumą tarp tinkle esančių prietaisų.

Tinklo architektūroje kiekvienas tinkle esantis prietaisas turi galimybę prieiti prie bangos ilgio ir statistiškai pasidalija tinklo pralaidumą (kiekvieno duomenų kanalo). [13]

Keturių bangų ilgių ir keturių jų tinkle esančių prietaisų loginė architektūra:



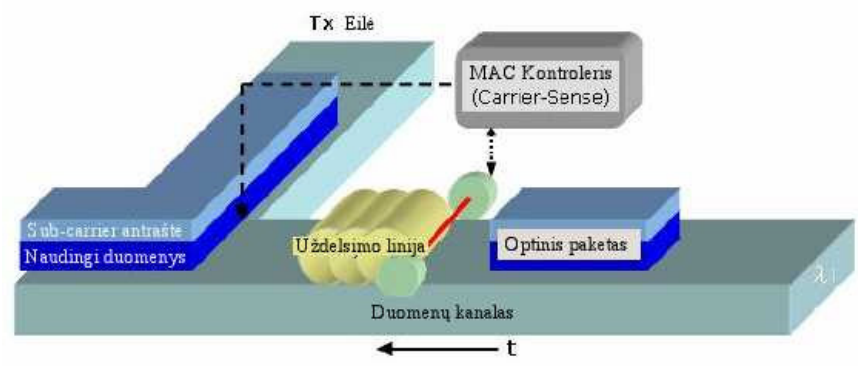
2.1 pav. Loginė architektūra.

Norint išvengti paketų kolizijos ir naudingai valdyti pralaidumą, buvo pasiūlyta „carrier preemtion Medium Access Control“ (MAC) protokolas, kuris pagrįstas nešančiojo mechanizmo jautimo schema. Nešančiojo mechanizmo jautimas naudojamas imtuve, norint išsiaiškinti „sub-carrier“ perduodamą signalą, patvirtinantį apie persiųstus paketus į optinį kabelį. Kiekvienas bangos ilgis *a* pririštas prie „sub-carrier“ dažnio. Tinkle esantys prietaisai aptinka bangų ilgių prieinamumą stebėdami „sub-carrier“, esantį RF srityje. [13]

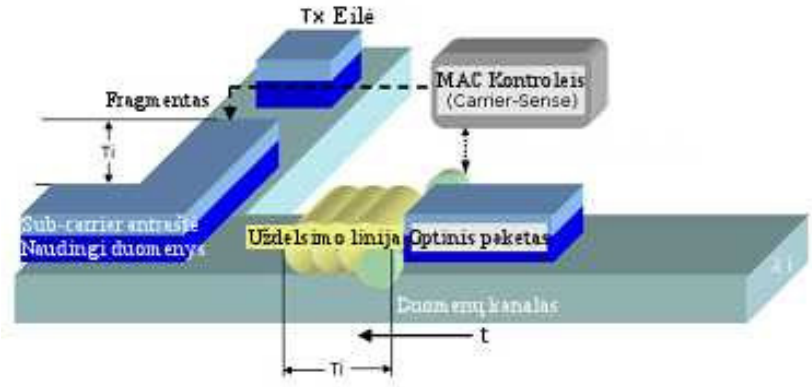
Norint išspręsti priėjimo kolizijas tinkle, kiekvienas prietaisas stebi bangų ilgius ir bando rasti atvirą langą paketų siuntimo kanaluose. Siunčiant į pasirinktą kanalą, kol kiti paketai iš siunčiančio tinklo prietaiso „atvyksta“ į numatytąjį tinko prietaiso tą patį kanalą, įvyksta kolizija. Kolizijos

atsiradimo priežastis yra ta, kad tinko prietaisai neturi pakankamai informacijos ar atsidarantis langas yra pakankamai didelis patalpinti paketą. [13]

Nešėjo išsirinktoje schemoje, susidūręs paketas, kuris nenutraukia siuntimo, bus nedelsiant sudalintas į dvi dalis: viena dalis bus išsiunčiama, kita vis dar turi būti eilėje 2.2-2.3 pav.. Persiuntimo prietaisas gali nesustodamas tęsti pirmųjų siuntimą, kol „atvykstantis“ duomenų „nešėjas“ bus patalpintas į „delay-line“. Kai duomenų „nešimas“ yra užvėlintas T nano sekundėms, duomenų siuntėjas pabaigia siųsti ankstesnį siuntimą. Fragmentas buvęs eilėje išsiunčiamas tuo pačiu kanalu arba esančiu laisvu kanalu. [13]

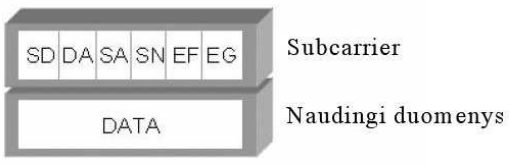


2.2 pav. „Carrier“ „jautimas“ (i kanalas).



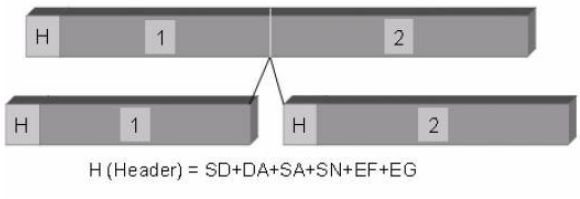
2.3 pav. „Cartier“ pasinaudojimas (i kanalas).

Norint užtikrinti „Carrier“ pasinaudojimo schemą, suformuotas rėmo formatas. 2.4 pav.



2.4 pav. Duomenų rėmo fragmentavimas.

Jis yra pritaikytas adresavimo galimybėms spręsti ir fragmentavimo mechanizmams. Dažniausiai jis savyje talpina pradžios ribotuvą (SD), pažymintį duomenų rėmų antraštes, kurias persiunčia į duomenų kanalą paketais, a fragmentais. Galutinio tikslo adreso (DA) bei duomenų paėmimo adreso (SA) laukai įrašomi į tinklo adresų informaciją. Eiliškumo numeris (SN) nurodo fragmento serijos numerį. Pabaigos fragmentas (EF) yra naudojamas aptikti paskutinį fragmentą. Galiausiai, pažymėtasis laukas (FG) yra rezervuojamas praplėstomis protokolo funkcijomis, tokiomis kaip kitos naudingos duomenų aptarnavimo klasės aptikimas. 2.5 pav. parodo kaip paketas yra suformuojamas atsiradus paketu kolizijai. Persiunčiant ar fragmentuojant, kiekvienas paketas turi būti papildomas rėmo antrašte. [13]



2.5 pav. Duomenų freimo fragmentacija.

2.2 IEEE 802.15.4 CSMA-CA protokolas (Zigbee)

Kai daugiau nei vienas prietaisas tuo pačiu metu išsiunčia rėmą „frame“, įvyksta kolizija. Kolizijai įvykus, visi joje dalyvavę rėmai „frames“ yra sugadinami. Standartinis mechanizmas sutrikimų pašalinimui kompiuteriniuose tinkluose yra vadinamas „carrier-sense multiple access“ (CSMA). CSMA algoritmai suardo nepasisekusio persiųsti simetrijas, bei perkrovimus, tuo pačiu metu naudoja pasirinktinius dvigubos eksponentės funkcijos atsarginės kopijos išsaugojimo procedūras. Laidais sujungti prietaisai gali atlikti tinklo klausymąsi tuo pačiu metu kai atlieka siuntimą ir taikyti CSMA su kolizijos radimu (CSMA/CD). Darbo stotys bevieliuose tinkluose negali klausytis jų pačių persiuntimų, todėl siuntimų susidūrimai gali būti aptikti tik po susidūrimo. Tuo pasėkoje bevieliai prietaisai naudoja CSMA kolizijos išvengimui (CSMA/CA arba CSMA-CA). [14]

Anksčiau tikimybių tikrinimo modelis buvo sėkmingai taikomas ginčytinų sprendimų protokoluose IEEE 802.3 - vieliniuose „Ethernet“ (CSMA/CD), bei IEEE 802.11 bevieliuose „Wireless LAN“ (CSMA/CA) tinkluose. [14]

2.2.1 CSMA-CA algoritmo veikimo principas:

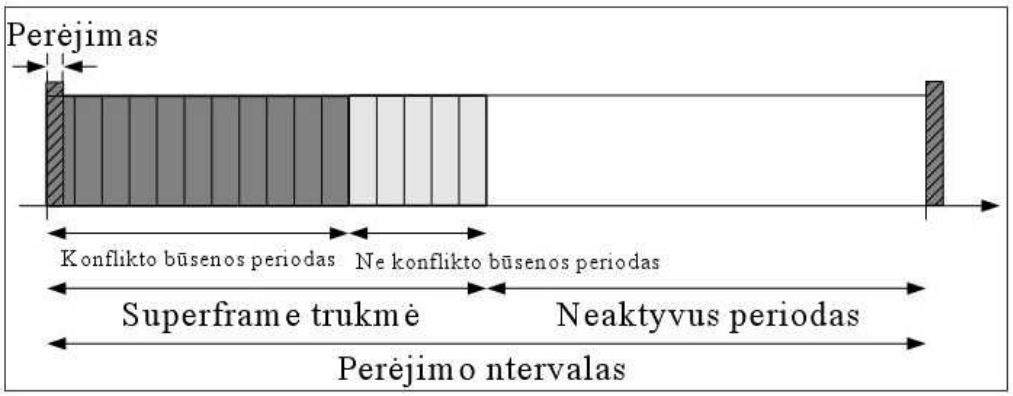
1. *Prieš išsiuntimą:* Jeigu prietaisas nori persiųsti rėmą „frame“ naudojant CSMA-CA, pirmiausia inicijuoja vietinius kintamuosius BE:=macMinBE atsarginėje kopijoje, išsaugant eksponentes, bei pieš išsiuntimą nustato NB:=0, sėkmingų atsarginių kopijų skaičių.
2. *Atsarginės kopijos:* Prieš pradedant vykdyti rėmo „frame“ persiuntimą, laukiama atsitiktinai parenkamo sveikojo skaičiaus tarp 0 ir 2^BE-1 atsarginės kopijos periodo ilgio užbaigimui aUnitBackoffPeriod. Jeigu yra naudojamas rezervuotas CSMA-CA, persiuntimas sinchronizuojamas su paženklinimu. Atsarginė kopija prasideda sekančios atsarginės kopijos periodo pradžioje. Jeigu naudojamas nerezervuotas CSMA-CA, atsarginė kopija prasideda nedelsiant. Pirmasis kiekvieno „superframe“ atsarginės kopijos periodas prasideda iškart paženklinus persiuntimą. Jeigu atsarginė kopija nebuvo pabaigta, iki pasibaigiant CAP, ji prasitęsia startavus sekančiame „superframe“. [14]
3. *Švaraus kanalo įvertinimas:* Padarius atsarginę kopiją, persiuntimo stotis vykdo švaraus kanalo įvertinimą „clear channel assessment“ (CCA). Jeigu po aštuonių simbolių periodo kanalas pasirodo užimtu, abu BE ir NB yra pridedami iškart iki maksimumo aMaxBE (BE) ir macMaxCSMABackoffs+1 (NB). Jeigu NB pralenkia macMaxCSMABackoffs, protokolas nustato negalimą kanalo priėjimą. Jeigu NB nepralenkia macMaxCSMABackoffs, protokolas grįžta prie atsarginės kopijos punkto. Jeigu kanalo priėjimas yra laisvai prieinamas, duomenų freimas gali būti išsiųstas, iš anksto numatytoje CSMA-CA du CCAs, kurių kiekvienas prasideda atsarginės kopijos sudarymo pradžioje. [14]
4. *Siuntimo pradžia:* Numatytame CSMA-CA, siuntimas gali prasidėti tik pasiekus atsarginės kopijos ribą ir jeigu (du CCAs, freimo persiuntimas, ir patvirtinimas) įvykdomi visi – iki vieno, mažiausiai „interframe“ tarpo periodo (IFS) žingsniai, prieš pasibaigiant CAP. [14]
5. *Patvirtinimas:* Jeigu pradininkas neužklausė patvirtinimo, laikoma, kad persiuntimas įvyko sėkmingai. Jeigu patvirtinimas buvo užklaustas, siuntėjas privalo aTurnaroundTime perjungti iš siuntimo į priėmimo rėžimą arba atvirkščiai. Gavėjas pradeda patvirtinimo persiuntimą aTurnaroundTime po paskutinio duomenų simbolio gavimo ar MAC komandos rėmo „frame“. Naudojamas nenumatytas CSMA-CA. Jis prasideda nuo atsarginės kopijos periodo ribos tarp aTurnaroundTime ir aTurnaroundTime + aUnitBackoffPeriod, po paskutinio simbolio duomenų ar MAC valdymo rėmo (naudojant numatytą CSMA-CA) gavimo. Jeigu pradininkas gauna patvirtinimą iš gavėjo su laiku macAckwaitDuration,

duomenų persiuntimas sėkmingas. Negavus patvirtinimo iki tam tikro laiko, freimas bus persiųstas pakartotinai iki maksimumo aMaxFrameRetries kartų. Po aMaxFrameRetries kartų protokolas nutraukiamas ir išleidžiama komunikacijos nesėkmės klaida. [14]

