UNIVERZITET CRNE GORE
ELEKTROTEHNIČKI FAKULTET

Jelena Kovač

Predlog rješenja za primjenu distribuiranog snimanja saobraćaja u velikim bežičnim senzorskim mrežama

- magistarski rad -

Podgorica, 2021.

UNIVERZITET CRNE GORE
ELEKTROTEHNIČKI FAKULTET

Jelena Kovač

Predlog rješenja za primjenu distribuiranog snimanja saobraćaja u velikim bežičnim senzorskim mrežama

- magistarski rad -

Podgorica, 2021.

PODACI I INFORMACIJE O MAGISTRANTU

Ime i prezime: **Jelena Kovač**
Datum i mjesto rođenja: 26.02.1996. Nikšić, Crna Gora
Prethodno završene studije:

Osnovne studije: Elektrotehnički fakultet Podgorica, Univerzitet Crne Gore,
smjer: Elektronika, telekomunikacije i računari, 180 ECTS kredita, 2017.godine
Specijalističke studije: Elektrotehnički fakultet Podgorica, Univerzitet Crne Gore,
smjer: Telekomunikacije, 60 ECTS kredita, 2018.godine

INFORMACIJE O MAGISTARSKOM RADU

Elektrotehnički fakultet
Studijski program: Elektronika, telekomunikacije i računari - Telekomunikacije
Naslov rada: **Predlog rješenja za primjenu distribuiranog snimanja saobraćaja u velikim bežičnim senzorskim mrežama**
Mentor: Prof. dr Enis Kočan

UDK, OCJENA I ODBRANA MAGISTARSKOG RADA

Datum prijave magistarskog rada: 29.01.2021. god.
Datum sjednice Vijeća na kojoj je prihvaćena tema: 26.05.2021. god.
Komisija za ocjenu teme i podobnosti magistranta:

1. Prof. dr Milica Pejanović-Đurišić
2. Prof. dr Zoran Veljović
3. Prof. dr Enis Kočan

Komisija za ocjenu rada:

1. Prof. dr Milica Pejanović-Đurišić
2. Prof. dr Zoran Veljović
3. Prof. dr Enis Kočan

Komisija za odbranu rada:

1. Prof. dr Milica Pejanović-Đurišić
2. Prof. dr Zoran Veljović
3. Prof. dr Enis Kočan

Datum odbrane: 12.01.2022. god.

Sadržaj

Sažetak

U fazama razvoja i testiranja bežičnih senzorskih mreža (WSN - *Wireless Sensor Networks*), veoma je važno imati uvid u saobraćaj koji se razmjenjuje u mreži. Za tu svrhu, sniferi imaju nezamjenjivu ulogu. U slučaju da WSN pokriva veliko područje, zbog slabljenja bežičnog signala i ograničene osjetljivosti prijemnika, jedan snifer ne može pružiti snimak cjelokupnog saobraćaja. Predmet istraživanja ovog rada je predlog distribuiranog rješenja za monitoring saobraćaja, koje zahtijeva implementaciju više snifera u WSN. Uvođenje više od jednog snifera dovodi do pojave višestruke detekcije istih paketa, od strane više bliskih snifera. Da bi snimak mrežnog saobraćaja sadržao samo originalno emitovane pakete, prvo se odgovarajućim algoritmom moraju detektovati kopije paketa, a onda i filtrirati, odnosno ukloniti.

Predloženo rješenje se zasniva na Argus softveru, u kojem je realizovan udaljeni pristup mrežnom saobraćaju u višekanalnim WSN u realnom vremenu, uz pomoć MQTT (*Message Queuing Telemetry Transport*) protokola. Program koji se pokreće na klijentskom računaru je proširen sa pomenutim algoritmom filtracije. Validacija rješenja je izvršena simulacionim i eksperimentalnim putem. Za simulacionu provjeru korišćen je snimak mrežnog saobraćaja iz OpenTestbed-a, dok eksperimentalna provjera, pored BeamLogic 802.15.4 Site Analyzer snifera, je uključivala rekonfigurizaciju OpenTestbed čvorova u ulogu snifera, kao i adaptaciju Argus softvera u radu sa rekonfigurisanim senzorskim uređajima.

Najvažniji rezultat rada je realizacija distribuiranog snifera u velikim WSN, što podrazumijeva rješenje koje će omogućiti upotrebu više od jednog snifera za efikasan monitoring mreže. Zatim, u radu je izvršeno povezivanje predloženog rješenja sa postojećim Argus softverom, čime je omogućen monitoring mrežnog saobraćaja na daljinu. Preko Interneta, snimak saobraća je dostupan sa bilo koje lokacije i svi klijenti sa pravom pristupa će dobiti podatke iz mreže u kojoj je postavljen distribuirani snifer.

Rješenje se može upotrebljavati prilikom ispitivanja performansi novih protokola, prije procesa standardizacije ili industrijske proizvodnje, kao i u stvarnim aplikacijama, u cilju praćenja funkcionalnosti i prikupljanju statistike.

Ključne riječi: 802.15.4 Site Analyzer, Argus, Distribuirani snifer, OpenWSN, Snifer, Bežične senzorske mreže, OpenTestbed

Abstract

In the development and testing phases of Wireless Sensor Networks (WSNs), it is of paramount importance to get insight into the traffic exchanged within the network. For this purpose, the sniffers are an irreplaceable tool. In the case where WSN covers a large area, a single sniffer cannot provide a trace of the complete data traffic, due to the attenuation of the wireless signal and the limited sensitivity of the receiver. The subject of research in this paper is the proposal of a distributed solution for traffic monitoring that includes the implementation of multiple sniffers in the WSN. Overhearing the same data traffic by different sniffers is likely in such a scenario, resulting in, multiple copies of the same data packet appearing in the merged trace. To overcome this issue, real time traffic filtering needs to be applied. As a result, traffic capture from the network contains only originally transmitted packets.

The proposed solution is based on the Argus software that allows real-time remote access to recorded network traffic in multi-channel WSN, using the MQTT (*Message Queuing Telemetry Transport*) protocol. The program running on the client computer is extended with a mentioned filtering algorithm. Simulation, as well as, experimental evaluation was performed. For the simulation verification, a snapshot of the actual network from OpenTestbed was used. Experimental testing of the extended Argus software, in addition to the use of a BeamLogic 802.15.4 Site Analyzer sniffer, required the configuration of OpenTestbed nodes in the role of a sniffer. In addition, it was necessary to adapt the software to work with these devices.

The most important result of the work is the realization of a distributed sniffer in large WSNs. A solution enables the use of more than one sniffer for efficient monitoring of network traffic. Except that, the proposed solution connects with the existing Argus software, which enables the sharing of the saved traffic traces through a cloud. The traffic capture is available from any location, any client with the right of access will receive data from the network in which the distributed sniffer is installed.

The solution can be used during the performance testing of novel protocols, before the standardization process or industrial production, as well as in real world-applications, in order to monitor functionality and collect statistics.

Keywords: 802.15.4 Site Analyzer, Argus, Distributed sniffer, OpenWSN, Sniffer, WSNs, OpenTestbed

Glava 1

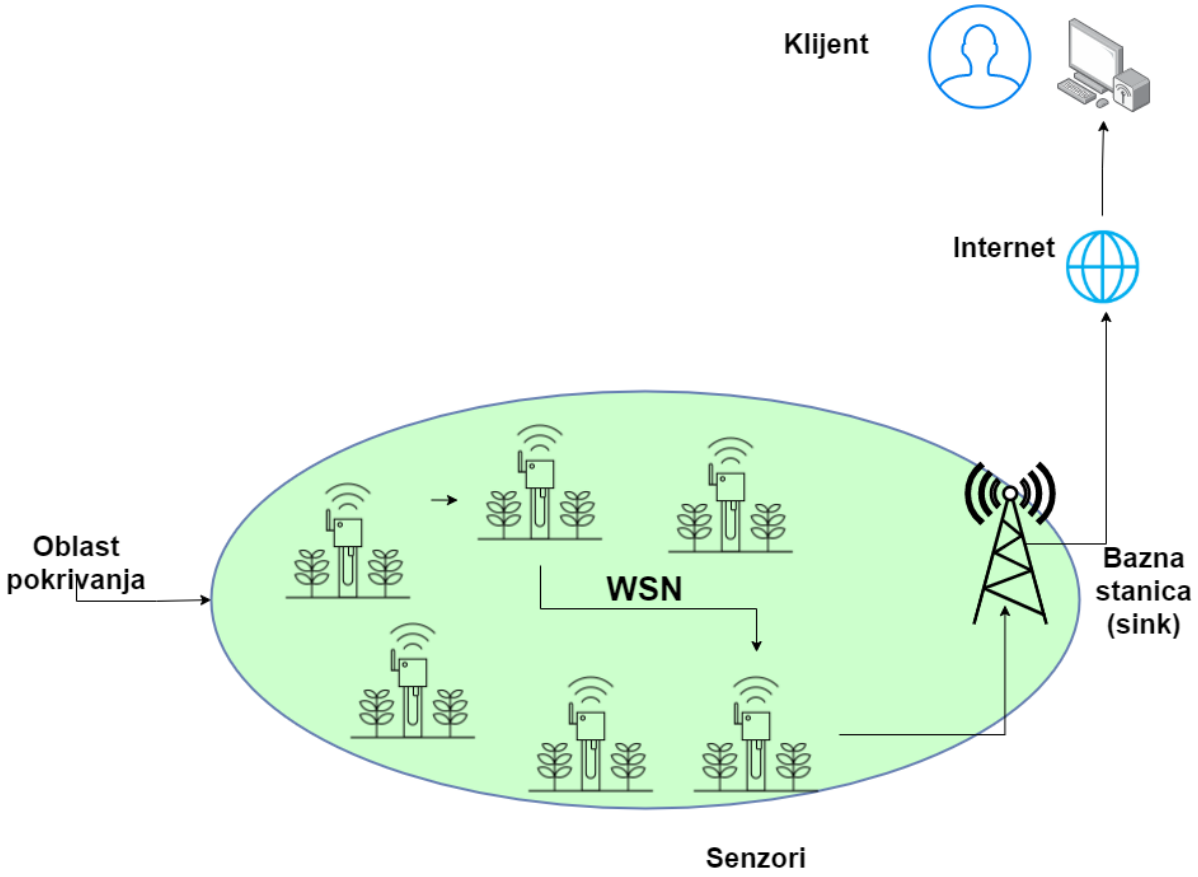
Uvod

Bežične senzorske mreže se generalno mogu opisati kao mreža komunikacionih čvorova povezanih bežično, koji zajedno prikupljaju informacije i omogućavaju interakciju ljudi, ili računara, sa okruženjem [1].

Bežičnu senzorsku mrežu čine senzorski čvorovi i bežična bazna stanica (*sink*). Senzori prate pojave u okruženju, kao što su: temperatura, zvuk, vibracija, pokret, zagađenje i nakon detekcije, preko mreže prosleđuju podatke bežičnoj baznoj stanici. Informacije se mogu prenositi preko više komunikacionih linkova. *Sink* predstavlja vezu između korisnika i WSN i obično je povezan na spoljnu mrežu (npr. Internet) preko *gateway*-a. Podaci prikupljeni od strane bežične bazne stanice se prosleđuju na obradu i analizu. Korisnik uz pomoć bazne stanice može prikupljati informacije iz WSN unosom jednostavnih upita. Primjer jedne WSN mrežne arhitekture je prikazan na slici 1.1. U mreži može biti jedna ili više baznih stanica, zavisno od potreba mreže. Sa jedne strane, bežične senzorske mreže pokreću razvoj novih aplikacija kao i novih tržišta, dok sa druge strane, na dizajn utiče nekoliko ograničenja koje zahtijevaju nove paradigme. Prikupljanje/obrada informacija i komunikacija pod ograničenom količinom energije, memorije, kao i procesorske moći, zahtijeva upotrebu višeslojnog mrežnog dizajna koji sadrži distribuiranu obradu signala/podataka, kontrolu pristupa medijumu, kao i nove komunikacione protokole [2].

Bežične senzorske mreže, zbog svoje fleksibilnosti, primjenu nalaze u različitim domenima, kao što su: industrija, vojska, medicina, kao i u pametnim automobilima, domovima i sl.. Ove mreže, koje podrazumijevaju M2M (*Machine to Machine*) tip komunikacija, predstavljaju osnovu koncepta Interneta stvari (*IoT - Internet of Things*). Glavni cilj ovog koncepta je digitalizacija i automatizacija različitih procesa, u cilju povećanja efikasnosti, kvaliteta, fleksibilnosti, uz istovremeno smanjenje troškova. Koncept se oslanja na upotrebi jeftinih uređaja, sa ograničenim hardverskim sposobnostima, koji imaju mogućnost prikupljanja i slanja prikupljenih podataka. Za najveći broj aplikacija, glavni zahtjevi osim energetske efikasnosti, jesu sigurnost i pouzdanost. Tokom dizajna ovih mreža, potrebno je

riješiti razne probleme koji se mogu javiti zbog ograničene memorije, kao i procesorskih moći senzorskih uređaja.



Slika 1.1 Primjer WSN mrežne arhitekture

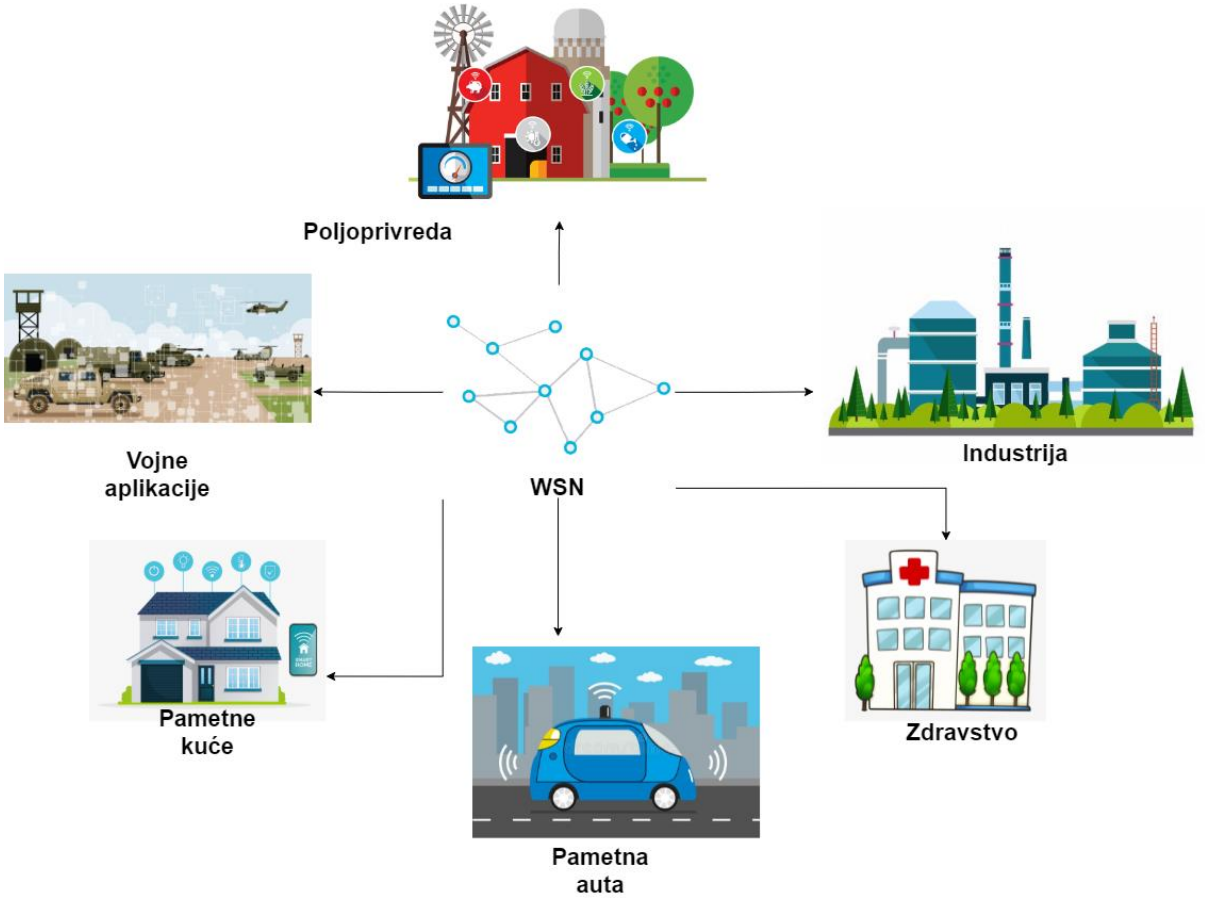
Širok spektar primjene bežičnih senzorskih mreža izaziva sve veće interesovanje istraživačke zajednice, kao i industrije. Neke od najznačajnih oblasti primjene WSN-a se mogu vidjeti na slici 1.2. Vojni domen nije samo prvo polje korišćenja WSN, nego se smatra da je motivisao pokretanje istraživanja o senzorskim mrežama [3]. *Smart Dust* [4] je tipičan primjer početnih istraživačkih napora, koji su izvedeni krajem devedesetih godina prošlog vijeka, da bi se razvili senzorski čvorovi koji bi, uprkos svojoj maloj veličini, bili sposobni za obavljanje špijunskih aktivnosti. Tehnološki napredak koji je postignut od tada učinio je WSN sposobnim da podrže različite operacije kao što su: nadzor bojnog polja, nadgledanje borbe, kao i otkrivanje uljeza. U domenu zdravstva, WSN koristi napredne medicinske senzore za praćenje pacijenata unutar zdravstvenih ustanova, bolnica ili u okviru doma pacijenta, kao i za praćenje vitalnih parametara pacijenata u realnom vremenu uz pomoć nosivog hardvera [5]. Primjene u životnoj sredini, koje zahtijevaju stalno praćenje ambijentalnih uslova u nepoznatim sredinama, ili na udaljenim lokacijama, mogu se poboljšati korišćenjem WSN-ova. Glavne

grane primjene su: praćenje parametara u vodi i vazduhu, kao i detekcija opasnosti (požari, zemljotresi, vulkani i sl.). Domen flore i faune je vitalan za svaku zemlju. Nadzor plastenika, usjeva i primjena u stočarstvu predstavljaju glavne primjene u ovoj oblasti [6]. Automatizacija navodnjavanja omogućava efikasnije korišćenje vode i smanjuje gubitke.

WSN u industriji svoju ulogu može pronaći u domenu robotike, logistike i održavanja mašina. Čeličane, rafinerije ulja i slične industrijske platforme zahtijevaju kontrolu i skalabilni transport. Veliki broj senzora je potreban u cilju dostavljanja informacija o temperaturi, pritisku i nivou popunjenosti rezervoara kontrolnom centru. Centar koristi prikupljene informacije da upravlja aktuatorom, započinje novi proces proizvodnje, raspoređuje održavanje ili uključuje alarm [7]. WSN su zaista alat za mjerenje prostornih i vremenskih karakteristika bilo kojih pojava u urbanom okruženju, pružajući neograničen broj aplikacija. Najpoznatije aplikacije u urbanom domenu se odnose na pametne kuće, pametne gradove i sisteme transporta [8]. Senzorske mreže su pokrenule sledeću revoluciju u informacionim i komunikacionim tehnologijama [9].

Pošto bežične senzorske mreže postaju sve veće, kompleksije i njihova uloga značajnija u različitim privrednim i industrijskim granama, projektovanje i monitoring postaju nezamislivi bez detaljnog uvida u mrežni saobraćaj, za šta su neophodni uređaji koji se nazivaju sniferi (njuškala). Sniferi su uređaji sa specijalizovanim softverom, koji mogu snimati i analizirati mrežni saobraćaj. Pošto preuzimaju kopije paketa sa nivoa linka, omogućavaju analizu mrežnih performansi, detektovanje interferencije u mreži i analizu paketa, bez uticaja na performanse mreže.

Senzorske mreže se obično primjenjuju u složenim okruženjima, sa različitim faktorima ometanja okoline i otkaz jednog čvora može dovesti do dinamičke promjene topologije mreže. Takođe, senzorski čvorovi imaju ograničene resurse, kao što su: napajanje, memorija, mogućnost komunikacije i obrade podataka na čvoru. Stoga, sniferi svoju ulogu mogu pronaći kako u fazama razvoja i testiranja novih mreža, tako i u stvarnim aplikacijama. Mogu pomoći u analizi radio-frekvencijskog (RF - *Radio frequency*) okruženja. Za efektivnu analizu, potrebno je uzeti u obzir cjelokupni razmijenjeni saobraćaj.

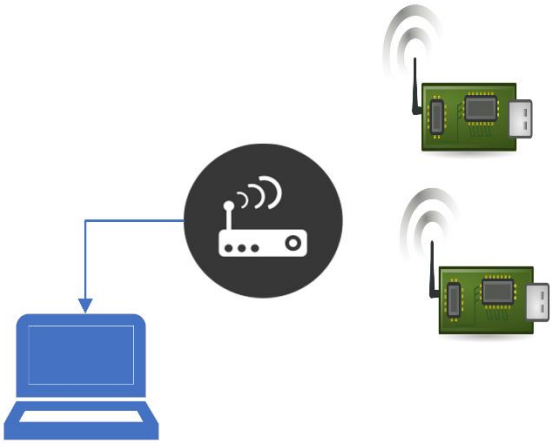


Slika 1.2. Primjeri primjene WSN u različitim oblastima

Računarske simulacije WSN koriste teorijske modele i teško je precizno modelovati stvarno fizičko okruženje i karakteristike bežičnih komunikacionih veza, poput kolizije na radio kanalima i multipleksiranje signala, što dovodi do određenog jaza između efekta validacije i stvarnog stanja. Rezultati dobijeni pomoću testnih eksperimentalnih postavki (*testbed* okruženje) mogu sadržati obuhvatnije faktore koji utiču na mrežu i izbjeći greške koje su rezultat pojednostavljenog modela u simulaciji. Međutim, *testbed*-ovi se obično postavljaju na manjem prostoru i čini ih veliki broj senzorskih uređaja, što se razlikuje od realnog okruženja. Zato je teško precizno predvidjeti mrežne performanse WSN-ova, nakon primjene u stvarnim aplikacijama. Radi održavanja normalnog funkcionisanja, WSN su potrebni alati za praćenje rada u realnom vremenu u stvarnom okruženju.

Upravo su sniferi uređaji koji omogućavaju snimanje mrežnog saobraćaja u nekoj WSN, a time i praćenje funkcionisanja kompletne mreže, njenih performansi, predikciju mogućih problema i otklanjanje postojećih, kao i planiranje eventualnog proširenja broja senzorskih čvorova, oblasti pokrivanja, itd. Kao takvi, postali su nezamjenjiv alat u

monitoringu rada postojećih WSN, planiranju i projektovanju novih WSN, analizi rada i performansi novopredloženih protokola, usvojenih standarda, i sl. Tipičan scenario upotrebe snifera za snimanje saobraćaja u jednoj WSN podrazumijeva njegovo povezivanje sa računarom, odgovarajućom kablovskom konekcijom, slika 1.3.



Slika 1.3 Primjena snifera za snimanje saobraćaja u WSN

Na računaru se vrši analiza snimljenog mrežnog saobraćaja. Međutim, ovakav sistem ima jedan nedostatak. Naime, predloženi sistem se može koristiti samo od strane korisnika na lokaciji nadgledane mreže, odnosno samo na računaru povezanom sa sniferskim uređajem. Stoga, istraživači iz OpenWSN radne grupe su razvili softver Argus, koji omogućava udaljeni pristup u realnom vremenu, preko Interneta, snimljenom mrežnom saobraćaju [10].

Motiv ovog rada je kreiranje alata za analizu i monitoring bežičnih senzorskih mreža koje imaju veliki broj senzora, raspoređenih na velikom prostoru. U slučaju kada bežična mreža pokriva veliki prostor, jedan snifer ne može ostvariti uvid u cjelokupni saobraćaj, zbog ograničene osjetljivosti prijemnika, kao i slabljenja signala. Rješenje problema zahtijeva implementaciju više snifera u WSN mreži, koji trebaju da budu povezani u jednu cjelinu, na odgovarajući način. Takođe, potrebno je omogućiti monitoring mreže na daljinu, tako da je snimak mrežnog saobraćaja dostupan sa bilo koje lokacije, putem Interneta, za sve klijente koji imaju pravo pristupa.

Cilj ovog istraživanja je proširiti postojeće Argus softversko rješenje, tako da se omogući upotreba većeg broja snifera za snimanje saobraćaja u jednoj WSN. U takvim scenarijima neminovno će doći do višestruke detekcije istih paketa, od strane susjednih snifera.

