



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY



UNIVERSITY OF GOTHENBURG

---

# Nya applikationer för kooperativ perception

Kandidatarbete vid Institutionen för Data- och Informationsteknik

Elias Niklund  
Emma Bodin  
Hugo Carlsson  
Isabella Hall  
Jesper Persson  
Jonathan Sunnerstam



KANDIDATARBETE 2025

# Nya applikationer för kooperativ perception

Elias Niklund  
Emma Bodin  
Hugo Carlsson  
Isabella Hall  
Jesper Persson  
Jonathan Sunnerstam



UNIVERSITY OF  
GOTHENBURG

---



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

Institution för Data- och Informationsteknik  
CHALMERS TEKNISKA HÖGSKOLA  
GÖTEBORGS UNIVERSITET  
Göteborg, Sverige 2025

Nya applikationer för kooperativ perception  
Elias Niklund Emma Bodin Hugo Carlsson Isabella Hall Jesper Persson Jonathan Sunnerstam

© Elias Niklund, Emma Bodin, Hugo Carlsson, Isabella Hall, Jesper Persson, Jonathan Sunnerstam 2025.

Handledare: Elad M. Schiller, Institution för Data- och Informationsteknik  
Examinatorer: Patrik Jansson och Arne Linde, Institution för Data- och Informationsteknik  
Rättande lärare: Roger Johansson, Institution för Data- och Informationsteknik

Kandidatarbete 2025  
Institution för Data- och Informationsteknik  
Chalmers Tekniska Högskola och Göteborgs Univeristet  
SE-412 96 Göteborg  
Telefon +46 31 772 1000

Satt med hjälp av L<sup>A</sup>T<sub>E</sub>X  
Göteborg, Sverige 2025

## Abstract

As connected and autonomous vehicles become more prevalent, cooperative perception is increasingly vital for ensuring traffic safety and efficiency in urban environments. This project investigates how cooperative perception can enhance intelligent transportation systems by reducing collision risks and improving traffic efficiency through real-time data exchange between vehicles and infrastructure.

The system is developed in a simulation framework based on SUMO called ms-van3t. Which also integrates decentralized communication, collision avoidance, and route planning. We introduce a new concept called ITS-fairy, which uses a Server Local Dynamic Map to support real-time data sharing and hazard detection. In addition, a safety mode based on Time-to-Collision and Space-to-Collision estimations provides proactive warnings to prevent crashes. Additionally, a centralized planner is introduced to reduce congestion by assigning routes based on global traffic conditions.

The implementations were evaluated through simulations of urban traffic scenarios such as intersections and roundabouts. The results demonstrate ensured safety in urban scenarios and more coordinated traffic behavior under high-density conditions. We conclude that the developed platform offers a robust foundation for continued research and development in cooperative vehicular systems.

Keywords: Cooperative Perception, Intelligent Transportation Systems, Autonomous Vehicles, S-LDM, Simulation

## Sammandrag

I takt med att uppkopplade och självkörande fordon blir allt vanligare blir samverkande perception allt viktigare för att säkerställa trafiksäkerhet och effektivitet i stadsmiljöer. Detta projekt undersöker hur kooperativ perception kan förbättra intelligenta transportsystem genom att minska kollisionsrisker och förbättra trafikeffektiviteten genom datautbyte i realtid mellan fordon och infrastruktur.

Systemet är utvecklat i ett simuleringsramverk baserat på SUMO som kallas ms-van3t. Detta integrerar även decentraliserad kommunikation, kollisionsundvikande och ruttplanering. Vi introducerar ett nytt koncept som kallas ITS-fairy, som använder en Server Local Dynamic Map för att stödja datadelning i realtid och riskdetektering. Dessutom ger ett säkerhetsläge baserat på uppskattningar av tiden och avståndet till kollision proaktiva varningar för att förhindra krascher. Vidare introduceras en centraliserad planerare för att minska trafikstockningar genom att tilldela rutter baserat på globala trafikförhållanden.

Implementeringarna utvärderades genom simuleringar av trafikscenarier i stadsmiljö, t.ex. korsningar och rondeller. Resultaten visar på ökad säkerhet i stadsmiljöer och ett mer samordnat trafikbeteende under förhållanden med hög trafikdensitet. Vi drar slutsatsen att den utvecklade plattformen erbjuder en robust grund för vidare forskning och utveckling inom kooperativa fordonssystem.

Nyckelord: Kooperativ Perception, Intelligenta Transportsystem, Autonoma fordon, S-LDM, Simulering

## Författarnas tack

Vi vill rikta ett varmt tack till våran handledare Elad Schiller, för hans stöd, expertis och vägledning under projektets gång.

Vi vill även tacka Lars Norén som hjälpte oss med införskaffandet och uppsättningen av den nya datorn till labbet. Samt rikta ett tack åt Yenan Wang för den hjälp och erfarenhet han bidrog med under uppstarten av simulatorn.

Slutligen vill vi tacka gruppen *Utveckling samt evaluering av lokalisering och kooperativa styrsystem* vars sällskap och delande av labbsal varit till stor nytta under arbetets gång.

Elias Niklund, Emma Bodin, Hugo Carlsson, Isabella Hall, Jesper Persson, Jonathan Sunnerstam, Göteborg, juni 2025

## Ordlista

- CAM** *Cooperative Awareness Message*. Meddelande som innehåller information om det sändande fordonet såsom position och hastighet.
- CPM** *Collective Perception Message*. Möjliggör för fordon att dela sensordata med varandra och därmed utöka sin uppfattning bortom direkt siktlinje.
- CPS** *Collective Perception System*. Ett ramverk som integrerar CPM-meddelanden med annan data från flera fordon för att förbättra den gemensamma uppfattningen och öka trafiksäkerheten.
- DENM** *Decentralized Environmental Notification Message*. Ett varningsmeddelande som används i intelligenta transportsystem för att varna andra fordon om faror eller oväntade händelser i trafiken.
- ITS** *Intelligent Transport Systems*. Integrering av informationsteknologi i transportsystem för att förbättra säkerhet, effektivitet och hållbarhet.
- LDM** *Local Dynamic Map*. En digital karta som i realtid hanterar både statisk och dynamisk information om trafikmiljön för att stödja säkrare och effektivare trafiksystem.
- MAPF** *Multi Agent Path Finding*. Ett problem där flera agenter ska hitta kollisionsfria vägar på en karta från start till mål.
- ms-van3t** *Multistack VANET framework for ns-3*. Kommunikationssimulator som möjliggör simulering av V2X-meddelanden i trafikmiljö.
- RSU** *Roadside Unit*. En ITS-station som behandlar Vehicle-to-Everything och Vehicle-to-Infrastructure kommunikation.
- S2C** *Space-to-Collision*. Är det minsta uppskattade avståndet som kommer uppstå mellan två fordon, under antagandet att de följer sina beräknade banor.
- S-LDM** *Server Local Dynamic Map*. En serverbaserad karta som samlar och delar information från fordon och sensorer för att stödja säker och effektiv trafik.
- SUMO** *Simulation of Urban Mobility*. En trafiksimulator som möjliggör modellering av verklighetstroga trafiksituationer.
- T2C** *Time-to-Collision*. Tiden det tar att nå S2C-avståndet mellan två fordon.
- TraCI** *Traffic Control Interface*. En API till SUMO som syftar till att interagera med simuleringen i realtid.
- V2X** *Vehicle-to-Everything*. Kommunikationssystem som möjliggör informationsutbyte mellan fordon och andra enheter såsom infrastruktur, fotgängare och nätverk.

# Innehåll

<b>Figurer</b>	<b>x</b>
<b>Tabeller</b>	<b>xii</b>
<b>1 Inledning</b>	<b>1</b>
1.1 Syfte . . . . .	1
1.2 Problembeskrivning . . . . .	2
1.3 Relaterat arbete . . . . .	2
1.4 Vårt bidrag . . . . .	3
1.5 Avgränsningar . . . . .	4
<b>2 Bakgrund</b>	<b>5</b>
2.1 Simulatorn . . . . .	5
2.1.1 SUMO - Simulation of Urban Mobility . . . . .	5
2.1.2 ms-van3t - Multistack VANET framework for ns-3 . . . . .	5
2.2 Virtuella trafikljus . . . . .	6
2.3 MAPF - Multi Agent Path Finding . . . . .	6
2.3.1 Lösning av MAPF-problemet . . . . .	6
2.3.2 C-CBS - Continous Conflict Based Search . . . . .	7
2.4 Kommunikation inom Vehicle to Everything . . . . .	8
2.5 ITS - Intelligent Transport System . . . . .	9
2.5.1 CPS - Collective Perception Service . . . . .	9
2.5.2 CPM - Collective Perception Message . . . . .	9
2.5.3 CAM - Cooperative Awareness Message . . . . .	9
2.5.4 LDM - Local Dynamic Map . . . . .	10
2.5.5 S-LDM - Server Local Dynamic Map . . . . .	10
<b>3 Metod</b>	<b>13</b>
3.1 ITS-fairy . . . . .	13
3.1.1 Funktionalitetslägen för ITS-Fairy . . . . .	13
3.1.2 Centraliserad och decentraliserad kollisionsundvikning . . . . .	15
3.1.3 Kostnadsanalys för centraliserad kooperativ kollisionsundvikning	15
3.1.4 Förklaring av pseudokod . . . . .	16
3.2 Centraliserat ingripandeläge . . . . .	18
3.2.1 Systemöversikt för centraliserade ingripandeläge . . . . .	18
3.2.2 Riskbedömning av Time-to-Collision (T2C) och Space-to-Collision (S2C) . . . . .	19

3.2.3	Vinkelbaserad kollisionsklassificering . . . . .	20
3.2.4	Kommunikation och perceptionsstandarder . . . . .	21
3.2.5	Mjukvaruarkitektur för kollisionsundvikande system . . . . .	21
3.2.6	Algoritmisk säkerhetslogik och pseudokod . . . . .	22
3.3	Kollektiv ruttplanering . . . . .	26
3.3.1	Hantering av integritet . . . . .	27
3.3.2	Prestanda av ruttplanering . . . . .	27
3.3.3	Kontroll av fordon . . . . .	28
3.3.4	Användning av Virtuella trafikljus . . . . .	28
3.3.5	Mjukvaruarkitektur för kollektivt ruttplanerande system . . . . .	30
<b>4</b>	<b>Implementering</b>	<b>33</b>
4.1	Installation av simulator . . . . .	33
4.2	Kommunikation mellan S-LDM och ms-van3t . . . . .	33
4.3	Centraliserade ingripandeläget . . . . .	33
4.4	Situationsmedvetna stödläget . . . . .	34
4.5	Kollektiva ruttplaneringssystemet . . . . .	34
<b>5</b>	<b>Utvärdering</b>	<b>37</b>
5.1	Undersökningsfrågor . . . . .	37
5.2	Utvärderingsmiljö . . . . .	37
5.3	Funktionalitet . . . . .	39
5.4	Utvärderingskriterier . . . . .	39
5.5	Experiment . . . . .	39
5.5.1	Centraliserade ingripandeläget . . . . .	39
5.5.2	Situationsmedvetna stödläget . . . . .	40
5.5.3	Kollektiva ruttplaneringssystemet . . . . .	40
<b>6</b>	<b>Resultat</b>	<b>43</b>
6.1	Centraliserade ingripandeläget . . . . .	43
6.2	Situationsmedvetna stödläget . . . . .	46
6.3	Kollektiva ruttplaneringssystemet . . . . .	47
<b>7</b>	<b>Diskussion och slutsats</b>	<b>51</b>
7.1	Diskussion . . . . .	51
7.2	Samhälleliga och etiska aspekter . . . . .	52
7.3	Framtida arbeten . . . . .	53
7.3.1	Centraliserade ingripandeläget . . . . .	53
7.3.2	Situationsmedvetna stödläget . . . . .	53
7.3.3	Kollektiva ruttplaneringssystemet . . . . .	54
7.4	Slutsats . . . . .	54
	<b>Litteraturförteckning</b>	<b>57</b>

# Figurer

2.1	Fem typer av konflikter: a) Nodkonflikt b) Kantkonflikt, c) Flera agenter rör sig i en cirkel, d) Två agenter byter plats och e) Agenter följer varandra. . . . .	7
2.2	Forden från ms-van3t visualiseras i S-LDM via ett webbgränssnitt. . .	11
3.1	Exempel där den röda bilen blir medveten om faran i en dold kurva med hjälp av ITS-fairy. . . . .	13
3.2	Lyckat säkerhetsscenario. . . . .	16
3.3	S-LDM assisterat scenario. . . . .	16
3.4	Fyrvägskorsning med riktningar. . . . .	17
3.5	Konfliktmatris där 1:orna motsvarar en potentiell konflikt. . . . .	17
3.6	Flödet av information och kommunikationsmönster. . . . .	19
3.7	Om ett fordon's T2C- och S2C-tröskelvärden underskrids samtidigt ( $T2C < T2C_{th}$ , $S2C < S2C_{th}$ ) identifieras en kollisionsrisk och ett DENM skickas. . . . .	20
3.8	Kollisionsscenarioer baserat på vinkelskillnaden mellan fordon ( $\Delta\theta$ ). . .	20
3.9	Systemets arkitektur. (1) fordonets attribut, (2) lagring av fordon's-data i LDM, (3) kollisiondetektering och generering av DENM-meddelanden, (4–6) kommunikationsflöde mellan S-LDM och ms-van3t via AMQP, (7–10) hantering, avkodning och vidarebefordran av meddelanden, (11) SUMO-gränssnitt för justering av fordon's attribut, samt (12) uppdatering av fordon's beteende i trafiksimulatoren. . . . .	22
3.10	Systemets arkitektur. (1) fordonets attribut, (2) lagring av fordon's-data i LDM, (3) schemaläggare som förutspår hur fordonen kommer färdas, (4–6) kommunikationsflöde mellan S-LDM och ms-van3t via AMQP, (7–10) hantering, avkodning och vidarebefordran av meddelanden, (11) SUMO-gränssnitt för justering av fordon's attribut, (12) fordon mottar kommandon baserat på schemat, samt (13) hämtar vägar, korsningar och annan infrastruktur ur kartan. . . . .	31
5.1	De nio kartorna som används har fått följande namn: a) Manhattan, b) Solängen, c) Gaffel, d) Häger, e) Litet spindelnät, f) Van Gogh, g) Picasso, h) Stort spindelnät och i) Schackbräde. . . . .	38
6.1	Graf trevägskorsning. Experiment för T2C-värden för det centraliserade ingripande läget i ITS-fairy, när SUMOs kollisionundvikande system utsetts för störningsförhållanden. . . . .	43

6.2	Graf trevägskorsning. Experiment för hur uppdateringsfrekvensen för fordonsinformation påverkar systemets säkerhet. . . . .	44
6.3	Graf fyrvägskorsning. Experiment för T2C-värden för det centraliserade ingripande läget i ITS-fairy, när SUMOs kollisionsundvikande system utsetts för störningsförhållanden. . . . .	45
6.4	Graf fyrvägskorsning. Experiment för hur uppdateringsfrekvensen för fordonsinformation påverkar systemets säkerhet. . . . .	45
6.5	Graf rondell. Experiment för T2C-värden för det centraliserade ingripande läget i ITS-fairy, när SUMOs kollisionsundvikande system utsetts för störningsförhållanden. . . . .	46
6.6	Graf rondell. Experiment för hur uppdateringsfrekvensen för fordonsinformation påverkar systemets säkerhet. . . . .	46
6.7	Visualisering av resultat med och utan AMQP. . . . .	47
6.8	Minsta latensen i olika scenarier . . . . .	48
6.9	Maximala latensen i olika scenarier . . . . .	48
6.10	Genomsnittliga latensen för olika scenarier . . . . .	49
6.11	Variansen i latensen för olika scenarier . . . . .	49

# Tabeller

6.1 Tabellen visar resultat från fem test med och utan AMQP. . . . . 47



# 1

## Inledning

Varje år dör omkring 1,19 miljoner människor världen över [1], och bara i USA besökte nästan 2,1 miljoner människor en akutmottagning efter en trafikrelaterad olycka. Detta är inte bara en förlust av liv och en försämrad livskvalitet utan även en kostnad på cirka 430 miljarder amerikanska dollar i årliga sjukvårdsrelaterade kostnader [2]. Utöver olyckorna står transportsektorn världen över för en femtedel av de globala koldioxidutsläppen, varav endast vägtransport utgör cirka 75% av utsläppen [3].

Samtidigt sker en snabb utveckling av automatiserade körsystem och förarassistans, vid år 2030 uppskattas nästan alla sålda bilar vara uppkopplade [4]. Den globala marknaden för bilar med automatiska körsystem låg dessutom på 41 miljarder amerikanska dollar år 2024 och förväntas öka till cirka 115 miljarder år 2029 [5].

En potentiell lösning till att minska olyckor och utsläpp är just uppkopplade fordon. När fordon i realtid kommunicerar med varandra och sin omgivning möjliggörs en säkrare trafik då fordonen kan uppfatta och därmed agera på faror i god tid [6]. Informationen kan även användas för att estimerast hastigheten inför trafikljus, korsningar samt vilka vägval fordonet tar till dess destination vilket bidrar till en minskad bränsleförbrukning [7].

Projektet kommer därför bygga en grund för system hos självkörande bilar i en trafiksimulator med fokus på kommunikation, kollisionsförebyggande funktioner och kollektiv ruttplanering.

### 1.1 Syfte

Syftet är att lära sig hur ett större projektarbete i grupp läggs upp och vikten av samarbete för att skapa en god arbetsmiljö som leder arbetet framåt. Genom att utveckla ett koncept inom ett aktuellt växande ämne ges möjlighet till lärorik och kreativ problemlösning. Projektet ger även en förberedelse inför arbetslivet på ett ingenjörsmässigt sätt.

Utöver detta syftar projektet till att utveckla nya och effektiva tillämpningar som hanterar olika trafiksituationer i en simulerad miljö. Detta genom att förbättra kommunikation- och informationsdelning mellan fordon och infrastruktur i realtid för att garantera säkerhet i trafiken och bidra till att förbättra trafikflödet. Dessa

lösningar kommer att ge stöd för automatiserade körsystem och förarassistans.

## 1.2 Problembeskrivning

Moderna fordon är idag ofta utrustade med assisterande hjälpmedel som syftar till att bidra till ett säkert framförande av fordonet. Farthållare, sensorer och kameror är några av de vanligaste teknikerna, men det är bara ett steg mot automatiserade och uppkopplade fordon. Idag existerar däremot ITS-system som med hög noggrannhet och precision kan skicka information mellan fordon och infrastruktur [8]. Det kommer fortfarande dröja många år innan autonoma fordon integreras i trafiken, mycket forskning och pengar har även investerats i detta globala projekt. Inom 5 år förväntas 95% av alla nyproducerade fordon vara uppkopplade och utrustade med avancerade sensorer som erbjuder information om trafiksituationer i realtid. Många av de fördelar som autonoma fordon erbjuder kan alltså inom en snar framtid realiseras med teknik som förväntas bli vanlig inom några få år.

Utmaningen ligger i att utnyttja fordonens realtidsinformation och därför kommer projektet behandla utvecklingen av ett sådant system. Tanken är att informationen tillsammans med en digital karta möjliggör centraliserade fordonsbeslut i syfte att garantera säkerhet och planera trafiken. Detta resulterar i ett nytt koncept kallat ITS-fairy som möjliggör informationsdelning samt ett kollisionsundvikande- och kollektivt ruttplanerande system.

För att undersöka detta ställs följande frågor:

- Kan ITS-fairy bidra till en mer fullständig och enhetlig uppfattning av trafiksituationen?
- Kan ITS-fairy vidta några effektiva säkerhetsåtgärder?
- Vad är avvägningen mellan kommunikationskostnaden och de ökade säkerhetsmarginalerna?
- Kan ett system som syftar till att effektivisera trafikflöden implementeras?

## 1.3 Relaterat arbete

Detta arbete bygger på en trafiksimulator som kallas SUMO. SUMO står för "Simulation of Urban MObility"[9] och med hjälp av denna har olika trafiksituationer kunnat simuleras för att testa våra implementationer. Ett par tillägg till SUMO har också nyttjats och dessa heter ms-van3t som står för "Multistack VANET framework for ns-3"[10] och S-LDM som står för Server Local Dynamic Map [11]. ms-van3t är en utbyggnad av SUMO som simulerar och utvärderar kommunikation mellan fordon och dess omgivning [10]. Även den centraliserade databasen, S-LDM, ger möjlighet att ta emot och skicka meddelanden mellan fordon och omgivningen [11].

Under planering av arbetet och utveckling av psuedokoden inom ITS-fairy har rapporten skriven av Jame, Rapelli och Casetti [12] använts tillsammans med rapporten skriven av Casimiro, Ekenstedt och Schiller [13].

