Rheinland-Pfälzische Technische Universität Kaiserslautern Landau

Performance Evaluation of Vehicular-to-Everything (V2X) Technologies and Machine Learning (ML) based PHY assistance

Leistungsbewertung von Vehicular-to-Everything (V2X) Technologien und auf maschinellem Lernen (ML) basierender PHY-Unterstützung

von

Raja Sattiraju, M.Sc.

geboren in Chennai, Indien

Vom Fachbereich Elektrotechnik und Informationstechnik
der Rheinland-Pfälzischen Technischen Universität Kaiserslautern-Landau
zur Verleihung des akademischen Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
genehmigte Dissertation

D 386

Tag der mündlichen Prüfung:	13. November 2025
Dekan des Fachbereichs:	Prof. Dr.-Ing. Daniel Görges
Prüfungsvorsitzender:	Prof. Dr.-Ing. Daniel Görges
1. Berichterstatter:	Prof. Dr.-Ing. Hans D. Schotten
2. Berichterstatter:	Prof. Dr.-Ing. Horst Wieker

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Kaiserslautern, 20.11.2024 _____

Raja Sattiraju

Abstract

The first four generations of mobile cellular networks were primarily utilized for personal communications and focused solely on increasing system capacity and user data rates.The fifth generation (5G) mobile communications standard revolutionized the cellular connectivity by adding support for new use cases such as enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communications (URLLC) and massive Machine Type Communications (mMTC). This enabled the foray of cellular communication technologies into new verticals such as Vehicular-to-Everything (V2X) communication, Internet of Things (IoT), industrial communications etc.

Vehicular-to-Everything (V2X) allows vehicles to directly communicate with each other, roadside infrastructure, and other road users to deliver an array of services. It forms a compelling usecase for 5G due to its potential to provide benefits such as road safety, traffic efficiency, smart mobility, environmental sustainability, and driver convenience. Currently, two Radio Access Technologys (RATs) exist for V2X communication: Intelligent Transportation Systems (ITS)-G5, based on the IEEE802.11 standard that is well over 20 years in research and development and Cellular Vehicle-to-everything (C-V2X), based on the newer but more established and global 3GPP standard. Both the technologies have their own pros and cons with ITS-G5 offering ease of deployment due to its light weight protocol and complete decentralized operation whereas C-V2X offers better radio performance, tighter integration with cellular networks etc.

Before deploying any RAT, it is important to extensively evaluate it by means of analytical, empirical and monte-carlo methods. This forms the core of the first part of this thesis where a detailed link and system level performance comparison has been carried out for ITS-G5 and C-V2X. A new link-level simulation framework, namely *pycv2x* has been developed from scratch for evaluating C-V2X whereas for ITS-G5, open source implementation from *ublox* was used. An extensive library of digital signal processing blocks for coding, modulation, channel estimation & equalization, frequency and timing offset correction were developed for C-V2X. The developed simulation framework is tested with the Third Generation Partnership Project (3GPP) reference channel models and the simulation results match the expected values from 3GPP Rel.14 specification.

Once, link level simulations are ready for both C-V2X and ITS-G5, the next step is to run the monte-carlo simulation for a wide variety of channel models. The simulations considered varying channel models ranging from simple AWGN models to multi-path fading models with varying delay profiles. After extensive literature survey, a total of 8 different channel models (from ITU and DSRC) specifically designed for vehicular scenarios were selected and the simulations are carried out. The results show that,in single transmission scheme, C-V2X outperforms ITS-G5 in almost all of the considered channel models with some exceptions with 16QAM and higher coding schemes. With one blind retransmission enabled, C-V2X exhibits a gain of atleast 6 dB and in some cases, reaching as high as 10 dB over ITS-G5.

The link level simulation encompasses the PHY layer performance with a single link only. In order to understand the system capacity, a system level analysis is necessary that considers the upper Media Access Control Layer (MAC) layer schemes. The thresholds for calculating packets errors at system level are derived from link level simulation by means of link-system level mapping curves (SNR - BLER curves). In this regard, a realistic system level simulation framework using real world maps and traffic was developed to evaluate the MAC schemes of ITS-G5 and C-V2X - namely Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) and Semi Persistant Subchannel Selection (SPSS) respectively. The simulation approach is based on bidirectionally coupling both the vehicular traffic simulator (Simulation of Urban Mobility (SUMO)) and the network simulator. The analysis was carried out for European Telecommunications Standards Institute (ETSI) defined scenarios such as highway and Manhattan grid. As an additional novelty, complex scenarios using real world maps and traffic for the cities of Kaiserslautern and Merzig were also considered for simulation that capture the traffic characteristics of both urban and rural areas. Overall, it can be seen that C-V2X has a higher range than ITS-G5 inline with the specification. This can be seen in the form of Packet Error Rate (PER) where ITS-G5 maxes out before C-V2X. In terms of distance, this translates to a range gain of almost 100 m for QPSK coding schemes and almost 200 m for 16QAM schemes. C-V2X also makes use of spectrum efficiently compared to ITS-G5 which can be seen in the form of higher Channel Busy Ratio (CBR). The retransmission gain with C-V2X is significant at higher MCS schemes, lower vehicular densities and higher speeds.

Machine Learning (ML) / Artificial Intelligence (AI) is a topic that generated huge interest among the academia and industry alike due to the availability of data and the processing power necessary to analyze this data. Therefore, it is widely being evaluated and used in almost every engineering domain. The wireless community also started to embrace ML over the last 5 years and many works were carried out to assess its applicability across the entire protocol stack. This forms the core of the second part of this thesis where different ML strategies were evaluated for different signal processing operations such as channel coding, channel estimation, reliability prediction, obstacle detection etc.

Autoencoders are one of the first thought applications of ML at Physical Layer (PHY) layer where a well trained ML model can mimic the behavior of any signal processing block (eventually the entire PHY pipeline). The idea is that if we design a moderately complex ML model and train it with the input and output samples of any given signal processing block (such as coding / modulation), the autoencoder can learn the inherent patterns in the data and after enough iterations will start generating the same data outputs as the considered signal processing block. In this regard, two ML models were developed for turbo decoding and channel estimation. The first one is based on Recurrent Neural Network (RNN) architecture that is good at understanding time series patterns and therefore can decode the inherent sequence dependency of turbo codes. The second is based on Convolutional Neural Network (CNN) architecture that is good at understanding spatial dependencies and is therefore a better candidate for channel estimation which involves averaging operations across frequency subcarriers. The models were trained and compared with legacy signal processing blocks and the results show that the ML based models perform on par or sometimes even outperforming their legacy counterparts in some cases. (CNN based channel estimation is seen to improve the system performance especially at high speed scenarios where there is a high Doppler).

Proactive link adaptation and management is another key area in wireless communication where ML based solutions can bring a huge gain in performance and add to the overall

value in the RF chain. In this regard, a novel reliability prediction scheme based on Long Short Term Memory (LSTM) networks that predicts the Signal to Interference plus Noise Ratio (SINR) for subsequent transmissions based on previous values obtained after channel equalization. This knowledge can help the transmitter to adjust the Modulation & Coding Scheme (MCS) schemes before a Channel Quality Indicator (CQI) feedback from the receiver. By combining the received CQI with the prediction value, the model can be tuned dynamically to make accurate predictions and subsequently optimize link adaptation

Another novel method for detecting the presence/absence of obstacles with ML algorithms on raw Ultra-Wide Band (UWB) waveforms is also investigated in this thesis. Raw UWB waveforms were collected in indoor/outdoor scenarios and were labeled appropriately. 1 denotes the presence of an obstacle (human) and 0 denotes no obstacle. A suite of supervised ML models were trained on the training data set and were used to predict on the test set. The result show that just by using raw waveforms (without any need for further filtering), supervised ML based methods can detect the obstacles with accuracies close to 95 %

Zussammenfassung

Die ersten vier Generationen von Mobilfunknetzen wurden in erster Linie für die persönliche Kommunikation genutzt und konzentrierten sich ausschließlich auf die Erhöhung der Systemkapazität und der Nutzerdatenraten. Der Mobilfunkstandard der fünften Generation (5G) revolutionierte die zellulare Konnektivität, indem er die Unterstützung für neue Anwendungsfälle wie Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communications (URLLC) und Massive Machine Type Communications (mMTC) einführte. Dies ermöglichte das Vordringen der Mobilfunktechnologien in neue Bereiche wie die Vehicular-to-Everything (V2X)-Kommunikation, das Internet der Dinge (IoT), die industrielle Kommunikation usw.

V2X ermöglicht es Fahrzeugen, direkt miteinander, mit der straßenseitigen Infrastruktur und mit anderen Verkehrsteilnehmern zu kommunizieren, um eine Reihe von Diensten anzubieten. Aufgrund des Potenzials, Vorteile wie Verkehrssicherheit, Verkehrseffizienz, intelligente Mobilität, Umweltverträglichkeit und Fahrerkomfort zu bieten, ist dies ein überzeugender Anwendungsfall für 5G. Derzeit gibt es zwei Funkzugangstechnologien (RATs) zur Realisierung der V2X-Kommunikation - Intelligent Transportation Systems (ITS)-G5, das auf dem IEEE802.11-Standard basiert, der seit über 20 Jahren erforscht und entwickelt wird, und Cellular Vehicle-to-everything (C-V2X), das auf dem neueren, aber etablierteren und globalen 3GPP-Standard basiert. Beide Technologien haben ihre eigenen Vor- und Nachteile, wobei ITS-G5 aufgrund seines leichtgewichtigen Protokolls und seines vollständig dezentralisierten Betriebs eine einfache Einführung bietet, während C-V2X eine höhere Leistungsfähigkeit, eine engere Integration mit Mobilfunknetzen usw. bietet.

Vor dem Einsatz einer RAT ist es wichtig, diese mit Hilfe von analytischen, empirischen und Monte-Carlo-Methoden umfassend zu bewerten. Dies bildet den Kern des ersten Teils dieser Arbeit, in dem ein detaillierter Leistungsvergleich auf Verbindungs- und Systemebene für ITS-G5 und C-V2X durchgeführt wurde. Für die Evaluierung von C-V2X wurde ein neues Simulationsframework auf Verbindungsebene, nämlich *pycv2x*, von Grund auf neu entwickelt, während für ITS-G5 die Open-Source-Implementierung von ublox verwendet wurde. Für C-V2X wurde eine umfangreiche Bibliothek von digitalen Signalverarbeitungsblöcken für Kodierung, Modulation, Kanalschätzung und sowie, Frequenz- und Zeitversatzkorrektur entwickelt. Der entwickelte Simulationsumgebung wurde mit den 3GPP-Referenzkanälen getestet und die Simulationsergebnisse stimmen mit den erwarteten Werten der 3GPP Rel.14 Spezifikation validiert.