2.2.2 „Superframe“ struktūra

Norint užsigarantuoti mažus paraiškos sugaišimo laiko tarpus ir specifinio duomenų pralaidumo paraiškas, IEEE 802.15.4 tinklai gali būti sinchronizuojami komunikacijai pagal „superframe“ struktūrą. Kiekvienas „superframe“ susideda iš 16 vienodai padalintų tarpų ir yra apribojamas tinklo ženklais, kurie yra pastoviai siunčiami suprojektuoto koordinuojančio prietaiso.

Duomenų freimų persiuntimui, iškilus ginčytinam priėjimo periodui, naudojamas CSMA-CA sugaišties būdas. Kol persiuntimo procesas yra ginčytinas, įsigali laisvas periodas, užtikrinantis skirtąjį laiko tarpą. Tinkluose, neturinčiuose pažymėtosios sinchronizacijos, duomenų freimai persiunčiami nesuteikiant laiko tarpų CSMA-CA. [14]



2.6 pav. „Superframe“ struktūra.

2.3 Taupantis energiją, be kolizijų, vidutinio priėjimo kontrolės (MAC), bevieliai tinklai

Prie srauto prisitaikantis vidutinio priėjimo kontrolės protokolas (TRAMA) yra pristatomas kaip energiją taupantis, be kolizijų, bevielis tinklas. TRAMA sumažina energijos naudojimą užtikrina, kad siuntimuose „unicast, multicast, brouadcast“ nebus kolizijų, bei leidžia tinklo mazgams (nodes) pereiti į mažesnę galią naudojantį rėžimą. TRAMA daro prielaidą, jog laikas kontaktavimui yra išskirtas, ir naudoja paskirstyto išrinkimo schemą, pagrįstą informaciją apie kiekvieno tinklo mazgo (node) judėjimą, nustato, kuris tinklo mazgas gali persiųsti tam tikrą laiko išskyrimą. TRAMA išvengia laiko gaišaties su tuščiu apkrovimu paskyrimo tinklo mazgams, taip pat leidžia tinklo mazgams nustatyti,

kada gali pereiti į laukimo rėžimą (nesiklausyti kanalo naudojant srauto informacijos). TRAMA parodoma teisinga ir tiesi, tinklo mazgas esantis ne laukimo rėžime yra numatytasis gavėjas ir joks kitas prietaisas nepatiria kolizijos. TRAMA našumas yra įvertintas per ištęstą modeliavimą, naudojant abu „syntetic-„ bei „sensor-network“ scenarijus. Rezultatas parodo, kad TRAMA dirba geriau „contention-based“ protokolais (pvz. CSMA, 802.11 ir S-MAC), taip pat „scheduling-based“ protokolais (pvz. NAMA) su stulbinančiais energijos išsaugojimais. [15]

2.4 IEEE 802.11 standarto problemos, sprendimai

2.4.1 MAC lygio „paslėpto taško” problema

Kita savita MAC lygio problema yra “paslėpto terminalo” problema – kai dvi *2* stotys gali gauti signalus iš prijungimo mazgo, bet negali viena iš kitos, dėl per didelio atstumo ar kliūčių. [16]

Norint išspręsti šią problemą 802.11MAC lygyje, naudojamas RTS/CTS protokolas (Request to Send/Clear to Send). Stotys siuntėjos pasiunčia RTS ir laukia prijungimo mazgo atsakymo CTS. Iš prijungimo mazgo gautas CTS signalas priverčia stotis atidėti jų duomenų siuntimą ir viena stotis gali siųsti duomenis ir gauti ACK paketą be kolizijų. Kadangi RTS/CTS yra brangus duomenų siuntimo metodas, jis paprastai naudojamas dideliems duomenų paketams siųsti, kurių pakartotinis siuntimas yra per brangus. [16]

Galiausiai, 802.11 standarto MAC lygis numato paketų suskaidymą ir CRC skaičiavimą. Kiekvienas paketas turi nustatytą ir patvirtintą CRC sumą. Tai skiriasi nuo Etherneto tinklų, kur klaidas apdoroja aukštesnio lygio protokolai (pvz.: TCP). Duomenų paketų skaidymas leidžia didelius duomenų paketus prieš juos siunčiant suskaidyti į mažesnius. Tai naudinga kai aplinka yra labai apkrauta ar labai triukšminga, todėl, kad tokiu būdu mažesni paketai yra mažiau apgadinami. Šis metodas sumažina pakartotinių siuntimų skaičių, taip pagerindamas viso bevielio tinklo darbą. Mac lygis atsako už gautų fragmentų sujungimą, kad procesas būtų aiškus ir aukštesnio lygio protokolams. [16]

2.4.2 Kanalų (Data Link) lygis 802.11 standarte

Duomenų lygis susideda iš 2 pakatalogių: LLC (Logical Link Control) ir MAC (Media Access Control). 802.11 standartas naudoja tokį patį LLC ir 48 bitų adresavimą kaip ir kiti 802 standarto tinklai. Tai suteikia galimybę prisijungti prie paprastų ir bevielių tinklų. [16]

MAC lygis yra visiškai kitoks.

802.11 standarto MAC lygis yra panašus į vieną realizuotą 802.3 standarte, kuris palaiko daug vartotojų bendroje aplinkoje. Vartotojas gali pasitikrinti aplinką prieš į ją įeidamas. [16]

802.3 standarto Etherneto tinkluose naudojamas CSMA/CD protokolas (Carrier Sence Multiple Access with Collision Detection). Šis protokolas aprašo kaip Etherneto stotys prieina prie paprastų tinklų ir kaip jos aptinka ir pašalina kolizijas, kurios atsiranda, kai keletas prietaisų bando veikti tame pačiame tinkle, tuo pačiu metu. Kad aptiktų koliziją, stotys turi siųsti ir gauti duomenis. 802.11 standartas numato vienpusį siųstuvo/imtuvo panaudojimą, todėl stotis negali nustatyti kolizijų, kai duomenys perduodami bevieliame 802.11 standarto tinkle. [16]

Kad pašalinti šį trūkumą 802.11 standartas naudoja pakeistą CSMA/CA protokolą (Carrier Sence Multiple Access with Collision Avoidance) arba DCF (Distributed Coordination Function). CSMA/CA protokolas stengiasi išvengti kolizijų naudodamas ACK paketo patvirtinimą. Tai reiškia, kad imtuvas turi nusiųsti siųstuvui ACK paketą, kad patvirtintų, jog gavo nesugadintą duomenų paketą. [16]

CSMA/CA dirba tokiu būdu. Stotis, pasiruošusi siųsti duomenis, patikrina kanalus ir jei neaptiktas tų kanalų aktyvumas, palaukusi atsitiktinį laiko tarpą, pradeda siųsti duomenis, jei aplinka vis dar yra laisva. Jei duomenų paketas gautas nesugadintas, stotis - gavėja persiunčia ACK paketą stočiai - siuntėjai, kurį gavus jį baigia darbą. Jei stotis - siuntėja ACK paketo negauna, ji nusprendžia, kad galėjo įvykti kolizija ir palaukus atsitiktinį laiko tarpą siuntimą pakartoja. [16]

Norint išsiaiškinti ar kanalas laisvas, naudojamas CCA (Chanel Clearance Algorithm). Jis matuoja signalo energiją ir apibūdina gauto signalo (RSSI) galią. Jei gauto signalo galia yra mažesnė už nustatytą bazinę galią, tada kanalas laikomas laisvu ir MAC lygis suteikia CTS (Clear to Send) statusą. Jei galia didesnė nei bazinė, siuntimas atidedamas remiantis protokolo taisyklėmis. Standartas siūlo dar vieną būdą, kaip nustatyti ar kanalas laisvas ar ne, kuris gali būti naudojamas kartu arba atskirai nuo RSSI. Tai – CFM (Carrier Frequency Method). Šį metodą labiau renkasi tie, kurie tikrina to paties tipo nešamo dažnį kaip ir 802.11 standarte. Kuris metodas geresnis, priklauso nuo triukšmo lygio darbo aplinkoje. CSMA/CA protokolas suranda būdą, kaip efektyviai spręsti triukšmo problemas, tačiau tai

reikalauja daug išlaidų, todėl 802.11 standarto tinklai dirba lėčiau nei ekvivalentūs Etherneto LAN'ai. [16]

2.4.3 FHSS metodas

FHSS metode 2.4 GHz juosta yra dalinama į 79 kanalus po 1 MHz. Gavėjas ir siuntėjas suderina tarpusavyje įjungimo/išjungimo schemas (jų yra 22) ir duomenys sėkmingai persiunčiami, naudojantis viena iš schemų. Kiekvienas duomenų persiuntimo metodas 802.11 standarto tinkle yra įgyvendinamas naudojant skirtingas perjungimo schemas. Schemos sudarytos taip, kad kaip galima labiau sumažintų galimybę dviem siuntėjams tuo pačiu metu pasinaudoti vienu kanalu. Tačiau esant labai dideliam seansų skaičiui, atsiranda kolizijos tikimybė.