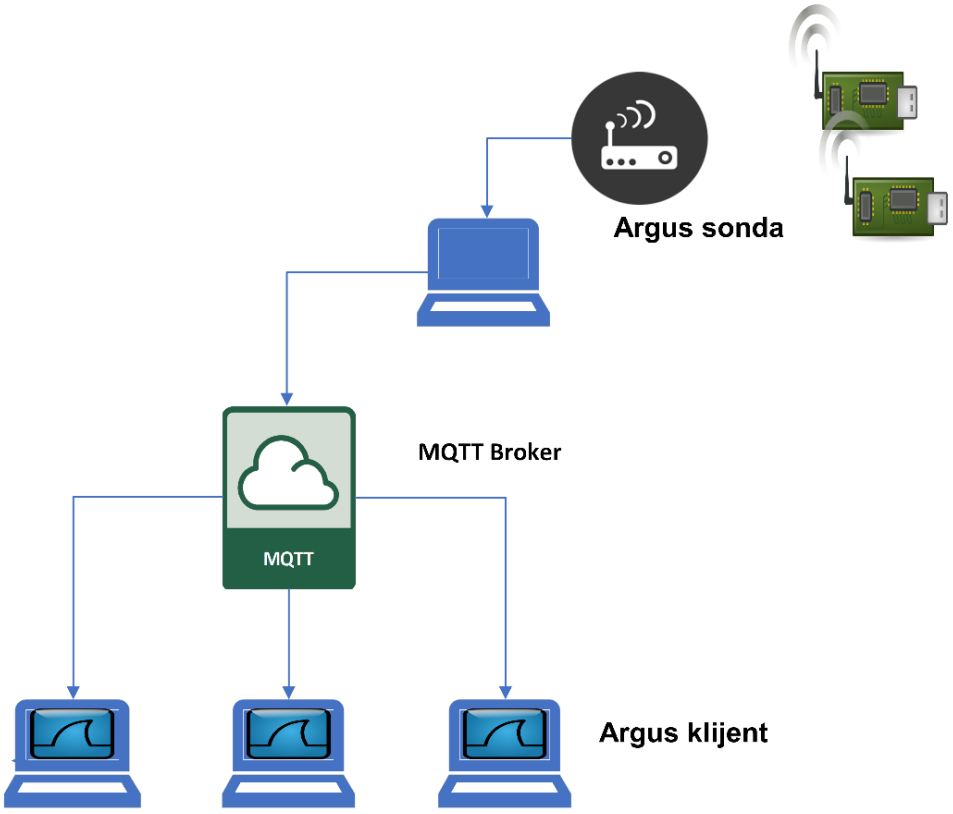
Predloženo rješenje će omogućiti filtriranje višestruko detektovanih istih paketa i na taj način omogućiti efikasnu analizu bežičnih senzorskih mreža sa velikim brojem senzora, koje se prostiru na velikim površinama.

Rad je organizovan na sledeći način. Argus softver i njegove komponente su predstavljene u drugoj Glavi. Osim detaljnog opisa načina funkcionisanja, adresirana su i njegova ograničenja. Takođe, objašnjena je 6TiSCH (*The IETF IPv6 over the TSCH mode of IEEE802.15.4e*) grupa protokola i njihova primjena, kao bitan segment predloženog rješenja. U trećoj Glavi je opisano predloženo rješenje distribuiranog snifera i njegova arhitektura. Posebna pažnja je usmjerena na algoritam filtriranja višestruko detektovanih paketa, koji na osnovu kontrolnih podataka (vrijeme, kanal transmisije i oznaka snifera) i sadržaja paketa, procjenjuje da li je uhvaćeni paket jedinstven, ili je došlo do višestrukog osluškivanja od strane više snifera. Osim toga, postojeće softversko rješenje Argus je prošireno, tako da je za snimanje mrežnog saobraćaja, osim specijalizovanog snifera, moguće koristiti i jednostavne senzorske uređaje koji su priključeni lokalno ili u testbedu. U četvrtoj Glavi, izvršena je simulaciona i eksperimentalna provjera predloženog algoritma. U simulacionoj provjeri se modeluju višestruki sniferi u WSN, a zatim se filtrira snimak saobraćaja iz stvarne WSN, dok se u eksperimentalnoj provjeri upoređuje saobraćaj uhvaćen od strane dva snifera na testbedu. U poslednjoj Glavi su sumirani rezultati i istaknut njihov značaj.

Glava 2

Argus softver

Kao što je već objašnjeno, sniferi su uređaji koji uz pomoć specijalizovanog softvera snimaju i analiziraju razmijenjeni mrežni saobraćaj. Omogućavaju analizu performansi mreže, otkrivaju smetnje i analiziraju razmijenjene pakete, bez uticaja na rad mreže. Sniferi predstavljaju nezamjenljiv alat tokom razvoja i testiranja novih standarda i protokola, dizajniranja novih mreža i implementacije sistema u realnom okruženju. Na slici 2.1 predstavljeno je pomenuto Argus rješenje, koje omogućava udaljeni pristup snimljenom saobraćaju u nekoj bežičnoj senzorskoj mreži. Podacima je moguće pristupiti u realnom vremenu preko Interneta [10]. Međutim, ovo rješenje je ograničeno na korišćenje samo jednog snifera, koji često ne može obezbjediti snimak cjelokupnog razmijenjenog saobraćaja u mreži.

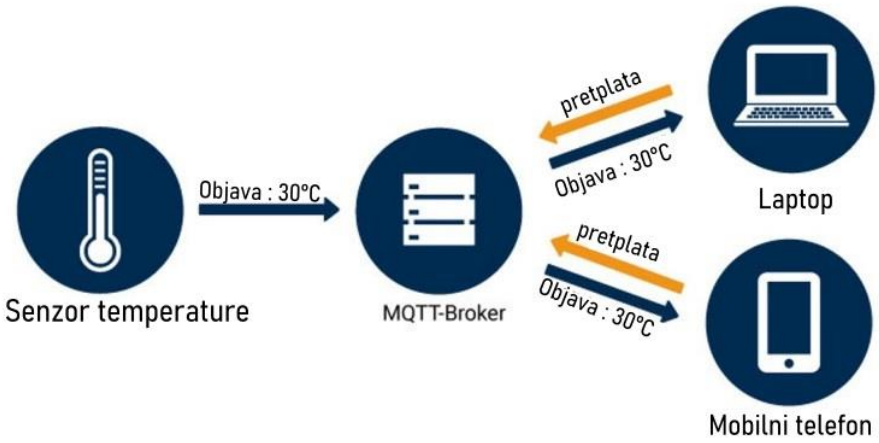


Slika 2.1 Argus rješenje za udaljeni pristup snimljenom mrežnom saobraćaju

Argus softver (slika 2.1) je razvijen u sklopu OpenWSN ekosistema. OpenWSN [11] je softverski projekat koji je nastao na Univerzitetu u Kaliforniji - Berkliju i trenutno se održava od strane INRIA (*National Institute for Research in Computer Science and Automation*) instituta i UOC (*Open University of Catalonia*) univerziteta. Glavni cilj projekta je da obezbjedi implementaciju grupe protokola otvorenog koda, koja se bazira na IoT (*Internet of Things*) standardima i koja je kompatibilna sa različitim hardverskim platformama. OpenWSN je projekat otvorenog koda koji je postao referentna implementacija grupe protokola zasnovanih na standardima za industrijski IoT (IIoT- *Industrial Internet of Things*). Implementiran je u programskom jeziku C i sastoji se od dva glavna dijela: firmver (*firmware*), koji radi na uređajima i softver (*software*), koji se pokreće na računaru. Firmver se može koristiti na 11 različitih hardverskih platformi, uključujući popularne TelosB, OpenMote [12] i IoT-LAB mote [13]. Takođe, podržan je od velikog broja mikrokontrolera male snage. Softver – *OpenVisualizer* sadrži skup podržanih skripti, uključujući vizualizaciju čvora u mreži i konfiguraciju jednog čvora kao *root*-a u mreži.

Argus omogućava dijeljenje 802.15.4 Site Analyzer-a preko Interneta [10]. Ova funkcionalnost omogućava pristup istom sniferu od strane velikog broja korisnika. Argus se bazira na MQTT (*Message Queuing Telemetry Transport*) komunikacionom protokolu. MQTT je protokol za razmjenu poruka, koji je razvijen od strane Organizacije za unapređenje strukturiranih informacionih standarda (OASIS - *Organization for the Advancement of Structured Information Standards*) [14]. Misija OASIS-a je: unaprijediti transparentan razvoj softvera i standarda otvorenog koda kroz moć globalne saradnje i zajednice [15].

MQTT je protokol aplikativnog nivoa, koji je idealan za povezivanje udaljenih uređaja sa minimalnom propusnošću mreže. Zasnovan je na principu pretplate i objave (*publish-subscribe*), posredstvom komponente zvane broker (posrednik). Klijenti se pretplaćuju i dobijaju određeni informacioni sadržaj na takozvane teme (*topics*) za koje su zainteresovani. Po MQTT protokolu, broker je centralna komponenta mreže, i njegov zadatak je da, poruke koje pristižu, pravilno preusmjeri onim klijentima koju su na tu temu pretplaćeni. U najvećem broju slučajeva klijent uspostavlja perzistentnu TCP (*Transmission Control Protocol*) konekciju sa brokerom, ali *je moguće i da se MQTT protokol oslanja na UDP (User Datagram Protocol)* transportni protokol, kada se zahtijeva da se koristi što manje resursa mreže. Uobičajena primjena MQTT protokola je predstavljena na slici 2.2.



Slika 2.2 Primjer primjene MQTT komunikacionog protokola

Argus koristi MQTT protokol za slanje snimljenog saobraćaja do udaljenih klijenata koji su pretplaćeni preko brokera na temu *'inria-paris/beamlogic'*. Snimljeni saobraćaj se analizira nekim od softvera za analizu protokola (protokol analizatori). Klijenti mogu sačuvati saobraćaj u lokalnoj memoriji za *offline* obradu.

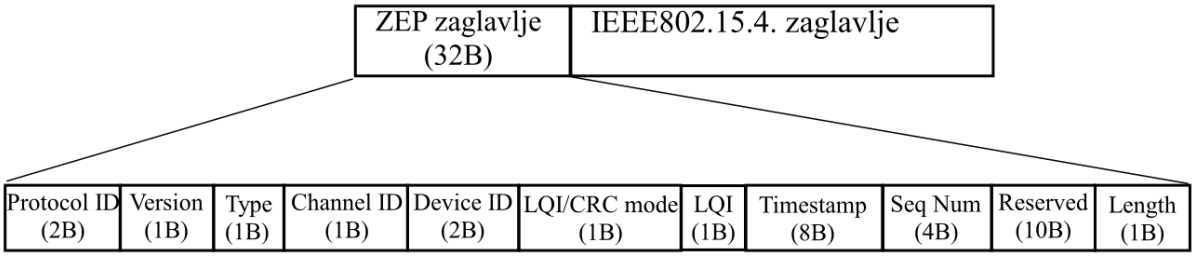
Wireshark je besplatni (*open-source*) protokol analizator, koji je našao široku upotrebu zbog efikasnosti i preglednog grafičkog interfejsa. Koristi se za analizu rada komunikacionih mreža, detekciju problema i razvoj komunikacionih protokola, kao i u edukativne svrhe. Njegova politika otvorenog koda omogućava talentovanim ekspertima iz oblasti mreža i komunikacija da ga nesmetano nadograđuju. Može se pokretati na Unix, Linux i Windows operativnom sistemu. Mrežni administratori ga mogu koristiti da lokalizuju probleme u mreži, inženjeri za bezbjednost da izučavaju probleme sigurnosti, razvojni programeri u razvoju protokola, dok studenti u učenju TCP/IP protokola, snimanju i analizi strukture paketa, itd. [16]. Wireshark softver se sastoji iz dva dijela. Prvi dio – detektor osluškuje pakete na nivou linka koji se razmjenjuju u mreži i čuva kopije u lokalnoj memoriji. Paketi sa viših protokolskih nivoa su enkapsulirani u frejmove nivoa linka i kao takvi se razmjenjuju. Druga komponenta sistema je analizator, koji ima mogućnost da analizira i razumije polja karakteristična za primijenjene komunikacione protokole, i prikazuje sadržaj polja detektovanih paketa. Kako bi Wireshark prikazao pakete, analizator paketa mora poznavati strukturu poruka koje koriste različiti protokoli u mreži.

Argus se sastoji od dvije komponente:

- **Argus Sonda (ArgusProbe.py)** je Python skripta koja se pokreće na računaru, koji je fizički povezan sa Site Analyzer-om. Njegov zadatak je da enkapsulira detektovane pakete preko MQTT protokola i šalje ih na MQTT broker.
- **Argus Klijent (ArgusClient.py)** je Python skripta koja se pokreće na klijentskom računaru. Klijent se pretplaćuje na specifične teme u cilju prikupljanja podataka. Tokom pokretanja ove skripte, Wireshark je otvoren u cilju analize paketa koji se klijentu predstavljaju kao da je na samoj lokaciji mreže. Višestruki klijenti mogu prikupljati podatke sa MQTT brokera u isto vrijeme. Argus Sonda i Argus Klijent su povezani preko MQTT brokera koji rade na *cloud*-u kao u uobičajenom MQTT objavi-pretplati podešavanju.

Argus softver je razvijen za pristup na daljinu saobraćaju snimljenom u bežičnoj senzorskoj mreži zasnovanoj na IEEE 80.15.4 standardu. IEEE 802.15.4 je osnovni standard za mnoge bežične senzorske mreže male snage. Definiše fizički i MAC (*Medium Access Control*) nivo. Aktuelne verzije standarda podrazumijevaju primjenu nelicenciranih frekvencijskih opsega na 868 MHz, 915 MHz i 2450 MHz. Najčešće se koristi frekvencijski opseg 2.4−2.485 GHz. Standard definiše format MAC zaglavlja i komunikaciju između čvorova na jednom kanalu. IEEE 802.15.4e radna grupa je redizajnirala postojeći MAC sloj sa TSCH (*Time Slotted Channel Hopping*) tehnikom višestrukog pristupa, kako bi se obezbijedila visoka pouzdanost i energetska efikasnost. Kroz vremensku sinhronizaciju i frekvencijsko skakanje (*frequency hopping),* riješeni su problemi RF smetnji i smanjena je potrošnja energije [17]. Primjer snifera koji se može koristiti u WSN mreži zasnovanoj na IEEE 802.15.4 grupi standarda, je BeamLogic 802.15.4 Site Analyzer [18].

Nakon detekcije IEEE 802.15.4 paketa u mreži, 802.15.4 Site Analyzer kreira svoje zaglavlje o primljenom paketu, koje sadrži podatak o nivou snage primljenog signala (RSSI-*Received Signal Strength Indicator),* oznaku kanala na kojem je primljen paket i vremensku referencu (*timestamp*). Na osnovu zaglavlja i dodatnih informacija o paketu, Argus formira ZEP (*ZigBee Encapsulation Protocol*) zaglavlje sa 12 polja. Skripta Argus Sonda enkapsulira IEEE 802.15.4 pakete koji su snimljeni u mreži, kao *payload* ZEP protokola. Kreirani paket sadrži dva polja i takav se prenosi preko MQTT brokera do klijenata, (slika 2.3). Na osnovu primljenog paketa, Argus Klijent dodaje nova zaglavlja (Ethernet, IPv6 - *Internet Protocol v6*, UDP) i prosleđuje Wireshark-u na prezentaciju klijentu.



Slika 2.3 Struktura paketa koja je formirana od strane Argus Sonda skripte

2.1. IEEE 802.15.4 standard

Najpoznatiji i najrasprostranjeniji standard u *low-power* radio komunikacionoj tehnologiji za za bežične personalne mreže (WPAN – *Wireless Personal Area Nework*) je IEEE 802.15.4. IEEE 802.15.4 balansira između energetske efikasnosti, komunikacionog dometa i protoka podataka. Razvijen je sa ciljem da definiše niže nivoe u OSI (*Open Systems Interconnection)* referentnom modelu za jeftine WSN male snage. Definiše fizički (npr. modulaciona šema koja se koristi) kao i MAC nivo (npr. određuje koji uređaj se uključuje i kad, na kom kanalu), ostavljajući višim nivoima mogućnost razvoja za specifične standarde kao što su ZigBee, 6LoWPAN (*IPv6 over Low -Power Wireless Personal Area Networks*), Thread i sl. [19].

Razvoj IEEE 802.15.4 standarda

IEEE 802.15.4 standard je imao više verzija nakon inicijalne IEEE802.15.4 – 2003 [19]. Opis pregleda razvoja standarda i njegovih karakteristika u različitim verzijama je dat u tabeli 2.1.

Tabela 2.1. Pregled razvoja IEEE802.15.4 standarda

IEEE 802.15.4 verzija	Detalji
IEEE 802.15.4 - 2003	Inicijalna verzija IEEE 802.15.4 standarda. Obezbjeđuje rad na 2 frekvencijska opsega, na nižim 868/915 MHz kao i na 2.4 GHz.
IEEE 802.15.4 - 2006	Ova verzija je omogućila povećanje protoka podataka koje se može postići na nižim frekvencijskim opsezima. Definiše 4 nove modulacione šeme koje se mogu koristiti – tri za niže frekvencijske opsege i jednu za 2.4 GHz.

IEEE 802.15.4a-2007	Ova verzija definiše dva nova fizička nivoa. Jedan je namijenjen za UWB (*Ultra wideband*) tehnologiju, dok drugi omogućava rad na proširenom spektru na 2.4 GHz.
IEEE 802.15.4c	Nadogradnja za 2.4 GHz, 868 MHz i 915 MHz, UBW i 779-787 MHz opseg (Kina).
IEEE 802.15.4d	2.4 GHz, 868 MHz, 915 MHz i 950 - 956 MHz (Japan) opseg.
IEEE 802.15.4e	Ova verzija je usvojena kao dopuna postojećem standardu 802.15.4-2006 na MAC nivou koja koristi *channel hopping* radi podrške industrijskom tržištu, povećanju otpornosti na spoljne smetnje i *multipath fadding*.
IEEE 802.15.4f	Novi fizički nivoi za UWB, 2.4 GHz i 433 MHz.
IEEE 802.15.4g	Ova verzija je predložena sa ciljem upotrebe u kontrolnim aplikacijama velikih razmjera kao što je *smart grid*.

Frekvencijski opsezi

Trenutni standard omogućava rad u tri opsega (868/915/2450 MHz). Najčešće je u upotrebi globalni nelicencirani opseg od 2.4 - 2.485 GHz. U ovom opsegu koristi *Offset quadrature phase-shift keying* (O-QPSK) mapiranje sa protokom podataka od 2Mb/s. Iz perspektive korisnika, protok podataka je 250kb/s. Tehnika koja je korišćena, zbog robusnosti, je prenos proširenim spektrom sa direktnom sekvencom (DSSS - *Direct-sequence spread spectrum*). Na opsegu od 2.4 GHz, IEEE 802.15.4 standard definiše primjenu 16 frekvencijskih kanala, označenih kao Kanali 11 do 26, koji su međusobno udaljeni po 5 MHz, na frekvencijama između 2.405 GHz i 2.480GHz, slika 2.4. Osim ovih kanala, u Evropi je definisana i mogućnost primjene Kanala 0, na opsegu između 868 i 868.6 MHz, a Kanali 1 do 10, na opsegu od 902 do 928 MHz se mogu koristiti u Sjedinjenim Američkim Državama (SAD) i širine su po 2MHz. Radio može slati/ primati podatke na bilo kojem kanalu i može izvršiti prelazak sa jednog na drugi kanal nakon vremenskog intervala od 192 μs. [20]

Mrežna topologija

Postoje dva glavna tipa mrežnih topologija koje se mogu koristiti u okviru IEEE 802.15.4 standarda. Ove topologije mogu biti primijenjene u različitim aplikacija i obje imaju svoje prednosti [19]. Glavne topologije su:

- Topologija zvijezda – mreža ima jedan centralni uređaj koji se naziva koordinator koji komunicira sa svim uređajima u mreži. Primjena ovog tipa topologije ograničava domet mreže do jednog hopa.
- *Peer to peer* topologija - mreža posjeduje koordinatora ali komunikacija je omogućena između uređaja, ne samo direktno preko koordinatora. Čvorovi mogu da rutiraju podatke iz mreže i samim tim mrežna pokrivenost je veća.

Mrežni čvorovi

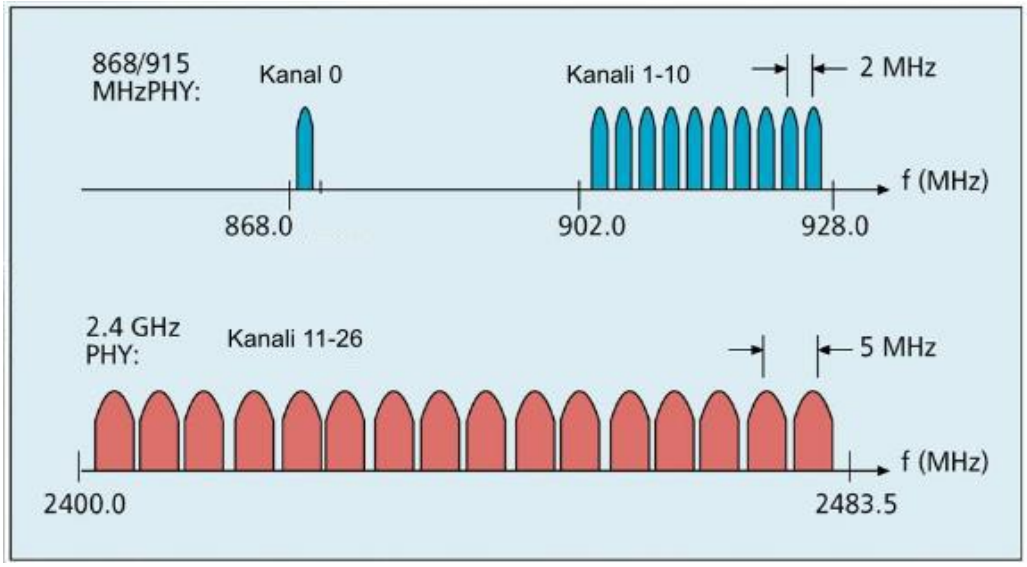
U mreži mogu postojati tri tipa uređaja, [19]:

- FFD (*Full Function device*) – uređaj koji može obavljati sve uloge. Može se koristi za slanje i prijem podataka ali i za potrebe rutiranja podataka kroz mrežu.
- RFD (*Reduced Function Device*) – uređaj koji ima smanjeni nivo funkcionalnosti. U mreži ovu ulogu obično ima krajnji uređaj koji može biti senzor ili *switch*. Pošto uređaj ne vrši rutiranje podataka, može se koristiti u neaktivnom modu kada nije u statusu slanja ili prijema podataka.
- Koordinator – uređaj koji upravlja IEEE 802.15.4 mrežom. U odnosu na FFD uređaj, ima ulogu u formiranju mreže, odnosno ulogu mrežnog menadžera kada je mreža formirana.

Rješenja zasnovana na IEEE 802.15.4 standardu

IEEE 802.15.4 standard se koristi kao osnova za veliki broj energetski efikasnih komunikacionih rješenja malog dometa, kod kojih se na različite načine definišu protokoli iznad MAC nivoa. Najpoznatije nadogradnje IEEE 802.15.4 standarda su ZigBee, Wireless HART, Thread, 6LoWPAN, 6TiSCH i drugi. ZigBee pruža više protokolske nivoe potrebne za radio sisteme male snage i malog protoka podataka, kao što su automatizacija domova, industrijski kontrolni sistemi, detekcija dima, prikupljanje podataka u medicini i sl. Iako su dostupne novije verzije IEEE 802.15.4 standarda, ZigBee koristi inicijalno izdanje iz 2003. godine [19]. Wireless HART je otvoreni standard koji je razvijen od strane HART *Communication Foundation* za korišćenje u 2.4 GHz ISM opsegu. Sistem koristi IEEE 802.15.4-2006 verziju standarda za fizički i nivo linka, dok je mreža organizovana u vremenski sinhronizovanu, samo-organizovanu i samo-održivu *mesh* arhitekturu [19]. *Thread* je mrežni protokol zasnovan na IPv6. Razvijen je namjenski za podršku *Internet of Things* (IoT) uređajima, odnosno najčešće se koristi u klijentskim aplikacijama u pametnom domu. Dizajniran je da bude lak i jednostavan, ali i da obezbjedi sigurnu vezu između hiljade uređaja

kao i vezu sa *cloud*-om. Kao osnovu na fizičkom i nivou linka ima IEEE 802.15.4 – 2006 standard [21]. 6LoWPAN omogućava prenos podataka u obliku IPv6 paketa preko IEEE 802.15.4 zasnovanih mreža. Obezbjeđuje *end-to-end* vezu i na taj način je moguće povezivanje sa velikim brojem mreža, uključujući direktno povezivanje sa Internetom. Može se koristiti u domenima automatizacije, kao i u industrijskom monitoringu. Na nivou linka koristi IEEE 802.15.4e verziju standarda, dok se na fizičkom nivou zasniva na IEEE 802.15.4g standardu [22]. 6TiSCH je komunikaciona arhitektura koja omogućava upotrebu IPv6 sa TSCH tehnikom višestrukog pristupa IEEE 802.15.4 standarda, namijenjena za upotrebu industrijskim kontrolnim aplikacijama. Na fizičkom nivou koristi IEEE 802.15.4 -2006 verziju standarda, a na nivou linka IEEE 802.15.4e [20].



