Komunikacija među vozilima unutar VANET mreže

Kovačević, Mario

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

KOMUNIKACIJA MEĐU VOZILIMA UNUTAR VANET MREŽE

Diplomski rad

Mario Kovačević

Osijek, 2020.

SADRŽAJ

1. UVOD

Razvojem 5G mreže očekuje se veliki napredak u znanosti i tehnologiji. Jedno od bitnijih ostvaranja je svakako autonomna vožnja. Da bi se autonomna vozila mogla pustiti u stvarni svijet, moraju biti sigurna i udobna za vožnju. Za potrebno je izvršiti brojna testiranja i provjere. VANET mreža (Vehicular Ad Hoc Networks) je mreža koja omogućuje bežičnu komunikaciju i razmjenu informacija među vozilima, što je jedan od ključnih preduvjeta za daljnji razvoj i širenje autonomnih vozila.

Ad Hoc mreže su zahtjevne za osigurati učinkovitu komunikaciju zbog dinamične topologije i velikih brzina vozila u stvarnosti a kako se radi o ljudskim životima unutar tih vozila mjesta za greške nema. Dakle nije dovoljno samo prenjeti informaciju od jednog vozila do drugog nego se to mora učiniti u realnom vremenu. Da bi se nešto odradilo u realnom vremenu, potreban je cijeli skup uređaja, algoritama, arhitektura sustava i protokola da bi VANET mreža ispravno funkcionirala. Svi ti dijelovi će se opisati pojedinačno te objasniti kako međusobno komuniciraju i nadopunjuju jedni druge. Osim toga razradit će se i glavna podjela komunikacije, komunikacija među vozilima, komunikacija između vozila i infrastrukture te komunikacija između dvije infrastrukutre. Osim fizičke podjele razmjene podataka, postoje i dvije vrste razmjene podataka ovisno o generiranju poruka. VANET mreža automatski generira poruke, pogotovo one koje su bitne za sigurnost u prometu, međutim postoje i poruke koje korisnik sam može zahtjevati, kao što su odabir najkraće rute. Upravo ta razlika između automatskih poruka i poruka na zahtjev će se analizirati u drugom dijelu rada.

Nakon teorijskog dijela slijedi simulacija i analiza rezultata. Analiza se sastoji od simulacija u dva programska okruženja, CupCarbon i Veinsa. Identične simulacije će biti izvedene u oba programa. Provesti će se sustavno istraživanje oba programa, opisati rad unutar njih te napraviti usporedna analiza komunikacijskih tehnologija koje se koriste u VANET mrežama. Napravit će se komparacija između slanja poruka pomoću VANET mreže i slanja poruka pomoću običnih senzora i usmjerivača, kao i razlika između automatski generiranih poruka i poruka na zahtjev korisnika. Rezultati će se grafički prikazati kako se mogao izvesti zaključak.

2. VANET MREŽA

VANET mreža (Vehicular Ad Hoc Network) je bežična mreža decentraliziranog tipa koja je nastala od MANET mreže (Mobile Ad Hoc Network). MANET mreža je prva ad-hoc mreža koja je bila u potpunosti razvijena. Obilježja ad-hoc mreža su da nemaju stabilnu infrastrukturu, već svaki čvor unutar same mreže sudjeluje u slanju i primanju podataka.

Ad-hoc mreže mogu biti velikih razmjera što dozvoljava veliki broj korisnika i neograničenu površinu. Brzina samih korisnika, odnosno vozila može biti dosta velika, kao npr. vožnja na autocesti, međutim onda promet ne smije biti pregust, kako bi mreža mogla učinkovito funkcionirati. Što je manja brzina vozila, ad hoc mreža dozvoljava veću gustoću prometa. Što se tiče izvora napajanja, ad hoc mreže nemaju ograničenja. To je jedan od preduvjeta da računalne performanse budu visoke. Za samo kretanje vozila je već poznat uzorak kretanja zbog poznavanja cesta. Osim toga poznate su i veličine čvorova kao visina i širina čvorova.

MANET mreža je određena standardom IEEE 802.11, koji govori o mobilnosti čvorova [4]. S razvojem tehnologije i potrebom za sve većom komunikacijom, pogotovo među vozilima, nastala je VANET mreža. 2001. godine je nastala prva inačica te mreže, pod nazivom *car-to-car ad hoc mobile communication and networking*. Glavni zadatak ove mreže je omogućiti da vozila komuniciraju ili međusobno ili sa infrastrukturom na cesti.

Kako se autonomna vozila sve više razvijaju i šire, VANET mreža je ključna za komunikaciju istih. Ono što predstavlja glavni problem kod realizacije takve komunikacije je velika brzina kojom se kreću vozila. Također zahtjevi za ovu mrežu su učinkovito isporučivanje svakog paketa, velika brzina isporučivanja paketa i malo kašnjenje, odnosno latencija. To je sve bitno prvo zbog sigurnosti u prometu, no i zbog drugih stvari kao što su gužva u prometu, kvaliteta i udobnost vožnje [1].

2.1 Tipovi komunikacije VANET mreže

VANET se dijeli na tri tipa komunikacije unutar same mreže, a to su

-komunikacija između vozila (V2V)

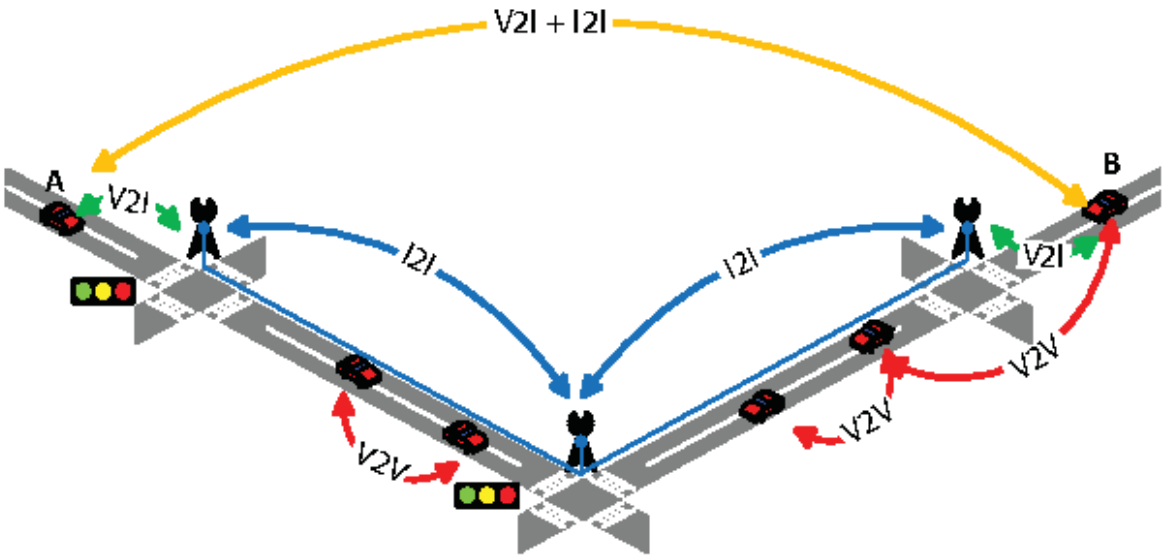
-komunikacija između vozila i infrastrukture (V2I)

-komunikacija između infrastrukture i infrastrukture (I2I)

Vehicle to vehicle (V2V) mreža omogućuje direktnu komunikaciju među vozilima. U takvoj mreži nije potreba infrastruktura i koristi se za sigurnost na cesti. Takva direktna komunikacija se naziva one hop komunikacija.

Vehicle to Infrastructure (V2I) mreža se koristi za komunikaciju vozila sa stabilnom infrastrukturom. Ovakav način komunikacije je potreban za skupljanje informacija o stanju na cesti. Ovakav tip komunikacije se naziva multihop komunikacija, jer kao što ime samo kaže, koristi više skokova, odnosno vozila i infrastrukturu. Hibridna arhitektura povezuje gore navedene vrste komunikacije V2V i V2I.

Infrastructure to Infrastructure (I2I) mreža je zadužena za prijenos podataka na udaljenije lokacije. Takav tip komunikacije isto spada u multihop komunikaciju.



Sl.1 Prikaz tipova komunikacije unutar VANET mreže

Na slici 1 je prikazan primjer realnog scenarija u prometu, gdje se raskrižju sa semaforom približava nekoliko vozila različitim brzinama. Da bi sva vozila imala pravovremene i točne informacije o stanju na cesti, za analizu je potrebno koristiti sva tri tipa komunikacije. Crvenim strelicama je prikazana V2V komunikacija, zelenim strelicama V2I komunikacija, plavim strelicama I2I komunikacija dok je žutim strelicama prikazana kombinacija dva tipa komunikacije, u ovom konkretnom primjeru V2I i I2I.

Kao što je već navedeno, glavna svrha postojanja komunikacije među vozilima je siguran i učinkovit promet. Ta dva pojma su međusobno usko povezana. Naprimjer ako se u prometu dogodi nesreća, ona vozila koju su jako blizu tog mjesta, za njih je prvenstveno to bitno zbog sigurnosti, međutim udaljena vozila će tu informaciju upotrijebiti kako bi zaobišli to mjesto i uštedili na vremenu i tako se povećava sama učinkovitost. Za takvu vrstu komunikacije, pokrivenost mreže mora biti jako velika, isto kao i sama robusnost. Zato se infrastruktura pored same ceste ima u cilju koristiti i kao gateway za internet [2]. U tom slučaju se informacije koje su potrebne za komunikaciju mogu prikupljati, spremati i analizirati.

Osim tipova komunikacije, VANET mreža se može podijeliti i na dva tipa okruženja, okruženje infrastrukture i ad-hoc okruženje [2]. Razlika između ta dva okruženja se nalazi u vremenskom trajanju povezanosti. Okruženje infrastrukture je trajno povezano i upravlja sa samim prometom. Također se svako vozilo promatra kao pojedinačno. Zadatak okruženja je prepoznavanje registracije vozila.

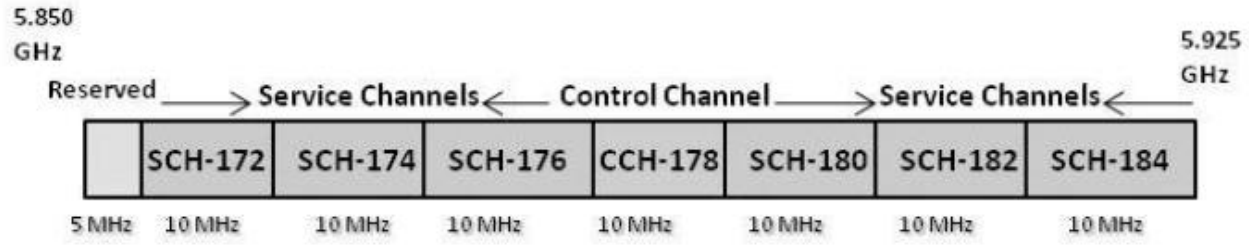
Drugo okruženje VANET mreže, koje je usko povezano s temom ovog rada je ad-hoc okruženje, odnosno promjenjivim okruženjem, ovisno vozilima koja se kreću unutar mreže. Takva vozila sadrže tri uređaja, komunikacijsku jedinici i dva senzora. Komunikacijska jedinica je zadužena za već objašnjene V2V, V2I i I2I komunikacije. Dva tipa senzora služe za mjerenje potrošnje i za mjerenje okruženja, odnosno koliko troši goriva i kolika je udaljenost do drugog vozila. Za sam proračun se koristi dodatna platforma a to je TPM- *Trusted Platform Module.*

Prva dva tipa, V2V i V2I komunikacija su određene standardom DSRC *(Dedicated Short Range Communication).* DSRC je bežična komunikacija na malu ili srednju udaljenost i ta komunikacija može biti u jednom ili u oba smjera. Da bi taj standard mogao ispravno funkcionirati, pored ceste moraju biti uspješno postavljeni uređaju koji se nazivaju *Road-Side Unit* [2]. Takvi uređaji imaju mogućnost velike brzine slanja i primanja podataka kao i veliku memoriju za pohranu podataka. Road Side Unit je poveznica između vozila koji se kreću i infrastrukture.