FHSS metodas siūlo labai paprastą siųstuvo/imtuvo konstrukciją, bet jis yra apibrėžtas 2 Mbps greičiu. Taip yra todėl, kad 1 kanalas užima lygiai 1 MHz pločio juostą, o FHSS metodas turi naudoti visą 2.4 GHz diapazoną. Tai įtakoja dažną kanalų perjungimą (pvz.: JAV minimalus perjungimo greitis yra 2.5 perjungimo per sekundę). Tačiau visa tai labai daug kainuoja. [16]

2.4.4. DSSS metodas

Šis metodas 2.4 GHz juostą padalina į 14 kanalų (JAV jų yra 11). Jei kanalai naudojami vienu metu toje pačioje vietoje, jie turi būti atskirti 25 MHz juostomis, kad išvengtų triukšmo. Tai reiškia, kad tik 3 kanalai gali būti naudojami toje pačioje vietoje. Duomenys yra perduodami vienu iš tų kanalų, nepersijungiant į kitus kanalus. Norint kompensuoti nepageidaujamą triukšmą, naudojama Barker'io 11-bitų seka. Kiekvienas vartotojo duomenų bitas yra konvertuojamas į 11 duomenų bitų persiuntimą. Toks aukštas kiekvieno bito dubliavimas leidžia žymiai padidinti perdavimo patikimumą su žemesne signalo galia. Net jei dalis signalo ir butų prarasta, jis dažniausiai yra atstatomas. Tai labai sumažina pakartotinių siuntimų skaičių. [16]

2.5 Virtualus FIFO „Back-Off" algoritmas bevielių tinklų kolizijų sprendimui.

FIFO (VFIFO) bevieliams tinklams „back-off" algoritmas. Privalumas yra „central unit" (CU) bevieliuose tinkluose visiems vartotojams transliuojamas „back-off" lango dydis. Žymiai sumažinamas neteisingas bangos pločio utilizavimas, tradiciniai dvejetainiai eksponentinei „back-off" (BEB)

algoritmai. Pasiūlyta schema išnaudoja CU trūkumus, gebėjimą nustatyti koliziją, įvertinti kartu besivaržančius (siunčiančių užklausą) vartotojus.[17]

Papildomai, sugeneruoti paketai pagal duotuosius ciklus yra padalinami į grupes pagal atvykimo laiką ir yra garantuoti, kad jų aptarnavimas ateis vienas po kito, kartu su sekančiu ciklu. Pasiūlytasis algoritmas nėra griežtai apibrėžtas aptarnauti pirmumo teise (first come, first served). Standartinis vėlinimas gali būti pagerintas su daugiau nei dviem užsakymais, o pralaidumas gali būti palaikomas ant 0.42, kai vartotojų kiekis pasiekia begalybę. Pagavimo efektas dar labiau sustiprina sistemos greitaveiką. [17]

2.6 SELECT: savaime apsimokantys kolizijos išvengiantys bevieliai tinklai

SELECT yra „sender-side-only“ kolizijos išvengimo mechanizmas. Jis perkelia paslėpto/neapsaugoto informacijos gavėjo (hidden/exposed-receiver) problemą į paketų laiko skaidymo lygį. Vien tik ant įrangos perspektyva nepagrįsta (pvz. daugiakanalės komunikacijos galimybė, kaip siūloma BAPU). Vietoj to, SELECT naudoja vien momentinius RSS išmatavimus. Standartizuota sensorinė funkcija įmontuota daugumoje bevielių siųstuvų - imtuvų siuntimo aptikimui (pvz. Off-th-shelf 802.11 bevieliai prietaisai). SELECT nėra bevielio tinklo analitinis signalo persiuntimo modelis, įtakojamas daug faktorių, todėl labai sunku tai analitiškai įvertinti. Galiausiai SELECT suderinamas su 802.11 fiziniu lygmeniu (PHY), vidutinio priėjimo kontrolės (MAC) nustatymais ir su kitais non-SELECT 802.11 prietaisais. SELECT įtraukimui į 802.11 DCF, reikalaujami tik maži negriaunamieji pataisymai kolizijos vengimui. [18]

2.6.1 Savaime apsimokantys kolizijos išvengimai

Intuityviai stipri koreliacija tarp siuntėjo RSS ir gavėjo RSS, leidžia siuntėjui nustatyti RSS bei kanalo prieinamumą. Šis priėjimas buvo pasirinktas ne veltui, to priežastys:

- pirmiausia pradedant RSS koreliaciją būtina, kad informacijos gavėjas turėtų RSS grįžtamąjį ryšį laike. Šis galinis ryšys išvengiamai įtrauks kai kuriuos signalus, esančius tarp siuntėjo ir gavėjo, kuris komplikuoja MAC sluoksnį bei/arba PHY sluoksnį.

- numatant imtuvo RSS aptinkami tik pažeidžiami imtuvai. RSS esantis paslėptame imtuve gali būti silpnas. [18]

Imtuvas nebeatsako į CTS per du scenarijus. Pažeidžiamame imtuve RTS susiduria su išeinančiąja transliacija, tuo tarpu paslėptame imtuve kanalai yra iš anksto rezervuoti CTS

išeinančiosios transliacijos. Iš siuntėjo pozicijos pasekmės šių dviejų scenarijų yra vienodos ir neturi skirtis. Tuo tarpu tikslus RSS numatymas taip pat ne pakankamas (kai imtuvas paslėptas) nei būtinas (kadangi siuntėjui reikia žinoti tik, kad imtuve yra žemas RSS). Vietoj to siuntėjas gali praleisti numatytąjį imtuvo RSS bei tiesiogiai užmegzti paskirstymą tarp kanalų RSS ir SR. Kadangi RTS sėkmingumas yra apibrėžiamas pagal CTS grįžimą, RTS nesėkmė yra apibrėžiama CTS „time out“. Jokie kiti signalai tarp siuntėjo ir gavėjo nėra būtini. [18]

2.6.1.1 RSS-SR paskirstymo priežiūra

Paslėpti/pažeidžiami imtuvai komplikuoja bendravimą tarp siuntėjo RSS bei kanalų priėjimo SR. Šis sudėtingumas bei reikalavimai pritaikomumui paneigia prieš tai pareikštą mintį apie paskirstymus analitinėmis formomis. Žinant, kad daugiausia 802.11 tinklo sąsajų „kortos“ (NIC‘s) turi bent 128 Kbytes integruotus statinio RAM (SRAM), mes pasirinkome išlaikyti tiesią histogramą, mainais į palyginamai mažą saugyklą (šimtai baitų). Tai daroma dėl mažesnio projektavimo sudėtingumo bei paprastesnių apskaičiavimų. [18]

Konkrečiai, mes padaliname RSS nuotolį, [RSS^{min}, CS^{thred}], į N intervalų [RSS_i^{min}, RSS_i^{max}] (i=1,…,N), kur RSS^{min} yra nustatytas į tam tikrą garso lygį arba pagal nutylėjimą matuojamas minimalus RSS bei CS^{thred}, nešančiojo dažnio slenkstis. Norėdami išgauti efektyvę išvadą (lookup), mes padaliname intervalus po lygiai. Kiekvienam RSS intervalui I_i, gaunamos trys kintamųjų būsenos:

S_i sėkmingi bandymai prieiti prie kanalų,

F_i nesėkmingi bandymai prieiti prie kanalų,

T_i^{upd} laiko anspaudas nurodantis paskutinį S_i ar F_i atnaujinimą.

Iš esmės histogramą galime gauti kaip vienmatį masyvą su N elementų iš $I_i=<S_i,F_i,T_i^{upd}>$. Masyvo dydis N gali būti apibrėžtas laisva atmintimi. RSS matavimai yra sukaupiami ir pristatomi vieno I_i. Tai gali ir nuslopinti RSS matavimo klaidas. [18]

2.7 pav. parodo siuntėjo pseudo kodus, norint atnaujinti atvaizdavimą esant bandymui, taip pat atnaujinti kanalų priėjimus. Parodoma pasisekė ar nepasisekė tai atlikti. Kaip matome histograma padengia tik kai $RSS<CS^{thred}$, tuo metu nebus jokio bandymo prieiti prie kanalo jeigu kanalas nėra laukimo būsenoje. Taip pat nustatinėjame RSS^{min} numatomą ar pamatuotą triukšmo lygį tol, kol RSS nustatymai žemiau triukšmo lygio tampa nepatikimi. [18]

// Input - rss: Nešantysis dažnis RSS
// - sf: 1 jei pasiseka, 0 jei ne, -1 jei nėra nauju įrašų
Upd_RSS_SR(rss, sf)
1. $i = \lfloor \left(rss - RSS^{min}\right) / I_{width} \rfloor$; // Rasti elementą I_i
// $I_{width} = (CS^{thred} - RSS^{min})/N$
2. $\alpha = 1 - (t - T_i^{upd})/T_{win}$; // Prisitaikantis senėjimo efektas
3. if ($\alpha < 0$) then $\alpha = 0$;
// if ($t - T_i^{upd} > T_{win}$) švarus S_i ir F_i
4. if ($sf == 1$)
then $S = 1, F = 0$; // Pavykęs kanlų priėjimas
else if ($sf == 0$)
then $S = 0, F = 1$; // Nepavykęs kanlų priėjimas
else $S = 0, F = 0$; // Nėra naujų įrašų
5. $S_i = \alpha \cdot S_i + S$; // Update # Nėra sėkmingų bandymų
6. $F_i = \alpha \cdot F_i + F$; // Update # Nėra nesėkmingų bandymų
7. $T_i^{upd} = t$; // Atnaujinimo laiko žymė

2.7 pav. Siuntėjas atnaujina atvaizdavimą su įrašais {rss,sf}.

Norint atvaizduoti esamą darbo aplinką, pasenę įrašai turi būti panaikinti. Taip pat priverstinai S_i bei F_i turi būti kaip skaičius, sėkmės/nesėkmės kanalų priėjimo mėginimai, su esamu laiko langu T_{win}. Pastebėkite, kad esami T_{win} nustatymai priklauso nuo aplinkos dinamikos, *tokios kaip* apkrovos kombinacijos, tinklo tipologijos, bei signalo sklidimo. Jeigu mes žymėsime naują įrašą kaip <rss, sf>, kur sf = 1 - sėkmingų kanalų bandymų skaičius, sf = 0 - nepavykusių kanalų bandymų skaičius. Standartinei langų sumos aproksimacijai, S_i, bei F_i periodiškai pritaikysime senėjimo faktorių α: $S_i = \alpha S_i + S$ bei $F_i = \alpha F_i + F$, kur F=~S. Atnaujinimo periodas yra nustatytas $T_{period} << T_{win}$ ir $\alpha = 1 - T_{period}/T_{win}$. Įrašus siuntėjai visada gauna periodiškai. Be to, atnaujinant visus N S_i ir F_i į O(N) įvyksta priešlaikinis atnaujinimas. Mes nukreipiame šias dvi problemas adaptuodami senėjimo faktorių α pagrįstą paskutinio atnaujinimo laiku T_i^{upd}: [18]

$$\alpha = \begin{cases} 1 - \dfrac{t - T_i^{upd}}{T_{win}} & if \quad t - T_i^{upd} < T_{win} \\ 0 \end{cases}$$

Rezultatai gali būti gaunami, imituojant periodiškai mažus atnaujinimo laiko periodus. Su dinaminiu senėjimo faktoriumi atvaizduojamas atnaujinamas tik pagal pareikalavimą:

1) po to kai siuntėjas gauna naujus įrašus;

2) prieš tai kai siuntėjas pareikalauja kanalo priėjimo SR atvaizdavimo.

Greito grįžimo užtikrinimui, priešlaikinio atnaujinimo sudėtingumas taip pat sumažinamas iki 0 (1). [18]

2.6.1.2 RSS-SR atvaizdavimo informacijos ieškojimas

Su RSS-SR atvaizdavimo palaikymu, siuntėjas gali sudaryti eilę atvaizdavimų, naudojant tam tikrą nešantįjį dažnį RSS, gauti SR kanalų priėjimų bandymų istoriją. 2.8 pav. parodo pseudokodus. Kai informacijos ieškojimo užklausa gauta, atvaizdavimas atnaujinamas pirmiausia, kad būtų galima panaikinti pasenusius įrašus (tuos įrašus, kurie iškrenta iš T_{win} lango). Tada aptinkame atnaujintą intervalą I_i, atsižvelgiant į užduotą RSS, ir išnagrinėjame ar yra pakankamai įrašų. Istorinis SR grąžinamas į bendrą skaičių, pasisekimų - nepasisekimų slenkstį. Kitaip tariant SR grąžinama 100%. Sumanymo esmė ta, kad pagal nutylėjimą mažesnis negu nešančiojo dažnio slenkstis RSS, CS^{thred} yra laikomi laukimo būsenos kanalu. [18]

```
// Input - rss: Nešantysis dažnis RSS
// Output:     Kanalų sėkmingų bandymų istorija
RSS_SR_LookUp(rss)
1.  if (rss ≥ CS^thred) return 0%;
                // Siuntimo kanalas užimtas
2.  Upd_RSS_SR(rss, −1); // Panaikinami pasenę duomenys
3.  i = ⌊(rss − RSS^min) / I_width⌋; // Aptikti elementą I_i
                // I_width = (CS^thred − RSS^min)/N
4.  if (S_i + F_i > Min_Num_Rec) then // Užtenkamai įrašų
        return S_i/(S_i + F_i); // Sėkmingumas
    else                        // Neužtenkamai įrašų
        return 100%; // Pagal nutylėjimą kanalas "idle" lauk.b.
```

2.8 pav. Siuntėjas sustato eiliškumą SR kanalų priėjimų istoriją pagal esamus RSS.