Det kollisionsundvikande systemet är baserat på pseudokoden som återfinns i doktorsavhandlingen av Marco Malinverno [14] (sidan 162, Appendix B), och i doktorsavhandlingen av Giuseppe Avino [15] (sidan 38, Algoritm 1). Den del av arbetet med fokus på ruttplanering har utöver det som nämnts ovan även byggts på en konfliktbaserad sökalgoritm C-CBS (Continuous Conflict Based Search) som letar den snabbaste konfliktfria vägarna för ett grupp av agenter som ska ta sig från deras start- och slutpunkt [16].

Vårt arbete är också inspirerat av tidigare forskning inom kooperativ perception och intelligenta transportsystem. Särskilt relevanta är Savic et al. [17] som presenterar algoritmer för kollisionsundvikande vid vägkorsningar även vid kommunikationsfel, samt Petig et al. [18] som studerar körfältsbyten med säkerhet i fokus. Morales-Ponce et al. [19, 20] adresserar hur asymmetrisk paketförlust kan påverka samarbete mellan fordon.

Vidare erbjuder Casetti et al. [21] en aktuell översikt över AI/ML-baserade tjänster för 6G-anslutna autonoma fordon, vilket breddar förståelsen för hur framtida kommunikationsinfrastruktur kan användas i liknande system. Dessa verk förstärker vår förståelse av systemens robusthet, informationsutbyte, och realtidsbeslutstagande.

Slutligen anknyter även vår användning av S-LDM och lokala dynamiska kartor till tidigare arbete inom Gullivertestbädden [22, 23], som även betonar säkerhetsarkitekturer för samverkande fordonssystem.

## 1.4 Vårt bidrag

Vi studerar i detta arbete ett aktuellt problem, att använda redan distribuerad teknik för att garantera trafiksäkerheten och trafikflödet. Detta genom en utbyggnad i verkliga och befintliga system. Samtidigt som flera förväntade fördelar av automatiserade körsystem fås, är driftsättningstiden ännu inte känd. Till skillnad från idag, där fordonstillverkarna satsar miljardbelopp på autonoma fordon som först väntas ge ut resultat inom 15 år [5][24]. Visar vi att man kan leverera betydelsefulla resultat redan här och nu.

Det kollisionsundvikande systemet i ITS-fairy höjer säkerheten i stadstrafik avsevärt jämfört med simulatorn SUMOs inbyggda system under störningsförhållanden. Vi stoppar fordonen från en kollision i 100% av fallen, jämfört med 73% av fallen för SUMOs system. Därmed gav vi i genomsnitt 0.7 sekunder extra tidsmarginal för inbromsningen av fordon vid en kollisionsrisk. I tre- och fyrvägs korsningar ökade tidsmarginalen med cirka 0.4 sekunder och i rondeller med drygt 1.3 sekunder.

Gällande ruttplanering och trafikflöde visar vi både på möjligheter och begränsningar. Beroende på kartans utseende är det inte alltid säkert att en betydande förbättring av trafikflödet uppnås. Däremot visar våra undersökningar att den totala latensen förbättras i 7 av 9 scenarier när de koordinerade rutterna jämförs med de

okoordinerade. När vi analyserar stadsliknande nätverk finns det en klar fördel med att använda vårt förslag, där våra mätningar visar att den totala restiden kan minska med över 35%.

### 1.5 Avgränsningar

Detta arbete har endast fokuserat på vissa specifika kartor och testfall såsom trafiksituationer med fokus på korsningar och rondeller. Dessa kan se olika ut och ha olika komplexitet i den verkliga världen. Därför har arbetet begränsats till att endast simulera en eller några typer av korsningar och rondeller.

Eftersom arbetet baserats på simuleringar har det inte tagits någon hänsyn till olika vägförhållanden såsom väder, underlag och liknande. Spontana händelser där den mänskliga faktorn påverkar en trafiksituation, såsom farliga beslut eller att föraren somnar, går inte heller att ta hänsyn till.

# 2

## Bakgrund

Projektet har utförts i en simulerad labbmiljö för att analysera fordonsautomation och intelligenta transportsystem. För en djupare förståelse av arbetet presenteras i detta kapitel värdefulla begrepp och koncept som berörs i rapporten.

### 2.1 Simulatore

Simulatore som används i projektet är uppbyggd av flera komponenter, vilket är SUMO, ms-van3t och S-LDM. Kärnan av simulatore består av SUMO som skapades av German Aerospace Center [9]. Forskare från tekniska högskolan Politecnico di Torino har sedan utökat funktionaliteten hos SUMO med ramverket ms-van3t. Detta ramverk möjliggör kommunikation mellan fordon genom att integrera nätverksimulatore ns-3 i SUMO [10]. Forskarna har även vidareutvecklat funktionalitet till ms-van3t i form av S-LDM [11]. Simulatore möjliggör simulering av olika väg-scenarier där fordonen kan kommunicera med varandra över etablerade kommunikationstekniker såsom IEEE 802.11 och LTE.

#### 2.1.1 SUMO - Simulation of Urban Mobility

SUMO kan simulera enskilda fordon och fotgängare samt modellera hur dessa interagerar med varandra i en trafikmiljö. Miljön skapas antingen manuellt eller så används en existerande karta över ett geografiskt område. Eftersom simulatore har öppen källkod lämpas den bra för integrering med andra verktyg såsom nätverkssimulatore ns-3. Genom denna integrering kan även internet-trafik inom SUMO simuleras och detta är grunden som ms-van3t är byggd på.

#### 2.1.2 ms-van3t - Multistack VANET framework for ns-3

ms-van3t möjliggör internet-kommunikation genom att integrera nätverks-simulatore ns-3 i SUMO. ms-van3t är ett verktyg för att simulera och utvärdera fordons kommunikation, Vehicle to Everything (V2X), och erbjuder en rad olika funktioner som syftar till att återskapa en verklig miljö för fordon.

ms-van3t stödjer även meddelandetyper specificerade av European Telecommunications Standards Institute (ETSI). Dessa meddelandetyper är Cooperative Awareness Message (CAM), Decentralized Environmental Notification Messages (DENM) och Infrastructure to Vehicle Information Message (IVIM). Meddelandetyperna syftar till

kommunikationen mellan fordon och annan trafikrelaterad infrastruktur. ms-van3t möjliggör dessutom användningen av förinspelad GNSS-data, vilket är data från en GPS-mottagare som på förhand spelats in. Detta gör det möjligt att återskapa ett riktigt scenario i simulatorm [10].

## 2.2 Virtuella trafikljus

Virtuella trafikljus är en metod för att tvinga ett fordon att stanna i en korsning. Detta uppnås genom att en signal meddelar varje fordon om det är tillåtet att köra eller inte. Men till skillnad från fysiska trafikljus, skickas denna signalen direkt till fordonet över exempelvis Wi-Fi standarden 802.11p [25]. Den trådlösa förmedlingen av trafikljus möjliggör att varje korsning i praktiken kan ha ett trafikljus, vilket kan förkorta den genomsnittliga restiden avsevärt. Eftersom trafikljusen har en övergripande information om de fordon som förväntas nyttja korsningen kan dessa vara mer effektiva än fysiska [26].

## 2.3 MAPF - Multi Agent Path Finding

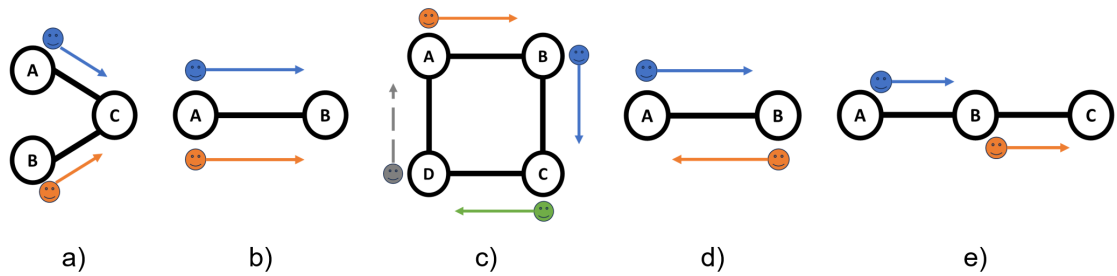
MAPF är ett problem vars lösning syftar till att hitta konfliktfria vägar för flera agenter. Genom kartor bestående av noder och kanter beskrivs miljön som agenterna navigerar i [27]. Alla agenter ska alltså samtidigt kunna följa sina vägar utan att kollidera med någon annan, vilket har ett brett användningsområde inom exempelvis robotteknik, automatiserade fordon och lager [28].

### 2.3.1 Lösning av MAPF-problemet

Lösningen av ett klassiskt MAPF-problem innehåller oftast [27]:

- **Karta utan riktning:** En karta kan beskrivas av  $G = (V, E)$  som är en uppsättning av noder ( $V$ ) och kanter ( $E$ ) [29].
- **Agenter:** Där  $A = \{1, 2, \dots, k\}$  beskriver hur många agenter som finns i lösningen.
- **Agenternas startposition s:** Vilken nod ( $V$ ) varje agent ( $A$ ) startar ifrån.
- **Agenternas slutdestination t:** Vilken nod ( $V$ ) varje agent ( $A$ ) ska till.

För att lösa problemet delas tiden upp i diskreta steg där agenterna vid varje tidssteg utför en rörelse, vilket betyder att de antingen står stilla eller rör sig mot nästa nod. Varje agent utför flera rörelser i rad för att nå sitt mål och tillsammans bildar rörelserna en individuell plan för just den agenten. Planen för agenten  $i$  kan skrivas som  $\pi_i = \{a_1, a_2, \dots, a_n\}$  där  $a_1, a_2, \dots, a_n$  är alla rörelser som agenten utför mellan noderna  $s(i)$  och  $t(i)$ . MAPF-problemet är löst när det finns en individuell kollisionsfri plan för varje agent.



**Figur 2.1:** Fem typer av konflikter: a) Nodkonflikt b) Kantkonflikt, c) Flera agenter rör sig i en cirkel, d) Två agenter byter plats och e) Agenter följer varandra.

Utöver agenternas planer får de inte heller kollidera med varandra under något tidssteg för att problemet ska räknas som löst [28]. Kollisionerna kallas för konflikter och det finns fem olika som kan ses i Figur 2.1 och följer:

- **a) Nodkonflikt:** Två agenter rör sig mot samma nod samtidigt och kan beskrivas av  $\pi_i(x) = \pi_j(x)$ .
- **b) Kantkonflikt:** Två agenter rör sig på samma kant i samma riktning samtidigt. Detta kan matematiskt beskrivas av  $\pi_i(x) = \pi_j(x)$ ,  $\pi_i(x+1) = \pi_j(x+1)$ . Om en nodkonflikt inte är tillåten kan en kantkonflikt aldrig uppstå.
- **c) Flera agenter rör sig i en cirkel:** Agenterna flyttar sig så att varje agent rör sig mot en nod som steget innan var upptagen av en annan agent. Detta kan inte ske om fallet där en agent följer en annan inte är tillåten.
- **d) Två agenter byter plats:** Konflikt där två agenter rör sig över samma kant samtidigt i olika riktningar och därmed byter plats med varandra. Kan beskrivas av  $\pi_j[x+1] = \pi_i[x]$  och  $\pi_i[x+1] = \pi_j[x]$
- **e) Agenter följer varandra:** Konflikt när två agenter följer varandra, den ena agenten tar alltså över en nod som tidssteget innan var upptagen av en annan agent. Detta beskrivs med  $\pi_i(x+1) = \pi_j(x)$ .

Det finns oftast flera lösningar till samma MAPF-problem och det finns olika sökalgoritmer för att lösa dessa. Beroende på hur algoritmerna utförs kan olika konflikter tillåtas i lösningen. Vanligast är att inga nod- eller kantkonflikter är tillåtna, men exakt vilka konflikter som används bestäms av algoritmen.

### 2.3.2 C-CBS - Continuous Conflict Based Search

C-CBS är en typ av MAPF-algoritm som utökar funktionaliteten till Conflict-Based Search (CBS). CBS är en metod som först planerar individuella vägar för varje agent utan hänsyn till potentiella konflikter. Därefter genomförs en sökning efter konflikter mellan de olika vägarna. Om en konflikt identifieras, exempelvis om två agenter befinner sig på samma nod vid samma tidpunkt, delas problemet upp i två nya delproblem. För varje delproblem införs en begränsning som utesluter konflikten för en av agenterna. ICTS delar upp sökningen i två nivåer. Först konstrueras ett

kostnadsträd där varje nod representerar en vektor av individuella kostnader för alla agenter. Därefter utförs en sammansatt sökning som försöker hitta vägar som överensstämmer med dessa kostnader och samtidigt undviker konflikter. Constraint Programming använder de olika begränsningarna och försöker hitta en lösning där alla begränsningar uppfylls. I detta projekt används en modifierad version av CBS som kallas Continuous Conflict-Based Search (C-CBS). Här genomförs sökningen i kontinuerlig tid istället för diskreta tidssteg. Algoritmen är utvecklad av Anton Andreychuk, Konstantin Yakovlev, Dor Atzmon och Roni Stern och lämpar sig väl för fall då agenter rör sig kontinuerligt istället för i diskreta steg [16].

C-CBS använder även algoritmen Safe interval path planning (SIPP) som kollar att agenten har ett säkert avstånd runt sig och ytterligare en algoritm som beräknar planen för en agent i kontinuerlig tid. Det som skiljer C-CBS från andra MAPF-algoritmer är att agenternas planer beräknas för kontinuerlig tid och konflikter identifieras genom att undersöka om två agents geometri överlappar vid ett tidssteg.

Likt CBS börjar algoritmen med att hitta vägar för alla agenter var för sig. Sedan kollar algoritmen efter om det finns några kollisioner eller begränsningar för någon agent. Sökningar sker i lågnivå och högnivå. Lågnivåsökningen letar efter vägar för agenter med hänsyn till om de har några begränsningar. Högnivåsökningen letar efter begränsningar för agenten med hjälp av Constraint Programming.

## 2.4 Kommunikation inom Vehicle to Everything

Vehicle to Everything (V2X) innefattar olika kommunikationstyper som möjliggör trådlös dataöverföring mellan fordon och omgivningen, vilket inkluderar andra fordon, fotgängare, väginfrastruktur och nätverk. Genom realtidsutbyte av information kan V2X bidra till ökad trafiksäkerhet, bättre trafikflödeshantering och en energieffektivare transport [30].

Kommunikationstyperna är:

- **Vehicle to Vehicle (V2V):** Direkt kommunikation mellan fordon för att dela information om hastighet, position och färdriktning. Detta möjliggör exempelvis varningssystem för att förebygga olyckor.
- **Vehicle to Infrastructure (V2I):** Kommunikation mellan fordon och väginfrastruktur såsom trafikljus och vägskyltar, som i sin tur effektiviserar trafikflödet och minskar trängsel.
- **Vehicle to Pedestrian (V2P):** System som varnar både förare och fotgängare om potentiella faror genom att dela positionsdata, vilket garanterar säkerheten för oskyddade trafikanter.
- **Vehicle to Network (V2N):** Användning av mobilnät för att ansluta fordon till molntjänster, som ger realtidsuppdateringar om trafikförhållanden, väder och andra relevanta data. Kommunikation mellan fordon och nätverk via 4G, LTE och 5G, vilket möjliggör realtidsuppdateringar om trafik, väder och andra relevanta data genom molntjänster och nätverksbaserade system.

Implementeringen av V2X-systemet förväntas ha en stor betydelse på trafiksäkerheten. Enligt USA:s National Highway Traffic Safety Administration (NHTSA) kan ett fullt implementerat V2V-system minska trafikolyckor med 13% – 18% per år [31]. Dessa system möjliggör ett snabbt och effektivt informationsutbyte mellan fordon och omgivningen, vilket gör det till en central del av framtidens transportsystem.

## 2.5 ITS - Intelligent Transport System

ITS är ett samlingsnamn för användningen av olika IT-lösningar inom transportsektorn. Genom att tillämpa informations- och kommunikationsteknik på vägtransporter skapas en smartare transportlösning som omfattar infrastruktur, fordon, trafikledning och mobilitetshantering. Målet är att garantera en säkrare, effektivare och mer hållbar transport, vilket regleras inom EU genom ITS-direktivet 2010/40/EU [32]. En central komponent inom ITS är Intelligent Transport Systems Stations (ITS-S), vilket avser enheter eller noder som deltar i systemet. ITS-S utgörs av fordon, Roadside Units (RSU) eller annan infrastruktur som kommunicerar och utbyter information med andra enheter för att förbättra trafikflöde och trafiksäkerhet [33].

### 2.5.1 CPS - Collective Perception Service

CPS möjliggör att flera ITS-stationer kan dela information om andra trafikanter och hinder som upptäckts med lokala sensorer som exempelvis radar och kameror. Genom att varje ITS-station bidrar med data om de detekterade objekten, oavsett om trafikanterna är anslutna eller inte, förbättras den individuella kunskapsbasen om omgivningen enligt ETSI TR 103 562 [34]. CPS kommunicerar med olika applikationer såsom LDM med Collective Perception Messages (CPM) [35].

### 2.5.2 CPM - Collective Perception Message

CPM är meddelanden som möjliggör för fordon och infrastruktur att dela information om objekt som upptäckts med dess sensorer, vilket skickas och tas emot av CPS. Genom datautbytet mellan fotgängare, hinder eller andra fordon kan mottagande enheter utvidga sin omgivningsbild utöver vad deras egna sensorer ser. Ett CPM innehåller flera informationskategorier, såsom uppgifter om sändarens position och rörelseriktning, detaljer om den använda sensorutrustningen samt en lista över de objekt som har detekterats. Detta inkluderar position, hastighet och dimensioner. Trafiksäkerhetsstandard är ETSI TS 103 324, som säkerställer att meddelandena är kompatibla mellan olika tillverkare och system. Detta är centralt för att möjliggöra en robust och pålitlig kooperativ perception i intelligenta transportsystem [35].

### 2.5.3 CAM - Cooperative Awareness Message

CAM är meddelanden som skickas periodiskt av fordon för att dela position, hastighet, riktning och andra relevanta fordonsparametrar. Dessa meddelanden skickas till andra ITS-stationer som delar det vidare för att bygga upp en gemensam situationsbild av trafiken. Skillnaden mellan CAM och CPM är att CAM-meddelanden innehåller

information om det aktuella fordonets status och attribut, medan CPM-meddelanden möjliggör delning av objekt som upptäckts av fordonets sensorer [36].

### 2.5.4 LDM - Local Dynamic Map

LDM är en karta som hanterar både statisk och dynamisk information om trafikmiljön i realtid. Genom att kombinera statisk och dynamisk information kan LDM hjälpa fordon och trafiksystem att fatta bättre beslut. Detta är avgörande för att garantera säkerheten i trafiken och förbättra trafikflödet [37]. LDM är uppbyggd i fyra nivåer enligt European Telecommunications Standards Institute (ETSI) [38], vilket illustreras i listan nedan.

- **Nivå 1 – Permanent statisk:** Det första lagret innehåller framför allt geografiska kartor med vägar och korsningar.
- **Nivå 2 – Övergående statisk:** Det andra lagret utökar det statiska lagret med bland annat trafikattribut, ITS-stationer och data som omfattar körfält.
- **Nivå 3 – Övergående dynamisk:** Det tredje lagret innehåller temporär regional information, såsom väder- och trafikförhållanden.
- **Nivå 4 – Starkt dynamisk:** Det fjärde lagret innehåller snabbt skiftande dynamisk information, såsom V2X-meddelanden med GPS-positioner för fordon.

### 2.5.5 S-LDM - Server Local Dynamic Map

S-LDM är en serverbaserad karta som hanterar och organiserar realtidsdata från fordon och andra ITS-stationer. Varje fordon delar löpande sina sensordata i form av CPM och CAM. Den insamlade datan används för att skapa en individuell LDM i varje fordon, som är uppbyggd i flera nivåer beroende på datans kategorisering. Genom att samla in och kombinera flera LDM:er tillsammans med information från andra ITS-stationer, exempelvis RSU:er (Roadside Units), kan en gemensam och mer heltäckande karta skapas på servernivå, en S-LDM. Denna serverkarta möjliggör effektivare informationsutbyte mellan fordon och stöder avancerade funktioner som kollisionsundvikande, trafikstyrning och kommunikation via V2X-teknik.

Genom en 5G-baserad S-LDM med syfte att förbättra kollektiv uppfattning av trafiken. S-LDM använder insamlad data från ms-van3t för att visualisera informationen på en karta i ett webbgränssnitt som presenterar fordonens positioner och attribut i realtid, detta visas i Figur 2.2. Systemet stödjer endast CAM-meddelanden och för att simulera CPM-meddelanden används det virtuella CAM-formatet [11].