U samoj mreži postoji nekoliko protokola usmjeravanja kako bi sve poruke stigle na svoje odredište. Dijele se na *unicast* i *multicast.* Unicast označava pojedinačno slanje poruke dok multicast šalje poruku na nekoliko različitih odredišta. Usmjeravanje poruka se može temeljiti na nekoliko različitih faktora. Može ovisiti o geografskom položaju, o topologiji same mreže ili o nakupini čvorova. Čvorovi se nakupljaju da se mreža poveže i postane što stabilnija.

Da se izbjegne gubitak paketa na putu do vozila, već spominjani IEEE 802.11 standard koristi TDMA (Time Division Multiple Access) pristup [4]. Takav pristup omogućava da svaki korisnik koristi odabrani kanal, no kao što samo ime kaže, vremenski je određeno kada će koji korisnik koristiti taj kanal.

Širina pojasa je 75 MHz i taj pojas se dijeli na 6 servisnih kanala i jedan kontrolni kanal [4].



Sl.1.1 Prikaz raspodjele kanala kod DSRC-a [4].

2.2 Algoritam za V2V model

V2V model komunikacije je model slanja i primanja podataka, tj. poruka između dva vozila koja se kreću. Algoritam kojeg taj model koristi se može razdijeliti na dvije grane- jedna grana je pristup u kojem pošiljatelj poruke bira odašiljača ovisno o njegovoj daljini te broju skokova. Druga grana, odnosno metoda, je podređena primatelju poruke, gdje taj isti primatelj poruke sam bira da li će proslijediti poruku.

Ovaj algoritam radi tako da svaki čvor sadrži informaciju o svojim susjednim čvorovima koji su udaljeni jedan skok. Informacije koje sadrži su kada su sreli taj susjedni čvor, njegova udaljenost, GPS pozicija i broj susjeda tog čvora [1]. Kada neki čvor ima uvid u sve te informacije, onda može i izračunati vrijednosti susjeda.

Standard DSRC, koji omogućuje razmjenu podataka među vozilima radi na frekvenciji 5.9 GHz. Njegov maksimalni domet je 1 km, što je i više nego dovoljno za izravnu komunikaciju između dva vozila.

2.2.1 Beacon poruke

Beacon poruka u VANET mreži su kratke poruke koje sadrže status vozila s nekoliko osnovnih informacija o vozilu [1]. Veličina takvih poruka je puno manja od normalnih poruka koje sadrže podatke, što je velika prednost kod dinamičnih mreža. Beacon poruke se prosljeđuju i kroz jednostruki skok i kroz višestruke skokove, i to na aplikacijskom sloju.

Interval slanja beacon poruka se razlikuje i iznosi od minimalno 0.1 s do maksimalno 5.s [1]. Što je navedeni interval manji, same informacije koje se šalju su novije i točnije, no to ima i negativnih strana. Ako je prevelika gustoća prometa, mali interval slanja beacon poruka će zagušiti mrežu, i onda dolazi do problema, jer drugi podaci onda stižu s velikim zakašnjenjem.

Najbolje je koristiti srednji interval kod beacon poruka, jer ima najbolji omjer učinkovitosti i troškova.

ID	Ime	Prioritet	Adresa primatelja (ID primatelja ili 0 za *razašiljanje*)	GPS koordinate trenutne pozicije	Broj 1-skok susjeda

Sl.2.1 Sadržaj jedne beacon poruke

Kao što je prikazano na slici, beacon poruka sadrži:
-ID
-Ime
-Prioritet
-Adresu primatelja
-GPS koordinate pozicije
-Broj jednostrukog skoka susjeda

Ono što je bitno kod primanja beacon poruka je to da li je ta poruka stigla od novog ili starog susjeda. U slučaju da je pristigla od starog susjeda, podaci se brišu iz liste susjeda. Vremenski rok za pojavljivanje starog susjeda se računa po formuli 1.5* interval beacon poruke. (0.1s-5s). Podaci o

svakom susjedu se pohranjuju u listu čvorova [1]. Sva vozila koja se nalaze na toj listi čvorova mogu međusobno izravno komunicirati, a po broju vozila s te liste se računa gustoća prometa.

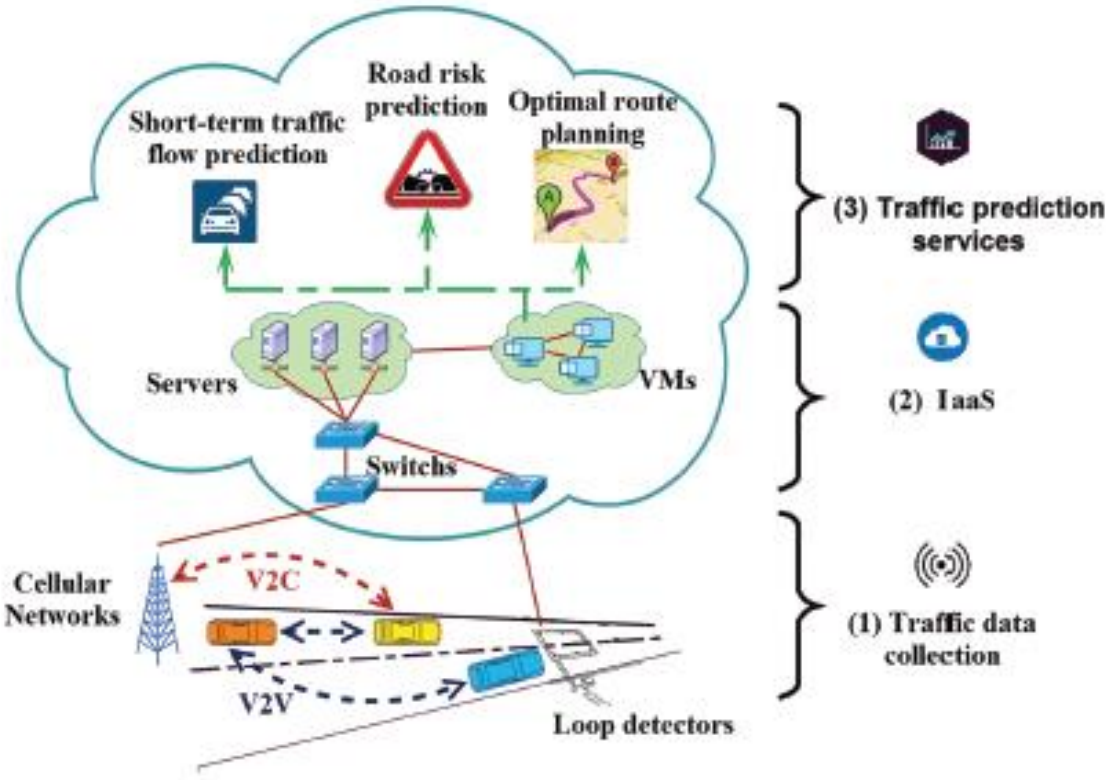
2.3 Arhitektura VANET-a

Prikupljanje i obrada podataka VANET mreže se može podijeliti u nekoliko razina, a sve te razine zajedno čine samu arhitekturu mreže. Svi podaci se čuvaju i razmjenjuju u VANET-Cloud arhitekturi. Svrha postojanja ovakve arhitekture je smanjenje grešaka koje se događaju prilikom detekcije podataka, kašnjenja i same kvalitete [3]. Postoje tri sloja arhitekture i oni su:

-*Traffic data collection layer*

-*Infrastructure as a service layer*

-*VANET Cloud layer*



Sl.2.2 Arhitektura VANET mreže [3].

2.3.1 Traffic data collection layer

Ovaj sloj ima ulogu prikupljanja podataka. Sve podatke koje obrađuje i prikuplja dobiva iz same mreže prometa. Ono što je bitno napomenuti da podaci s kojima barata se moraju odrađivati u *real-time* vremenu, odnosno gotovo trenutno. Nekoliko parametara s kojima ovaj sloj radi su: brzina samih vozila, stanje ceste i prometni tok [3].

Da bi se moglo efikasno upravljati autonomnim vozilima i usmjeravati ih na najkraći put, potrebno je detektirati zagušena područja, tj gužve na cesti. Također je potrebno znati same brzine kojima se vozila kreću. Uz sve to za neke stvari je bitna i prošla situacija na tim dijelovima ceste. Za potpun rad ovog sloja arhitekture koriste se dva sučelja, komunikacijsko sučelje, i sučelje 'osjetljivih' podataka [3]. Zadatak sučelja osjetljivih podataka je da prosljeđuje podatke u *cloud.* Cloud će biti opisan u nastavku rada. Komunikacijsko sučelje je zaduženo za pristup prometnim

uslugama u stvarnom vremenu [3]. Vozila koriste 3G, 4G ili 5G mrežu, a poželjno je da je mreža sa što većom brzinom i što manjom latencijom.

2.3.2 Infrastructure as a service layer

Sloj u kojemu se infrastruktura koristi kao uslužni servis ima dva podsloja, fizički podsloj resursa i virtualizirani podsloj, [3]. Fizički podsloj resursa grupira same poslužitelje ovisno o njegovim karakteristikama. Poslužitelji su odgovorni za spremanje podataka o prometu i to u stvarnom vremenu. Osim toga, oni dozvoljavaju pristup raznim već postojećim bazama podataka o prometu i stanju na cestama. To su fizički poslužitelji, koji se sastoje od preklopnika i servera. Preklopnik se može promatrati kao svojevrsna veza između prvog sloja za skupljanje podataka i drugog, servisnog sloja. Osim fizičkih postoje i računalni poslužitelji čiji je jedini zadatak da omoguće rad s visokom kvalitetom. Virtualizirani podsloj je podsloj koji služi za izbjegavanje pada kvalitete mreže. Pad kvalitete se događa kada u nekom trenutku dođe do velikog broja vozila u samoj mreži, samim time i ogromne količine podataka koja mora biti obrađena u stvarnom vremenu, a to za uspješan rad VANET mreže nije dozvoljeno. Upravo zato se cloud javlja kao učinkovito rješenje.

2.3.3 Sloj VANET oblaka

VANET cloud sloj, odnosno oblak je zadužen za obradu prikupljenih informacija iz prometa. Nakon što sve podatke uspješno obradi, sljedeći korak je optimizacija predviđanja podataka. Osim predviđanja bitno je da podaci koji se šalju svakom vozilu budu stvarno i dostavljeni tom istom vozilu a ne nekom drugom. Predviđeni podaci u oblaku sadrže informacije kao što su vrijeme putovanja, protok prometa, brzina, zatvorene ceste i slično, [3].

Već navedno prikupljanje i obrada podataka je centar rada ovog sloja i to omogućuju detektori koji se nalaze kraj ceste, RSU. *Road side unit* takve neobrađene podatke prosljeđuje u ovaj sloj pomoću UC 7420 uređaja. To je mrežni uređaj koji služi za povezivanje uređaja za prikupljanje podataka i semafora pomoću UMTS mreže.

Oblak nudi dva tipa usluge. Prvi tip usluge je na zahtjev samog korisnika, a drugi tip usluge je automatska usluga, [3]. Usluga koja se određuje na zahtjev izračunava podatke koji su bitni samom korisniku kao što su planiranje plana puta i vrijeme putovanja. Automatska usluga je ona koja je bitna za sve sudionike prometa, kao što je prometna nesreća. Takvi događaji se sami šalju korisnima.