Slenkančio laiko langų įrašai histogramoje yra pagrindinis kriterijus sistemos prisitaikymui. Per pilną ciklą iš MAC modulio išeinantys įrašai susiję su sėkme, - nesėkme į SELECT modulį patenka detaliam RSS-SR atvaizdavimui sukurti. Kai ciklas uždaromas, MAC modulis vengs bandymų kreiptis į RSS kanalus, pažymėtus SELECT, kaip mažo prieinamumo SR. Tokiu atveju, jokių kitų įrašų nebus įgyjama, kol RSS intervalas bus atvaizduojamas kaip žemo kanalo prieinamumo SR ir artimo ciklo sistema sustings prie RSS intervalų, turėdami galimybę pasikeisti. Per esamą langą, ribojant tinkamus įrašus bei panaikinant susidūrimą su senaisiais, artimojo ciklo sistema gali pamažu atsikurti. Taigi MAC modulis bandys pamažu prieiti prie RSS kanalų, kurie buvo manyti, jog turi būti priskirti kaip nesėkmingi. [18]

2.7 Bevielių tinklų sparta naudojant kolizijos atsparumo moduliaciją

2.7.1 Apibendrinimas

Siekiant pagerinti bevielių tinklų spartą („nublankinant“ „multiple access“ kanalą), taip pat išnaudojant kanalų kodavimo klaidų taisymo galimybę, buvo pasiūlytos kelios kodavimo technologijos. Esant kolizijoms, šios technologijos leidžia sumažinti neigiamą įtaką sistemai. Rezultatai rodo, kad galima pasiekti stebėtinų laimėjimų, vietoj tradicinių moduliacinių schemų, pagrįstų „Slotted ALOHA“ protokolu, naudojant CRM (Collision Resistant Modulations). Pagrindinis privalumas - šio signalo lauko kodavimo prieš tradicinio kodavimo schemas, yra tai, kad koduojant nepadidėja pralaidumas rezervinės informacijos perdavimui. [19]

Išnagrinėsime CRM potencialius privalumus kartu su labiau kompleksuotu MAC protokolu, kuris turi kelias siuntimo galimybes: „random Access“ bei „collision free“. [19]

2.7.2 Kolizijos atsparumo moduliacija

Kolizijos kanalą sumodeliuojame kaip on-off vektorinį kanalą.

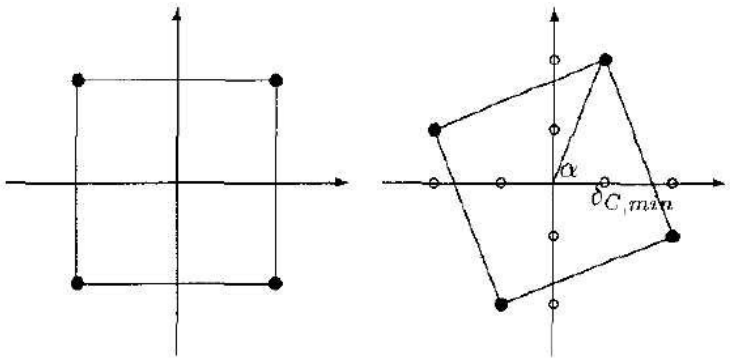
$$y = C\,(G_s + n) \qquad (1)$$

kur: $s = (s_1,...s_D)^T$ yra išsiųsto signalo vektorius (ar „simbolis“), paimtas iš signalo S, |S| esant Euklido dydžio D, $n = (n_1,\ldots.n_D)^T$ yra pridedamas. Gauso triukšmas su $n_i \sim N(0, N_0/2)$, $y = (y_1,...y_D)^T$ - gautasis signalas, $G = \mathrm{diag}\,(g_1,....g_D)$ kanalų persidengimas, bei $C = \mathrm{diag}\,(c_1,....c_D)$, su $c_i \in \{0,1\}$, kolizijos šablonas (0 apibrėžia koliziją). Iš formulės (1) akivaizdu, kad ištransliavus s komponentą į lizdą kur buvo įvykusi kolizija, jis nedelsiant ištrinamas. [19]

C ir G yra gerai žinomi, kai imtuvas turi puikią kanalų būsenos informaciją (CSI). Maksimalaus tikėtumo (ML) nuosprendžio taisyklė gaunama minimizuojant visus $s \in S$, esant modifikuotam Euklido nuotoliui:

$$d_C^{\,2}(s, y) = \sum_{i=1}^{D} c_i (y_i - g_i s_i)^2 \quad . \qquad (2)$$

Šis taškas gaunamas kaip atsakas iš signalo taškų S(C) į minimalaus atstumo kriterijų Euklido I matmenų erdvėje. S(C) yra laikomas S projekcija paralelinėje erdvėje, sugeneruotoje I ašių, kuri neatsiliepia į nulinius c_i. Pagal šiuos kriterijus, norint išvengti pasikartojančių klaidų, reikia, kad taškai S(C) būtų skirtingi. [19]



2.9 pav. CRM esant D = 2 gauto iš 4-PSK nustatant signalo sukimąsi pavyzdys.

Apibrėžimas: kolizijoms atspari moduliacija (CRM) yra D-matmens signalas, su nustatyta S tikimybe. Bet kuri projekcija į kurią nors koordinatę, esančią paralelinėje erdvėje, yra signalas su tuo pačių išnykusių taškų kiekiu. |S(C)| = |S| - visoms C reikšmėms, kurios nėra lygios nuliui. [19]

S vektoriai taip pat privalo turėti skirtingas koordinates, arba tiksliau D dydžio „Hamming“ nuotolį tarp bet kurių vektorių porų. [19]

2.8 Bevielių tinklų su polinkiu į koliziją, pakartotinos būsenos įranga

2.8.1 Apibendrinimas

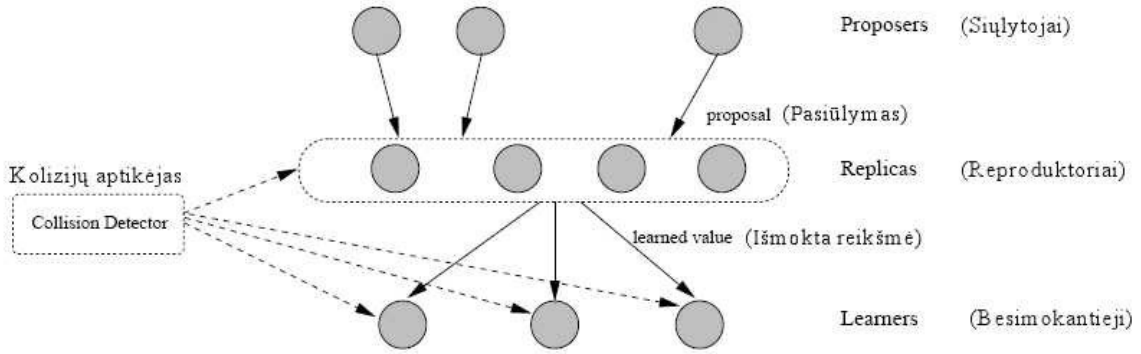
Tinkluose, kuriuose yra didelė tikimybė klaidų, pakartotinos būsenos įranga ilgą laiką buvo naudojama kaip priemonė tolerancijos klaidų aptarnavimui. Išplečiame pakartotinos įrangos būsenos paradigmą į bevielius „ad hoc“ tinklus, kurie yra linkę į kolizijas, pranešimo praradimus, bei darbo nutrūkimus. Mūsų algoritmas yra specialiai pritaikytas „ad hoc“ tinklams, kur tinklo dalyviai nežinomo prioriteto, bei prisitaikantys prie kintančio tinklo dalyvių skaičiaus. Galiausiai mūsų algoritmas yra efektyvus dėl nekintamo dydžio protokolo pranešimų, bei dėl pakartotinos įrangos būsenos, pastoviu laiku siunčiamų atsakymų. [20]

2.8.2 Būsenos be kolizijų įranga

Parodysime kaip perdaryti gerai žinomą įrangą, kurią galėtume naudoti kolizijos polinkio aplinkose. Pristatome būsenos be kolizijų įrangą, tai yra įranga, kuri gali aptikti kolizijas, praradus įeinančia ar išeinančią informaciją. [20]

Formaliai, būsenos be kolizijų įranga yra „automatas“ susidedantis iš: (i) juridinės būsenos rinkinio, V, (ii) pradinės būsenos, v_0, (iii) pasiūlymų rinkinio, „inputs“, (iii) rinkinio reikšmių kurias reikia išmokti, „outputs“, bei (iv) persiuntimo funkcijos, δ kuri gaunama iš $V \times P("inputs") \cup \{\pm\}$ į $V \times "outputs"$. Pastebėkite, kad esant kolizijai δ galima atnaujinti būseną.

Kiekvienas, dalyvaujantis mazgas (node), yra priskirtas atlikti vieną iš trijų rolių: siūlytojo (proposer), reproduktoriaus (replica), ar mokymosi (learner). Ši įranga dirba „rato“ modeliu. Kiekviena ratu einanti įranga susideda iš trijų dalių: (i) siūlytojai (proposer), siuntimo pasiūlymai, *p Є "inputs"* įrangai. Reproduktoriai (ii) (replica), gauna pasiūlymus atnaujinti būseną bei išsiuntimo reikšmes. Jas išsiunčia. Besimokantieji (iii) (learners), gauna išsiųstas reikšmes iš būsenos įrangos, $l \in outputs \cup \{\pm\}$, jas persiunčia. Būsenos be kolizijų įrangos, komponentai aprašyti 2.10 pav. [20]



2.10 pav. Būsenos be kolizijų įrangos, komponentai.

Formaliai sakome, kad algoritmas A įgyvendina būsenos įrangą, jeigu jos rezultatas nuoseklus ir atitinka automatiškai įvykdytą rezultatą. Tai yra kiekvienai būsenos įrangai ciklas r yra būsena $v_r \in V$ ir siūlymų rinkinys $Q_r \subseteq inputs \cup \{\pm\}$, kuris patenkina sekančias ypatybes: (i) būsenos eiliškumas yra nuoseklus kartu su perėjimo funkcija, t.y. $v_{r+1} = \delta(v_r, Q_r)$. Išmoktos (ii) reikšmės yra pastovios kartu su persiuntimo funkcijomis, t.y. jeigu $l \neq \pm$ išmoktos būsenos įrangai r, tada $l = \delta(v_r, Q_T)$ išeinančioji žinutė. (iii) kiekvienas Q_T yra siūlymų poaibis įrangos būsenai esant apie r. Jeigu yra keli pasiūlymai r ir $\notin Q_r$, tuomet įvyksta kolizija, $\pm \in Q_r$. (iv) esant nesėkmingam r, būsenos įranga neišmoksta reikšmės nei per nei po ciklo r. Kol kas mūsų nustatymai apima tik saugumo ypatybes. Taip pat reikalaujama, kad: (v) kiekvienas būsenos įrangos ciklas būtų vykdomas kaip konstanta nuo komunikavimo ciklų. (vi) po galimos kolizijos (po ciklo r_{ecf}) ir po galutinio patikslinimo (po ciklo r_{acc}) būsenos įranga nebepatiria kolizijų. (vii), jei bent vienas reproduktorius išlieka nesusikirtęs, tai ir būsenos įranga nesusikerta. [20]

2.9 Bevielių sensorinių tinklų, kolizijų vengimas taikomosios programos pagrindu

2.9.1 Apibendrinimas

Bevieliai sensoriniai tinklai charakterizuojami pagal prijungtų, mažų, mažos galios tinklo mazgų (node), renkančių informaciją apie fizinę aplinką, kiekį. Sutampantys persiuntimai atsiranda esant gerai žinomai paslėpto terminalo problemai, jie sudaro kolizinę situaciją, kurios metu persiunčiami paketai sugadinami. Kadangi sugadinti paketai turi būti persiunčiami, kolizijos prideda papildomą apkrovą energijos suvaržytai sistemai. Pristatome programos pagrindu pagrįstą priėjimą prie kolizijų vengimo. Siūlome du specifinius algoritmus: [21]

- pirmasis stebi TCP perkrovos algoritmą ir esant kolizijai pritaiko siuntimo spartą;
- antrasis, norint sumažinti kolizijų kiekį, sukeičia paketų persiuntimo laikus.