**Figur 2.2:** Forden från ms-van3t visualiseras i S-LDM via ett webbgränssnitt.



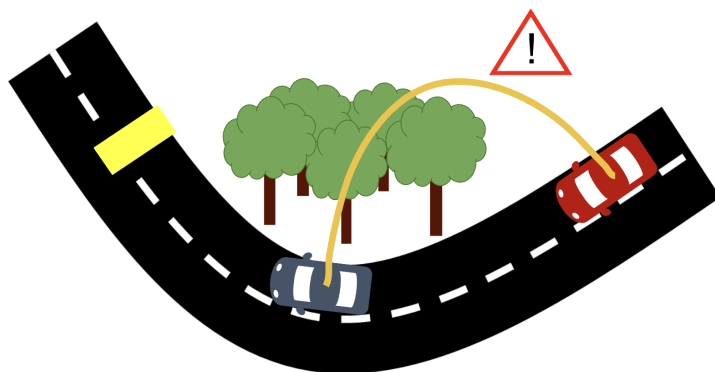
# 3

## Metod

I detta avsnitt redogörs upplägget över beräkningar och tillvägagångssätt som utgör grunden för implementeringen av det kooperativa kollisionsundvikande systemet i ITS-Fairy och ruttplaneringssystemet. Här redovisas även metoder som en grund för framtida arbeten.

### 3.1 ITS-fairy

ITS-fairys syfte är att öka systemets robusthet genom en intelligent stödstruktur vid identifierad trafikfara i S-LDM systemet. När en fara upptäckts i S-LDM ska fordon som befinner sig inom en viss given räckvidd informeras genom ett meddelande om att en närliggande fara har identifierats. Informationen som används i S-LDM samlas in genom CPM-meddelanden från fordonen, vilket möjliggör en tillförlitlig och uppdaterad bild av trafiksituationen. ITS-fairy analyserar dessa CPM-meddelanden för att avgöra om en fara är närvarande och vidarebefordrar informationen till de fordon som saknar den i sin lokala databas, vilket ses i Figur 3.1.



**Figur 3.1:** Exempel där den röda bilen blir medveten om faran i en dold kurva med hjälp av ITS-fairy.

#### 3.1.1 Funktionalitetslägen för ITS-Fairy

ITS-fairy kan agera i flera olika lägen, där varje läge motsvarar en viss grad av autonomi i systemet och olika nivåer av stöd till fordonet. Nedan förklaras de tre

olika lägena som ingår i ITS-fairy:

**Centraliserat ingripandeläge:** Den första modellen i ITS-fairy hör till det kollisionsundvikande systemet, som i ett tidigt skede identifierar situationer med förhöjd kollisionsrisk mellan fordon. Baserat på insamlad trafikinformation analyserar den centrala enheten den aktuella trafiksituationen och fattar beslut om lämpliga åtgärder. Besluten kommuniceras som instruktioner till det berörda fordonet, som förväntas följa dem direkt. Detta läge kännetecknas av en hög grad av extern styrning, där fordonets egna beslutsutrymme minimeras till förmån för ett samordnat och ingripande system.

**Situationsmedvetet stödläge:** Den andra modellen i ITS-fairy analyserar potentiella risker och bedömer utifrån det senaste CPM-meddelandet om fordonet har tillgång till all relevant information för att uppfatta sin omgivning korrekt. Om kritisk information saknas kompletterar systemet detta för att säkerställa fullständig situationsmedvetenhet. Modellen syftar alltså inte till att direkt styra fordonets beteende, utan att istället skapa förutsättningar för att fordonet självt ska kunna fatta informerade och säkra beslut. Genom att fylla i informationsluckor när exempelvis ett fordon inte har sikt mot ett hinder eller ett annat fordon leder detta till att den kooperativa förståelsen av trafikmiljön förbättras.

**Assisterad prioritering av fordon:** I den tredje modellen fungerar ITS-fairy som ett stödsystem för att identifiera vilka fordon som bör prioriteras vid informationsdelning. Genom att anpassa sändningshastigheten för CPM-meddelanden och fokusera på de mest relevanta objekten säkerställs en effektivare informationsspridning i nätverket. Prioritering sker baserat på aktuell trafiksituation och konfliktanalys, där vissa fordon som exempelvis har bäst överblick i en korsning identifieras som särskilt viktiga informationsbärare. ITS-fairy instruerar dessa fordon att temporärt öka sin sändningsfrekvens, medan övriga fordon tillfälligt kan minska, alternativt behålla normal sändningsfrekvens för att undvika överbelastning.

Tillsammans möjliggör dessa tre funktionslägen en flexibel och skalbar arkitektur för samverkan mellan fordon och infrastruktur. Genom att växla mellan olika nivåer av central styrning kan ITS-fairy anpassa sitt beteende efter både trafikmiljöns komplexitet och det individuella fordonets informationsbehov. Denna styrning sträcker sig från direkt ingripande vid akuta kollisionsrisker, till stöd för situationsmedvetenhet och selektiv prioritering av informationsspridning. På så vis balanseras systemets reaktionsförmåga mot kommunikationsbelastning, vilket inte bara bidrar till säkrare trafikflöden, utan även till en mer effektiv och tillförlitlig informationshantering. I särskilda fall, där ett snabbt ingripande krävs kan systemet dessutom gå in och ge explicita instruktioner till fordon, vilket förstärker den kooperativa säkerheten ytterligare.

### 3.1.2 Centraliserad och decentraliserad kollisionsundvikning

S-LDM övervakar fordon för att exempelvis hantera kollisionsundvikning och skicka ut varningsmeddelanden (DENMs) när fördefinierade värden såsom Time-to-Collision (T2C) och Space-to-Collision (S2C) underskrids. Detta representerar en centraliserad och reaktiv strategi som används i det ingripandeläget.

I kontrast introducerar ITS-fairy ett kooperativt decentraliserat angreppssätt, vilket är de resterande lägena. Istället för att direkt styra fordonen delas information om vad som kan hända och vad som händer runt omkring. Detta gör det möjligt för fordonen att själva fatta informerade beslut, vilket möjliggör skalbar koordinering mellan flera fordon, snabbare reaktioner genom lokal beslutsfattning och integration med CPS. Förändringen innebär ett skifte från låg nivå kontroll till högre grad av samarbete vid kollisionsundvikning, där fordonen agerar mer självständigt och i samverkan.

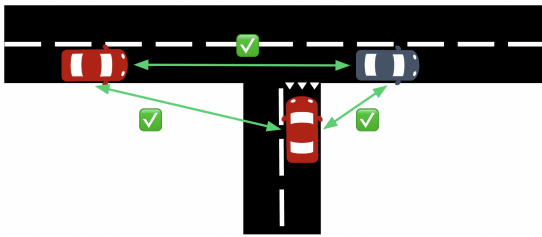
### 3.1.3 Kostnadsanalys för centraliserad kooperativ kollisionsundvikning

För att utvärdera kommunikationskostnaden i ett centraliserat system baserat på S-LDM analyseras två scenarier:

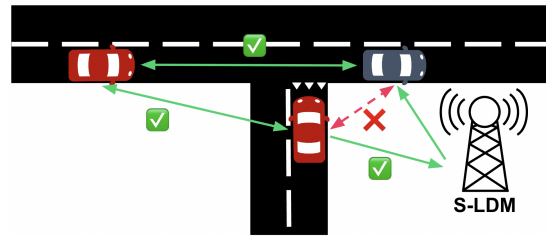
**Lyckat säkerhetsscenario** (Figur 3.2): Alla säkerhetskritiska CPMs tas emot i tid via direkt V2V-kommunikation utan någon fördröjning, och ingen åtgärd från S-LDM krävs. Varje fordon sänder CPMs med en frekvens på  $1\text{ Hz}$ , vilket ger en minimal kommunikationskostnad.

**S-LDM assisterat scenario** (Figur 3.3): Vissa säkerhetskritiska CPMs kan förloras eller försenas i den direkta kommunikationen och måste därför återskapas via S-LDM. S-LDM samlar inte bara in och distribuerar den saknade informationen, utan kan även kombinera kartor med annan sensordata och agera händelsestyrt. Detta möjliggör realtidsidentifiering av risksituationer och automatiserade säkerhetsreaktioner i nätverket. I sin tur leder detta till en ökad kommunikationsbelastning på minst  $n + 1$  meddelanden per sekund (beroende på antalet involverade RSU:er). Vanligtvis krävs två kommunikationsrundor per händelse, en för att samla in den saknade informationen och en för att distribuera den.

Även om S-LDM-assistans kan bidra till att återställa kritisk information, introducerar den samtidigt en risk särskilt under nätverksbelastning. Den extra trafiken kan bidra till ett överbelastat nätverk och förlorad information, därigenom minska systemets totala tillförlitlighet. Detta skapar ett dilemma där försöken att öka pålitligheten genom centraliserad återhämtning kan istället försämra nätverksförhållandena och minska sannolikheten för att meddelanden levereras i tid.



Figur 3.2: Lyckat säkerhetsscenario.



Figur 3.3: S-LDM assisterat scenario.

### 3.1.4 Förklaring av pseudokod

ITS-fairy kan också förklaras genom en pseudokod som ska se till att all information kring faror kommuniceras till alla bilar som riskerar att hotas av faran. Pseudokoden nedan kopplas till det andra funktionsläget av ITS-fairy.

---

#### Algorithm 1 ITS-fairy: Situationsmedvetet stödläge

---

```

1: procedure HANDLE_CONFLICT_AWARENESS( $V, pos$ )
2:    $V \leftarrow \text{getVehs}(V, pos)$  ▷ Returnerar delmängd av fordon
3:    $C \leftarrow \text{conflict}(V)$  ▷ Returnerar faror som alla fordon känner till
4:   for all  $v \in V$  do
5:      $missing\_hazards \leftarrow \emptyset$  ▷ Faror  $v$  inte känner till
6:     for all  $c \in C$  do
7:       if  $v \in c.V$  then ▷ Om  $v$  hotas av  $c$ 
8:         if  $\neg \text{hazard\_reported}(c, v)$  then
9:            $missing\_hazards \leftarrow missing\_hazards \cup \{c\}$ 
10:        end if
11:       end if
12:     end for
13:     if  $missing\_hazards \neq \emptyset$  then
14:        $\text{send\_CPM\_hazard}(v, missing\_hazards)$  ▷ Skicka faror till  $v$ 
15:     end if
16:   end for
17: end procedure

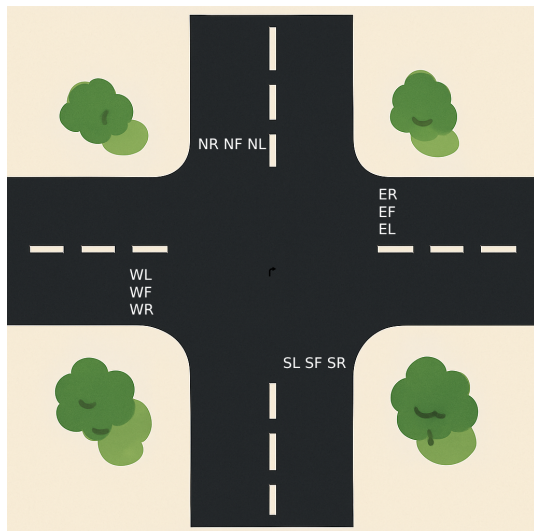
```

---

I början av algoritmen kallas funktionen *getVehs* som använder fordonets aktuella position tillsammans med kartdata från S-LDM för att identifiera och returnera en delmängd av fordon som befinner sig inom ett visst avstånd. Detta görs med hjälp av en medlemstjänst utvecklad av Casimiro, Ekenstedt och Schiller [13]. Denna tjänst uppdaterar regelbundet en uppsättning medlemskap för varje fordon, där varje medlemskap innehåller de fordon som kan utgöra en potentiell prioritetskonflikt i samband med en specifik manöver. Detta kan till exempel röra sig om en vänstersväng. För att avgöra vilka fordon som är relevanta i ett sådant scenario tas hänsyn till faktorer som prioritet enligt trafikregler, samt fordonens position och avstånd till den kritiska punkten. Tjänsten hjälper alltså till att skapa dynamiska och situationsberoende medlemskap för att avgöra vilka fordon som är relevanta för situationen. En mer detaljerad beskrivning av tjänsten finns i rapporten Membership-

based Manoeuvre Negotiation in Autonomous and Safety-critical Vehicular (2021)[13].

Därefter används en delmängd av fordon för att anropa metoden *conflict* som beräknar alla faror som finns inom scenariot. För att identifiera en fara kan flera olika metoder användas. Exempelvis i en korsning kan en konfliktmatris konstrueras som representerar trafikrörelserna i korsningen, se Figur 3.4 och 3.5. Om trafikrörelserna överlappar är en fara upptäckt som därefter läggs till i en lista av konflikter. Därefter itereras alla fordon för att säkerställa att de är medvetna om konflikterna.



Figur 3.4: Fyrvägskorsning med riktningar.

	NR	NF	NL	ER	EF	EL	SR	SF	SL	WR	WF	WL
NR	1	0	0	0	0	0	0	0	0	0	0	0
NF	0	1	0	0	1	0	0	0	1	0	1	1
NL	0	0	1	0	1	1	0	1	0	0	0	1
ER	0	0	0	1	0	0	0	0	0	0	0	0
EF	0	1	1	0	1	0	0	1	0	0	0	1
EL	0	0	1	0	0	1	0	1	1	0	1	0
SR	0	0	0	0	0	0	1	0	0	0	0	0
SF	0	0	1	0	1	1	0	1	0	0	1	0
SL	0	1	0	0	0	1	0	0	1	0	1	1
WR	0	0	0	0	0	0	0	0	0	1	0	0
WF	0	1	0	0	0	1	0	1	1	0	1	0
WL	0	1	1	0	1	0	0	0	1	0	0	1

Figur 3.5: Konfliktmatris där 1:orna motsvarar en potentiell konflikt.

För att prioritera fordonen som är mest utsatta för konflikter används pseudokoden som visas i Algoritm 2. Sändningsfrekvensen höjs för de fordon med flest rapporterade konflikter. Algoritmen kopplas till den tredje modellen av ITS-fairy, *Assisterad prioritering av fordon*. Det första steget i algoritmen är att fordonen sorteras utifrån hur många konflikter de har inrapporterat. De fordon som har identifierat flest konflikter utsätts också för mest risk och därmed ska dessa fordon prioriteras i kommunikationen. För att avgöra hur många av dem som ska tilldelas en högre sändningsfrekvens används variabeln  $n$  som ett gränsvärde.

**Algorithm 2** ITS-fairy: Assisterad prioritering av fordon

---

```
1: procedure INCREASE_BROADCASTING_FREQ( $V, n$ )  $\triangleright V$ : alla fordon,  $n$ : antal
   fordon som ska prioritera sin sändningsfrekvens
2:   sort( $V$ ) efter antal konflikter per fordon  $\triangleright$  Sortera fordon efter flest konflikter
3:    $i \leftarrow 0$ 
4:   for all  $v \in V$  do
5:     if  $i = n$  then
6:       break  $\triangleright$  Avsluta om vi har prioriterat  $n$  fordon
7:     end if
8:      $v.broadcastFrequency \leftarrow$  HIGH_FREQUENCY  $\triangleright$  Exempel: 2 Hz
9:      $i \leftarrow i + 1$ 
10:  end for
11: end procedure
```

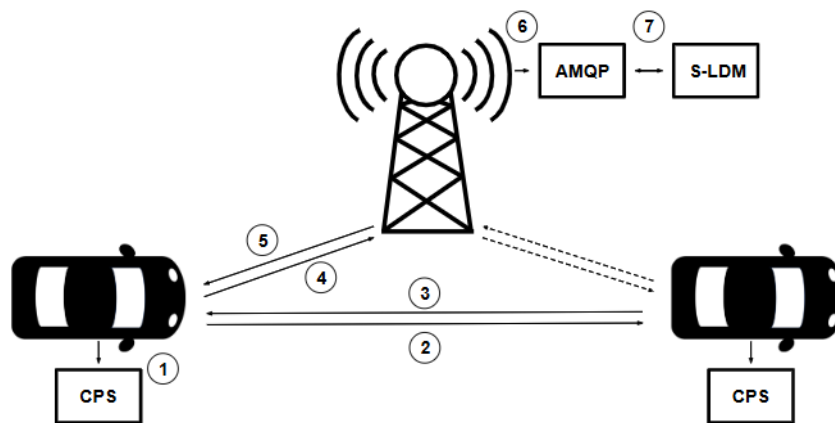
---

## 3.2 Centraliserat ingripandeläge

Det kollisionsundvikande systemet bygger på att fordon kontinuerligt samlar in information från omgivningen genom ett centraliserat system. För att undvika kollisioner i tät trafik och stadsmiljöer krävs det att fordon inte enbart förlitar sig på sina egna sensorer, utan även använder sig av kritisk information från utomstående fordon. Med hjälp av S-LDM, V2X-kommunikation och riskberäkningar kan systemet identifiera en kollisionsrisk och generera varningsmeddelande (DENM) till de inblandade fordonen.

### 3.2.1 Systemöversikt för centraliserade ingripandeläge

Systemet är baserat på pseudokoden som återfinns i doktorsavhandlingen av Marco Malinverno [14] (sidan 162, Appendix B), och i doktorsavhandlingen av Giuseppe Avino [15] (sidan 38, Algoritm 1). Systemet använder S-LDM för att observera fordons positioner och rörelser i realtid, för att upptäcka kollisionsrisker i tid. S-LDM jobbar som en centraliserad server vid en RSU och dess position säkerställer en pålitlig mottagning av meddelanden med minskad störning jämfört med V2V-kommunikation. När systemet upptäcker en risk för potentiell kollision skickas DENM till de involverade fordonen, som innehåller information om att sakta ner eller stanna.



**Figur 3.6:** Flödet av information och kommunikationsmönster.

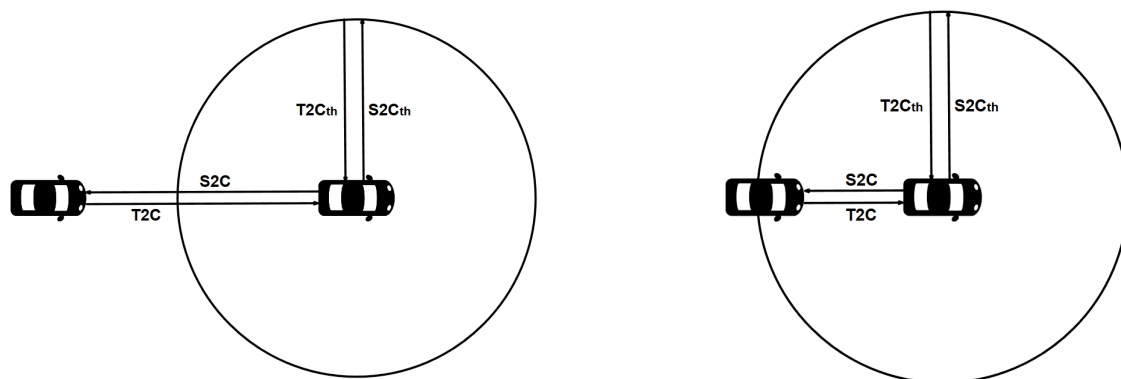
Figur 3.6 visar de olika stegen som sker när systemet bearbetar, och skickar vidare data för att undvika potentiella kollisioner. Processen förklaras med följande steg:

- (1) CPS bearbetar sensor- och V2X-data för att förbättra situationsmedvetenheten.
- (2) Fordonet skickar V2V-meddelanden direkt till närliggande fordon i realtid.
- (3) Fordonet lyssnar efter inkommande V2V-meddelanden och uppdaterar sin LDM med andra fordons realtidsdata och upptäckta objekt.
- (4) Fordonet skickar CPMs till en RSU och delar med sig av sin realtidsdata och upptäckta objekt.
- (5) RSU bearbetar CPMs och svarar med DENMs.
- (6) AMQP är ett protokoll för hantering av meddelandeköer mellan system och används för tillförlitlig överföring av V2X-data mellan enheter.
- (7) S-LDM samlar in trafikdata från RSUs och fordon, vilket ger en centraliserad vy över kollisionsrisker.

### 3.2.2 Riskbedömning av Time-to-Collision (T2C) och Space-to-Collision (S2C)

För att bedöma potentiella kollisioner i stadstrafik analyserar systemet kontinuerligt T2C- och S2C-avstånden mellan fordon. S2C definieras som det minsta beräknade avstånd som kan uppstå mellan två fordon, förutsatt att de följer sina förväntade banor. T2C anger tiden det tar att nå S2C-avståndet givet fordonens aktuella position, hastighet och acceleration. En identifierad kollisionsrisk förekommer om båda parametrarna underskrider sina respektive tröskelvärden, vilket visas i figur 3.7. Dessa värden kan anpassas beroende på väder och andra miljöfaktorer.

Beräkningarna genomförs i realtid i S-LDM enheten, som kontinuerligt tar emot rörelsedata från andra fordon. Vid identifierad risk skickas ett DENM till alla involverande fordon. Detta gör det möjligt för S-LDM att ingripa när fordonens egna sensorer är otillräckliga, vilket hjälper till att undvika potentiella kollisioner och garantera säkerhet.

