



CHALMERS



Tekniker för kommunikation mellan arbetsfordon

Utvärdering av DSRC samt C-V2X

Examensarbete inom högskoleingenjörsprogrammet Elektroteknik

TEDD AHLBERG

KARIN KJELLKVIST

EXAMENSARBETE

Tekniker för kommunikation mellan arbetsfordon

Utvärdering av DSRC samt C-V2X

TEDD AHLBERG
KARIN KJELLKVIST



CHALMERS

Institutionen för Rymd-, geo- och miljövetenskap
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2019

Tekniker för kommunikation mellan arbetsfordon
Utvärdering av DSRC samt C-V2X
TEDD AHLBERG
KARIN KJELLKVIST

© TEDD AHLBERG & KARIN KJELLKVIST, 2019.

Institutionen för Rymd-, geo- och miljövetenskap
Chalmers Tekniska Högskola
SE-412 96 Göteborg
Telefon: +46 (0)31-772 1000

Omslag: Piktogram av kommunikation mellan olika arbetsfordon.

Göteborg, Sverige 2019

Communication techniques between construction vehicles

Evaluation of DSRC and C-V2X

TEDD AHLBERG

KARIN KJELLKVIST

Department of Space, Earth and Environment

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2019

Bachelor's thesis

Abstract

In a transport system where vehicles are on their way of becoming autonomous, wireless transmission can be a key factor for success. Information is being shared between vehicles as well as with their surroundings. CPAC Systems in Mölndal, Sweden, are using intervehicular communication as part of their development of control systems for the automotive industry. In the present project, technologies for data transmission are evaluated based on certain criteria, such as latency, range and data rate. The project is a prestudy that lays a foundation for further testing. The evaluated technologies are *Dedicated Short Range Communications* (DSRC) and *Cellular Vehicle-To-Everything* (C-V2X). DSRC is recommended when fast data transfer is required and safety has high priority. C-V2X is recommended for data transfer with long distance between units.

The report is written in Swedish.

Keywords: C-V2X, DSRC, network, V2X, wireless transmission

Sammanfattning

Som ett led i att göra fordon helt eller delvis autonoma, är trådlös informationsöverföring mellan fordon en viktig del. Överföringen sker mellan två eller fler fordon, samt mellan fordon och deras omgivning. På CPAC Systems i Mölndal utvecklas styrsystem till fordonsindustrin. Vissa av företagets projekt kräver fordonskommunikation och där eftersöks lämpliga överföringsteknologier. Detta projekt syftar till att kartlägga marknaden för fordonskommunikation samt att utvärdera teknologier utifrån kriterier såsom latens, räckvidd och datamängd per tidsenhet. Det är en förstudie som ger rekommendationer inför tester av överföring. Resultatet innehåller en utvärdering av teknologierna *Dedicated Short Range Communications* (DSRC) och *Cellular Vehicle-To-Everything* (C-V2X). DSRC rekommenderas där snabb överföring krävs och säkerhet är prioriterat. C-V2X rekommenderas där lång räckvidd krävs.

Nyckelord: C-V2X, DSRC, nätverk, trådlös överföring, V2X

Förord

Detta examensarbete har varit en del av den sista terminen på programmet Elektroteknik högskoleingenjör, på Chalmers Tekniska Högskola. Utbildningen omfattar 180 högskolepoäng och av dessa utgör examensarbetet 15 poäng. Projektet har ägt rum hos CPAC Systems i Mölndal, ett företag som utvecklar styrsystem till fordon.

Stort tack till CPAC som gav oss ett fantastiskt bemötande från första början. Vi har fått mycket stöd och vägledning från våra två handledare Oscar Bodell och Jacob Brickner. Även andra personer på företaget har bidragit med kunskap om V2X, däribland Christian Eggertsen. Detta har hjälpt oss framåt i vårt arbete och vi uppskattar det mycket.

Vi vill också rikta ett varmt tack till Arto Heikkilä, lektor i flera kurser vi läst samt programansvarig för det treåriga elektroprogrammet. Med erfarenhet och engagemang har han som examinator guidat oss genom projektet.

Tedd Ahlberg & Karin Kjellkvist, Göteborg, maj 2019

Förkortningar

Tabell 0.1: Förteckning över de förkortningar som förekommer i rapporten. Tabell 2.1 innehåller ytterligare förkortningar som inte listas här.

AM	Amplitude Modulation (Amplitudmodulering)
BPSK	Binary Phase Shift Keying
C-V2X	Cellular Vehicle-To-Everything
CAN	Central Area Network
CC-BY-SA	Common Creative Attribution Share-Alike
CCH	Control Channel
CDM	Code-Division Multiplexing
CPU	Central Processing Unit
DSB	Double Sideband
DSRC	Dedicated Short Range Communications
ETSI	European Telecommunications Standards Institute
FDM	Frequency-Division Multiplexing
FM	Frequency Modulation (Frekvensmodulering)
GSM	Global System for Mobile Communications
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IMAP	Internet Message Access Protocol
ISM	Industrial, Scientific and Medical
ITU	International Telecommunication Union
LLC	Logical Link Control
LOS	Line Of Sight
LTE	Long-Term Evolution
LTE-A	Long-Term Evolution Advanced
MAC	Media Access Control
NLOS	No Line Of Sight
NVRAM	Non-Volatile Random Access Memory
OBU	On-Board Unit
OFDM	Orthogonal Frequency Division Multiplexing
OSI	Open System Interconnection
PDU	Protocol Data Unit
PLR	Packet Loss Rate
PM	Phase Modulation (Fasmodulering)
PSK	Phase Shift Keying
PTS	Post- och Telestyrelsen
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RAM	Random Access Memory
ROM	Read-Only Memory
RSU	Road-Side Unit
RTT	Round-Trip Time
SCH	Service Channel

SNR	Signal to Noise Ratio
SSB	Single Sideband
TCP	Transmission Control Protocol
TDM	Time-Division Multiplexing
UDP	User Datagram Protocol
UV	Ultraviolet strålning
V2I	Vehicle-To-Infrastructure
V2N	Vehicle-To-Network
V2P	Vehicle-To-Pedestrian
V2V	Vehicle-To-Vehicle
V2X	Vehicle-To-Everything
VANET	Vehicular Ad-Hoc Network
VSF	Vestigial Sideband
WLAN	Wireless Local Area Network

Innehåll

1	Inledning	1
1.1	Bakgrund	1
1.2	Syfte	2
1.3	Avgränsningar	2
1.4	Precisering av frågeställningen	3
1.5	Rapportens disposition	3
2	Teoretisk bakgrund	5
2.1	Trådlös kommunikation	5
2.1.1	Sändare och mottagare	5
2.1.2	Vågornas resa genom luften	5
2.1.3	Analoga moduleringsmetoder	7
2.1.4	Digitala moduleringsmetoder	9
2.1.4.1	PSK	9
2.1.4.2	OFDM	12
2.1.5	Elektromagnetiskt spektrum	12
2.1.6	Bandbredd	14
2.1.7	Multiplexering	15
2.2	Nätverk	15
2.2.1	OSI-modell	15
2.2.1.1	Fysiska lagret	16
2.2.1.2	Datalänkslagret	16
2.2.1.3	Nätverkslagret	17
2.2.1.4	Transportlagret	17
2.2.1.5	Sessionslagret	17
2.2.1.6	Presentationslagret	17
2.2.1.7	Applikationslagret	18
2.2.2	Router	18
2.2.3	CAN	19
2.3	V2X	20
2.3.1	DSRC	21
2.3.2	C-V2X	22
3	Metod	25
3.1	Test av RTT med V2X-enheter	25

4	Resultat	27
4.1	Relevanta studier av V2X	27
4.1.1	Utveckling av DSRC-system	27
4.1.2	Jämförelse av DSRC och C-V2X med avseende på räckvidd . .	27
4.1.3	Jämförelse av DSRC och C-V2X under olika körhastigheter . .	28
4.2	Eget test av V2X-enheter	29
4.3	Sammanställning av produkter	31
5	Slutsatser och diskussion	33
5.1	Rekommendation av tekniker utifrån frågeställning	33
5.2	Diskussion kring resultatets kvalitet	34
5.3	V2X ur ett etiskt och hållbarhetsperspektiv	34
5.4	Förslag till fortsatta studier	34
	Referenser	37
A	Grafer från V2X-test	I
B	Testkod	V

1

Inledning

I inledningskapitlet förklaras bakgrunden till att detta projekt genomförs. Därefter följer en genomgång av uppdragets syfte och mål, samt omfattningen av arbetet.

1.1 Bakgrund

Detta projekt gjordes på CPAC Systems i Mölndal. De utvecklar till största del styrsystem till fordon och deras kunder inkluderar olika delar av Volvo. Vissa av deras projekt behandlar trådlös kommunikation mellan arbetsfordon. Denna sorts kommunikation mellan två eller fler fordon kallas *Vehicle To Vehicle* (V2V). Vårt arbete fokuserar på två av dessa projekt, därför att bättre prestanda i kommunikationen är önskvärd. *Projekt 1*: Det första projektet innefattar en grupp med arbetsfordon som kör i formation, med små avstånd mellan fordonen. Körningen ska ske helt eller delvis autonomt. Detta ställer stora krav på säkerhet, som fås av snabb och tillförlitlig kommunikation. Därför behöver CPAC en produkt som klarar detta. *Projekt 2*: Det andra projektet gäller en grupp arbetsfordon som är utspridda över en större yta. Fordonen behöver kunna dela positionsdata med varandra och skicka produktionsdata till en central enhet. Här krävs produkter med lång räckvidd som kan hantera större mängd data. När det gäller överföring mellan fordon och alla godtyckliga enheter de kan kommunicera med (såsom bilar, nätverk, infrastruktur) har detta samlingsnamnet *Vehicle-To-Everything* (V2X), se Avs. 2.3. På uppdrag av CPAC gjordes en undersökning av V2X-marknaden, med syfte att sammanställa lämpliga V2X-teknologier och produkter. Arbetet är en förstudie som förbereder för praktiska tester av produkter. I denna rapport kommer främst beteckningen V2X användas, även när det gäller specifikt V2V. Begreppen “överföring” och “kommunikation” är i sammanhanget i princip synonyma och kommer användas växelvis.

1.2 Syfte

Företaget CPAC önskar förbättring av den trådlösa kommunikationen mellan arbetsfordon inom två olika projekt där de arbetar med automatisering och styrning.

- *Projekt 1:* En grupp arbetsfordon kör tätt i formation, liknande en konvoj. Fordonen bör kunna köra så nära varandra som möjligt utan att kollidera. Körningen sker helt eller delvis autonomt. Fordonen utbyter positionsdata samt informationsmeddelanden om körningen. Då fordonen är utomhus kan kommunikationen påverkas av regn eller snö.
- *Projekt 2:* Produktion sker i ett större utomhusområde som kan vara ensligt beläget och kan i vissa fall sakna mobiltäckning. Arbetsfordonen befinner sig på långa avstånd från varandra. Insamling av positions- och produktionsdata sker. Miljön kan innehålla både damm och regn/snö. Terräng och utrustning kan komma i vägen för den trådlösa överföringen.

Examensarbetet syftar till att kartlägga marknaden för V2X och hitta lämpliga teknologier och produkter för dessa scenarion. Vad som är lämpligt beror på vissa kriterier som specificeras i Avs. 1.4. Resultatet blir en förstudie med en översikt över aktuella V2X-teknologier, en sammanställning av lämpliga produkter för överföring samt förslag på hur testning av en sådan produkt kan genomföras.

1.3 Avgränsningar

Under projektet konstrueras ingen egen hårdvara eller mjukvara, utan det är färdiga produkter som finns ute på marknaden som undersöks. Testning av överföring görs mellan två stycken enheter, trots att fler enheter är inblandade i de verkliga situationerna. Det kommer heller inte ske någon storskalig testning, utan bara förslag på framtida tester. Wi-Fi och Bluetooth är överföringstekniker som är vanliga i dagens bilar, de är uteslutna ur detta projekt. Även aspekten kring eventuella säkerhetsrisker i fordonskommunikationen, såsom attacker, är utanför detta arbetes omfattning.

1.4 Precisering av frågeställningen

Följande kriterier för överföring är utgångspunkt för arbetet:

- *Projekt 1:* Trådlös kommunikation mellan fordon i tät formation
 - Den viktigaste parametern är snabb överföring med låg fördröjning på ungefär 4ms.
 - Räckvidden som krävs är runt 30 meter.
 - Små datamängder på ungefär 100 byte överförs.
- *Projekt 2:* Trådlös kommunikation mellan fordon utspridda över större område
 - Det är mycket information som överförs såsom t.ex. position, lastvikt och annan produktionsdata.
 - Räckvidden bör vara lång, ca 1 kilometer.
 - Fördröjning är inte lika kritisk parameter här.

1.5 Rapportens disposition

Kapitel 2, Teoretisk bakgrund, inleds med en beskrivning av hur trådlös kommunikation generellt fungerar. Därefter förklaras några av de delar som behövs för att ett nätverk ska fungera. Till sist ges en introduktion till V2X-tekniker. Kapitel 3, Metod, är ett kort kapitel som beskriver upplägget på vår litteraturstudie samt beskrivning av hur ett V2X-test sätts upp. I kapitel 4, Resultat, hittar man en sammanfattning av några studier som gjorts på V2X samt resultat från ett egenproducerat enkelt test. Resultatkapitlet innehåller också en sammanställning av olika produkter för V2X-kommunikation. I det femte kapitlet, Slutsatser och diskussion, används resultaten för att kunna ge rekommendationer om lämpliga tekniker för de scenarion detta projekt utgår från. Kapitlet innehåller även en kort diskussion om hur vi bedömer kvaliteten på resultaten, reflektioner kring V2X ur ett hållbarhetsperspektiv och till sist förslag på tester utifrån de kriterier som satts upp.

2

Teoretisk bakgrund

För att få en bättre förståelse av V2X behöver man gå in på dess olika beståndsdelar. Så i detta kapitel förklaras grundläggande teori om hur den fysiska kommunikationen sker i form av elektromagnetiska vågor och de olika sätt att modulera dessa för V2X. Även dess frekvensspektrum och vilka frekvensband som används för olika applikationer kommer gås igenom. Då V2X-tekniken bygger på Wi-Fi-standarden tas även grundläggande teori om nätverk upp och därefter de tekniker som används för V2X, som är DSRC och C-V2X.

2.1 Trådlös kommunikation

All kommunikation sker med någon form av informationsöverföring. Elektronisk överföring kan delas upp i trådbunden och trådlös, där den trådbundna kräver någon form av ledare medan den trådlösa sker via luften. Båda varianterna kräver en sändare och en mottagare. Det är med hjälp av elektromagnetiska vågor som information kan skickas genom fria luften utan någon ledare. I antennledaren hos sändaren finns elektroner som frekvent ändrar riktning på grund av växelspanning. Denna rörelse ger upphov till förändringar i elektriska och magnetiska fält, alltså elektromagnetiska vågor. De breder ut sig i luften och når till slut mottagaren där de alstrar en likadan växelspanning som fanns hos sändaren.

2.1.1 Sändare och mottagare

Sändarens uppgift är att omvandla signalerna till en form som matchar mediet de ska transporteras genom, samt att förstärka dem så de klarar avståndet de ska färdas. Mottagarens uppgift är att ta emot samt förstärka de försvagade signalerna och att omvandla dem till en form som mottagarsidan kan tolka. Om det i en överföring går att använda fler än en antenn både hos sändaren och mottagaren, blir signalöverföringen mer pålitlig.