2.4 Razmjena podataka u VANET mreži

Postoji nekoliko modela prema kojima se poruke šalju i primaju u VANET mreži. Svaki model radi na drugačiji način, međutim zajedničko im je to da osiguravaju jednostavan mehanizam razmjene podataka. Postoje tri modela razmjene podataka i oni su

-reaktivna razmjena podataka

-proaktivna razmjena podataka

-hibridna razmjena podataka

Reaktivna razmjena se temelji na protokolima koje rade samo kada korisnik traži nekakvu informaciju stoga takvi protokoli daju bolje rezultate. Glavni reaktivni protokol je AODV (Ad hoc on demand distance vector). Njegov zadatak je pronaći rutu do odredišta, međutim samo kada ta ruta nije unaprijed poznata. Princip rada ovog protokola je taj da svaki čvor unutar mreže sadrži svoju vlastitu tablicu usmjeravanja. U toj tablici se nalazi nekoliko informacija, ali ona najbitnija je informacija o dolasku na samo odredište. U slučaju da se otkrije novi, kraći put do odredišta, AODV će sačuvati taj novi, a staru rutu će obrisati.

U slučaju da nam je bitno nekoliko puteva do odredišta, tada je moguće u mreži koristiti drugi reaktivni protokol a to je AOMDV (Ad hoc on demand multi path distance vector). Kao što mu i samo ime kaže, on podržava nekoliko ruta i taj protokol nam je bitan u slučaju kada se na jednoj ruti dogodi neočekivan zastoj ili prometna nesreća. Ono što je problem kod ovog protokola je to što on ne radi u realnom vremenu. Kako bi poboljšao taj dio, nova verzija ovog protokola je uzimala u obzir kombinaciju brzine i smjera čvorova

Proaktivna razmjena podataka se temelji na periodičnom ažuriranju podataka, točnije usmjerivačke tablice. Glavni protokol kod ovakve razmjene je usmjerivački protokol temeljen na gustoći. Sam protokol rangira prometnice po gustoćama i slaže ih u hijerarhiju. Ovaj protokol radi u realnom vremenu. Kada računa gustoću na prometnicama, prvo šalje testni paket, te se tek nakon toga odabire ruta za vozilo.

Drugi proaktivni protokol je OLSR (*Optimized Link state routing*). Razlika između dva proaktivan protokola je ta što OLSR čuva cijelu rutu od izvora do odredišta u tablici koja se periodično ažurira. U slučaju da se neka ruta izgubi, automatski će tražiti novu rutu umjesto te pomoću Bellman Ford algoritma. Nedostatak ovog algoritma je zagušenje mreže zbog stalnog ažuriranja velikog broja podataka te velika potrošnja energije.

Hibridna razmjena podataka se temelji na adaptivnom usmjerivačkom protokolu, koji je kombinacija reaktivnog i proaktivnog. Ono što ovaj protokol uzima u obzir su brzina i gustoća čvorova. Gustoću čvorova računa pomoću LET-a (*Link expiration time*).Ako se čvorovi kreću velikom brzinom, a gustoća im je mala, onda će LET biti kratak, dok obratno ako je LET relativno dug, znači da je mreža nešto statičnija. Nedostatak adaptivnog usmjerivačkog protokola je to što koristi veliki broj periodičnih poruka pa može doći do zagušenja.

Još jedan tip protokola koji postoji su protokoli temljeni na poziciji, a najpoznatiji među njima je GSR (*Geographic Source Protocol*). Kao što samo ime kaže, ovaj protokol koristi karte gradova i povlači informacije iz njih koristeći RLS, *reactive location service*. Drugi pozicijski protokol je GPSR (*Greedy perimeter stateless routing*) koji ima dva načina rada. Jedan način je da prosljeđuje pakete čvoru koji mu je geografski najbliži, a drugi način rada je da se paketi prosljeđuju duž niza čvorova. GPSR je nepouzdan i ima veliki gubitak paketa, zato se pozicijski protokoli ne koriste u VANET mrežama.

Same poruke koje se šalju u VANET mreži također se razlikuju po nekoliko stvari, da li u arhitekturi mreže idu kao uplink ili kao downlink, da li se šalju nakon zahtjeva, da li se šalju periodično ili ne.

Event description message je poruka koja je jedna od najvažnijih za primjenu VANET mreže kod autonomnih vozila. Ove poruke obavještavaju ako se dogodio neki neočekivan scenarij na cesti,

kao naprimjer prometna nesreća. Lokacija takvog događaja se otkriva pomoću GPs-a. Ovaj tip poruke se šalje u oblak i tako javlja opis i lokaciju događaja svim ostalim vozačima, uz vrijeme kada se događaj dogodio i kojem se točno vozilu radi.

On-demand message je poruka koju sam vozač šalje zbog raznih upita oko općenitih informaciju u prometu. Primjer je trajanje vožnje. Ova poruka sadrži tri informacije, podatak o kojem se vozilu radi, zahtjev za neku informaciju te lokacija samog vozila.

Data collection message skupina poruka koje oblak skuplja i obrađuje kako bi imao detaljnije informacije o svakom pojedinačnom vozilu, što pomaže i kod kasnije predikcije događaja. Za skupljanje ovih podataka uključeni su i V2V i V2I načini komunikacije. Data collection poruke se dijele na *uplink* i *downlink*. Downlink su zahtjevi za poruke od strane oblaka, dok su uplink, odgovori na te zahtjeve, odnosno slanje poruka vozila prema oblaku.

Traffic information message su kao što i samo ime kaže, poruke s informacijama o prometu. Na temelju ovih poruka se računa najkraća ruta za neko određeno putovanje.

Periodic message su poruke koje se šalju periodično kako bi se mreža konstantno osvježavala u slučaju neke promjene na cesti, novih puteva i slično.

Event driven message je poruka čiji je zadatak da oblak osim što sadrži informacije o stanju na cesti, također sudjeluje u sigurnosti u prometu.

Kao što je već napomenuto, *event description messages* su najvažnije za ljudsku sigurnost. Upravo zato u mreži postoji algoritam koji kada se dogodi neočekivan događaj, dozvoljava da te poruke generiraju i šalju samo vozila u blizini tog događaja, točnije svjedoci. Ostala vozila nisu u mogućnosti kreirati vlastite poruke s opisom događaja da ne dođe do zagušenja.

Komunikacija među dva vozila može biti nedovoljno dobra ili čak u potpunosti prekinuta. Da bi se izbjegao taj problem VANET mreža koristi različite frekvencije koje ovise o tome u kakvom se području koriste, naseljeno područje ili ruralno područje. Kada vozilo prelazi iz jednog u drugo područje zadatak robusnih usmjerivačkih protokola je da se veza ne prekine i da se podaci uspješno

razmjenjuju. Protokoli će uvijek imati spremnu rezervnu vezu koju će aktivirati kada originalna veza oslabi.

U V2V modelu postoje dvije metode razmjene poruka. Prva metoda ima pristup usmjeren prema pošiljatelju, dok druga metoda ima pristup usmjeren prima primatelju poruke. Kada se metoda temelji na pošiljatelju, on sam bira koji će od njegovih susjednih mobilnih čvorova proslijediti poruke dalje. Njegovi susjedni čvorovi se smatraju oni koji se nalaze 1 skok. Kod druge metode, koja se još naziva i *BackUp* metoda, sami primatelji biraju hoće li oni proslijediti poruke dalje.

3. SIMULACIJA I ANALIZA REZULTATA

Praktični dio ovog rada će se izvesti u dva simulacijska okruženja, CupCarbon simulatoru i Veinsu. Oba programa služe kao alat kojim se simuliraju kretnje mobilnih stanica, te komunikacija između njih. Veins je software koji simulira točno VANET komunikaciju između vozila i infrastrukture, dok CupCarbon koristi jednostavniji pristup i modele, točnije senzore. Cilj je provesti identične simulacije komunikacije među vozilima unutar u oba programa, te usporediti rezultate.

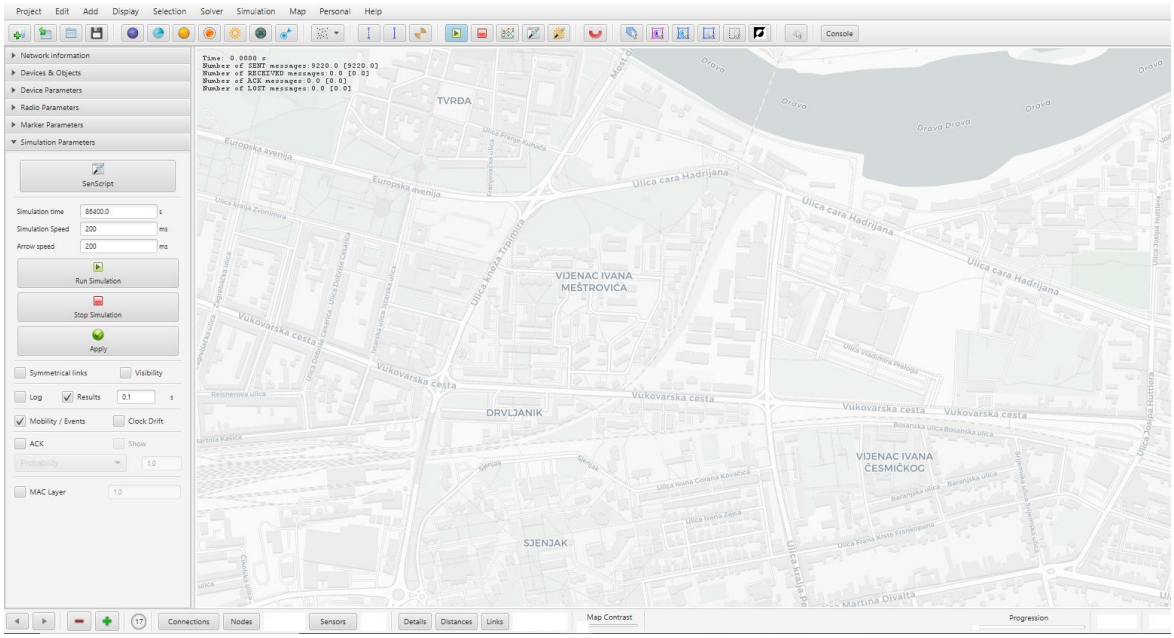
Provesti će se dva tipa simulacije u oba programa. Simulirat će se kretnje dva vozila, a prvo vozilo će nenadano stati, što će predstavljati prometnu nesreću. Kako je cilj prenjeti tu informaciju do drugog vozila, analizirat će se vrijeme potrebno da paket s tom porukom stigne do vozila. Drugi tip simulacije je zahtjev od drugog vozila da sazna gdje se nalazi prvo i da li je prvo vozilo zastalo. Taj tip zahtjeva se naziva *ad hoc on demand distance vector*. Grafički će se prikazati usporedba vremena slanja za oba tipa komunikacije u istom programu, te za isti tip komunikacije unutar dva programa i analizirati rezultate.

3.1 CupCarbon simulator

CupCarbon pripada u SCI-WSN simulatore. SCI-WSN je kratica za Smart City and Internet of Things Wireless Sensor Network. Pomoću njega se uspješno može dizajnirati i vizualizirati razne algoritme. Ono što je bitno za sam ovaj rad je vizualiziranje rada mrežnih senzora, mrežnih topologija i protokola. CupCarbon ima jedan nedostatak a to je da ne podržava rad svih slojeva mreže, nego se temelji na aplikacijskom sloju.

Postoje dvije vrste simulacija u CupCarbon simulatoru. Prva vrsta simulacija omogućuje dizajn mobilnih scenarija te generiranje prirodnih događaja. Druga vrsta simulacije koju nudi je simulacija okoliša s diskretnim događajima preko bežičnih mrežnih senzora.

CupCarbon koristi Open Street Map (OSM). To je vrlo bitno jer je moguće testirati razne scenarije u pravim gradovima, na pravim prometnicama. Zahvaljujući tome moguće je postaviti mrežne senzore na bilo koju poziciju na karti.. Postoji nekoliko osnovnih dijelova ovog softvera.



Sl.3.1 Prikaz grafičkog sučelja CupCarbon simulatora

Kao što je vidljivo na slici 3.1, CupCarbon se sastoji od nekoliko dijelova, a to su traka izbornika, alatna traka, sučelje s parametrima uređaja i simulacije, mape, komandnog prozora i trake s detaljima simulacija, [5].