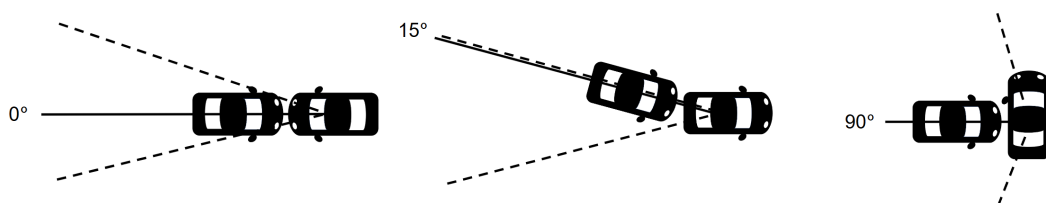


**Figur 3.7:** Om ett fordon's T2C- och S2C-tröskelvärden underskrids samtidigt ( $T2C < T2C_{th}$ ,  $S2C < S2C_{th}$ ) identifieras en kollisionsrisk och ett DENM skickas.

### 3.2.3 Vinkelbaserad kollisionsklassificering

För att identifiera olika trafiksituationer klassificeras potentiella kollisionstyper utifrån riktningen på fordonet. Klassificeringen sker genom beräkningen av vinkelskillnaden mellan fordonens färdriktningar, enligt ekvationen 3.1. Vinklar upp till  $15^\circ$  betraktas som frontalkollisioner eller påkörning bakifrån [39], som Figur 3.8 visar. Däremot indikerar vinklar mellan  $15^\circ$  och  $165^\circ$  sidokollision [39].

$$\Delta\theta(t) = |\theta_v(t) - \theta_w(t)| \quad (3.1)$$



Frontalkollision:  
 $\Delta\theta \leq 15^\circ$

Kollision bakifrån:  
 $\Delta\theta \leq 15^\circ$

Sidokollision:  
 $15^\circ < \Delta\theta < 165^\circ$

**Figur 3.8:** Kollisionsscenarier baserat på vinkelskillnaden mellan fordon ( $\Delta\theta$ ).

Klassificeringarna påverkar både riskbedömningen och ingripandet som S-LDM gör. Vid sido- och frontalkollisioner krävs mer drastiska ingrepp jämfört med påkörning bakifrån, på grund av den ökade säkerhetsrisken [40]. Detta innebär att det inkommande fordonet behöver stanna vid en identifierad sido- och frontalkollision. Vid en identifierad påkörning bakifrån saktar fordonet däremot ner tillräckligt mycket för att överskrida T2C-tröskeln. Kombinationen av vinkelanalys, T2C och S2C skapar en flexibel metod som kan upptäcka och hantera olika typer av farliga trafikscenarier med hög precision.

### 3.2.4 Kommunikation och perceptionsstandarder

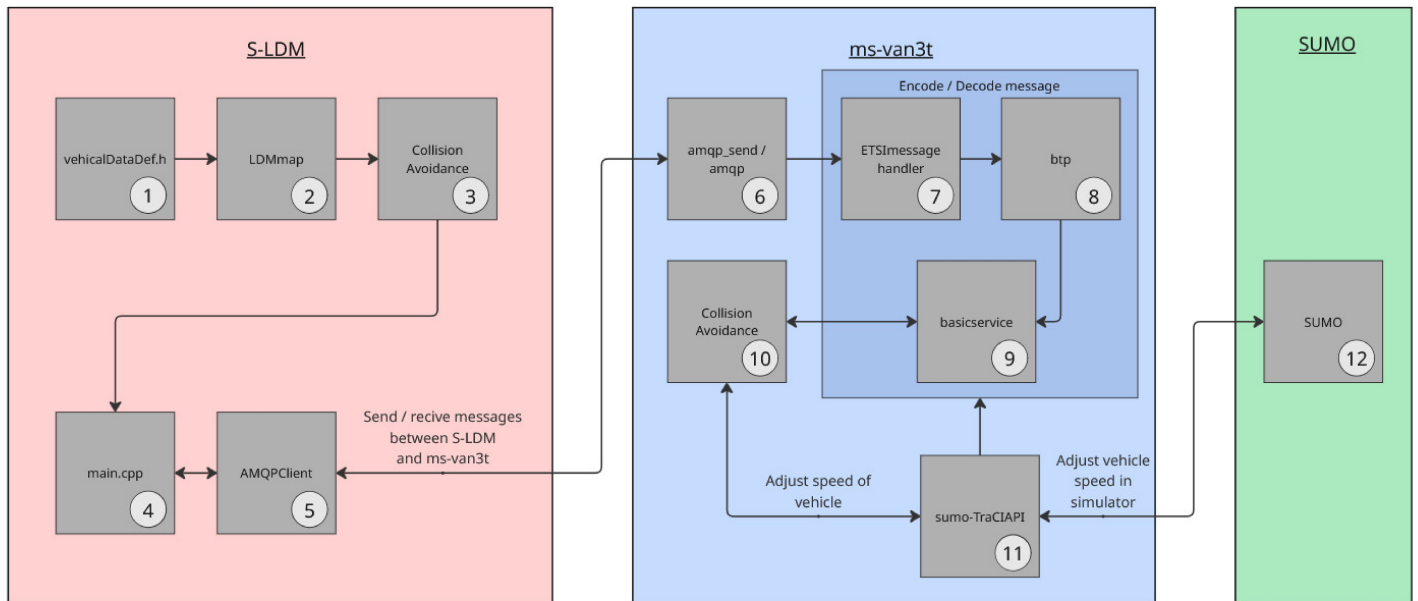
Systemet integrerar CPS, vilket använder sig av CPMs och annan inkommande data för att förbättra perceptionen och garantera säkerheten i trafiken. ETSIs CPM standard, TS 103 324, definierar hur fordon och infrastruktur ska utväxla denna information [41], vilket skickas kontinuerligt med exempelvis frekvensen 1.0 Hz. Informationen skickas i protokoll med slumpmässig åtkomst, utan garanti att nå fram eller med någon specifik prioritering. Därmed behandlas säkerhetskritiska och icke kritiska meddelanden lika på kommunikationskanalen. Detta kan potentiellt leda till att meddelanden med kritisk information försenas eller försvinner.

Det kollisionsundvikande systemet kräver att beslut och åtgärder sker i realtid. För att detta ska vara möjligt krävs det att tillgången av kritisk data garanteras inom en viss tidsram. Den nuvarande standarden saknar förmågan att säkerställa detta, vilket riskerar att meddelanden försenas, förloras helt eller förhindrar de kollisionsundvikande åtgärderna att ske. För att garantera säkerheten måste kommunikationsprotokollen inte bara ge möjlighet till generell tillförlitlighet, utan även garantera realtidsåtkomst av säkerhetskritisk information.

### 3.2.5 Mjukvaruarkitektur för kollisionsundvikande system

Informationen som kommer från den simulerade miljön SUMO skapar en dynamisk återkoppling mellan ms-van3t och S-LDM, vilket ger en realistisk grund för att analysera beteenden under olika trafik scenarier.

Arkitekturen i Figur 3.9 ger en övergripande bild över systemet och dess kommunikationsflöde. Figuren visar bland annat stegen för var fordonets attribut definieras och hur LDM lagrar och uppdaterar sin information. Den visar dessutom var kollisioner detekteras, var varningsmeddelanden genereras och hur dessa meddelanden kommuniceras via AMQP. Slutligen visas också var informationen används för att justera fordonets hastighet i simuleringsmiljön. Flödet som visas i figuren tydliggör grunden för hur testmiljön stödjer testningen av realtidskritisk kommunikation och säkerhetsbeslut.



**Figur 3.9:** Systemets arkitektur. (1) fordonets attribut, (2) lagring av forondsdata i LDM, (3) kollisionsdetektering och generering av DENM-meddelanden, (4–6) kommunikationsflöde mellan S-LDM och ms-van3t via AMQP, (7–10) hantering, avkodning och vidarebefordran av meddelanden, (11) SUMO-gränssnitt för justering av fordonets attribut, samt (12) uppdatering av fordonets beteende i trafiksimulatorn.

### 3.2.6 Algoritmisk säkerhetslogik och pseudokod

Systemets säkerhetslogik bygger på beräkningar av de två parametrarna T2C och S2C, vilket används för att i realtid bedöma risken för potentiella kollisioner mellan fordon. S-LDM enheten genomför beräkningarna kontinuerligt, vilket utgör grunden för att identifiera de situationer där ett DENM ska skickas för att möjliggöra en anpassad och säker reaktion.

Den övergripande logiken beskrivs i Algoritm 3 klassificerar det aktuella fordonet som fordon  $v$  och de närliggande fordonen som fordon  $w$ . Algoritmen börjar med att beräkna fordonens vinkelskillnad ( $\Delta\theta$ ) för att klassificera kollisionstypen. Därefter sker det en distansberäkning som avgör om de involverade fordonens framtida banor hamnar inom ett kritiskt avstånd. Om fordonen bekräftas vara tillräckligt nära varandra, beräknas både T2C och S2C mellan fordonen. Om kollisionstypen är en påkörning bakifrån och T2C är mindre än sitt tröskelvärde kommer ett DENM meddelande skickas som gör att fordon  $v$  saktar ner för att överskrida tröskelvärdet. Om båda parametrarna understiger tröskelvärderna, skickas ett DENM-meddelande till de involverade fordonen som gör att vid en sidokrock stannar fordon  $v$ , medan vid en frontalkrock stannar båda fordonen.

**Algorithm 3** DetectCollision

---

```

1: procedure DETECTCOLLISION(id_vehicle_v, id_vehicle_w, pvx, pvy, pwx,
  pwy, svx, svy, swx, swy, avx, avy, awx, awy, v_heading, w_heading, th_heading,
  T2C_threshold, S2C_threshold)
2:   heading_diff  $\leftarrow$  |v_heading - w_heading|  $\triangleright$  Uträkning av vinkelskillnaden
3:   if heading_diff < th_heading then  $\triangleright$  Påkörning bakifrån eller frontalkrocks
  kontroll
4:     r  $\leftarrow$  IS_IN_RANGE(pvx, pvy, pwx, pwy, svx, svy, swx, swy,
  T2C_threshold)
5:     if r = true then
6:       t2c  $\leftarrow$  TIME_TO_COLLISION(pvx, pvy, pwx, pwy, svx, svy, swx, swy,
  avx, avy, awx, awy)
7:       if 0 < t2c or t2c > T2C_threshold then
8:         continue  $\triangleright$  Fordon för lågt bort tidsmässigt
9:       end if
10:      SEND DENM(to v and w)  $\triangleright$  Sakta ner meddelande till fordon v
11:      s2c  $\leftarrow$  SPACE_TO_COLLISION(pvx, pvy, pwx, pwy, svx, svy, swx, swy,
  avx, avy, awx, awy, TC2)
12:      if s2c  $\leq$  T2C_threshold then  $\triangleright$  Frontalkrock identifierad
13:        SEND DENM(to v and w)
14:      end if
15:    end if
16:  else  $\triangleright$  Sidokrocks kontroll
17:    r  $\leftarrow$  IS_IN_RANGE(pvx, pvy, pwx, pwy, svx, svy, swx, swy,
  T2C_threshold)
18:    if r = true then
19:      t2c  $\leftarrow$  TIME_TO_COLLISION(pvx, pvy, pwx, pwy, svx, svy, swx, swy,
  avx, avy, awx, awy)
20:      if 0 < t2c or t2c > T2C_threshold then
21:        continue  $\triangleright$  Fordon för lågt bort tidsmässigt
22:      end if
23:      s2c  $\leftarrow$  SPACE_TO_COLLISION(pvx, pvy, pwx, pwy, svx, svy, swx, swy,
  avx, avy, awx, awy, T2C)
24:      if s2c  $\leq$  S2C_threshold then  $\triangleright$  Sidokrock identifierad
25:        SEND DENM(to v and w)
26:      end if
27:    end if
28:  end if
29: end procedure

```

---

För att möjliggöra kontinuerlig kollisions identifiering krävs hantering och beräkning av varje fordons data. Detta görs med hjälp av Algoritm 4. Först kontrolleras om en LDM-karta finns att utgå ifrån. Stämmer detta hämtas aktuell fordonsdata om position, hastighet och acceleration kontinuerligt från LDM-kartan för varje fordon  $v$ . Sedan omvandlas fordonsdatan till komponenter i x- och y-led. Därefter görs en sökning i ett radieintervall kring varje fordon  $v$ , vilket syftar till att identifiera närliggande fordon  $w$ . Det största värdet utav T2C-, S2C-tröskelvärdena eller hastigheten används för att bestämma radieintervallet. För varje identifierat fordon  $w$  kontrolleras datans tidsstämpel. Om datan är aktuell anropas Algoritm 3 för att utvärdera och klassificera potentiella kollisionsrisker.

---

**Algorithm 4** HandleVehicleData

---

```
1: procedure HANDLEVEHICLEDATA(Idmmap, stationID)
2:   if not LOOKUP(Idmmap, stationID) then
3:     Error: Vehicle not found
4:     return
5:   end if
6:   // kalkylera hastigheten för fordon  $v$  i x- och y-led
7:    $svx \leftarrow$  CALCULATE_X_COMPONENT(sv, v_heading)
8:    $svy \leftarrow$  CALCULATE_Y_COMPONENT(sv, v_heading)
9:   // kalkylera accelerationen för fordon  $v$  i x- och y-led
10:   $avx \leftarrow$  CALCULATE_X_COMPONENT(av, v_heading)
11:   $avy \leftarrow$  CALCULATE_Y_COMPONENT(av, v_heading)
12:   $radius \leftarrow$  max(sv, T2C_THRESHOLD, S2C_THRESHOLD)
13:  if not RANGESELECT(Idmmap, radius) then
14:    Error: Inga fordon i närheten
15:    return
16:  end if
17:  for each vehicle  $w$  near vehicle  $v$  do
18:     $t\_now \leftarrow$  GET_TIMESTAMP_US
19:    if  $t\_now - w\_timestamp > 3$  seconds then
20:      continue ▷ Ignorera gammal fordon  $w$  data
21:    end if
22:    // kalkylera hastigheten för fordon  $w$  i x- och y-led
23:     $swx \leftarrow$  CALCULATE_X_COMPONENT(sw, w_heading)
24:     $swy \leftarrow$  CALCULATE_Y_COMPONENT(sw, w_heading)
25:    // kalkylera accelerationen för fordon  $w$  i x- och y-led
26:     $awx \leftarrow$  CALCULATE_X_COMPONENT(aw, w_heading)
27:     $awy \leftarrow$  CALCULATE_Y_COMPONENT(aw, w_heading)
28:    DetectCollision(stationID, w_stationID,
29:      pvx, pvx, pwy, svx, svy, swx, swy,
30:      avx, avy, awx, awy, v_heading, w_heading,
31:      HEADING_THRESHOLD, T2C_THRESHOLD, S2C_THRESHOLD)
29:  end for
30: end procedure
```

---

De tre hjälpalgoritmerna Algoritm 5 , Algoritm 6 och Algoritm 7 som stödjer Algoritm 3 genom att utföra specifika delberäkningar. Algoritm 5 beräknar avståndet som fordonen  $v$  och  $w$  hinner färdas innan en potentiell kollision kan inträffa. Tidsgränsen för detta definieras av T2C-tröskeln. Därefter räknar vi ut ett räckviddsområde för båda bilarna och tar ut den största av de två. Detta används därefter för att räkna ut sträckan fordonet förväntas färdas under tiden för T2C-tröskelvärdet. Om avståndet mellan fordon  $v$  och  $w$  är mindre än räckviddsområdet är bilarna tillräckligt nära varandra för att resten av Algoritm 3 ska fortsätta.

---

**Algorithm 5** *Is\_in\_range*


---

```

1: procedure IS_IN_RANGE( $pvx$ ,  $pvy$ ,  $px$ ,  $py$ ,  $svx$ ,  $svy$ ,  $swx$ ,  $swy$ ,
   T2C_threshold)
2:    $pvx\_new \leftarrow pvx + svx \cdot T2C\_threshold$ ;
3:    $pvy\_new \leftarrow pvy + svy \cdot T2C\_threshold$ ;
4:    $px\_new \leftarrow px + swx \cdot T2C\_threshold$ ;
5:    $py\_new \leftarrow py + swy \cdot T2C\_threshold$ ;
6:    $distance\_v \leftarrow \sqrt{(pvx\_new - pvx)^2 + (pvy\_new - pvy)^2}$ 
7:    $distance\_w \leftarrow \sqrt{(px\_new - px)^2 + (py\_new - py)^2}$ 
8:    $distance \leftarrow \max(distance\_v, distance\_w)$ 
9:    $d\_max \leftarrow \sqrt{2 \cdot distance^2}$ 
10:   $distance\_V2W \leftarrow \sqrt{(pvx - px)^2 + (pvy - py)^2}$ 
11:  return  $distance\_V2W < d\_max$ 
12: end procedure

```

---

Algoritm 6 räknar ut den tidsmässiga utvecklingen med hjälp av informationen från fordonen  $v$  och  $w$  längs x- och y-led. Genom att räkna ut derivatan med avseende på tid kan man räkna ut tidpunkterna då avståndet mellan fordonen förväntas vara minimalt. Eftersom ekvationen kan ha flera lösningar väljer algoritmen den minsta icke-negativa roten. Detta görs genom att först kalkylera de olika koefficienterna i andragradsekvationen för att få fram diskriminanten. Därefter används diskriminanten för att räkna ut de olika rötterna där de minsta icke-negativa blir T2C-värdet.

**Algorithm 6** Time\_To\_Collision

---

```

1: procedure TIME_TO_COLLISION(pvx, pvy, pwx, pwy, svx, svy, swx, swy, avx,
  avy, awx, awy)
2:   Output: Space-to-collision (S2C)
3:   kalkylera koefficienterna i andragradsekvationen:
4:    $a \leftarrow 0.5 \cdot (avx - awx)^2 + 0.5 \cdot (avy - awy)^2$ 
5:    $b \leftarrow (svx - swx) \cdot (avx - awx) + (svy - swy) \cdot (avy - awy)$ 
6:    $c \leftarrow (pvx - pwx) \cdot (svx - swx) + (pvy - pwy) \cdot (svy - swy)$ 
7:    $discriminant \leftarrow b^2 - 4ac$ 
8:    $t1 \leftarrow \frac{-b + \sqrt{discriminant}}{2a}$ 
9:    $t2 \leftarrow \frac{-b - \sqrt{discriminant}}{2a}$  ▷ kalkylera rötterna  $t1, t2$ 
10:  if  $t1 < 0$  &  $t2 < 0$ 
11:     $t2c \leftarrow -1$ 
12:  else
13:     $t2c \leftarrow \min(t1, t2)$ 
14:  end if
15:  return  $t2c$ 
16: end procedure

```

---

Det Algoritim 7 gör är att räkna ut de uppskattade positionerna för fordon  $v$  och  $w$  i x och y-led vid T2C för att räkna ut S2C.

**Algorithm 7** Space\_To\_Collision

---

```

1: procedure SPACE_TO_COLLISION(pvx, pvy, pwx, pwy, svx, svy, swx, swy,
  avx, avy, awx, awy, T2C)
2:   Output: Space-to-collision (S2C)
3:    $Pvx \leftarrow pvx + svx \cdot t2c + 0.5 \cdot avx \cdot t2c^2$ 
4:    $Pvy \leftarrow pvy + svy \cdot t2c + 0.5 \cdot avy \cdot t2c^2$ 
5:    $Pwx \leftarrow pwx + swx \cdot t2c + 0.5 \cdot awx \cdot t2c^2$ 
6:    $Pwy \leftarrow pwy + swy \cdot t2c + 0.5 \cdot awy \cdot t2c^2$ 
7:    $S2C \leftarrow \sqrt{(Pvx - Pwx)^2 + (Pvy - Pwy)^2}$ 
8:   return S2C
9: end procedure

```

---

Säkerhetslogiken gör det möjligt för systemet att reagera omedelbart på förändringar i trafiken, vilket i sin tur bidrar till att fordon ges tillräckligt med tid att agera proaktivt och därmed garanterar trafiksäkerheten.

### 3.3 Kollektiv ruttplanering

Nästa system som presenteras syftar till att effektivisera trafikflöden samt hur implementeringen av ett sådant system kan se ut. För att effektivt dirigera trafik krävs ett system som är medveten om alla fordons positioner och baserat på detta ta beslut som främjar ett effektivt trafikflöde. Syftet med detta är att undvika

köer och onödiga stopp. Ett sådant system kan beskrivas som en "svart låda" där trafiksituationen är indata och beslutet om hur trafiksituationen skall hanteras är utdata. Det finns flera olika sätt att konstruera ett sådant system och en presentation av den nuvarande implementeringen samt andra teoretiska tillämpningar kommer att göras här. Grundidéen är att varje fordon skall få körningsanvisningar som syftar till att effektivisera trafikflöden. Det finns flera olika sätt att konstruera ett sådant system, men de tre centrala aspekterna är integritet, prestanda och kontroll. Detta innebär att systemet existerar på ett spektrum där olika implementeringar kan fokusera på olika områden i varierande utsträckning.