Slika 2.4 Frekvencijski opsezi za IEEE 802.15.4 standard

MAC nivo

IEEE 802.15.4 standard definiše i protokole na nivou linka. Nivo linka direktno komunicira sa fizičkim nivoom, definiše format zaglavlja (sa poljima kao što su izvorišna i destinaciona adresa) i način komunikacije između *low-power* uređaja. Koristi zvijezda topologiju u kojoj svi uređaji direktno komuniciraju sa specijalnim uređajem za koordinaciju. Ovaj protokol nije bio odgovarajući zbog zahtjeva da neki uređaji (npr. uređaj za koordinaciju) imaju *duty cycle* = 100% kao i rad na samo jednom kanalu.

Duty Cycle predstavlja trajanje aktivnosti uređaja u posmatranom vremenskom intervalu, izraženo u procentima. Na primjer, ako uređaj na jednom kanalu emituje u dužini od dva vremenska slota na svakih 10 vremenskih slotova, njegov *duty cycle* je 20%. Ako uređaj

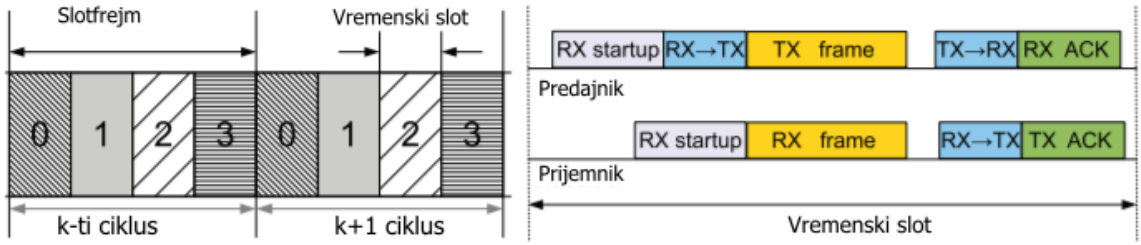
emituje na tri kanala u dužini od 2 vremenska slota, na svakih 10 vremenskih slotova, uređaj će biti aktivan u 6 od 10 vremenskih slotova, pa je njegov *duty cycle* 60% [23].

IEEE 802.15.4e

Zbog različitih propagacionih efekata koji se mogu javiti prilikom prostiranja signala bežičnim medijumom, nije moguće ostvariti pouzdanost koja se zahtijeva u kritičnim aplikacijama. Iz tog razloga je IEEE 802.15.4e radna grupa redizajnirala postojeći protokol uz pomoć vremenski sinhronizovane promjene kanala (TSCH - *Time Synchronized Channel Hopping*) kao tehnike višestrukog pristupa, u cilju prilagođenja potrebama industrijskih aplikacija. Cilj novog dizajna je prevazilaženje dva najveća problema prethodnog protokola: neoptimalna upotreba energije *low-power* uređaja, koja nije zavisila od saobraćaja u mreži, kao i problema interferencije zbog rada na jednom frekvencijskom kanalu. Novi protokol se zasniva na vremenskoj sinhronizaciji (zbog energetske efikasnosti) i *channel hopping*-u, koji mu omogućava veliku pouzdanost, dok je moguće održati nizak *duty cycle* koji se preporučuje za IoT kritičnih aplikacija, [20].

2.1.1. *Slotframe*

Svi uređaji u TSCH mreži sa više hopova su sinhronizovani. Vrijeme je podijeljeno na vremenske slotove (*timeslot*). Kao što je prikazano na slici 2.5 vremenski slot je dovoljne veličine za frejm maksimalne veličine (od uređaja A do uređaja B) i poruku potvrde uspješnog prijema (od uređaja B do uređaja A). Trajanje vremenskog slota nije definisano od strane IEEE802.15.4e standarda i može se prilagoditi potrebama aplikacije [20].



Slika 2.5 Slotframe i sadržaj jednog vremenskog slota

TSCH definiše brojač vremenskih slotova koji se naziva *Absolute Slot Number* (ASN). Kada se mreža kreira, ASN je postavljen na nulu i uvećava se za jedan kada dođe do isteka slota. Tokom pristupanja mreži, uređaj uči informaciju o trenutnom ASN. Pošto su uređaji

sinhronizovani, vrijednost ASN-a je poznata u svakom trenutku. ASN je potreban za izračunavanje frekvencije na kojoj se komunicira, zajedno sa kanalnim *offset*- om.

Vremenski slotovi su grupisani u jedan ili više *slotframe*-ova. *Slotframe*-ovi se ponavljaju tokom vremena. IEEE802.15.4e ne definiše tačnu veličinu *slotframe*-a. Zavisno od potreba aplikacije, može varirati od 10 do 1000s vremenskih slotova. Što je *slotfrejm* kraći, češće se određeni vremenski slot ponavlja rezultujući većim opsegom i manjim kašnjenjem, ali takođe i većom potrošnjom energije.

2.1.2.TSCH *scheduling*

U mreži se kreira mrežni raspored (*scheduling* funkcija), koji govori svakom uređaju koju funkciju treba da obavlja. U dodijeljenom slotu uređaj može biti aktivan (predaje/prima paket), ili u režimu uštede snage (*sleep* režimu). Dok je u *sleep* režimu, radio je isključen. Za svaki aktivni slot, *scheduling* funkcija definiše sa kojim susjednim uređajem se ostvaruje prenos, kao i koji je kanalni *offset*. Kada je uređaj aktivan u vremenskom slotu, dodjeljuje mu se MAC *slotframe* identifikator, kao i MAC adresa, [20].

Kada je paket generisan od strane višeg nivoa, predaje se nivou linka na čuvanje u predajnom baferu. U svakom slotu za predaju (TX slot), uređaj provjerava da li postoji paket za susjeda sa kojim je raspoređen. Ako nema paketa, isključuje radio u trajanju od jednog vremenskog slota, odnosno vraća se u *sleep* mod. Ako postoji paket, vrši prenos paketa i čeka potvrdu o uspješnom prijemu. U slučaju uspješnog prenosa (prijema poruke o uspješnom prijemu) paket se briše iz predajnog bafera. U suprotnom slučaju, uređaj čuva paket do ponovnog slanja. Postoji određeni broj pokušaja za slanje paketa prije brisanja iz predajnog bafera.

U svakom prijemnom slotu, uređaj uključuje radio tačno prije prijema paketa. U slučaju prijema, šalje potvrdu o uspješnom prenosu, isključuje radio i prosleđuje paket višem nivou na obradu. Ako nema paketa za prijem tokom trajanja *timeout*-a, vraća se u *sleep* mod. Ovo znači da predajnik nema ništa da pošalje ili da je došlo do gubitaka tokom prenosa.

U slučaju da između određenih korisnika postoji veći saobraćaj, *scheduling* funkcija može dodijeliti korisnicima više ćelija u istom *slotframe*-u. Modifikovanjem broja dodijeljenih ćelija mijenjaju se resursi koji su alocirani između susjeda. Dodavanjem/uklanjanjem ćelija između susjeda, uređaj može prilagoditi potrebe specifičnoj aplikaciji, [24].

Moguće je:

- kreirati "rijedak" raspored za aplikacije kojima je potreban rad sa velikom uštedom energije, po cijenu smanjenog opsega.
- kreirati "gust" raspored za aplikacije koje generišu veliku količinu podataka, po cijenu velike potrošnje energije.

2.1.3. Mehanizmi raspoređivanja

IEEE 802.15.4e definiše raspodjelu vremenskih slotova. Raspodjelu slotova je potrebno pažljivo napraviti tako da je uređaj A u slotu koji predaje, dok uređaj B u istom tom slotu osluškuje paket. Ako A više nije susjed sa korisnikom B (ako je pomjeren ili isključen), B ne bi trebalo da osluškuje pakete od korisnika A. Iako su ova pravila intuitivna, ilustruju potrebu za pažljivim kreiranjem rasporeda, konstantnim monitoringom kretanja uređaja u mreži i modifikacijom rasporeda ukoliko je to potrebno. Potrebno je napomenuti da IEEE 802.15.4e ne definiše kreiranje rasporeda.

Raspoređivanje se može obavljati na centralizovan ili distribuiran način [20]:

▪ Centralizovani pristup podrazumijeva postojanje centralnog entiteta koji održava mrežni raspored. Svaki uređaj u mreži ažurira upravljačkom uređaju listu drugih uređaja sa kojima može komunicirati i količinu podataka koju generiše. Uz pomoć tih informacija, menadžer kreira raspored. Kada je raspored napravljen, menadžer informiše svaki uređaj o linkovima u kojima učestvuju. Kada dođe do promjene (npr. uređaj izgubi svog susjeda), informiše sve na koje ta promjena utiče.

▪ U distribuiranom pristupu, uređaji samostalno odlučuju preko kojih će linkova komunicirati sa susjedima. Ovaj pristup se primjenjuje u mobilnim mrežama ili kada mreža ima previše *gateway*-a. Najjednostavnije rješenje jeste da svaki uređaj kreira raspored za svakog susjeda.

2.1.4. Sinhronizacija

Sinhronizacija između uređaja je potrebna za održavanje konekcije sa susjedima u TSCH mreži. Svaki uređaj periodično sinhronizuje svoj sat u odnosu na proizvoljno izabrani drugi uređaj. IEEE 802.15.4e standard ne definiše kako uređaj bira susjeda koji mu pruža podatak o vremenu, to zavisi od viših protokolskih nivoa, koji moraju voditi računa o kreiranju petlji prilikom ponovne sinhronizacije. Uređaji u mreži periodično šalju *Enhanced Beacons* (EB), u cilju objavljivanja svog prisustva mreži. Kada novi uređaj želi da pristupi mreži,

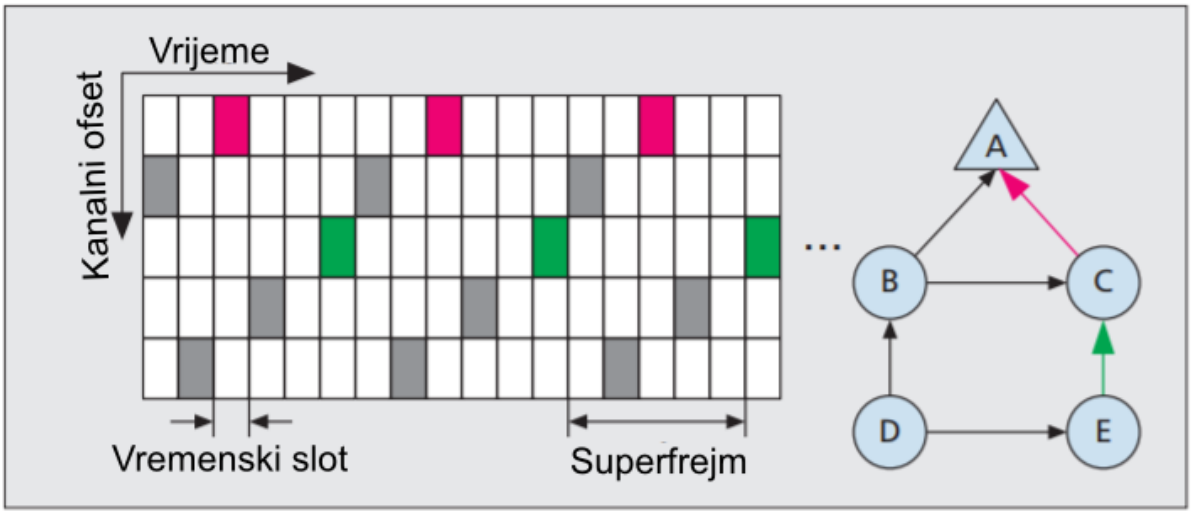
osluškuje EB poruke i sinhronizuje se. EB sadrže informacije o dužini vremenskog slota, *slotframe*-u i vremenskim slotovima koje posjeduje uređaj koji se oglasio.
IEEE 802.15.4e definiše dva metoda za sinhronizaciju uređaja u odnosu na svog susjeda:

1) Sinhronizacija bazirana na poruci potvrde (ACK - *Acknowledgement*)
2) Sinhronizacija na bazi razmjene frejma *(Frame-Based* sinhronizacija).

U oba slučaja prijemnik računa razliku u vremenu između očekivanog i stvarnog dolaska frejma. U sinhronizaciji koja se bazira na poruci potvrde, prijemnik upisuje izmjereni *offset* u polje ACK poruke i šalje predajniku. Na taj način izvršena je sinhronizaciju između prijemnika i predajnika. U sinhronizaciji na bazi razmjene frejma, prijemnik koristi izmjereni *offset* da podesi svoje vrijeme. U ovom slučaju, prijemnik je taj koji se sinhronizuje na sat predajnika, [20].

2.1.5. *Channel Hopping*

TSCH tehnika višestrukog pristupa, dodaje promjenu kanala (*channel hopping)* višestrukom pristupu IEEE802.15.4 MAC protokola. *Channel hopping* koristi frekvencijski diverziti, koji izbjegava interferenciju i *multipath feding*. Korišćenje višestrukih frekvencija povećava kapacitet mreže, jer omogućava transmisiju više frejmova u isto vrijeme, korišćenjem različitih kanala, slika 2.6. *Channel hopping* sa pristupom na bazi slotova povećava pouzdanost, jer pošiljalac i primalac mijenjaju frekvenciju pri svakoj transmisiji paketa. U slučaju da je transmisija bila neuspješna, retransmisija se vrši na drugoj frekvenciji. Takav pristup povećava vjerovatnoću uspješnog prenosa u odnosu na korišćenje iste frekvencije. Ideja *channel hopping*-a je ključ koji omogućava zahtijevanu pouzdanost za bežičnu komunikaciju u industrijskim IoT mrežama [7].



Slika 2.6 Raspored u mreži je podijeljen u domenu vremena i domenu učestanosti

Uređeni par*(t, chOf)* predstavljaju vremenski slot, kao i *channel offset* na određenom linku. *Channel offset* se sledećom jednačinom prevodi u frekvenciju:

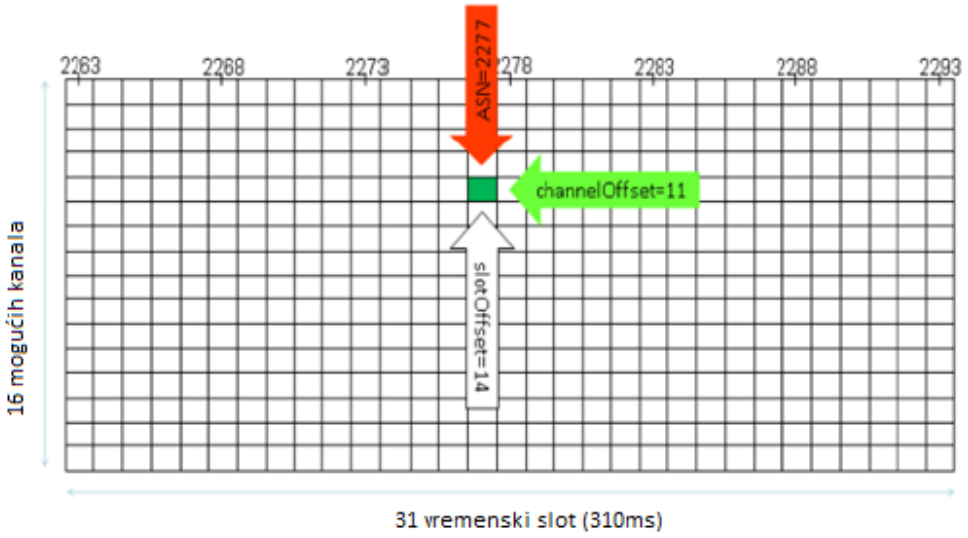
$$f = F\{(ASN + chOf) mod\ nch\}, \quad (2.1)$$

ASN predstavlja ukupan broj slotova od uspostavljanja mreže, koji se uvećava nakon svakog realizovanog slota, i poznat je svim uređajima u mreži. Analitički se može predstaviti kao:

$$ASN = (k * S + t), \quad (2.2)$$

gdje *k* predstavlja ciklus a *S* dužinu *slotframe-a*. Funkcija *F* se realizuje uz pomoć tabele. Vrijednost *nch* predstavlja broj dostupnih fizičkih frekvencija, kao i veličinu *look-up* tabele. Vrijednosti *t* i *chOf* moraju biti u okviru $0 \leq t \leq S\text{-}1$, $0 \leq chOf \leq nch\text{-}1$. U IEEE 802.15.4e standardu, broj dodijeljenih kanala je 16.

Na slici 2.7 je prikazana ćelija koja se nalazi u *slotframe-u*. Njen *slotOffset* je 14, *channelOffset 11* i ASN 2277. Prilikom komunikacije, uređaj primjenjuje relaciju (2.1) da izračuna frekvenciju. U sledećoj iteraciji (ciklusu *slotframe-a*), ćelija će imati isti *slotOffset* i *channelOffset,* ali se ASN promijenio, i primjenjivanjem (2.1), doći će do korišćenja nove frekvencije, [24].



Slika 2.7 Primjer ćelije

2.1.6. Formiranje mreže

Formiranje mreže uključuje dvije komponente: obavještavanje i pridruživanje. Novi uređaj pokušava da se pridruži mreži osluškujući frejmove obavještavanja. Kada je neki od frejmova primljen, novi uređaj se pridružuje slanjem *Join Request-a,* komandnog frejma uređaju koji se oglasio.

U centralizovanom upravljačkom sistemu, *Join Request* frejmovi se rutiraju do menadžera mreže. U distribuiranom sistemu se mogu slati direktno. Kada se novi uređaj priključio mreži, menadžer postavlja linkove između novog i postojećih uređaja u mreži. Linkovi se mogu modifikovati ili izbrisati nakon pristupanja mreži, [20].

Kada su se svi uređaji pridružili mreži, odnosno kada mrežni menadžer ne prima više *Join Request*-ove tokom *timeout* intervala, oglašavanje može biti isključeno. Nakon toga, procedura se može ponovo izvršiti, u cilju provjere da li postoji još uređaja koji žele da pristupe mreži.

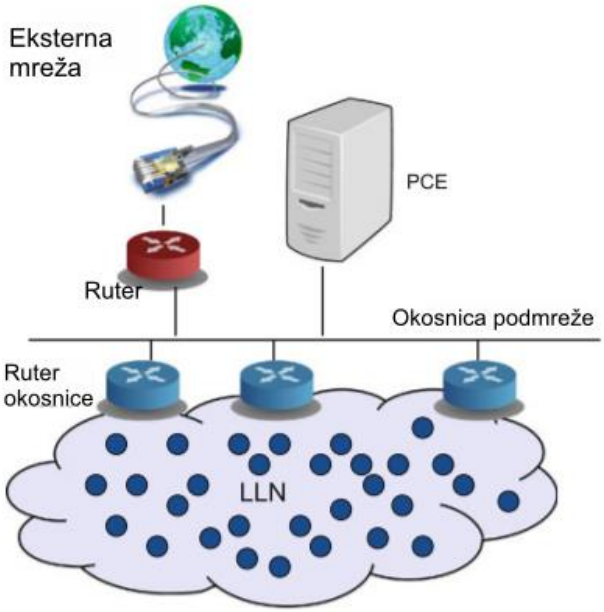
2.2. 6TiSCH

Mreže male snage sa gubicima (LLN - *Low-power and Lossy Network Architecture*) se mogu primjenjivati u različitim industrijskim okruženjima, gdje je u upotrebi do nekoliko desetina hiljada senzorskih čvorova, koji dijele isto radio okruženje. Memorija, kao i procesorske sposobnosti senzorskih uređaja, su ograničeni, pa sve funkcionalnosti, uključujući adresiranje i rutiranje, moraju biti svedene na minimum.

Industrijska automatizacija zahtijeva novu generaciju komunikacionih protokola, kompatibilnih sa IP, koji mogu zadovoljiti industrijske zahtjeve uz olakšanu interakciju sa kontrolnim centrom, kao i omogućiti prikupljanje podataka preko Interneta. Više od decenije, industrija se oslanjala na TSCH tehnologiju u ispunjavanju traženih zahtjeva kroz standarde kao što su WirelessHART i ISA100.11a. Mreže zasnovane na TSCH-u su pokazale pouzdanost od kraja do kraja (*end-to-end*) nivoa 99,999%, uz trajanje baterije duže od deset godina. Međutim, ove tehnologije ne mogu ispuniti zahtjeve današnjih industrijskih procesa uključujući funkcije upravljanja mrežom i orkestracije resursa.

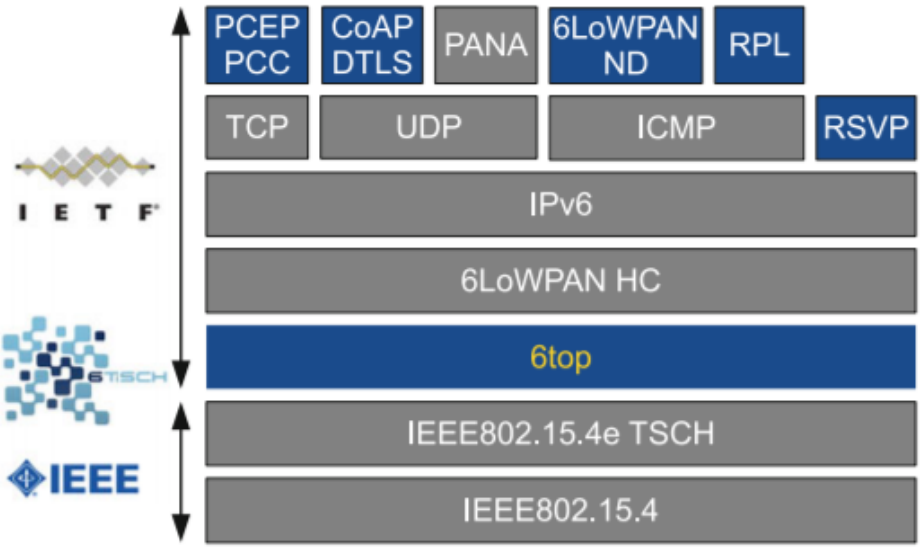
IETF (*Internet Engineering Task Force*) radna grupa 6TiSCH je osnovana da omogući upotrebu IPv6 sa TSCH tehnikom višestrukog pristupa IEEE 802.15.4 standarda, koji nudi industrijske performanse u smislu pouzdanosti i potrošnje energije. Radna grupa je kreirala arhitekturu u kojoj bežični uređaji male snage formiraju LLN mrežu. LLN se povezuje na Internet preko jednog ili više LLN graničnih rutera (LBR - *LLN Border Routers)* [24]. 6TiSCH je nadogradila pomenuti model, dokumentovala kao arhitekturu zasnovanu na otvorenim standardima, istakla najbolju praksu i standardizovala komponente koje nedostaju da bi se postigle performanse industrijskog stepena, u smislu varijacije kašnjenja, kašnjenja, skalabilnosti, pouzdanosti i rada sa uređajima male snage sa IPv6 preko IEEE 802.15.4e TSCH. Struktura balansira između propusnosti, kašnjenja i potrošnje energije, dok održava veliku pouzdanost. Integracija sa IP olakšava upravljanje mrežom i osigurava upotrebu u budućnosti kao i kompatibilnost tehnologije.

Kao što je ilustrovano na slici 2.8 arhitekturu mreže može činiti potencijalno veliki broj IPv6 podmreža, koje mogu obuhvatiti hiljade LLN uređaja, koji su ujedinjeni preko okosnice. Kada je to moguće, arhitektura koristi postojeće protokole kao što su IPv6 *Neighbor Discovery* (ND), IPv6 za bežične mreže male snage (6LoWPAN - *IPv6 over Low -Power Wireless Personal Area Networks*) i protokol za rutiranje za mreže male snage sa gubicima (RPL - *Routing Protocol for Low-Power and Lossy Networks*) sa minimalnom adaptacijom potrebnom za zadovoljavanje kriterijuma pouzdanosti i skalabilnosti preko okosnice [24].



Slika 2.8 Model arhitekture protokola za WSN

U cilju uvezivanja IEEE 802.15.4e TSCH nivoa sa IETF protokolima kao što su 6LoWPAN, RPL i IPv6 *Neighbor Discovery*, potrebno je raditi na standardizacionim 'rupama' i kreirati okvir koji će biti prikladan za bežične *mesh* topologije [7].



Slika 2.9 6TiSCH grupa protokola

IETF 6TiSCH skup protokola se oslanja na IEEE802.15.4 TSCH nivo linka koji dopunjuje sa setom protokola, kao što je opisano u 6TiSCH arhitekturi [25]. Slika 2.9 prikazuje glavne blokove koje čine 6TiSCH grupu protokola. Na samom "dnu" se nalazi fizički nivo sa

veoma malom potrošnjom, koji balansira između energetske efikasnosti, opsega i protoka podataka. Na njega se oslanja IEEE802.15.4e MAC nivo, koji se zasniva na vremenskoj sinhronizaciji i *channel hopping*-u, koji omogućava veliku pouzdanost dok održava nizak *duty cycle*. U sledećem sloju, arhitektura implementira 6LoWPAN kao nivo adaptacije [26]. Ovaj sloj omogućava grupi protokola da komprimuje IPv6 zaglavlja. Procedura pruža sigurnost da će frejmovi standarda IEEE 802.15.4e imati dužinu od najviše 127 bajta, odnosno smanjiće se obim informacija potrebnih za prenos [27]. U osnovi, ideja je da se iz zaglavlja uklone informacije koje nisu ključne, dok se druge informacije komprimuju, poput izvorišne i destinacione adrese. Ovaj sloj takođe ima mehanizam koji se naziva Granični ruter male snage (LBR - *Low-Power Border Router*), koji vraća zaglavlje na standardnu veličinu IPv6 zaglavlja [28].