Als nächster Schritt die Monte-Carlo-Simulation für eine Vielzahl von Kanalmodellen durchgeführt. Bei den Simulationen wurden verschiedene Kanalmodelle berücksichtigt, die von einfachen AWGN-Modellen bis hin zu Mehrwege-Schwundmodellen mit unterschiedlichen Verzögerungsprofilen reichen. Nach einer umfangreichen Literaturrecherche wurden insgesamt 8 verschiedene Kanalmodelle (von ITU und DSRC) ausgewählt, die speziell für Fahrzeugszenarien entwickelt wurden, ausgewählt und die Simulationen

durchgeführt. Die Ergebnisse zeigen, dass C-V2X bei einem einzigen Übertragungsschema in fast allen betrachteten Kanalmodellen besser abschneidet als ITS-G5, mit einigen Ausnahmen bei 16QAM und höheren Kodierungsschemata. Mit einer aktivierten blinden Neuübertragung zeigt C-V2X einen Gewinn von mindestens 6 dB und erreicht in einigen Fällen sogar 10 dB gegenüber ITS-G5.

Während die Simulation auf Link-Ebene lediglich die Leistungsfähig eines einzelnen Links betrachtet, werden bei der Analyse auf Systemebene die oberen MAC-Schichten (Media Access Control Layer) berücksichtigt. Die Schwellenwerte für die Berechnung von Paketfehlern auf Systemebene werden aus der Simulation auf Link-Ebene mit Hilfe von Link-System-Level-Mapping-Kurven (SNR-BLER-Kurven) abgeleitet. In diesem Zusammenhang wurde ein realistischer Simulationsumgebung auf Systemebene mit realen Verkehrsszenarien entwickelt, um die MAC-Schemata von ITS-G5 und C-V2X zu bewerten - nämlich Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) bzw. Semi Persistant Subchannel Selection (SPSS). Der Simulationsansatz basiert auf der bidirektionalen Kopplung des Fahrzeugverkehrssimulators (Simulation of Urban Mobility (SUMO)) und des Netzsimulators. Die Analyse wurde für Szenarien wie Autobahn und Manhattan-Netz durchgeführt, welcher von der European Telecommunications Standards Institute (ETSI) definiert wurden. Als zusätzliche Neuheit wurden auch komplexe Szenarien mit realen Karten und Verkehr für die Städte Kaiserslautern und Merzig für die Simulation berücksichtigt, die die Verkehrscharakteristiken sowohl städtischer als auch ländlicher Gebiete erfassen. Insgesamt zeigt sich, dass C-V2X eine höhere Reichweite als ITS-G5 gemäß der Spezifikation aufweist. Dies zeigt sich in Form der Paketfehlerrate (PER), bei der ITS-G5 vor C-V2X das Maximum erreicht. In Bezug auf die Entfernung bedeutet dies einen Reichweitengewinn von fast 100 m bei Quadrature Phase Shift Keying (QPSK)-Kodierungsschemata und fast 200 m bei 16-Quadrature Amplitude Modulation (QAM) Schemata. C-V2X nutzt das Spektrum im Vergleich zu ITS-G5 auch effizienter, was sich in Form einer höheren Channel Busy Ratio (CBR) zeigt. Der Gewinn bei der Weiterübertragung mit C-V2X ist bei höheren Modulations- und Kodierungsverfahren (MCS), niedrigeren Fahrzeugdichten und höheren Geschwindigkeiten erheblich.

Machine Learning (ML) / Künstliche Intelligenz (KI) ist ein Thema, welches aufgrund der Verfügbarkeit von Daten und der zur Analyse dieser Daten erforderlichen Rechenleistung sowohl in der Wissenschaft als auch in der Industrie auf großes Interesse stößt. Daher wird es in fast allen technischen Bereichen untersucht und eingesetzt. Auch die Wireless-Gemeinschaft hat in den letzten 5 Jahren begonnen, sich mit ML zu beschäftigen, und viele Arbeiten wurden durchgeführt, um die Anwendbarkeit von ML über den gesamten Netzwerkprotokollstapel hinweg zu bewerten. Dies bildet den Kern des zweiten Teils dieser Arbeit, in dem verschiedene ML-Strategien für verschiedene Signalverarbeitungsoperationen wie Kanalcodierung, Kanalschätzung, Zuverlässigkeitsvorhersage, Hinderniserkennung usw. evaluiert wurden.

Autoencoder sind eine der ersten denkbaren Anwendungen von ML auf der PHY-Schicht. Dabei versucht ein gut trainiertes ML-Modell das Verhalten eines beliebigen Signalverarbeitungsblocks (eventuell der PHY-Pipeline) nachzuahmen. Die Idee ist, dass, wenn ein mäßig komplexes ML-Modell entwerfen und es mit den Eingangs- und Ausgangsmustern eines beliebigen Signalverarbeitungsblocks (z. B. Codierung/Modulation) trainiert wird, der Autoencoder die inhärenten Muster in den Daten erlernen kann und nach genügend Iterationen die gleichen Datenausgaben wie der betrachtete Signalverarbeitungsblock erzeugt. In diesem Zusammenhang wurden zwei ML-Modelle für Turbo-Decodierung und Kanalschätzung entwickelt. Das erste Modell basiert auf der Architektur eines Recurrent

Neural Network (RNN), welches für Zeitreihenmuster geeignet ist und daher die inhärente Sequenzabhängigkeit von Turbocodes dekodieren kann. Das zweite Modell basiert auf der Architektur des Convolutional Neural Network (CNN), Welches räumliche Abhängigkeiten berücksichtigt und daher ein besserer Kandidat für die Kanalschätzung ist, die Mittelungsoperationen über Frequenzunterträger beinhaltet. Die Modelle wurden trainiert und mit herkömmlichen Signalverarbeitungsblöcken verglichen. Die Ergebnisse zeigen, dass die ML-basierten Modelle gleich gut oder in einigen Fällen sogar besser abschneiden als die herkömmlichen Modelle. Die CNN-basierte Kanalschätzung verbessert die Systemleistung insbesondere bei Hochgeschwindigkeitsszenarien mit hohem Doppleranteil.

Proaktive Linkadaption und -verwaltung ist ein weiterer Schlüsselbereich in der drahtlosen Kommunikation, in dem ML-basierte Lösungen einen enormen Leistungszuwachs bringen und den Leistungsfähigkeit in der RF-Kette erhöhen können. In diesem Zusammenhang wurde ein neuartiges Zuverlässigkeitsvorhersageverfahren auf der Grundlage von LSTM-Netzwerken (Long-Short-Term Memory) entwickelt, das die SINR für nachfolgende Übertragungen auf der Grundlage früherer, nach dem Kanalausgleich erhaltener Werte vorhersagt. Dieses Wissen kann dem Sender helfen, die MCS-Schemata vor einer CQI-Rückmeldung des Empfängers anzupassen. Durch die Kombination des empfangenen CQI mit dem Vorhersagewert kann das Modell dynamisch abgestimmt werden, um genaue Vorhersagen zu treffen und die Anpassung der Verbindung zu optimieren.

Eine weitere neuartige Methode zur Erkennung des Vorhandenseins von Hindernissen mit ML-Algorithmen auf rohen UWB-Wellenformen wird ebenfalls in dieser Arbeit untersucht. Rohe UWB-Wellenformen wurden in Innen- und Außenszenarien aufgenommen und entsprechend gekennzeichnet. 1 steht für das Vorhandensein eines Hindernisses (Mensch) und 0 für kein Hindernis. Eine Reihe von überwachten ML-Modellen wurde auf dem Trainingsdatensatz trainiert und zur Vorhersage auf dem Testdatensatz verwendet. Die Ergebnisse zeigen, dass überwachte ML-basierte Methoden allein durch die Verwendung roher Wellenformen (ohne weitere Filterung) die Hindernisse mit einer Genauigkeit von fast 95% erkennen können.

Contents

List of Figures

List of Tables

Publications

As First Author

1. R. Sattiraju, D. Wang, A. Weinand and H. D. Schotten, "Link Level Performance Comparison of C-V2X and ITS-G5 for Vehicular Channel Models," *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, Antwerp, Belgium, 2020, pp. 1-7.
2. Sattiraju, Raja, Andreas Klein, Lianghai Ji, Chan Zhou, Omer Bulakci, Josef Eichinger, Nandish P. Kuruvatti, and Hans D. Schotten. "Virtual cell sectoring for enhancing resource allocation and reuse in network controlled D2D communication." In *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pp. 1-6. IEEE, 2015
3. Sattiraju, Raja, Jasper Siemons, Mohammad Soliman, Wasim Alshrafi, Fabian Rein, and Hans D. Schotten. "Design of a highly reliable wireless module for ultra-low-latency short range applications." In *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*, pp. 1-4. IEEE, 2017
4. Sattiraju, Raja, Jacob Kochems, and Hans D. Schotten. "Machine learning based obstacle detection for Automatic Train Pairing." In *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*, pp. 1-4. IEEE, 2017
5. Sattiraju, Raja, Andreas Weinand, and Hans D. Schotten. "Channel Estimation in C-V2X using Deep Learning." In *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp. 1-5. IEEE, 2019
6. Sattiraju, Raja, Jacob Kochems, and Hans D. Schotten. "To Supervise or not-ML based UWB Obstacle Detection." In Mobile Communication-Technologies and Applications; *24. ITG-Symposium*, pp. 1-6. VDE, 2019
7. Sattiraju, Raja, Andreas Weinand, and Hans D. Schotten. "AI-assisted PHY technologies for 6G and beyond wireless networks.", *1st 6G Wireless Summit*, Levi, Finland, March 2019
8. Sattiraju, Raja, Andreas Weinand, and Hans D. Schotten. "Performance analysis of deep learning based on recurrent neural networks for channel coding." In *2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp. 1-6. IEEE, 2018
9. Sattiraju, Raja, Pratip Chakraborty, and Hans D. Schotten. "Reliability analysis of a wireless transmission as a repairable system." In *2014 IEEE Globecom Workshops (GC Wkshps)*, pp. 1397-1401. IEEE, 2014

10. Sattiraju, Raja, and Hans D. Schotten. "Reliability modeling, analysis and prediction of wireless mobile communications." In *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, pp. 1-6. IEEE, 2014

11. Schotten, H. D., Sattiraju, R., Serrano, D. G., Ren, Z., & Fertl, P. (2014, June). Availability indication as key enabler for ultra-reliable communication in 5G. In *2014 European Conference on Networks and Communications (EuCNC)* (pp. 1-5). IEEE

As Second Author and contributor

1. Wang, Donglin, Raja R. Sattiraju, Andreas Weinand, and Hans D. Schotten. "System-level simulator of LTE sidelink C-V2X communication for 5g." In *Mobile Communication-Technologies and Applications; 24. ITG-Symposium*, pp. 1-5. VDE, 2019

2. Wang, Donglin, Raja R. Sattiraju, and Hans D. Schotten. "Performances of C-V2X Communication on Highway under Varying Channel Propagation Models." In *2018 10th International Conference on Communications, Circuits and Systems (ICCCAS)*, pp. 305-309. IEEE, 2018

3. Wang, Donglin, Raja R. Sattiraju, Anjie Qiu, Sanket Partani, and Hans D. Schotten. "Effect of Retransmissions on the Performance of C-V2X Communication for 5G." In *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, pp. 1-7. IEEE, 2020.