2.1.2 Vågornas resa genom luften

Trådlös överföring sker med hjälp av elektromagnetiska vågor. De utbreder sig i luften med ljusets hastighet. Nästan allt material som vågorna måste passera sänker utbredningshastigheten. Signalens effekt dämpas också längs vägen [1]. Effektförlusten kan beskrivas i dB med ekvationen:

$$P_L \text{ (dB)} = \lg \frac{P_t}{P_r} \quad (2.1)$$

där P_L = effektförlust (L från engelskans *path loss*), P_t = effekt vid sändare (t från engelskans *transmitter*) och P_r = effekt vid mottagare (r från engelskans *receiver*) [2]. Signalen kan också bli drabbad av störningar och brus, vilket kan göra det svårt för mottagarsidan att tolka signalen korrekt. Hur stor del brus en mottagen signal består av beskrivs med ett signal-till-brus-förhållande (skrivs vanligen SNR, förkortning av *Signal to Noise Ratio*). SNR är ett mått på hur bra kvalitet signalen har [2].

Om vägen mellan sändare och mottagare består av klar sikt, *Line Of Sight* (LOS), är detta en optimal situation för överföring. Oftast är så inte fallet, utan olika sorters hinder (terräng, natur, utrustning, byggnader...) är i vägen för utbredningsriktningen. Detta kallas *No Line Of Sight* (NLOS). NLOS orsakar flervägsutbredning [3], att signalen delas upp i två eller fler signalvägar med olika riktning och olika mycket dämpning. Dessa olika vågor kommer nå mottagaren vid olika tidpunkter, då de har olika långa avstånd att färdas. Detta gör att de kan vara fasskiftade gentemot varandra då de ankommer till mottagaren. När mottagaren har tagit emot signalerna som gått olika vägar, är det inte säkert att den totala mottagna signalen är likadan som den skickade signalen. Konstruktiv eller destruktiv interferens kan ha uppkommit på grund av fasskillnaden [4] [2]. De fysiska fenomen som påverkar överföringen är till exempel reflektion, diffraktion och spridning [3]. Om vågen slår emot en slät yta som är mycket större än signalens våglängd, uppstår reflektion. Vågen studsar ut från ytan och får en ny färdriktning. Diffraktion kallas det när vågen rör sig mot ett ogenomträngligt material med ett mindre hål och den del av vågen som färdas genom det mindre hålet böjs. Det kan också uppstå om vågen når en skarp kant. Om vågen stöter på ett hinder som är mindre än dess våglängd, uppstår spridning.

Ifall sändare och mottagare rör sig i förhållande till varandra - något som ofta händer inom fordonskommunikation - uppstår ett fenomen som kallas Dopplereffekt. Det innebär att frekvensen på vågen som mottas skiljer sig från frekvensen på vågen som skickades. Om sändaren närmar sig mottagaren blir frekvensen högre än den skulle varit om båda var stillastående och om sändaren avlägsnar sig från mottagaren blir frekvensen lägre. Grovt förenklat är anledningen att våglängderna komprimeras respektive sträcks ut [2]. En ekvation för Dopplereffekt kan skrivas:

$$f_D = f_0 \frac{v \cos \theta}{c} \quad (2.2)$$

där f_D = den frekvensskillnad som uppstår mellan sändare och mottagare, f_0 = sändarens frekvens, v är mottagarens hastighet i sin rörelse relativt sändaren, θ = vinkeln mellan signalriktningen vid mottagaren och mottagarens rörelseriktning, och c = ljusets hastighet (en konstant, ca $3 \cdot 10^8$ m/s).

Vid trådlös kommunikation är det vanligt att använda en så kallad bärvåg, med högre frekvens än den meddelandevåg (signal) som skall skickas. Meddelandevågen appliceras på bärvågen i sändaren med hjälp av en komponent som kallas blandare. Mottagaren tar sedan emot bärvågen och extraherar meddelandevågen från den [1].

2.1.3 Analoga moduleringsmetoder

Elektromagnetiska vågor kan anta alla möjliga former, men beskrivs oftast av sinusvågor, då dessa är lätta att generera, samt att alla former kan beskrivas av flera stycken sammansatta sinusvågor. För att kunna skicka någon form av information med hjälp av dessa måste de moduleras, alltså tvingas att frångå grundformen. Detta kan ske på olika sätt. Alla sinusvågor har egenskaperna amplitud, frekvens och fasvinkel. Utifrån dessa skapas amplitudmodulering (AM), frekvensmodulering (FM) och fasmodulering (PM, p:et efter engelskans *phase*). [1]. FM och PM brukar sammanfattas som vinkelmodulering. Modulering i allmänhet innebär att den aktuella egenskapen höjs eller sänks vid olika ställen på vågen för att representera en signal, t. ex. logiska ettor och nollor. Ett exempel (med godtyckliga värden) skulle kunna vara frekvensmodulering och att frekvensen 5 Hz representerar en logisk nolla och frekvensen 10 Hz representerar en logisk etta. Mottagarsidan tar då samplingar av frekvensen med jämna mellanrum, vilket skapar ett mönster av nollor och ettor.

Frekvensmodulering innebär att man på bärvågen varierar frekvensen beroende på spänningsvärdet hos meddelandevågen. Exempelvis, om den har negativ spänning blir det lägre frekvens och om den har positiv spänning blir det högre frekvens. Frekvensen varierar alltså över tiden i takt med meddelandet. Momentanspänningen på en FM-signal med entonigt meddelande kan uttryckas som:

$$v_{FM}(t) = V_c \cos(\omega_c + m_f \sin \omega_m t) \quad (2.3)$$

där V_c = bärvågens amplitud, ω_c = bärvågens vinkelfrekvens, m_f = frekvensmoduleringsindex och ω_m = meddelandets vinkelfrekvens [5]. Vid fasmodulering är det i stället fasan som varierar beroende på spänningsvärdet hos meddelandevågen. Momentanspänningen på en PM-signal med entonigt meddelande blir då:

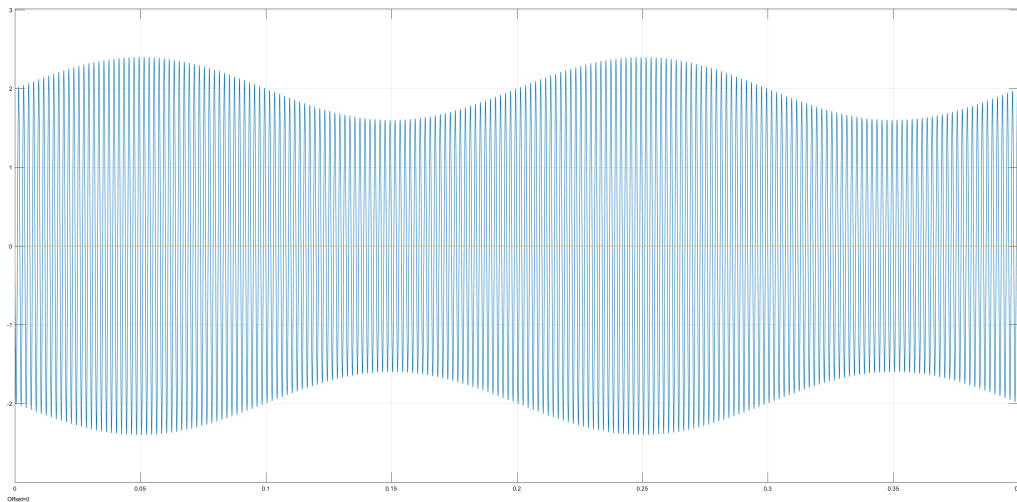
$$v_{PM}(t) = V_c \cos(\omega_c + m_\theta \cos \omega_m t) \quad (2.4)$$

där m_θ = fasmoduleringsindex.

Med hjälp av multiplikation kan en meddelandevåg amplitudmodulera en bärvåg. Toppvärdena på den modulerade vågen följer en kurva som kallas envelopp. Momentanspänningen för en AM-signal med ett entonigt meddelande uttrycks som följer:

$$v_{AM}(t) = V_c \cos \omega_c t + V_m \cos \omega_m t \cdot \cos \omega_c t \quad (2.5)$$

Den amplitudmodulerade signalen innehåller två sidband. Exempel på andra varianter av AM-signaler är *Vestigial Sideband* (VSB), *Double Sideband Suppressed Carrier* (DSB-SC) och *Single Sideband* (SSB) [6]. I Fig. 2.1 och Fig. 2.2 kan man se en DSB-modulerad våg och hur olika meddelandesignaler påverkar hur signalen ser ut. Från första figuren kan man urskilja meddelandesignalen i topparna och dalarna hos enveloppen, där topparna har en amplitud på 0,4 V och oscillerar runt medelnivån 2 V.

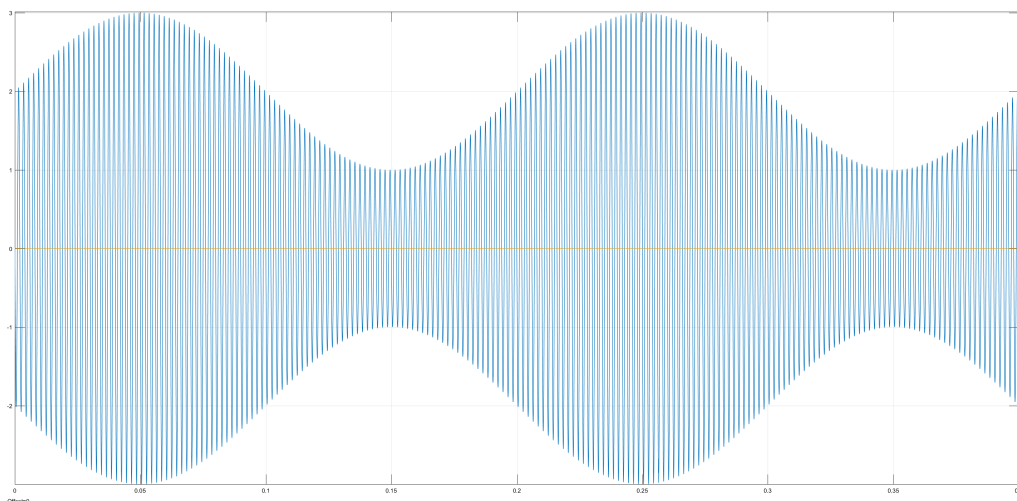


Figur 2.1: Figur av en AM-modulerad våg där $V_c=2$ V och $V_m=2$ V.

Om man ändrar amplituden för meddelandesignalen ändrar man också modulationsgraden, som i Fig. 2.1 var bärvågens amplitud delat med meddelandevågens amplitud.

$$\frac{V_{max} - V_{min}}{V_{max} + V_{min}} = \frac{2,4 - 1,6}{2,4 + 1,6} = 0,2 = 20\% \quad (2.6)$$

Denna grad beskriver hur mycket amplituden på vågen moduleras och har man en modulationsgrad över 100% är den övermodulerad. En konsekvens av detta är att signalen blir distorderad och slutar följa meddelandesignalen. Vid modulationsgrad på 100% släcks dalarna till noll och vid högre "korsar" meddelandesignalen tidsaxeln [6].



Figur 2.2: Figur av en amplitudmodulerad våg där $V_c=2$ V och $V_m=0,5$ V.

För Fig. 2.2 kan man se att toppar och dalar fluktuerar mer och modulationsgraden är (med samma ekvation som Ekv. 2.6):

$$\frac{3 - 1}{3 + 1} = 0,5 = 50\% \quad (2.7)$$

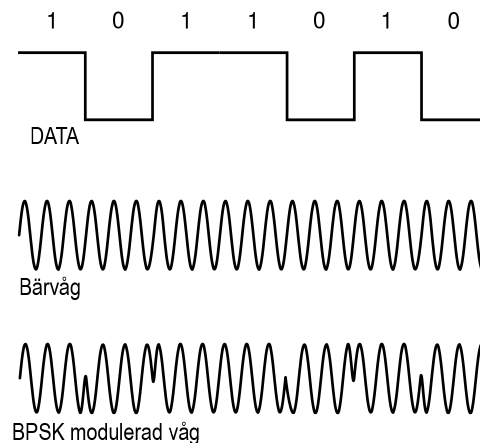
2.1.4 Digitala moduleringsmetoder

En av de större nackdelarna med endast analog modulering, är begränsningar i hur mycket och hur fort man kan skicka information jämfört med digitala. Andra fördelar med digital modulering är att den är mindre känslig för brus, samt dess användning tillsammans med programkod. Överföringshastigheten för digitala signaler kan man definiera med spektral effektivitet, som är ett mått på hur snabbt data kan överföras för en viss bandbredd. Enheten för detta mått mäts i hur många bitar per sekund (bps) som kan överföras för ett visst frekvensband bps/Hz [7]. Till exempel, med ett frekvensband på 1000 Hz och 1000 bitar som ska överföras får man 1 bps/Hz. För fordonskommunikation behöver man snabb överföring, som är en av anledningarna till att man använder sig av digitala moduleringsmetoder. Då dagens fordon använder sig av i princip bara digital kommunikation via *Controller Area Network*-bussar (CAN) (se Avs. 2.2.3), vill man helst undvika att använda sig av analog modulering då denna kräver ännu en digital till analog omvandling som kan ge felkällor. Enligt *European Telecommunications Standards Institute* (ETSI) skall signalen vara av typen *Orthogonal Frequency Division Multiplexing* (OFDM) och minst kunna hantera hastigheter på 3, 6 och 12 Mbps [8]. De moduleringstekniker som används för att uppnå dessa hastigheter är *Binary Phase Shift Keying* (BPSK), *Quadrature Phase Shift Keying* (QPSK) samt *Quadrature Amplitude Modulation* (QAM).

2.1.4.1 PSK

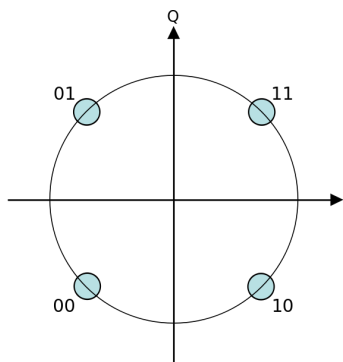
Ett sätt att modulera informationsöverföring är *Phase Shift Keying* (PSK). Det är en moduleringsteknik som bygger på att man i bärvågen gör fasskift vid olika tidpunkter.

BPSK skiftar fas beroende på om data är en logisk etta eller nolla och det är därför den definieras som binärt fasskiftande. Bärvågen för BPSK har en konstant amplitud som fås genom att addera två sinusliknande signaler som är fasförskjutna 90° från varandra. Sedan fasskiftar man bärvågen 180° , beroende på vad för data som skall skickas. Utseendet för den modulerade signalen kan ses i Fig. 2.3. Överst är ett exempel på data som skall föras över, den mittersta signalen är bärvågen med sin konstanta amplitud och längst ner är den modulerade signalen som vid varje förändring av data fasskiftas med 180° [7]. Notera hur den modulerade vågen är i fas då datan är en logisk etta och 180° förskjuten då data är en nolla.