### 3.3.1 Hantering av integritet

Integritet är ett viktigt ämne när känsliga personuppgifter hanteras. I detta fall visar den känsliga informationen var olika fordon är på väg. För att undvika hantering av sådan data finns en rad olika metoder för att implementera det föreslagna systemet. Vid användning av en MAPF-algoritm krävs både position och mål, men det är även möjligt att ge systemet ett slumpat mål som ligger i närheten av fordonets verkliga destination. På så sätt kan en effektiv rutt för ett fordon fortfarande beräknas utan att delge information om fordonets sanna destination. Det är även möjligt att dela upp fordonets rutt i mindre delar och endast ge MAPF-algoritmen ett delmål. På så sätt undviks att systemet har kännedom om fordonets verkliga mål samtidigt som rutten kan planeras med hänsyn till övriga fordon.

Istället för att varje fordon tvingas delge information om sin rutt kan man förutspå vart ett fordon är på väg. Fördelarna med detta är att systemet endast behöver information om varje fordons nuvarande position [42]. Korsningar kan i sådant fall effektivt hanteras baserat på hur ett fordon förväntas agera och om det avviker från det förutspådda valet finns det en logik som hanterar dessa fall.

### 3.3.2 Prestanda av ruttplanering

Att genomföra en sökning med en MAPF-algoritm kan kräva mycket beräkningskraft som ökar drastiskt med fler agenter. Detta kan göra det helt omöjligt att genomföra en sökning på en större karta. Däremot finns det några sätt att öka prestandan, samtidigt som målet att planera effektiva rutter bibehålls. En sådan metod kan vara att använda en icke-optimal MAPF-algoritm. Dessa algoritmer garanterar inte att de hittar den optimala lösningen, men samtidigt kräver dessa algoritmer mindre beräkningskraft [43].

Ytteligare en lösning kan vara att dela upp kartan i mindre bitar och planera rutter för fordonen i detta begränsade område. På så sätt kräver varje sökning mindre resurser. Detta ger fördelen att fordonens sanna destination inte alltid avslöjas för systemet, vilket ökar integriteten. Istället för att planera rutterna för varje fordon kan antalet fordon som får vistas på en vägsträcka begränsas. I detta fall kan en algoritm som bygger på begränsandet av antalet fordon på en specifik väg. I detta fall är det möjligt att använda positionsdata från S-LDM. Ett sådant

system kan sedan kombineras med Virtuella Trafikljus (VTL) för att förhindra att trafikdensiteten blir allt för hög på en väg.

Sista implementeringsförslaget är en kombination av att planera rutter för fordon och att begränsa fordons åtkomst till vissa vägar. När ett fordon färdas på mindre vägar som exempelvis i ett villaområde eller i en stadskärna planeras rutterna för fordonen med MAPF-algoritmen. Däremot om ett fordon skall åka på en motorväg eller en större landsväg begränsas trafikdensiteten. Fördelen med denna lösning är att komplexa trafiksituationer som stadskörning kan planeras optimalt, samtidigt som enklare trafiksituationer som motorvägar endast hanteras baserat på fordonsdensitet [44].

#### 3.3.3 Kontroll av fordon

Den sista aspekten som systemet bör ta hänsyn till är hur mycket kontroll systemet bör ha över varje fordon. Om systemet har total kontroll över varje fordon behöver systemet inte beräkna nya rutter för varje fordon lika ofta eftersom systemet vet att ett fordon kommer följa den givna ruten. Ett problem som uppstår är att all trafik blir beroende av att systemet fungerar och kommunikationsstörningar kan då få stora negativa konsekvenser. Istället kan centralt planerade rutter användas som vägbeskrivningar för varje enskilt fordon. Föraren har fortfarande makten att avvika från ruten eller stanna om detta önskas. Fördelarna med detta är att ett sådant system kan implementeras i uppkopplade fordon eftersom en mänsklig förare kan ha kontroll över fordonet. Redan inom 5 år förväntas 95% av alla nyproducerade fordon vara utrustade med ITS-Stationer, vilket möjliggör denna implementering inom en snar framtid [45]. Även användningen av VTL ger föraren kontroll över fordonet och tillvägagångssättet kan implementeras i fordon med ITS-Stationer.

#### 3.3.4 Användning av Virtuella trafikljus

Givet ett bestämt körschema behövs en metod för att kontrollera en förares agerande vid korsningar. En given rutt kan innehålla planerade stopp och för att en förare skall följa ruttens anvisningar måste en mekanism finnas för att uppfylla detta. I ett sådant scenario kan virtuella trafikljus användas där. Ett rödljus visas för föraren vid ett specificerat tillfälle enligt ett körschema.

Ett sådant system kan integreras både för fordon framförda av mänskliga förare eller självkörande fordon. Detta har stor potential eftersom ett sådant system tillåter dagens fordon att dra nytta av vissa fördelar som självkörande fordon i framtiden kommer att ha. Även om detta system inte gör att fordon kör autonomt tar det ett steg närmare till en värld där fordon faktisk är självkörande.

Implementering av ett sådant system kan beskrivas i fyra steg. Första steget är att skapa ett schema för varje fordon som beskriver när de bör stanna, alltså när ett virtuellt rödljus skall skickas till fordonet. Detta beskrivs av Algoritm 8.

---

**Algorithm 8** Schedule Vehicle Stops

---

```

1: procedure SCHEDULESTOPS(schedule)
2:   vtlSchedule  $\leftarrow$  [] ▷ Skapa tom lista över alla stopp
3:   for all vehicle in schedule do ▷ Gå igenom varje fordons schema
4:     for all step in agent.steps do ▷ Gå igenom varje steg i fordonets schema
5:       if step.startPos == step.goalPos then ▷ Kolla om fordonet ska vänta
6:         vtlSchedule.append((step.goalPos, step.duration, vehicle)) ▷
           Registrera position, varaktighet och fordon för stoppet
7:       end if
8:     end for
9:   end for
10:  return vtlSchedule ▷ Returnera schemat över alla stopp
11: end procedure

```

---

När schemat för alla stopp har skapats behöver dessa schemaläggas för varje fordon. Detta beskrivs av Algoritm 9.

---

**Algorithm 9** Schedule Vehicle Traffic Lights

---

```

1: procedure STOPVEHICLES(vtlSchedule)
2:   for all (intersectionToStopAt, stopDuration, vehID)  $\in$  vtlSchedule do ▷
     Gå igenom alla schemalagda stopp
3:     timeToStop  $\leftarrow$  CALCULATETIME TO INTERSECTION(vehID,
     intersectionToStopAt) ▷ Beräkna ankomsttiden till korsningen
4:     SCHEDULEVTL(vehID, timeToStop, stopDuration) ▷ Schemalägg ett
       stopp i trafikljuset
5:   end for
6: end procedure

```

---

I Algoritm 10 beräknas den förväntade tiden som tar fordonet att nå den specificerade korsningen där det skall stanna.

**Algorithm 10** Calculate Time to Intersection

---

```

1: function CALCULATETIMETOINTERSECTION(vehicleRoute, stopIntersection)
2:   timeToIntersection  $\leftarrow$  0 ▷ Initiera tidräkning
3:   for all road in vehicleRoute do ▷ Iterera genom alla vägar i rутten
4:     timeToIntersection  $\leftarrow$  timeToIntersection +  $\frac{vehSpeed}{road.length}$  ▷ Lägg till
     restiden för aktuell väg
5:     if road == stopIntersection then
6:       return timeToIntersection ▷ Returnera tiden då fordonet når rätt
     korsning
7:     end if
8:   end for
9: end function

```

---

Till sist skall ett virtuellt röd ljus exekveras. Detta görs enligt Algoritm 11.

**Algorithm 11** Schedule Virtual Traffic Light

---

```

1: procedure SCHEDULEVTL(vehID, timeToStop, stopDuration)
2:   ▷ Planera det virtuella trafikljuset i timeToStop sekunder
3:   ▷ Virtuella trafikljuset pågår i stopDuration sekunder
4:   SCHEDULEYELLOWLIGHT(vehID, timeToStop - 3, 3) ▷ Gult ljus tre
     sekunder innan stopp
5:   SCHEDULEREDLIGHT(vehID, timeToStop, stopDuration) ▷ Rött ljus
     under stopDuration sekunder
6:   SCHEDULEREDYELLOWLIGHT(vehID, timeToStop + stopDuration, 1) ▷
     Rödgult ljus före grönt ljus
7:   SCHEDULEGREENLIGHT(vehID, timeToStop + stopDuration + 1) ▷ Visa
     grönt ljus
8: end procedure

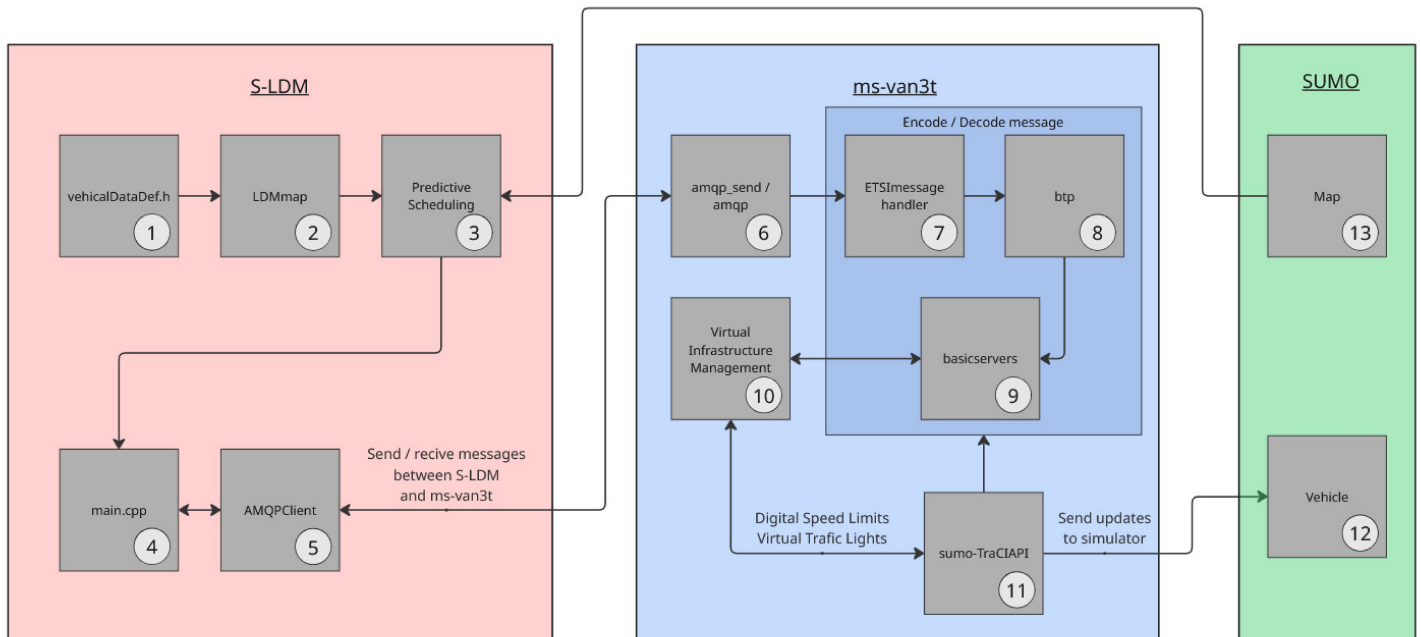
```

---

Denna pseudokod beskriver ett system som syftar till att påverka hur bilar agerar givet ett körschema. Systemet tillåter då förare att följa en planerad rutt utan att ett centralt system behöver kontrollera varje fordon.

### 3.3.5 Mjukvaruarkitektur för kollektivt ruttplanerande system

Figur 3.10 visar hur ett system integrerat med S-LDMen kan se ut. I ett sådant system finns ingen information om fordonens destination och därför behöver det finnas ett annat sätt att ta reda på hur fordon förväntas agera. Detta är syftet med steg (3) i Figur 3.10. Det som skiljer denna struktur från den i Figur 3.9 är vilken information som hämtas i SUMO och hur den används. En karta över vägar och korsningar hämtas tillsammans med information om fordonen. Kartan tillsammans med fordonens start- och målpositioner används för att schemalägga rutter. Dessa rutter skickas till fordonen som sedan förväntas följa dem.



**Figur 3.10:** Systemets arkitektur. (1) fordonets attribut, (2) lagring av fordonsdata i LDM, (3) schemaläggare som förutspår hur fordonen kommer färdas, (4–6) kommunikationsflöde mellan S-LDM och ms-van3t via AMQP, (7–10) hantering, avkodning och vidarebefordran av meddelanden, (11) SUMO-gränssnitt för justering av fordons attribut, (12) fordon mottar kommandon baserat på schemat, samt (13) hämtar vägar, korsningar och annan infrastruktur ur kartan.



# 4

## Implementering

I metoden beskrivs koncept för ITS-fairy i olika funktionalitetslägen och system för ruttplanering. I följande kapitel beskrivs implementeringen för dessa samt installation av simulatören och kommunikationsimplementering mellan S-LDM och ms-van3t.

### 4.1 Installation av simulator

Det rekommenderade operativsystemet som installerades var Ubuntu 22.04. Därefter installerades SUMO tillsammans med ms-van3t och S-LDM enligt medföljande guider. Till dessa medföljde även exempelfiler på trafikscenarion som kunde byggas på och användas som grund för projektet. Eftersom S-LDM utvecklades för flera år sedan krävdes anpassningar för att den skulle fungera med en nyare kompilator. För att etablera kommunikationen mellan ms-van3t och S-LDM behövde en git patch-fil appliceras, som lade till den kod som behövdes för att bygga och skicka meddelanden till S-LDM. Patch-filen saknade nödvändig kod för att kommunikationen skulle fungera, men genom analysering av exempelfilerna i ms-van3t löstes kommunikationsproblemet.

### 4.2 Kommunikation mellan S-LDM och ms-van3t

Bristande kommunikation mellan ms-van3t och S-LDM identifierades, specifikt i riktningen från S-LDM tillbaka till ms-van3t. För att säkerställa framtida funktionalitet och tillförlitlighet i systemet utformades ett stabilt kommunikationsprotokoll. Protokollet liknade den kommunikation från ms-van3t till S-LDM där activeMQ används som förmedlare för meddelandena. S-LDM publicerar meddelanden på ett ämne hos activeMQ som ms-van3t prenumererar på för att ta emot meddelandena.

### 4.3 Centraliserade ingripandeläget

Det kollisionundvikande säkerhetssystemet implementerades som en central del i S-LDM. Syftet med systemet var att i realtid analysera fordonens rörelsedata från SUMO och avgöra om en potentiell kollision skulle förekomma. Vid identifierad risk skickades ett DENM till de inblandade fordonen för att undvika potentiella kollisioner genom att sakta ner eller stanna bilarna.

Systemet utformades för att ta emot fordonsdata via AMQP från ms-van3t. Varje inkommande meddelande behandlades och lagrades temporärt i S-LDM som kontinuerligt uppdaterades med nya meddelanden. Baserat på informationen som skickades från ms-van3t analyserade S-LDM systemet trafiksituationen och fattade beslut om eventuella ingrepp som behövde göras. Analysen och beslutslogiken delades upp i separata funktionsenheter för datahantering, analys och respons.

När ett varningsmeddelande skickades från S-LDM publicerades det till ett specifikt ämne i activeMQ som ms-van3t också var prenumererad på. Genom kanalen kunde meddelanden skickas från S-LDM till ms-van3t. Därefter skickade ms-van3t ett stopp- eller sakta ner-kommando till rätt fordon i SUMO. Detta skapade ett realtidssystem mellan den inkommande datan som S-LDM analyserade, till de visuella ingreppen som ms-van3t gjorde i SUMO.

### 4.4 Situationsmedvetna stödläget

I S-LDM databasen lagras objekt som representerar fordon och innehåller information såsom fordonstyp, position, riktning, hastighet, fordonets storlek samt tidpunkten för senaste uppdateringen. Däremot sparar S-LDM ingen information om vilken typ av väg fordonet befinner sig på, CPM-meddelanden skickas heller inte till S-LDM. Därför begränsades implementationen av det situationsmedvetna stödläget till den data som samlas in från CAM-meddelanden. Det implementerades en algoritm som kan identifiera ett stillastående fordon på en väg för att sedan skicka ett varningsmeddelande till de bilar som påverkas av konflikten. Utifrån pseudokoden i Algoritm 1 representerar denna algoritm den metod som kallas `conflict(V)`. Även detta system utformades för att skicka varningsmeddelanden via AMQP från S-LDM, liksom det centraliserade ingripandeläget.

### 4.5 Kollektiva ruttplaneringssystemet

Implementeringen av effektiviseringssystemet gjordes med hjälp av MAPF-algoritmen C-CBS. Ruttplaneringen har gjorts under antagandet att destinationen för varje fordon är känd samt att algoritmen endast hanterar ett begränsat antal fordon. Dessa antaganden gjordes för att validera konceptet utan att behöva ta hänsyn till komplexiteten som integritet och prestanda medför.

För att hitta en kollisionsfri rutt för varje fordon användes fordonens aktuella position och destination. Detta hämtades med hjälp av TraCI API:n, vilket är en del av SUMO och sparades sedan till en XML-fil som C-CBS programmet kan läsa. Även kartan konverteras till ett format som C-CBS kan läsa. Detta skapade vissa komplikationer eftersom algoritmen detekterar om två agenter befinner sig på samma plats på grafen. När kartan konverteras till en graf tas det endast hänsyn till korsningarnas position samt hur många vägar som leder in och ut ur korsningen. Detta innebär att kurvor, höjdskillnader eller avståndet mellan två filer inte inkluderas. Eftersom varje väg definieras som en rak linje mellan två noder

innebär detta att två motgående filer ligger ovanpå varandra. C-CBS identifierar kollisioner genom att undersöka om två agenter kommer befinna sig på samma plats samtidigt. Därav skapas en konflikt när två fordon förväntas färdas i motsatt riktning på samma vägsträcka. För att kringgå problemet skapades en extra nod för varje in- och utgående väg i korsningen. Dessa ansluts till varandra och placeras ut i en cirkel vilket säkerställer att två vägar inte hamnar ovanpå varandra.

När all information om de nuvarande trafikläget samt kartan är insamlad går C-CBS igenom informationen och hittar en optimal lösning. Denna lösning läses sedan och varje fordon får därefter en uppdaterad rutt som de kommer att följa. Denna rutt behandlas i SUMO, som en typ av vägvisning. Även detta är en förenkling eftersom C-CBS inte tar hänsyn till fordonens acceleration eller de-acceleration.



# 5

## Utvärdering

V2X-kommunikation i uppkopplade fordon ger möjligheter för framtida trafiksystem, men innan tekniken kan implementeras i verkliga fordonsmiljöer krävs en noggrann utvärdering av dess funktionalitet. I detta kapitel presenteras undersökningsfrågor och utvärderingskriterier av de tre systemen som arbetet fokuserar på.

### 5.1 Undersökningsfrågor

För att utvärdera det centraliserade ingripandeläget analyseras tröskelvärdena för parametrarna T2C och S2C. S2C definieras som det minsta beräknade avståndet som kan inträffa mellan två fordon. T2C anger tiden det tar att nå S2C-avståndet. De frågor som ställs är: Hur påverkar tröskelvärdena för T2C och S2C systemets förmåga att upptäcka kollisioner? Samt, hur påverkar T2C- och S2C-tröskelvärdena systemets förmåga att förhindra kollisioner?