RPL se implementira na sledećem nivou. Protokol za rutiranje može podržati linkove sa velikim gubicima i obično se koristi u kombinaciji sa ruterima koji imaju vrlo ograničene resurse, kao u automatizaciji zgrada/kuća i u industrijskim okruženjima. Vrši lako uspostavljanje ruta, distribuiranje informacije između uređaja i adaptaciju na veoma efikasan način. Sa predstavljenim karakteristikama, pogodan je za *smart grid* komunikacije. Ovaj protokol je osmislila radna grupa IETF ROLL (*Routing Over Low power and Lossy networks*) i posebno je razvijen za rad sa mrežama male snage [29]. Uređaji u mreži su povezani putanjom sa više hopova (*multi-hop*) do grupe uređaja koji služe za skupljanje podataka i koordinaciju. RPL kreira usmjereni aciklični (bez petlji) DODAG (*Direction-Oriented Directed Acyclic Graph*) graf za svaki uređaj, sa definisanim težinskim faktorima na linkovima i atributima, tj. informacijama o statusu čvorova. Rutiranje se uspostavlja na osnovu metrike koja računa udaljenost svakog uređaja u odnosu na njegov *root.* Težinski faktor na pojedinačnom linku se smanjuje duž grafa bliže destinacijom čvoru [20]. RPL može funkcionisati sa različitim vrstama saobraćaja i signalne informacije koje se razmjenjuju između uređaja zavise od zahtijevanih tokova podataka.

CoAP (*Constrained Application Protocol*) protokol definiše podskup RESTful (*Representational State Transfer*) specifikacija, adaptiranih za ograničenja prisutna u senzorskim mrežama [30]. Ovaj protokol je konstruisan kao zaglavlje koje se postavlja na UDP protokol i omogućava implementaciju aplikacija koje se mogu postaviti na uređaje za istraživanje funkcija hardvera i softvera. CoAP cilja da ponudi jednostavnost, veoma mala

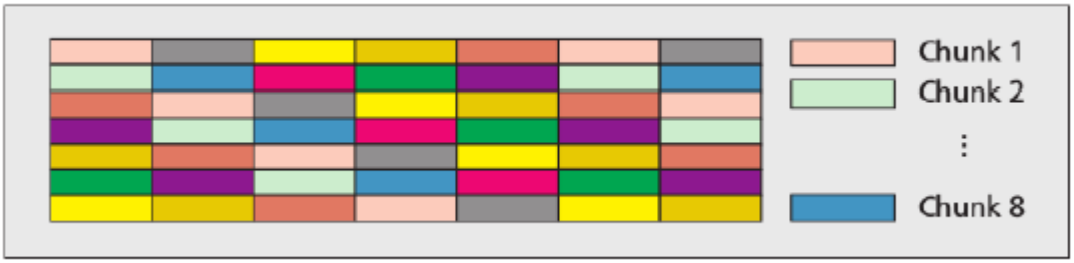
zaglavlja i ograničenu fragmentaciju a da održi interoperabilnost sa HTTP (*Hypertext Transfer Protocol*) protokolom.

IEEE 802.15.4e TSCH definiše koji čvor izvršava raspoređivanje (*scheduling*), ali ne govori kako se kreira i održava taj raspored. Slično, RPL organizuje postojeću topologiju u *multi-hop* rutirajuću strukturu, ali nije jasno kako je to usklađeno sa nivoom linka. U predloženom komunikacionom okviru nedostaje sloj koji dozvoljava *scheduler*-u da upravlja rasporedom u mreži.

U TSCH mreži, *scheduler* kontroliše komunikaciju u mreži. Izgradnja komunikacionog rasporeda uključuje dodjeljivanje vremenskih slotova susjedima. Pridruživanje višestrukih vremenskih slotova istim susjedima povećava propusnost (broj paketa koji susjedi mogu razmijeniti u jedinici vremena) i smanjuje kašnjenje u komunikaciji. Međutim, pošto je radio češće uključen, to rezultuje u povećanju prosječne potrošnje energije, odnosno u kraćem trajanju baterije.

IETF 6top je kontrolni nivo, koji se nalazi između IP nivoa i nivoa linka, a čija je uloga apstrakcija linka koji je potreban za IP operacije. Uključuje funkcionalnost prikupljanja statistika, koja višem nivou (RPL protokol rutiranja) može koristiti za informaciju o konektivnosti. Pruža višim nivoima komande upravljanja u cilju konfigurisanja i omogućavanja podrške traženog kvaliteta servisa QoS (*Quality of Service*) na MAC nivou. Monitoring proces se koristi za detekciju greške u radu mreže. 6top posjeduje interfejs koji omogućava pristup rasporedu ćelija u *slotframe*-ovima eksternoj upravljačkoj jedinici i dozvoljava dodavanje komplementarne funkcije, [31].

6TiSCH definiše novi koncept koji se naziva *Channel distribution/usage (CDU)* matrica. CDU je matrica u kojoj su kolone jednake broju dozvoljenih frekvencija (indeksirano sa *ChannelOffset*-om) dok vrste predstavljaju broj vremenskih slotova (indeksirano sa *slotOffset* -om). Presjek kolone i vrste čini ćeliju. Veličina ćelije je jednaka vremenskom slotu, koji može varirati od 10 do 15 milisekundi, kako bi obezbjedio transmisiju frejma i prijem potvrde uspješnog prenosa, uključujući sigurnosne provjere na prijemnoj strani. CDU matrica može biti podijeljena na djelove (*chunks*). Kao što se vidi na slici 2.10, *chunk* je niz ćelija koji dijeli CDU matricu po vremenu i frekvenciji i koji se koristi od istog komunikacionog para.



Slika 2.10 Primjer chunk-ova u CDU matrici

Pozicija *chunk-a* u CDU je poznata svim uređajima u mreži prilikom procesa usvajanja, koji se odvija nakon pregovaranja o dodatnim ćelijama. Uređaj koji odobrava *chunk* mora da odluči koja transmisija neće izazvati interferenciju. Generalizovano, *chunk* predstavlja dio opsega i može se posmatrati kao transmisioni kanal u vremenskom/frekvencijskom domenu.

6TiSCH uključuje i minimalnu konfiguraciju za pokretanje mreže [32], koja predstavlja scenario koji sadrži samo osnovne protokole koje je potrebno podržati, preporučene konfiguracije i režime rada dovoljne da omoguće funkcionalnu 6TiSCH mrežu. *Minimal Scheduling Function* (MSF) objašnjava način na koji se uređaj pridružuje mreži, kao i na koji način se kreira raspored komunikacije na distribuirani način [33].

Cilj 6TiSCH-a je kreiranje kontrolnog nivoa koji će upravljati komunikacionim rasporedom i prilagoditi ga potrebama u mreži. Nova arhitektura omogućava bežičnim uređajima, koji funkcionišu sa malom potrošnjom energije, da se ponašaju kao tradicionalni uređaji na internetu, imaju svoje IPv6 adrese, interaguju sa internet klijentima i serverima kroz standardizovane aplikativne protokole [7].

2.3. BeamLogic 802.15.4 Site Analyzer

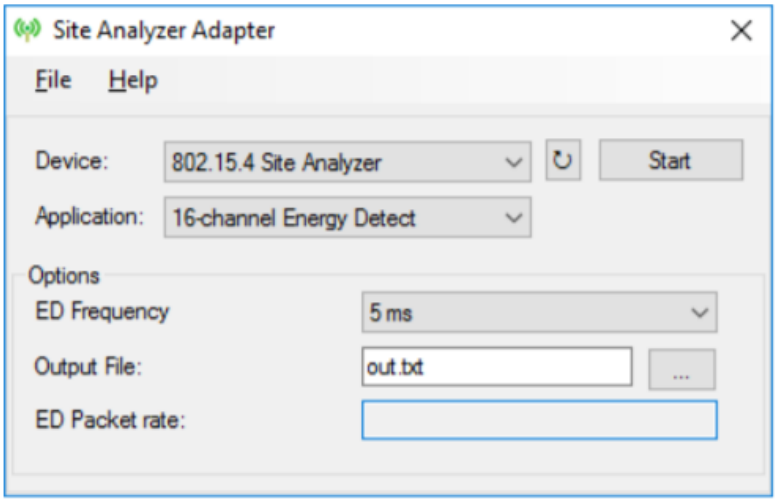
BeamLogic 802.15.4 Site Analyzer je uređaj koji se može koristiti za detektovanje i analizu IEEE 802.15.4 saobraćaja (slika 2.11). Prikaz hardverskih i softverskih karakteristika 802.15.4 Site Analyzer-a je prikazan u tabeli 2.2. BeamLogic je kreirao softver koji komunicira sa uređajem, vrši konfiguraciju, hvata razmijenjene pakete i prosleđuje ih analizatoru paketa Wireshark-u [34], ili ih upisuje u pripremljeni fajl [18]. Saobraćaj može snimati istovremeno na svih 16 kanala na 2.4 GHz opsegu. Pomenuta karakteristika je veoma značajna u mrežama gdje se koristi TSCH tehnika višestrukog pristupa, jer uređaji mijenjaju frekvenciju na kojoj

funkcionišu. Snifer se uz pomoć USB-a konektuje na računar koji pokreće specijalizovani softver.

Slika 2.11 BeamLogic 802.15.4 Site Analyzer

Kada snifer detektuje paket, on ga prihvata i šalje na računar na logovanje i obradu. 802.15.4 Site Analyzer takođe pruža i režim rada za 'Detekciju energije'. U ovom režimu rada, uređaj osluškuje istovremeno na svih 16 kanala i mjeri RSSI vrijednosti na skali od 0 do 32. Prikupljanje RSSI vrijednosti se može vršiti kontinualno, ili periodično, sa unaprijed definisanim periodom. 802.15.4 Site Analyzer je integrisan sa Wireshark-om, tako da snimljeni paketi mogu biti prikazani u trenutku prijema, ili sačuvani u lokalnoj memoriji na dalju analizu.

Pokretanje rada 802.15.4 Site Analyzer-a se vrši uz pomoć adaptiranog softvera SiteAnalyzerAdapter.exe, slika 2.12.



Slika 2.12 Grafički interfejs 802.15.4 Site Analyzer Adapter softvera

Tabela 2.2. Hardverske i softverske karakteristike BeamLogic 802.15.4 *Site Analyzer*-a

Hardverske karakteristike

Radna frekvencija	2.4GHz
Kanali	16 kanala simultano
Snimanje paketa	2.4GHz 802.15.4-kompatibilni saobraćaj
Detekcija energije	Obično preko 8 simbola
Osjetljivost prijemnika	-101 dBm
Interfejs	USB
Maksimalni protok podataka	do 1 MB/s

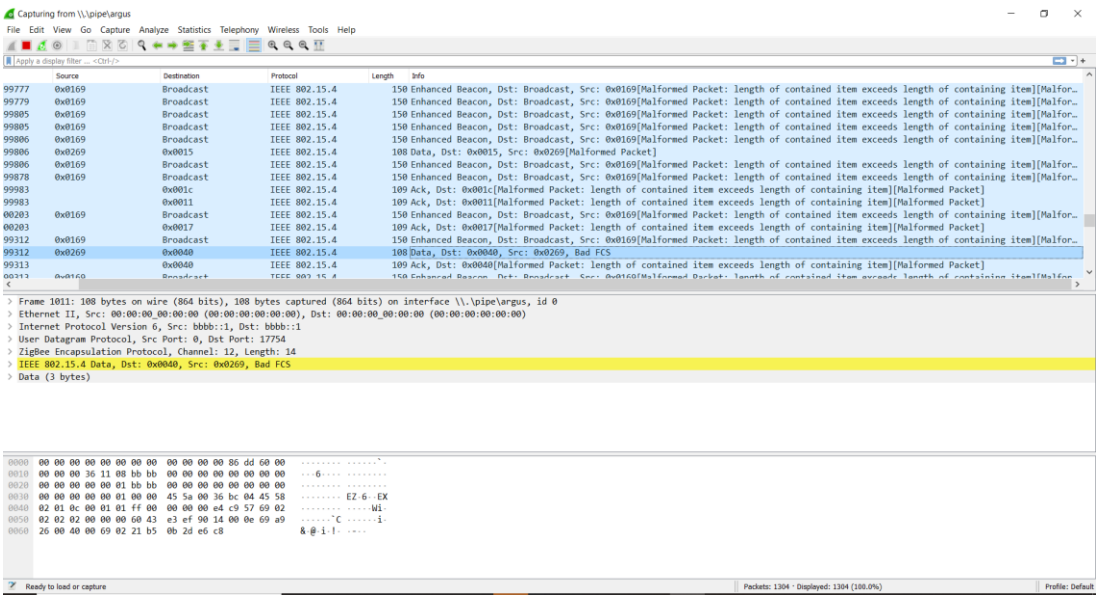
Softverske karakteristike

Podržani OS:	Windows, Linux
Integracija	Wireshark
Aplikacije	Snifer, Detekcija Energije

U nastavku će biti objašnjeni način pokretanja 802.15.4 Site Analyzer-a, kao i pojedina polja koja su prikazana na grafičkom interfejsu softvera:

- Polje *Device* - Uređaj: Ako je uređaj povezan u trenutku pokretanja aplikacije, naziv uređaja će se pojaviti u padajućoj listi u ovom polju. U suprotnom, padajući meni će biti prazan. Osvježavanje padajuće liste se radi uz pomoć dugmeta 'Osvježi' sa desne strane.
- Polje *Application* - Aplikacija: Trenutno postoje dvije podržane aplikacije: 16-kanalni snifer i 16-kanalni *Energy Detect*. 16-kanalni snifer sluša pakete na svim kanalima i prijavljuje kada je paket uspješno primljen. 16-kanalni *Energy Detect* sluša sve kanale istovremeno sa unaprijed definisanom učestanosti (može se izabrati sa padajuće liste) i objavljuje rezultujući RSSI na svakom kanalu.

- Polje *ED Frequency* – ED Učestalost: Ovo podešavanje predstavlja učestalost sa kojom će funkcionisati *Energy Detect*. Dostupne su sledeće opcije: Kontinuirano, 5 ms, 50ms, 100ms, 500ms, 1s, 2s, 5s, 10s.
- Polje *Output File* - Izlazna datoteka: Ovo polje postavlja izlaznu datoteku za tok detekcije energije. Nakon što je započeto mjerenje, rezultati će biti dodati u određenoj datoteci.
- Polje *ED Packet Rate* – ED brzina detekcije paketa: Ovo polje služi samo za pregled i prikazuje koliko će se mjerenja detekcije energije izvesti u sekundi.
- Polje Start - Startovanje: Ovo dugme se koristi za pokretanje aplikacije sa konfigurisanim parametrima. Za 16-kanalni snifer, pokreće se Wireshark i primljeni paketi će biti prikazani na grafičkom interfejsu. Za opciju detekcije energije, mjerenja se mogu sačuvati samo u datoteci u lokalnoj memoriji.



Slika 2.13 Prikaz Wireshark-a nakon aktivacije

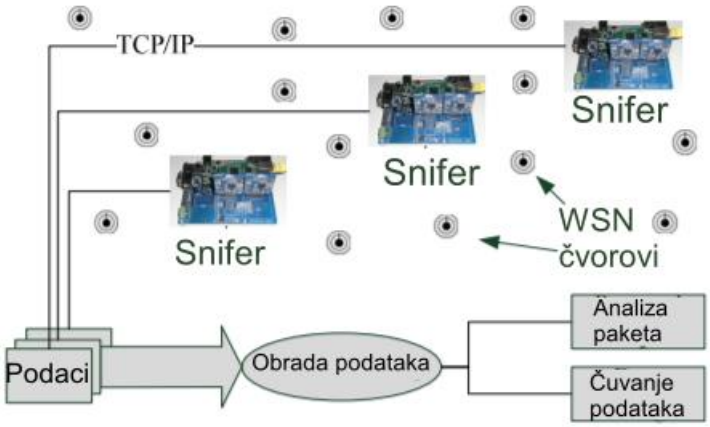
Paketi se prosleđuju Wireshark-u sa dodatim informacijama o RSSI, kanalu i preciznim vremenom (*timestamp*), slika 2.13. *Timestamp* je broj koji predstavlja vrijeme u kojoj je primljen indikator „*Start of Frame*“ izraženo u 1/3 mikrosekunde [18].

2.4. Dosadašnja istraživanja

Zbog karakteristika WSN, njihov razvoj zahtjeva alate za analizu rada i otklanjanje grešaka. Trenutno su u razvoju mnoga rješenja koja predlažu nove i efikasnije načine za

monitoring mreža. Osim toga, često postoji potreba da se taj snimljeni mrežni saobraćaj učini dostupnim za analizu što većem broju korisnika preko Interneta. Neka od realizovanih dosadašnjih rješenja će biti prikazana u nastavku.

SNDS (*Sensor Network Distributed Sniffer*) predstavlja distribuirani sistem za nadzor i analizu protokola velikih i složenih senzorskih mreža. Sniferi se postavljaju u mrežu koju je potrebno nadgledati. Uglavnom, koriste se za osluškivanje određenog kanala i prenos podataka do servisnog programa, koji analizira dobijene podatke i grafički ih prikazuje u realnom vremenu. Za prenos podataka koriste se Ethernet i TCP kako bi se obezbjedila stabilnost, fleksibilnost, kao i mogućnost prenosa u realnom vremenu [35]. Sniferi, kao i računar na kojem se analizira i prikazuje mrežni saobraćaj, se nalaze u istoj LAN (*Local area network*) mreži. Međutim, sniferi su ograničeni na nadgledanje samo jednog frekvencijskog kanala, što je značajan nedostatak, obzirom da bežične senzorske mreže često koriste višekanalnu komunikaciju.



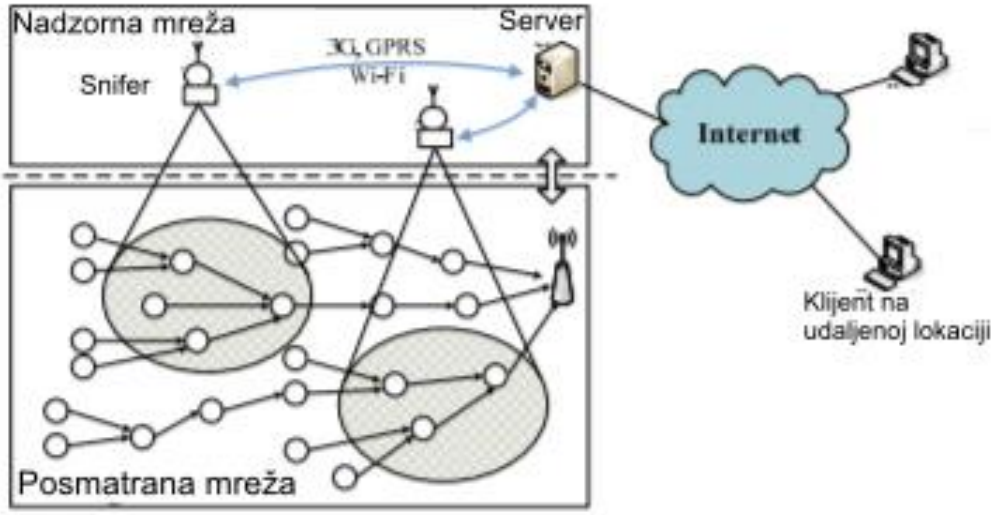
Slika 2.14 Arhitektura SNDS rješenja

Na slici 2.14 je prikazana arhitektura SNDS rješenja. Centralni dio predstavljaju mrežni sniferi koji snimaju pakete i prosleđuju ih servisnom programu na dalju analizu.

Još jedan od alata za bnadgledanje funkcionisanja bežičnih senzorskih mreža je NSSN (*Network monitoring and packet Sniffing tool for wireless Sensor Networks*). Arhitekturu NSSN čini: mreža koju je potrebno pratiti, nadzorna mreža koju čine sniferi, server na kojem se prikupljaju i čuvaju informacije o mreži, kao i klijenti koji se povezuju na server putem Interneta (slika 2.15). Sniferi osluškuju pakete koji su razmijenjeni i prosleđuju ih serveru. Sniferi mogu pratiti statuse uređaja u mreži, pronalaziti mrežne probleme i optimizovati

mrežnu konfiguraciju, bez uticaja na rad WSN. Na serveru se može vršiti obrada podataka, mrežna dijagnostika, mjerenje performansi, kao i vizualizacija podataka. Udaljeni klijenti se konektuju na server preko postojećih mreža, preuzimaju potrebne podatke i vrše analizu testnih podataka [36].

Kao i u SNDS rješenju, sniferi osluškuju pakete na određenim kanalima, odnosno nemaju mogućnost višekanalne detekcije. Monitoring se može vršiti uz pomoć softvera koji omogućava automatsko pretraživanje spektra i detekciju kanala na kojem WSN funkcioniše, ili uz ručno podešavanje kanala.



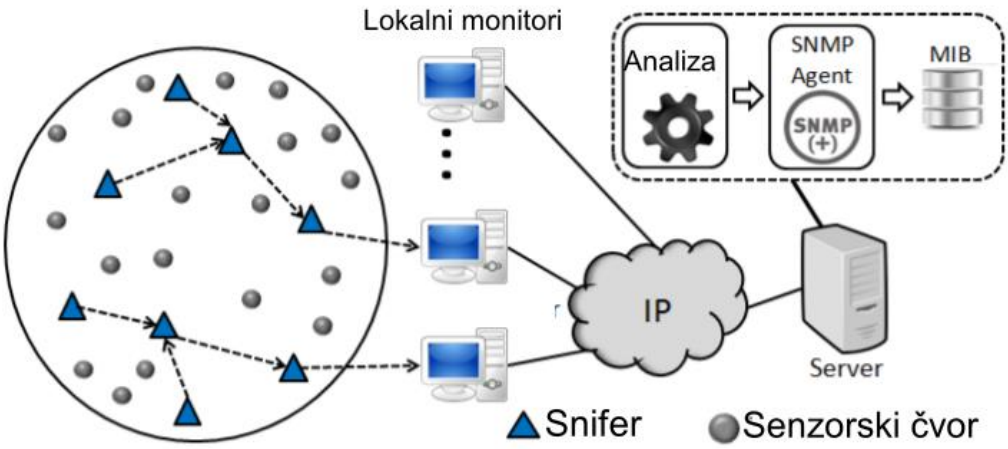
Slika 2.15 Arhitektura NSSN rješenja

Energetski efikasan pasivni sistem praćenja je neophodan kada je potrebno duže vremena nadgledati WSN u realnom scenariju. U suprotnom, vijek trajanja mreže za nadzor može biti mnogo kraći od vijeka trajanja ciljne mreže. Istraživači u radu [37], ukazuju na potrebu postojanja energetski efikasne mreže za monitoring bežičnih senzorskih mreža i predlažu sistem pod nazivom EPMOSt (*Energy-efficient Passive MOnitoring System),* koji pokušava da poveća energetsku efikasnost mreže za monitoring, dok pruža informacije o posmatranoj mreži uz pomoć SNMP (*Simple Network Management Protocol*) agenta.

Na slici 2.16 prikazan je EPMOSt sistem, u kojem je mreža sa sniferima raspoređena zajedno sa mrežom koju je potrebno nadgledati. Snifer hvata pakete, dodaje vremensku oznaku za svaki uhvaćeni paket i dopunjuje poruku za nadgledanje sa zaglavljem detektovanog paketa. Zatim šalje ovu poruku lokalnom monitoru, koristeći mrežu za nadzor. Lokalni monitor prima nadzorne poruke od više snifera i dodaje informacije u bazu podataka

koja se nalazi na serveru. Server analizira informacije koje generišu jedan ili više lokalnih monitora i dobija nekoliko informacija o posmatranoj mreži (vrijeme u kojem se svaki čvor budi, gubitak paketa, ponovno pokretanje čvorova, broj poslatih i primljenih paketa od svakog čvora itd.). Ove informacije su dostupne administratoru mreže, a takođe se čuvaju i upravljačkoj informacionoj bazi (MIB - *Management Information Base*) kojoj pristupa SNMP agent.

U odnosu na rješenje koje je predloženo u ovom radu, EPMOSt se fokusira na energetskoj efikasnosti. Stoga, umjesto uobičajene konfiguracije slanja detektovanih paketa, pruža mogućnost agregacije uhvaćenih paketa i slanje istih do najbližeg lokalnog računara. Takođe, sistem pruža algoritam za odabir broja aktivnih snifera. Algoritam predlaže najmanji broj potrebnih snifera u mreži i samim tim smanjuje potrošnju energije.