Grafičko sučelje je vidljivo čim se uđe u simulator, i ono služi da postavljanje mobilnih stanica (vozila), baznih stanica,senzora i markera, direktno na kartu. Te iste senzore se konfigurira u SenScript prozoru.

SenScript prozor je dio simulatora u kojemu se može svaki senzorski čvor konfigurirati da radi ovisno o potrebama. Može se podesiti da radi kao usmjerivač, kao izvor ili kao odredište slanja paketa. Kako je pisan u Java programskom okruženju, moguće je dodavati razne mogućnosti ovom simulatoru.

Ovaj program nudi nekoliko mogućnosti simulacija raznih scenarija i primjene različitih algoritama. Trenutna verzija omogućava dinamičko konfiguriranje čvorova, što znači da jedan čvor može raditi u više mreža. Ovisno u kojim mrežama čvor sudjeluje dodjeljuju mu se adrese i

kanali. Osim vremena potrebnih za izvršavanje simulacije, odnosno vremena puta mobilnih čvorova, CupCarbon nudi i prikaz potrošnje energije, što je još jedan bitan faktor za provjeru prije primjene na autonomna vozila.

- *Sensor Node* – čvor koji predstavlja klasični senzorski čvor te nema dodatnu ulogu
- *Media Sensor Node* – također senzorski čvor, no za razliku o sensor node-a, ovaj čvor prikuplja podatke iz okoline u koju je postavljen. Konfiguracija čvora se vrši u SenScript prozoru.
- *Base Station* (*Sink*) – statični senzorski čvor, ono što je jako važno kod baznih stanica da nemaju potrošnju baterije nego je neograničena
- *Mobile* – Mobila stanica koja simulira bilo koje uređaje koji se kreću, u ovom radu simulira kretnje vozila.
- *Markers* – markeri su uređaji koji također imaju nekoliko različitih uloga. Prvo i najbitnije je da određuju rutu po kojoj će se mobilna stanica odnosno vozilo kretati. Druga uloga je da ograničavaju područje rada senzora, ukoliko je to potrebno [5]

3.2 VEINS

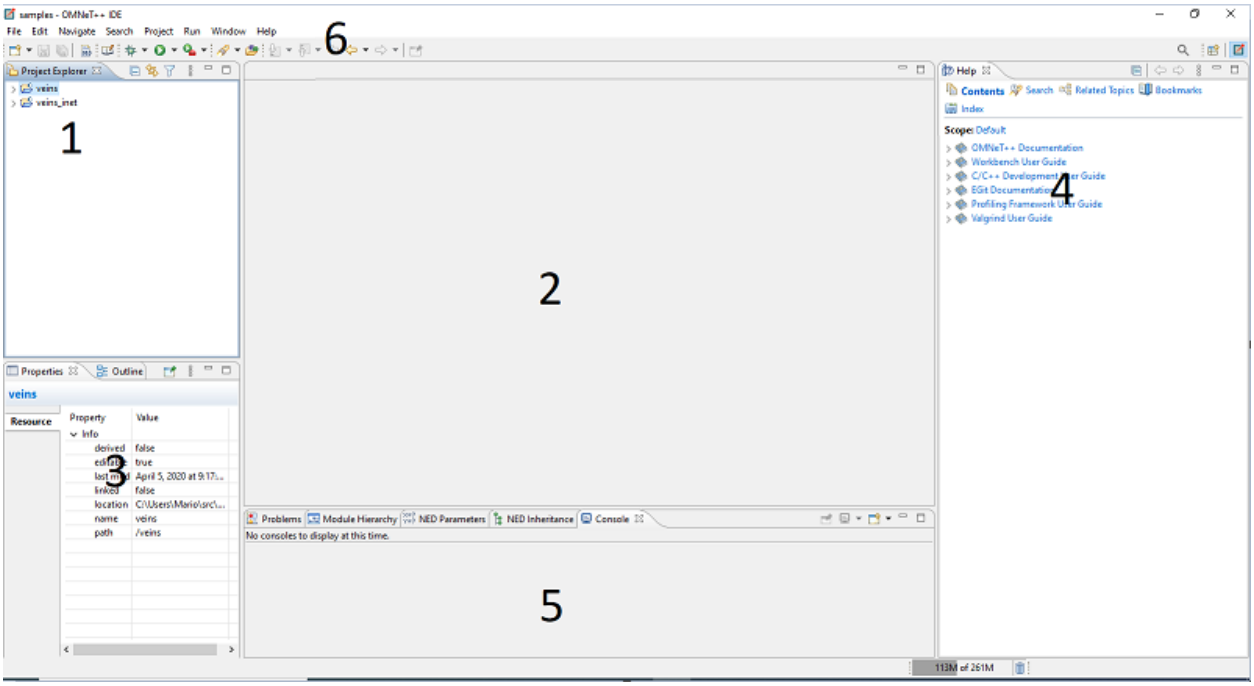
VEINS je simulacijsko okruženje koje provodi mrežne simulacije vozila. Povezuje i koristi dva simulatora, OMNet++ i SUMO. Pomoću tog skupa modela Veins provodi realne mrežne simulacije i što je bitno napomenuti, bez gubitka brzine.

OMNeT++ (Objective Modular Network Testbed in C++) se koristi za izgradnju mrežnih simulatora. Od 1997. godine je postao javno dostupan. OMNet++ je stvoren simulacijom komunikacijskih mreža i multiprocesora i rađen je tako da bude što jednostavniji. Upravo se iz tog razloga koristi u mnogim simulacijama, od simulacija bežične i ad-hoc mreže do simulacija mreže

peer to peer. Pomoću OMNeT je moguće kreirati nekoliko vrsta mreža, a to mogu biti žične i bežične komunikacijske mreže, senzorske mreže i što je posebno bitno *ad-hoc* mreže. Također OMNeT++ koristi veliki niz alata od kojih su najbitniji Eclipse, integrirano razvojno okruženje za programiranje i grafička mogućnost postavljanja i izvođenja simulacije. Komponente koje se koriste u OMNeT-u su programirane u C++ jeziku.

SUMO je kratica za *Simulation of urban mobility* je simulator za promet na cesti. Osim što je besplatan i lako dostupan, SUMO ima veliki kapacitet s kojim može istovremeno obrađivati 10 000 cesta. SUMO se može promatrati kao alat s kojim se pronalaze rute, vizualiziraju postavljene mreže te se računa vrijeme kod usmjeravanja paketa. Kako može podržati veliki broj čvorova u mreži, mogu se detaljno postaviti vozila, pješaci i javni prijevoz. Da bi rezultati bili što precizniji, dizajniran je tako da se mikroskopski točno zna uzdužna i bočna pokretljivost svakog vozila. Da bi se ad-hoc mreža mogla detaljno izvesti, svakom od vozila u mreži se rute mogu dodijeliti statički, dinamički ili treći tip, ruta se generira prema voznom redu. Da ne bi dolazilo do zagušenja mreže, koristi se algoritam za izvođenje stabilne raspodjele protoka. Rezultat tog algoritma je optimalno korištenje cesta.

Veins uspješno generira interakciju između prometa na cesti i mrežnih simulatora u stvarnom vremenu zahvaljujući korištenju već opisanih modela, OMNeT-u i SUMO-u- Također podržava simulaciju bežičnih komunikacijskih protokola u VANET mreži.



Sl. 3.2 Prikaz OMNeT++ sučelja

Nakon instalacije, OMNeT++ se pokreće iz Windows Comand skripte pomoću naredbe *omnetpp.* Nakon nekog vremena otvori se prikaz kao na slici 3.2. Sučelje se može podijeliti u 6 dijelova:

1: *project explorer* ili *workspace*- dio OMNeT-a u koji se mogu implementirati gotovi projekti koji se kasnije pokreću kao OMNeT++ simulacije

2: *code source editor*- "glavni" dio okruženja, editor u kojemu se nalaze zaglavlja i cijeli kod

3: *property view*- sadrži sva svojstva odabranih elemenata, kao što su ime i tip datoteke

4: *help*- dio u samom okruženju koji služi za pomoć pri pronalasku stvari ili uputa

5: *console*- prozor u kojem se ispisuju izlazne vrijednosti tj. *output*

6: alatna traka pomoću koje se između ostalog provjeravaju greške u pisanom kodu, te se pokreće sama simulacija

3.3 Simulacija VANET komunikacije

3.3.1 V2V komunikacija u Veinsu

Prvi scenarij koji će se provesti u programu Veins je simulacija automobilske nesreće. Pojednostavljeno, tri vozila se kreću istom rutom jedno za drugom. Nakon nekog vremena prvo vozilo se zaustavi te je potrebno poslati poruku drugim vozilima kako bi izbjegli nesreću te odredili alternativnu rutu.

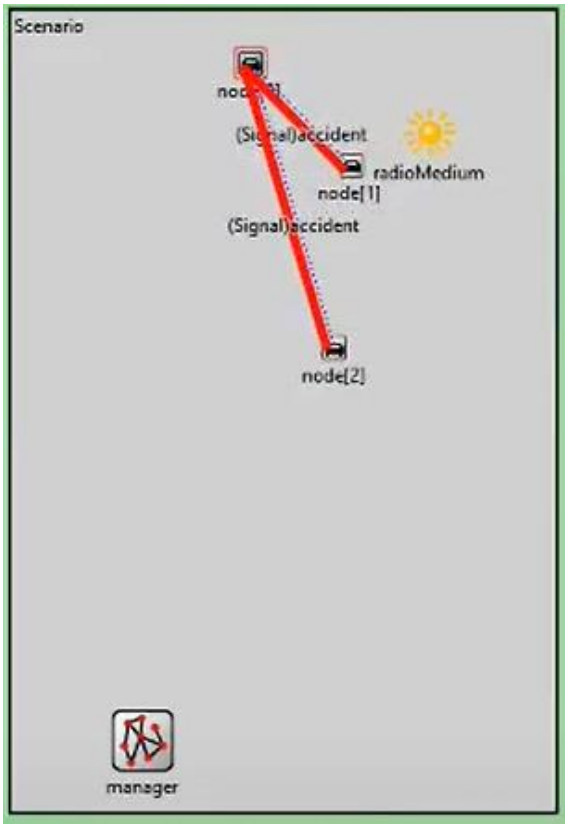
Scenario.ned | VeinsInetSample | VeinsInetSample | VeinsInetMobilit | VeinsInetMobilit | VeinsInet

```
Define_Module(VeinsInetSampleApplication);

VeinsInetSampleApplication::VeinsInetSampleApplication()
{
}

bool VeinsInetSampleApplication::startApplication()
{
    // host[0] should stop at t=20s
    if (getParentModule()->getIndex() == 0) {
        auto callback = [this]() {
            getParentModule()->getDisplayString().setTagArg("i", 1, "red");

            traciVehicle->setSpeed(0);

            auto payload = makeShared<VeinsInetSampleMessage>();
            timestampPayload(payload);
            payload->setChunkLength(B(100));
            payload->setRoadId(traciVehicle->getRoadId().c_str());

            auto packet = createPacket("accident");
            packet->insertAtBack(payload);
            sendPacket(std::move(packet));
        };
        timerManager.create(veins::TimerSpecification(callback).oneshotAt(SimTime(20, SIMTIME_S)));
    }

    return true;
}

bool VeinsInetSampleApplication::stopApplication()
{
    return true;
}
```

Sl.3.3 Dio koda koji prikazuje zaustavljanje i slanje poruke

Na slici 3.3 je prikazan dio koda koji je zadužen nakon 20 sekundi simulacije normalne vožnje, prvo vozilo, odnosno čvor 0, zaustaviti tako što mu se brzina postavlja na 0. Također čvor postaje

crvene boje, kako bi bilo uočljivije. Nakon toga se kreira poruka u zaglavlju *VeinsInetSampleMessage* veličine 100 bajta te se šalje ostalim vozilima u mreži. Drugi čvorovi ili vozila kada prime tu poruku pod nazivom 'nesreća' postaju zelena. Kada su čvorovi zeleni onda mijenjaju rutu.