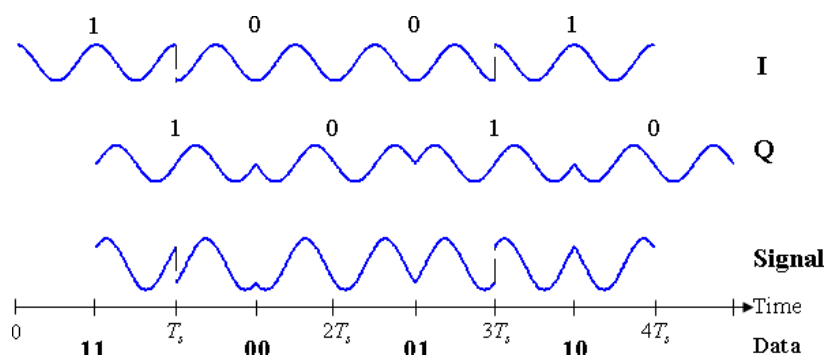


Figur 2.3: Figur av hur en BPSK-modulering kan se ut beroende på data.

QPSK är mycket lik BPSK med skillnaden att den istället för ett fasskifte mellan två lägen skiftar mellan fyra [7]. Därför kallas denna metod även för *quadri*(fyra)-*phase shift keying*. Fasskillnaden mellan bitparen är 90° och förskjuten 45° , vilket man kan urskilja ur Fig. 2.4. Man använder sig av symboler med två bitar i par istället för en som i BPSK, vilket gör att man kan skicka mer information per symbol. Den modulerade signalen som kan ses längst ner i Fig. 2.5 är kombinerad av två vågor som betecknas I och Q. Ur konstellationsdiagrammet i Fig. 2.4 kan man urskilja att x-axeln betecknas med I. Detta innebär representationen av en signal i fas med bärvågsoscillationen (In-phase). Y-axeln betecknas med Q som är en signal i kvadratur (Quadrature) d.v.s 90° förskjuten till I. Med denna representation av bärvågsoscillationen kan man beskriva hur moduleringen av en signal ser ut för de olika moduleringsteknikerna BPSK, QPSK och QAM. Om man lägger ihop signalerna I och Q, som beror på datan, får man den modulerade vågen. Om man börjar titta från tiden noll till vänster och fram till T_s , kan man se att signalen är summan av I och Q signalerna som är förskjutna 90° från varandra (se konstellationsdiagram i Fig. 2.4). Tittar man återigen på konstellationsdiagrammet ser man att vid T_s skiftar I till noll, det vill säga fasförskjuts 180° och vid strecket 00 skiftar även Q och man får en signal som är $\pm 180^\circ$ förskjuten från strecket vid 11. Vid strecket 01 skiftar signalen med -90° från streck 00 och så vidare. För att undvika felsignaler brukar man ha bitarna i en graykodad formation. Detta innebär att bitarna ändras endast en i taget mellan lägena, till exempel i Fig. 2.4 kan man se att mellan 45° och 135° ändras endast första biten.

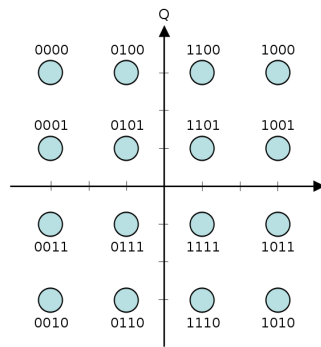


Figur 2.4: Konstellationsdiagram för en graykodad QPSK-signal. Från [9], Common Creative Attribution Share-Alike (CC-BY-SA)



Figur 2.5: Exempel på hur en QPSK-våg kan se ut av två sammansatta I- och Q-vågor.

Med moduleringsmetoden QAM ändrar man på både fas och amplitud, för att få så mycket information som möjligt i en signal [7]. Denna fas- och amplitudmodulering kan man beskriva med ett konstellationsdiagram, se Fig. 2.6. I figuren kan man se en 16-QAM, vilket innebär att man kan skicka data med symboler på fyra bitar medan för 256-QAM kan man skicka åtta bitar. Teoretiskt är det möjligt att använda högre konstellationer än 256-QAM, men redan för denna finns det stor risk för störningar och fel och det ställer höga krav på felkorrigering [5]. Från Fig. 2.6 kan man urskilja att datan 1101 motsvaras av en fas på 45° , att 1000 har samma fas men en mycket högre amplitud och det är så man skiljer på data. Denna signal brukar också graykodas för att minska fel.



Figur 2.6: Konstellationsdiagram för en graykodad 16-QAM signal. Från [10], CC-BY-SA

2.1.4.2 OFDM

För att få in mer information per bandbredd använder man sig av OFDM. Det är en modulerings teknik som används främst för mobila nätverk och Wi-Fi-nätverk. Tekniken innebär att man delar upp frekvenserna med jämnt delbara frekvensmellanrum. T. ex. skall man ha tio stycken bärvågor med bandbredden 10 MHz är bredden mellan varje våg 1 MHz. Denna teknik är mycket flexibel och kan innehålla många bärvågor i en viss bandbredd, detta för att vågorna ej skall störa ut varandra [7]. Varje bärvåg kan moduleras var för sig av typerna BPSK, QPSK och QAM för den standard som används för *Dedicated Short Range Communication* (DSRC) [8] (se avsnitt 2.3.1).

2.1.5 Elektromagnetiskt spektrum

Det elektromagnetiska spektrumet brukar delas upp i ett antal standardintervall utifrån frekvens eller våglängd. Sambandet mellan frekvens och våglängd visas i Ekv. 2.8 nedan, där f = frekvens (Hz), c = ljusets hastighet i vakuum (ca $3 \cdot 10^8$ m/s) och λ = våglängd (m) [5]:

$$f = \frac{c}{\lambda} \quad (2.8)$$

Ett flertal av standardintervallen återfinns i Tabell 2.1. Tabellen innehåller namn på de olika intervallen, förkortningar (där sådana är vedertagna), samt frekvens- och våglängdsintervall. På ännu kortare våglängder hittas synligt ljus, ultraviolett strålning (UV), röntgenstrålning samt gammastrålning [11]. Det finns även andra vanliga indelningar av spektrumet, de avhandlas ej här.

Tabell 2.1: Ett vanligt sätt att dela in det elektromagnetiska spektrumet. Tabellen visar ett urval av intervall. Från [5]. Omarbetad med tillstånd.

Beteckning	Förkortning	Frekvens	Våglängd
Extremely low frequency	ELF	3 Hz - 30 Hz	10^8 m - 10^7 m
Super low frequency	SLF	30 Hz - 300 Hz	10^7 m - 10^6 m
Ultra low frequency	ULF	300 Hz - 3 kHz	10^6 m - 10^5 m
Very low frequency	VLF	3 kHz - 30 kHz	10^5 m - 10^4 m
Low frequency	LF	30 kHz - 300 kHz	10 km - 1 km
Medium frequency	MF	300 kHz - 3 MHz	1 km - 100 m
High frequency	HF	3 MHz - 30 MHz	100 m - 10 m
Very high frequency	VHF	30 MHz - 300 MHz	10 m - 1 m
Ultra high frequency	UHF	300 MHz - 3 GHz	1 m - 10 cm
Super high frequency	SHF	3 GHz - 30 GHz	10 cm - 1 cm
Extremely high frequency	EHF	30 GHz - 300 GHz	1 cm - 1 mm
Infraröd strålning	IR	300 GHz - 300 THz	1 mm - $1 \mu\text{m}$

Några exempel på vad som karakteriserar olika frekvenser [1]:

- Frekvenser lägre än IR sammanfattas som radiovågor.
- Mänskligt tal sker på ULF.
- Sändning av AM-radio sker på MF och sändning av FM-radio sker på VHF.
- Frekvenser runt UHF och SHF kallas mikrovågor. Mikrovågsugnar opererar på UHF.
- IR, synligt ljus och UV kallas tillsammans för det optiska spektrumet.

För att inte olika typer av överföringar ska störa ut varandra måste det finnas regleringar för hur det elektromagnetiska spektrumet får användas och av vem. Allt eftersom trådlös överföring blir en mer och mer använd teknologi, blir det trängre bland frekvenserna. Louis E. Frenzel skriver i sin bok *Principles of Electronic Communication Systems* [1]: “*The electromagnetic spectrum is one of our most precious natural resources.*” Olika länder har olika organisationer som bestämmer hur frekvensallokering ska fungera för respektive. Det finns också mer globala organisationer där länderna möts och man bestämmer hur det ska fungera globalt. FN-organet *International Telecommunication Union* (ITU) ansvarar för det delade globala frekvensutrymmet [12]. I Sverige är det Post- och Telestyrelsen (PTS) som på nationell nivå arbetar med att fördela frekvensbanden på det sätt som ger mest samhällsnytta [13]. De deltar också i det globaliserande arbetet å Sveriges vägnar. Många frekvensintervall har blivit licensierade så att de ska vara tillägnade ett specifikt användningsområde, t. ex. 5,9 GHz-bandet, som har med fordonskommunikation att göra. Andra delar av spektrumet är licensfria, det innebär att vem som helst får använda dem. Exempelvis 2,45 GHz är globalt licensfritt och kallas *Industrial, Scientific and Medical*-bandet (ISM). Här huserar bland annat teknikerna Bluetooth och Wi-Fi [14].

2.1.6 Bandbredd

När en meddelandevåg med frekvensen f_m modulerar en bärvåg med frekvensen f_c (c från engelskans *carrier*), tar den resulterande signalen upp ett spann av frekvenser. Dessa frekvenser, som ligger under och över f_c , kallas för sidband. Ofta definieras signalens bandbredd B som det spann av frekvenser där dämpningen jämfört med f_c är mindre än 3 dB. De frekvenser som avgränsar bandbredden kan betecknas som f_1 och f_2 . Ekvationerna för att få fram dessa är:

$$f_1 = f_c - f_m \quad (2.9)$$

$$f_2 = f_c + f_m \quad (2.10)$$

Bandbredden blir helt enkelt $f_2 - f_1$ [1]. (Illustrerande exempel för AM-signal med dubbla sidband: Om bärvågsfrekvensen $f_c=1000$ kHz och meddelandevågen $f_m=10$ kHz, blir $f_1=990$ kHz, $f_2=1010$ kHz och bandbredden $B=20$ kHz.) Olika modulerings tekniker utnyttjar sin bandbredd olika effektivt. Måttet för detta kallas spektral effektivitet och beskrivs närmare i avsnitt 2.1.4. Som tidigare nämnts, är det nuförtiden rätt trångt i den lägre delen av frekvensspektrumet. Genom att utvidga spektrumet uppåt skapas mer plats för kommunikation. Därför lägger teknikvärlden mycket kraft på att utveckla produkter som kan operera på höga frekvenser. En signal med en viss given bandbredd kommer ta upp procentuellt mindre del av spektrumet, om även höga frekvenser kan användas som bärvågor. Detta kan illustreras med Ekv. 2.11:

$$\text{Procent av spektrum} = \frac{B}{f_c} 100 \quad (2.11)$$

Kommunikation på höga frekvenser är önskvärd även av en annan anledning - det tillåter större bandbredd. Om bärvågsfrekvensen är väldigt låg måste bandbredden vara mycket smal, då det lägre sidbandet breder ut sig ner mot 0 Hz. Höga frekvenser medför däremot att stor bandbredd kan tillåtas och att många signaler ändå får plats i utrymmet. Stor bandbredd innebär att mer data kan överföras per tidsenhet. Bandbredd relaterar alltså till bithastighet - hur många bitar som kan överföras per tidsenhet i digital överföring. Nuförtiden används ofta begreppet bandbredd felaktigt när det egentligen är bithastighet som åsyftas [15]. Men bithastigheten ändras med bandbredden tillsammans med brus och detta kan beskrivas av Shannon-Hartley teoremet som följer [1]:

$$C = B \log_2(1 + S/B) \quad (2.12)$$

Detta för en godtycklig kanal och där C är kanalens kapacitet uttryckt i bitar per sekund. B är bandbredden för kanalen och S/B är kvot mellan signal och brus. Utifrån denna kan man se att man kan öka kanalens kapacitet att överföra en signal, genom att antingen öka bandbredden eller kvoten S/B genom att minska bruset.

2.1.7 Multiplexering

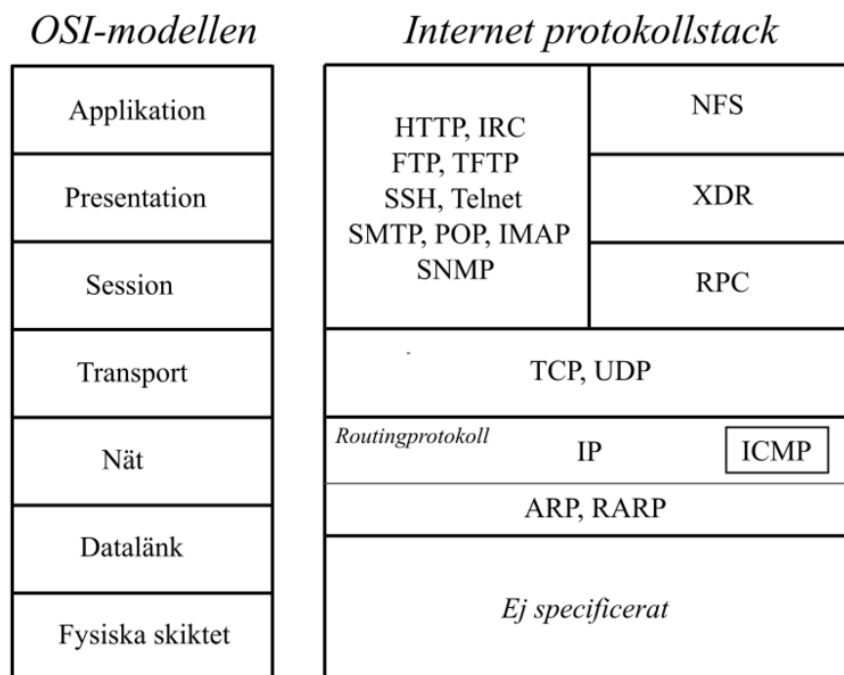
Multiplexering är ett samlingsnamn för tekniker som medför att flera olika sändare kan skicka meddelanden samtidigt, på en och samma informationsbärare. Olika metoder används för att dela upp informationsutrymmet mellan de respektive sändarna. Med *Time-division multiplexing* (TDM) turas sändarna om att skicka små bitar av relativt lågfrekventa meddelanden över en bärvåg med högre överföringshastighet. Det krävs alltså flera "omgångar" av överföring för ett helt meddelande att bli emottaget, och parallellt har meddelanden till andra mottagare också blivit skickade över samma bärvåg [16]. *Frequency-division multiplexing* (FDM) innebär att meddelanden med skilda bärvågsfrekvenser skickas över en mer bredbandig kanal. Mottagaren separerar sedan de olika sändarna med t.ex. heterodyntekniker eller med hjälp av ett bandpassfilter (ett filter som sorterar bort allt utom det önskade frekvensbandet) [17]. En annan metod för multiplexering är *Code-division multiplexing* (CDM) där varje överföringskanal tilldelas sin egen unika kod och där mottagaren sedan avkodar meddelandet.

2.2 Nätverk

För att föra över information mellan olika enheter behöver man använda sig av nätverk så att överföringen sker på rätt sätt. Det finns mycket att nämna om nätverk, men de beståndsdelar som är mest relevanta för kommunikation mellan fordon är *Open System Interconnection*-modellen (OSI) som anger standarder för hård- och mjukvara, routern som är en sorts dator, samt *Central Area Network*-bussen (CAN) som bidrar till att flera olika noder kan skicka information till varandra.

2.2.1 OSI-modell

För att lätt kunna använda nätverk för att kommunicera använder man sig av OSI-modellen. Den består av sju olika lager som kan användas individuellt, men som tjänar lagret över och betjänas av lagret under. Detta sätt att sköta telekommunikationen gör att olika tekniker kan användas inom varje lager och mellan lager är det inget som förändras vid olika teknikval. De olika lagren finns beskrivna i Fig. 2.7 tillsammans med tillhörande internetprotokoll.