Mes įvertinome abu algoritmus, atliekant simuliacijas, ir mūsų rezultatai rodo, kad toks priėjimas gali sumažinti priverstinai sukeltą kolizijų kiekį retransliuojant, bei sutaupyti energijos iki 50%. [21]

2.9.2 Kolizijų vengimas

Paprastas ir nesudėtingas priėjimas kolizijų sumažinimui yra tikimybės schemos naudojimas. Laikas išskirstomas į stambius lūžius, kur kiekvienas šaltinis atsitiktinai pasirenka siuntimo laiką. Kadangi lūžiai, lyginant su paketo siuntimo laiku, yra gan dideli, dviejų šaltinių persidengimas yra mažai tikėtinas. Šią schemą naudoja MOAP (Multihop Over the Air Programming) protokolas.[21]

Pagal aprašymą, kiekvienas šaltinis gali siųsti tik vieną paketą per lūžį. Dėl to sugaištas laikas tiesiogiai proporcingas lūžio dydžiui. Palyginus didelis lūžis mažiems tinklams labai skiriasi nuo lūžio dideliuose tinkluose. [21]

Sprendimas turėtų būti pritaikomas tada, kai galima sumažinti sugaišimą, bet taip pat reaguoti į įvykusias kolizijas. Persiuntimų kiekio padidinamas idealiai, be žymios gaišties, sumažina būsimų kolizijų kiekio tikimybę. [21]

Pritaikomos kolizijų schemos gali būti išskirstytos į du tipus: „source-based“ bei „receiver-based“.

„Source-based“ metode nėra aiškaus grįžtamojo ryšio iš šaltinio į imtuvą, išskyrus tą dalį, kuri aprašyta kaip dalis protokolo. Kaip pavyzdys „source-based“ metodo yra Van Jacobson TCP apkrovos

išvengimas. TCP perkrovos vengimas naudoja TCP protokolų dublikatus ACKS, bei laiko intervalus, per kuriuos padaromos išvados į tinklo perkrovą ir į tai sureaguojama. [21]

„Receiver-based“ metodas leidžia imtuvui išsiųsti grįžtamąjį ryšį siųstuvui, tai vyksta kaip reakcija į koliziją. Kai imtuvas aptinka informaciją susijusią su kolizijos vengimu, persiunčia informaciją šaltiniui. [21]

2.9.3 „Source-Based“ kolizijų vengimas

Perkrovos vengimas tinklo paketų pasikeitime yra panašu į kolizijų vengimą bevieliuose tinkluose. Abiem atvejais šaltinis ima savo duomenų srautą, net esant nenormaliai veiklai (perkrovai ar kolizijai). Esant kolizijos vengimui, imamas duomenų srautas pakeičia duomenų atsiuntimo periodą, tokiu būdu išvengiama kolizijų, susidariusių dėl persidengiančių periodų. Didelei apkrovai esančiai imtuve leidžiama išsisklaidyti, kai sulėtiname šaltinį. Šis rezultatas netiesiogiai numano, kad visas šaltinio srautas turi ribotą ilgį. Galiausiai po kurio tai laiko šaltinis baigsis. [21]

Esant TCP perkrovos protokolui, netiesioginis konstravimas numano, kad prarasti paketai buvo dėl perkrovos. Šis manymas bevieliams tinklams netinkamas. Duomenys gali būti prarasti dėl imtuvo kaltės, esant kolizijai ar nuorodos praradimui. Nuorodos praradimas apima daugybę priežasčių, susijusių su paketų sugadinimu esant atspindžiams, aplinkos pasikeitimui, persidengimui ir t.t. Mes teigiame, kad šaltinis turėtų nereaguoti į nuorodų praradimus, kadangi jie nėra apibrėžiami ir yra už šaltinio kontrolės ribų. Kita vertus koliziją galima išvengti, jeigu šaltinis imsis papildomų veiksmų. Mūsų „source-based“ metodas neskiria skirtumo tarp nuorodos ir kolizijos praradimų. Šį metodą mes vadiname „uninfomed TCP-like collision avoidance“. [21]

2.9.4 „Receiver-Based“ kolizijų vengimas

Kaip minėjome prieš tai, MOAP protokolas siunčia NACK šaltiniui informaciją apie prarastus paketus. Jeigu imtuvas gali atskirti patirtus nuostolius, esant priverstinei kolizijai bei nuorodos praradimui, tada šaltinis gali naudoti arba atmesti šią informaciją, priimant geriausią sprendimą duomenų perdavimo spartinimui. Šį metodą vadiname „informed TCP-like collision avoidance“ ir galime pažymėti, kad našumas priklauso nuo kolizijos aptikimo schemos tikslumo. [21]

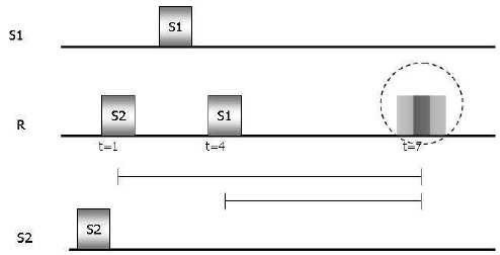
Darant prielaidą, kad visi šaltiniai persiunčia paketus tuo pačiu dažniu 1/T t.y. per laiką T kiekvienas aktyvus šaltinis siunčia paketus. Tada imtuvas tuo pačiu metu gali stebėti iš visų šaltinių atvykstančius paketus, bei aptikti didelius neaktyvumo periodus, įskaitant duotą laiko rėmą. Jeigu

paketas prarandamas dėl kolizijos, imtuvas gali atsiųsti didžiausią neaktyvumo periodą pradiniam šaltiniui. Tuo metu šaltinis, norėdamas sumažinti tolimesnes kolizijas, gali bandyti siųsti. Iš esmės šaltinis bando sukeisti persiuntimus išlaikant duomenų srautą. Šį metodą vadiname „Phase-offset collision avoidance“. [21]

2.9.5 Kolizijų aptikimas

Kolizijos aptikimas imtuve yra nesudėtingas, kaip ir pažeistų paketų tikrinimas. Pažeistas paketas gali būti priežastis nuorodų praradimui bei kolizijos atsiradimui. Esant sugadintam paketui ne visada galima nustatyti jo kilmę. Vienintelę išvadą, kurią galima padaryti yra ta, kad buvo paketo praradimas. [21]

Idealiu atveju imtuvas žinotų tikslų šaltinio paketų siuntimo laiką. Jeigu keli paketai būtų siunčiami tuo pačiu metu, imtuvas galėtų nustatyti įvyksiančią koliziją, bei informuoti šaltinį. Būtų idealus sprendimas į protokolą pridėti determinizmo. Daugumoje sąsajų, šaltinis išsiųs paketą ir nustatys laikmatį kitam siuntimui. Jeigu mes nustatysime sekančio siuntimo laiką prieš išsiunčiant paketą, tada galėsime įdėti šią informaciją į naudingą apkrovą (payload). Informavus imtuvą, paketai laukiami apytiksliai nurodytu laiku. Kiekvienam šaltiniui laikantis šių taisyklių, imtuvas gali susidaryti atvykstančių paketų eilę. Jeigu pirminis šaltinis ir, bet kokie galiniai šaltiniai nustato, kad paketų atvykimas bus apytiksliai tuo pačiu metu, galime iškelti hipotezę, kad laiku atvykęs sugadintas paketas bus kolizijos rezultatas. 2.11 pvz. Apibudina šią situaciją. [21]



2.11 pav. Kolizijos aptikimas imtuve.

Šaltinis S_2 siunčia, kai t = 1 ir informuoja imtuvą, kad sekantis siuntimas bus t = 7. Šaltinis S_1 siunčia, kai t = 4 ir informuoja imtuvą, kad sekantis siuntimas bus, kai t = 7. Jeigu nors vienas blogas paketas atvyks, kai t = 7, imtuvas užfiksuos koliziją. [21]

CSMA MAC turi „backoff“ mechanizmą, leidžiantį išvengti kolizijų. Šis mechanizmas gali pradėti nedeterminuotą vėlinimą laiko, kuriuo šaltinis išsiunčia į MAC sluoksnį ir nustato laikmatį, ir

laiko, kai imtuvas gauna paketą, bei sudaro laiko poslinkį. Norėdamas žinoti tikslesnį sekančio paketo persiuntimo laiką, šaltinis gali išsiųsti į imtuvą dispersiją kartu su siuntimo intervalu. [21]

Siuntimo intervalas ir siuntimo dispersija nurodo numatomą paketo atsiuntimo laiką. Kolizija apibrėžta daugialypiu laiko persidengimu. Kadangi paketai gali būti gaunami bet kuriuo laiku, tam tikrame intervale yra įmanoma, kad galima persidengimo kolizija neįvyks. Taigi kolizijos aptikimo modelis negali būti taikomas tiksliam būsimų kolizijų nustatymui. Tačiau galimas panaudojimas „post-fakto“ kolizijos nustatymui. [21]

Pagal anksčiau minėtą medžiagą, parastas paketas klasifikuojamas kaip priverstinės kolizijos praradimas. Pasibaigus tikėtinam atvykimo laikui, plius žinant siuntimo subtilybes, susidarė išvados:

- jokių duomenų paketų iš pirminio šaltinio negauta;
- atvykimo laikas parodė persidengimą tarp pirminio šaltinio atsiuntimo laiko ir galinio šaltinio atsiuntimo laiko;
- gautas bent vienas sugadintas paketas. [21]

Pastebime, kad mūsų kolizijos nustatymo mechanizmas yra *numatytojas* – ne visada garantuojamas teisingas atsakymas. Neatkeliavus šaltinio paketui lauktu laiku, galime paketą laikyti susidūrusiu. Galimas variantas, kad aptikta kolizija buvo nuspėta susidūrimui su kitu šaltiniu, bet ištiktųjų buvo sugadintas dėl triukšmų (nuorodos praradimo). Tokiu atveju mūsų *numatytojas* pateiktų teigiamą,- neigiamą atsakymą. [21]

Negavus paketų lauktu laiku, nežinant jo šaltinio, nustatomas paketo praradimas dėl radijo skilimo. Jeigu kiti šaltiniai buvo numatę siųsti tuo laiko momentu, paketų likimas neaiškus. Paketas gali būti prarastas dėl radijo sklidimo problemų ar kolizijos. Jeigu kolizija atsiranda preambulėje ar pradinio simbolio, tada imtuvas negalės užrakinti siuntėją ir nebus gauta jokių paketų. Tai vestų į „false negatyve“ atsakymą. [21]