4. Weinand, Andreas, Raja Sattiraju, Michael Karrenbauer, and Hans D. Schotten. "Supervised Learning for Physical Layer based Message Authentication in URLLC scenarios." In *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pp. 1-7. IEEE, 2019.

5. Weinand, Andreas, Michael Karrenbauer, Raja Sattiraju, and Hans Schotten. "Application of machine learning for channel based message authentication in mission critical machine type communication." In *European Wireless 2017; 23th European Wireless Conference*, pp. 1-5. VDE, 2017.

6. Klein, Andreas, Alexander Rauch, Raja R. Sattiraju, and Hans D. Schotten. "Achievable Performance Gains Using Movement Prediction and Advanced 3D System Modeling." In *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, pp. 1-5. IEEE, 2014.

7. A. Qiu, D. Wang, S. Partani, R. Sattiraju and H. D. Schotten, "Mitigating Broadcast Storm Problem in VANET when Parked Cars being awaken as Relays," *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chongqing, China, 2020, pp. 631-639

8. Ji, Lianghai, Andreas Klein, Nandish Kuruvatti, Raja Sattiraju, and Hans D. Schotten. "Dynamic Context-aware optimization of D2D communications." In 2014 *IEEE 79th Vehicular Technology Conference (VTC Spring)*, pp. 1-5. IEEE, 2014.

9. Kuruvatti, Nandish P., Andreas Klein, Lianghai Ji, Chan Zhou, Omer Bulakci, Josef Eichinger, Raja Sattiraju, and Hans D. Schotten. "Robustness of location based

D2D resource allocation against positioning errors." In 2015 *IEEE 81st Vehicular Technology Conference (VTC Spring)*, pp. 1-6. IEEE, 2015.

10. D.F. Kuelzer, M. Kasparick, A. Palaios, R. Sattiraju, O. D. Ramos-Cantor, D. Wieruch, H. Tchouankem, F. Goettsch, P. Geuer, J. Schwardmann, G. Fettweis, H.D. Schotten and S. Stanczak, "AI4Mobile: Use Cases and Challenges of AI-based QoS Prediction for High-Mobility Scenarios." *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring).* Helsinki, Finland, 2021

11. A. Palaios, P. Geuer, J. Fink, D.F. Kuelzer, F. Goettsch, M. Kasparick, D. Schaeufele, R. Hernangomez, S. Partani, R. Sattiraju, A. Kumar, F. Burmeister, A. Weinand, C. Vielhaus, F.H.P. Fitzek, G. Fettweis, H.D. Schotten, and S. Stanczak, "Network Under Control: Multi-Vehicle E2E Measurements for AI-Based QoS Prediction." *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)* 2021, September 13-16, in Helsinki, Finland

12. A. Palaios, C. Vielhaus, D.F. Kuelzer, P. Geuer, R. Sattiraju, J. Fink, M. Kasparick, C. Watermann, G. Fettweis, F.H.P. Fitzek, H.D. Schotten, S. Stanczak, "Effect of Spatial, Temporal and Network Features on Uplink and Downlink Throughput Prediction.", *IEEE 4th 5G World Forum (5GWF)* 2021, October 13-15, Virtual Conference, (Best Workshop Paper Award)

1 Introduction

The year 1886 is regarded as the year of birth of the modern automobile when the German inventor Carl Benz filed for a patent titled *vehicle powered by a gas engine*. The Benz Patent-Motorwagen was a three wheeled automobile with a rear engine that contained many innovations such as steel spoked wheels, rubber tires, toothed rack steering, leaf spring suspension and a chain drive with single-speed transmission. However, it was Henry Ford with his Ford Model T and the implementation of a moving assembly line that turned the automobile from a luxury into a necessity by making it cheap, versatile and easy to maintain.This became the technological starting point for the automobile as a medium for new ideas with which future developers could work. The early 20th century witnessed many technological breakthroughs for the automobile such as electric ignition system, independent suspension, four wheel brakes, multi-valve and overhead camshaft engines, hydraulic brakes, torque converters, automatic gearbox etc. By 1930, most of the mechanical technology used in today's automobiles had been invented.

The introduction of Government regulations for exhaust emissions and the development of low cost per function solid state digital electronics in 1970's ushered in a new digital era for the automobile. The relevance of electronics in the automobile have grown substantially with breakthrough developments such as Electronic fuel injection, Adaptive transmission control, Cruise control including safety features such as Airbags, Anti-lock braking systems etc. Eletronics dominated more than 90% of all the automotive technological innovations in the modern era and continues to do so in the early 21st century. There are a multitude of sensors performing both critical and non critical tasks. Current generation of vehicles are equipped with enough electronics to realize full autonomous functionality.

With increasing vehicle penetration in almost all the corners of the globe, new traffic problems arose. Congestion and traffic jams were a constant sight in addition to increasing number of accidents as well as Green House Gas (GHG) emissions. Therefore, early 21st century shifted the focus to increasing traffic safety and efficiency. Early systems included simple traffic announcements over radio based on crowdsourced information. Later on, navigation systems based on Global Positioning System (GPS) were developed that can dynamically adapt the driving route based on traffic information. Currently, providers such as Google Maps, HERE maps are being used by almost every manufacturer that provide rich and curated navigational as well as contextual information.

Fully autonomous driving was seen as a game changer inorder to mitigate traffic accidents as well as increase traffic efficiency. The ability of a car to navigate by itself provides endless possibilities to revolutionize the travel habits of people. Contrary to *ownership*, fully autonomous driving introduced new concepts centered around *usership* such as on-demand mobility, Mobility as a Service (MaaS) etc.

The continuous evolution of mobile communication systems resulted in support for different industry verticals in addition to people communication. Out of these verticals, three domains stand out - vehicular communication, industrial communication and IoT. The integration of mobile communication systems with these domains brings in benefits in terms

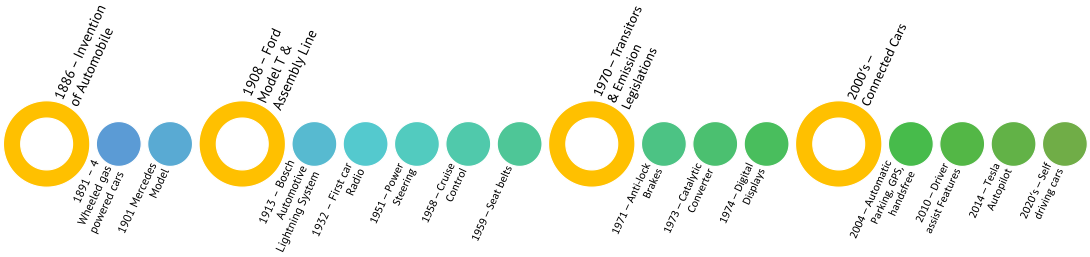


Figure 1.1: Technological Evolution of Vehicles

of increased safety and efficiency at decreased operating costs. Connected vehicles, i.e., vehicles that can talk to each other and other traffic entities provides unique oppurtunities to make our transport systems safer, cleaner, more efficient and more user-friendly. Since the beginning of 20th century, connected vehicles, electromobility and autonomous driving dominated the innovations carried out in the automotive sector. Each one of them is a big study by itself, we limit our discussion to connected vehicles within the scope of this thesis.

1.1 A brief history of communications

It was in the year 1729 that an English scientist named Stephen Gray discovered the property of electrical conduction and insulation, that laid the foundation for electrical communication. Thanks to the inventions of electro-meter in 1746 by Daniel Gralath and electromechanical generator in 1831 by Michael Faraday, people could generate and measure electrical charge (or more specifically the electrical potential difference) between any two contact points. The real breakthrough for electrical communication came in the form of telegraph developed by Samuel Morse and other inventors who commissioned the first telegraphic link between Washington and Baltimore in 1844. The telegraph worked by transmitting electrical signals over a wire laid between stations. Although highly successful, the telegraph was limited to sending or receiving one message at a time. Later, Alexander Graham Bell theorized the possibility of transmitting multiple messages at varying tones simultaneously, thus leading him to invent the telephone in 1870. Although limited to short distances at the beginning, the availability of vacuum triodes for amplification of the signal at regular intervals led to long distance telephony and the establishment of companies such as Bell Telephone company (in 1877 and subsequently became American Telephone and Telegraph Company (AT&T) in 1885) to manage telephone connections.

The first communication systems were analog in the sense that they used continuous analog signals to transmit data, generally using analog modulation methods. Hence, they were highly susceptible to various distortions and interference. The invention of transistor in 1960s pave the way for digital communication where the analog signals were digitized using various techniques prior to transmission. The subsequent Very Large Scale Integration (VLSI) technology made it possible to implement, in hardware, the digital encoders/decoders based on Pulse Coded Modulation (PCM) (developed by Alec Reeves in 1937[150]). Based on these inventions, Bell Laboratories were able to install the first carrier system called T-1 in 1962[146] that used Time Division Multiplexing (TDM) to simultaneously transmit 24 telephone calls over a single copper line. This paved the way for Integrated Services Digital Network (ISDN), a system that allowed multiple services such

as voice, data and video to be integrated using TDM and transmitting them simultaneously in digital form.

Telephone communication took a giant leap forward with the demonstration of Marconi in 1901, in which he showed the possibility of wireless communication by successfully transmitting a radio signal from Cornwall, England to Neufoundland, Canada over a distance of more than 2500 km. Subsequent inventions such as vacuum triode by De Forest in 1906 and AM superheterodyne receiver by Armstrong during World War I led to the establishment of Amplitude Modulation (AM) Radio broadcasting[146]. This was followed by other useful inventions such as Frequency Modulation (FM) by Armstrong and Oscilloscope by Allen Dumont that helped realize wireless networks and devices in hardware. The launch of the first communication satellite, the *Early Bird* in 1967 made it possible to transmit multiple phone calls over very long distances.