Sl.3.4 Prikaz slanja paketa drugim vozilima

Event#	Time	Relevant Hops	Name	ID / Source	Kind / Destination	Length / Protocol	Type	Length	Info
#231	20	node[0] --> node[1]	accident	10.0.0.10:9001	224.0.0.1:9001	UDP	DATA	175 B	(UNKNOWN) VeinsInetSampleMessage, length = 100 B | 9001-
#231	20	node[0] --> node[2]	accident	10.0.0.10:9001	224.0.0.1:9001	UDP	DATA	175 B	(UNKNOWN) VeinsInetSampleMessage, length = 100 B | 9001-
#267	20.000210378973	node[2] --> node[0]	relay	10.0.0.132:9001	224.0.0.1:9001	UDP	DATA	175 B	(UNKNOWN) VeinsInetSampleMessage, length = 100 B | 9001-
#267	20.000210378973	node[2] --> node[1]	relay	10.0.0.132:9001	224.0.0.1:9001	UDP	DATA	175 B	(UNKNOWN) VeinsInetSampleMessage, length = 100 B | 9001-
#287	20.000485612340	node[1] --> node[0]	relay	10.0.0.71:9001	224.0.0.1:9001	UDP	DATA	175 B	(UNKNOWN) VeinsInetSampleMessage, length = 100 B | 9001-
#287	20.000485612340	node[1] --> node[2]	relay	10.0.0.71:9001	224.0.0.1:9001	UDP	DATA	175 B	(UNKNOWN) VeinsInetSampleMessage, length = 100 B | 9001-
#307	20.000825795283	node[0] --> node[1]	relay	10.0.0.10:9001	224.0.0.1:9001	UDP	DATA	175 B	(UNKNOWN) VeinsInetSampleMessage, length = 100 B | 9001-
#307	20.000825795283	node[0] --> node[2]	relay	10.0.0.10:9001	224.0.0.1:9001	UDP	DATA	175 B	(UNKNOWN) VeinsInetSampleMessage, length = 100 B | 9001-

Sl. 3.5 Prikaz poslanih paketa

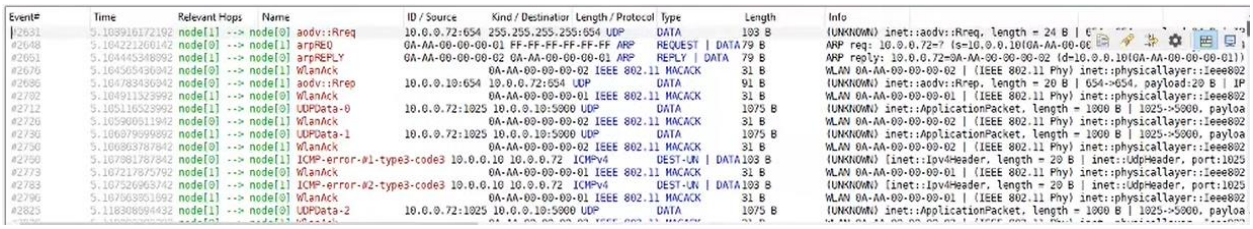
Na slici 3.5 se vidi prikaz slanja paketa. Prva dva paketa šalje čvor 0 koji se zaustavi. Nakon što generira poruku nesreća, paket pristiže čvorovima 1 i 2. Protokol kojim se šalju paketi je UDP *(User Datagram Protocol),* a veličina paketa 100 B. Vrijeme trajanja slanja paketa bližem čvoru, čvoru 1, je 252 ns, a vrijeme potrebno da paket stigne čvoru 2 je 771 ns.

3.3.2 VANET komunikacija na zahtjev u Veinsu

Drugi tip simulacije u Veinsu je slanje poruka između vozila, no za razliku od prvog primjera, ovdje se poruka šalje tek na zahtjev vozila. U konkretnom primjeru drugo vozilo u nizu, čvor 1 šalje vozilu ispred sebe, čvoru 0 zahtjev te nazad dobiva povratnu informaciju. U ovom slučaju paketi koriste usmjerivački protokol, AODV (ad-hoc distance vector).

Sl.3.6 Prikaz AODV komunikacije

Prijašnji primjer i ovaj su jako slični, jedina razlika je kod postavljanja podmodula vozila, tj čvorova. Razlika između čvorova u prvom primjeru i ovom sada je što sada postoji usmjerivački podmodul. Zahvaljujući njemu moguće je implementirati razne protokole, ne mora nužno biti AODV, no u ovom slučaju je. Druga promjena je dodana veza između dispečera poruke i usmjerivačkog ulaza.



Sl. 3.7 Prikaz poslanih paketa kod AODV-a

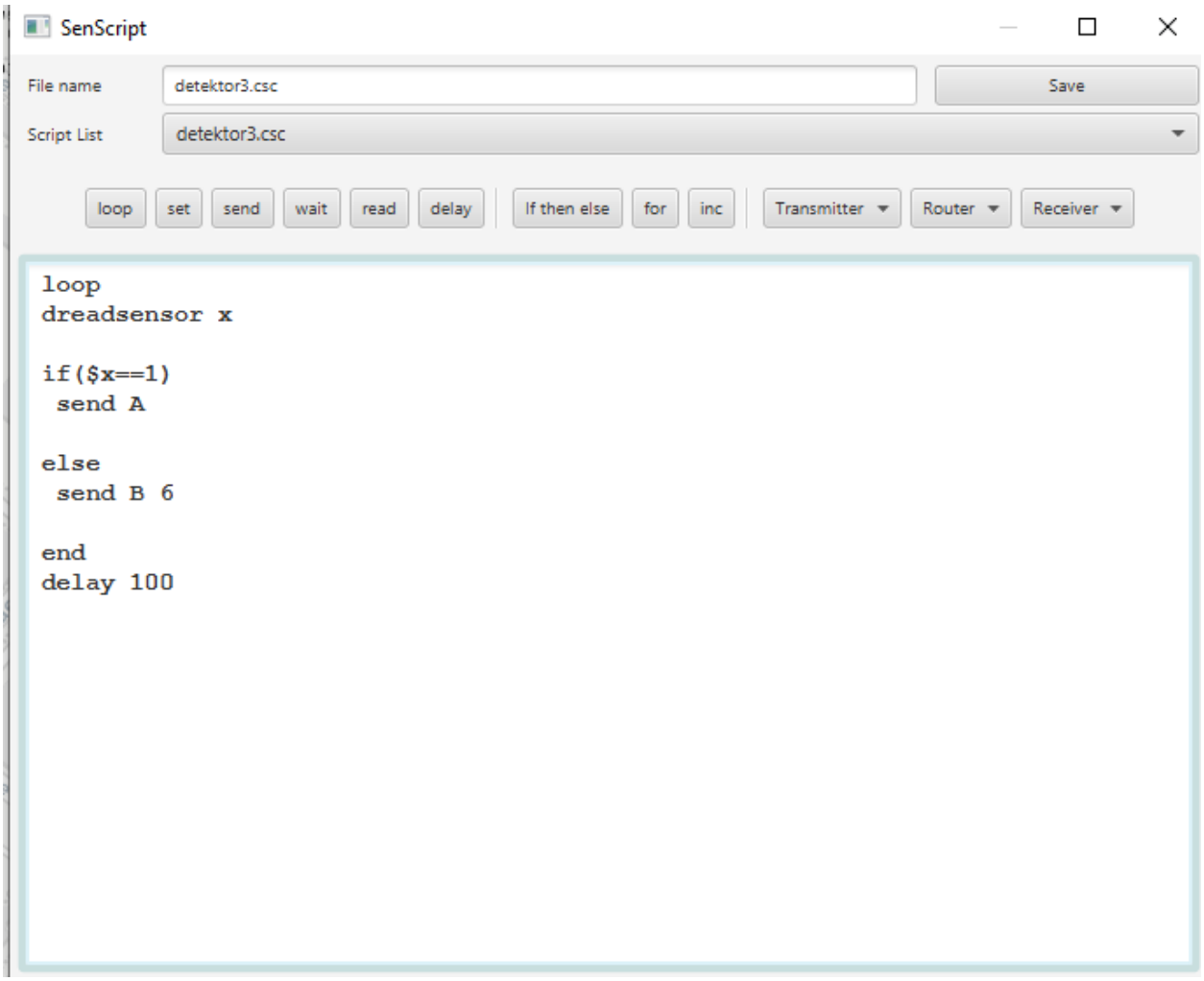
Ono što se prvo uočava kod liste poslanih i primljenih paketa je da je prvi paket poslan od strane drugog čvora prema prvome, i jasno se vidi *AODV request.* Nakon toga slijed komunikacije je normalan i nastavlja se prirodnim tokom. Također je vidljivo da se ne koristi svaki puta UDP protokol, već ovdje postoji nekoliko tipova kao što su ARP (Adress Resolution Protocol) te ICMP (Internet Control Message Protocol). Ukupno vrijeme potrebno da se dobije odgovor o stanju na cesti je vrijeme potrebno da se pošalje zahtjev plus odgovor. To vrijeme u ovom primjeru simulacije iznosi 256 mikrosekundi, što je značajno više nego kad se automatski generira takva poruka.

3.3.3 V2V komunikacija u CupCarbonu

Nakon što su provedene dvije simulacije u programskom okruženju Veins, cilj je izvesti što sličnije simulacije u CupCarbonu software-u kako bi se mogli usporediti vremena simulacije i slanja paketa. Prva simulacija je V2V komunikacija, međutim kako u ovom okruženju ona nije još izvediva, V2V se izvodi pomoću senzora koji predstavljaju *Road Side Unit.* Također zbog