Galiausiai *numatytojas* yra tinkamas tik atvykstančių paketų laiko eilei sudaryti (įmontuotas į imtuvą). Esant bet kokiems šaltinio paketų praradimams, kartu su paketu prarandama ir informacija apie būsimus paketų laikus. Šiuo atveju imtuvas gali neteisingai pažymėti koliziją kaip nuorodos praradimą. Mūsų kolizijos aptikimo tikslumas yra tiesiai proporcingas nuorodos kokybei tarp imtuvo, bei jį supančių šaltinių. [21]

2.10 Greito kolizijos sprendimo (FCR) MAC algoritmas bevieliams tinklams

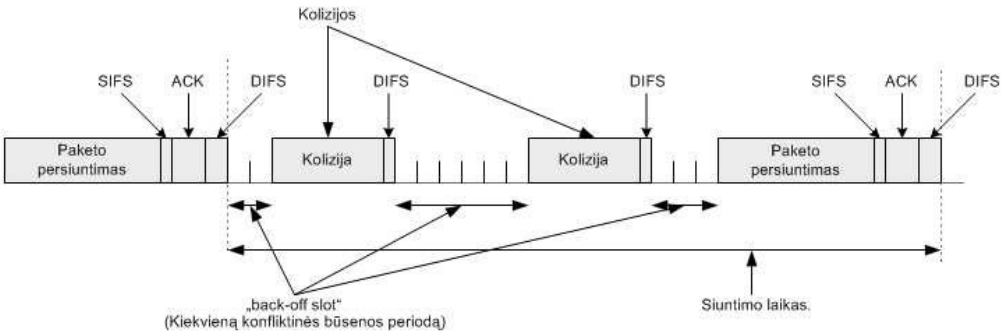
2.10.1 Apibendrinimas

Norint padidinti konflikto būsena paremto MAC protokolo pralaidumą ir sumažinti perkrovimus, (tokius kaip paketų kolizija ir „idle slot“) turi būti panaudotas efektyvus kolizijų skaidymo algoritmas kiekviename konflikto būsenos cikle. Šiam metodui buvo pasiūlyta daug originalių algoritmų ir pakoreguoti abejotinam lango dydžiui taip pat atsitiktinai parenkamos „backoff“ reikšmes. „Out-band-busy-tone“ signalizavimas yra aktyviai naudojamas informuoti kitus dėl užimto kanalo statuso. Abejotina konflikto būsenos išsiųstų paketų informacija taip pat gali būti panaudota kaip pagalba kolizijos sprendimui. Cali, Conti ir Gregori pasiūlė įdomų algoritmą, kuris pagerina IEEE802.11 MAC protokolo vykdymą. Pagrindinė šio algoritmo idėja yra abejotinų lango dydžių pakoregavimas kiekviename įrenginyje, kuris paremtas apskaičiuotų aktyvių įrenginių skaičiumi. Tačiau tikrame bevieliame tinkle aktyvių įrenginių panaikinimas nėra lengva užduotis. [22]

Nors ir buvo pasiūlyta daug naujų, konflikto būsena paremtų MAC protokolo idėjų, nėra lengva patenkinti visus norimus nustatymus ir tuo pačiu išsaugoti realizavimo paprastumą bevieliuose tinkluose. Čia mes siūlome išplatintą, naują, veiksmingą, konflikto būsena paremtą MAC algoritmą, kuris yra tarsi greito kolizijos sprendimo FCR algoritmas. Pastebime, kad pagrindinis daugumos kolizijų paremtų MAC algoritmų trūkumas kyla iš paketų kolizijos ir išnaudotų „slots“ dėl „back-off“ kiekviename abejotiname konflikto būsenos cikle. Pvz. IEEE802.11 MAC protokolas, kai aktyvių stotelių skaičius padidėja, yra per daug atitolusių stotelių su mažais abejotinais konflikto būsenos langais, taigi daug retransliuotų paketų vis vien susidurs ateityje, o tai sulėtins kolizijų išskaidymą. FCR algoritmas, didindamas susiduriančių ir atidėtų abejotinų stotelių langų dydžius dėl prioriteto praradimo procedūroje, mėgina greitai išspręsti kolizijas, t.y. mes sugalvojame algoritmą, kuris perskirstys jų „back off“ laiko intervalus tam, kad išvengti susidūrimų ateityje. Neveikiančių „lot“ mažinimui, FCR algoritmas kiekvienai stotelei suteikia trumpą neveikiančių „back off“ periodą su sėkmingu paketo perdavimu. Be to, kai stotelė aptinkama su neveikiančiu „slot“, ji pradeda eksponentiškai mažinti „back off“ laiko intervalus, o ne tiesiškai kaip IEEE 802.11 MAC. Mes siekiame pasiūlyti MAC paremta konflikto būsena, kuri yra lengvai realizuojama realiuose bevieliuose tinkluose. [22]

2.10.2 Greitas kolizijų sprendimas: pagrindinė idėja

Yra du pagrindiniai faktoriai, įtakojantys IEEE 802.11 MAC protokolo našumą: Nesėkmingi persiuntimai (atsiradę dėl paketų kolizijos), bei neveiksnūs „slots“, atsirandantys atsitraukiant kiekvienam turinio ciklui. 2.12 pav. [22]



2.12 pav. Pagrindinės CSMA/CA operacijos.

Esant dideliam srautui (visos M stotys turi paketų persiuntimą) ir numanymui apie ergodiškumą, galima gauti pralaidumo išraiškas. 3.12 pav. galime išnagrinėti vieno persiuntimo ciklą. [22]

$$p = \frac{\overline{m}}{E[N_c](E[B_c]\cdot t_s + \overline{m} + DIFS) + (E[B_c]\cdot t_s + \overline{m} + SIFS + ACK + DIFS)} \quad ,$$

kur $E[N_c]$ yra vidutinis kolizijų kiekis (virtualaus siuntimo cikle),

$E[B_c]$ vidutinis kiekvieno periodo „idle slot“ kiekis., [22]

t_s „slot“ ilgis.,

$\overline{m}$ vidutinis paketų ilgis.

Iš to galime iššvelgti geriausią situaciją 2.12 pav., kuri turi didžiausią pralaidumą: po sėkmingo paketo persiuntimo turi sekti kitas paketo siuntimas nenaudojant „overhead“, kur mūsų atveju, $E[N_c]=0$, $E[B_c]=0$, ir tada pralaidumas būtų: [22]

$$p_{best} = \frac{\overline{m}}{\left(\overline{m} + SIFS + ACK + DIFS\right)}$$

Tai gali būti pasiekta tik dėl puikaus laiko paskirstymo. Šioje situacijoje kiekvienas prietaisas turi galimybę paketų persiuntimui, $p_{trans}(i) = \begin{cases} 1 \\ 0 \end{cases}$; 1 – jeigu prietaisas *i* persiunčia paketą per esamą periodą. 0 – kitais atvejais.[22]

Kai kuriose schemose galime sakyti, kad laiko intervalas parenkamas atsitiktinai, tada paketo persiuntimo tikimybė iš prietaiso *i* priklausytų nuo „back-off“ laiko intervalo. [22]

$$p_{trans}(i) = \frac{1}{(B_i + 1)}$$

Kur B_i yra prietaiso *i* „back-off“ laiko intervalas. [22]

Tai reiškia, kad jeigu prietaisas *i* turi „back-off“ laiko intervalą nustatytą ant 0. Esant tokiai situacijai prietaisas *i* nedelsiant persiųs paketą. Tai gali būti interpretuojama kaip prietaiso *i* galimybė persiųsti paketą. Jeigu prietaiso *i* „back-off“ laiko intervalas nustatytas ties ∞ (esant 0) tai rodo apie paketo persiuntimo galimybę. [22]

$B_i = \begin{cases} 0 \\ \infty \end{cases}$ 0 - prietaisui *i* persiunčiant paketą. ∞ - kitais atvejais.

Išvada: jeigu išeitų sukurti konflikto būsena pagrįstą MAC algoritmu, kuris prietaisui priskiria „back-off“ laiko intervalą 0, tuo tarpu kitiem prietaisam, kiekvienam periodui priskiria „back-off“ laiko intervalą ∞, būtų sudaroma ideali laiko seka maksimaliam pralaidumui pasiekti. Šitoks MAC algoritmas tikrovėje neįgyvendinamas. Tačiau tai leidžia susidaryti pagrindinę idėją kaip padidinti pralaidumą MAC protokole. Vienas iš būdų tai įgyvendinti - yra suprojektuoti MAC protokolą taip, kad būtų stengiamasi priartėti prie idealaus laiko suskirstymų. [22]

2.10.3 Greito kolizijos sprendimo (FCR) algoritmas

Pagrindinis sunkumas IEEE 802.11 MAC protokole yra lėtas kolizijų sprendimas, didėjant vartotojų kiekiui. Aktyvus prietaisas gali būti dviejų būsenų - siuntimo būsena bei atidėjimo būsena. Prietaisui išsiunčiant paketą galima viena ar abi būsenos: sėkmingas paketo išsiuntimas ar kolizija. Prietaisas bus vienoje iš trijų būsenų kiekvieną konflikto būsenos periodą: esant sėkmingai paketo perdavimo būsenai, esant kolizijos būsenai, esant atidėjimo būsenai. Dažniausiai konflikto būsena pagrįstose MAC algoritmuose nėra skirtingiems prietaisams skirtingų konflikto būsenos lango dydžių, o bei aptikus „idle“ prietaisą „back-off“ laiko intervalas bus sumažintas per vieną „slot“. Pasiūlytame greito kolizijos sprendimo (FCR) algoritme pakeisime konflikto būsenos lango dydį kiekvienam prietaisui ir pakartotinai paleisime „back-off“ laiko intervalą. Visi potencialūs siuntimui prietaisai su paleistais laiko intervalais ateityje išvengs potencialių kolizijų, tokiu būdu sprendžiamas greitas kolizijų vengimas. [22]

3. KOLIZINIŲ SITUACIJŲ SUDARYMAS

3.1 Bandymai sukurti koliziją WLAN tinkle

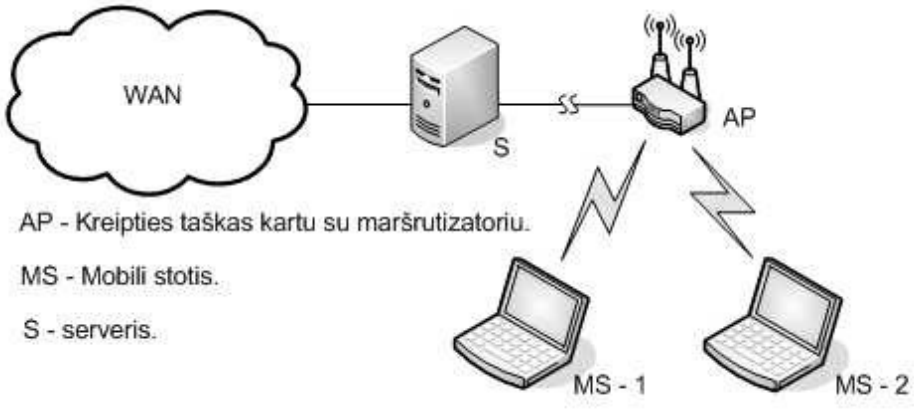
Bandymo tikslas: sukurti kolizinę situaciją.