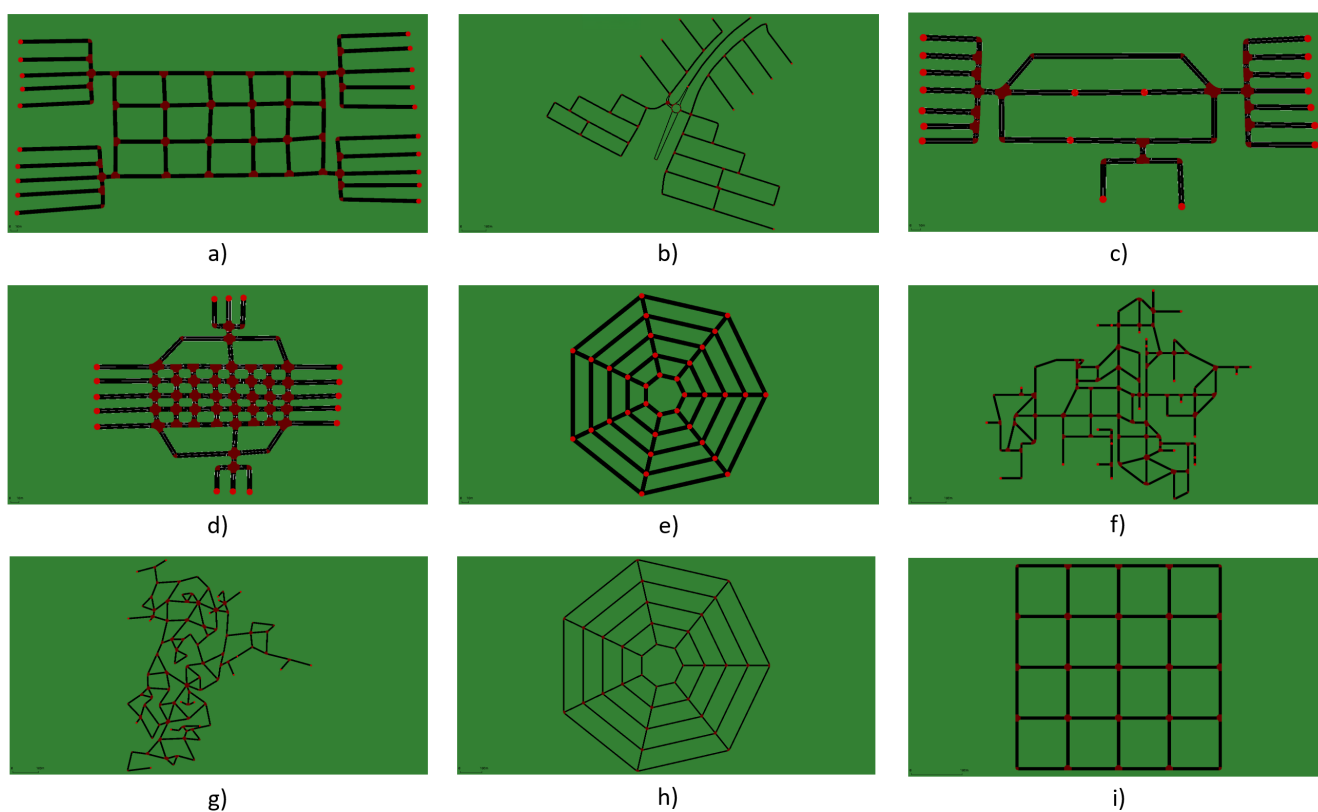
Systemets tillämpning i stadsmiljö kräver även att kommunikation sker i realtid, och utvärderingen syftar till att hitta en uppdateringsfrekvens som garanterar säkerheten utan att kompromissa med kommunikationens tillförlitlighet. Därför ställs undersökningsfrågan: Hur påverkar uppdateringsfrekvensen systemets säkerhetsnivå i en stadstrafiksituation?

I det situationsmedvetna stödläget, jämförs fordonens lokala kartor med varandra. Vilket leder till frågan: Kan ITS-fairy bidra till en mer korrekt och enhetlig uppfattning av omgivningen.

Syftet med det kollektiva ruttplanerande systemet är att effektivisera trafikflödet i olika scenarier. För att utvärdera systemet ställs då frågan: Kan den totala restiden för alla fordon minskas om varje rutt planeras med hänsyn till övriga fordon i ett givet scenario?

### 5.2 Utvärderingsmiljö

För att testa de implementerade systemen krävs en utvärderingsmiljö i form av en karta i simulatoren SUMO. Alla testkartor ses i Figur 5.1 där kartorna fått olika namn. Samtliga kartor har använts till att utvärdera ruttplaneringssystemet. Delar i karta b) används för att utvärdera det centraliserade ingripandeläget. Bitarna som då används är rondellen i mitten, en trevägs- och en fyrvägs korsning.



**Figur 5.1:** De nio kartorna som används har fått följande namn: a) Manhattan, b) Solängen, c) Gaffel, d) Häger, e) Litet spindelnät, f) Van Gogh, g) Picasso, h) Stort spindelnät och i) Schackbräde.

## 5.3 Funktionalitet

De funktioner som kommer utvärderas i ITS-fairy är framförallt det centraliserade ingripandeläget, vilket innefattar det kollisionssundvikande systemet. Systemet upptäcker risk för kollisioner baserat på tröskelvärden för T2C och S2C. Upptäckta kollisionrisker klassificeras som korsningskollisioner, påkörningar bakifrån eller frontalkrockar genom vinkelanlys. Vid identifierad risk för kollision skickas meddelanden ut och åtgärd tas av de inblandade fordonen. Om V2V-kommunikationen är fördröjd eller utebliven, hanterar S-LDM det kollisionssundvikande arbetet. Därmed utvärderas S-LDM i dess kollisionssundvikande algoritm och förmåga för realtidskommunikation. Dessutom kommer det situationsmedvetna stödläget i ITS-fairy utvärderas utifrån den konfliktidentifiering som är implementerad.

## 5.4 Utvärderingskriterier

För det centraliserade ingripande läget i ITS-fairy analyseras T2C, där tiden för kollisionrisk beräknas och dess korrelation med hur väl S-LDM ingriper. Detta jämförs med värdena från SUMOs ursprungliga kollisionssundvikande system samt dess system med störningsförhållanden. Vidare utvärderas systemets uppdateringsfrekvens där fokus ligger på hur ofta ny fordonsdata mottas och hur detta påverkar tillgången till aktuell information för beslutstagande.

För det situationsmedvetna stödläget kommer tiden från att en fara identifieras till dess att den når fordonet att utvärderas. Denna tidsperiod är avgörande för att kunna bedöma systemets förmåga att reagera proaktivt och vidta lämpliga åtgärder i god tid.

Målet för det kollektiva ruttplaneringssystemet är att minska den totala latensen för alla fordon. Med detta menas att den tid det tar för varje fordon att nå sin destination summeras. Den totala latensen ger en inblick i hur rutter planerade med hänsyn till andra fordon kan minska köer och därmed också utsläpp från fordon.

## 5.5 Experiment

I kommande avsnitt presenteras experimenten som utförs för att validera systemen utefter funktionaliteten och de undersökningsfrågor som ställts.

### 5.5.1 Centraliserade ingripandeläget

Två experiment genomförs för att testa hur väl det centraliserade ingripandeläget för ITS-fairy upptäcker och agerar i kollisionrisk, samt hur uppdateringsfrekvensen påverkar säkerheten. Alla experiment är baserade på simulerade scenarier i SUMO på kartan Solängen i Figur 5.1. Fokus ligger på hur det kollisionssundvikande systemet i ITS-fairy identifierar och ingriper där det finns kollisionrisk. Framförallt då simulerade förluster av information sker i SUMO och ITS-fairy är den enda

komponenten som har nödvändig information.

Det första experimentet utvärderar hur T2C- och S2C-tröskelvärden påverkar systemets förmåga att upptäcka och förhindra kollisioner. Tre vägsценarier har valts ut, trevägskorsning och fyrvägskorsning med vänstersväng, samt inkörningsscenario i rondell. Dessa valdes då de utgör högrisksценarier i trafiken, där avstånd- och hastighetsbedömningen för en förare ofta är komplex [46] [47]. Varje vägsscenario testas i tre omgångar, alla med ingripanden av ITS-fairy där tröskelvärdena skiftas. Ett standardläge där T2C-tröskelvärdet =  $4.8s$  och S2C-tröskelvärdet =  $4.2m$ , samt höga och låga tröskelvärden med förändring på  $\pm 30\%$ . Tidigare studier visar att T2C-tröskelvärdet ligger inom intervallet  $4.5 - 5s$  [48]. Men även ett basvärde på T2C-tröskelvärdet =  $4s$  samtidigt som S2C-tröskelvärdet =  $4m$ , men beroende på trafiksituation och kommunikationssätt kan dessa värden ändras [15]. Utifrån de trafiksituationer som testas var detta rimliga tröskelvärden. Fem förartyper, säker till aggressiv, modelleras via parametrarna  $\tau$  som styr förarens tidslucka,  $\sigma$  för bristande perfektion i körning,  $accel$  och  $decel$  för acceleration och inbromsning, samt  $speedFactor$  för hur mycket föraren skiljer från hastighetsbegränsningen [49].

Det andra experimentet utförs i samma scenarier men varierar uppdateringsfrekvensen av fordonsinformation till  $0.5 Hz$ ,  $1 Hz$  (ETSI-standard [35]),  $2 Hz$  och  $5 Hz$ . Låg frekvens förväntas missa kritisk information. Medan högre frekvens förväntas ge mer precis information kring fordonen, men riskerar också att förlora kritisk information till följd av hög kommunikationsbelastning. Resultaten analyseras utefter T2C-värden för att bedöma hur säkerheten skiljer sig för de olika uppdateringsfrekvenserna. Samt utifrån hur ofta fordonsinformation uppdateras och hur detta påverkar systemets tillgång till realtidsinformation.

### 5.5.2 Situationsmedvetna stödläget

Experimentet för det situationsmedvetna läget implementerades utifrån Algoritm 1, vilken beräknar tiden det tar att identifiera en konflikt i S-LDM och skicka meddelande till de fordon som utsätts för konflikten. Mer specifikt handlar testet om ett simulerat scenario där det befinner sig ett stillastående fordon längs med en höghastighetsväg. Därefter skickas ett varningsmeddelande till fordonen längre bort att ett annat fordon är stillastående. Två test genomförs för att undersöka latensen i systemet. I det första experimentet genereras ett varningsmeddelande som systemet skickar till AMQP. Det andra experimentet genererar endast varningsmeddelandet utan att skickas. I båda fallen mäts den totala latensen.

### 5.5.3 Kollektiva ruttplaneringssystemet

Ett experiment genomförs i flera olika scenarier för att utvärdera huruvida planerade rutter minskar den totala latensen.

Experimenten utgår från 9 olika kartor där förutbestämda rutter har satts. De förutbestämda rutterna är utvalda genom att först välja en väg där ett

fordon skall starta och sedan välja ett mål som fordonet ska ta sig till. SUMO väljer därefter en effektiv rutt mellan dessa två vägar baserat på den totala tiden som ett fordon förväntas färdas. Denna rutt baseras endast på den förväntade tiden det bör ta för fordonet att nå sitt mål och tar inte hänsyn till övriga fordon.

Experimenten genomförs i tre steg. Först mäts den totala latensen för alla fordon när de samtidigt följer den förutbestämda effektiva rutten. Fordonen färdas då utan någon global koordinering. Den koordinering som sker är endast i korsningar, där fordonen väntar på varandra i syfte att inte kollidera. Därefter genomförs samma scenario, men nu med den globala ruttplaneringen aktiverad. Alla rutter planeras i början av scenariot och de planerade rutterna skickas sedan till varje fordon. Även här summeras latensen för varje fordon. I den sista delen genomför varje fordon sin egen rutt ensamt. Det finns alltså bara ett fordon på kartan vilket innebär att trafik och köbildning inte påverkar fordonets latens. Detta blir det ideala fallet och används som en utgångspunkt för de två andra testerna.

Eftersom C-CBS ger en deterministisk lösning genomförs testerna för varje scenario endast en gång. Upprepade genomföranden av samma scenario ger alltså samma resultat.



# 6

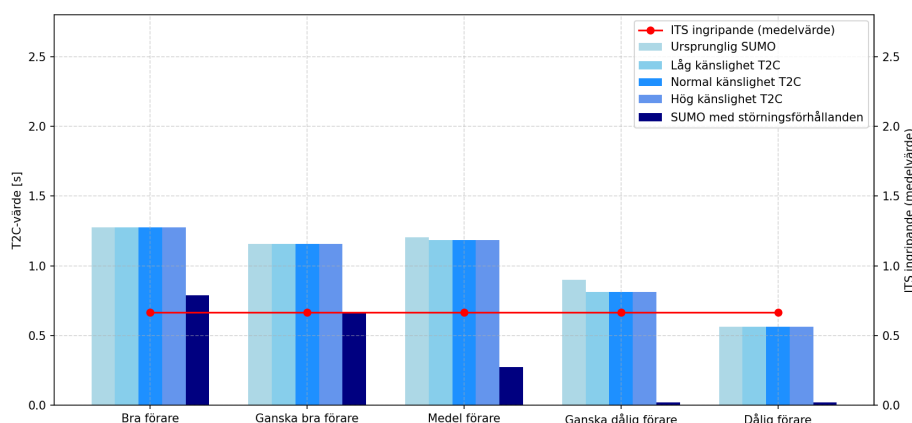
## Resultat

Detta kapitel redovisar samtliga resultat från de experiment som utförts. Här analyseras resultaten utifrån de grafer som presenteras.

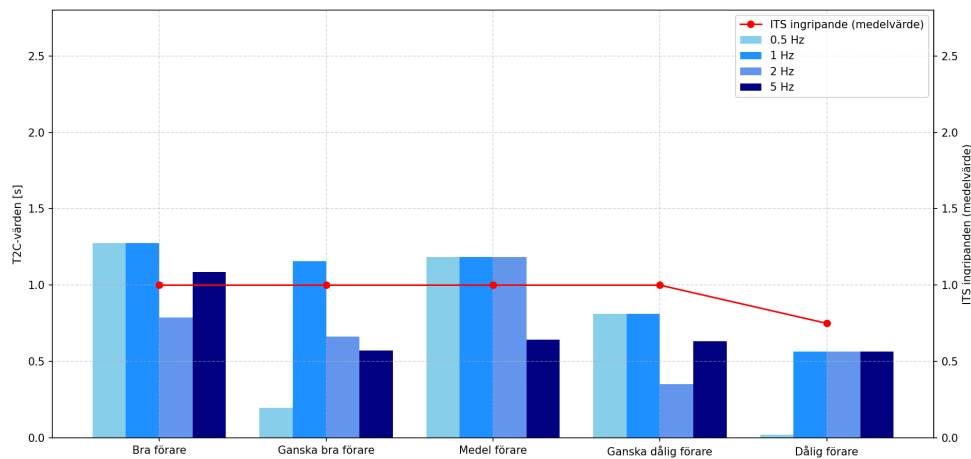
### 6.1 Centraliserade ingripandeläget

I Figur 6.1 jämförs T2C-värden för det centraliserade ingripande läget i ITS-fairy med SUMOs ursprungliga system och SUMO under störningsförhållanden i en trevägskorsning. Resultatet visar att T2C-värdena minskar vid sämre förarbeteende. SUMOs ursprungliga kollisionsundvikande funktionalitet ger samma resultat som ITS-fairy, med undantag för enstaka tillfälle. SUMOs system under störningsförhållanden ger osäkra T2C-värden, som resulterar i krock för de två mest riskfyllda förarna. Medelvärdet för ITS-fairy ingripanden är genomgående samma för alla förarbeteenden genom testet.

I Figur 6.2 visas hur olika uppdateringsfrekvenser, 0.5 till 5  $Hz$ , påverkar T2C-värden i trevägskorsningen under störningsförhållanden. Vid 0.5  $Hz$  missas kritisk information i två förarbeteenden och leder till krock på en av de. Kritisk information tappas i några av lägena för 2  $Hz$  och 5  $Hz$  men kollisionsrisken upptäcks. Vi ser även att medelvärdet av antalet ITS-fairy ingripanden sjunker för den mest kritiska föraren, på grund av krocken som sker vid 0.5  $Hz$ .



**Figur 6.1:** Graf trevägskorsning. Experiment för T2C-värden för det centraliserade ingripande läget i ITS-fairy, när SUMOs kollisionsundvikande system utsetts för störningsförhållanden.

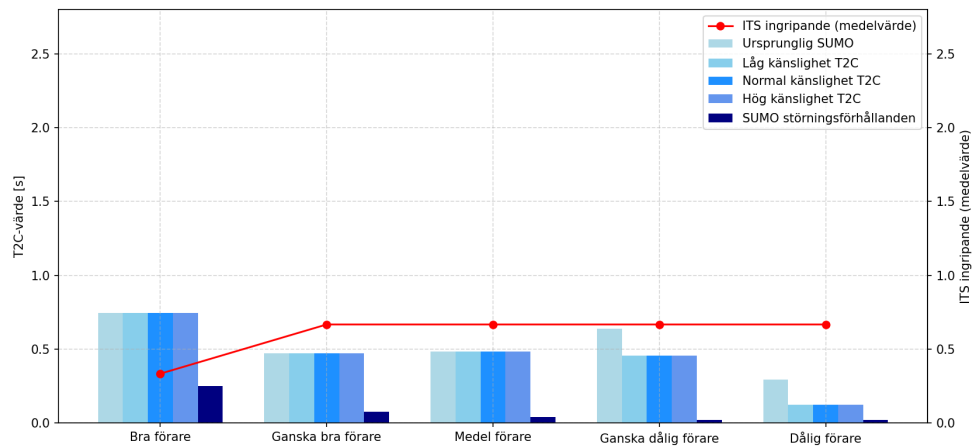


**Figur 6.2:** Graf trevägskorsning. Experiment för hur uppdateringsfrekvensen för fordonsinformation påverkar systemets säkerhet.

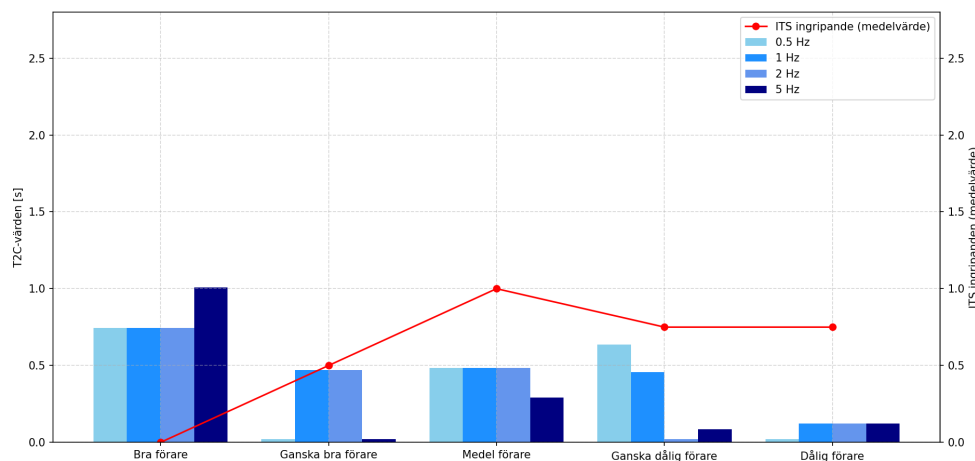
Likt scenariet ovan, visar Figurerna 6.3 och 6.4 samma experiment, men i vägscenariot fyrvägskorsning.

Figur 6.3 visar att resultatet för T2C-värdena går nedåt desto sämre föraren blir. SUMOs ursprungliga kollisionsundvikande funktionalitet ger samma resultat som det ingripande läget i ITS-fairy för de första tre förarbeteendena, men för de sista två är det högre. SUMOs system under störningsförhållanden ger osäkra T2C-värden, och resulterar i krock för de två mest riskfyllda förarna. Medelvärdet för ITS-fairy ingripanden ökar efter den säkraste föraren, men håller sig sedan på en jämn nivå.

För det andra experimentet som visas i Figur 6.4, kan det ses att vid 0.5 Hz upptäcks kritisk information för sent vid två tillfällena, vilket leder till en krock i scenariot. Däremot en kollisionsrisk upptäcks även tidigare i ett av förarbeteendena. För 2 Hz tappas kritisk information i de två mest kritiska förarna, där ett av dem leder till krock. För 5 Hz upptäcks en kollision i tidigare skede för den säkraste föraren, men information tappas på de andra fyra och leder till krock på en utav dem.



**Figur 6.3:** Graf fyrvägs korsning. Experiment för T2C-värden för det centraliserade ingripande läget i ITS-fairy, när SUMOs kollisionssundvikande system utsetts för störningsförhållanden.



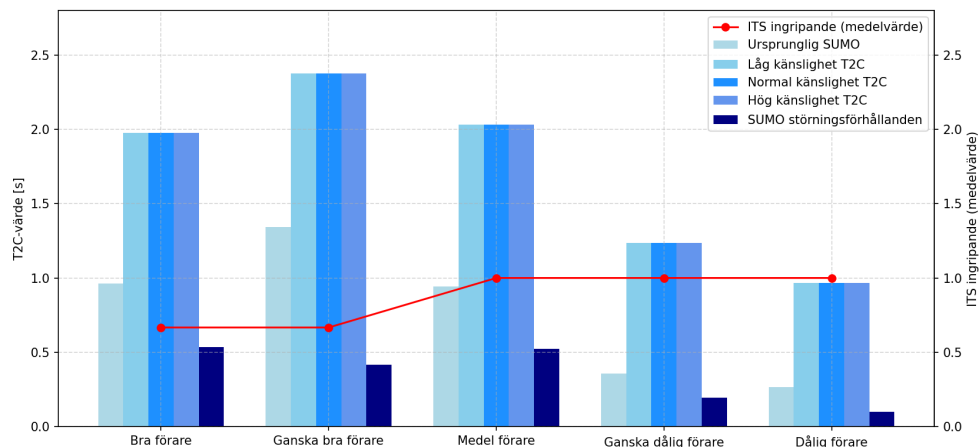
**Figur 6.4:** Graf fyrvägs korsning. Experiment för hur uppdateringsfrekvensen för fordoninformation påverkar systemets säkerhet.