Fuelled by the demand for telephone connectivity during periods of mobility prompted AT&T to launch a commercial Mobile Telephone Service (MTS) in 1949. This system allowed users to place and receive telephone calls from automobiles. However, due to the constraints of technology, this system was not economically sustainable nor scalable due to the availability of only three radio channels effectively meaning that only three users in any given city would be able to make or receive calls at any given time. The concept of cellular technology was proposed by AT&T Bell Laboratories in 1968[47] , which allowed frequency reuse in small adjacent areas (called *cells*) made widespread adoption of mobile telephones economically feasible. However, due to the unavailability of required technology, the first generation (1G) commercial cellular system was launched in Japan by Nippon Telegraph and Telephone (NTT) only in 1979. From then, a new generation of cellular networks were developed and rolled out every decade ranging from 2G systems in 1990s to the current launch ready 5G systems.

Attention then turned towards achieving higher transmission rates using more robust modulation techniques and at the same time to have a reliable data transfer. Some of these goals were achieved by the use of adaptive equalization of Lucky proposed in 1965 [113] and some pioneering works done by G. Ungerboeck [162] in the area of channel coding and modulation. Higher data rates were achieved by using higher order modulation techniques such as M-Phase Shift Keying (PSK), M-QAM etc. However, it was not until 1948, when Claude Shannon, with his famous paper[163] laid solid theoretical foundations for digital communication, that proved the possibility to have *error free* transmission of information over a channel corrupted with noise as long as the rate of transmission is less that what is defined as channel capacity. This spawned a whole new research into the field of information theory that forms the basis for modern digital communication. Although it was Morse who recognized the need for such coding efficiency (which was evident from his usage of binary Morse code for the telegraph) a full century before, it was Shannon who laid out principles on how the symbols of communication are transmitted, how the transmitted symbols convey meaning, and the effect of the received meaning. His work led to huge amounts of research in the field of source coding resulting in error correcting block codes such as Golay [69], Hamming[73] and Bose Chaudhuri Hocquenghem (BCH)[24] codes. In these block coding methods, a block of k information bits is mapped into n digits forming a codeword. However, this one to one mapping process is time consuming and cannot be implemented in real-time. A new class of error-correcting codes, called convolutional codes, first proposed by Forney in 1960s [61] and subsequently developed during 1970s, emerged that were easy to implement and could work in real-time. However, they were

still far behind in terms of achieving the optimum Shannon capacity. Turbo codes, proposed by Berrou, Glavieux and Thithimajshima in 1993 [22], filled this gap by achieving near Shannon capacities and are used in almost all of the current digital communication systems.

A careful look at the mobile communication evolution shows that the first cellular phones were initially used in vehicles before general everyday human use. The cellular technologies made a full circle by *re-supporting* V2X services as part of the 5G communication system.

1.2 Vehicular Communication - An Overview

Vehicular Communication / Vehicular Networking in general can be classified into two categories - intra vehicular communication referring to the communication between various subsystems inside the vehicle and inter vehicular communication that refers to the communication between vehicles and other road entities. In this section, we briefly cover the wired and wireless network technologies that are used in the vehicles today.

1.2.1 Intra-Vehicular Networks

Traditional in-vehicle networking deals with wired databus systems where the communications between components are multiplexed over a shared medium. The word "bus" comes from the Latin word "omnibus", which means "for all". Thus, the point-to-point links between the devices and their controlling Electronic Control Units (ECUs) were replaced by a single digital link on which all the information is transmitted serially and multiplexed in time. Consequently, this requires definition of new rules (or protocols) for managing communications and, in particular, for granting bus access. The following points outline the distinguishing characteristics of a serial bus compared to the point-to-point links.

Single Input and Output Each connected ECU and its subcomponents (herewith referred to as a nodes) share a single input and output. Thus, when one ECU sends information on a bus, the other ECUs receive the information at almost the same time.

Message Addressing Since all the nodes can receive the information simultaneously, it is imperative for them to understand to whom the message is addressed to. In this regard, a general distinction is made between sender-selective and receiver-selective addressing. With sender-selective addressing, the sender defines the intended receiver using a unique address. In contrast, receiver-selective addressing identifies the information to be sent which is in turn subscribed by various receivers.

Channel Access and Contention Due to the shared nature of the access medium, the nodes have to compete with each other inorder to get their messages transmitted on the bus. This procedure is called a channel access method. Techniques such as Time Division Multiple Access (TDMA), Carrier Sense Multiple Access (CSMA) allow for a single channel to be shared by multiple nodes with little or no collisions of messages. A collision occurs when two or more nodes transmit the same information at the same time.

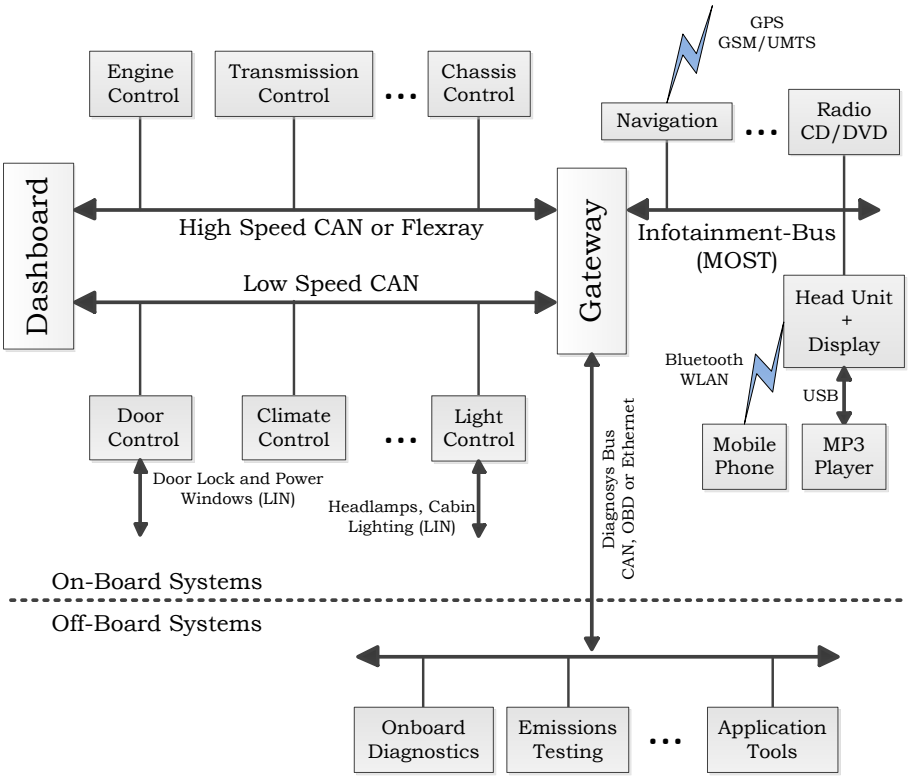


Figure 1.2: Bus Networks in today's cars

In the early 1980s, automotive Original Equipment Manufacturers (OEMs) started to develop their own fieldbus technologies. An example of such an early bus system is the Cyclic Redundancy Check (CRC) display system introduced in Buick vehicles by General Motors (GM) in 1984-85 [42]. Over time, with many automotive manufacturers sharing the same supply vendors, there was a need for standardization [131]. This gave birth to the first standardized fieldbus technology called Controller Area Network (CAN) which went to become the most successful and the most used fieldbus in the automotive industry. Soon, it became clear that a single solution cannot satisfy the diverse and contrasting communication requirements of the exploding electronic devices and this led to the development of application specific bus networks such as Flexray, Local Interconnect Network (LIN), Media Oriented Systems Transport (MOST). Today's cars encapsulate the in-vehicle features into several functional domains that use heterogenous networks depending upon functional requirements as outlined in Figure 1.2. It can be seen that the different bus networks are interconnected by so called gateways that facilitate inter-domain communication [45].

Apart from the standardized databus systems, there are also many proprietary solutions developed by individual companies. Most notable of these include the Vehicle Area Network (VAN) bus developed by PSA Peugeot Citroen and Renault. VAN is a serial protocol capable of speeds up to 125 kbit/s with PHY layer similar to CAN but uses differential manchester encoding. It is used in Peugeot 406 models for body electronics control, entertainment systems, steering wheel controls, multifunctional display etc. Philips Lite Automotive NETwork (PLANET) is an in-vehicle serial network designed by Philips for safety critical passenger systems (airbags) that uses 2 wires for transmisson of data, power and clock with its custom protocol. The rapid growth of embedded microcontrollers in the car also led to the use of Serial Peripheral Interface (SPI) and Inter-Integrated Circuit (I2C)

technologies for connecting the peripheral Integrated Circuits (ICs) with micro controllers over short distance as well as for intra-board communication. Safe-by-Wire is another master-slave network architecture intended mainly for smart airbag deployment where the current CAN and Local Interconnect Network (LIN) architectures cannot provide adequate reliability and deterministic properties [26]. Two other proprietary bus protocols based on master-slave architecture were also designed by Motorola namely Motorola Interconnect (MI) that is similar to LIN to be used in comfort electronics such as seats, mirrors etc. and Distributed Systems Interface (DSI) for safety related applications [131].

Some other technologies for entertainment and data intensive applications include Domestic Digital Bus (D2B) system [39] from Motorola with a ring/star optic fibre supporting communications up to 20 MB/s and Mobile Multimedia Link (MML) from Delphi systems with a master-slave optic fibre supporting up to 100 MB/s. Modified versions of computer networking technologies such as ITS Data Bus (IDB)-1394 (Automotive Firewire) and Universal Serial Bus (USB) have also been proposed for media applications [44].

The proliferation of wireless technologies inside vehicles started with Bluetooth and is still the defacto technology for interfacing the user device (i.e., smartphone) with the vehicle's infotainment system. In December 1999, Bluetooth Special Interest Group (SIG) formed the Car Working Group whose task is to primarily investigate the possibilities of using Bluetooth for in-vehicle communication. The first in-vehicle application that was standardized as part of its efforts was the Hands-Free profile, that enabled making and taking calls from the mobile phone using the in-vehicle audio system. The subsequent standards enabled deeper integration with applications such as wireless music streaming, application control, remote keyless systems along with support for various in-vehicle wearables. Bluetooth is also extensively used under the hood for connecting to diagnostic systems (e.g., via a Bluetooth enabled On Board Diagnostics-II (OBD-II) adapter) for the purpose of transferring diagnostic information and alerts in real-time.