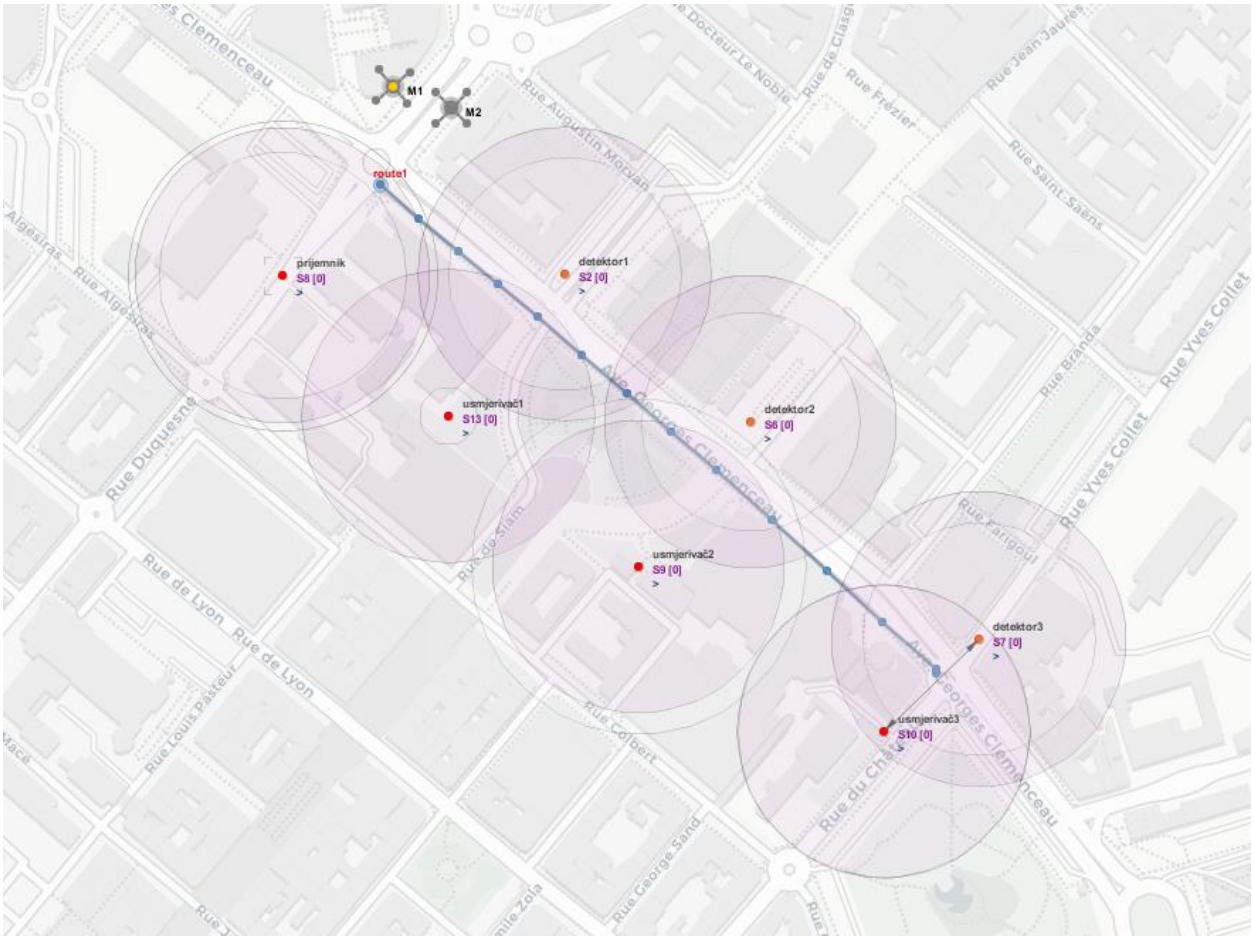
ograničenja programa, mobilni čvorovi ili vozila u našem slučaju moraju imati predodređenu rutu prije nego što bi krenuli. Kako se inače taj proces odvija automatski a ovdje će se morati ručno unjeti, što uzima puno vremena, u rezultatima i analizi se neće računati to vrijeme ručnog unošenja rute drugom vozilu. Dakle, kao i u prvom primjeru, vozilo jedan se kreće određenom rutom, te iznenada staje i potrebno je tu poruku prenjeti do drugog vozila. Također, radi lakšeg određivanja zaustavljanja, ovdje se neće podešavati vrijeme nakon kojeg staje nego će se jednostavno odrediti kraća ruta od prave.

Road Side Unit u stvarnosti ili senzori u simulaciji se postavljaju tako da detektiraju kada je vozilo prošlo pored njih te tu informaciju šalju drugim senzorima

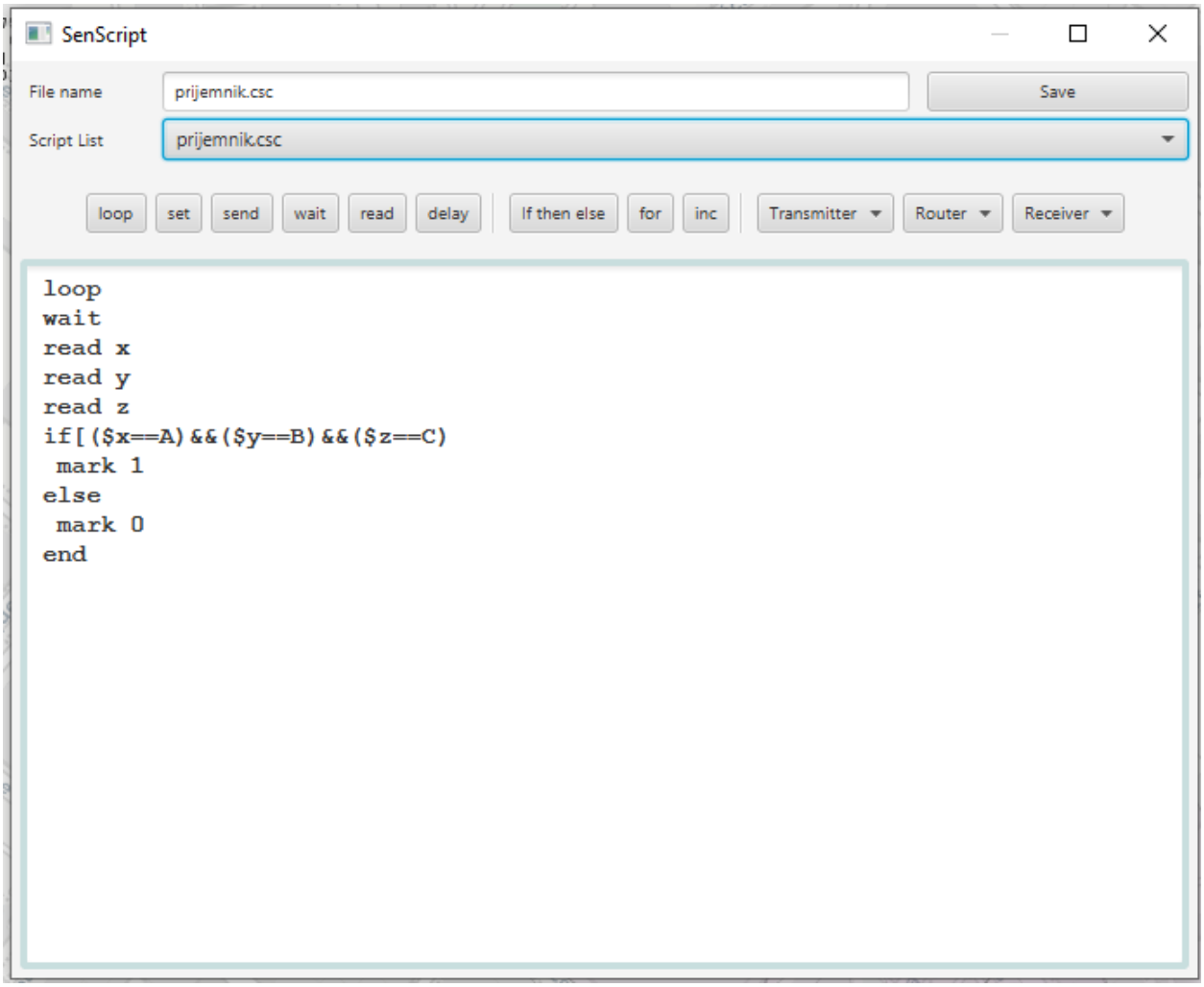


Sl 3.8 Primjer detektora vozila

U SenScript prozoru, kao što je već spomenuto, konfiguriraju se senzorski čvorovi. Da bi se moglo detektirati kod kojeg se čvora dogodila nesreća te javiti ostalim mobilnim čvorovima, mora se znati gdje se nalazi prvo vozilo. Pomoću koda sa slike 3.8 se detektira vozilo ako se nalazi u radijusu samog senzora. Senzori, ili RSU, se zato postavljaju što bliže cesti kako bi se što efikasnije iskoristili resursi. Postavljaju se senzori s obje strane ceste, jer je program postavljen tako da senzor može imati samo jednu ulogu, ili detektora ili usmjerivača. To nije efikasna upotreba, međutim tako se postavlja zbog lakše izvedbe simulacije.



Sl.3.9 Prikaz rasporeda senzora



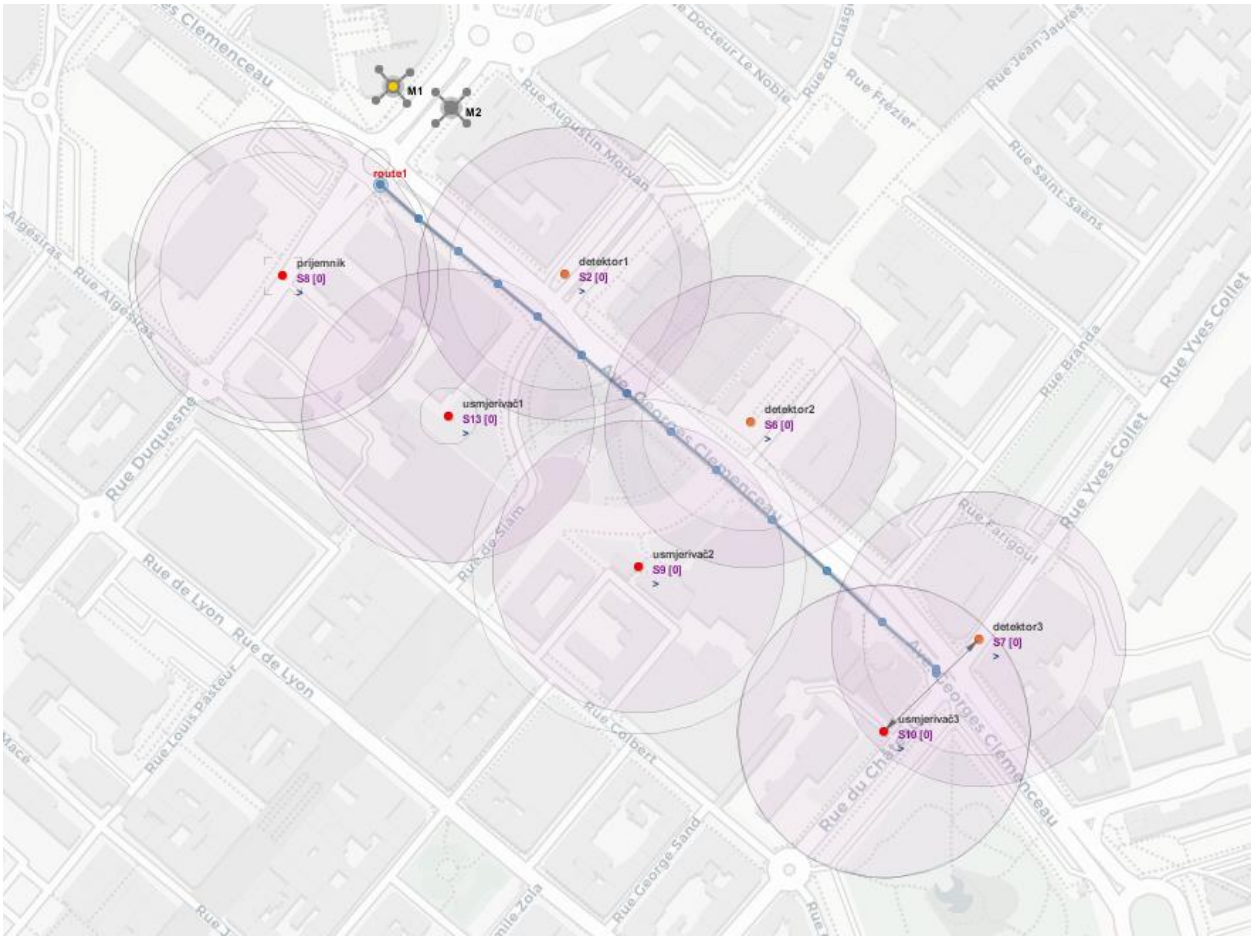
Sl.3.10 Senscript prikaz prijemnika

Na slici 3.10 se nalazi kod koji omogućuje prijemniku da očita ako je vozilo došlo do zadnjeg senzora. Budući da u primjeru postoje 3 detektora i tri usmjerivača, za svaki detektor koji uspješno detektira vozilo u svom radijusu, podešeno je da šalje drugu vrijednost. Jedan detektor šalje vrijednost A, drugi vrijednost B, treći C. Pomoću usmjerivača i očitanih vrijednosti je u prijemniku postavljen uvjet da samo ukoliko su očitane sve tri vrijednosti, prijemnik se označi, što daje vozilu broj 2 informaciju gdje se vozilo 1 zaustavilo. Još jedanput dolazi do izražaja neefikasnost komunikacije kada se koristi infrastruktura, za razliku od izravne komunikacije.