Figur 2.7: Översikt av OSI-modellens olika lager och deras ordning, där fysiska lagret benämns som det första lagret och applikationslagret som det sjunde. Till höger i figuren ses tillhörande protokoll. Från [18], CC-BY-SA

Varje lager hanterar information på ett eget sätt. Dessa olika sätt kallas för *Protocol Data Unit* (PDU) och beskriver hur protokollen ser ut för de olika internetprotokollstackarna från Fig. 2.7. För att bestämma hur man skall hantera PDU:n i varje lager, kan man lägga till en header eller trailer. Headerinformation läggs till före PDU:n och trailer efter. Denna information är specifik för varje lager så att om man i sändarens nätverkslager lägger till en header, används den i mottagarens nätverkslager, vilket leder till att sändare och mottagare måste använda samma protokoll [19]. Nedan följer en översiktlig beskrivning av de olika lagrena.

2.2.1.1 Fysiska lagret

Detta lager sköter själva överföringen alternativt mottagning av data och kan ses som det första lagret i Fig. 2.7. Datahanteringen sker genom att konvertera från elektriska, radio- eller optiska signaler till digitala elektriska signaler via analog-till-digital konvertering vid mottagning och tvärtom vid sändning. Denna omvandling gör att man kan behandla signalen som en bit. Därav är PDU:n i form av en bit [19]. Den tekniska specifikationen bestäms här som binär spänningsnivå, moduleringsmetod, överföringshastighet och liknande [1].

2.2.1.2 Datalänkslagret

Datalänkslagret bidrar med nod-till-nod-överföring mellan två noder som är direkt-kopplade. PDU i detta lager hanteras som en struktur [19]. Den upptäcker och rättar även de fel som uppstår i det fysiska lagret. Lagret kan delas upp i två olika delar:

- Medium Access Layer (MAC): Ser till att rätt enhet får tillgång till rätt medium genom att identifiera enheter i nätverket med specifika adresser. Adresserna är sex byte stora och varje byte motsvarar åtta bitar. Den sköter även tillstånd att skicka data.
- Logical Link Control (LLC): Ansvarar för att kontrollera nätverkslagrets protokoll, styra felkontroll och se till att synkronisera dataramarna.

Det finns inget internetprotokoll specificerat för detta lager.

2.2.1.3 Nätverkslagret

I nätverkslagret finns procedurer och funktioner för hur kommunikationen inom nätverket skall ske och dess väg. Detta sker med hjälp av rätt adressering på noder i nätverket. PDU:n i detta fall hanteras som ett paket.

2.2.1.4 Transportlagret

För att få en tillförlitlig överföring slutanvändare till slutanvändare behövs transportlagret. I detta lager lägger man informationen i segment av datapaket, därför är PDU:n för detta lager av typen segment [19]. De vanligaste protokollen som används i transportlagret är *Transmission Control Protocol* (TCP) som är förbindelsebaserad, samt *User Datagram Protocol* (UDP) som är förbindelselös [19]. Det som definierar förbindelsebaserad är att den är uppdelad i tre faser: anslutning, dataöverföring och nedkoppling [19].

2.2.1.5 Sessionslagret

Sessionslagret planerar jobben för applikationslagret genom presentationslagret. Det hanterar de systemresurser som behövs i dessa fall och sköter även de kommandon som skall skickas till transportlagret [20]. En av dess viktigare egenskaper är att det hanterar information som läggs i presentationslagrets entitet. Detta för att kunna hantera flera uppkopplingar samtidigt [19]. I sessionslagret upprättar man en virtuell koppling som sedan stängs ner när överföringen är klar. Beroende på vilken typ av kommunikation som skall utföras - simplex, halv-duplex eller full-duplex - bestämmer lagret vem som får sända och hantera de som eventuellt sänder samtidigt. Sessionslagret har ingen egen PDU.

2.2.1.6 Presentationslagret

Då det finns många olika typer av dataarkitekturer, såsom 8-, 16-, 32- och 64-bit har alla dessa regler för att hantera data. Presentationslagrets huvuduppgift är därför att se till att datan har rätt syntax för applikationslagret. Man behöver även se till att dataformatet är rätt med avseende på byteordningen, som bestäms utifrån om det är little-endian (används av Intel och andra) eller big-endian (IBM och andra) [20]. För little-endian skickar man den minst signifikanta biten först och för big-endian skickar man den mest signifikanta biten först. Kryptering och avkryptering brukar även ske i detta lager och precis som sessionslagret så har den inte någon egen PDU.

2.2.1.7 Applikationslagret

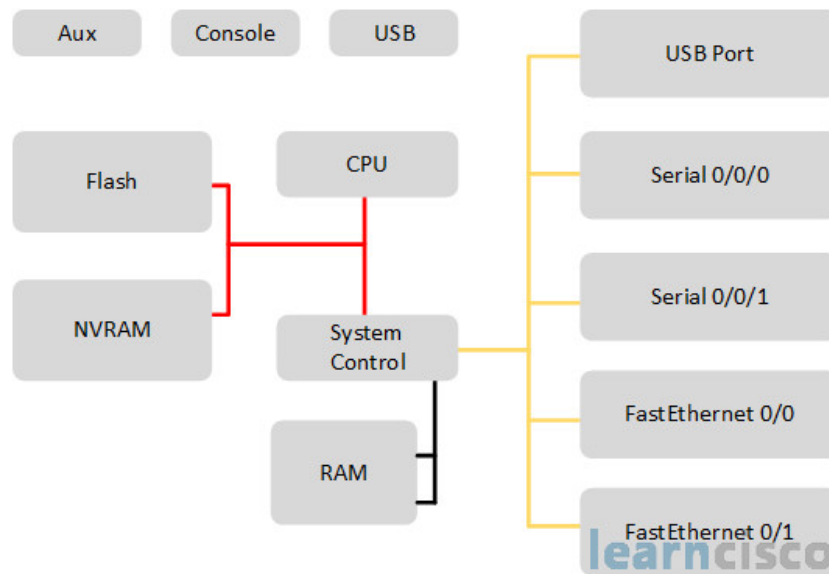
Applikationslagret är det sista lagret i OSI-modellen och är det som är närmast slutanvändaren. Beroende på vad för typ av data man skall skicka använder man olika typer av protokoll, såsom *Hypertext Transfer Protocol* (HTTP) och *Internet Message Access Protocol* (IMAP) [20]. Så om det är röstdata som skall skickas använder man ett protokoll och om det är e-mail använder man ett annat. Man kan se det som ett grafiskt användargränssnitt, t. ex. en webbsida där användaren klickar på en nedladdningslänk och ett protokoll för att göra detta används. I detta lager är QoS (Quality of Service) en stor del och det är en mätning och kontroll av kvaliteten man vill uppnå. Kriterierna skiljer sig även här beroende på vilken typ av data som skall skickas.

2.2.2 Router

En router är enkelt förklarad en avskalad persondator och består av en *Central Processing Unit* (CPU), någon typ av minne samt kringutrustning. CPU:n är själva hjärnan och utför de instruktioner som krävs av operativsystemet. Olika typer av minne behövs för behålla konfigurationsfiler, operativsystem och andra nödvändiga filer. De olika typerna av tillverkare skiljer sig väldigt lite i uppbyggnad. Om man ser till Cisco, som är den största tillverkaren [21], är minnet uppdelat i följande typer:

- *Random Access Memory* (RAM) är arbetsminnet och håller både tillfälliga värden och uträkningar. Efter att routern är spänningslös försvinner denna information.
- *Read-Only Memory* (ROM) innehåller loggar för felsökning. Denna information är kvar efter att router är spänningslös.
- Flash-minne behåller som ROM också data efter spänningslöshet och består av flytande grindtransistorer i rutnät och programmeras elektriskt att hålla data [22]. Denna innehåller operativsystemet IOS för Cisco-enheter.
- *Non-Volatile Random Access Memory* (NVRAM) håller startkonfigurationer för routern och kan till exempel vara IP-adresser för olika enheter och interna kringutrustningar. Som namnet indikerar behåller denna typ av minne datan även efter spänningsbortfall [23].

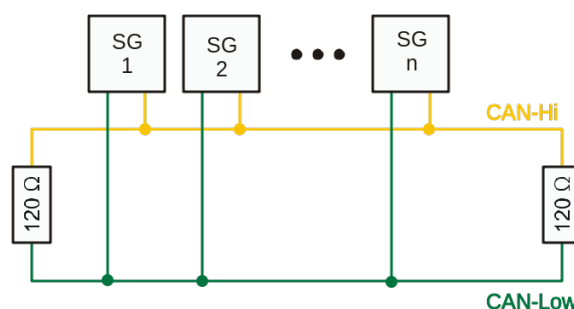
För att kunna kommunicera med andra enheter och konfigurera den egna enheten behövs en del kringutrustning. Detta är återigen beskrivet utifrån Ciscos utrustning men gäller även allmänt. Komponenterna illustreras i Fig. 2.8 tillsammans med minnen och CPU.



Figur 2.8: Översikt av de olika komponenterna i en router. Från [24], CC-BY-SA

2.2.3 CAN

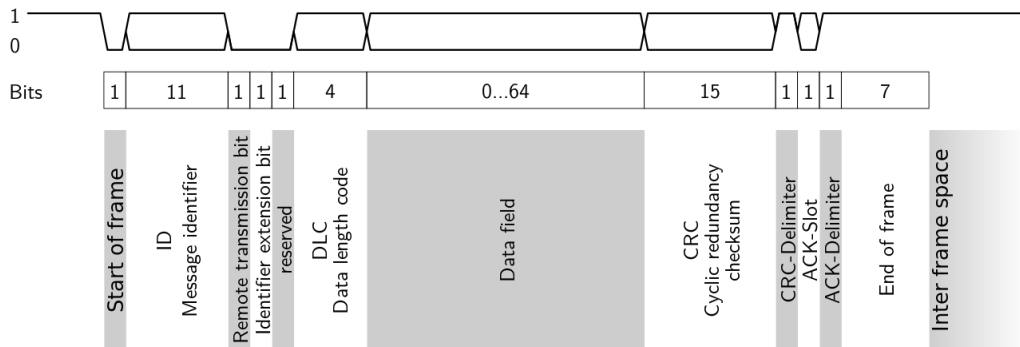
Då man i mindre system - såsom bilar - och även större, behöver skicka mycket signaler från olika sensorer och enheter behövs det ett gemensamt system för att spara på resurser som plats, energi och material. Bussystem är en lösning på detta som innebär att flera enheter har en gemensam databuss där information kan skickas och tas emot av alla parallellt uppkopplade enheter, se Fig. 2.9. Utförandet är mycket likt det som används för kommersiella nätverk och är baserat på olika typer av lager för att kunna hanteras universellt. De olika lagrena som används är det fysiska, datalänks- och applikationslagret [25] se Avs. 2.2.1. Den största skillnaden från internet är att CAN inte är adressbaserad, det vill säga att meddelandet inte skickas bara mellan nod till nod, utan till alla som lyssnar på bussen och är då meddelandebaserad istället. Varje meddelande som skickas har ett visst ID. Det är upp till varje nod att avkoda meddelandet och se om den behöver lyssna till det.



Figur 2.9: Översikt av hur CAN är uppbyggt (med ändresistorer). Från [26], CC-BY-SA

Meddelandet som skickas är uppbyggt enligt meddelanderamen i Fig. 2.10. Det börjar med en logisk nolla för att definiera start på meddelandet och efterföljs av ett

ID som definierar vad det är för typ av meddelande. För att skicka information till specifika noder kan man använda biten som i Fig. 2.10 är definierad som “remote transmission bit”. Meddelanden kan vara olika långa, därför behöver man definiera längden. Det gör man med fyra bitar innan man skickar datan, som kan variera i antalet bitar upp till 64 [25].



Figur 2.10: Meddelanderam för CAN-signaler.

2.3 V2X

V2X är ett paraplybegrepp för trådlös kommunikation där fordon är inblandade. Det gäller främst bilar av olika slag - flygplan, tåg och båtar brukar inte räknas in i begreppet [27]. Det finns ett flertal vedertagna begrepp som ingår i V2X, såsom [28]:

- V2V - kommunikation mellan fordon
- Vehicle-To-Infrastructure (V2I) - kommunikation mellan fordon och infrastruktur
- Vehicle-To-Network (V2N) - kommunikation mellan fordon och nätverk
- Vehicle-To-Pedestrian (V2P) - kommunikation mellan fordon och fotgängare

Kommunikationen kan alltså se ut på olika sätt och användas till olika syften. Helheten - V2X - kan bidra till “smartare” trafik och transport. Användningsområdena är bland annat trafiksäkerhet, att skapa smidigare trafikflöden och att effektivisera produktion på arbetsplatser där fordon används. Några exempel på situationer där V2X kan användas [29]:

- Ett fordon detekterar en kollision på vägen framför sig och skickar varningsmeddelanden till de fordon längre bak, som då kan bromsa in eller svänga undan.
- Alla fordon i ett område skickar ut positionsdata. Platser med tät trafik och risk för bilkö kan då kartläggas och bilar kan lägga om sina rutter för att undvika kö.
- Självkörande arbetsfordon kan med hjälp av V2X vara igång dygnet runt i exempelvis en gruva. Detta effektiviserar produktion.

Det finns olika sätt att skapa de nätverk inom vilka kommunikationen sker. De två populäraste, som är utformade för just fordonsanvändning, benämns DSRC respektive C-V2X. DSRC är ett dynamiskt nätverk som saknar fast infrastruktur, ett slags trådlöst lokalt nätverk (vanligare beteckning: Wireless Local Area Network (WLAN)). C-V2X använder sig av det befintliga mobilnätet, där meddelanden mellan fordon går via basstationer, alltså mobilmaster. Det är mer optimalt att kunna använda en kombination av de båda teknikerna, framför att använda bara en av dem. Det går då att nyttja respektive tekniks fördelar och få en mer flexibel kommunikation.

2.3.1 DSRC

Ett *Vehicular Ad-hoc Network* (VANET) är ett dynamiskt trådlöst nätverk som inte kräver någon fast infrastruktur. Alla de fordon som befinner sig inom räckvidd för överföring och har stöd för kommunikationen, utgör de noder som nätverket byggs upp av. Det behövs inga stationära routrar, då varje nod är en router i sig. När ett nytt fordon kommer inom räckvidd ansluts detta. När ett fordon som tidigare varit en del av nätverket kör iväg, försvinner dess anslutning. Detta skapar alltså ett rörligt nät som kan förändras snabbt, när fordon passerar varandra [27]. I ett VANET befinner sig ofta sändare och mottagare på ungefär samma höjd, någon eller några meter från marken. Detta innebär att det kan vara saker i vägen för vågutbredningen, som fordon, hus eller kullar. Detta försvårar kommunikationen och behövs tas hänsyn till vid design av V2X-applikationer [3].

DSRC använder sig av VANET-strukturen. DSRC är speciellt framtaget för kommunikation mellan fordon och det är flera standardiseringsorgan som tagit fram standarder som stödjer DSRC. Kommunikationen är designad för att ha väldigt låg latens med en kort till medellång räckvidd, det ska också vara en stabil anslutning med hög säkerhet [29]. Tiden det tar att skicka ett meddelande från en enhet till en annan är runt 4 ms och ett inbördesavstånd på ungefär 300 m se Bilaga A. Dataöverföringshastigheten hos DSRC ligger på ca 3-27 Mbps [30].