1.2.1.1 Current Trends & Roadmap

The development of novel applications such as Advanced Driver Assistance Systems (ADAS) and the increased penetration of hybrid vehicles with many electric/electronic drive components called for bounded latencies and increased bandwidth. In this context, it is foreseen that Ethernet is going to play a major role in the next generation cars allowing multiple in-vehicle systems to simultaneously access information from various subsystems [17]. A new IEEE 802.3bp task force named Reduced Twister Pair Gigabit Ethernet (RTPGE) is in the process of standardizing a new PHY layer for transmission of 1 Gbit/s over 15 m of single twisted wire pair.

These PHY layer modifications are complemented by many novel upper layer techniques, most notably the Society of Automotive Engineers (SAE) AS6802, time-triggered Ethernet [23] and the IEEE 802.1 Time Sensitive Networking (TSN) group. The latter uses a comprehensive set of operations based on standards such as IEEE 802.1AS for time synchronization, IEEE 802.1Qat for stream reservations, IEEE 802.1Qav for traffic shaping, IEEE 802.1Qcb and IEEE 802.1Qbu for redundancy etc. Other next-gen novel features include energy efficiency by using IEEE 802.11at Power over Ethernet (PoE) standard that transmits both the power and data simultaneously, thus saving on wiring, weight and the number of connectors.

The wiring harness itself has been an optimization factor in the quest to make cars lighter and energy efficient. Modern automobiles contain on average 2 km of wiring weighing more than 30 kg, thus representing a significant cost in terms of money, weight, and the space needed. This has motivated research efforts into investigating the feasibility of replacing some of the connectors with wireless links using existing wireless technologies such as Bluetooth, Wi-Fi, ZigBee, UWB etc. In particular,the mmWave technology such as the 60 GHz links have shown promise for connecting microcontrollers over a short range due to the high bandwidth they offer. However, wireless technologies have inherent flaws in terms of security, reliability, real-timeliness etc. Therefore, apart from tire pressure sensors (where their usage is obvious), wireless technologies are yet to find widepread acceptance for in-vehicle networking.

Traditional wiring is made using copper, a metal known for its properties such as high electrical conductance, flexibility, anti-corrosiveness and ease of manufacture/assembly. Owing to the high penetration of the electric vehicles that require longer and thicker cables, current market demand for copper in automotive wire harnesses is expected to grow to ≈1.15 million tonnes by 2020 compared to ≈ 1 million tonnes in 2015. Aluminium has been considered as replacement to copper owing to its lighter weight and cost albeit offering lower flexibility & electrical conductivity and being highly corrosive, However, with advances in metallurgy [179, 87] that provided cost effective solutions to these problems, aluminum is finding increased attention to be used in automotive wiring harnesses.

With advances in Light Emitting Diode (LED) technology and manufacturing, Visible Light Communication (VLC) is also investigated as a potential replacement for some cable connectors[136]. VLC offers high data rates and ultra-wide bandwidth, while bringing no electromagnetic inferences. VLC spectrum has the added advantage of being license free, operating at frequencies between 400 THz and 750 THz. However, the technology needs to overcome challenges such as noise resilience and robustness required to meet automotive standards.

1.2.2 Inter-Vehicular Networks

With the advent of ITS, it became necessary for the vehicles and other traffic entities to cooperate to exploit it's benefits. In this regard, research into investigating suitable radio protocols for the purpose of facilitating V2X communication. Existing candidates such as ZigBee, Bluetooth, Wi-Fi have proved to be insufficient owing to their lack of reliability and the ability to guarantee deadlines. Therefore, a new radio protocol was deemed necessary. After consideration, two competing technologies were selected - one based on the IEEE 802.11p standard named ITS-G5 (or DSRC in US) and the other one is based on the 3GPP standard and is named C-V2X. Extensive evaluations and demonstrations have been carried out for both the technologies and there is an ongoing debate on which one to use for mass deployment. Currently, it is foreseen that both the current vehicles will be equipped with both the technologies and allow branching out of supported V2X applications. Since these technologies have been explained in adequate detail in the subsequent sections, their description here is omitted

1.3 Motivation, Scope & Contribution of the Thesis

The thesis deals primarily with performance analysis and optimization of C-V2X technology. In this regard, the contribution of this thesis is two-fold and is divided into two parts.

1.3.1 Part 1. Link & System Level Analysis of ITS-G5 and C-V2X

Before deploying any new RAT, it is important to extensively evaluate it by means of analytical and empirical methods. Therefore, the first part of this thesis deals primarily with the link and system level performance of both ITS-G5 and C-V2X. To this direction, the following contributions are made

1. The link level pipelines of both ITS-G5 and C-V2X have been explained in adequate detail. There exists already several link level simulators for ITS-G5 and therefore an open source version in Matlab (from ublox) was used for the analysis.
2. Since no open source implementation exist for C-V2X, a new link level simulation framework with complete control, data and broadcast channels was developed in Python. Additionally, fading channel models with support for different delay profiles were also implemented from scratch.
3. The link level analysis was carried out for both the radio technologies for AWGN as well as different multi-path fading channel models proposed by ITU and the DSRC Tiger team.
3. The Signal to Noise Ratio (SNR) - Block Error Rate (BLER) curves obtained after the link level simulation were used to evaluate the system level performance. To do this, V2X simulator, an open source implementation in Matlab developed by university of Bologna was used [29]. The system level analysis was carried out for different scenarios including highway, manhattan grid as well as real world scenarios such as Kaiserslautern and Merzig. Vehicular traffic was generated using SUMO depicting both the peak and off-peak traffic conditions.

There are many works that separately evaluate the link and system level performance of both ITS-G5 and C-V2X. Most of them use the ITU channel models whereas the DSRC tiger team channel models were only used for ITS-G5. However, to the knowledge of the author, there exist no works that comprehensively compare both the technologies for all the channel models side by side. This thesis bridges that gap by evaluating multiple classes of channel models for the link level comparison.

1.3.2 Part 2. AI assistance to offload and optimize PHY Signal Processing

ML Traditional Wireless Network optimization was done by means of acquired domain knowledge in the form of mathematical models / heuristics and designing solutions / algorithms based on this knowledge. This approach, though perfected over the years, will be insufficient when dealing with future hyper-connected society and anticipated services such as autonomous driving, eHealth, Industrial Communication etc. This can be due to *model deficit*, where no physics-based mathematical models exist for the problem due to insufficient domain knowledge or due to *algorithm deficit*, where even though a sound

mathematical model is available, the existing algorithms optimized on the basis of this model are too complex to be implemented [168]. Therefore, it can be said that the future networks need to have intelligence that is spread throughout the infrastructure. In this regard, a promising solution comes in the form of AI/ML where, instead of relying on domain knowledge, a learning flow is used by means of using sufficiently large examples of desired behavior for the algorithm of interest. Such ML algorithms have been applied to various domains such as computer vision, natural language processing, social network filtering, drug design etc. where they have produced results comparable to and in some cases superior to human experts.

In this regard, the second part of this thesis deals with the modeling and evaluation of different ML models to perform traditional signal processing operations. These models would essentially act as *auto-encoders* for the considered operation. The primary objective is to demonstrate the effectiveness of ML techniques using a subset of signal processing operations as follows

1. Using RNNs for encoding and decoding of turbo codes
2. Using CNNs for channel estimation
3. Reliability prediction based on post-equalited SINR using LSTM
4. Obstacle detection using supervised learning based on UWB radar.

1.4 Organization of the Thesis

This thesis is organized into two parts. The first part deals with the link and system level analysis of both C-V2X and ITS-G5 and is comprised of three chapters. chapter 2 provides motivates the need for V2X and presents the history and evolution of both C-V2X and ITS-G5. chapter 3 explains the PHY layer of C-V2X in detail and also presents the architecture and components of the developed link level simulator. Since, ITS-G5 is matured technology and many papers already explain it in sufficient detail, its PHY layer description in not included here. However, for the sake fo completeness, a thorough description of ITS-G5 PHY is provided in Appendix A. chapter 3 also provides the link level results (in terms of SNR - BLER graphs) of both the technologies for both AWGN as well different fading channel models. chapter 4 continues further with the system level analysis of C-V2X and ITS-G5. Firstly, the MAC schemes, namely SPSS used in C-V2X and CSMA/CA used in ITS-G5 are presented in detail. Next, the different scenarios used for the analysis are explained, namely Highway, Manhattan grid and real world maps. chapter 4 concludes with the results of the simulation that are presented in terms of PER over distance and CBR.

The second part deals with the application of ML for optimizing the PHY blocks. In this regard, chapter 5 introduces ML motivates its use for wireless communication. The applications of ML in the context of wireless communication is also presented by an extensive state of the art analysis along with a survey of existing ML algorithms.chapter 6 demonstrates the applicability of ML as autoencoders for different PHY processes. It does so by using different ANN architectures for the operations of channel coding (turbo codes) and channel estimation. chapter 7 shows the applicability of ML for predicting the reliability of the transmission using post-equalized SINRs. chapter 8 demonstrates the efficacy of ML in detecting obstacles on time domain waveforms. For this, UWB radar was used for data

generation in real world environments. Finally chapter 9 provides the overall summary of the thesis and provides concluding remarks.

Part I

Evaluation of V2X Technologies

2 An overview of available V2X Technologies

Automobiles play an important role in human life in terms of on-demand transportation, mobility, and convenience. thereby allowing people to geographically increase their social and economic interactions. Developed economies depend very much on them since it is a more popular/preferred mode of transport as compared to developing countries. The European Union (EU) is among the world's biggest producers of motor vehicles and this sector represents the largest private investment in research and development. The european's love for car has not yet faded away which is visible from the fact that the vehicle per capita is showing a steady increase as seen from Figure 2.1

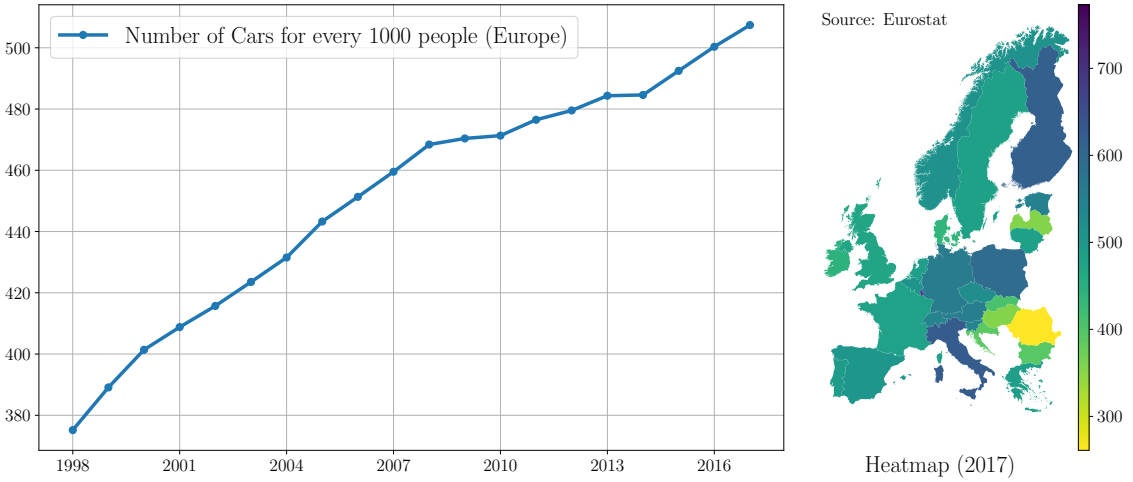


Figure 2.1: Car Ownership Statistics - Europe

With the increasing number of cars, the number of traffic related accidents are also on the rise. Road traffic accidents are the largest cause of injury-related deaths worldwide. Figure 2.2 shows the per capita of number of injuries/deaths due to traffic accidents in Germany. It is important to note that these casualities are decreasing every year due to safer cars with a wide range of acive safety systems as well as government regulations. The country of Sweden has gone to an extreme length of defining Vision2030 where they envision zero traffic accidents.