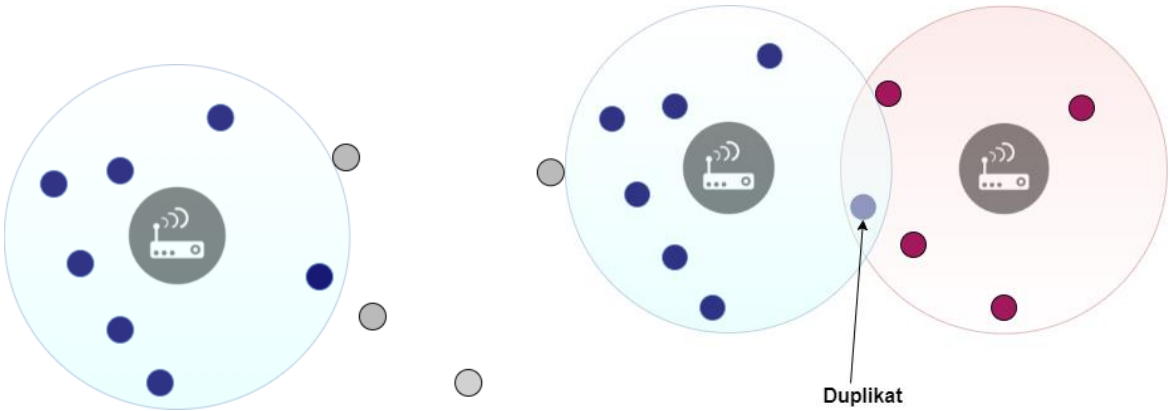


Slika 2.16 Arhitektura EPMOSt rješenja

Glava 3

Distribuirani snifer

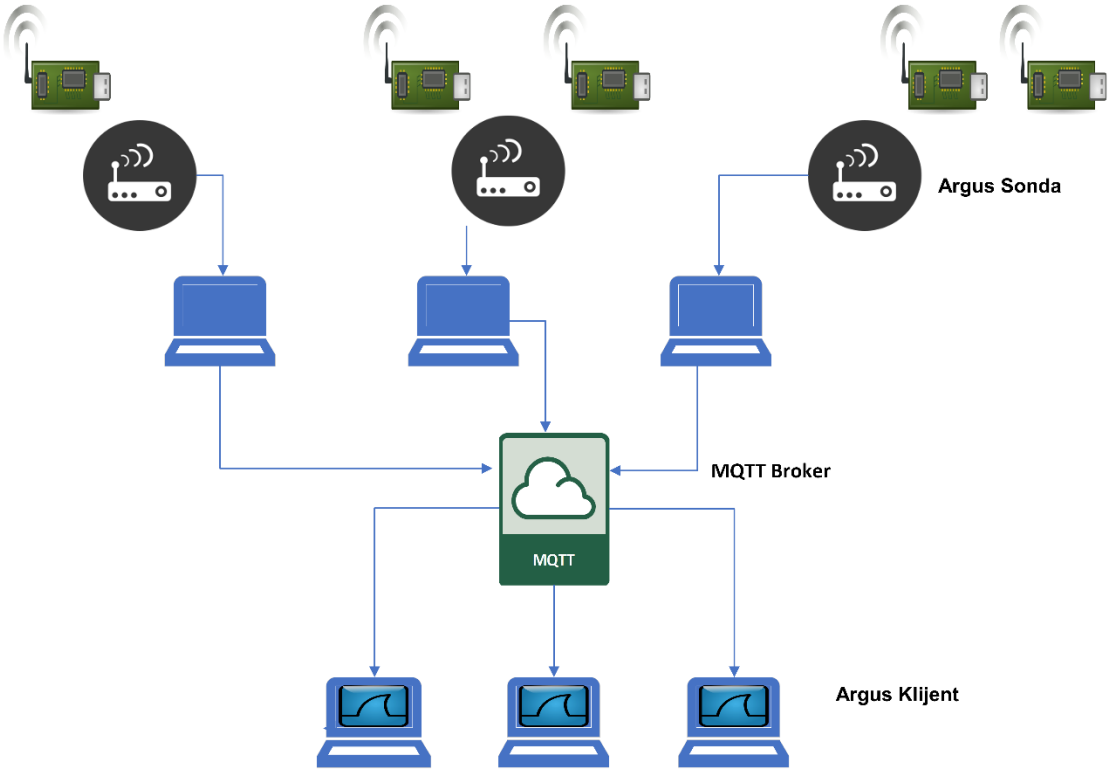
Eksperimentalni testbedovi predstavljaju neophodan alat za testiranje performansi novih protokola i mreža, prije procesa standardizacije ili industrijske proizvodnje. U fazi testiranja rada mreže na eksperimentalnim testbedovima, uobičajena praksa je da se koriste sniferi za snimanje mrežnog saobraćaja. Svaki snifer ima prag osjetljivosti i postoji velika vjerovatnoća da paketi koji dolaze sa udaljenog dijela mreže neće biti uhvaćeni, ako se koristi samo jedan snifer. To može biti otežavajući faktor u analizi mreže koja pokriva veliku oblast i/ili više spratova u zgradi. Implementacija distribuiranog snifera predstavlja logičko rješenje problema. Međutim, pojedini sniferi će primiti kopije istih paketa, koji mogu biti pogrešno protumačeni, ako nije primijenjeno filtriranje na klijentskom dijelu. Na slici 3.1, sa lijeve strane, se može vidjeti sistem u kojem jedan sniferski uređaj nije u mogućnosti da uhvati sve pakete razmijenjene u mreži, dok se sa desne strane vidi upotreba dva snifera. U prikazanom sistemu, dva snifera su dovoljna za pokrivanje cijele mreže, ali postoji vjerovatnoća da će neki paketi biti detektovani od strane oba snifera. Na taj način se javljaju duplikati paketa u mrežnom analizatoru. Proširenje Argus rješenja - Distribuirani Argus, je prilagođeno za sinhrono korišćenje više snifera i ispravnu interpretaciju uhvaćenog saobraćaja, prezentujući korisniku jedinstveni prikaz saobraćaja u mreži.



Slika 3.1 Upotreba jednog snifera (lijevo), upotreba dva snifera sa pojavom duplikata paketa (desno)

3.1. Arhitektura

Predloženo rješenje na slici 3.2 koristi više snifera, koji simultano snimaju saobraćaj i uz pomoć MQTT protokola prosleđuju podatke do klijentskih računara. Na ovaj način, moguće je uhvatiti pakete koji se razmjenjuju u mreži koja se prostire na velikom području, što omogućava analizu velikih WSN mreža, gdje jedan snifer nije dovoljan. Neki paketi će vjerovatno biti snimljeni od strane više snifera. Kao rezultat, višestruke kopije istog paketa će biti prikazane na klijentskom računaru. Zbog toga, potrebno je filtrirati paketski saobraćaj na klijentskom računaru i prikazati realni uhvaćeni saobraćaj.

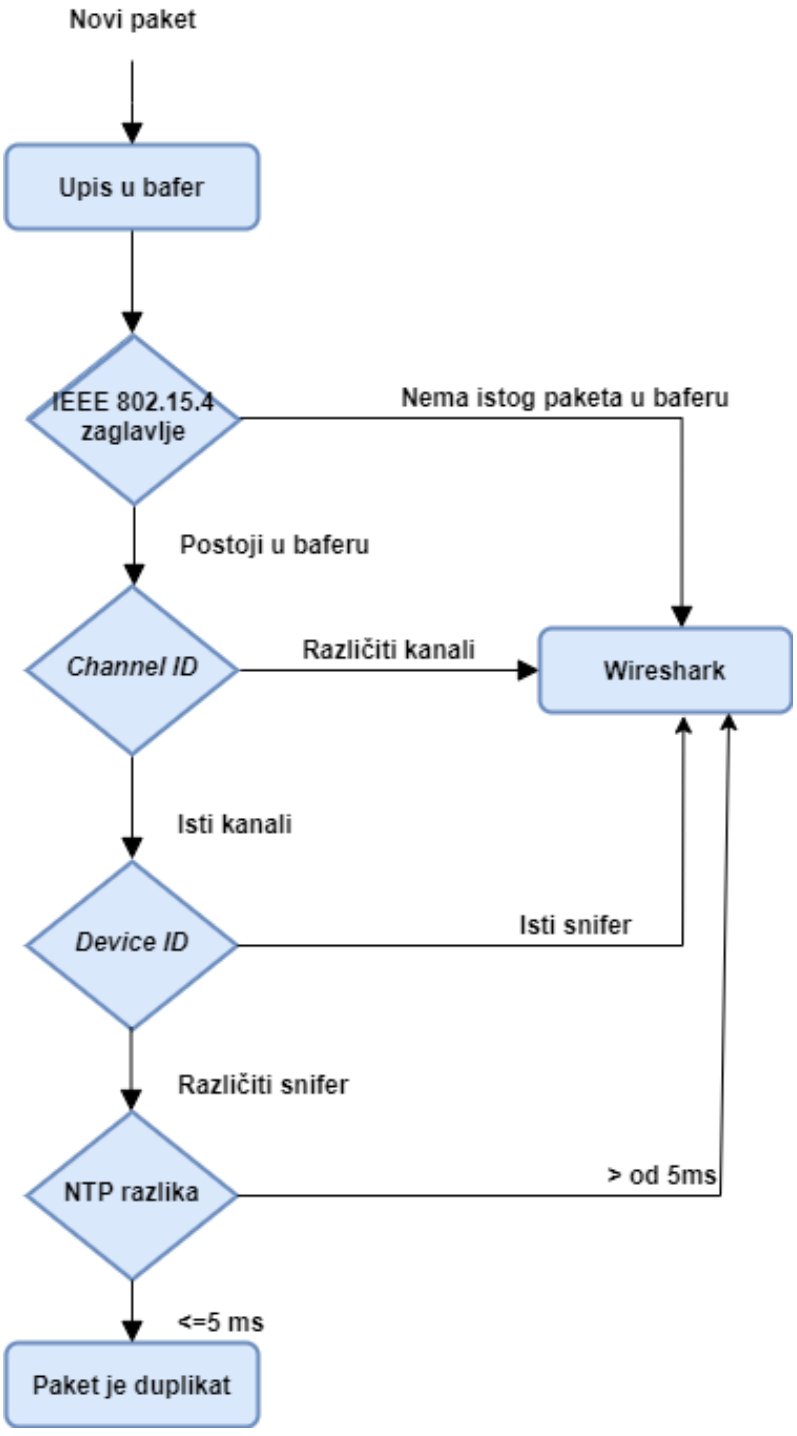


Slika 3.2 Distribuirani Argus koji koristi više snifera

3.2. Algoritam filtracije

Nakon pretplaćivanja na definisanu temu, MQTT broker šalje detektovane pakete svim zainteresovanim klijentima. Paketi, neposredno prije prikazivanja u mrežnom analizatoru, prolaze kroz funkciju koja se zove *DuplicateCheck* u Argus Klijentu [38]. Funkcija filtrira pakete koji su detektovani više puta, odnosno od strane više snifera. U cilju filtriranja paketa, na klijentskoj strani je kreiran bafer u kojem se čuvaju novi pristigli paketi. Veličina bafera se

može podešavati i treba se prilagoditi u odnosu na dinamiku mreže i broj uređaja. Inicijalno, veličina bafera je podešena na 200 paketa.



Slika 3.3 Algoritam filtracije paketa, funkcija DuplicateCheck

Filtriranje paketa na klijentskoj strani je podijeljeno u 4 faze. Algoritam filtracije je prikazan na slici 3.3. Na početku analize, IEEE 802.15.4 zaglavlje primljenog paketa se poredi sa istim segmentom paketa u baferu. Ako algoritam detektuje da isto zaglavlje ne postoji u baferu, zaključuje se da je u pitanju novi, jedinstveni paket. Paket se šalje mrežnom analizatoru Wireshark-u na prikaz klijentu. U suprotnom, analiza se nastavlja.

Dodatna analiza se vrši u slučajevima kada uređaji u senzorskoj mreži razmjenjuju popuno iste IEEE 802.15.4 pakete, kao na primjer tokom podešavanja mreže, *broadcasting*-a kao i specijalnog testiranja. Sledeće faze uključuju analizu polja iz ZEP zaglavlja. ZEP zaglavlje sadrži 12 polja od kojih se 3 koristi za filtriranje:

- *Channel ID,*
- *Device ID,*
- NTP *Timestamp*.

Polje *Channel ID* može uzimati vrijednosti između 11 i 26. Ovo polje predstavlja kanal na kojem je došlo do razmjene paketa u mreži. Ako analiza pokaže da su novi i paket iz bafera razmijenjeni na istom kanalu, to može značiti da je došlo do višestruke detekcije paketa i potrebno je nastaviti analizu. U suprotnom slučaju, paket se prosleđuje klijentu.

Sledeći parametar koji se analizira je *DeviceID*. *DeviceID* predstavlja jedinstveni identifikator svakog snifera. Prilikom povezivanja snifera sa računarom preko USB-a, *DeviceID* polje iz ZEP zaglavlja se automatski popunjava uz pomoć *Python* funkcije koja se naziva *random*. Ovaj metod se koristi kako bi se smanjila potrebna konfiguracija prije pokretanja programa za snimanje saobraćaja. Vjerovatnoća da će sniferima biti pridružena ista vrijednost *DeviceID* polja je:

$$p = 1 - N!\binom{2^{16}}{N}/2^{16N}, \tag{3.1}$$

gdje *N* predstavlja broj snifera koji se koristi u mreži, (:) predstavlja binomni koeficijent, dok je veličina polja *DeviceID* je jednaka 16 bita. Vjerovatnoća *p* definisana relacijom (3.1), se povećava sa povećanjem broja snifera u mreži i ova jednačina predstavlja slučaj koji se u matematici naziva *birthday problem* [39]. Ako su primljeni paket i paket iz zaglavlja snimljeni od strane istog snifera, zaključuje se da primljeni paket nije duplikat i može se prikazati klijentu. U suprotnom, analiza se nastavlja.

Sledeće polje koje se koristi u analizi je NTP (*Network Time Protocol*) zaglavlje. U ovom polju, računar koji pokreće Argus Sonda skriptu upisuje vrijeme kada je paket proslijeđen MQTT brokeru. Mjerenje vremena se vrši uz pomoć *Python* funkcije *time,* koja obezbjeđuje vrijeme lokalne mašine u milisekundama (ms). Kako bi predloženi sistem ispravno funkcionisao, svi računari koji pozivaju skriptu za snimanje saobraćaja moraju biti sinhronizovani. Kada klijentski računar primi dva paketa sa istim IEEE 802.15.4 zaglavljem, koji koriste isti kanal za transmisiju i uhvatili su ih različiti sniferi, potrebno je pogledati NTP *timestamp* polja. Ako se ove vrijednosti razlikuju za više od 5 ms, smatra se da paket nije duplikat. U ovom slučaju, novi primljeni paket se čuva u baferu i prosleđuje klijentu. Najstariji paket u baferu se uklanja i prilikom prijema novog paketa dolazi do ponavljanja čitavog procesa. Interval od 5 ms je izabran zbog nepreciznosti NTP protokola, kao i kašnjenja na serijskim portovima između snifera i kompjutera. Nakon filtracije, klijent u Wireshark-u može vidjeti jedinstvene pakete koji su razmijenjeni u mreži i saobraćaj se može snimiti za dalju analizu.

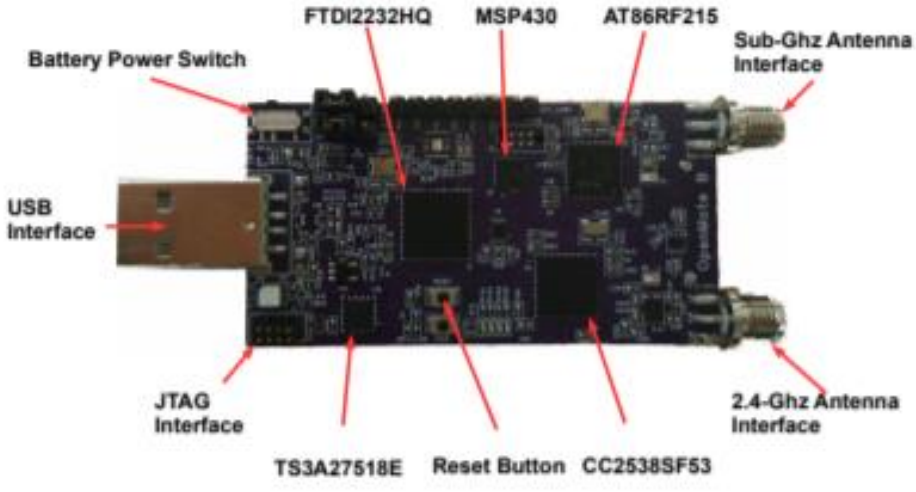
3.3. Implementacija distribuiranog snifera u Argus softveru

Za potrebe testiranja algoritma filtracije, neophodna su najmanje dva snifera. Međutim, u OpenTestbedu, gdje je u toku istraživanja vršeno snimanje mrežnog saobraćaja, dostupan je samo jedan BeamLogic 802.15.4 Site Analyzer. Stoga, ideja je da se iskoristi potencijal OpenMote B uređaja, odnosno da se određeni broj OpenMote B uređaja konfiguriše na način da predstavljaju snifere na po jednom kanalu.

3.3.1. OpenMote B

OpenMote B (slika 3.4) je platforma za razvoj hardvera i izradu prototipova za IIoT, posebno za istraživače i programere koji rade na sledećoj generaciji bežičnih mreža male potrošnje energije i velikog dometa zasnovanih na IPv6. Posjeduje mikrokontroler Texas Instruments CC2538 ARMCortex-M3, i odlikuje je istovremeni rad na više frekvencijskih kanala na ISM opsezima od 2.4 GHz i 868 / 915MHz. Podržava najnovije IEEE 802.15.4 standarde, uključujući MR-OFDM (*Multi-rate and multi-regional orthogonal frequency division multiplexing*) modulacije, IEEE 802.15.4g-2012, kao i FSK (*Frequency-shift keying),* OQPSK (*Offset quadrature phase-shift keying)* i OFDM (*Orthogonal frequency-division multiplexing)* modulacije. Sadrži dva primopredajnika: CC25381 IEEE802.15.4 radio, koji

koristi kanale na opsegu 2.4 GHz, i AT86RF215 2 IEEE802.15.4g radio, koji koristi kanale na opsegu ispod 1GHz.



Slika 3.4 OpenMote B platforma

OpenMote B je specijalno dizajniran za rad u testbedu [40]. U tabeli 3.1. su prikazane glavne karakteristike mikrokontrolera i primopredajnika koji se nalaze na ploči.

Karakteristike ove platforme su:

• Korisnički interfejs: Glavna ploča uključuje četiri LED diode (zelena, žuta, narandžasta i crvena) i korisničko dugme koje je namijenjeno za detektovanje grešaka. Osim toga, glavna ploča uključuje dugme za reset koje pruža mogućnost hardverskog resetovanja.

• Komunikacija preko serijskog porta: Glavna ploča može komunicirati sa računarom preko UART (*A universal asynchronous receiver-transmitter*) porta na CC25358. Rješenje se bazira na FTDI (*Future Technology Devices International Limited*) FT2232H čipu, serijskom-USB konverteru. U dodatku, FTDI čip omogućava programiranje CC2538 direktno preko internog programa za pokretanje i cc2538-bsl Python skripte.

• Proširenje ploče: Glavna OpenMote B ploča uključuje port za proširenje, koji sadrži 8 pinova sa 2.54 mm razmaka, a koji se mogu koristiti za detekciju grešaka ili povezivanje pomoćnih ploča, npr. OpenMote B ploče sa senzorima. Ploča za proširenje uključuje VCC (*Voltage Common Collector*) (2.5V) i GND (*Ground*) pin, kao i šest podesivih pinova.

• Sigurnosna podrška: Glavna OpenMote B ploča uključuje hardversku podršku za kriptografske funkcije koristeći SHA2, AES-128/256, ECC-128/256 i RSA algoritme.

Tabela 3.1. Karakteristike mikrokontrolera i promopredajnika na OpenMote B platformi

Mikrokontroler (Texas Instr, CC2538)	**Primopredajnik 1 (Texas Instr. CC2538)**	**Primopredajnik 2 (ATMEL, AT86RF215)**
ARM Cortex-M3 sa *pre-fetch* kodom - 16 MHz ili 32 MHz - 32 KB RAM - 512 KB FLASH **Ulazno-izlazni priključci:** - 4x za opštu upotrebu, 1x *sleep timer* - 1x 12 bit ADC sa 8 kanala - 2x SPI, 2x UART, 1x I2C **Režim uštede energije:** Aktivni mod: 7/13mA (16/32 MHz)	2.4 GHz ISM opseg sa podrškom IEEE 802.15.4-2006	868/915MHz i 2.4 GHz ISM opsezi sa podrškom IEEE 802.15.4g-2012
	Modulacija: OQPSK sa DSSS	Modulacija: MR-FSK/OFDM/O-QPSK
	Protok: 250 kb/s	Protok: 6.25 kb/s to 2400 kb/s
	Osjetljivost prijemnika: -97 dBm	Osjetljivost prijemnika: -123 dBm
	Snaga predajnika: 7 dBm	Snaga predajnika: 14.5dBm
	Struja predajnika: 24 mA na 0 dBm	Struja predajnika: 62 mA na 14 dBm
	Struja prijemnika: 20 mA	Struja prijemnika: 28 mA

Za firmver je korišćen openwsn-fw, koji je dio OpenWSN-a [41]. Firmver posjeduje već razvijene projekte koje mogu koristiti svi podržani uređaji. Jedan od projekata je *oos_sniffer*, koji na određenom kanalu osluškuje sve pakete i štampa na serijskom portu. Kako jedan uređaj može osluškivati na samo jednom kanalu, da bi se zamijenio jedan 802.15.4 Site Analyzer, potrebno je konfigurisati 16 OpenMote B uređaja. Prije početka rada sa hardverskim uređajima, potrebno je provjeriti da li sve ispravno funkcioniše u simulaciji.

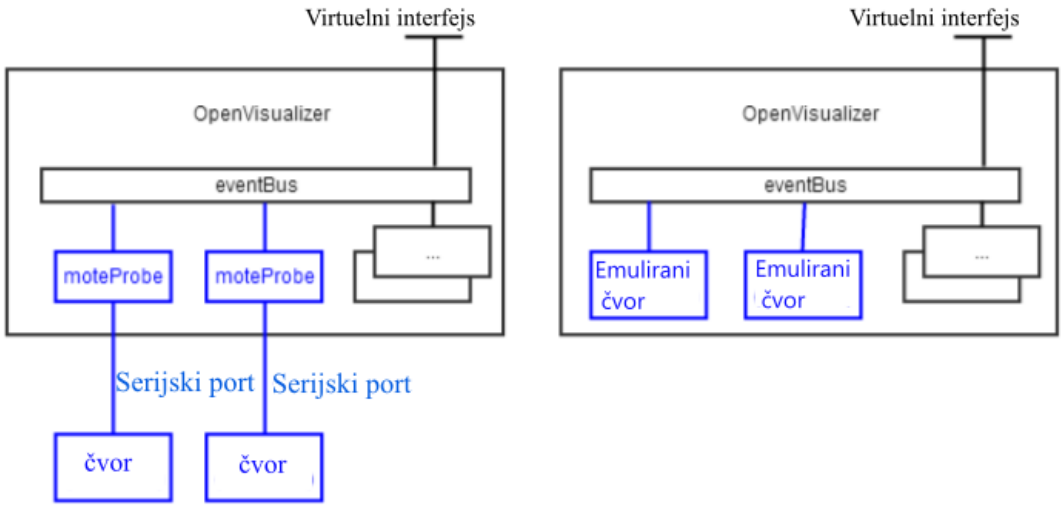
3.3.2. OpenSim emulator

OpenSim je emulator koji je dio projekta OpenWSN [42]. OpenSim kombinuje elemente iz OpenWSN firmvera i OpenWSN softvera. Emulirani senzori se kompajliraju, kako Python ekstenzija OpenWSN-a kreira instancu klase za svaki emulirani čvor i pokreće ih unutar OpenVisualizer-a.

Debugging informacije od hardverskih ili emuliranih uređaja se prenose preko serijskog porta i sakupljaju se uz pomoć OpenVisualizer-a. Za pokretanje OpenVisualizer-a potrebno je instalirati neophodne biblioteke. Konkretno, potrebno je instalirati PySerial, PyDispatcher,

PyWin32 (Python ekstenzija za Windows), Scons (koristi se za izvršenje naredbi) i TAP za Windows (IPv6 *tunnel drive*).

OpenSim je dio OpenVisualizer-a. OpenSim oponaša senzor koji generiše informacije i prosleđuje ih na magistralu podataka (EventBus). EventBus obezbjeđuje razmjenu poruka. Takođe, posjeduje komponente koje pružaju uslugu komunikacije bežičnih uređaja preko serijskog porta i eksterne korisničke aplikacije preko IPv6 TUN interfejsa. Emulirana mreža se ponaša potpuno isto kao i mreža sa stvarnim uređajima. Slika 3.5 prikazuje kako emulirani čvorovi (*mote*) komuniciraju sa EventBus-om.