Prietaisai bandymo atlikimui:

1. Nešiojamas kompiuteris Lenovo, aprašymas priede Nr. 1
2. Nešiojamas kompiuteris Vector, aprašymas priede Nr. 2
3. Kreipties taškas kartu su maršrutizatoriu, Linksys Wap11, aprašymas priede Nr. 3

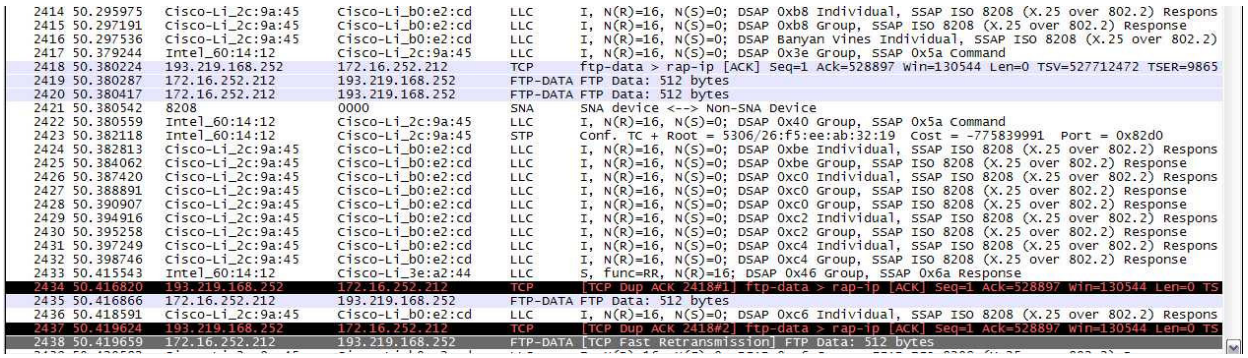
Bandymams parinktas 20MB failas. Naudojame centralizuotą WLAN tinklo tipologiją.

Parinkta situacija, eiga: bandymą vykdomoje aplinkoje paliekamas veikti tik vienas AP. AP padedamas tarp dviejų nešiojamųjų kompiuterių. Kompiuteriai atitraukiami nuo AP maksimaliu atstumu neprarandant ryšio su AP (bandoma sukurti paslėpto terminalo problemą). Į kompiuterius įdiegiamas Wireshark srauto analizatorius. Abiejuose kompiuteriuose startuojama programa Wireshark. MS – 1 pradeda siųsti 20MB paketą į FTP serverį S. Po 10 sek. MS – 2 pradeda taip pat siųsti 20MB paketą į FTP serverį S. Baigus abiem MS duomenų persiuntimą, sustabdomas Wireshark veikimas.



3.1 WLAN bandymo struktūrinė schema.

Rezultatai: failai į serverį S persiųsti nesugadinti. MS – 2 turintis geresnį ryšį su AP pirmasis baigia siuntimą. Gauti Wireshark duomenys (log).



2414	50.295975	Cisco-Li_2c:9a:45	Cisco-Li_b0:e2:cd	LLC	I, N(R)=16, N(S)=0; DSAP 0xb8 Individual, SSAP ISO 8208 (X.25 over 802.2) Respons
2415	50.297191	Cisco-Li_2c:9a:45	Cisco-Li_b0:e2:cd	LLC	I, N(R)=16, N(S)=0; DSAP 0xb8 Group, SSAP ISO 8208 (X.25 over 802.2) Response
2416	50.297536	Cisco-Li_2c:9a:45	Cisco-Li_b0:e2:cd	LLC	I, N(R)=16, N(S)=0; DSAP Banyan vines Individual, SSAP ISO 8208 (X.25 over 802.2)
2417	50.379244	Intel_60:14:12	Cisco-Li_2c:9a:45	LLC	I, N(R)=16, N(S)=0; DSAP 0x3e Group, SSAP 0x5a Command
2418	50.380224	193.219.168.252	172.16.252.212	TCP	ftp-data > rap-ip [ACK] Seq=1 Ack=528897 Win=130544 Len=0 TSV=527712472 TSER=9865
2419	50.380287	172.16.252.212	193.219.168.252	FTP-DATA	FTP Data: 512 bytes
2420	50.380417	172.16.252.212	193.219.168.252	FTP-DATA	FTP Data: 512 bytes
2421	50.380542	8208	0000	SNA	SNA device <--> Non-SNA Device
2422	50.380559	Intel_60:14:12	Cisco-Li_2c:9a:45	LLC	I, N(R)=16, N(S)=0; DSAP 0x40 Group, SSAP 0x5a Command
2423	50.382118	Intel_60:14:12	Cisco-Li_2c:9a:45	STP	Conf. TC + Root = 5306/26:f5:ee:ab:32:19 Cost = -775839991 Port = 0x82d0
2424	50.382813	Cisco-Li_2c:9a:45	Cisco-Li_b0:e2:cd	LLC	I, N(R)=16, N(S)=0; DSAP 0xbe Individual, SSAP ISO 8208 (X.25 over 802.2) Respons
2425	50.384062	Cisco-Li_2c:9a:45	Cisco-Li_b0:e2:cd	LLC	I, N(R)=16, N(S)=0; DSAP 0xbe Group, SSAP ISO 8208 (X.25 over 802.2) Response
2426	50.387420	Cisco-Li_2c:9a:45	Cisco-Li_b0:e2:cd	LLC	I, N(R)=16, N(S)=0; DSAP 0xc0 Individual, SSAP ISO 8208 (X.25 over 802.2) Respons
2427	50.388891	Cisco-Li_2c:9a:45	Cisco-Li_b0:e2:cd	LLC	I, N(R)=16, N(S)=0; DSAP 0xc0 Group, SSAP ISO 8208 (X.25 over 802.2) Response
2428	50.390907	Cisco-Li_2c:9a:45	Cisco-Li_b0:e2:cd	LLC	I, N(R)=16, N(S)=0; DSAP 0xc0 Group, SSAP ISO 8208 (X.25 over 802.2) Response
2429	50.394916	Cisco-Li_2c:9a:45	Cisco-Li_b0:e2:cd	LLC	I, N(R)=16, N(S)=0; DSAP 0xc2 Individual, SSAP ISO 8208 (X.25 over 802.2) Respons
2430	50.395258	Cisco-Li_2c:9a:45	Cisco-Li_b0:e2:cd	LLC	I, N(R)=16, N(S)=0; DSAP 0xc2 Group, SSAP ISO 8208 (X.25 over 802.2) Response
2431	50.397249	Cisco-Li_2c:9a:45	Cisco-Li_b0:e2:cd	LLC	I, N(R)=16, N(S)=0; DSAP 0xc4 Individual, SSAP ISO 8208 (X.25 over 802.2) Respons
2432	50.398746	Cisco-Li_2c:9a:45	Cisco-Li_b0:e2:cd	LLC	I, N(R)=16, N(S)=0; DSAP 0xc4 Group, SSAP ISO 8208 (X.25 over 802.2) Response
2433	50.415543	Intel_60:14:12	Cisco-Li_3e:a2:44	LLC	S, func=RR, N(R)=16; DSAP 0x46 Group, SSAP 0x6a Response
2434	50.416820	193.219.168.252	172.16.252.212	TCP	[TCP Dup ACK 2418#1] ftp-data > rap-ip [ACK] Seq=1 Ack=528897 Win=130544 Len=0 TS
2435	50.416866	172.16.252.212	193.219.168.252	FTP-DATA	FTP Data: 512 bytes
2436	50.418591	Cisco-Li_2c:9a:45	Cisco-Li_b0:e2:cd	LLC	I, N(R)=16, N(S)=0; DSAP 0xc6 Individual, SSAP ISO 8208 (X.25 over 802.2) Respons
2437	50.419624	193.219.168.252	172.16.252.212	TCP	[TCP Dup ACK 2418#2] ftp-data > rap-ip [ACK] Seq=1 Ack=528897 Win=130544 Len=0 TS
2438	50.419659	172.16.252.212	193.219.168.252	FTP-DATA	[TCP Fast Retransmission] FTP Data: 512 bytes

3.2 pav. Dalis *.log failo.

Išvada: išnagrinėjus *log failą buvo nustatyta, kad MS - 1 siunčiant informaciją į S, bei MS – 2 nepradėjus siųsti, kolizijų nepatiria. Po 10sek. pradėjus MS – 2 siuntimą teigiame, kad tinkle atsirado kolizijos. Sugadinami paketai, atsiranda jų persiuntimai.

3.2 Bandymai sukurti koliziją Bluetooth tinkle.

Bandymo tikslas: sukurti kolizinę situaciją.

Prietaisai bandymo atlikimui:

1. Nešiojamas kompiuteris Dell Latitude D630 NB (schemoje MS 1), aprašymas priede Nr. 4
2. Mobilusis telefonas Sony Ericsson T610 (schemoje MS 2), aprašymas priede Nr. 5
3. Bluetooth adapteris Cambird BTD-002 (schemoje MS 3).

Bandymams parinktas 1MB bei 10MB failai. Naudojame populiariąją „Ad hoc“ tinklo tipologiją. MS 1, MS 2 ir MS 3 išdėstyti netoli (nedaugiau 1m atstumu) vienas kito. Tinklo prietaisų prioritetų pasiskirstymas:

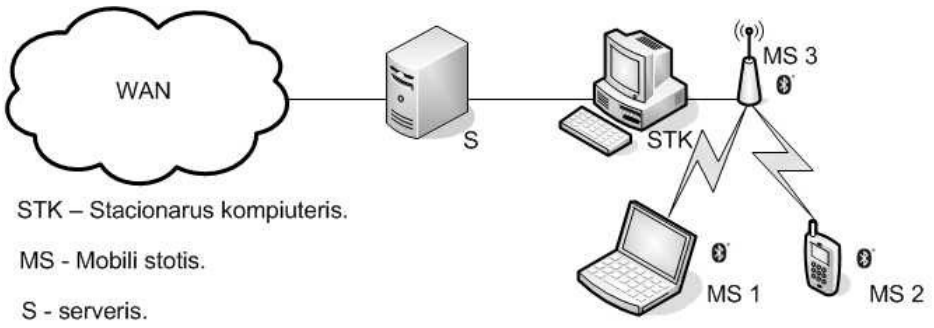
MS 1 – priskirtasis įrenginys (slave).

MS 2 – priskirtasis įrenginys (slave).

MS 3 – pagrindinis įrenginys (master).

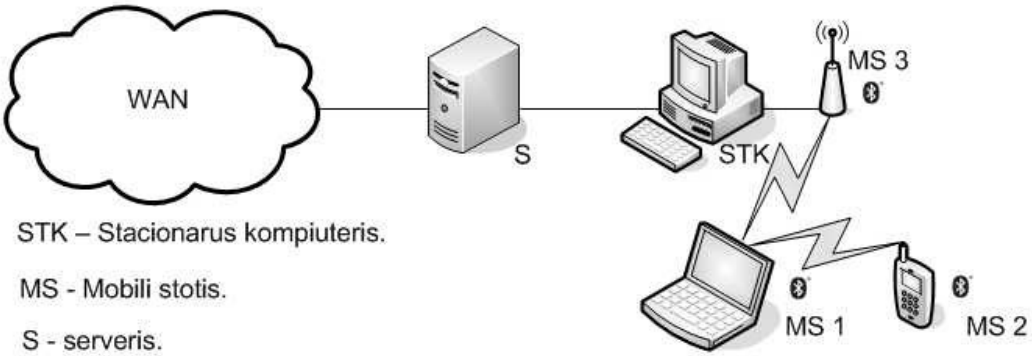
Esant brangiai „third party“ įrangai, bendravimas oru tarp MS įrenginių stebimas nebus. Sudaromos 2 situacijos:

a) Išjungiamas MS 2. MS1 siunčia 10MB duomenų kiekį į MS 3. MS 1 siunčiant duomenis į MS 3 įjungiamas MS 2. Įjungus bandoma perduoti 1MB duomenų kiekį į MS 3. 2.3 pav.



3.3 pav. Duomenų perdavimas į MS 3.

b) Išjungiamas MS 2. MS1 siunčia 10MB duomenų kiekį į MS 3. MS 1 siunčiant duomenis į MS 3 įjungiamas MS 2. Įjungus bandoma perduoti 1MB duomenų kiekį į MS 1. 2.4 pav.



3.4 pav. Duomenų perdavimas į MS 3 ir MS 1.