En standard som stödjer DSRC är *Institute of Electrical and Electronics Engineers* (IEEE) 802.11p [31]. 802.11p är ett tillägg till 802.11-familjen av standarder, mer känd som Wi-Fi. Standarden hanterar de två lägsta lagren i OSI-modellen, (fysiska lagret och MAC). Värt att notera är att 802.11p inte fastställer vilket frekvensband kommunikationen ska äga rum på, enbart hur kommunikationen ska utföras. Frekvenser allokeras lokalt i olika regioner. Här i Europa är frekvensbandet 5855 MHz - 5925 MHz (ett spann på 70 MHz) avsatt för DSRC. Detta brukar benämnas som 5,9 GHz-bandet. En fördel med att använda detta frekvensband är att det är avsatt för fordonskommunikation enbart, det är inte lika trångt som vissa andra frekvensband. Många V2X-enheter inom en mindre area kan orsaka störningar som äventyrar trafiksäkerheten [32]. En nackdel med denna relativt höga frekvens (jämfört med signaler på lägre frekvens) är att vågorna, som har en våglängd på runt fem centimeter [3], inte är lika bra på att penetrera hinder. 802.11p är uppdelat i sju kanaler som var och en har en bandbredd på 10 MHz. En är kontrollkanal

(*control channel*, (CCH)) och de resterande sex är servicekanaler (*service channel*, (SCH)). Det finns en maxgräns för hur mycket uteffekt en sändare får ha i 802.11p. För SCH är denna gräns 23 dBm och för CCH är gränsen 33 dBm (dBm innebär förstärkningen i jämförelse med 1 mW, uttryckt i dB). DSRC använder OFDM som moduleringsteknik. För att komplettera 802.11p i DSRC används också IEEE 1609-familjen som definierar kommunikationen för de övre lagren i OSI-modellen.

2.3.2 C-V2X

C-V2X, är beteckningen för när ett cellulärt nätverk används för kommunikation mellan fordon. Det cellulära nätverket i fråga är det befintliga utbyggda mobilnätet - det som mobiltelefonin använder sig av - där mobilmasterna fungerar som basstationer [27]. Ett cellulärt nätverk fungerar som så att ett landområde delas upp upp i mindre celler. Varje cell har en eller flera basstationer som har till uppgift att ta emot och sända vidare information till mobila enheter. En basstation består enkelt förklarad av sändare, mottagare och antenn. Tidigare sände antennerna i alla riktningar, men nu kan antennernas strålning göras mer riktad. Om man tänker sig området kring en basstation som en 360° cirkel, så är en antenn konstruerad för att täcka 120°. Har man tre av dessa jämnt utspridda, så fås tre olika sändningssystem som tillsammans täcker hela cirkeln utan att gå in i varandra. Detta ger ett mer frekvenseffektivt nät, då samtliga tre "sidor" kan använda samma frekvens utan att störningar uppstår [1]. De som sköter utbyggnaden av det cellulära nätverkets infrastruktur är mobiloperatörerna. Vissa mobiloperatörer i ett land bygger egna nätverk, medan andra mobiloperatörer köper in sig på existerande nätverk. Därför kan en och samma basstation användas av flera mobiloperatörer. Tekniken för cellulära nätverk har utvecklats i olika steg, så kallade generationer, där varje generation består av en eller flera standarder. Utvecklandet av nya generationer innebär inte att de abrupt ersätter de gamla som därmed skulle försvinna. I stället är det så att när nya generationer lanseras stannar de gamla kvar som komplement, även om de tidigare generationerna används mindre och mindre och till slut fasas ut. Anledningen till att nya generationer lanseras är att nyutvecklad teknik tillåter högre datahastigheter, större bandbredd och andra funktioner som förbättrar eller effektiviserar den mobila kommunikationen. I Tabell 2.2 nedan följer en enkel översikt av utvecklandet av generationer.

Tabell 2.2: Översikt av olika generationer för cellulära nätverk.

Generation	Bithastighet	Kommentar
1G	-	Enbart analog överföring, dvs ingen dataöverföring [33].
2G	9,6 kbps [34]	Första generationen med digital överföring. Flera standarder lanserades, <i>Global System for Mobile Communications</i> (GSM) används mest [1].
3G	12-16 Mbps [35][34]	Består av flera standarder. Internet i mobilen börjar användas i högre utsträckning i och med 3G.
4G	10-300 Mbps (LTE), 300-750 Mbps (LTE-A) [36]	Enbart en standard, <i>Long-Term Evolution</i> (LTE). Utvecklades senare till <i>LTE-Advanced</i> (LTE-A). Tillåter strömning av videofilmer i mobilen med tillfredsställande kvalitet.
5G	Några Gbps [37]	Under utbyggnad. Planerad global lansering år 2020. Medför mer smalt riktade signaler, vilket minskar störningar.

De olika generationerna av mobilnät opererar på diverse frekvensband som tillsammans sträcker sig från några hundra MHz upp till ett par GHz [38]. 5G i Sverige kommer troligen också uppta området kring 26 GHz [39]. Det är där den extremt snabba dataöverföringen kommer kunna äga rum.

C-V2X är, till skillnad från ett VANET, ett centraliserat nätverk. Information skickas från en enhet till en basstation (kallas *uplink*), basstationen skickar sedan vidare informationen till en annan enhet (*downlink*). LTE stödjer även *sidelink*, som innebär att informationen skickas direkt mellan två enheter utan att basstationen behöver användas. Detta påminner mycket om DSRC.

3

Metod

Arbetet inleddes med en grundlig litteraturstudie. Då projektet var av den undersökande typen, krävdes att vi först gjorde en kartläggning av V2X-marknaden och vilka tekniker som används. Det kom fram att det finns överföringstekniker specifikt designade för fordonskommunikation. Den information vi hittade var mestadels av akademisk karaktär, forskningsrapporter samt information om olika överföringsprotokoll och dylikt. För att få mer konkret kunskap om vilka produkter som finns på marknaden och vilka tekniker som är populära, fick vi ändra vår sökmetodik. Vi började söka efter produkter med hjälp av söktermer som *V2X hardware* eller *V2X unit*. Detta ledde oss till olika tillverkares hemsidor där vi kunde dra slutsatser om vilka tekniker de använde sig av - genomgående DSRC samt C-V2X. En sammanställning gjordes av kompletta produkter som var möjliga att genomföra tester på. Vi hittade också separata chip för V2X-överföring som hade gått att bygga en egen enhet utifrån. Dessa valdes bort då det var utanför projektets omfattning. Ett av målen med projektet var att utvärdera tekniker utifrån vissa kriterier. För att kunna göra detta sökte vi efter tester som andra gjort på V2X-överföring. Utgångspunkten i det arbetet var specifikt teknikerna DSRC och C-V2X och söktermerna utformades därefter. I slutskedet av projektet fick vi tillgång till utrustning från tillverkaren Rendits (se Tabell 4.3) och valde då att genomföra ett enkelt *Round-Trip Time* (RTT)-test för att ha några egenproducerade värden att utgå från. Ett annat mål med projektet var att ge förslag till framtida tester. Dessa förslag utformades utifrån våra kriterier, andras tester, samt vår litteraturstudie.

3.1 Test av RTT med V2X-enheter

Utifrån sitt val av utrustning kan man behöva att testa sin enhet i simulerad miljö. För att åstadkomma detta kan man sätta upp en testmiljö för en PC. Det man behöver för testning är två V2X-enheter, en switch, en dator och antenner för enheterna. Anledningen till att man använder sig av en PC för både sändning och mottagning, är för att undvika att ha två osynkroniserade klockor och för att underlätta programmering. I följande uppsättning av test används en manual för Rendits [40], men metodiken gäller även för andra typer av tester.

- *Utvecklingsmiljö*: Först behöver man sätta upp sin utvecklingsmiljö för att kunna göra tester från PC. För detta test behöver man först installera Java SDK 8 för att kunna köra testprogrammet. Man behöver även installera Maven för att kunna kompilera och bygga upp existerande programmeringsprojekt.

- *Lagerhantering:* För att kunna hantera det fysiska lagret som för över information mellan PC och V2X-enhet används i detta test ett färdigt bibliotek från Alex Voronovs ETSI ITS G5 Geonetworking stack [41], som bygger på protokoll 802.11p, se Avs. 2.3.1. För att installera laddar man ner filerna och kör kommandot “mvn clean install” i en terminalkonsoll och Maven installerar och laddar ner de filer som behövs för att installera.
- *Testkod:* Att testa hur lång tid det tar att skicka och sedan ta emot meddelande betecknas som RTT. För att beräkna detta behöver man en testkod som mäter tid från det att man skickar tills det att man tar emot. Detta görs i flera iterationer så man får ett bra medelvärde på RTT. Koden som används finns i bilaga B. Resultatet och illustration visas i Avs. 4.2.

4

Resultat

För att få sig en uppfattning av tekniken har ett antal rapporter med tester studerats och ett litet eget test har gjorts. Här presenteras resultat från dessa, samt en sammanställning av produkter.

4.1 Relevanta studier av V2X

Detta avsnitt sammanfattar tre olika projekt där tekniker för V2X testats med avseende på prestanda.

4.1.1 Utveckling av DSRC-system

År 2012 genomfördes ett projekt i Kanada med syfte att utveckla ett DSRC kommunikationssystem [42]. I och med detta genomfördes ett antal tester på systemet. Hårdvaran utgjordes centralt av en DSRC-modul från tillverkaren Unex och mjukvaran var Linux-baserad. Ett av testen innebar att sätta upp två V2X-enheter inomhus, med 15 meters mellanrum, och mäta *Packet Loss Rate* (PLR) (alltså paketförluster) samt RTT. RTT mättes med hjälp av Linux inbyggda ping-verktyg och meddelandet var 64 byte stort. Resultatet av testet var 0% i PLR samt ca 0,6 ms i RTT. I ett annat test monterades två V2X-enheter i var sin bil med antenner monterade på biltaken. I del 1 av testet körde bilarna åt samma håll med 150 meter mellan sig, i 60 km/h. Den ena enheten pingade kontinuerligt den andra under tiden. PLR samt RTT mättes. Resultatet blev 1% i PLR samt 0,658 ms i RTT. I del 2 av testet var syftet att mäta den maximala räckvidden hos DSRC-överföringen. En av bilarna var stillastående och medan den andra bilen körde iväg därifrån, pingade den hela tiden den första bilen. Vid den sträcka då pingandet inte längre kom fram till den stillastående bilen visste man att räckvidden var överskriden. Resultatet blev en räckvidd på 230 meter. Det kan noteras att vid detta tillfälle hade bilen redan kommit utanför LOS.

4.1.2 Jämförelse av DSRC och C-V2X med avseende på räckvidd

Biltillverkaren Ford och chiptillverkaren Qualcomm har utfört tester i USA för att jämföra hur DSRC och C-V2X presterar [43]. Detta gjordes på en 1350 meter lång testbana. Här sammanfattas tre av dessa tester. Uteffekten var satt till 17 dBm för

både DSRC- och C-V2X-enheterna.

- *Test 1:* Här mättes räckvidden för DSRC- respektive C-V2X-tekniken under LOS-förhållanden. En bil var stillastående medan en annan bil körde bort från den. Samtidigt skickades meddelanden mellan bilarna.
- *Test 2:* Detta test var uppbyggt för att simulera en korsning med skymd sikt (NLOS). En bil stod parkerad med en lastbil på var sida om sig. En annan bil körde så att den närmade sig dem, passerade deras fronter och sedan körde därifrån. Meddelanden skickades mellan de två bilarna.
- *Test 3:* En bil stod parkerad med en stillastående lastbil precis framför sig. En annan bil som befann sig framför lastbilen körde framåt så att lastbilen hela tiden skymde den bakre bilen totalt (större grad av NLOS än i test 2). Även här skickades meddelanden mellan de två bilarna.

Resultaten från dessa tester återfinns i Tabell 4.1 nedan. Gemensamt för alla tester är att när väl signalen börjat försämrats går andel mottagna meddelanden relativt snabbt ner mot noll. I tabellen anges dock endast sträckan där all information mottas (100%), samt det avstånd där andelen mottagen information har sjunkit till 90%.

Tabell 4.1: Resultat från tester av överföringsräckvidd för DSRC och C-V2X. I tre olika scenarier har dataförluster uppmätts vid olika avstånd.

Andel mottagna meddelanden	Test 1 (avstånd)	Test 2 (avstånd)	Test 3 (avstånd)
100% (DSRC)	575 m	350 m	100 m
90% (DSRC)	675 m	375 m	150 m
100% (C-V2X)	1075 m	650 m	400 m
90% (C-V2X)	1175 m	850 m	550 m

4.1.3 Jämförelse av DSRC och C-V2X under olika körhastigheter

Det finns fler exempel på jämförelser mellan DSRC och C-V2X. I Kina gjordes detta år 2017 på en testbana för kommunikation mellan fordon [44]. DSRC-systemet bestod av både *On-Board Units* (OBU) samt stationära *Road-Side Units* (RSU). C-V2X-systemet var ett nätverk med 4G på 1,89 GHz. Nätverket var indelat i fyra celler, där varje cell hade en nedladdningshastighet på 80 Mbps. Testerna utgick från tre scenarier, varav två stycken kommer återges här, *Collision Avoidance* samt *Traffic Text Message Broadcast*. I båda två uppmättes PLR samt RTT.

- *Collision Avoidance:* Ett statusmeddelande på 100 bytes skickas från en bil till en annan med 10 meddelanden per sekund. Meddelandena kan innehålla information om närliggande bilars körning, så att kollisioner kan undvikas. Då detta är en säkerhetsapplikation skall RTT inte överstiga 100 ms. Syftet med testet är att jämföra RTT och PLR för DSRC och C-V2X vid olika körhastigheter för fordon: 30, 60, 90 samt 120 km/h. Ena bilen kör alltså

bakom den andra med ett säkerhetsavstånd mellan dem, båda kör i samma hastighet.

- *Traffic Text Message Broadcast*: Traffic Text Message Broadcast innebär korta textmeddelanden som skickas ut till fordonens OBU:s från en central informationstjänst. Meddelandena kan t. ex. innehålla uppgifter om trafikflöde samt väder- och vägförhållanden. Storleken på textmeddelandena är max 350 bytes och de skickas med 10 meddelanden per sekund. Informationen når bilarna via RSU:s i DSRC-fallet, respektive basstationer i C-V2X-fallet. Testerna mäter räckvidden för DSRC-överföring mellan en RSU samt ett fordon som kör i 120 km/h. Man gör också ett likadant test som i Collision Avoidance, med skillnaden att det är ett annat sorts meddelande som skickas.

Nedan följer en kort sammanfattning av resultaten från testerna. Figurerna som hänvisas till återfinns i Bilaga A. I figurerna används benämningen LTE, vilket här kallas C-V2X.