Slika 3.5 Arhitektura OpenVisualizer-a sa stvarnim uređajima (lijevo) i sa emuliranim uređajima (desno)

Prilikom pokretanja OpenSim simulacije, emulirani čvorovi komuniciraju sa ostatkom OpenVisualizer arhitekture. OpenSim omogućava korisnicima da simuliraju OpenWSN mrežu bez fizičkih uređaja i emuliraju mrežu u Python i C programskom jeziku. OpenSim je kompatibilna sa Windows i Linux operativnim sistemima. Više emuliranih čvorova može se istovremeno pokrenuti pomoću OpenSim okvira za simulaciju.

3.3.3. Proširenje Argus softvera

Kako bi OpenMote B uređaji mogli uspješno da proslijede snimljene pakete do klijenata, potrebno je proširiti Argus Sonda skriptu. Postupak proširenja Argus Sonda skripte je tekao u dva koraka:

1. Nadogradnja skripte za interfejs sa sniferom priključenim lokalno, preko serijskog porta,
2. Nadogradnja skripte za interfejs sa OpenTestbed-om.

Zavisno od režima rada u kojem je potrebno pozvati skriptu Argus Sonda, prosleđuju se odgovarajući ulazni parametri. Za upotrebu 802.15.4 Site Analyzer snifera potrebno je u pozivu skripte dodati *--probetype beamlogic*. Korišćenje OpenMote B uređaja preko serijskog porta se poziva sa *--probetype serial --serialport COM3* (naziv serijskog porta na kojem je priključen OpenMote B uređaj), dok upotreba u testbedu zahtjeva *--probetype opentestbed --testbedmote 00-12-4b-00-14-b5-b5-bf* (EUI64- *Extended Unique Identifier* adresa uređaja koji se koristi kao snifer) [43].

OpenWSN uređaji, u ovom slučaju sniferi, enkapsuliraju podatke koji se prenose preko serijskog porta u HDLC formatu (*High-Level Data LinkControl*). To znači da su podaci (statistika sa uređaja, sadržaj paketa i slično) enkapsulirani u HDLC frejmove i da posjeduju *check* sumu, kako bi se provjerilo da li je tokom prenosa preko serijskog porta došlo do greške. Kod treba da dekodira HDLC, vrši provjeru tipa serijske poruke i da na osnovu tipa serijske poruke dalje vrši obradu i prosleđuje je različitim komponentama (npr. TUN interfejsu za Wireshark, ili statističkoj obradi).

HDLC je grupa protokola nivoa linka, koja služi za prenos podataka između čvorova u mreži, a kreirana od strane Međunarodne organizacije za standardizaciju (ISO-*International Organization for Standardization*). Pošto se radi o protokolu nivoa linka, podaci su organizovani u frejmove. Frejm se putem mreže prenosi do destinacije koja potvrđuje njegov uspješan prijem [44]. Sastoji se od 6 polja, slika 3.6:

- Oznaka (*Flag*) - 8-bitna sekvenca koja označava početak i kraj frejma;
- Adresa (*Address*) - Sadrži adresu primaoca. Ako primarni čvor šalje frejm, on sadrži adresu sekundarnog čvora, dok ako ga šalje sekundarni čvor, polje sadrži adresu primarne stanice;
- Kontrola (*Control*) - Sadrži informacije o protoku i grešci;
- Podaci (*Data)* – U ovom polju se nalaze korisni podaci. Njegova dužina može varirati od zavisno od potreba mreže;
- FCS (*Frame check sequence*) – Provjera frejma koja se može sadržati od 2 ili 4 bajta za otkrivanje grešaka. Standardni kod koji se koristi je ciklični redundantni kod (CRC- *Cyclic redundancy check*).

Slika 3.6 Struktura HDLC frejma

Klasa koja obezbjeđuje rad sa lokalno priključenim sniferom otvara serijski port, čita snimljene pakete sa porta, dekodira HDLC pakete i prosleđuje na dalju obradu. U Pseudokodu 3.1 prikazano je otvaranje serijskog porta sa definisanim protokom i čitanje podataka. Vrijednost koju treba proslijediti je naziv serijskog porta računara na kojem je priključen OpenMote B uređaj. Za svaki bajt primljene poruke, poziva se procedura *NewByte*.

Proširenje ArgusProbe skripte I - Otvaranje serijskog porta:

Ulaz: serial_port = COM3, *baudrate* = 115200 b/s

```
// Otvaranje serijskog porta
1 serial_handler=Serial(serial_port,baudrate)
2 while True do
3     waitingbytes = serial_handler.inWaiting()
4     if waitingbytes ≠ 0 then
          // Čitanje informacija sa serijskog porta
5         rx_bytes= serial_handler.read(waitingbytes)
6         for byte in rx_bytes do
              // Poziv funkcije za svaki novi bajt
7             newByte(byte)
```

Pseudokod 3.1 Otvaranje serijskog porta, čitanje podataka i prosleđivanje na analizu

Ova procedura je prikazana u Pseudokodu 3.2 HDLC frejm (slika 3.6) koristi oznaku (*flag*) za označavanje njegovog početka i kraja. U OpenTestbedu, za oznaku početka i kraja je korišćena ista oznaka – '~' (\x7e u heksadecimalnom zapisu). Procedura *NewByte*, na osnovu pomenute oznake detektuje HDLC frejm. Bajtovi detektovanog paketa se upisuju u promjenljivu, koja prije nastavka obrade mora proći kroz validaciju. Greške u detekciji se mogu javiti, jer je moguće da će procedura detektovati prvo kraj jednog paketa, a zatim početak sledećeg.

Proširenje ArgusProbe skripte II - Detekcija HDLC frejma:

```
   Ulaz: Byte
1  HDLC_flag = ~
2  prijem = False
3  hdlc_flag = False
   // Parsiranje primljenih bajtova na serijal portu
4  if prijem ≠ True then
       // Prijem paketa nije još počeo
5      if hdlc_flag = True ∧ Byte ≠ HDLC_flag then
           // Prvi bajt paketa nakon HDLC oznake
6          prijem = True
7          hdlc_flag = False
8          rxBuffer += Byte
9      else if Byte = HDLC_flag then
           // HDLC oznaka na početku frejma
10         rxBuffer = []
11         hdlc_flag = True
12     else
13         Error
14 else
       // U toku je prijem paketa
15     if Byte ≠ HDLC_flag then
           // Sredina frejma
16         rxBuffer += Byte
17     else
           // Kraj frejma
18         hdlc_flag = True
19         prijem = False
20         rxBuffer += Byte
21         valid_frame = handle_frame()
22         if valid_frame = True then
               // Paket je ispravan
23             hdlc_flag = False
24             newFrame(rxBuffer)
```

Pseudokod 3.2 Funkcija NewByte: Detekcija HDLC frejma

Funkcija validacije u pseudokodu 3.3 kao ulazni parametar ima detektovani HDLC frejm. Za dalju analizu potrebno je prvo ukloniti prvi i poslednji bajt, koji predstavljaju oznaku. Zatim je potrebno provjeriti *check* sumu. *Check* suma nosi informaciju o uspješnom prenosu ili o grešci koja se potencijalno desila. Ako je provjera uspješna, potrebno je ukloniti i provjeriti prvi bajt korisne informacije u paketu. Prvi bajt označava tip poruke i u OpenTestbed mreži paketi koji u sebi sadrže korisne informacije su označeni sa karakterom 'P'. Stoga, sve poruke

koje su razmijenjene sa drugim tipom se odbacuju. Ako paket prođe sve provjere, prosleđuje se na dalju analizu. Ako paket ne ispunjava sve uslove, briše se iz prijemnog bafera.

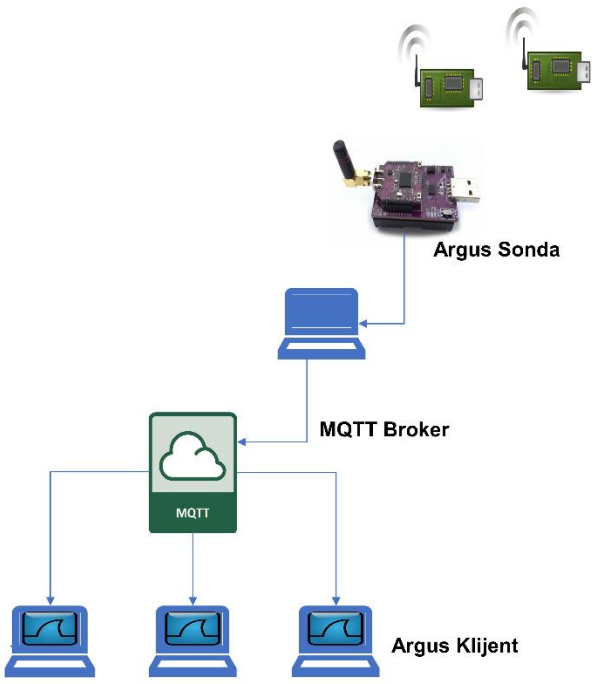
Proširenje ArgusProbe skripte III - Validacija HDLC frejma:

Ulaz: rxBuffer

```
// Uklanjanje prvog i poslednjeg bajta
rxBuffer = rxBuffer[1:-1],
// Provjera ček sume
crc = check_CRC(rxBuffer[:-2])
if crc = True then
    // Uspješna CRC provjera
    rxBuffer = rxBuffer[:-2]
    // Posmatraju se paketi koji imaju oznaku P
    if rxBuffer[0] = P then
        rxBuffer = rxBuffer[1:]
        valid_frame = True
else
    Error: Pogrešna ček suma
```

Pseudokod 3.3 Validacija HDLC frejma

Nakon dekodiranja, potrebno je paket prilagoditi za slanje Argus brokeru. Paketi koji se šalju od strane 802.15.4 Site Analyzer-a su enkapsulirani sa ZEP zaglavljem, stoga je potrebno kreirati zaglavlje na isti način. Paketi se šalju preko MQTT protokola sa odgovarajućom temom, slika 3.7.



Slika 3.7 Arhitektura Argus rješenja koji kao snifer na jednom kanalu koristi OpenMote B

Klasa koja radi sa OpenTestbed-om, umjesto čitanja sa serijskog porta, prima podatke preko MQTT-a. OpenTestBed se sastoji od 'OtBox' paketa, koji je prikazan na slici 3.8. [40]. OtBox sadrži:

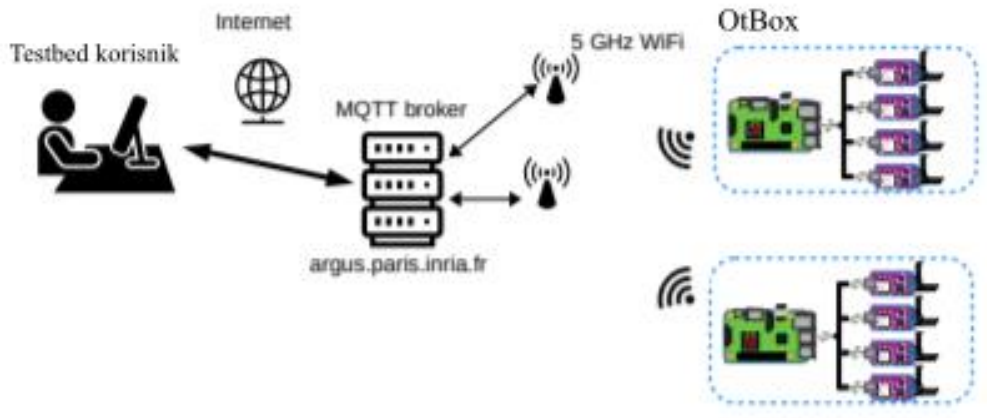
- Raspberry Pi – „Mali“ kompjuter koji pokreće OpenTestbed softver i povezuje se sa *back-end* mrežom preko WiFi konekcije. Za WiFi komunikaciju se koristi opseg od 5 GHz, kako bi se spriječila pojava interferencije sa OpenMote B uređajima koji funkcionišu na 2.4GHz;
- 4 OpenMote B uređaja;
- Ekran;
- QR kod (*Quick Response code*).

Svaki OtBox pokreće *otbox.py* Python program [45]. Uz pomoć programa, Raspberry Pi komunicira sa OpenMote B uređajima preko serijskog porta i nudi servise korisniku preko jednostavnog API-ja (*Application programming interface*):

- Menadžment uređaja – obuhvata reprogramiranje koda sa firmverom, resetovanje uređaja, kao i onemogućavanje rada uređaja;
- OtBox upravljanje: prikupljanje informacija o statusu OtBox-ova, prikupljanje informacija o MAC adresi povezanih uređaja, nadogradnja softvera, mijenjanje slike koja se prikazuje na ekranu;
- Prosljeđivanje sa serijskog porta: objavljivanje bajtova koji su poslati od strane sniferskog uređaja, kao i slanje bajtova korisnicima.

Slika 3.8 OtBox: Raspberry Pi, 4 OpenMote B, ekran i QR kod u staklenom zvonu

OtBox se konektuje na centralni MQTT broker, u uobičajenoj konfiguraciji klijent-server. Korisnik može pokrenuti eksperiment konektovanjem na pomenuti broker, zadati komande koje je potrebno izvršiti i primiti informacije koje šalju OtBox-ovi. Kao što se vidi na slici 3.9, ne postoji server za OpenTestbed.



Slika 3.9 Arhitektura OpenTestbed-a

Arhitektura Argus-a u režimu rada sa OtBox čvorovima izgleda kao na slici 3.10. OtBox čvorovi se nalaze na lokaciji testbeda. Pošto je potrebno snimati saobraćaj na svim kanalima, neophodno je koristiti 16 OpenMote B uređaja, odnosno 4 OtBox-a. Svaki od njih je programiran da radi na različitom kanalu. Raspberry Pi svakog OtBox-a se koristi za prikupljanje informacija od uređaja preko serijskog porta i slanje na MQTT broker sa definisanom temom: *'opentestbed/deviceType/mote/deviceId'*. Taj korak predstavlja uobičajen način rada elemenata u testbedu. Informacije koje se prosleđuju od uređaja do brokera mogu biti podaci sa senzora, informacije o statusu čvorova (identifikator i rang čvora, podaci o sinhronizaciji, rasporedu čvora, susjedima i redu za prenos) iz mreže, dok u ovom slučaju, kada uređaji rade kao sniferi, informacije na MQTT brokeru predstavljaju uhvaćene pakete u mreži.

Izmjena u odnosu na postojeći model komunikacije jeste Argus Sonda skripta, koja se kao na slici 3.10, pretplaćuje na MQTT broker koji sadrži podatke proslijeđene putem Raspberry Pi-a, od OpenMote B uređaja. Cijeli kod proširene Argus Sonda skripte se može vidjeti na [46]. Na pseudokodu 3.4, uređaj se pretplaćuje na definisanu temu. Zatim, funkcija pokušava da prikupi podatke sa MQTT brokera i nakon toga ih prosleđuje u bafer (*queue).*

Proširenje ArgusProbe skripte IV - Rad sa OpenTestbedom:

Ulaz: mqtt_broker, testbedmote_eui64, Tema = 'opentestbed/deviceType/mote/deviceId'
Izlaz: Queue

```
// Uredjaj se pretplaćuje na temu
1 mqtt_connect(Tema,mqtt_broker,testbedmote_eui64)
2 try
    // Učitavanje paketa
3   bytes = load['serialbytes']
4 else
5   try
      // Upisivanje u queue
6     Queue.put(bytes)
7   except
8     Queue.Full
```

Pseudokod 3.4 Pretplaćivanje na temu i smještanje podataka u queue

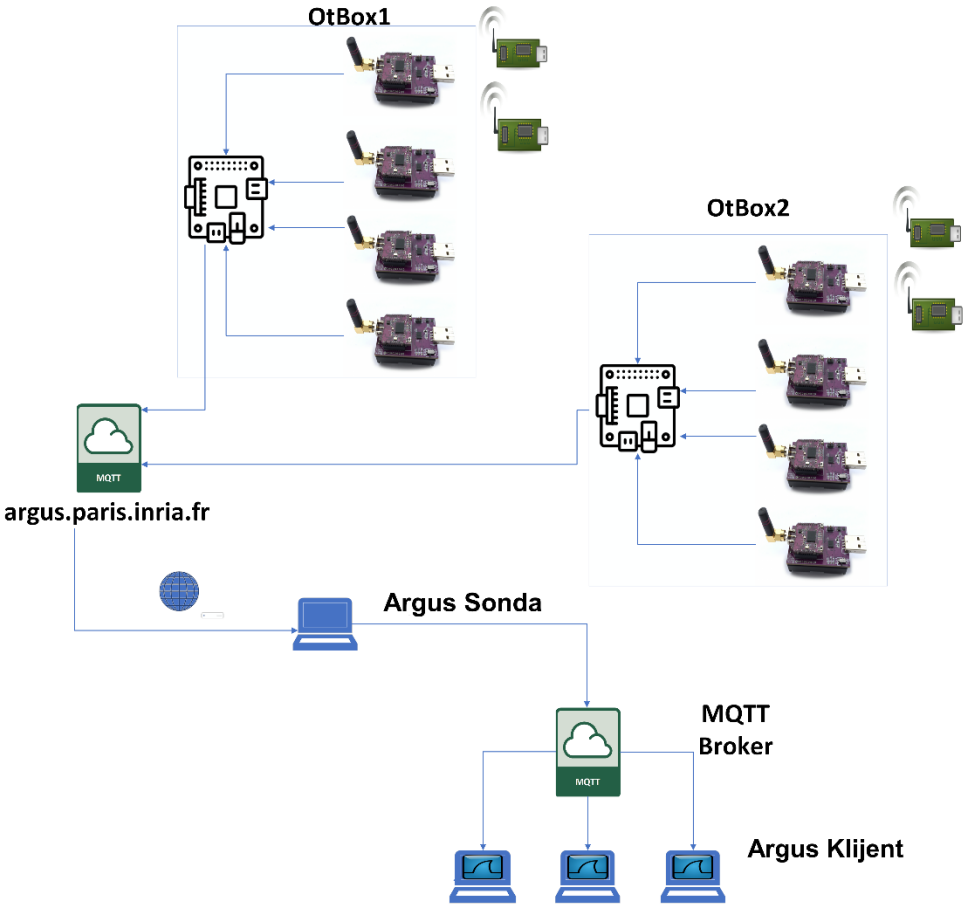
Nastavak obrade obuhvata preuzimanje podataka iz *queue*-a i poziv funkcije *NewByte* za svaki bajt detektovanog paketa (vidi Pseudokod 3.5). Nakon detekcije HDLC paketa i njegovog dekodiranja, vrši se dodavanje ZEP zaglavlja i slanje na MQTT broker, koji je standardni dio Argus konfiguracije. Klijenti sa udaljene adrese, pokretanjem Argus Klijent skripte, se konektuju na MQTT broker i primaju pakete u Wireshark analizatoru paketa. Izmjena Argus sistema nije uticala na klijentski dio, tako da je rad klijenata ostao isti.

Proširenje ArgusProbe skripte V - Prosledjivanje podataka iz queue-a:

Ulaz: *Queue*

```
1 rx_bytes = Queue.get()
2 for byte in rx_bytes do
    // Poziv funkcije za svaki novi bajt
3   newByte(byte)
4 newByte(byte)
```

Pseudokod 3.5 Prosleđivanje podataka iz queue-a



Slika 3.10 Arhitektura Argus klijenta kada radi sa OpenTestBed-om

Opisana nadogradnja Argus rješenja u konfiguraciji sa slike 3.10 je omogućila eksperimentalno testiranje predloženog algoritma filtriranja istih paketa snimljenih od strane 2 ili više susjednih snifera. Argus Sonda skripta je proširena da radi sa OpenMote B uređajima iz OpenTestbed-a, koji su konfigurisani da obavljaju ulogu snifera na jednom kanalu. Skripta prosleđuje detektovane pakete do svih zainteresovanih klijenata. U eksperimentu koji je vršen korišćena su 4 Otbox-a, odnosno 16 OpenMote B uređaja koji osluškuju na 16 različitih kanala i na taj način mogu zamijeniti jedan Site Analyzer. Site Analyzer, kao i OpenMoteB sniferi, su postavljeni u istoj kancelariji, na približno istom mjestu, kako bi se njihov uhvaćeni saobraćaj mogao uporediti i analizirati.

Glava 4

Verifikacija predloženog rješenja

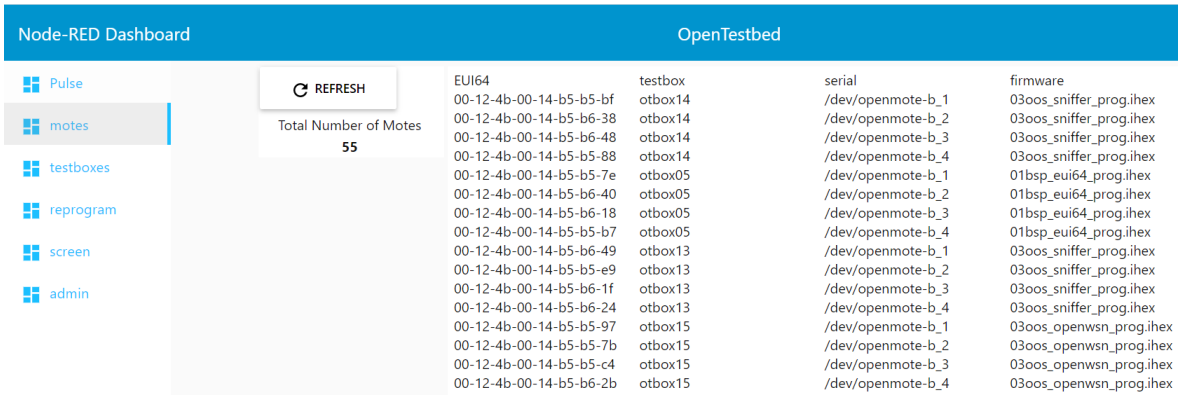
Kao i svaka mrežna tehnologija, IoT rješenja moraju biti detaljno testirana i potvrđena, prije upotrebe u realnim aplikacijama. Pokretanje simulacije koja modeluje funkcionisanje IoT mreže je generalno prvi korak. Drugi korak je pokretanje firmvera, koji se u potpunosti implementira na testbedu. Iako je testbed skup bežičnih uređaja male snage u kontrolisanom okruženju, ključna usluga koju nudi je mogućnost učitavanja novog firmvera na uređaje i posmatranje njihovog ponašanja u mreži. Naprednije usluge uključuju rezervaciju testbeda od strane korisnika, *toolchain* za kompajliranje firmvera, mjerenje nivoa utroška energije i skladištenje logova.

Rad funkcije filtriranja provjeren je simulacionim i eksperimentalnim putem. Za potrebu simulacionog testiranja, korišćen je snimak saobraćaja uhvaćen uz pomoć 802.15.4 Site Analyzer-a na Inria-Paris testbedu, koji se sastoji od uređaja koji se baziraju na IEEE 802.15.4 standardu. Paketi se obrađuju uz pomoć Python skripte, koja koristi pomenuti snimak saobraćaja.

Simuliran je scenario gdje više snifera u mreži šalje prikupljene podatke na MQTT broker i primjenjuje se filtriranje na prijemu. *DuplicateCheck* funkcija iz Argus Klijenta provjerava primljene pakete i vraća vrijednost *False*, ako je paket duplikat. U suprotnom slučaju, funkcija vraća *True*. Tokom testiranja, sve faze filtriranja su primijenjene. Problem koji se može javiti jeste zagušenje MQTT brokera, koji ne može obraditi sve pristigle pakete. Zbog toga se može desiti da neki paketi ne budu proslijeđeni klijentima. U tom slučaju, skripta pokazuje poruku o zagušenju. Takođe, veoma je bitno da uređaji koji pokreću program imaju sinhronizovano NTP vrijeme, sa preciznošću ispod nekoliko milisekundi. Ako sinhronizacija nije izvršena sa ovim nivoom preciznosti, filtracija na klijentskom dijelu neće biti uspješna. Da bi se olakšao problem sinhronizacije, moguće je konfigurisati NTP server na jednom od računara sa kojima su sniferi povezani. Konfigurisani računar postaje server za sve ostale računare u mreži, dok on preuzima vrijeme od eksternog NTP servera.