In order to achieve such strict goals, it is imperative that the vehicles need to sense their environment and be prepared to take corrective action. However, adding too many sensors would only result in an overdesigned system and sometimes, even the best sensor systems fail to anticipate the events due to their limited range and inherent latency involved in sensing operation. V2V communication, a technology that lets cars

A promising solution is to let cars exchange information about their speed, position, steering and brake status within a few hundred meters inorder to avoid accidents, ease conges-

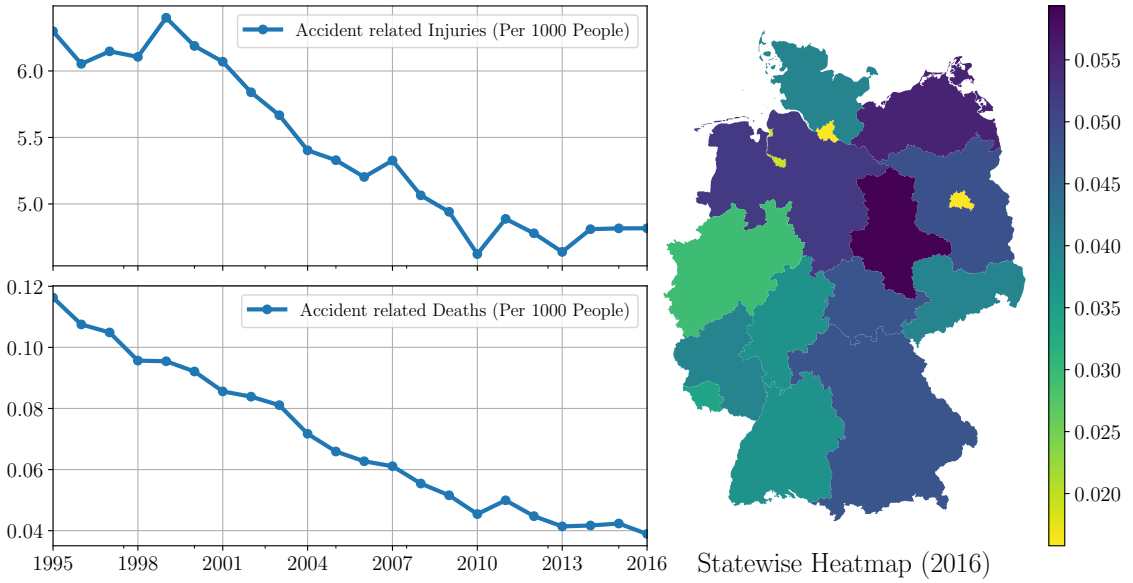


Figure 2.2: Accident Statistics - Germany

tion and improve the enviroment as well. For example, consider a car that is equipped with sensors to detect if the vehicle in front has braked harshly. It takes somewhere between tens of milliseconds to seconds for the vehicle's sensors to detect this event after which corrective measures can be applied. In constrast, a vehicle equipped with a communication device can disseminate this information (almost instanteneously) when it has applied harsh braking thereby saving precious time. Moreover, the communication devices deployed with the road side infrastructure can also help disseminate local information such as traffic jams and other speed regulations.

These forms of communication are termed Vehicular-to-Everything (V2X) and encompasses communication scenarios between vehicles, Road Side Units (RSUs) and other traffic stakeholders such as pedestrians, traffic lights, traffic control centres etc. The main goal of V2X is to increase traffic safety and efficiency. The technology behind V2X communication allows vehicles to broadcast and receive broadcast messages from other traffic entities (up to 10 times per second), creating a 360-degree *awareness* of other traffic entities in proximity. Vehicles equipped with appropriate software (or safety applications) can use the messages from surrounding traffic entities to determine potential crash threats as they develop. Depending upon the type of traffic entity involved, V2X communication modes can be broadly classified into four types. They are

Vehicle-to-Vehicle (V2V) Communication

This mode encompasses the communication between the vehicles within a certain proximity to each other. V2V enables vehicles to broadcast information such as their position, speed, steering-wheel position, brake status etc. to other vehicles within a few hundred meters. The other cars can use such information to build a dynamic local map of the environment around them and to anticipate any potential accidents or other undesirable events. The V2V can be a mesh network enabling every car to send, capture and retransmit messages.

Vehicle-to-Infrastructure (V2I)

This mode encompasses communication between road side infrastructure called RSUs with other traffic entities such as vehicles, pedestrians and cyclists. The RSU can be thought of as a mini-basestation that contains local information about traffic and other road side warnings. The RSUs can be configured to broadcast this local information to all the traffic entities in its proximity and also receive messages from them. It can also filter these messages to build local maps of its own and if necessary, share this data with the traffic control centre and/or any other traffic regulatory via a cellular uplink. Additionally, RSUs can also be used to broadcast local advertisements.

Vehicle-to-Pedestrian (V2P)

This mode of communication encompasses communication between vehicles with pedestrians and cyclists. This is especially useful in cities and rural areas where the number of accidents involving pedestrians and cyclists is on the rise. Vehicle-to-Pedestrian (V2P) encompasses traffic entities such as pedestrians at crosswalks, cyclists, passengers embarking/disembarking buses and people using wheelchairs or other mobility devices issuing warnings to the drivers of vehicles about their presence. This is by far the most challenging mode requiring pedestrians/cyclists to be equipped with hardware that supports sending and receiving traffic related messages which is usually energy intensive.

Vehicle-to-Network (V2N)

This mode encompasses communication between vehicles and a cellular network and is useful for applications such as browsing and access to applications requiring internet connectivity. It can also be used to disseminate traffic related messages (referred to as Decentralized Environment Notification Messages (DENMs)) over a broader geographical area. This mode is similar to traditional cellular communication except that it eliminates the need for smartphone and interfaces directly with the car.

V2X Usecases

Usecases form the paradigm for the use of V2X communication - when, where and for what purpose do we use V2X communication. Since the main motivations for V2X are to increase traffic safety & efficiency and to enhance driving experience (convenience), many usecases were proposed and classified accordingly. These usecases were proposed by projects such as C-ITS Platform, EU FP7 funded project PRE-DRIVE C2X, SimTD, DRIVE etc. as well as standardizing bodies such as ETSI, 3GPP, SAE.

Safety usecases increase the overall road and traffic safety. These usecases are targeted towards avoiding crashes and/or to minimize damage when the crashes become unavoidable. These can be further divided into hard safety and soft safety usecases depending upon the stringency of requirements. Efficiency usecases strive to increase the overall traffic efficiency. These are not safety critical and thereby have relayed reliability and real-time requirements. Convenience/Mobility usecases provide entertainment services during the commute. These usecases focus on making the driving more enjoyable and may also be delivered through consumer electronic devices such as smartphones of using the in-vehicle

embedded devices. These usecases can tolerate huge delays but generally demand very high throughput. An overview of these usecases can be seen in Figure 2.3

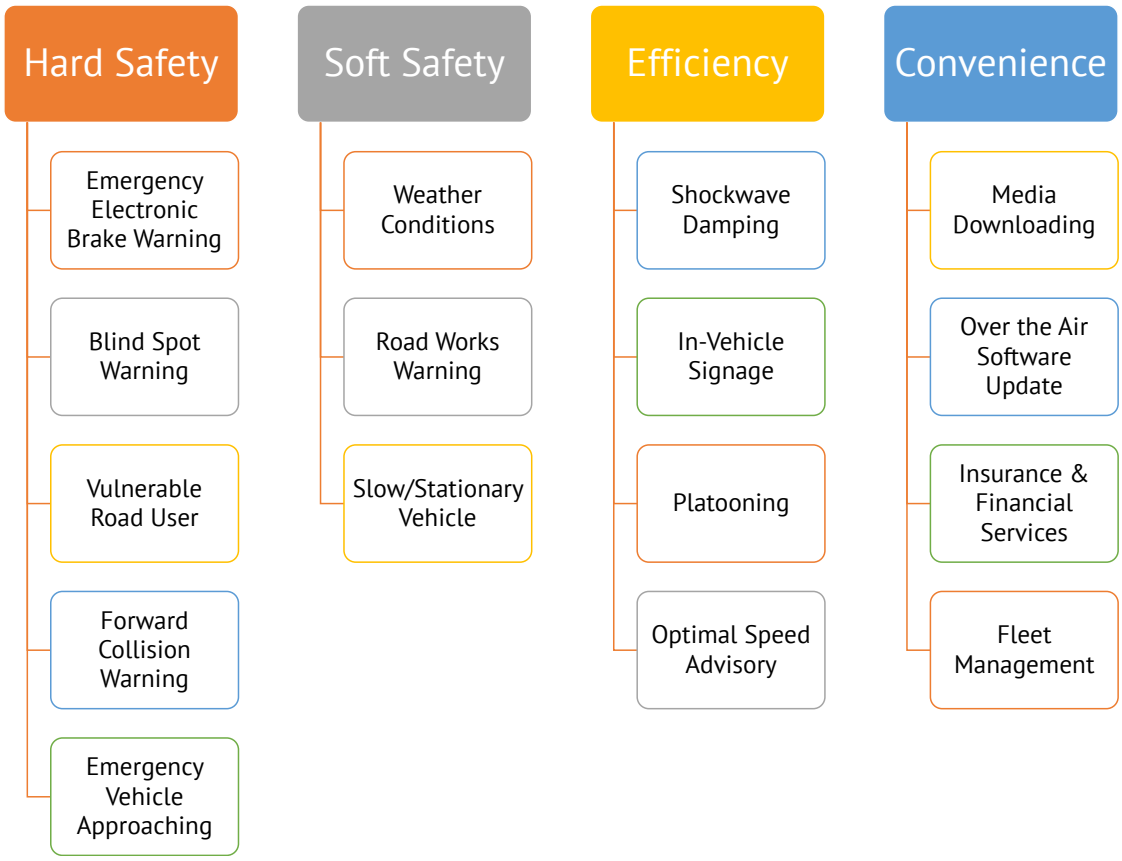


Figure 2.3: Example V2V Usecases

However, with the advent of applications such as autonomous driving, connected driving coupled with enabling technologies such as 5G, Network Slicing, Mobile Edge Cloud etc and the need for green mobility, the differentiating margins between these categories is slowly disappearing. For example, connected driving is more safer, efficient and at the same time very convenient. Considering this into account, the 5G Automotive Association (5GAA) has identified the following groups of usecases with a possibility that a given usecase can fall into more than one group

Safety includes usecases that provide enhanced safety for vehicle and the driver. Examples include Emergency Braking, Intersection Management Assist, Collision warning etc.