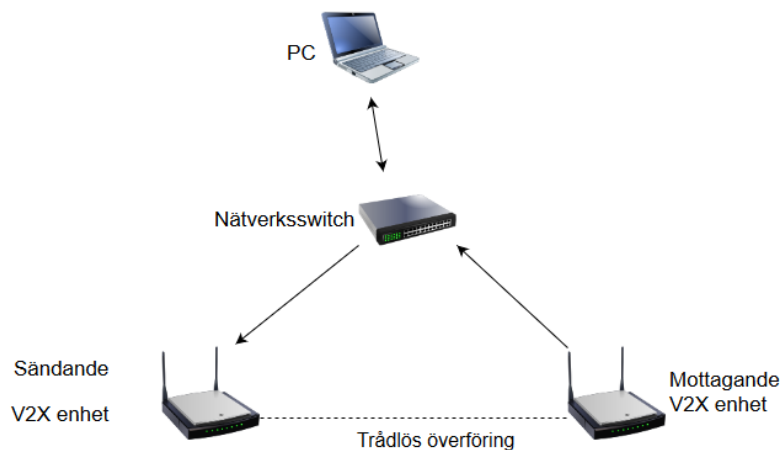
- *Collision Avoidance*: Fig. A.1 visar hur DSRC bara påverkas marginellt av hastighetsökningen, medan C-V2X har en tydligare ökning på flera procent. När det kommer till RTT (Fig. A.2) har DSRC en extremt låg fördröjning som inte påverkas alls av hastigheten, medan C-V2X ligger på ett par hundra millisekunder och får sämre prestanda av hastighetsökningen.
- *Traffic Text Message Broadcast*: Fig. A.3 samt Fig. A.4 visar att både PLR och RTT ökar drastiskt när avståndet mellan OBU och RSU överstiger ca 300 m, då den tappar kontakten. I detta sammanhang har alltså DSRC en räckvidd på 300 m. Resultatet från testet med olika körhastigheter uppvisar samma mönster som resultatet från Collision Avoidance: DSRC-överföringen påverkas inte av hastigheten, medan C-V2X-överföringen får sämre prestanda allt eftersom hastigheten ökar. Dock har man inte tagit hänsyn till fördröjningar i den styrande elektroniken, vilket kan ha stor påverkan på fördröjningen. Men C-V2X (LTE) har genomgående lägre PLR och RTT i detta test (se Fig. A.5 och A.6), jämfört med Collision Avoidance (Fig. A.1 och A.2).

Enligt [44] är det till stor del Dopplereffekten som är anledning till C-V2X försämrade prestanda. DSRC drabbas inte lika mycket av denna effekt som man kan se i Fig. A.6. Detta gäller både när OBU-enheten rör sig mot RSU-enheten och från, som man kan urskilja i Fig. A.4 där negativt värde på x-axel syftar på att den rör sig mot RSU och positivt syftar på att den rör sig ifrån.

4.2 Eget test av V2X-enheter

I frågeställningen rörande *Projekt 1* (se Avs. 1.4) ställdes kraven till att överföringen skall vara snabb med liten fördröjning och med kortare avstånd och små datapaket. För att få en uppfattning av hur snabb överföring med DSRC kan vara, kopplades två V2X-enheter från utvecklaren Rendits till en PC via en switch. Det togs inte hänsyn till typen av nätverksswitch eller nätverkskort då detta test endast var till för att få en uppfattning. Kopplingen kan illustreras i Fig. 4.1 där PC:n kör ett testprogram

som skickar 100 meddelanden via switchen till sändarenheten som mottar och skickar meddelandena ut i luften utan att ta hänsyn till om någon nod lyssnar. Mottagaren som lyssnar efter meddelanden registrerar de 100 meddelandena och vidarebefordrar dessa till PC:n som räknar ut hur lång tid det tog att få över meddelandet mellan sändare och mottagare.



Figur 4.1: Översikt över hur test av RTT genomförts.

Testprogrammet kör tio stycken överföringar av 100 meddelanden på 100 byte och delar på tusen för att få fram RTT, det vill säga hur lång tid det tar från PC genom V2X-enheter och tillbaka. I Tabell 4.2 ses resultat från ett antal testfall som är relevant för *Projekt 1*. Värdena i tabellen kommer från ett medelvärde av att köra testprogrammet tre gånger för varje scenario och detta motiveras med att vi får tillräckligt lika resultat varje gång.

Tabell 4.2: Test av RTT för tre olika scenarier med avseende på avstånd mellan V2X enheter och LOS.

Parameter	RTT
LOS 5 meter	3,887 ms
LOS 20 meter	3,885 ms
NLOS 20 meter	3,930 ms

Utifrån de resultat som uppnåddes kan man se att de är väldigt liten tidsskillnad i dessa testfall. Inom LOS spelar det ingen roll med kortare avståndsskillnader, men NLOS har en liten men viss skillnad. För NLOS användes 5 cm tjockt trä. Dessa resultat kan jämföras med tester från Avs. 4.1.3 och därifrån från Fig. A.4 kan man utläsa ett värde på ca 5 ms. Detta är rimligt då vårt test var utformat för två stillastående enheter och i inomhusmiljö, vilket bör leda till bättre resultat. Jämför man dock med resultat från Avs. 4.1.1 där man har en RTT på 0,6 ms för 64 byte så skiljer det sig mycket, men i det fallet tog man bort fördröjningen mellan pingen

så det går inte riktigt att jämföra. Detta understryker åter igen hur viktigt det är att optimera elektronik och kodning.

4.3 Sammanställning av produkter

I Tabell 4.3 följer en sammanställning av de V2X-produkter som hittats under projektets gång, som bedöms som lämpliga för projektets syfte. Samtliga produkter är kompletta kommunikationsenheter med hårdvara och mjukvara och ska kunna användas för test av överföring. De benämns av utvecklarna bl. a. som *evaluation kits*, *development platforms* och *on-board units*. Utifrån vad vi sett på hemsidor och i produktspecifikationer är alla ungefär samma sorts produkt. Alla använder DSRC och/eller C-V2X, då dessa är de mest vedertagna teknologierna för kommunikation mellan fordon.

Tabell 4.3: En sammanställning av V2X-enheter som kan användas för tester. Produkterna listas utan någon inbördes ordning.

Utvecklare	Modell	V2X-teknologi	Utvecklarens hemvist
Intrinsyc [45]	Qualcomm CV2X Development Platform	C-V2X	Kanada
Nordsys [46]	WaveBEEplus Universal 64Bit C2X Development Platform	DSRC	Tyskland
Unex [47]	OBU-201E	DSRC	Taiwan
Savari [48]	MobiWAVE On-Board-Unit	DSRC	USA
Kapsch TrafficCom [49]	EVK-3300 V2X Evaluation Kit	DSRC	Österrike, finns kontor i Sverige
Cohda Wireless [50]	MK5 OBU	DSRC, C-V2X	Australien, finns kontor i Tyskland
Commsignia [51]	ITS-OB4	DSRC, C-V2X	USA
Rendits [40]	Vehicle-to-Anything Router	DSRC	Sverige

I Fig. 4.2 ses ett exempel på hur dessa V2X-enheter kan se ut. På bilden syns några av enhetens gränssnitt, såsom uttag för CAN-buss och Ethernet. För att upprätta en kommunikation krävs minst två av dessa enheter. Antenner kan kopplas in för att öka räckvidden på överföringen.



Figur 4.2: Exempelbild på enhet för V2X-kommunikation. Från [52]. CC-BY-SA

5

Slutsatser och diskussion

I detta kapitel, som rundar av projektet, presenteras slutsatser kring vilka överföringsteknologier som passar bäst för respektive projekt. Sedan följer en kort reflektion kring resultatets kvalitet och vad som hade kunnat förbättra det. Vi diskuterar också V2X ur ett större perspektiv, samt ger förslag till fortsatta studier.

5.1 Rekommendation av tekniker utifrån frågeställning

Frågeställningen i detta projekt (se Avs. 1.4) handlar om ifall det går att hitta tekniker som uppfyller de kriterier som ställts av CPAC för de två aktuella projekten. Utifrån de studier vi tagit del av, samt vårt egna test, rekommenderar vi följande:

- *Projekt 1:* Till projektet med fordonen som kör i tät formation bedöms DSRC vara en lämplig teknologi. RTT ligger på ett par millisekunder, vilket Avs. 4.1.1 och Avs. 4.1.2 samt Tabell 4.2 visar. Detta är tillräckligt låg fördröjning för säkerhetsaspekten när fordon kör alldeles intill varandra. Det är inga större datamängder som ska överföras, så DSRC:s datahastighet på några tiotal Mbps (se Avs. 2.3.1) klarar av detta. Angående räckvidd på överföring så krävs det bara ett par tiotal meter, så de flera hundra meter DSRC uppnår (se Avs. 4.1.1 och Avs. 4.1.3 samt Tabell 4.1) räcker mer än väl. C-V2X rekommenderas ej till detta projekt på grund av den höga latensen, några hundra millisekunder enligt studien i Avs. 4.1.3.
- *Projekt 2:* Till projektet med fordonen som är utspridda över ett stort område, tror vi att C-V2X kan vara en lämplig teknik. Det bör mer specifikt vara 4G, då det är detta som testats. I detta projekt är räckvidden det största problemet, men C-V2X uppnår avstånd på drygt en kilometer vid LOS och runt 500 meter vid NLOS i ett av de experiment vi granskat (se Tabell 4.1). Med möjlighet att överföra relativt stora mängder data, några hundra Mbps enligt Tabell 2.2, passar C-V2X bra för projektet förutsatt att man bygger ut nät om det saknas täckning. Fördröjning är ingen kritisk parameter, så några hundra millisekunder (se Avs. 4.1.3) ska inte vara några problem. Även DSRC hade troligen passat för detta projekt, om det hade räckt med en överföringsräckvidd på några hundra meter.

5.2 Diskussion kring resultatets kvalitet

Om projektet hade utförts i större skala, hade fler tester kunnat göras på V2X-enheterna från Rendits. Detta hade troligen gett ett mer värdefullt resultat, eftersom testerna hade kunnat skräddarsys utifrån de två scenarion projektet utgår från. Det faktiska resultatet är i stället en förstudie som bör kunna vara till stor hjälp för den som eventuellt genomför de praktiska testerna. Vi anser att de tekniker vi hittat matchar kriterierna för respektive scenario bra. Dock är slutsatserna dragna främst från ett fåtal andra studier. Fler prestandavärden från andra tester hade eventuellt bidragit till en mer nyanserad bild av respektive teknik. Sammanställningen av produkter i Tabell 4.3 kan vara till hjälp om man vill testa tekniker för V2X, men inte vet var man ska börja leta. Då tillverkarna sällan specificerar prestanda eller pris på sina webbsidor, ingår inte dessa parametrar i tabellen. Vi tror att många av produkterna presterar på liknande nivå, då det finns regleringar för mycket av tekniken, som maximal tillåten uteffekt.

5.3 V2X ur ett etiskt och hållbarhetsperspektiv

Många är rädda för den elektromagnetiska strålning våra kroppar utsätts för i och med den ökade användningen av trådlös teknologi. Strålsäkerhetsmyndigheten bedömer dock att det inte finns några hälsorisker med exponering av radiovågor från exempelvis trådlösa nätverk och basstationer [53]. Den största och mest komplexa frågan när det kommer till fordonskommunikationens påverkan på miljön, är troligtvis det faktum att användning av V2X kan bidra till utvecklingen av autonoma fordon. Detta ämne debatteras mycket ur både ett etiskt perspektiv samt när det kommer till hållbar utveckling. En självkörande bil som kommunicerar med andra fordon kan anpassa sin hastighet på ett smidigare sätt som minskar utsläppen, och bilköer kan troligen bli ett mindre problem. Ett annat tänkbart scenario är att självkörande bilar gör att folk som i dagsläget inte har körkort kan åka själv i en bil (då skaffar eventuellt fler bil) och att människor kan tänka sig längre pendlingstider, då de kan t. ex. äta eller jobba under tiden (bilar kör längre sträckor per dag) [54] [55]. Detta är bara några exempel på hur ett samhälle med smarta bilar kan ändra människors beteende och det är svårt att säga om detta kommer påverka miljön positivt eller negativt. Om man tittar på användning av V2X i arbetsfordon mer specifikt, kan V2X leda till smartare produktion och logistik, vilket skulle kunna minska onödiga körningar. Det kan också bidra till ökad säkerhet för människor om autonomi gör att man inte behöver vistas på riskfyllda arbetsplatser.

5.4 Förslag till fortsatta studier

Här presenteras några förslag på relevanta tester att genomföra utifrån de scenarion som beskrivits i Avs. 1.4. Dessutom förklaras varför vissa andra tester är mindre relevanta för just dessa scenarion.

- Räckvidd på överföringen testas lämpligen med flera olika “nivåer” av NLOS. Allt ifrån helt fri sikt, via mindre hinder utspridda, till att placera ett stort fordon framför någon av enheterna. Detta för att simulera de miljöer överföringen kommer ske i. Testerna kan ske genom att mäta PLR vid olika avstånd.
- Latens och datahastighet kan testas genom att dels pinga och dels skicka en större fil. Detta kan göras på olika avstånd.
- Om möjligt bör produkterna testas utomhus i olika sorters väder (klart väder, regn, snö) för att kunna avgöra om de kan prestera under årets alla dagar. Här bör PLR testas. För *Projekt 2* är det även relevant att testa för smuts och damm.
- Test av Dopplereffekt är något som kan göras om C-V2X är tekniken som används (DSRC drabbas knappt av Dopplereffekt). Dock kör stora arbetsfordon som regel i relativt låga hastigheter där effekten är mycket liten, varför vi inte anser att detta test är kritiskt.

Utöver förslag på framtida tester med utgångspunkt från denna förstudie, finns fler möjligheter till studier på ämnet. När 5G etablerats som teknik kommer det möjligen passa bra till fordonskommunikation, tack vare de höga hastigheterna. Därför föreslår vi tester på enheter som använder 5G när dessa väl finns på marknaden. Viktigt är då att mäta RTT för att se om det går att användas till säkerhetsapplikationer. Det är även intressant att titta närmare på sidelink-funktionen hos C-V2X och undersöka vilken prestanda denna har. Datasäkerhet är en viktig fråga när det kommer till fordonsindustrin och därför är det relevant att ta reda på hur känsliga olika V2X-tekniker är för intrång och attacker.

Referenser

- [1] L. E. Frenzel Jr., *Principles of Electronic Communication Systems*, 4. utg. New York, USA: McGraw-Hill Education, 2016.
- [2] A. Goldsmith, *Wireless Communications*. New York, USA: Cambridge University Press, 2005, [Online]. URL: <https://app.knovel.com/hotlink/toc/id:kpWC000002/wireless-communications/wireless-communications> (hämtad 2019-02-15).
- [3] A. Tomasi, "Reliable communication in mine environments for autonomous vehicles", [Online], masteruppsats, Stockholm, Sverige, 2016. URL: <http://www.diva-portal.org/smash/get/diva2:1078420/FULLTEXT01.pdf> (hämtad 2019-03-27).
- [4] R. K. Crane, "Radio-wave propagation", *AccessScience*, [Online]. URL: <https://www-accessscience-com.proxy.lib.chalmers.se/content/radio-wave-propagation/568500> (hämtad 2019-05-13).
- [5] M. J. N. Sibley, *Modern Telecommunications, Basic Principles and Practices*. Boca Raton, USA: CRC Press, 2018, [Online]. URL: <https://www-taylorfrancis-com.proxy.lib.chalmers.se/books/9781351263603> (hämtad 2019-05-10).
- [6] P. Wallander, *17 lektioner i TELEKOMMUNIKATION*. Älvsjö: PERANT, 2001, ISBN: 91-86296-10-8.
- [7] R. A. Poisel, *Electronic Warfare Receivers and Receiving Systems*. Boston, USA: Artech House, 2018, [Online]. URL: <https://app.knovel.com/hotlink/pdf/rcid:kpEWRRS001/id:kt011LUTL1/electronic-warfare-receivers/orthogonal-frequency?kpromoter=federation> (hämtad 2019-03-01).
- [8] ITS-G5, *ITS-G5 Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band*, REN/ITS-0040191, 2019.
- [9] Splash. (2006). Constellation diagram for Gray-coded 4PSK. [Elektronisk bild], URL: https://commons.wikimedia.org/wiki/File:QPSK_Gray_Coded.svg (hämtad 2019-05-13).
- [10] —, (2006). Constellation diagram for Gray coded 16-QAM. [Elektronisk bild], URL: https://commons.wikimedia.org/wiki/File:16QAM_Gray_Coded.svg (hämtad 2019-05-15).
- [11] Wikipedia. (2018). Elektromagnetiskt spektrum. [Online], URL: https://sv.wikipedia.org/wiki/Elektromagnetiskt_spektrum (hämtad 2019-05-10).
- [12] ITU. (2019). About International Telecommunication Union (ITU). [Online], URL: <https://www.itu.int/en/about/Pages/default.aspx> (hämtad 2019-05-09).