Na slici 3.9 je plavo označena ruta kojim će se kretati vozilo M1 i M2. Vozilu M1 je dodijeljena ruta da vozi do kraja postavljenih markera, a vrijeme simulacije je 20 sekundi, da bude identično kao kod simulacije u Veinsu. Nakon što se vozilo 1 zaustavi, gleda se vrijeme potrebno da paket dođe do prijemnika, te se nakon toga ručno postavlja ruta za vozilo 2. Ruta će biti ista kao za vozilo jedan, međutim bit će kraća, jer će vozilo dva prijeći samo onaj dio rute koji je označen da je vozilo 1 uspješno prošlo, odnosno da detektor više ne javlja njegovu prisutnost. Vrijeme potrebno da poruka stigne od trećeg detektora u nizu do prijemnika je 150 ms, jer svaki paket ima brzinu slanja od 50 ms.

3.3.4 Komunikacija na zahtjev u CupCarbonu

Cilj ove četvrte, ujedno i posljednje simulacije je provjeriti razliku između izravne komunikacije među vozilima i komunikacije na zahtjev. U CupCarbon simulatoru ne postoji *ad hoc on demand distance vector routing,* stoga će se pojednostaviti primjer tako da će se upit slati obrnutim redoslijedom, od prijemnika do detektora i nazad.



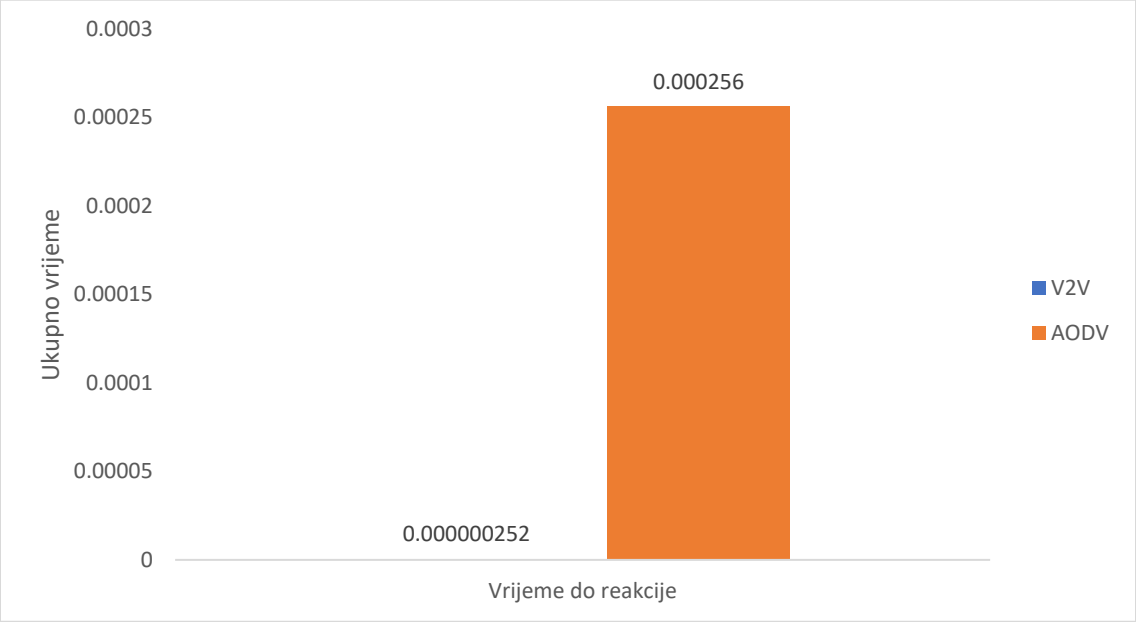
Sl.3.11 Prikaz rasporeda senzora

Ruta za vozila jedan i dva je identična kao u prošlom primjeru, isto kao i raspored detektora i usmjerivača. Razlika između dvije navedene simulacije je ta što sada vozilo dva prvo šalje zahtjev detektorima da očitaju da li se vozilo 1 nalazi u njihovom radijusu detekcije. Nakon toga detektor šalje nazad potvrdu.

4. GRAFIČKI PRIKAZ REZULTATA

Nakon provedene četiri simulacije, analizirat će se njihovi rezultati. Dvije simulacije su izvedene u programu Veins, u OMNeT++ okruženju, a dvije simulacije su izvedene u CupCarbon simulatoru. Simulacije su paralelno izvedene.

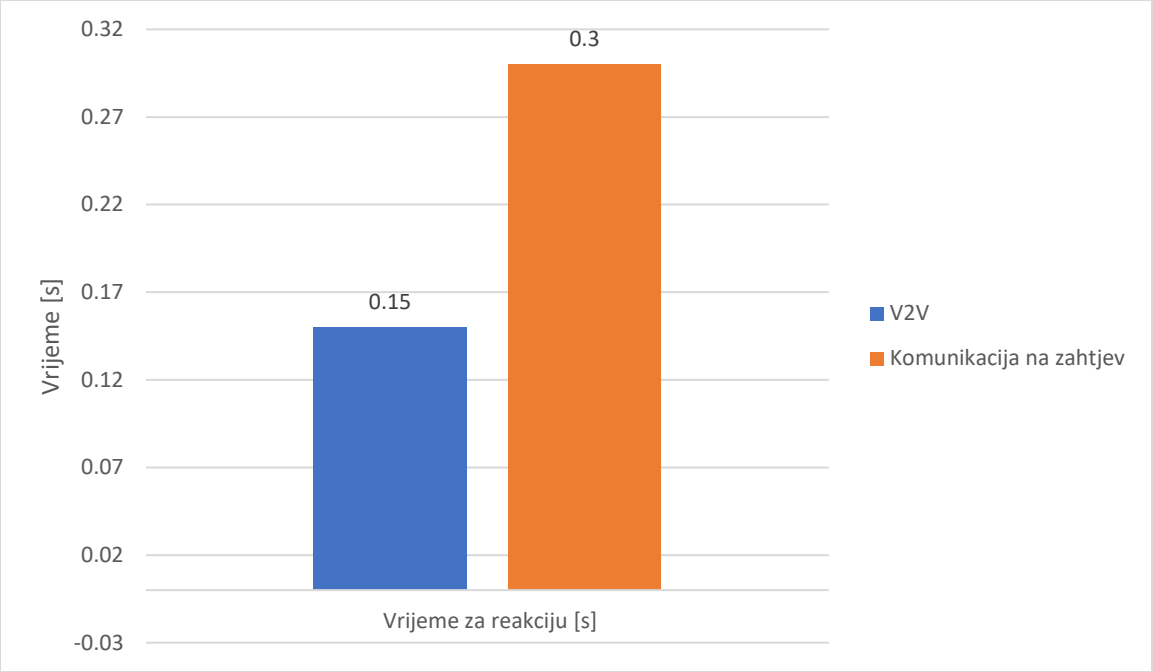
4.1 Usporedba V2V i AODV u Veinsu



Sl.4.1 Usporedba dvije vrste komunikacija u Veinsu

U programskom okruženju Veins su izvedene dvije skoro identične simulacije, međutim korišten je drugi tip komunikacije među vozilima. Prvi primjer je rađen tako da se vozila kreću istom rutom te nakon nekog vremena vozilo stane simulirajući automobilsku nesreću. Drugi primjer je sličan tome samo drugo vozilo prvo šalje zahtjev prvome za stanju na cesti. Stoga ne čudi što je vrijeme potrebno za reakciju značajno veće kod druge simulacije, iako je bitno napomenuti da se i dalje radi o realnim vremenima slanja informacija, ispod jedne sekunde.

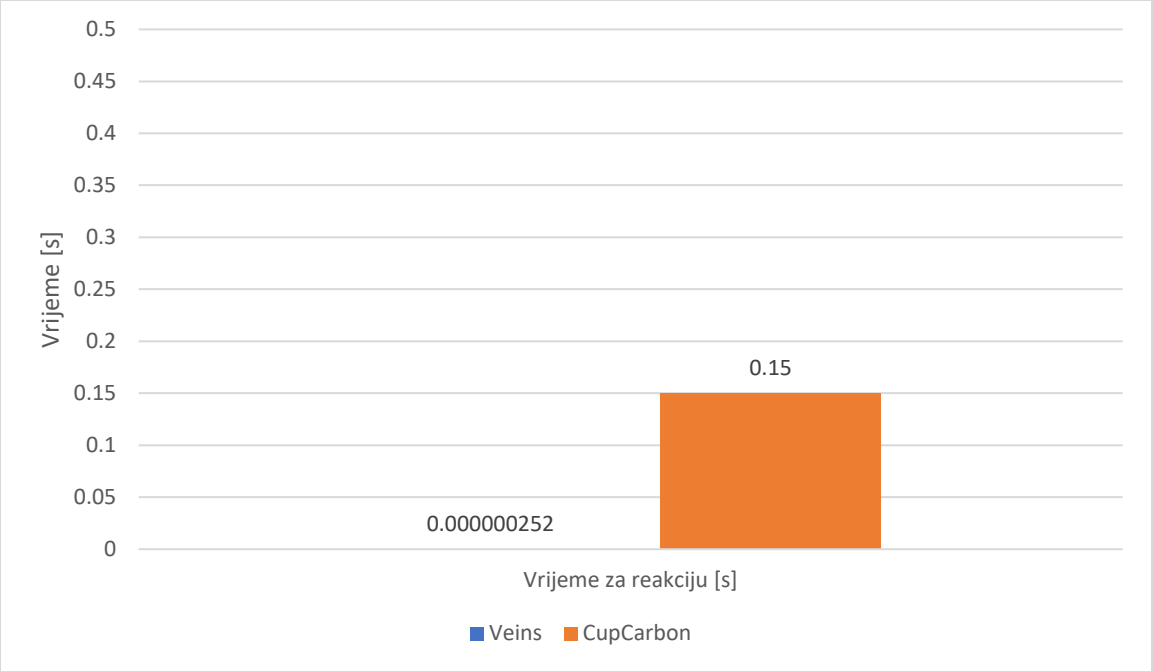
4.2 Usporedba V2V i komunikacije na zahtjev u CupCarbonu



Sl.4.2 Usporedba dvije vrste komunikacija u CupCarbonu

U CupCarbon simulatoru su provedene dvije simulacije koje su imale cilj biti identične kao u Veinsu, no radi ograničenja programa, izvedene su u dosta jednostavnoj varijanti. Korišteno je više cestovne infrastrukture kako bi se podaci mogli prenjeti do drugog vozila, što nije efikasno niti daje pravu sliku za implementaciju u autonomna vozila. No rezultati vidljivi s grafa pokazuju da se izravna komunikacija između vozila kao i u drugom programu izvodi s manje čekanja. Također je primjetno da su vremena do 'reakcije' dosta veća nego kada se koristi VANET mreža. Kod CupCarbona je zbog jednostavnosti programa, slanje paketa podešeno na 50 ms, tako da ovisi o broju senzora kroz koji prolazi. U našem primjeru prolazi kroz 3 senzora u 1. primjeru a 6 u drugom jer prvo šalje zahtjev pa se odgovor obrnutim putem vraća nazad.