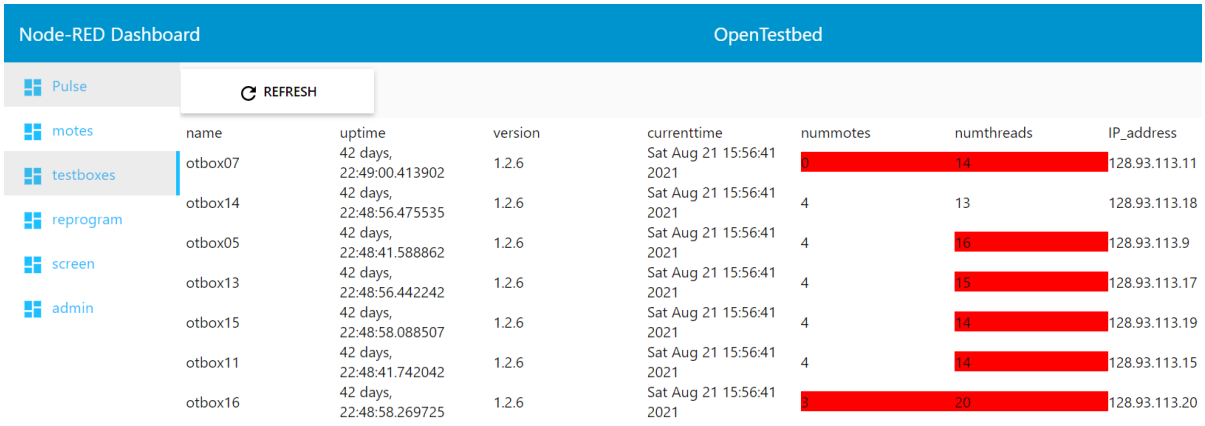
4.1. Eksperimentalna provjera filtriranja paketa

Eksperimentalna provjera algoritma filtracije je izvršena na testbedu Inria-Paris. OpenTestbed se sastoji od 80 uređaja, postavljenih u sklopu 20 OtBox-ova širom institucije. OtBox-ovi se nalaze u kancelarijama, salama za sastanke i glavnom hodniku [40]. Radi lakše upotrebe OpenTestbed-a, razvijen je *OpenTestBed dashboard* [47]. *Dashboard*, slika 4.1, ima ulogu kontrolne table, koja se konektuje na MQTT broker i koja omogućava korisniku interakciju putem API-ja klikom na web interfejsu. Kontrolna table nije testbed server i nije neophodna za pokretanje OpenTestbed-a.



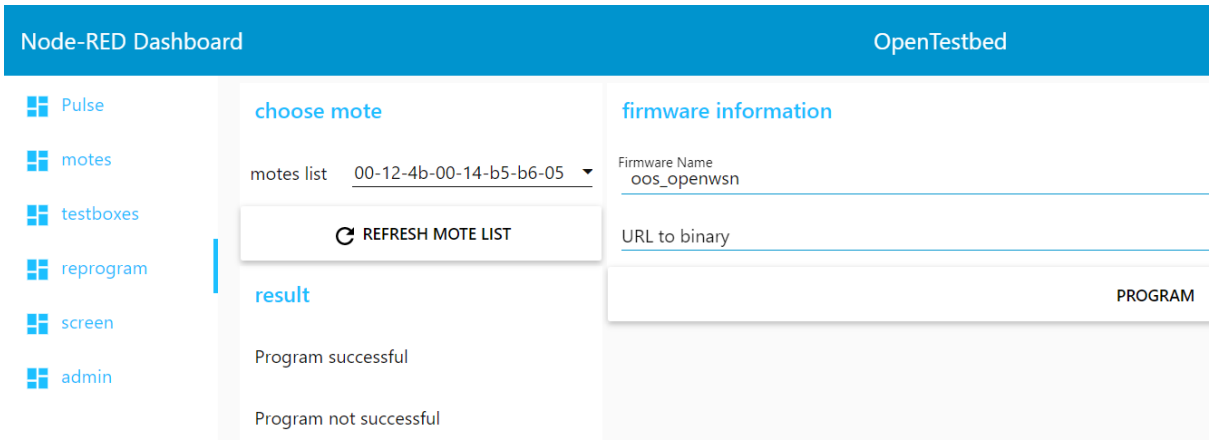
Slika 4.1 OpenTestbed dashboard

Jedinstveni identifikator uređaja je njegova IPv6 EUI64 adresa. Pomenuti web interfejs omogućava pregled OpenMote B uređaja označenih adresom i pruža informaciju koji firmver je trenutno konfigurisan. Takođe, pruža informaciju o položaju OtBox-ova, broju uređaja u OtBox-u, vremenu konfiguracije, kao i IP adresi, slika 4.2.



Slika 4.1 OpenTestBed informacije o OtBox-ovima

Osim preglednog tipa, interfejs pruža mogućnost promjene firmvera. Uređaj se može odabrati iz liste *motelist* i unijeti naziv firmvera koji je potrebno kompajlirati. Nakon izvršenja, na ekranu se prikazuje da li je naredba bila uspješna, slika 4.3. Osim promjene firmvera, moguće je resetovati sve uređaje kroz opciju '*Discover motes*'.



Slika 4.2 Upload firmvera korišćenjem dashboard-a

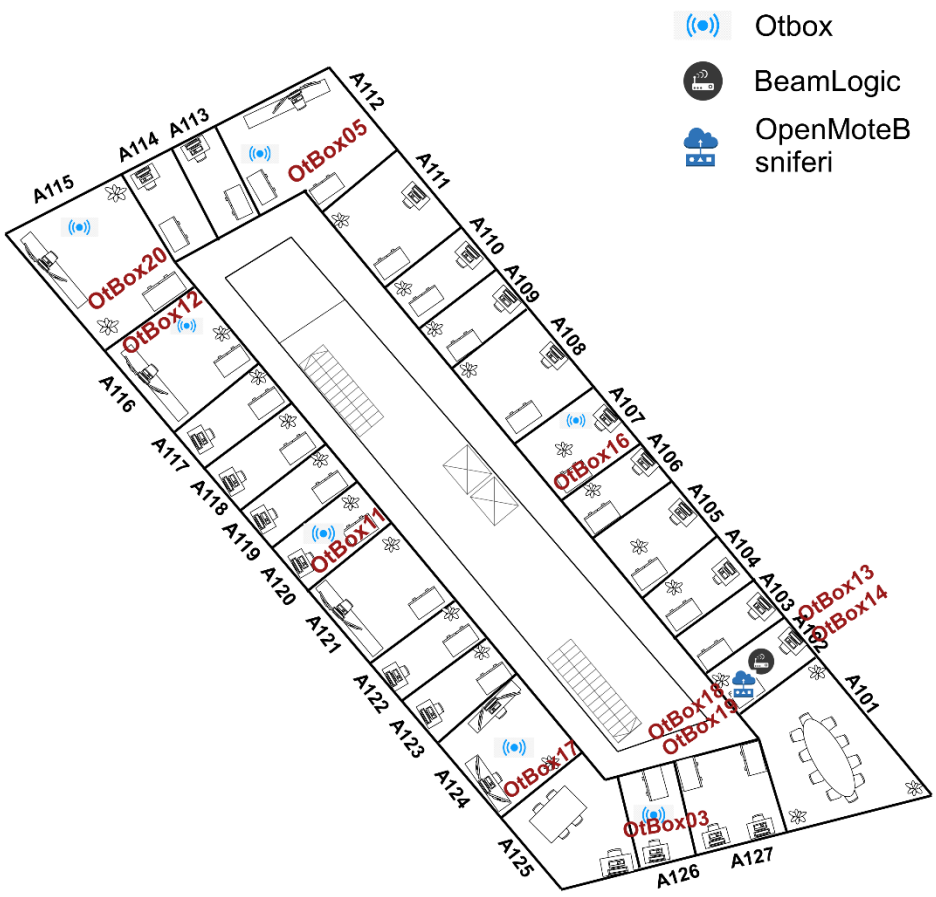
U eksperimentu je korišćen jedan 802.15.4 Site Analyzer, koji osluškuje na svih 16 kanala i 16 OpenMote B uređaja. OpenMote B uređaji se nalaze u sklopu OtBox-ova označenih brojevima 13,14,18 i 19 (slika 4.4). Uređaji su programirani da imaju ulogu snifera i svaki od njih osluškuje na različitom kanalu i na taj način predstavljaju zamjenu jednom Site Analyzer-u. Da bi jedan uređaj postao snifer, potrebno je u config.h biblioteci podesiti:

- #define BOARD_OPENSERIAL_SNIFFER 1
- #define IEEE802154E_SINGLE_CHANNEL 11

kao i u oos_snifer.h :

- #define CHANNEL 11

i izvršiti snimanje oos_sniffer firmer projekta na svim OpenMote B uređajima. Kanal može uzimati vrijednost između 11 i 26. U OpenWSN-u, firmver nudi komponentu *openserial*. Komponenta je odgovorna za prosleđivanje statusa čvorova, paketa do *root*-a ili Interneta i prijem komandi iz softvera. Uključena je kontrola veze podataka (HDLC). Status čvora uključuje identifikator čvora, rang čvora, trenutni ASN (*Apsolute Slot Number*), informaciju da li je čvor sinhronizovan, kao i detalje o rasporedu čvora, susjedima i rasporedu za prenos.



Slika 4.3 Lokacije OtBox-ova u Opentestbed-u

OpenMote B sniferi kao i Site Analyzer su postavljeni u istoj kancelariji, kako bi se rezultat snimanja mogao upoređivati (pogledaj sliku 4.4). OpenMote B uređaji pozivaju skriptu Argus Sonda, kojoj kao ulazni parametar *probetype* prosleđuju vrijednost *'opentestbed'*, dok Site Analyzer prosleđuje *'beamlogic'*. OpenMote B uređaji, kao i Site Analyzer, objavljuju podatke na MQTT brokeru sa temom *'inria-paris/beamlogic*'. Prilikom osluškivanja paketa Site Analyzer-u je pridružen snifer identifikator *(SnifferID)* '01', dok je grupi snifera dodijeljen '02'. Ta informacija predstavlja jedan od parametara za filtraciju paketa na prijemu.

Snimanje na svim OpenMote B sniferima u isto vrijeme je omogućeno pokretanjem uz pomoć skripte. Skripta na klijentskom računaru poziva proceduru filtracije duplikata. Snimak saobraćaja se sastoji od 20000 paketa. U prvom slučaju, veličina bafera je 200 paketa. Ukratko, bafer predstavlja memorijski prostor u kojem se smještaju novi, pristigli paketi. Pristigli paket se poredi sa sadržajem iz bafera i ako isti paket ne postoji u baferu (po kriterijumima provjere), dodaje se i prosleđuje klijentu. Ako je sadržaj u baferu popunjen, uklanja se najstariji paket u baferu. Stoga, veličinu bafera je potrebno prilagoditi mreži, broju uređaja, kao i brzini slanja

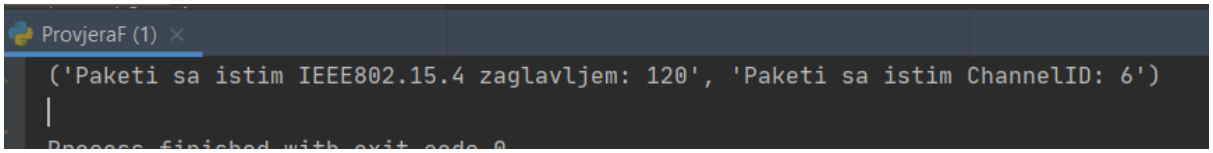
paketa. Ako je bafer malog kapaciteta, neki višestruki paketi se neće moći isfiltrirati i korisnik na svom prikazu neće imati jedinstveni saobraćaj. Takođe, povećanjem bafera se povećava i vrijeme čekanja uslijed obrade.

U eksperimentalnoj postavci napravljena su tri snimka saobraćaja. Jedan sa veličinom bafera 200 paketa, drugi sa 500, a treći sa veličinom od 50 paketa. Rezultati dobijeni u Wireshark-u su snimljeni na lokalnom računaru i analizirani. Vršena je provjera analogna algoritmu filtracije, na način da su traženi paketi koji imaju isto IEEE802.15.4 zaglavlje, koji su razmijenjeni na istom kanalu i čiji sadržaj su uhvatili različiti sniferi. U postavci sa veličinom bafera 200, od 20000 paketa koji su snimljeni, samo jedan duplikat nije isfiltriran u predloženom algoritmu. Na slici 4.5 se može vidjeti rezultat provjere snimka saobraćaja. Na prikazu se može uočiti da je IEEE 802.15.4 zaglavlje paketa jedan i dva isto, da su razmijenjeni na kanalu 16 i da su ih snimili dva različita snifera. Na osnovu tih informacija se može zaključiti da je paket duplikat i da ga skripta nije filtrirala. Na osnovu njihove pozicije u logu koja je prikazana (10724, 10927) se može vidjeti da je udaljenost između njih 203 i da bafer koji je veličine 200 paketa nije bio dovoljan. Odnosno, došlo je do brisanja prvog paketa iz bafera, prije nego što je njegova kopija uhvaćena drugim sniferom, stigla na obradu. Na samom dnu se može vidjeti da se u logu nalazi 71 par paketa sa istim IEEE 802.15.4 zaglavljem, od kojih su samo dva razmijenjena na istom kanalu, dok je jedan snimljen od strane različitih snifera. Isto IEEE 802.15.4 zaglavlje se može pojaviti na primjer, tokom podešavanja mreže, *broadcasting*-a, kao i specijalnog testiranja.

```
IEEE802.15.4 zaglavlje prvog paketa: 08 AB CD 04 FF FF 1F 00 69 01 C3 04 80 01 01 34 06 5D 63 47 0D
Channel ID prvog paketa: 16
Sniffer ID prvog paketa: 1
IEEE802.15.4 zaglavlje drugog paketa: 08 AB CD 04 FF FF 1F 00 69 01 C3 04 80 01 01 34 06 5D 63 47 0
Channel ID drugog paketa: 16
Sniffer ID prvog paketa: 2
('Pozicije paketa u logu: 10724', 10927)
('Paketi sa istim IEEE802.15.4 zaglavljem: 71', 'Paketi sa istim ChannelID: 2')
```

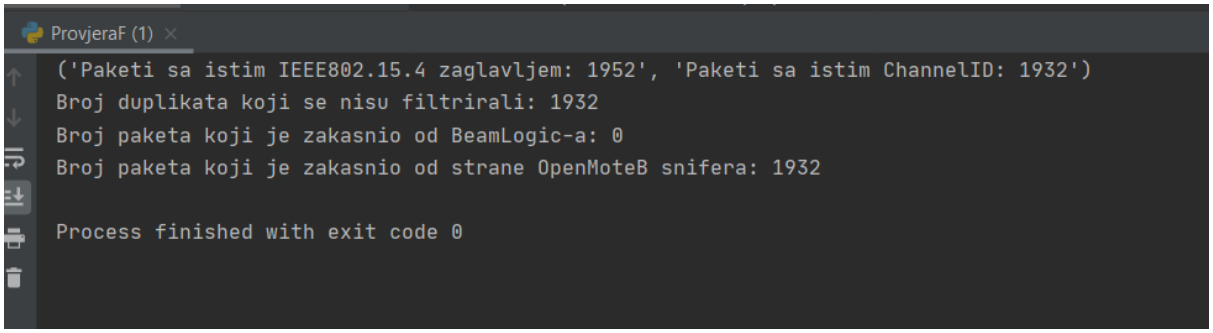
Slika 4.5 Rezultat provjere rezultata snimka saobraćaja sa baferom veličine 200 paketa

Sledeće snimanje saobraćaja je izvršeno sa veličinom bafera 500 paketa. U provjeri nije bilo duplikata poruka (slika 4.6). Prilikom analize je uočeno da postoji 120 parova paketa koji imaju isto IEEE802.15.4 zaglavlje, od kojih su 6 parova razmijenjeni na istom kanalu. Pošto nema duplikata, snimili su ih isti sniferi.



Slika 4.6 Rezultat provjere rezultata snimka saobraćaja sa baferom veličine 500 paketa

Naredni dobijeni rezultati su zanimljivi, jer prikazuju kolika je važnost dobro izabrane veličine bafera. U eksperimentu čiji su rezultati prikazani na slici 4.7, veličina bafera je 50 paketa. Ta veličina bafera je mala za jednu dinamičnu mrežu kakva je OpenTestbed, pa zbog toga veliki broj paketa nije filtriran. Preciznije, 1952 para paketa imaju isto IEEE802.15.4 zaglavlje, od kojih su 1932 razmijenjeni na istom kanalu i detektovali su ih različiti sniferi.



Slika 4.7 Rezultat provjere rezultata snimka saobraćaja sa baferom veličine 50 paketa

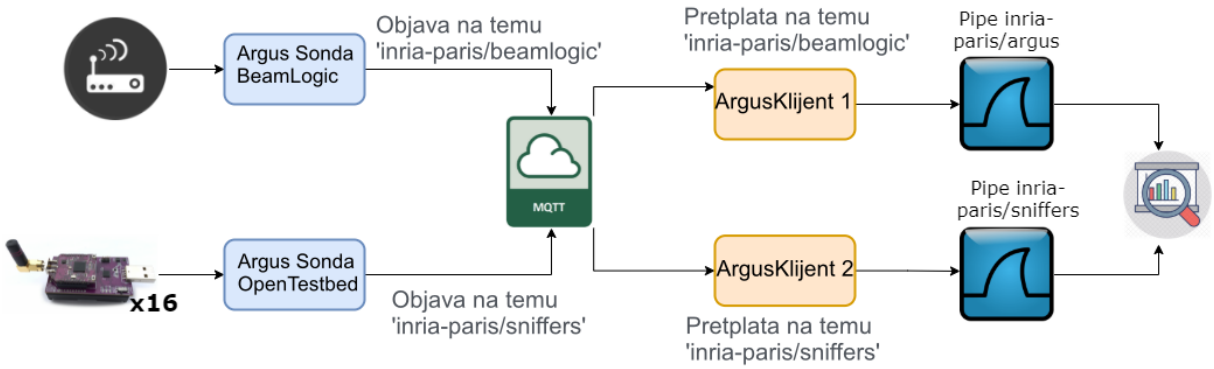
Analizom duplikata paketa uočeno je da paketi koji su detektovani od strane OpenMote B uređaja dolaze sa zakašnjenjem u odnosu na iste pakete detektovane od strane 802.15.4 Site Analyzer-a. Pretpostavlja se da je uzrok tome kašnjenje na serijskom portu uređaja, kao i kašnjenje uslijed prenosa preko MQTT brokera.

4.2. Verifikacija rada distribuiranog snifera

Proširenjem programa Argus Sonda omogućeno je OpenMote B uređajima da snimljene pakete prosleđuju zainteresovanim klijentima. OpenMote B i 802.15.4 Site Analyzer uređaji imaju različite osjetljivosti prijemnika, odnosno prag minimalne snage signala koju prijemnik može detektovati. Osjetljivost Site Analyzer-a je -101 dBm, dok je osjetljivost OpenMote B uređaja u OpenTestBed-u jednaka -123 dBm. Kako bi se provjerila funkcionalnost OpenMote B snifera, izvršena je sledeća eksperimentalna provjera.

Kao što je prikazano na slici 4.7 eksperimentalna postavka se sastoji od dva paralelna procesa osluškivanja saobraćaja u OpenTestBed-u. U procesu učestvuju Site Analyzer,

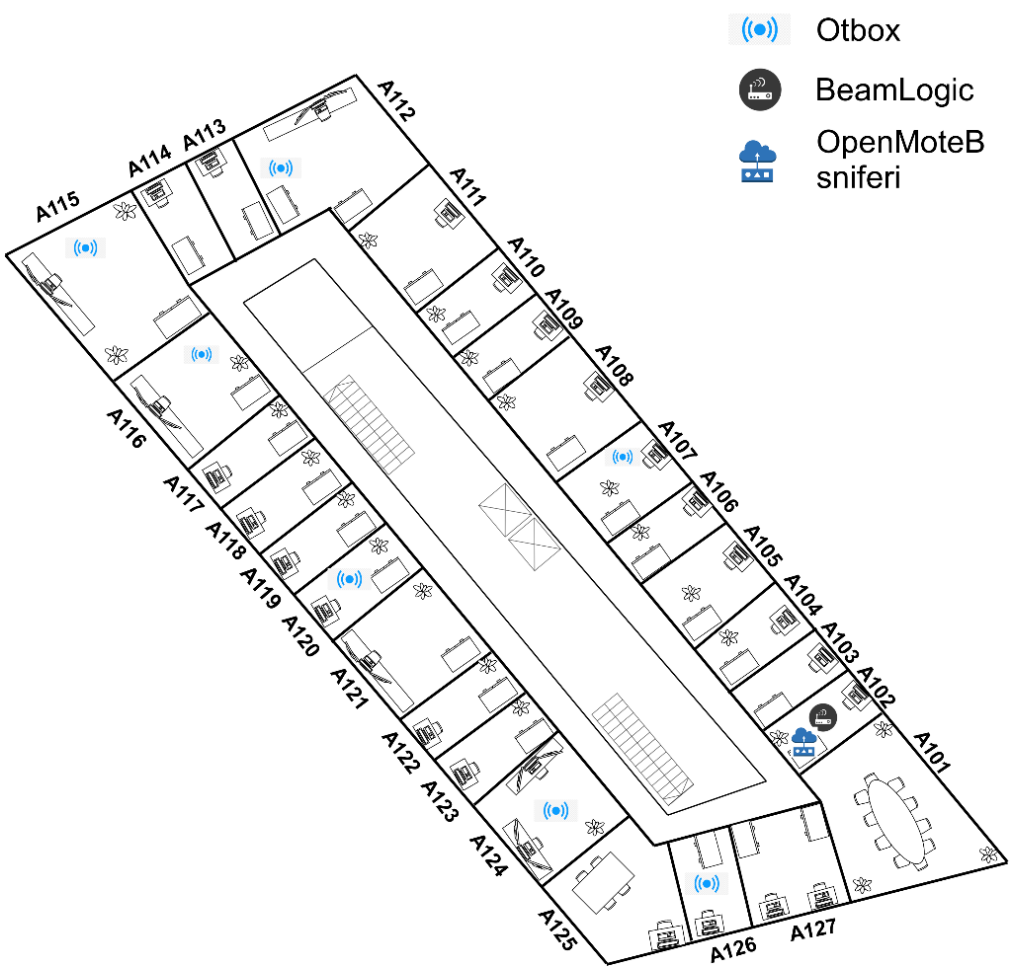
simultano na svih 16 kanala, kao i 16 OpenMote B uređaja, koji su konfigurisani da osluškuju na 16 različitih kanala. Na računaru koji je povezan sa Site Analyzer-om se pokreće Argus Sonda skripta, kojoj se kao ulazni parametar *probetype* prosleđuje vrijednost *'beamlogic'*, dok se na računaru koji vrši testiranje poziva ista skripta gdje je ovaj parametar podešen na vrijednost *'opentestbed'*. Kako bi se na prijemu razdvojili paketi koje je uhvatio Site Analyzer i paketi koje su uhvatili OpenMote B uređaji, kreirane su dvije MQTT teme. Paketi sa Site Analyzer-a se prosleđuju na MQTT broker sa temom: *'inria-paris/beamlogic'*, a drugi preko *'inria-paris/sniffers'*. Nakon toga, potrebno je otvoriti dva klijentska procesa koja se pretplaćuju na dvije pomenute teme na MQTT brokeru. U cilju pokretanja dva Wireshark procesa na jednom računaru, moraju se kreirati dva PIPE-a, u ovom slučaju Argus Klijent1 kreira PIPE *'inria-paris/argus'* dok Argus Klijent2 *'inria-paris/sniffers'*.



Slika 4.8 Eksperimentalna postavka

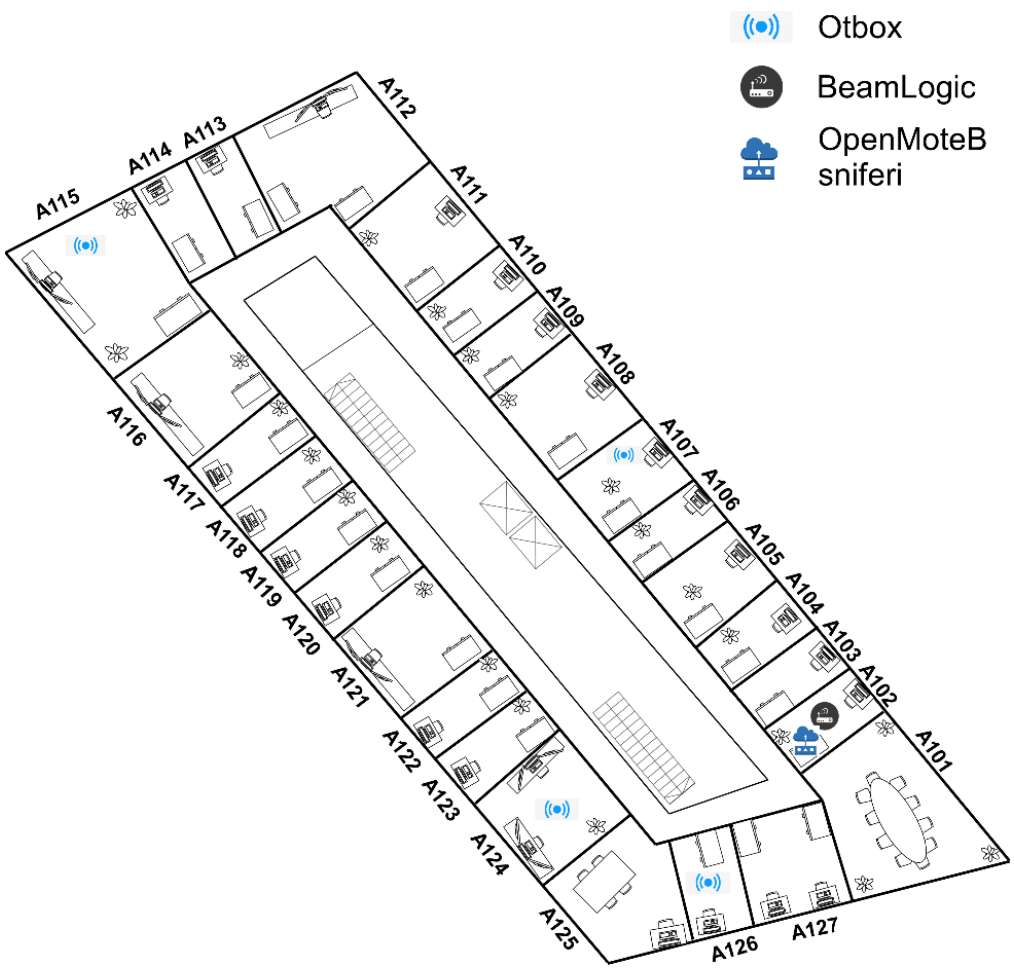
Kao što se može vidjeti na slici 4.8, OpenMote B uređaji i Site Analyzer, nezavisno jedni od drugih snimaju saobraćaj. Stanje u mreži se može pratiti u realnom vremenu na Wireshark-u, koji se otvara nakon pokretanja klijentske skripte. Nakon prikazivanja paketa u Wireshark-u i hvatanja dovoljnog broja paketa za analizu, snimanje se zaustavlja i uhvaćeni saobraćaj se čuva u lokalnoj memoriji. Dva kreirana fajla u sledećem koraku se analiziraju i upoređuju.