För samma två experiment som ovan men i vägscenariot rondell, visas resultaten i Figurerna 6.5 och 6.6.

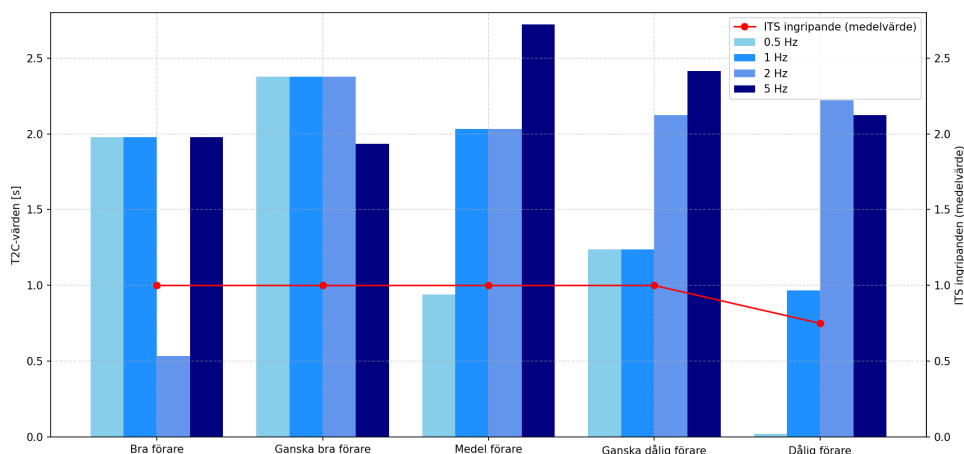
Figur 6.5 visar resultatet att T2C-värdena går nedåt desto mer kritisk föraren blir. SUMOs ursprungliga kollisionssundvikande funktionalitet ger ett lägre T2C-värde för alla förarbeteenden. Dock fortfarande högre än T2C-värdena för SUMOs system under störningsförhållanden. Dessa ger ett osäkert resultat men resulterar aldrig i krock. Medelvärdet för ITS-fairy ingripanden ökar efter de två första förarna, men håller sig sedan på en jämn nivå.

För det andra experimentet som visas i Figur 6.6, kan vi se för 0.5 Hz upptäcks kritisk information försent för tre förarbeteenden, där ett läge leder till krock. För 2 Hz tappas kritisk information i ett läge och leder till senare upptäckt kollisionsrisk. Men upptäcker en kollisionsrisk tidigare än 1 Hz testerna för två lägen. För 5 Hz tappas kritisk information i ett förarbeteende, men upptäcker tre kollisionsrisker

tidigare än 1 Hz testerna.



**Figur 6.5:** Graf rondell. Experiment för T2C-värden för det centraliserade ingripande läget i ITS-fairy, när SUMOs kollisionsundvikande system utsetts för störningsförhållanden.



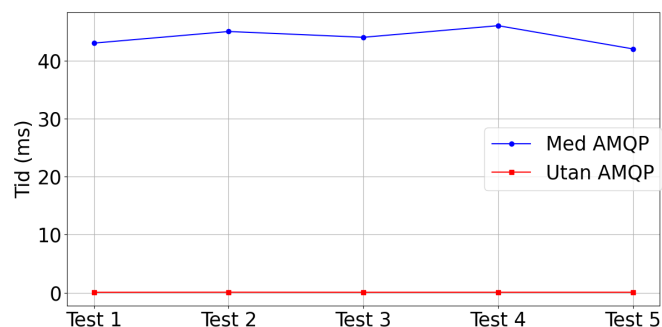
**Figur 6.6:** Graf rondell. Experiment för hur uppdateringsfrekvensen för fordonsinformation påverkar systemets säkerhet.

## 6.2 Situationsmedvetna stödläget

För det situationsmedvetna stödläget av ITS-fairy genomfördes ett test där systemets förmåga att identifiera en fara och skicka ett varningsmeddelande till bakomliggande fordon utvärderades i en verklighetstrogen situation. Tiden det tog att identifiera faran och skicka varningen mäts och presenteras i Tabell 6.1.

**Tabell 6.1:** Tabellen visar resultat från fem test med och utan AMQP.

Test	Med AMQP	Utan AMQP
1	43 ms	85 $\mu$ s
2	45 ms	97 $\mu$ s
3	44 ms	88 $\mu$ s
4	46 ms	90 $\mu$ s
5	42 ms	92 $\mu$ s



**Figur 6.7:** Visualisering av resultat med och utan AMQP.

Resultaten från de utförda testerna indikerar att exekveringstiden är nära 0 sekunder eller försumbar när ingen konflikt identifieras. Detta tyder på att det är själva sändningen av meddelandet som utgör det mest tidskrävande momentet i algoritmen. Därför gjordes även tester utan AMQP för att visa tiden att köra algoritmen utan att skicka något meddelande, detta tog ungefär 90  $\mu$ s som demonstreras i Tabell 6.1.

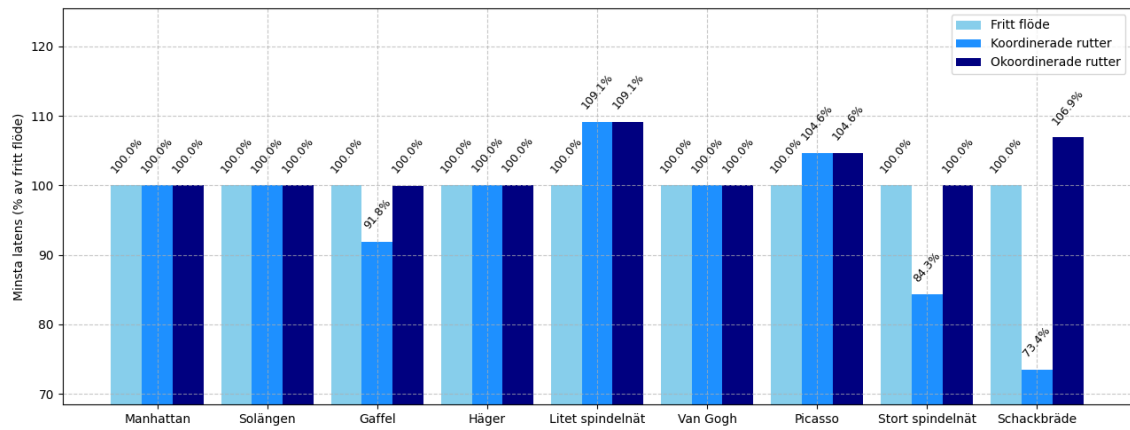
### 6.3 Kollektiva ruttplaneringssystemet

Värdet som undersöks är latensen mellan SUMOs egna ruttplanering och de centralt planerade rutterna av C-CBS. Dessa två värden jämförs med fallet då varje fordon färdas utan övrig trafik, när fordonen har fritt flöde.

De fyra parametrarna kring latensen som undersöktes var: den minsta latensen för ett fordon, den genomsnittliga latensen för alla fordon, den maximala latensen för ett fordon och variansen. För att underlätta avläsningen av graferna har den nedre delen av grafen begränsats till 5% under de minsta uppmätta värdet.

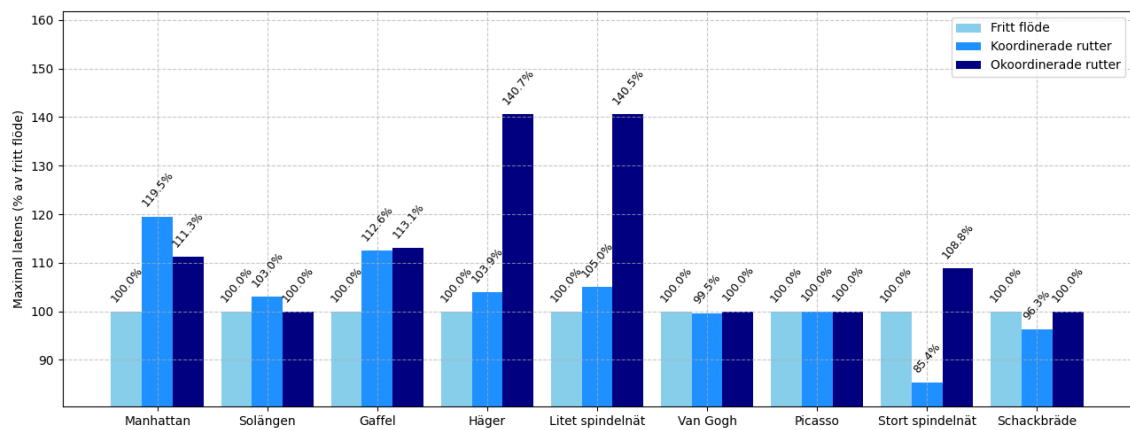
Figur 6.8 visar den kortaste tiden det tog för ett fordon att nå sitt mål som en procentdel av den minsta tiden för fordon i fritt flöde.

## 6. Resultat



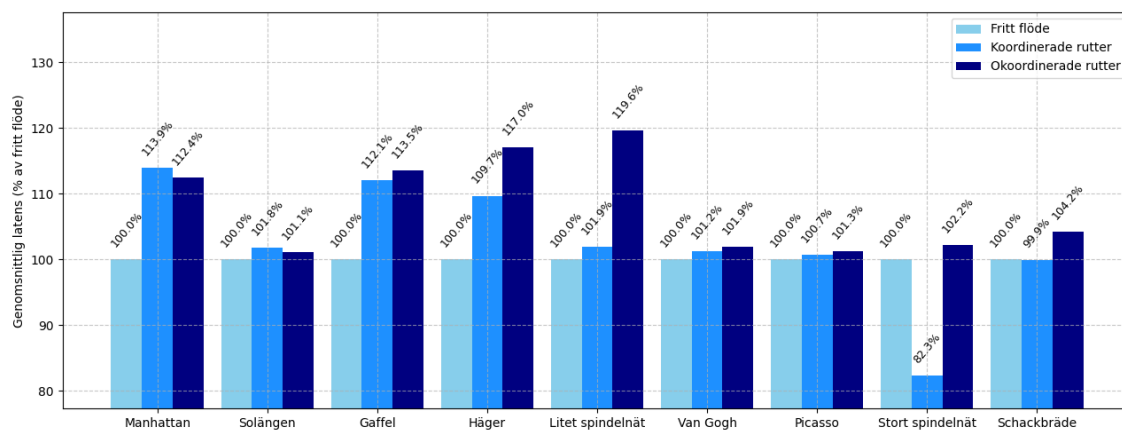
**Figur 6.8:** Minsta latensen i olika scenarier

Den andra grafen, Figur 6.9, visar den maximala tiden för ett fordon att nå sitt mål som en procentandel av den maximala tiden i fritt flöde.



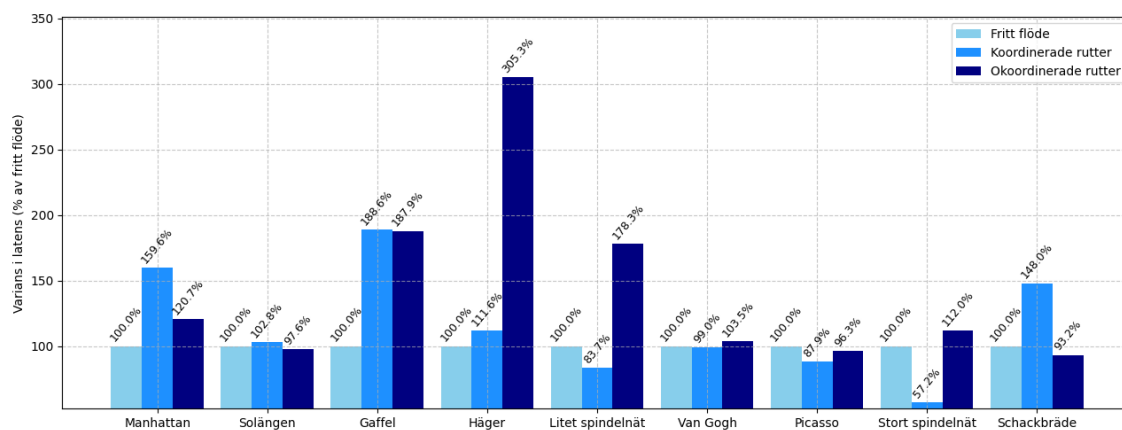
**Figur 6.9:** Maximala latensen i olika scenarier

I Figur 6.10 visar den genomsnittliga tiden det tog för ett fordon att nå sitt mål, även här som procent av fallet då fordonen färdades i fritt flöde.



**Figur 6.10:** Genomsnittliga latensen för olika scenarier

Den sista grafen 6.11 visar variansen mellan de olika tiderna för alla fordon. Som en baslinje användes även här fordonen i fritt flöde.



**Figur 6.11:** Variansen i latensen för olika scenarier



# 7

## Diskussion och slutsats

I följande kapitel diskuteras resultaten från testerna. Vidare diskuteras samhälleliga och etiska aspekter utifrån detta projekt, samt framtida utvecklingspotential. Till sist dras slutsatser utifrån resultatet.

### 7.1 Diskussion

I de tre genomförda trafik-scenarierna för det centraliserade ingripande läget, uppnås nästan lika höga T2C-värden som SUMOs inbyggda kollisionsundvikande system utan kommunikations-störningar. Detta beror på att det kollisionsundvikande systemet behöver göra prediktions-beräkningar på alla fordon, medan SUMOs interna system håller fordonen på ett säkert avstånd för att undvika kollisioner. Detta kan göra att SUMO ibland får ett högre T2C-värde på grund av den konstanta informations-tillgången, medan det ingripande läget har lika höga eller lite lägre T2C-värde, men med en bättre kontext om trafiken, vilket ökar funktionaliteten i verkliga miljöer. Däremot vid rondeller och kommunikations-störningar har det ingripande läget ett högre T2C-värde, vilket betyder att systemet har en större säkerhetsmarginal och kan säkerställa säkerheten.

Dessutom kan man se en korrelation mellan ingripningar av ITS-fairy och vårdslösare förare. Detta beror på att systemet identifierar lägre T2C- och S2C-värden, vilket indikerar ökad kollisionsrisk jämfört med de säkrare förarna som generellt sett har höga värden. Tröskelvärdena för T2C och S2C har en direkt inverkan på systemets känslighet och förmåga att upptäcka och skicka varningar i rätt tid. Höga tröskelvärden leder till tidigare varningar, vilket ökar chansen att undvika kollision, men även risken för falska alarm och onödiga ingripanden. Lägre tröskelvärden minskar antalet varningar och ingrepp, men innebär också en ökad risk för sen reaktionstid i kritiska situationer. Detta understryker betydelsen av ett balanserat tröskelvärde som gör systemet tillräckligt responsivt, utan att belasta trafiken med felaktiga signaler.

Samtidigt visar testerna att systemets robusthet minskar främst vid höga uppdateringsfrekvenser. En hög uppdateringsfrekvens innebär att ITS-fairy har en mer precis information kring fordon i närheten, vilket ökar precisionen i kollisionsbedömningen i snabba och dynamiska lägen. Däremot belastas kommunikationskanalen i högre grad och meddelanden riskerar att gå förlorade. Detta visar på vikten av en balanserad uppdateringsfrekvens som garanterar säkerheten utan att kompromissa

med kommunikationens tillförlitlighet.

Utifrån resultatet av det situationsmedvetna läget av ITS-fairy är det möjligt att identifiera faror i S-LDM och vidarebefordra dessa till de fordon som berörs. Det framgår också att kommunikationen är den mest tidskrävande delen av algoritmen. Om ingen fara identifieras kan funktionen däremot köras med hög frekvens, vilket möjliggör snabbare identifiering av nya faror.

Det förekom dock fall där felaktiga faror upptäcktes, exempelvis om den stillastående bilen står utmed vägen fast i en säker position utan att vara fara för trafiken. Detta berodde på bristande funktionalitet för att avgöra om en detektion verkligen utgjorde en fara. För att förbättra tillförlitligheten krävs ett mer avancerat system som beaktar fler parametrar i sin bedömning. Målet att förbättra uppfattningen av trafiksituationen uppnåddes därmed inte fullt ut.

När det gäller den kooperativa ruttplaneringen är resultatet inte entydigt. Systemet visar inte på förbättringar i varje scenario, vilket är fullt rimligt både med anseende på de förenklingar som gjorts när systemet implementerades, samt kartornas och rutternas utformning. I 7 av 9 scenarier ser vi förbättringar gällande den genomsnittliga latensen jämfört med fallet utan koordinerade rutter. Detta visar att systemet absolut har en möjlighet att effektivisera trafikflöden i många fall.

I scenarierna ”Gaffel”, ”Stort spindelnät”, ”Schackbräde” och ”Van Gogh” (se figur 5.1) ser vi även förbättringar jämfört med fritt flödande fordon, vilket tyder på att de planerade rutterna inte är optimala. Om rutterna skulle varit optimala borde det inte vara möjligt att få bättre resultat i avseende på minsta latens, genomsnittlig latens eller maximal latens. Detta eftersom den enda faktorn som bör spela någon roll är övrig trafik och därmed bör fordon i fritt flöde generera den allra minsta latensen, den minsta genomsnittliga latensen och den minsta maximala latensen. Trots detta visar resultaten på viss förbättring i dessa avseenden.

I 6 av 9 scenarier visar resultaten även på att de koordinerade rutterna skapar mer rättvisa för alla förare då variansen i dessa fall är mindre. Denna mätning visar på att det finns mindre skillnader i fordonens latens för ett och samma scenario. På så sätt skapas en till synes mer rättvis trafiksituation där fordon får mer liknande latens.

## 7.2 Samhälleliga och etiska aspekter

Det kollisionsundvikande system vi utvecklat syftar till att öka trafiksäkerheten, men väcker samtidigt etiska frågor kring hur systemet bör agera i farliga trafiksituationer. Beroende på det valda etiska ramverket påverkar det systemets prioriteringar i olika situationer. Ett utilitaristiskt perspektiv skulle innebära att rädda flest liv även om det kan skada föraren, medan ett etiskt egoistiskt perspektiv hade fokuserat på förarens säkerhet framför andra trafikanter. Valet av ett etiskt ramverk har därmed en direkt påverkan på hur systemet hanterar olycksrisker som i sin tur kan ha stora

implikationer.

Utöver säkerhetsfrågan medför MAPF-systemet till miljömässiga lösningar som bidrar till jämnare trafikflöden, minskad köbildning och växthusgasutsläpp. Detta stödjer och kan hjälpa länder nå målet att minska utsläppen av fossila bränslen enligt Parisavtalet [50]. Därmed bidrar projektet till både ökad trafiksäkerhet och hållbar utveckling när klimatförändringarna är en av de största utmaningarna för planeten och en hållbarare framtid [51].

## 7.3 Framtida arbeten

Under projektets gång har flera områden identifierats där systemet kan byggas ut eller förbättras. Dessa framtida arbeten syftar till att förbättra systemets användbarhet i olika scenarier och flexibilitet. Genom att analysera nuvarande begränsningar och möjliga förbättringar skapas en grund för vidareutveckling av systemet.

### 7.3.1 Centraliserade ingripandeläget

Inom det centraliserade ingripande läget kan framtida arbeten fokusera på att utvärdera systemets prestanda i olika trafiksituationer. Detta kan involvera fler förare i trafikscenarierna som kommer från olika håll med olika körbeteende, vilket bättre efterliknar en stadsmiljö. För att detta ska vara möjligt krävs en återhämtningsfunktion som gör att fordon som stannat eller saktat ner på grund av identifierad risk, ökar sin hastighet då T2C- och S2C-trösklarna inte längre underskrids. Detta skulle bidra till både trafiksäkerheten samt ett effektivare trafikflöde. Det kollisionsundvikande systemet skulle dessutom kunna integreras med MAPF-systemet, vilket skulle möjliggöra en bättre koordinering mellan fordon som kan förbättra säkerhetssystemet och trafikflödet.

Det finns även potential att förbättra den nuvarande kollisionsdetekteringen i scenarier med begränsad sikt, exempelvis då fordon eller hinder är runt hörn. Just nu hanterar SUMOs inbyggda system dessa scenarier effektivare än det ingripande läget. SUMO stödjer även simulering av fotgängare, vilket ger möjligheten att utvärdera om det centraliserade ingripande läget är snabbare eller effektivare än SUMOs inbyggda system.

### 7.3.2 Situationsmedvetna stödläget

Det finns brister i denna implementering av konfliktidentifiering där den är mycket specifikt anpassad för att fungera på en motorväg där fordon passerar snabbt förbi. En bättre lösning för framtida projekt skulle vara att använda positionen på det stillastående fordonet och skapa en metod som avgör om fordonet befinner sig på en motorväg eller inte. Vidare skulle det också behövas funktionalitet för att skicka CPM-meddelanden till S-LDM och använda dem för att förbättra konfliktidentifieringen. CPM-meddelanden används i expansionen för den högupplösta fordonssimulatorens CARLA, av ms-van3t för att visualisera bilarnas uppfattning av miljön, men de används inte i S-LDM.