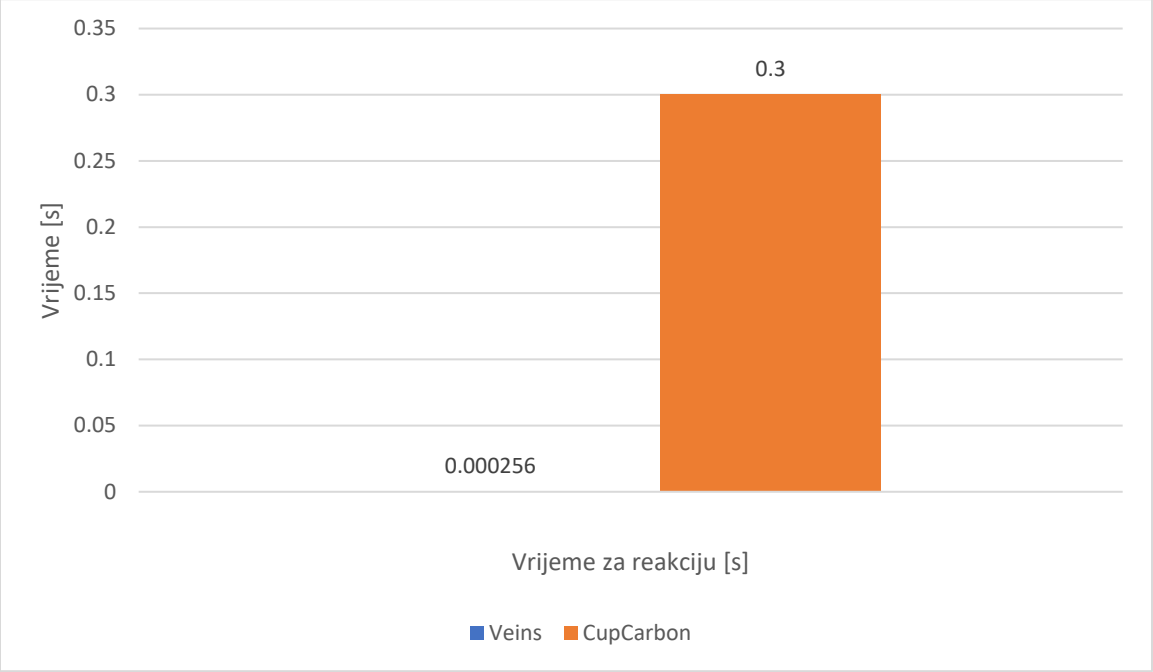
4.3 Usporedba V2V komunikacije u oba programa



Sl.4.3 Usporedba izravne komunikacije između dva programa

Ono što je zapravo i bila tema ovog diplomskog rada, a to je prikazati razliku između dva programa i dvije vrste komunikacije među mobilnim čvorovima se vidi na slici 4.3. Kada se radi o simulaciji nesreće i javljanju drugim vozilima o istoj, VANET mreža ima veliku prednost jer automatski generira poruku u realnom vremenu. Plavom bojom na grafu je označeno vrijeme potrebno za pristizanje poruke i ono je u mjernoj jedinici nanosekunda. Taj rezultat možda i nije realan, jer je očitanje ovisilo o ljudskoj pogrešci, međutim ono što je sigurno je to da kada se ne radi o VANET komunikaciji kao u CupCarbonu, vrijeme je veće. Budući da se radi o vozilima koja se kreću velikom brzinom, u takvim slučajevima je bitno da informacija stigne što prije.

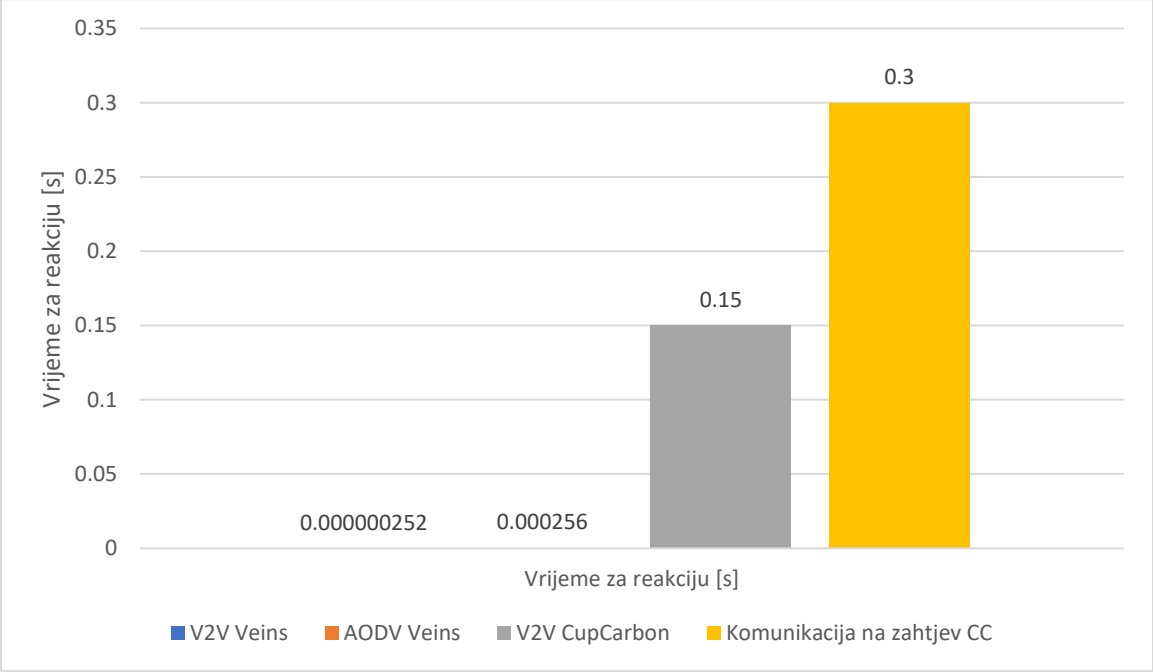
4.4 Usporedba komunikacije na zahtjev u oba programa



Sl.4.4 Usporedba drugog primjera u oba programa

Drugi primjer simulacije koji se izveo u oba programska okruženja je slanje zahtjeva drugog vozila prvome za određenu informaciju. Zbog nemogućnosti primjene i testiranja ad hoc on demand distance vector protokola kao u Veinsu, kod u CupCarbonu je postavljen tako da kroz iste puteve šalje zahtjev za informacijom i nazad pristiže odgovor. Zbog toga vrijeme potrebno za reakciju iznosi 30 ms, jer putem prolazi kroz 6 senzora. Isto kao i u prvom primjeru provedenom u oba programa i ovdje rezultati u CupCarbonu daju sporije rezultate, što još jednom potvrđuje da je VANET mreža dobar odabir kada brzina igra ulogu.

4.4 Prikaz svih rezultata



Sl. 4.5 Prikaz rezultata svih simulacija

Na grafu 4.5 su prikazani rezultati sve četiri provedene simulacije.Ono što je uočljivo na prvi pogled je to da su rezultati unutar Veinsa, tj. koristeći VANET mrežu puno brži za razliku od primjera kada ju ne koristimo. Također vidi se da izravna komunikacija među vozilima je otprilike zahtjeva duplo manje vremena nego kada se šalje zahtjev, bio to AODV ili običan upit jer se automatski generiraju poruke.

5. ZAKLJUČAK

Uspješna implementacija VANET mreže je jedan od bitnijih preduvjeta za korištenje autonomnih vozila. Kako postoji nekoliko tipova komunikacije unutar same mreže, potrebno je izvesti simulacije na nekoliko scenarija da se utvrdi koja komunikacija daje najbolje rezultate. U ovom radu su provedene simulacije da se utvrde razlike između komunikacije između vozila, i komunikacije koja se šalje na zahtjev korisnika. Simulacije su provedene u dva programa, Veinsu i CupCarbonu. Budući da Veins omogućuje simulacije s izravnom primjenom VANET mreže, dok je CupCarbon program s pojednostavljenom shemom i razmjenom podataka preko senzora, po rezultatima je vidljivo da je VANET mreža daleko brža i izvediva u realnom vremenu. Razlika između vremena slanja i primanja paketa je značajna. Drugi zaključak koji se nameće je to da između izravne komunikacije vozila i komunikacije na zahtjev, gdje korisnik traži željenu informaciju, izravna komunikacija je brža i izvodi se u realnom vremenu jer se poruke automatski generiraju. To je bitnije kod dojave o nekoj nesreći na cesti. S druge strane, kada korisnik traži neki upit o stanju na cesti, tada su informacije dostupne skoro duplo kasnije nego kada se automatski generiraju, međutim, dolazi do velike uštede resursa. VANET mreža je također isplativa uspoređujući je sa drugim vrstama komunikacije, jer je potrebno manje ulaganje u infrastrukturu, iako i ona zahtjeva veliku pokrivenost ceste.

6. LITERATURA

[1] Balen J., Učinkovito rasprostiranje poruka u mrežama vozila zasnovano na njihovom položaju, Osijek, 2014.

[2] Kasapović S., Banjanović_Mehmedović L., Sigurnosni uslovi i primjer aplikacije u mreži vozila, Tuzla, 2016.

[3] Marzak B., Abdelatif S., Derdour M., Ghoulami-Zine N., VANET: A novel service for predicting and disseminating vehicle traffic information

[4] Pattnaik O., Pattanayak B., Performance Analysis of MANET and VANET based on Throughput Parameter, Nalanda Institute of Technology

[5] Leovac M, Optimalno usmjeravanje mobilnih čvorova u bežičnim senzorskim mrežama, Osijek, 2019.

SAŽETAK

Tema ovog rada je opisati i analizirati rad VANET mreže. Prvi dio rada se sastoji od teorijskog dijela gdje je obrađena podjela VANET mreže na tri tipa komunikacije, vehicle to vehicle, vehicle to infrastructure i infrastructure to infrastructre komunikaciju. Također su opisani protokoli pomoću kojih se šalju paketi i poruke u samoj mreži. Prikazana je struktura beacon poruka, tj. kratkih poruka koje se šalju u mreži sa osnovnim informacijama o statusu vozila. Nakon toga u radu je objašnjen paket unutar VANET mreže i od kojih se dijelova sastoji. Sljedeći dio je arhitektura same mreže odnosno podjela na skupljanje informacija s prometnica, VANET oblak u kojem se analiziraju i obrađuju skupljeni podaci i infrastruktura koja služi kao veza između ta dva dijela. Zadnji dio teorijskog dijela je objašnjenje razmjene podataka. Drugi dio rada je praktični dio i izvedva četiri simulacije u dva programska okruženja, Veinsu i CupCarbonu. Provedene su dvije simulacije u oba programa te su međusobno analizirane i grafički prikazane, ovisno i vrsti komunikacije unutar samog programa, te razlike između istih simulacija u dva programa. Na kraju rada u zaključku su opisana zapažanja i zaključci.

Ključne riječi: Vehicular ad hoc network (VANET), V2V komunikacija, V2I komunikacija, mobilni čvorovi.

ABSTRACT

The topic of this thesis is to describe and analyze the work of the VANET network. The first part of the paper consists of a theoretical part where the division of the VANET network into three types of communication, vehicle to vehicle, vehicle to infrastructure and infrastructure to infrastructure communication is discussed. Protocols for sending packets and messages within the network are also described. The structure of beacon messages, short messages sent in the network with basic information about the status of the vehicle, is shown. After that, the paper explains the package within the VANET network and what parts it consists of. The next part is the architecture of the network itself, the division into the collection of information from roads, the VANET cloud in which the collected data is analyzed and processed, and the infrastructure that serves as a link between the two parts. The last part of the theoretical part is an explanation of data exchange. The second part of the paper is a practical part and performance of four simulations in two programming environments, Veins and CupCarbon. Two simulations were performed in both programs and were mutually analyzed and graphically presented, depending on the type of communication within the program, and the differences between the same simulations in the two programs. At the end of the paper, the observations and conclusions are described in the conclusion.

Key words: Vehicular ad hoc network (VANET), V2V communication, V2I communication, mobile nodes.

ŽIVOTOPIS

Mario Kovačević, rođen je 27.11.1995. godine u Slavonskom Brodu, gdje je završio "Osnovnu školu Antun Mihanović" te "Klasičnu gimnaziju fra Marijana Lanosovića s pravom javnosti". Tijekom pohađanja osnovne i srednje školu, bavio se izvannastavnim aktivnostima te je trenirao rukomet. Nakon srednje škole, 2014. godine upisuje preddiplomski studij na "Fakultetu elektrotehnike, računarstva i informacijskih tehnologija" u Osijeku. Nakon završene prve godine studija, odabire smjer Komunikacije i Informatika. Preddiplomski studij završava sa temom završnog rada: "Analiza RIP usmjerivačkog protokola pomoću Riverbed Modeler simulatora", te stječe zvanje sveučilišnog prvostupnika. Nastavlja sa školovanjem te upisuje diplomski studij, izborni blok Komunikacijske tehnologije.