Vehicle Operations Management includes usecases that provide operational and management value to the vehicle manufacturer. Examples include sensors monitoring, software updates, remote support etc.

Convenience includes usecases that provide value and convenience to the driver. Examples include infotainment, assisted and cooperative navigation, and autonomous smart parking

Autonomous Driving addresses usecases that are relevant for Autonomous vehicles (SAE level 4 and 5) along with control oriented usecases such as Tele-operation with augmented reality support for remote driver, dynamic map updates etc.

Platooning addresses usecases that are relevant for platooning and platoon management operations such as collect and establishment of platoons, dissolving platoon, passing through platoon etc.

Traffic Efficiency and Environmental friendliness includes usecases that provide enhanced value to infrastructure or city providers. Examples include Green Light Optimal Speed Advisory (GLOSA), smart routing etc.

Society & Community includes usecases that are of value and interest to the society and the public. Examples include Vulnerable Road User (VRU) protection, public and emergency services such as Emergency vehicle approaching, Accident reports etc.

In order for V2X applications and services to run, a Radio Access Network (RAN) is necessary at the backend that implements a RAT. RATs form the basis of any communication stack and their choice heavily influences the performance of wireless communication applications. A RAT can be defined as the underlying physical connection method for a radio based communication network and resides in a RAN and provides the interface between the RAN and the User Equipment (UE). Examples of RAT include Wi-Fi, Bluetooth, LTE, etc. and can be broadly differentiated into two types - cellular networks operating out of a centralized RAN architecture and short-range networks operating in a distributed way.

There are two main contenders for V2X communications. They are DSRC (also known as ITS-G5 in Europe) with a distributed architecture and C-V2X with a centralized architecture.

2.1 DSRC / ITS-G5

DSRC) is an open source protocol that enables two-way short-medium range wireless communications capability for vehicles to exchange critical messages with each other and with RSUs that are positioned along the road. The genisys of DSRC can be traced back to 1999 when the US Federal Communications Commission (FCC) granted 75 MHz of dedicated bandwidth in the 5.9 GHz region since existing technologies, such as the ones used for automatic tolling have too limited bandwidth to support modern ITS applications. In 2002, on the basis of extensive research and testing, the American Society for Testing and Materials (ASTM) published the first standard (ASTM E2213) that recommended that the RAT be based on a modified version of IEEE 802.11a [86]. This led to the formation of an IEEE study group in 2004 that drafted an amendment based on the ASTM recommendation and named it 802.11p. Similar to 802.11a, 802.11p uses Orthogonal Frequency Division Multiplexing (OFDM) at the PHY layer along with re-using the same preamble and pilot design for synchronization and channel estimation. The only difference is that 802.11p operates in *half-clocked* mode halving the 20MHz channel spacing to 10MHz and effectively doubling the timing parameters such as OFDM symbol duration and cyclic prefix. This enables 802.11p to better handle the high mobility scenarios when compared to 802.11a.

The 802.11 Wireless Local Area Network (LAN) and the 1609 DSRC working groups from IEEE developed the 1609 standard family that specified protocols over the top of 802.11p to make it suitable for wireless vehicular applications (Figure 2.5). Called Wireless Access in Vehicular Environments (WAVE), these protocols specify the rules for network association and disassociation. Traditionally, 802.11 defines Basic Service Set (BSS), a station

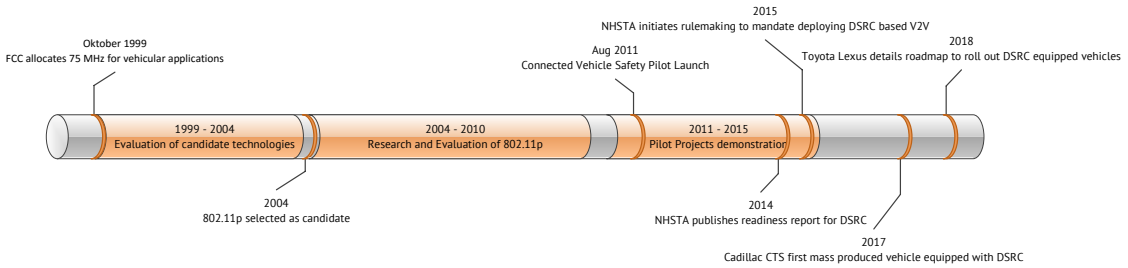


Figure 2.4: DSRC Timeline of events

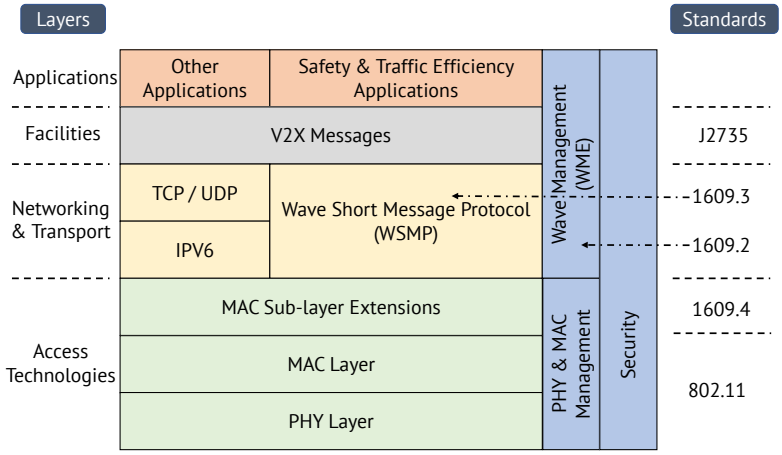


Figure 2.5: DSRC Protocol Stack [59]

grouping mechanism that is used to create different topologies such as master-client or mesh networks. An 802.11 device needs to be a member of a BSS in order to send/receive messages and joining a BSS implies managerial procedures such as scanning, authentication, association etc. For V2X communication, this adds an unnecessary overhead and sometimes, the vehicles may pass through their communication range before a connection gets established. The WAVE protocols defined a new mode of operation called Outside the Context of BSS (OCB) that eliminates the connection establishment procedures along with support for legacy infrastructure mode. The networking protocol used is Internet Protocol version 6 (IPv6) which is the default networking protocol for current networks along with a combination of transport protocols Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). These protocols were also modified to support direct communication modes in V2X. The Wave Short Message Protcol (WSMP) is a single hop network protocol defined in IEEE 1609.3 that acts as a transport protocol multiplexing the messages to upper layer protocol entities based on service IDs. IEEE 1604.4 defines a MAC management extension to enable channel switching between control and shared channels. IEEE1604.2 provides security mechanisms such as authentication and optional encryption of messages based on digital signatures and certificates.

At the facilities layer, the syntax and semantics for the V2X messages are defined by the SAE J2735 standard. A total of 15 message types were specified covering V2V and Vehicle-to-Infrastructure (V2I) communication modes. Most notable of these messages is the Basic Safety Message (BSM) that conveys the vehicle's state information such as position, speed, sensor status etc and is transmitted at the rate of 10 Hz.

2.1.1 ITS-G5

The standardization activities for ITS in Europe started as early as 1990s with the specifications for Traffic Message Channel (TMC) and Electronic Fee Collection (EFC) developed in the context of Real-Time Traffic Information (RTTI) as part of the DRIVE Program [85]. The standardization efforts for V2X in Europe started in parallel to those of USA and supported by a different research and development programs and promoted by different stakeholders [58]. Hence, this led to development of another set of standards namely Cooperative - Intelligent Transportation Systems (C-ITS) with underlying standard derived from 802.11a.

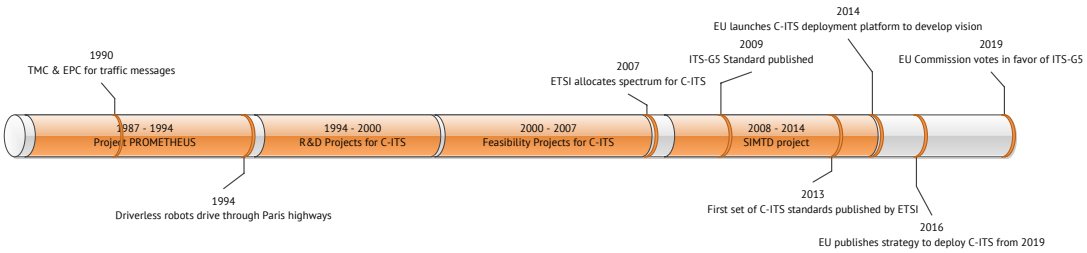


Figure 2.6: C-ITS Timeline

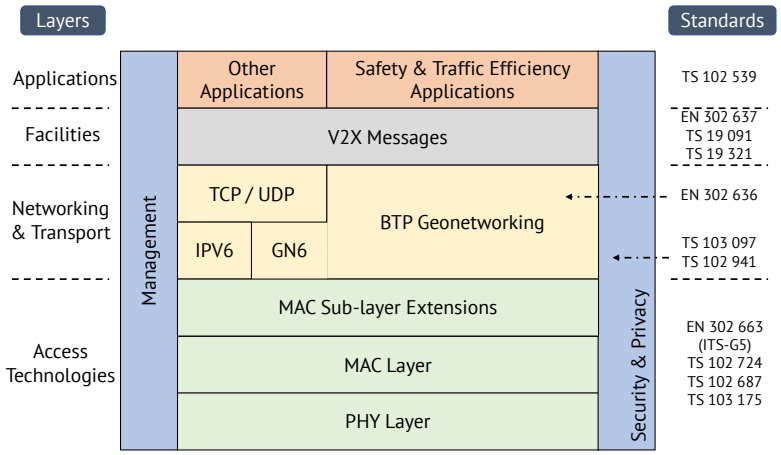


Figure 2.7: C-ITS Protocol Stack [59]

The 802.11p equivalent in the C-ITS stack covering PHY and MAC layers is termed as ITS-G5. Similar to DSRC, it also operates in the 5.9 GHz band which is divided into three subbands 1) ITS-G5A with 30 MHz that is dedicated for safety and traffic efficiency applications 2) ITS-G5B with 20 MHz for non-safety applications and 3) ITS-G5C that is shared with RLAN band. At the PHY layer, C-ITS applies OFDM at the same *half-clocked* mode but with the adapted spectrum masks. Even though the underlying network protocol is based on IPv6, C-ITS specifies an additional multi-hop routing protocol called Geonetworking that uses geographical coordinates for addressing and forwarding messages. Compared to WSMP in DSRC, Geonetworking is optimized for multi-hop communications with geo-addressing, providing enhanced support for applications albeit at an increased protocol complexity and overhead. At the facilities layer, the equivalent message type for BSM is termed Cooperative Awareness message (CAM) [52] and is used for exchanging periodic vehicle status messages. An additional message type called DENM [53] is also

specified that is tailored for Geonetworking layer and is used for disseminating safety information within a geographical area.