### 7.3.3 Kollektiva ruttplaneringssystemet

Det finns i nuläget stor potential till vidareutveckling av system relaterat till trafikplanering. Konceptet att använda en MAPF-algoritm för att planera effektiva rutter kan vidareutvecklas genom att lägga till fler mekanismer i syfte att kontrollera fordonen. Detta inkluderar VLT och digitala hastighetsbegränsningar. Fler MAPF-algoritmer bör även utvärderas, alternativt kan C-CBS-algoritmen som använts modifieras för att ta hänsyn till fordonens dynamik och geometri. För att förbättra den nuvarande implementeringen kan även fallet då sökningen sker flera gånger under ett givet scenario utvärderas. Detta kräver att grafen utökas till en högre detaljnivå då korsningars interna vägar hittills har ignorerats.

## 7.4 Slutsats

För att besvara de frågor som formulerades i kapitel 1.2 har vi utvecklat och utvärderat ett system för kooperativ perception och ett ruttplaneringssystem i en simulerad trafikmiljö. Genom simuleringar och analys av olika trafiksituationer, såsom rondeller och korsningar, har vi kunnat dra följande slutsatser:

Arbetet visar att ITS-fairy tydligt kan bidra till en mer fullständig och enhetlig uppfattning av trafiksituationen genom att samla in och sprida kritiska situationer i realtid. Det situationsmedvetna stödläget möjliggör att faror som inte direkt upptäckts av ett fordon ändå kan förmedlas, dock visar det att kommunikationen är det mest tidskrävande momentet. Vidare påvisar det centraliserade ingripandeläget att effektiva säkerhetsåtgärder kan vidtas, systemet stoppar fordonen i 100 % av de testade farliga situationerna, vilket är en signifikant förbättring jämfört med SUMO:s inbyggda skydd. Avslutningsvis visar experimenten att det är möjligt att implementera ett koordinerat ruttplaneringssystem som minskar restider och ökar flödet i vissa nätverkstyper. Särskilt i stadsliknande miljöer ger det kollektiva ruttplaneringssystemet märkbara fördelar.

## Användning av AI

I detta arbete har AI-verktyg såsom ChatGPT och GitHub Copilot använts främst för förståelse och felsökning av befintlig kod. Även enklare kodstycken har genererats. Verktögen har också använts vid rapportskrivningen för att få hjälp med flyt och till viss del struktur i skrivandet samt genererat enskilda figurer till rapporten. Däremot har stor hänsyn till verktygens risk att generera inkorrekt och missvisande tagits. Vid varje tillfälle då AI använts har därför resultaten noggrant granskats.



# Litteraturförteckning

- [1] W. H. Organization, *Global status report on road safety 2023: summary*. World Health Organization, 2023, hämtad: 16 Maj 2025.
- [2] S. K. Ahmed, M. G. Mohammed, S. O. Abdulqadir, R. G. A. El-Kader, N. A. El-Shall, D. Chandran, M. E. U. Rehman, och K. Dhama, “Road traffic accidental injuries and deaths: A neglected global health issue,” *Health science reports*, vol. 6, nr 5, s. e1240, 2023, hämtad: 21 Maj 2025.
- [3] H. Ritchie, “Cars, planes, trains: where do co emissions from transport come from?” *Our World in Data*, 2020, hämtad: 5 februari 2025. [Online]. Tillgänglig: <https://ourworldindata.org/co2-emissions-from-transport>
- [4] D. Papadimitriou, I. Koutla, och P. Duysinx, “Future move - a review of the main trends in the automotive sector at horizon 2030 in the great region,” Online, Tekn. rapp., 2022, hämtad: 30 januari 2025. [Online]. Tillgänglig: [https://orbi.uliege.be/bitstream/2268/291600/1/whitepaper\\_V1.1.pdf](https://orbi.uliege.be/bitstream/2268/291600/1/whitepaper_V1.1.pdf)
- [5] Statista, “Size of the global autonomous car market in 2024, with a forecast through 2029 (in billion u.s. dollars) [graph],” <https://www.statista.com/statistics/428692/projected-size-of-global-autonomous-vehicle-market-by-vehicle-type/>, okt. 2024, hämtad: 18 maj 2025.
- [6] J. Dong, J. Wang, och D. Luo, “Impact of connected and autonomous vehicles on traffic safety of mixed traffic flow: from the perspective of connectivity and spatial distribution,” *Transportation Safety and Environment*, vol. 4, nr 3, s. tdac021, 08 2022, hämtad: 18 maj 2025. [Online]. Tillgänglig: <https://doi.org/10.1093/tse/tdac021>
- [7] Y. Wen, Y. Wang, Z. Zhang, J. Wu, L. Zhong, M. Papageorgiou, och P. Zheng, “Effects of connected autonomous vehicles on the energy performance of signal-controlled junctions,” *Sustainability*, vol. 15, nr 7, s. 5672, 2023, hämtad: 18 maj 2025. [Online]. Tillgänglig: <https://doi.org/10.3390/su15075672>
- [8] C. Creß och A. C. Knoll, “Intelligent transportation systems with the use of external infrastructure: A literature survey,” *arXiv preprint arXiv:2112.05615*, 2021, hämtad: 18 maj 2025.
- [9] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, och E. Wießner, “Microscopic traffic simulation using sumo,” i *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018, hämtad: 15 Maj 2025. [Online]. Tillgänglig: <https://elib.dlr.de/124092/>
- [10] F. Raviglione, C. R. Carletti, M. Malinverno, C. Casetti, och C. Chiasserini, “ms-van3t: An integrated multi-stack framework for virtual validation of v2x communication and services,” *Computer Communications*, vol.

- 217, s. 70–86, 2024, hämtad: 4 februari 2025. [Online]. Tillgänglig: <https://www.sciencedirect.com/science/article/pii/S0140366424000227>
- [11] C. C. F. S. G. Y. F. V. C.M. Risma Carletti, F. Raviglione, “S-LDM: Server local dynamic map for 5G-based centralized enhanced collective perception.”
- [12] A. Jame, M. Rapelli, och C. Casetti, “Reducing pollutant emissions through virtual traffic lights,” *Computer Communications*, vol. 188, s. 167–177, 2022. [Online]. Tillgänglig: <https://doi.org/10.1016/j.comcom.2022.03.018>
- [13] A. Casimiro, E. Ekenstedt, och E. M. Schiller, “Membership-based manoeuvre negotiation in autonomous and safety-critical vehicular systems,” *CoRR*, vol. abs/1906.04703, 2019, hämtad: 2025-05-03. [Online]. Tillgänglig: <http://arxiv.org/abs/1906.04703>
- [14] M. Malinverno, “Safety applications and measurement tools for connected vehicles,” Doktorsavhandling, Politecnico di Torino, Turin, Italy, February 2021, hämtad 19 maj 2025. [Online]. Tillgänglig: [https://iris.polito.it/retrieve/e384c433-5c92-d4b2-e053-9f05fe0a1d67/conv\\_thesis.pdf](https://iris.polito.it/retrieve/e384c433-5c92-d4b2-e053-9f05fe0a1d67/conv_thesis.pdf)
- [15] G. Avino, “Development and performance evaluation of network function virtualization services in 5g multi-access edge computing,” Doktorsavhandling, Politecnico di Torino, Turin, Italy, 2021, hämtad: 18 maj 2025. [Online]. Tillgänglig: [https://iris.polito.it/retrieve/e384c433-06e4-d4b2-e053-9f05fe0a1d67/phd-thesis\\_giuseppeavino.pdf](https://iris.polito.it/retrieve/e384c433-06e4-d4b2-e053-9f05fe0a1d67/phd-thesis_giuseppeavino.pdf)
- [16] A. Andreychuk, K. Yakovlev, D. Atzmon, och R. Stern, “Multi-agent pathfinding with continuous time,” 2019, hämtad: 12 maj 2025. [Online]. Tillgänglig: <https://arxiv.org/abs/1901.05506>
- [17] V. Savic, E. M. Schiller, och M. Papatriantafilou, “Distributed algorithm for collision avoidance at road intersections in the presence of communication failures,” i *IEEE Intelligent Vehicles Symposium, IV 2017, Los Angeles, CA, USA, June 11-14, 2017*. IEEE, 2017, s. 1005–1012. [Online]. Tillgänglig: <https://doi.org/10.1109/IVS.2017.7995846>
- [18] T. Petig, E. M. Schiller, och J. Suomela, “Changing lanes on a highway,” i *18th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems, ATMOS 2018, August 23-24, 2018, Helsinki, Finland*, ser. OASICs, R. Borndörfer och S. Storandt, Red., vol. 65. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, s. 9:1–9:15. [Online]. Tillgänglig: <https://doi.org/10.4230/OASICs.ATMOS.2018.9>
- [19] O. Morales-Ponce, E. M. Schiller, och P. Falcone, “How to stop disagreeing and start cooperating in the presence of asymmetric packet loss,” *Sensors*, vol. 18, nr 4, s. 1287, 2018. [Online]. Tillgänglig: <https://doi.org/10.3390/s18041287>
- [20] —, “Cooperation with disagreement correction in the presence of communication failures,” i *17th International IEEE Conference on Intelligent Transportation Systems, ITSC 2014, Qingdao, China, October 8-11, 2014*. IEEE, 2014, s. 1105–1110. [Online]. Tillgänglig: <https://doi.org/10.1109/ITSC.2014.6957835>
- [21] C. Casetti, C. Chiasserini, F. Dressler, A. Memedi, D. Gasco, och E. M. Schiller, “Ai/ml-based services and applications for 6g-connected and autonomous vehicles,” *Comput. Networks*, vol. 255, s. 110854, 2024. [Online]. Tillgänglig: <https://doi.org/10.1016/j.comnet.2024.110854>

- 
- [22] A. Casimiro, O. Morales-Ponce, T. Petig, och E. M. Schiller, “Vehicular coordination via a safety kernel in the gulliver test-bed,” i *34th International Conference on Distributed Computing Systems Workshops (ICDCS 2014 Workshops), Madrid, Spain, June 30 - July 3, 2014*. IEEE Computer Society, 2014, s. 167–176. [Online]. Tillgänglig: <https://doi.org/10.1109/ICDCSW.2014.25>
- [23] C. Berger, O. Morales-Ponce, T. Petig, och E. M. Schiller, “Driving with confidence: Local dynamic maps that provide los for the gulliver test-bed,” i *Computer Safety, Reliability, and Security - SAFECOMP 2014 Workshops: ASCoMS, DECSoS, DEVVARTS, ISSE, ReSA4CI, SASSUR. Florence, Italy, September 8-9, 2014. Proceedings*, ser. Lecture Notes in Computer Science, A. Bondavalli, A. Ceccarelli, och F. Ortmeier, Red., vol. 8696. Springer, 2014, s. 36–45. [Online]. Tillgänglig: [https://doi.org/10.1007/978-3-319-10557-4\\_6](https://doi.org/10.1007/978-3-319-10557-4_6)
- [24] D. Leggett. Explainer: When will we have driverless vehicles? Just Auto. Hämtad: 21 Maj 2025. [Online]. Tillgänglig: <https://www.just-auto.com/features/explainer-when-will-we-have-driverless-vehicles/>
- [25] R. Zhang, F. Schmutz, K. Gerard, A. Pomini, L. Basseto, S. B. Hassen, A. Ishikawa, I. Ozgunes, och O. Tonguz, “Virtual traffic lights: System design and implementation,” i *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. IEEE, 2018, s. 1–5, hämtad: 15 Maj 2025.
- [26] M. Ferreira, R. Fernandes, H. Conceição, W. Viriyasitavat, och O. K. Tonguz, “Self-organized traffic control,” i *Proceedings of the seventh ACM international workshop on VehiculAr InterNETworking*, 2010, s. 85–90, hämtad: 13 Maj 2025.
- [27] R. Stern, *Multi-Agent Path Finding – An Overview*, 10 2019, s. 96–115, hämtad: 1 maj 2025.
- [28] R. Stern, N. Sturtevant, A. Felner, S. Koenig, H. Ma, T. Walker, J. Li, D. Atzmon, L. Cohen, T. K. S. Kumar, E. Boyarski, och R. Bartak, “Multi-agent pathfinding: Definitions, variants, and benchmarks,” 2019, hämtad: 1 maj 2025. [Online]. Tillgänglig: <https://arxiv.org/abs/1906.08291>
- [29] R. Diestel, *The Basics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2025, s. 1–35, hämtad: 1 maj 2025. [Online]. Tillgänglig: [https://doi.org/10.1007/978-3-662-70107-2\\_1](https://doi.org/10.1007/978-3-662-70107-2_1)
- [30] Blackberry, “Vehicle-to-Everything (V2X),” 2024, hämtad: 31 januari 2025. [Online]. Tillgänglig: <https://blackberry.qnx.com/en/ultimate-guides/software-defined-vehicle/vehicle-to-everything>
- [31] National Highway Traffic Safety Administration (NHTSA), “Federal Motor Vehicle Safety Standards: V2V Communications,” 2017, hämtad: 30 januari 2025. [Online]. Tillgänglig: <https://www.federalregister.gov/documents/2017/01/12/2016-31059/federal-motor-vehicle-safety-standards-v2v-communications>
- [32] Transportstyrelsen, “Intelligent transport system (its),” 2025, hämtad: 4 februari 2025. [Online]. Tillgänglig: <https://www.transportstyrelsen.se/sv/om-oss/det-har-ar-transportstyrelsen/forskning-och-innovation/its/>
- [33] International Organization for Standardization, “ISO 17438-5: Intelligent transport systems — Reference architecture and terminology — Part 5: Information layer,” 2024, hämtad: 13 februari 2025. [Online]. Tillgänglig: <https://www.iso.org/obp/ui#iso:std:iso:17438:-5:dis:ed-1:v1:en>

- [34] European Telecommunications Standards Institute (ETSI), “ETSI TR 103 562 V2.1.1 (2019-12),” European Telecommunications Standards Institute, Tekn. rapp., 2019, hämtad: 4 februari 2025. [Online]. Tillgänglig: <https://cdn.standards.iteh.ai/samples/53495/5584b341b34c4248bc979396d7f4c839/ETSI-TR-103-562-V2-1-1-2019-12-.pdf>
- [35] —, “Intelligent Transport System (ITS); Vehicular Communications; Basic Set of Applications; Collective Perception Service; Release 2 ,” ETSI, Tekn. rapp. ETSI TS 103 324 V2.1.1, 2021, hämtad: 4 februari 2025. [Online]. Tillgänglig: [https://www.etsi.org/deliver/etsi\\_ts/103300\\_103399/103324/02.01.01\\_60/ts\\_103324v020101p.pdf](https://www.etsi.org/deliver/etsi_ts/103300_103399/103324/02.01.01_60/ts_103324v020101p.pdf)
- [36] ETSI, “Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service,” 2019, hämtad 6 februari 2025. [Online]. Tillgänglig: [https://www.etsi.org/deliver/etsi\\_en/302600\\_302699/30263702/01.04.01\\_60/en\\_30263702v010401p.pdf](https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.04.01_60/en_30263702v010401p.pdf)
- [37] Eiter, Thomas and Füreder, Herbert and Kasslatter, Fritz and Parreira, Josiane Xavier and Schneider, Patrik, “Towards a Semantically Enriched Local Dynamic Map,” 2018, hämtad: 1 februari 2025. [Online]. Tillgänglig: <https://link.springer.com/article/10.1007/s13177-018-0154-x#citeas>
- [38] ETSI, “ETSI TR 102 863 V1.1.1: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM),” 2011, hämtad: 1 februari 2025. [Online]. Tillgänglig: [https://www.etsi.org/deliver/etsi\\_tr/102800\\_102899/102863/01.01.01\\_60/tr\\_102863v010101p.pdf](https://www.etsi.org/deliver/etsi_tr/102800_102899/102863/01.01.01_60/tr_102863v010101p.pdf)
- [39] Y. M. X. G. o. H. L. Shen Li, Qiaojun Xiang, “Crash risk prediction modeling based on the traffic conflict technique and a microscopic simulation for freeway interchange merging areas,” *International Journal of Environmental Research and Public Health*, vol. 13, nr 11, 2016. [Online]. Tillgänglig: <https://pmc.ncbi.nlm.nih.gov/articles/PMC5129367/>
- [40] N. S. Council, “Types of crashes,” 2023, hämtad: 2025-05-02. [Online]. Tillgänglig: <https://injuryfacts.nsc.org/motor-vehicle/overview/type-of-crash/>
- [41] ETSI, “Etsi ts 102 894-2 v2.1.1 (2022-11) - intelligent transport systems (its); users and applications requirements; part 2: Applications and facilities layer common data dictionary; release 2,” 2022, accessed: 2025-05-02. [Online]. Tillgänglig: [https://www.etsi.org/deliver/etsi\\_ts/102800\\_102899/10289402/02.01.01\\_60/ts\\_10289402v020101p.pdf](https://www.etsi.org/deliver/etsi_ts/102800_102899/10289402/02.01.01_60/ts_10289402v020101p.pdf)
- [42] D. C. Selvaraj, C. Vitale, T. Panayiotou, P. Kolios, C. F. Chiasserini, och G. Ellinas, “Edge-assisted ml-aided uncertainty-aware vehicle collision avoidance at urban intersections,” *IEEE Transactions on Intelligent Vehicles*, 2023, hämtad: 29 April 2025.
- [43] K. Okumura, M. Machida, X. Défago, och Y. Tamura, “Priority inheritance with backtracking for iterative multi-agent path finding,” *Artificial Intelligence*, vol. 310, s. 103752, 2022, hämtad: 19 maj 2025.
- [44] T. Darwish och K. A. Bakar, “Traffic density estimation in vehicular ad hoc networks: A review,” *Ad Hoc Networks*, vol. 24, s. 337–351, 2015, hämtad: 19 Maj 2025.
- [45] M. Bertoncetto, C. Martens, T. Möller, och T. Schneiderbauer, “Unlocking the full life-cycle value from connected-car data,” McKinsey & Company, Tekn.

- rapp., hämtad: 19 Maj 2025. [Online]. Tillgänglig: <https://www.mckinsey.com/~media/mckinsey/industries/automotive%20and%20assembly/our%20insights/unlocking%20the%20full%20life%20cycle%20value%20from%20connected%20car%20data/unlocking-the-full-life-cycle-value-from-connected-car-data.pdf>
- [46] M. Abdel-Aty och K. Haleem, “Analyzing angle crashes at unsignalized intersections using machine learning techniques,” *Accident Analysis & Prevention*, vol. 43, nr 1, s. 461–470, 2011, hämtad: 18 maj 2025. [Online]. Tillgänglig: <https://doi.org/10.1016/j.aap.2010.10.002>
- [47] L. Li, Z. Zhang, Z.-G. Xu, W.-C. Yang, och Q.-C. Lu, “The role of traffic conflicts in roundabout safety evaluation: A review,” *Accident Analysis & Prevention*, vol. 196, s. 107430, 2024, hämtad: 18 maj 2025. [Online]. Tillgänglig: <https://doi.org/10.1016/j.aap.2023.107430>
- [48] R. van der Horst och J. Hogema, “Time-to-collision and collision avoidance systems,” *ICTCT Workshop Proceedings*, 1994, hämtad: 18 maj 2025. [Online]. Tillgänglig: [https://www.researchgate.net/publication/237807114\\_TIME-TO-COLLISION\\_AND\\_COLLISION\\_AVOIDANCE\\_SYSTEMS](https://www.researchgate.net/publication/237807114_TIME-TO-COLLISION_AND_COLLISION_AVOIDANCE_SYSTEMS)
- [49] SUMO. (2025) Definition of vehicles, vehicle types, and routes. Hämtad: 18 maj 2025. [Online]. Tillgänglig: [https://sumo.dlr.de/docs/Definition\\_of\\_Vehicles%2C\\_Vehicle\\_Types%2C\\_and\\_Routes.html](https://sumo.dlr.de/docs/Definition_of_Vehicles%2C_Vehicle_Types%2C_and_Routes.html)
- [50] Joint Research Centre, “G20 climate strategies insufficient to meet paris agreement goals,” 2025, hämtad: 5 februari 2025. [Online]. Tillgänglig: [https://joint-research-centre.ec.europa.eu/jrc-news-and-updates/g20-climate-strategies-insufficient-meet-paris-agreement-goals-2025-01-31\\_en](https://joint-research-centre.ec.europa.eu/jrc-news-and-updates/g20-climate-strategies-insufficient-meet-paris-agreement-goals-2025-01-31_en)
- [51] Regeringskansliet, “Agenda 2030 – mål 13: Bekämpa klimatförändringarna,” 2022, hämtad 13 maj 2025. [Online]. Tillgänglig: <https://www.regeringen.se/regeringens-politik/globala-malen-och-agenda-2030/agenda-2030-mal-13-bekampa-klimatforandringarna/>