- [13] Post- och telestyrelsen. (2015). Spektrumnytta som ger samhällsnytta. [Online], URL: <https://www.pts.se/globalassets/startpage/dokument/icke-legala-dokument/faktablad/radio/sv-spektrum-april-2015.pdf> (hämtad 2019-05-09).
- [14] Wikipedia. (2019). ISM Band. [Online], URL: https://en.wikipedia.org/wiki/ISM_band (hämtad 2019-05-06).
- [15] R. Keim. (2018). Understanding Bandwidth in Communications and Computing. [Online], URL: <https://www.allaboutcircuits.com/technical-articles/understanding-bandwidth-in-communications-and-computing/> (hämtad 2019-05-13).
- [16] Wikipedia. (2019). Multiplexing. [Online], URL: <https://en.wikipedia.org/wiki/Multiplexing> (hämtad 2019-05-28).
- [17] —, (2018). Multiplex. [Online], URL: <https://sv.wikipedia.org/wiki/Multiplex> (hämtad 2019-05-28).
- [18] Svjo. (2016). OSI model compared with internet protocol. [Elektronisk bild], URL: <https://commons.wikimedia.org/wiki/File:Osi-vz-ip-model-sv.png> (hämtad 2019-05-09).
- [19] M. M. Alani, *Guide to OSI and TCP/IP Models*, ser. SpringerBriefs in Computer Science. Springer, 2014, [Online]. URL: <https://library-books24x7-com.proxy.lib.chalmers.se/toc.aspx?site=Y7V97&bookid=76720> (hämtad 2019-03-28).
- [20] L. M. Thompson och T. Shaw, i *Industrial Data Communications (5th Edition)*. ISA, 2016, ISBN: 978-0-87664-095-1. URL: <https://app.knovel.com/hotlink/khtml/id:kt0112ONER/industrial-data-communications/modeling>.
- [21] Synergi. (2017). Switching & Router Revenues Still Growing; Cisco Still Controls Half of the Market. [Online], URL: <https://www.srgresearch.com/articles/switching-router-revenues-still-growing-cisco-still-controls-half-market> (hämtad 2019-05-07).
- [22] Wikipedia. (2019). Flash memory. [Online], URL: https://en.wikipedia.org/wiki/Flash_memory (hämtad 2019-05-07).
- [23] learncisco. (2019). Router Internal Components Overview. [Online], URL: <https://www.learncisco.net/courses/icnd-2/network-management/router-internal-components-overview.html> (hämtad 2019-05-07).
- [24] —, (2019). Router Internal Components Overview. [Elektronisk bild], URL: <https://www.learncisco.net/courses/icnd-2/network-management/router-internal-components-overview.html> (hämtad 2019-05-08).
- [25] P. Zhang, *Part V. Embedded Networks in Industrial Control Systems, Advanced Industrial Control Technology*. Elsevier, 2010, [Online], ISBN: 978-1-4377-7807-6. URL: <https://app.knovel.com/hotlink/khtml/id:kt0098LCR1/advanced-industrial-control/part-v-embedded-networks>.
- [26] Stefan-Xp. (2008). CAN-Bus Topology (Electronic Two Wire Connection). [Elektronisk bild], URL: https://commons.wikimedia.org/wiki/File:CAN-Bus_Elektrische_Zweidrahtleitung.svg (hämtad 2019-05-15).
- [27] Z. MacHardy, A. Khan, K. Obana och S. Iwashina, "V2X Access Technologies: Regulation, Research, and Remaining Challenges", *IEEE Communications*

- Surveys & Tutorials*, årg. 20, s. 1858–1877, 3 2018, [Online]. URL: <http://proxy.lib.chalmers.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edb&AN=131486989&lang=sv&site=eds-live&scope=site> (hämtad 2019-02-22).
- [28] J. Wang, Y. Shao, Y. Ge och R. Yu, “A Survey of Vehicle to Everything (V2X) Testing”, *Sensors*, årg. 19, s. 334, 2 jan. 2019. URL: <https://www.mdpi.com/1424-8220/19/2/334> (hämtad 2019-06-12).
- [29] A. Ghirnikar, “Trådlöst som banar väg för uppkopplade fordon”, *Elektro-niktidningen*, s. 24–25, febr. 2018, [Online]. URL: <http://etn.se/index.php/technical-papers/64380-marvell-tradlost-som-banar-vag-for-uppkopplade-fordon.html> (hämtad 2019-03-28).
- [30] M. Patrick. (2018). V2X communications - LTE versus DSRC. [Online], URL: <https://www.mwee.com/design-center/v2x-communications-lte-versus-dsrc> (hämtad 2019-05-22).
- [31] T. V. Nguyen et al., “A comparison of cellular vehicle-to-everything and dedicated short range communication”, *2017 IEEE Vehicular Networking Conference (VNC)*, 2017, [Online]. URL: <https://ieeexplore.ieee.org/document/8275618> (hämtad 2019-06-12).
- [32] N. Vivek, S. V. Srikanth, P. Saurabh, T. P. Vamsi och K. Raju, “On field performance analysis of IEEE 802.11p and WAVE protocol stack for V2V & V2I communication”, *International Conference on Information Communication & Embedded Systems (ICICES2014)*, s. 1–6, 2014, [Online]. URL: <https://ieeexplore.ieee.org/document/7033960> (hämtad 2019-04-03).
- [33] P. Viswanathan. (2018). How Does a Mobile Network Work? [Online], URL: <https://www.lifewire.com/how-does-a-mobile-network-work-2373338> (hämtad 2019-04-04).
- [34] Wikipedia. (2018). Mobile broadband. [Online], URL: https://en.wikipedia.org/wiki/Mobile_broadband (hämtad 2019-05-22).
- [35] P. Davidsson. (2017). Bredbandskollen, Mobil surfhastighet i Sverige 2017. [Online], URL: https://internetstiftelsen.se/docs/Bredbandskollen_Mobil_surfhastighet_2017.pdf (hämtad 2019-05-22).
- [36] Kjell & Company. (2018). Mobilnäten. [Online], URL: <https://www.kjell.com/se/fraga-kjell/hur-funkar-det/mobilt/mobil-kommunikation/mobilnaten> (hämtad 2019-05-22).
- [37] S. Kavanagh. (u.d.). How fast is 5G? [Online], URL: <https://5g.co.uk/guides/how-fast-is-5g/> (hämtad 2019-05-22).
- [38] Post- och telestyrelsen. (u.d.). Mobiltelesystem. [Online], URL: <https://www.pts.se/sv/privat/telefoni/mobil-telefoni/mobiltelesystem/> (hämtad 2019-05-22).
- [39] P. Ottsjö, “Digitaliseringen i Sverige går långsammare”, *Ny Teknik*, s. 16–17, nr. 11 mars 2019.
- [40] *RENDITS MANUAL*, [Online], 2018. URL: <https://github.com/rendits/router/blob/master/doc/manual.pdf> (hämtad 2019-05-24).
- [41] A. Voronov, A. Zeverin och D. Heuven, *geonetworking: Vehicle Adapter*, [Repository], maj 2016. DOI: 10.5281/zenodo.51295. URL: <https://doi.org/10.5281/zenodo.51295>.

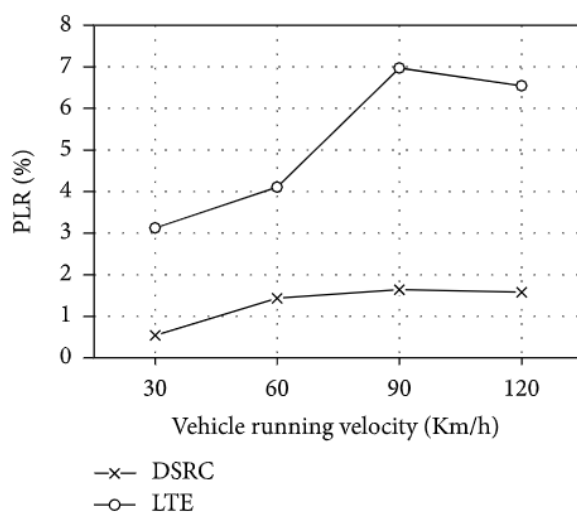
- [42] F. Kamal, E. Lou och V. Zhao, "Design and validation of a small-scale 5.9 GHz DSRC system for vehicular communication", *2012 25th IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)*, s. 1–4, 2012, [Online]. URL: <https://ieeexplore.ieee.org/abstract/document/6334893> (hämtad 2019-05-21).
- [43] 5GAA Automotive Association, *V2X Technology Benchmark Testing*, [Online], 2018. URL: <https://www.qualcomm.com/media/documents/files/5gaa-v2x-technology-benchmark-testing-dsrc-and-c-v2x.pdf> (hämtad 2019-05-21).
- [44] Z. Xu, X. Li, X. Zhao, M. H. Zhang och Z. Wang, "DSRC versus 4G-LTE for Connected Vehicle Applications: A Study on Field Experiments of Vehicular Communication Performance", *Journal of Advanced Transportation*, årg. 2017, aug. 2017, [Online]. DOI: 10.1155.2017.2750452. URL: <https://www.hindawi.com/journals/jat/2017/2750452/> (hämtad 2019-05-21).
- [45] Intrinsyc. (2019). Qualcomm Cellular Vehicle-to-Everything (CV2X) Development Platform. [Online], URL: <https://shop.intrinsyc.com/products/qualcomm-cellular-vehicle-to-everything-cv2x-development-platform> (hämtad 2019-05-30).
- [46] *waveBEEplus*, *UNIVERSAL 64BIT C2X DEVELOPMENT PLATFORM*, [Online], Braunschweig, Tyskland: Nordsys, 2016. URL: https://www.nordsys.de/images/sampledats/asimages/C2X/c2xdatasheets/NO_waveBEEplus_db_05_2016_rev1.5_en.pdf (hämtad 2019-04-29).
- [47] *OBU-201E*, *V2X On-Board Unit*, *ETSI TC-ITS protocol stack*, [Online], Unex Technology Corp., 2017. URL: <https://www.unex.com.tw/download/578f366c87d60/35?token=FFFFQDLJVIDIBqAYMHV1pzVJRcE2bLrRwkD8fsfnn> (hämtad 2019-04-29).
- [48] *MobiWAVE On-Board-Unit (OBU)*, *MW-1000 OBU Family*, [Online], Savari, 2017. URL: http://savari.net/wp-content/uploads/2017/05/MW-1000_April2017.pdf (hämtad 2019-04-29).
- [49] *EVK-3300*. *V2X Evaluation Kit*. [Online], Kapsch TrafficCom, u.d. URL: <https://www.kapsch.net/ktc/downloads/datasheets/v2x/Kapsch-KTC-DS-EVK-3300-EN-WEB.pdf?lang=en-US> (hämtad 2019-04-29).
- [50] *MK5 OBU*, *Cohda Wireless' 5th generation market ready On-Board Unit*, [Online], Cohda Wireless, u.d. URL: https://www.cohdawireless.com/wp-content/uploads/2018/08/CW_Product-Brief-sheet-MK5-OBU.pdf (hämtad 2019-04-29).
- [51] *ITS-OB4 Product Brief*, *Powerful V2X Onboard Unit*, [Online], Santa Clara, United States: Commsignia, 2018. URL: http://www.commsignia.com/wp-content/uploads/2019/04/Commsignia_ITS_OB4_ProductBrief.pdf (hämtad 2019-04-29).
- [52] Spielvogen. (2015). Commsignia connected car V2X router. [Elektronisk bild], URL: https://commons.wikimedia.org/wiki/File:Commsignia_connected_car_V2X_router._Spielvogel.jpg (hämtad 2019-05-17).
- [53] Strålsäkerhetsmyndigheten. (2017). Trådlös teknik. [Online], URL: <https://www.stralsakerhetsmyndigheten.se/omraden/magnetfalt-och-tradlos-teknik/tradlos-teknik/> (hämtad 2019-05-25).

- [54] A. Pernestål, “Varning för självkörande bilar”, *Forskning & Framsteg*, nr. 3 mars 2019, [Online]. URL: <https://fof.se/tidning/2019/3/artikel/varning-for-sjalvkorande-bilar> (hämtad 2019-05-25).
- [55] S. A. Miller och B. R. Heard, “The Environmental Impact of Autonomous Vehicles Depends on Adoption Patterns”, *Environmental Science & Technology*, årg. 50, s. 6119–6121, nr. 12 juni 2016, [Online]. URL: <https://pubs.acs.org/doi/pdf/10.1021/acs.est.6b02490> (hämtad 2019-05-25).
- [56] Z. Xu, X. Li, X. Zhao, M. H. Zhang och Z. Wang. (2017). The performance degradation of DSRC and LTE under Scenario I. [Elektronisk bild], URL: <https://www.hindawi.com/journals/jat/2017/2750452/> (hämtad 2019-05-23).
- [57] Z. Xu, X. Li, X. Zhao, M. H. Zhang och Z. Wang. (2017). The DSRC performance versus different communication distance when the vehicle velocity is 120 Km/h. [Elektronisk bild], URL: <https://www.hindawi.com/journals/jat/2017/2750452/> (hämtad 2019-05-23).
- [58] —, (2017). The performance degradation of DSRC and LTE under Scenario II. [Elektronisk bild], URL: <https://www.hindawi.com/journals/jat/2017/2750452/> (hämtad 2019-05-23).
- [59] Rendits, *Test suite for the Rendits V2X router*, [Repository]. URL: <https://github.com/rendits/router>.

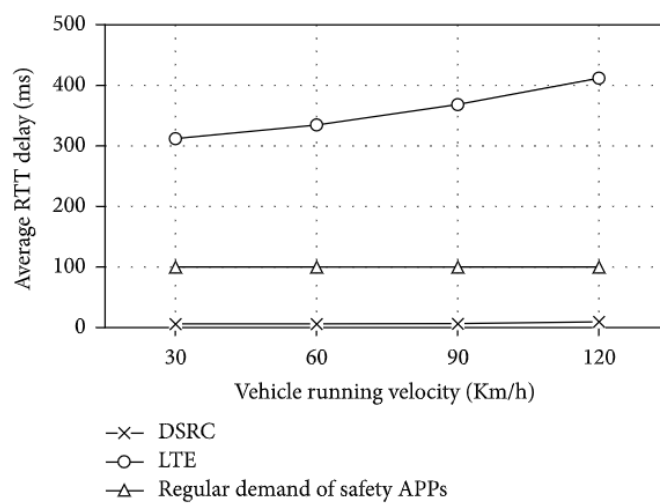
A

Grafer från V2X-test

I Avs. 4.1.3 beskrivs resultat från en studie av DSRC och C-V2X. Denna bilaga innehåller de grafer som refereras till i avsnittet.