Na slici 4.9 su predstavljeni raspoloživi OtBox-ovi u OpenTestbed-u. OtBox-ovi su postavljeni u zgradama A i C. Za eksperimentalnu verifikaciju korišćena je zgrada A, kao i OtBox-ovi koji se u njoj nalaze (otbox03, otbox05, otbox11, otbox12, otbox13, otbox14, otbox16,otbox17, otbox18, otbox19 i otbox20). Četiri OtBox-a sa 16 uređaja su korišćeni kao sniferi, dok ostalih 7 OtBox-ova sa 27 uređaja razmjenjuju pakete koje je potrebno detektovati.



Slika 4.9 Verifikacija rada distribuiranog snifera – eksperiment 1

Ideja za provjeru rada OpenMote B uređaja je sledeća. Potrebno je izvršiti snimanje saobraćaja sa različitim brojem aktivnih uređaja, a ukoliko OpenMote B ispravno rade, onda se očekuje da će u svim scenarijima biti približno jednako poklapanje uhvaćenog mrežnog saobraćaja. Prvo snimanje saobraćaja uključuje sve aktivne uređaje u mreži (slika 4.9), dok sledeći snimak predstavlja rad sa polovinom broja dostupnih uređaja (slika 4.10). Aktivni uređaji koriste *oos_openwsn*, dok neaktivni uređaji *bsp_eui64* projekat, koji predstavlja dio OpenWSN firmvera. *Oos_openwsn* projekt koristi 6TiSCH grupu protokola, dok *bsp_eui64* se koristi za neaktivne uređaje, jer ne vrši slanje paketa već omogućava samo treperenje lampica na hardveru.

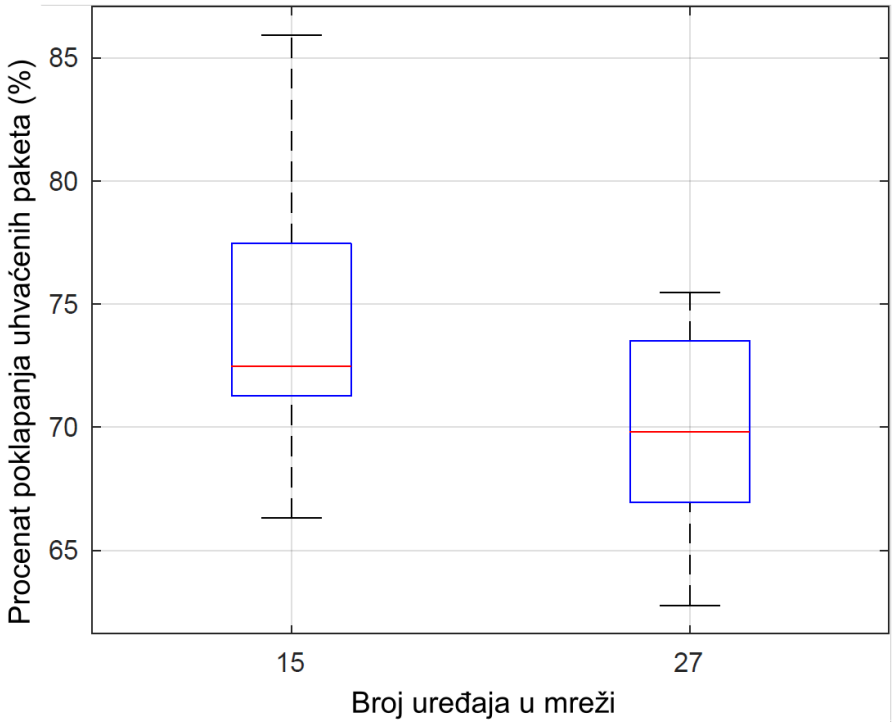


Slika 4.10 Verifikacija rada distribuiranog snifera – eksperiment 2

Sniferi u oba eksperimenta se nalaze na istoj lokaciji, u kancelariji A102. Postavljeni su na približno istom mjestu, sa ciljem da dijele što je moguće sličnije radio okruženje i da se nakon snimanja njihov uhvaćeni saobraćaj može porediti. Oba eksperimenta su ponovljena po deset puta. Nakon snimanja saobraćaja, izvršena je analiza. Analizom su detektovani isti uhvaćeni paketi. Pod pojmom isti paketi se podrazumijevaju paketi koji imaju isto IEEE 802.15.4 zaglavlje, kao i kanal na kojem su razmijenjeni.

Na slici 4.11 prikazan je rezultat poređenja paketa uhvaćenih od strane Site Analyzer-a i OpenMote B uređaja. Sa 15 aktivnih senzorskih uređaja, detektovano je poklapanje mrežnog saobraćaja do 86%, dok u slučaju kada je 27 senzorskih uređaja bilo aktivno, poklapanje snimljenog saobraćaja između ova 2 tipa snifera je iznosilo do 76%. Procenat poklapanja uhvaćenih paketa je izračunat na sledeći način: broj istih paketa koji su uhvatili sniferi je podijeljen sa brojem paketa koje je uhvatio Site Analyzer * 100%. Rezultat prikazuje manje

poklapanje saobraćaja u eksperimentu sa 27 senzorskih uređaja, u odnosu na eksperiment sa 15 uređaja. Razlog tome je manja opterećenost MQTT brokera u scenariju sa 15 aktivnih senzorskih uređaja, kao i manja vjerovatnoća da će doći do interferencije prilikom korišćenja manjeg broja senzora.



Slika 4.11 Rezultati poređenja uhvaćenog saobraćaja od strane 802.15.4 Site Analyzer-a i OpenMote B snifera

Tokom izvršavanja pomenutog eksperimenta, u prostorijama gdje je postavljen OpenTestbed je takođe vršeno testiranje i SmartMeshIP mreže. Pošto je u pitanju saobraćaj koji na nivou linka koristi IEEE 802.15.4 standard, sniferi su uhvatili pakete koji pripadaju i ovoj mreži. Rezultati koji su prikazani na slici 4.11 su filtrirani, tako da uključuju samo mrežni saobraćaj koji su generisali senzorski uređaji iz gore navedenih OtBox-ova. Međutim, pošto se filtriranje trenutno vrši na klijentskoj strani, svi detektovani paketi se prenose preko standardne Argus putanje: od uređaja do brokera, pa nakon toga od brokera do klijenata. Samim tim, pomenuti saobraćaj SmartMeshIP mreža dodatno opterećuje MQTT Broker, što se može potvrditi pojavom *overflow* poruke koja ukazuje da broker ne može obraditi sve pristigle pakete. Takođe, pretpostavlja se da SmartMeshIP povećava vjerovatnoću pojave interferencije, koja može uzrokovati da saobraćaj koji potiče od pojedinih udaljenih senzora, ne može biti

ispravno dekodiran. Neki od narednih koraka bi mogao da uključi filtriranje paketa prije slanja na broker, ili upotrebu dva brokera kako bi se smanjila pojava 'uskog grla'.

Kao što je ranije pomenuto, konfiguracija OpenMote B snifera podrazumijeva slanje podataka na MQTT broker dva puta. Paketi se prvo preuzimaju sa brokera u Argus Sonda skripti, enkapsuliraju, a zatim šalju preko brokera svim zainteresovanim klijentima. Pošto paketi 'putuju' dva puta duže nego u konfiguraciji sa Site Analyzer-om, vjerovatnoća gubitka paketa je veća.

U svakom od ovih scenarija, prikazuju se agregirani podaci za sve OtBox-ove u scenariju, a tokom agregacije se gube informacije koliko koji uređaj doprinosi ukupnom rezultatu. Sledeći korak u istraživanju bi bio razdvajanje rezultata, odnosno uhvaćenog saobraćaja, za svaki uređaj pojedinačno, uz istu postavku snifera. Ideja je da se ima uvid u broj paketa koje će senzori poslati i na kojem kanalu. Sa tim informacijama, moguće je izračunati koliko će tačno i koliko uspješno sniferi uhvatiti pakete.

Glava 5

Zaključak

Sniferi igraju značajnu ulogu u razvojnoj fazi bežičnih senzorskih mreža. Pomoću snifera se mogu analizirati razmijenjeni paketi, detektovati problemi i smetnje u mreži, analizirati performanse novih protokola i standarda, i to sve prije upotrebe u stvarnom okruženju. Sniferi ne utiču na rad mreže, tako da se mogu upotrebljavati i u komercijalnim aplikacijama za potrebe monitoringa i statistike. Sniferska rješenja kao što je Argus, koja omogućavaju udaljeni pristup snimljenom mrežnom saobraćaju, su od velikog značaja, jer olakšavaju upotrebu snifera koji je u uobičajenoj konfiguraciji bio ograničen na lokaciju posmatrane mreže. Osim mogućnosti praćenja mreže sa udaljene lokacije, Argus rješenje omogućava distribuciju uhvaćenog saobraćaja prema više korisnika.

U ovom radu, realizovan je *open source* distribuirani snifer, koji omogućava analizu saobraćaja u velikim bežičnim senzorskim mrežama, koje se mogu protezati na više spratova u *indoor* okruženjima. Upotrebom više snifera koji međusobno ne sarađuju, očekivano je da se na prijemu mogu naći kopije istih paketa. Stoga, u radu je predstavljen algoritam koji u mreži, baziranoj na IEEE 802.15.4 standardu, detektuje i uklanja višestruko snimljene pakete. Filtracija se odvija u četiri koraka i uključuje provjeru IEEE 802.15.4 zaglavlja, kanal na kojem je razmijenjen paket, identifikator snifera koji je uhvatio paket kao i vrijeme detekcije. Kao rezultat filtracije, korisnik ima jedinstveni prikaz saobraćaja u mreži. Prije instalacije više snifera u mreži, neophodno je utvrditi njihove lokacije, tako da sniferi mogu pokriti čitavu mrežu koja se posmatra. Lokacije snifera se mogu utvrditi intuitivno, na osnovu pozicije senzora u mreži, ili uz pomoć specijalizovanog softvera koji se naziva Mercator, koji omogućava uvid u kvalitet linkova na svim kanalima [48]. Takođe, predloženo rješenje za realizaciju višestrukih snifera u velikim senzorskim mrežama je uvezano sa postojećim Argus softverom, čime je omogućen monitoring mrežnog saobraćaja na daljinu. U radu je realizovano i proširenje Argus rješenja, koje sada osim rada sa Site Analyzer-om, može raditi sa lokalno priključenim sniferima, kao i sa OpenTestbed-om. Dopunom Argus rješenja, informacije dobijene iz testne mreže se šalju zainteresovanim klijentima na potpuno isti način kao u

originalnom Argus rješenju. Ova funkcionalnost se može iskoristiti u manjim i jeftinijim testbedovima, ili za potrebe istraživačkog rada.

Predloženo rješenje je verifikovano simulacionim kao i eksperimentalnim putem. U eksperimentalnoj provjeri algoritma filtriranja, uočeno je da je veoma bitno prilagoditi veličinu bafera mreži. Veličina bafera zavisi od dinamike generisanja saobraćaja u mreži, kao i od broja aktivnih uređaja u mreži. U slučaju da je bafer nedovoljne veličine za posmatranu mrežu, algoritam filtracije nije u mogućnosti da uhvati sve duplikate paketa. Kao rezultat, klijent koji radi analizu mreže nema jedinstven prikaz mrežnog saobraćaja. Sa druge strane, bafer velike veličine bafera će omogućiti da se u snimku saobraćaja ne pojave duplikati paketa, ali će znatno usporiti rad Argus softvera. Stoga, potrebno je pronaći optimalnu veličinu bafera za svaku mrežu.

U vremenu ekspanzije primjene bežičnih senzorskih mreža, koje su dovele do koncepta Interneta stvari (IoT), poseban značaj zauzimaju industrijske IoT mreže. 6TiSCH grupa protokola, kao nadogradnja na IEEE 802.15.4 standard, upravo omogućava nivo pouzdanosti WSN potreban za implementaciju u IIoT mrežama. Sa softverskim alatom otvorenog koda razvijenim u okviru sprovedenog istraživanja, biće znatno olakšan monitoring rada velikih IIoT mreža, u kojima je potrebno primijeniti više snifera, a omogućen je pristup snimljenom saobraćaju proizvoljnom broju korisnika, na daljinu. Razvijeno distribuirano snifersko rješenje će biti nezamjenljiv alat i pri razvoju novih protokola za WSN, novih standarda, kao i za monitoring svih performansi od značaja za rad IIoT mreža.

Literatura

[1] Culler D, Estrin D, Srivastava M, „Overview of sensor networks,“ *Computer 37(8),* p. 41–49, 2004.

[2] V. R, „Wireless sensor networks,“ *Proceedings of the 5th European conference,* 2008.

[3] Kandris D, Nakas C, Vomvas D, Koulouras, „G. Applications of Wireless Sensor Networks: An Up-to-Date Survey,“ *Applied System Innovation,* 2020.

[4] Warneke, B.; Last, M.; Liebowitz, B.; Pister, K.S. , „Smart dust: Communicating with a cubic-millimeter,“ *Computer,* t. 34, br. 44-51, 2001.

[5] S. Junnila et al., „Wireless, Multipurpose In-Home Health Monitoring Platform: Two Case Trials,“ *IEEE Transactions on Information Technology in Biomedicine,* t. 2, pp. 447-455, 2010.

[6] O. Elijah, T. A. Rahman, I. Orikumhi, C. Y. Leow and M. N. Hindia, „An Overview of Internet of Things (IoT) and Data Analytics in Agriculture: Benefits and Challenges,“ *IEEE Internet of Things Journal,* t. 5, br. 5, pp. 3758-3773, 2018.

[7] D. Dujovne, T. Watteyne, X. Vilajosana and P. Thubert, „6TISCH: Deterministic IP-enabled industrial Internet (of things),“ *IEEE Communications Magazine,,* t. 52, br. 12, pp. 36-41, 2014.

[8] Zantalis, F.; Koulouras, G.; Karabetsos, S.; Kandris, D., „ A review of machine learning and IoT in smart transportation,“ *Future Internet ,* t. 4, br. 94, 2019.

[9] Dong-Seong Kim, Hoa Tran-Dang, An Overview on Wireless Sensor Networks. In: Industrial Sensors and Controls in Communication Networks. Computer Communications and Networks., Springer, Cham, 2019.

[10] „Argus (Software),“ INRIA, [Na mreži]. Available: https://raweb.inria.fr/rapportsactivite/RA2016/eva/uid74.html. [Poslednji pristup 25 July 2021].

[11] „OpenWSN,“ 25 July 2021. [Na mreži]. Available: https://openwsn.atlassian.net/wiki/spaces/OW/overview.

[12] X. Vilajosana, P. Tuset, T. Watteyne, and K. Pister, „OpenMote: OpenSource Prototyping Platform for the Industrial IoT,“ u *International Conference on Ad Hoc Networks (AdHocNets),* San Remo, Italy, 2015.

[13] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele and T. Watteyne, „Fit iot-lab: A large scale open experimental iot testbed," *2015 IEEE 2nd World Forum on Internet of Things (W-iF-IoT),* p. 459–464, 2015.

[14] „MQTT: The Standard for IoT Messaging," [Na mreži]. Available: https://mqtt.org/. [Poslednji pristup August 2021].

[15] OASIS, „OASIS Open," [Na mreži]. Available: https://www.oasis-open.org/org/. [Poslednji pristup August 2021].

[16] Shaoqiang Wang, DongSheng Xu and ShiLiang Yan, „Analysis and application of Wireshark in TCP/IP protocol teaching," *2010 International Conference on E-Health Networking Digital Ecosystems and Technologies (EDT),* pp. 269-272, 2010.

[17] IEEE, „802.15.4e-2012 - IEEE Standard for Local and metropolitan area," 2012.

[18] BeamLogic, „THE 802.15.4 SITE ANALYZER," [Na mreži]. Available: http://www.beamlogic.com/802-15-4-siteanalyzer. [Poslednji pristup 25 July 2021].

[19] „IEEE 802.15.4 Standard: a tutorial / primer," [Na mreži]. Available: https://www.electronics-notes.com/articles/connectivity/ieee-802-15-4-wireless/basics-tutorial-primer.php. [Poslednji pristup October 2021].

[20] M. R. P. e. al., „Standardized Protocol Stack for the Internet of (Important) Things," *IEEE Communications Surveys & Tutorials,* t. 15, br. 3, pp. 1389-1406, 2013.

[21] T. I. W. Technology. [Na mreži]. Available: https://www.electronics-notes.com/articles/connectivity/ieee-802-15-4-wireless/thread-wireless-connectivity.php. [Poslednji pristup October 2021].

[22] O. Jonas, „6LoWPAN demystified," October 2014. [Na mreži].

[23] „Duty cycle," [Na mreži]. Available: https://www.thethingsnetwork.org/docs/lorawan/duty-cycle/. [Poslednji pristup August 2021].

[24] Palattella, Maria & Thubert, Pascal & Vilajosana, Xavier & Watteyne, Thomas & Wang, Qin & Engel, Thomas, „6TiSCH Wireless Industrial Networks: Determinism Meets IPv6," *Smart Sensors, Measurement and Instrumentation,* t. 9, 2014.

[25] P. Thubert, „An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4,“ 2019. [Na mreži]. Available: https://tools.ietf.org/id/draft-ietf-6tisch-architecture-15.html.

[26] J. Melorose, R. Perroy, and S. Careas, „Compression format for IPv6 datagrams in low power and lossy networks,“ *Statew. Agric. L. Use Baseline 2015,* t. 1, pp. 1-25, 2015.

[27] T. Watteyne et al, „OpenWSN: A standards-based low-power wireless development environment,“ *Eur. Trans. Telecommun.,* t. 23, br. 5, p. 480–493, 2012.

[28] T. Winter et al., „RPL: IPv6 routing protocol for low power and lossy networks,“ July 2011. [Na mreži]. Available: http//tools. ietf. org/html/draft-ietf-roll-rpl-19.

[29] D. Airehrour, J. A. Gutierrez, and S. K. Ray, „A Trust-Aware RPL routing protocol to detect blackhole and selective forwarding attacks,“ *Aust. J. Telecommun. Digit. Econ,* t. 5, br. 1, p. 50, 2017.

[30] Z. Shelby, K. Hartke, C. Bormann, B. Frank, „Constrained application protocol (CoAP),“ 2014. [Na mreži]. Available: https://datatracker.ietf.org/doc/html/rfc7252.

[31] „6TiSCH Operation Sublayer Protocol (6P),“ u *[draft-ietf-6tisch-6top-protocol-12]*, 2018.

[32] X. Vilajosana, K. Pister, and T. Watteyne, „Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration,“ u *Internet Engineering Task Force Std. RFC8180*, May, 2017.

[33] M. Vucini ˇ c, J. Simon, K. Pister, and M. Richardson, „Minimal Security Framework for 6TiSCH,“ *Internet Engineering Task Force Std,* November 2018.

[34] [Na mreži]. Available: https://www.wireshark.org/. [Poslednji pristup 25 July 2021].

[35] X. Kuang and J. Shen, „SNDS: A Distributed Monitoring and Protocol Analysis System for Wireless Sensor Network,“ *2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing,* pp. 422-425, 2010.

[36] Z. Zhao, W. Huangfu and L. Sun, „NSSN: A network monitoring and packet sniffing tool for wireless sensor networks,“ *2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC),* pp. 537-542, 2012.

[37] Garcia FP, Andrade RM, Oliveira CT, de Souza JN., „EPMOSt: an energy-efficient passive monitoring system for wireless sensor networks,“ *Sensors (Basel),* 2014.

[38] J. Crnogorac, J. Kovač, E. Kočan i M. Vučinić, „d-Argus: a Distributed IEEE 802.15.4 Sniffer," *2019 27th Telecommunications Forum (TELFOR), 2019,* pp. 1-4, 2019.

[39] „Birthday_problem," [Na mreži]. Available: https://en.wikipedia.org/wiki/Birthday_problem. [Poslednji pristup October 2021].

[40] J. Muñoz et al., „OpenTestBed: Poor Man's IoT Testbed," *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS),* pp. 467-471, 2019.

[41] „OpenWSN-fw," OpenWSN, [Na mreži]. Available: https://github.com/openwsn-berkeley/openwsn-fw. [Poslednji pristup August 2021].

[42] OpenWSN, „OpenSim emulator," 2021. [Na mreži]. Available: https://openwsn.atlassian.net.

[43] J. Kovac, „JelenaKovacc/argus," 2021. [Na mreži]. Available: https://github.com/JelenaKovacc/argus/tree/develop_o.

[44] „HDLC - High-level Data Link Control," [Na mreži]. Available: https://www.tutorialspoint.com/high-level-data-link-control-hdlc. [Poslednji pristup September 2021].

[45] OpenTestbed, August 2021. [Na mreži]. Available: https://github.com/openwsn-berkeley/opentestbed/blob/master/otbox.py.

[46] „JelenaKovacc/Argus/ArgusProbe," [Na mreži]. Available: https://github.com/JelenaKovacc/argus/blob/develop_o/ArgusProbe_Beamlogic.py.

[47] OpenWSN, „OpenTestbed dashboard," August 2021. [Na mreži]. Available: https://openwsn-dashboard.eu-gb.mybluemix.net/ui/#!/0.

[48] OpenWSN, „Mercator: Dense Wireless Connectivity Datasets for the IoT," [Na mreži]. Available: https://github.com/openwsn-berkeley/mercator/wiki. [Poslednji pristup August 2021].

Lista skraćenica

6LoWPAN - IPv6 over Low -Power Wireless Personal Area Networks

6TiSCH - The IETF IPv6 over the TSCH mode of IEEE802.15.4e

ACK - Acknowledgement

API - Application programming interface

ASN - Absolute Slot Number

CDU - Channel distribution/usage

CoAP - Constrained Application Protocol

CRC - Cyclic redundancy check

DAG - Direct Acyclic Graf

DODAG - Direction-Oriented Directed Acyclic Graph

DSSS - Direct-sequence spread spectrum

EB - Enhanced Beacons

EPMOSt - Energy-efficient Passive MOnitoring System

FCS - Frame check sequence

FSK - Frequency-shift keying

FTDI - Future Technology Devices International Limited

GND - Ground

HDLC - High-Level Data Link Control

HTTP - Hypertext Transfer Protocol

IEEE - Institute of Electrical and Electronics Engineers

IETF - Internet Engineering Task Force

IIoT - Industrial Internet of Things

INRIA - National Institute for Research in Computer Science and Automation

IPv6 - Internet Protocol v6

IoT - Internet of Things

ISO - International Organization for Standardization

LAN - Local area network

LBR - LLN Border Routers

LBR - Low-Power Border Router

LLN - A Low-power and Lossy Network Architecture

M2M - Machine to Machine

MAC – Medium Access Control

MIB - Management Information Base

MQTT - Message Queuing Telemetry Transport

MR-OFDM - Multi-rate and multi-regional orthogonal frequency division multiplexing

MSF - Minimal Scheduling Function

NSSN - Network monitoring and packet Sniffing tool for wireless Sensor Networks

NTP - Network Time Protocol

O-QPSK - Offset quadrature phase-shift keying

OFDM - Orthogonal frequency-division multiplexing

OQPSK - Offset quadrature phase-shift keying

QR code - Quick Response code

QoS - Quality of Service

RESTful - Representational State Transfer

RF – Radio Frequency

ROLL - Routing Over Low power and Lossy networks

RPL - Routing Protocol for Low-Power and Lossy Networks

RSSI - Received Signal Strength Indicator

SNDS - Sensor Network Distributed Sniffer

SNMP - Simple Network Management Protocol

SPI - Serial Peripheral Interface

TCP - Transmission Control Protocol

TSCH - Time Slotted Channel Hopping

UART - A universal asynchronous receiver-transmitter

UWB - Ultra wideband

UDP - User Datagram Protocol

UOC - Open University of Catalonia

USB - Universal Serial Bus

VCC - Voltage Common Collector

WG - Working Group

WPAN - Wireless Personal Area Nework

WSN - Wireless Sensor Network

ZEP - ZigBee Encapsulation Protocol