Rezultatai:

a) įjungus MS 2 aplinkiniai Bluetooth prietaisai buvo rasti. Bandant iš MS 2 persiųsti 1MB duomenų kiekį į MS 3 siuntimas nepavykdavo.

b) įjungus MS 2 aplinkiniai Bluetooth prietaisai buvo rasti. Bandant iš MS 2 persiųsti 1MB duomenų kiekį į MS 1 siuntimas įvykdavo.

Išvada: darome prielaidą, kad esant a) situacijai šokinėjantis dažnis (bandant išvengto kolizijos) patyrė koliziją (persiuntimas kitu kanalu neįvyko). Neatmetama, kad „Synchrinization Profile“ ne iki galo įvykdė sinchronizavimą, ar dėl nežinomų priežasčių nebuvo panaudotas TDD (Time Division Duplex) siuntimo vietai (slot) užimti.

Esant situacijai b), viskas praėjo sklandžiai, kolizija neaptikta.

IŠVADOS IR SIŪLYMAI

- Išnagrinėta labiausiai šiuo metu paplitę bevieliai tinklų tipai (WLAN, Bluetooth, Zigbee). Išanalizuota jų sandara, veikimo principai, naudojamos tipologijos.
- Apžvelgti jau taikomi, bei taikytini (esant poreikiui), kolizinių situacijų vengimo būdai.
- Eksperimentiškai įvykdytos dvi populiariausios (WLAN centralizuotoje tipologijoje „paslėpto terminalo“ problema, bei Bluetooth „ad hoc“ tipologijos šokinėjančio dažnio problema) kolizinės situacijos. Gauti rezultatai:

  1. (Wlan) išnagrinėjus *log failą buvo nustatyta, kad MS - 1 siunčiant informaciją į S, bei MS – 2 nepradėjus siųsti, kolizijų nepatiria. Po 10sek. pradėjus MS – 2 siuntimą teigiame, kad tinkle atsirado kolizijos. Sugadinami paketai, atsiranda jų persiuntimai.
  2. (Bluetooth) darome prielaidą, kad esant a) situacijai šokinėjantis dažnis (bandant išvengto kolizijos) patyrė koliziją (persiuntimas kitu kanalu neįvyko). Neatmetama, kad „Synchrinization Profile“ ne iki galo įvykdė sinchronizavimą, ar dėl nežinomų priežasčių nebuvo panaudotas TDD (Time Division Duplex) siuntimo vietai (slot) užimti.

  Esant situacijai b), viskas praėjo sklandžiai, kolizija neaptikta.

Remiantis iškeltais uždaviniais, gautais rezultatais, pasiūlymą galima išdėstyti sekančia tvarka:

Žinant, kad kolizijų kiekis sparčiai išauga tinkle atsirandant naujiems prietaisams, siūlau kolizijos vengimo sąvoką perkelti į fizinį lygmeny, t.y.:

Norint sumažinti kolizijų kiekį, pagerinti tinklo pralaidumą, pratęsti mobilaus prietaiso energijos šaltinio gyvavimo laiką (energijos suvartojimo kiekį) siūlau „paskleisti“ vartotojus (sudarinėti kiek įmanoma mažiau vartotojų turinčius tinklus).

Pvz. nagrinėtame WLAN centralizuotos tipologijos koliziniame atvejyje, reikia išdėstyti daugiau kreipties taškų AP. Taip būtų mažinamas vartotojų kiekis vienam AP.

Bluetooth „ad hoc“ tipologijai, turint kitokią struktūrą, šis pasiūlymas tiktų turint ne mažiau kaip tris vienu laiko momentu aktyvius prietaisus.

LITERATŪRA

1. 5ci svetainė. [žiūrėta 2007-12-01]. Prieiga per internetą: <http://www.5ci.lt/wlan/WLAN_intro_1.htm>.
2. Design Line svetainė. [žiūrėta 2007-12-02]. Prieiga per internetą: <http://www.rfdesignline.com/GLOBAL/electronics/designline/shared/article/showArticle.jhtm?articleId=194300237 &pgno=2>.
3. Software technologies group svetainė. Services for Product Developers. [žiūrėta 2007-12-03]. Prieiga per internetą: <http://www.stg.com/wireless/ZigBee_netw.html>.
4. Steven M., ZigBee/IEEE 802.15.4. University of Wisconsin. [žiūrėta 2007-12-03]. Prieiga per internetą: <http://pages.cs.wisc.edu/~suman/courses/838/f06/zigbee-myers-talk.pdf>.
5. Qusay H. Mahmoud. Wireless Application Programming with J2ME and Bluetooth. Sun Microsystems svetainė. [žiūrėta 2007-12-02]. Prieiga per internetą: <http://developers.sun.com/techtopics/mobility/midp/articles/bluetooth1/ >.
6. CDW svetainė. [žiūrėta 2007-12-03]. Prieiga per internetą: <http://www.cdw.com/webcontent/editorial/technologies/061803_Bluetooth.asp>.
7. Wisegeek svetainė. [žiūrėta 2007-12-03]. Prieiga per internetą: <http://www.wisegeek.com/what-is-irda.htm?referrer=adwords_campaign=irda_ad=031171&_search_kw=ir%20wireless &393108443>.
8. Search Mobile Computing svetainė. [žiūrėta 2007-12-03]. Prieiga per internetą: <http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci213689,00.html>.
9. GSM World svetainė. [žiūrėta 2007-12-03]. Prieiga per internetą: <http://www.gsmworld.com/technology/gprs/index.shtml>.
10. GSM World svetainė. [žiūrėta 2007-12-03]. Prieiga per internetą: <http://www.gsmworld.com/technology/edge/index.shtml>.
11. Šaltis A. Radijo technologijos vartotojų prieigų tinkluose. Daktaro disertacija. VGTU: technologijos mokslai, elektros ir elektronikos inžinerija - 01T. Vilnius 2004.
12. Zyren J. IEEE 502.11g to benefit WLANs. The Official Site of the Embedded Developement Community svetainė. [žiūrėta 2007-12-01]. Prieiga per internetą: <http://www.embedded.com/story/OEG20020628S0061>.
13. Kau S.W., Hwang S.W. Department of Computer Science and Information Engineering.

Ntional Kaohsiung Univercity. All-Optical IP-over-DWDM MAN Ring Network with CSMA/CP Protocol– 2002. [žiūrėta 2007-12-03]. Prieiga per internetą: <http://140.134.132.124:8080/dspace/bitstream/2377/1363/1/ce07ics002002000058.PDF>.

14. Prismmodelchecker svetainė. [žiūrėta 2007-12-03]. Prieiga per internetą: <http://www.prismmodelchecker.org/casestudies/zigbee.php>.
15. Rajendran V., Obraczka K., Garcia-Luna-Aceves. Energy-efficient, collision-free medium Access control for wireless sensor networks - 2006. [Hingham, MA, USA, 2006]. MA, 2006. p. 63-78
16. Liaukuvienė A. RadioEthernet standartas ir jo taikymas. Hakeriai svetainė. [žiūrėta 2008-03-03]. Prieiga per internetą: <http://www.hakeriai.lt/articles.php?id=1921>.
17. Chih-Peng LI. Virtual-FIFO Back-Off Algorithm for Cpllision in Wireless Networks – 2005 National Sun Yat-Sen University. [Taiwan, 2005]. 2005. p.4056-4063.
18. Chen C., Seo E., Kim H., Lao H. SELECT: Self-Learning Collision Avoidance of Wireless Networks – 2007. Uivercity of Illinois. [Illinois, USA, 2007]. Illinois 2007.
19. Improving performance ofwireless networks using collisionresistent modulations – 1998: Global telecommunications conference. [Sidnėjus, Australija 1998 m. 08 mėn. 11 d.]. Sidnėjus, 1998. 2186-2191 p.
20. Chockler G., Gilert S. Computer Science and Artificial Inteligence Lab. Replicated State Machines for Collision-Prone Wireless Networks. Kembridžas – 2006. [žiūrėta 2008-03-03]. Prieiga per internetą: <http://courses.csail.mit.edu/6.885/spring06/papers/ChocklerGilbert.pdf>.
21. Application-based collision avoidance in wireless sensor networks – 2004: IEEE International Conference. Pranešimo medžiaga. [2004 m. lapkričio 16-18 d.]. 2004. 506-514 p.
22. Kwon Y., Fang Y., Latchman H., Fast collision resolution (FCR) MAC algorythm for wireless local area networks – 2002: Global telecommunications conference. Pranešimo medžiaga. [2002 lapkričio 21d.] p. 2250-2254

Darbas buvo pristatytas Šiaulių universiteto, Technologijos fakulteto „Studentų mokslinių darbų“ konferencijoje. 2008 m. gegužės 14 d. Šiauliai.

PRIEDAI

1 PRIEDAS

NEŠIOJAMO KOMPIUTERIO LENOVO PARAMETRAI

[CD-ROM]- lenovo.nfo

2 PRIEDAS

NEŠIOJAMO KOMPIUTERIO VECTOR PARAMETRAI

[CD-ROM]- vector.nfo

3 PRIEDAS

KREIPTIES TAŠKO LINKSYS WAP11 PARAMETRAI

Linksys Wap11.

Prieigos taškas Linksys Wap11:

Dirbantis pagal IEEE 802.11b standartą;

Taktinis dažnis 2.4GHz;

Siųstuvo radijo dažnio išėjimo galia 4 Mw/MHz.

Dydžiai

• L = 7.5 inches (190.5mm)

• W = 4.6 inches (116.84mm)

• H = 1.375 inches (35mm)

Saugumas

• WPA-Enterprise

• WPA-Personal

• 64/128-bit WEP

• MAC Filtraimas

4 PRIEDAS

NEŠIOJAMO KOMPIUTERIO DELL LATITUDE PARAMETRAI

Dell Latitude D630 NB

Gamintojas	Dell
Kodas	3510854LTD630V
Gamintojo kodas	D944C
Garantija	36
Gamintojo puslapis	[nuoroda]
Papildoma informacija	-
Pagaminimo šalis	Airija
Ekrano įstrižainė	36 cm (14.1")
LCD matricos tipas	WXGA+ (1440x900)
Centrinio procesoriaus tipas	Intel Core 2 Duo T8100
Centrinio procesoriaus dažnis	2.1 GHz
Operatyvioji atmintis	2x 1024 MB DDR2 667 MHz
Standžiojo disko talpa	120 GB
Standžiojo disko sūkių skaičius	5400 aps/min
Vaizdo plokštė	nVidia Quadro NVS 135M
Optinio įrenginio tipas	DVD+-R/RW
Internetinė kamera	nėra
Bevielio tinklo sąsaja	802.11a/b/g
Bluetooth sąsaja	yra
PCMCIA lizdas	yra
Express Card lizdas	nėra
Modemas	yra
Tinklo jungtis	10/100/1000 Mbps
IEEE 1394 (firewire) jungtis	yra
TV-out jungtis	nėra
COM (serial) jungtis	yra
USB jungtis	yra, 4x USB 2.0
Operacinė sistema	Windows Vista Business
Svoris (kg)	2.4

5 PRIEDAS

MOBILAUS TELEFONO SONY ERICSSON PARAMETRAI

Sony Ericsson T610

Pagrindinės savybės	
Dažnis	900/1800/1900
Svoris su baterija	95
Matmenys (mm)	102x44x19
Ekranas	128x160, 65536
Išorinis ekranas	-
Baterija	
Baterija	Li-Polymer
Baterijos talpa (mAh)	770
Budėjimo laikas iki (val.)	315
Pokalbių trukmė iki (val.)	11
Duomenų perdavimas	
Naršyklė	2.0 versija
HSCSD	2+1
GPRS	4+1
EDGE	-
3G (UMTS)	-
3G (HSDPA)	-
WLAN	-
Sujungimas su kompiuteriu	
USB	✓
IR jungtis	✓
Bluetooth	✓
Sinchronizacija su kompiuteriu	✓