Figur A.1: Jämförelse av PLR hos DSRC och C-V2X vid olika hastigheter, när Collision Avoidance-meddelande skickas. Från [56], CC-BY-SA



Figur A.2: Jämförelse av RTT hos DSRC och C-V2X vid olika hastigheter, när Collision Avoidance-meddelande skickas. Från [56], CC-BY-SA

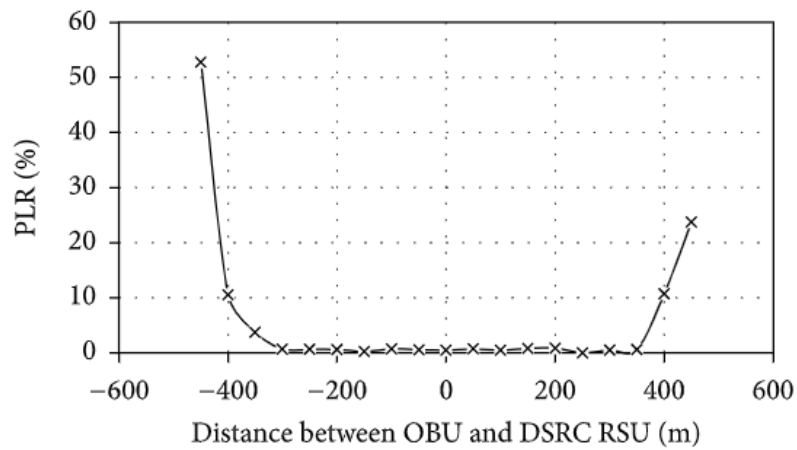


Figure A.3: Mätning av PLR hos DSRC vid olika avstånd mellan OBU och RSU, när Traffic Text Message Broadcast-meddelande skickas. Från [57], CC-BY-SA

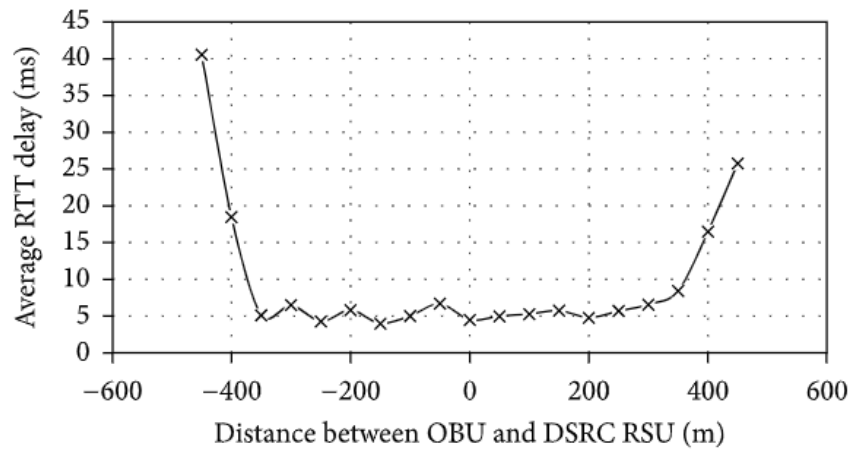
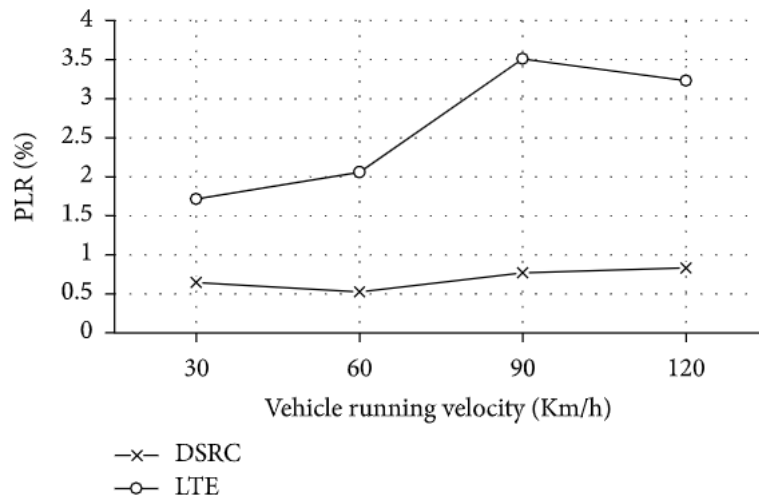
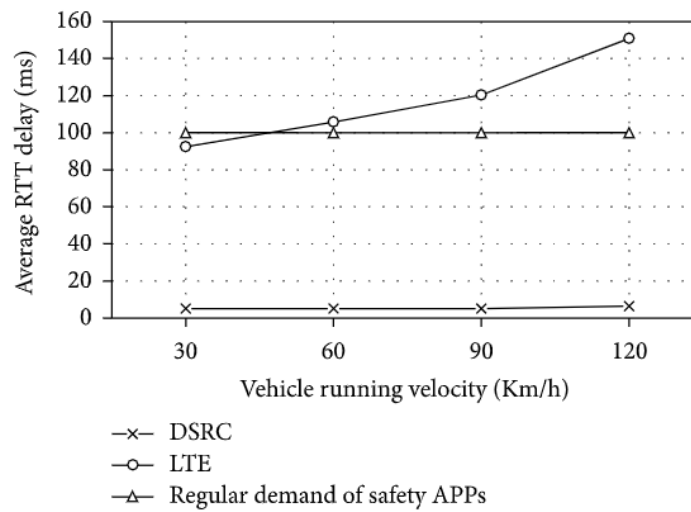


Figure A.4: Mätning av RTT hos DSRC vid olika avstånd mellan OBU och RSU, när Traffic Text Message Broadcast-meddelande skickas. Från [57], CC-BY-SA



Figur A.5: Jämförelse av PLR hos DSRC och C-V2X vid olika hastigheter, när Traffic Text Message Broadcast-meddelande skickas. Från [58], CC-BY-SA



Figur A.6: Jämförelse av RTT hos DSRC och C-V2X vid olika hastigheter, när Traffic Text Message Broadcast-meddelande skickas. Från [58], CC-BY-SA

B

Testkod

I Avs. 3.1 beskrivs hur man testar RTT och resultatet finns i Avs. 4.2. Bilagan innehåller testkod som används för att skicka 100 meddelanden 10 gånger och ta emot dem för att beräkna denna. Detta görs med två V2X-enheter och en PC via en switch. Koden är skriven av Rendits (återgiven med tillstånd av Albin Severinsson) [59].

```
/* Copyright 2018 Rendits
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you
 * may not
 * use this file except in compliance with the License. You may obtain a
 * copy of
 * the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See
 * the
 * License for the specific language governing permissions and
 * limitations under
 * the License.
 */

/* Test suite for the Rendits V2X router. Meant to verify that the router
 * is
 * working properly and to measure the transmission latency.
 */

package com.rendits.testsuite;

/* Standard Java */
import java.io.IOException;
import java.io.FileInputStream;
import java.io.PrintWriter;
import java.nio.ByteBuffer;
import java.util.Properties;
```

```
import java.net.Socket;
import java.net.DatagramSocket;
import java.net.DatagramPacket;
import java.net.InetAddress;
import java.util.concurrent.TimeUnit;
import java.util.concurrent.Executors;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.ScheduledExecutorService;
import java.util.concurrent.ConcurrentHashMap;

/* Logging framework */
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

/* Simple message classes */
import com.rendits.router.SimpleCam;
import com.rendits.router.SimpleDenm;
import com.rendits.router.SimpleIclcm;

public class TestSuite {
    private final static Logger logger =
        LoggerFactory.getLogger(TestSuite.class);
    private final DatagramSocket rcvSocket;
    private static final ExecutorService executor =
        Executors.newCachedThreadPool();
    private static final ScheduledExecutorService scheduler =
        Executors.newScheduledThreadPool(3);

    private final int MAX_UDP_SIZE = 300;
    private int stationID = 100;

    TestSuite(Properties props) throws IOException {
        logger.info("Starting test suite...");

        int portSendCam =
            Integer.parseInt(props.getProperty("portSendCam"));
        int portSendDenm =
            Integer.parseInt(props.getProperty("portSendDenm"));
        int portSendIclcm =
            Integer.parseInt(props.getProperty("portSendIclcm"));
        int portRcvFromRouter =
            Integer.parseInt(props.getProperty("portRcvFromRouter"));
        InetAddress routerAddress =
            InetAddress.getByName(props.getProperty("routerAddress"));

        int configPort =
            Integer.parseInt(props.getProperty("routerConfigPort"));

        rcvSocket = new DatagramSocket(portRcvFromRouter);
```

```
Vehicle vehicle = new Vehicle(stationID);

ConcurrentHashMap highResTimestamps = new ConcurrentHashMap();
CamService camService = new CamService(10, rcvSocket,
    routerAddress,
                                portSendCam, vehicle,
                                scheduler,
                                highResTimestamps);

//IclcmService iclcmService = new IclcmService(1000, rcvSocket,
    routerAddress, portSendIclcm, scheduler);

ReceiveService receiveService = new ReceiveService(rcvSocket);
LatencyTest latencyTest = new LatencyTest(receiveService,
    executor, highResTimestamps);
latencyTest.start();

ConfigService configService = new ConfigService(routerAddress,
    configPort);

/*
Properties newProps = new Properties();
newProps.setProperty("foo", "1");
newProps.setProperty("bar", "2");
newProps.setProperty("portSendIclcm", "6001");
configService.send(newProps);
*/
}

/* Service handling connecting to a router and sending config
 * updates.
 */
private class ConfigService {
    InetAddress routerAddress;
    int configPort;

    ConfigService(InetAddress routerAddress, int configPort) {
        this.routerAddress = routerAddress;
        this.configPort = configPort;
    }

    /* Send a properties object to the router */
    public void send(Properties props) throws IOException {

        /* Open a socket to the router */
        Socket socket = new Socket(routerAddress, configPort);

        /* Create a PrintWriter and send the properties object
```

```
        * over it.
        */
        PrintWriter out = new PrintWriter(socket.getOutputStream(),
            true);
        props.list(out);

        /* Close the socket */
        socket.close();
    }
}

/* Service handling receiving packets from the router */
private class ReceiveService {
    private DatagramSocket rcvSocket;
    private boolean running = true;

    ReceiveService(DatagramSocket rcvSocket) {
        this.rcvSocket = rcvSocket;
    }

    /* Receive a packet from the router */
    public DatagramPacket receive() {
        DatagramPacket packet = new DatagramPacket(new
            byte[MAX_UDP_SIZE], MAX_UDP_SIZE);
        try{
            rcvSocket.receive(packet);
        }catch(IOException e){
            logger.warn("IOException when receiving packet.");
            return null;
        }
        return packet;
    }
}

private class LatencyTest {
    private ReceiveService receiveService;
    private ExecutorService executor;
    private ConcurrentHashMap highResTimestamps;

    private boolean running = true;
    private final int WARMUP_COUNT = 10000;
    private final int TEST_LENGTH = 1000;

    LatencyTest(ReceiveService receiveService,
        ExecutorService executor,
        ConcurrentHashMap highResTimestamps) {

        this.receiveService = receiveService;
        this.executor = executor;
    }
}
```

```
        this.highResTimestamps = highResTimestamps;
    }

    /* Start the latency test */
    public void start() {

        /* Check for any connection */
        DatagramPacket packet;
        packet = receiveService.receive();
        logger.info("Successfully connected to the routing
            application.");

        /* Let the JIT warm up */
        int totalReceived = 0;
        logger.info("Warming up the JIT...");
        for(int i = 0; i < WARMUP_COUNT; i++) {
            packet = receiveService.receive();
            totalReceived++;
            if(totalReceived % 100 == 0) {
                logger.info((int) (totalReceived / (double)
                    WARMUP_COUNT * 100) + "% completed...");
            }
        }

        /* Start the test proper */
        logger.info("Starting latency test...");
        executor.submit(runner);
    }

    private Runnable runner = new Runnable() {
        @Override
        public void run() {
            ByteBuffer buffer;
            long totalDelay = 0;
            long totalReceived = 0;

            for(int i = 0; i < TEST_LENGTH; i++) {
                DatagramPacket packet = receiveService.receive();
                buffer = ByteBuffer.wrap(packet.getData());

                switch(packet.getData()[0]) {
                    case 2:
                        if(totalReceived % 100 == 0) {
                            logger.info((int) (totalReceived / (double)
                                TEST_LENGTH * 100) + "% completed...");
                        }

                        SimpleCam simpleCam = new
                            SimpleCam(packet.getData());
                }
            }
        }
    };
}
```

```
        /* Calculate high-resolution delay */
        int lowResTimestamp =
            simpleCam.getGenerationDeltaTime();
        if(highResTimestamps.containsKey(lowResTimestamp))
        {
            long nanoTimestamp = (long)
                highResTimestamps.get(simpleCam.getGenerationDeltaTime())
            long delay = System.nanoTime() -
                nanoTimestamp;
            highResTimestamps.remove(simpleCam.getGenerationDeltaTime())
            totalDelay += delay;
            totalReceived++;
        }
    }
    long averageDelay = totalDelay / totalReceived;
    double averageDelayMilliseconds = (double) averageDelay
        / 1e6;
    logger.info("Delay test completed. Received: " +
        TEST_LENGTH +
            " messages. Average round-trip delay: " +
            averageDelayMilliseconds + "ms.");
}

};

}

/* Service for periodically sending iCLCM messages */
private class IclcmService {
    private int period;
    private DatagramSocket socket;
    private InetAddress routerAddress;
    private int routerPort;

    private SimpleIclcm simpleIclcm;
    private byte[] buffer;

    IclcmService(int period,
        DatagramSocket socket,
        InetAddress routerAddress,
        int routerPort,
        ScheduledExecutorService scheduler) {

        this.period = period;
        this.socket = socket;
        this.routerAddress = routerAddress;
        this.routerPort = routerPort;

        /* Create a simple iCLCM manually */
    }
}
```

```
simpleIclcm = new SimpleIclcm(stationID,
    (byte) 128, //containerMask
    100, //rearAxleLocation
    0, //controllerType
    1001, //responseTimeConstant
    1001, //responseTimeDelay
    10, //targetLongAcc
    1, //timeHeadway
    3, //cruiseSpeed
    (byte) 128, //lowFrequencyMask
    1, //participantsReady
    0, //startPlatoon
    1, //endOfScenario
    255, //mioID
    10, //mioRange
    11, //mioBearing
    12, //mioRangeRate
    3, //lane
    0, //forwardID
    0, //backwardID
    0, //mergeRequest
    0, //safeToMerge
    1, //flag
    0, //flagTail
    1, //flagHead
    254, //platoonID
    100, //distanceTravelledCz
    2, //intention
    2); //counter

/* Format it as a byte array */
this.buffer = simpleIclcm.asByteArray();

/* Add the runner to the thread pool */
scheduler.scheduleAtFixedRate(runner, period, period,
    TimeUnit.MILLISECONDS);

logger.info("Starting iCLCM service...");
}

private Runnable runner = new Runnable() {

    @Override
    public void run() {
        try{
            DatagramPacket packet = new DatagramPacket(buffer,
                buffer.length);
            packet.setPort(routerPort);
            packet.setAddress(routerAddress);
```

```
        socket.send(packet);
    } catch(IOException e) {
        logger.error("IOException in iCLCM Service.");
    }
    }
};

}

public static void main(String[] args) throws IOException {
    Properties props = new Properties();
    FileInputStream in = new FileInputStream("testsuite.properties");
    props.load(in);
    in.close();
    TestSuite ts = new TestSuite(props);
}
}
```