These activities were later mirrored by other countries with Europe and Japan allocating similar frequency bands for ITS operations - called the ITS-G5 and the Association of Radio Industries and Businesses (ARIB) standards respectively.

2.2 C-V2X

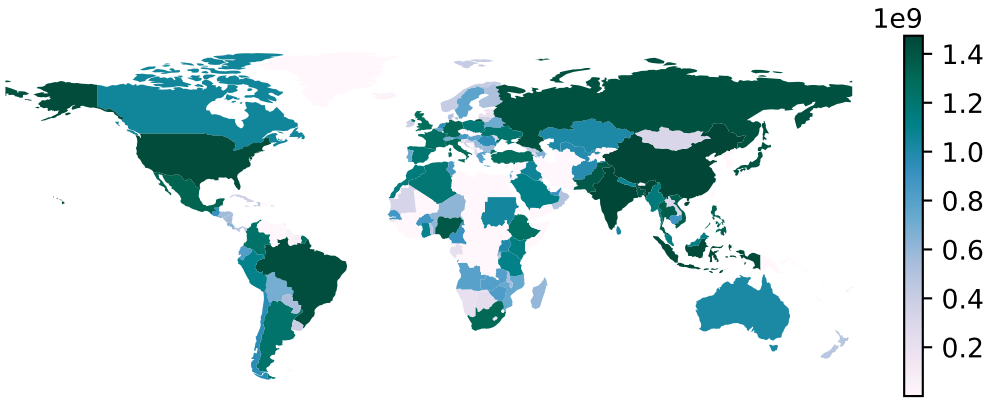


Figure 2.8: Number of Cellular Subscribers 2017

Cellular wireless technologies have enjoyed a dramatic growth since 1983 when the first commercial cellular telephone service was deployed by Ameritech in Chicago area. Cellular technology today has become ubiquitous owing to its continual evolution, thanks to the efforts of standardization bodies, namely 3GPP, Third Generation Partnership Project-2 (3GPP2) and to some extent IEEE. Our focus here is on 3GPP that is responsible for the standardization of Global System for Mobile communications (GSM), Universal Mobile Telecommunications System (UMTS) and LTE technologies that are commonly referred to as 2G, 3G and 4G. This evolution is standardized and published in the form of *Specification Releases*. Each Release(Rel.) is a collection of specifications that provides in detail the standards and guidelines for deploying cellular networks. The first specification for LTE was presented in the Rel.8 document with minor revisions in Rel.9. The currently used standard is LTE-Advanced (LTE-A) that compliments the legacy standard with technologies such as Carrier Aggregation (CA), multi-antenna techniques, Coordinated MultiPoint (CoMP) and support download speeds of upto 300 MB/s and upload speeds of upto 100 MB/s.

A plethora of literature exists describing in sufficient detail, the overall LTE network architecture. Here we consider a broader view of its structure that is in principle, based in its predecessor technologies. It comprises of three subsystems 1) the UE, 2) the Access Network (AN) and, 3) the Core Network (CN) The AN of LTE is called Evolved Universal Terrestrial Radio Access Network) (E-UTRAN) which uses OFDM in the radio interface to communicate with the UE. Evolved Packet Core (EPC) is used in the CN to provide a all-IP architecture to enable access to various services such as telephony and internet. An E-UTRAN AN contains multiple eNodeBs (eNBs) spread geographically that allow

connectivity between the UEs and the EPC CN. An eNB communicates with the other elements of the system by means of 3 interfaces - E-UTRAN Uu, S1 and X2 respectively.

The S1 interface is divided into control plane (S1-CP) and user plane (S1-UP) to allow eNB to connect with different nodes in the EPC. The control plane refers to the protocol stack that is used to manage the interface operations between multiple nodes wheres the user plane manages the user data. The interface E-UTRAN Uu, also known as LTE Uu or simply LTE radio interface, allows data transfer between the eNB and the UEs. All the functions and protocols needed for this transfer are implemented in the eNB. Optionally, the eNBs can connect between them using the X2 interface. These connections can be used to exchange signaling messages to handle the radio resources (e.g. to reduce interference) and also to manage traffic when users move from one eNB to another during a handover procedure.

Data transfer between different interfaces is handled by means of *channels*. LTE uses several different types of logical, transport and physical channels. Logical channels define the *type* of information that is to be transmitted and can be further divided into traffic and control channels. Traffic channels carry data in the user plane, while control channels carry signalling messages in the control plane. Transport channels define the *method* of data transport, i.e., the encoding, scrambling or any other source/channel coding mechanisms. Finally, physical channels define the *mapping* of data blocks onto the time domain symbols and frequency domain sub-carriers. For a complete list of all the channels used in LTE, please refer to Table.

2.2.1 LTE D2D as a precursor to Cellular based V2X

3GPP's Rel.12 included significant changes to the legacy LTE architecture by introducing the network assisted direct D2D communications that enables UEs that are in close proximity to directly establish a communication link between themselves instead of relying on the network infrastructure. The primary driver for D2D support came in the form of public emergency usecases (e.g., Group communication) wherein the mobile UEs need to take over some functionality of the cellular network in the absense of network coverage.Release 12 specifies the concept of Proximity Services (ProSe) that uses direct D2D communication. This communication mode is seen to support services such as ProSe direct discovery and ProSe direct communication to identify and communicate with nearby UEs respectively. Cellular resources in the Uplink (UL) are used for ProSe services mainly because of two reasons: 1) SC-FDMA, which is used in the UL and is much efficient than its DL counterpart, OFDM in terms of Peak to Average Power Ratio (PAPR) and energy, 2) UL transmissions are sporadic compared to DL where the eNB has always something to transmit. Moreover, due to the low transmission power and geographical separation of the UEs, interference is also less in the UL band.

2.2.1.1 Interfaces & Channels

Several new interfaces have been introduced with ProSe apart from the Uu interface that is traditionally used for facilitating communication between the UEs and the eNB. From the UE point of view, the important ones are the PC5 interface, which is a *one-to-many* communication interface to connect with other UEs and a PC3 interface used by the UE to contact the ProSe function to authorize itself.

The channel structures that are used in the air-interface to realize the ProSe application are termed as SL channels in line with UL and DL. There are two logical channels defined for communication - the Sidelink Traffic Channel (STCH) and Sidelink Broadcast Control Channel (SBCCH). STCH is used for user data transmission from the ProSe application and it interfaces with Sidelink Shared Channel (SL-SCH) at the transport layer and the PSSCH at the PHY layer. On other hand, SBCCH is used for synchronization and transmission of control information. It interfaces with Sidelink Broadcast Channel (SL-BCH) at the transport layer, Physical SideLink Broadcast Channel (PSBCH) and PSCCH at PHY layer respectively.

2.2.1.2 SL Modes

Since the SL communication is also expected to work in cases where the backend connection to the eNB is not available, additional communication modes have been defined in addition to the legacy point-to-point links.

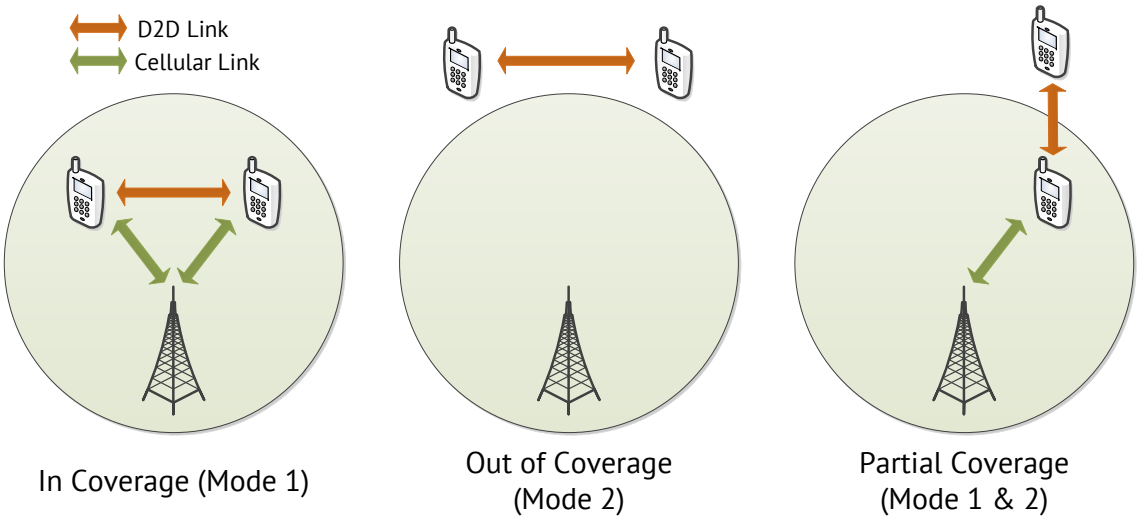


Figure 2.9: SL Communication Modes

Mode 1 Also referred to as the *in-coverage* scenario, the network controls the resources used for ProSe communication. It may assign specific resources to a transmitting UE, or may assign a pool of resources the UE selects from. This way, interferences with the cellular traffic is avoided and in addition the ProSe communication may be optimized.

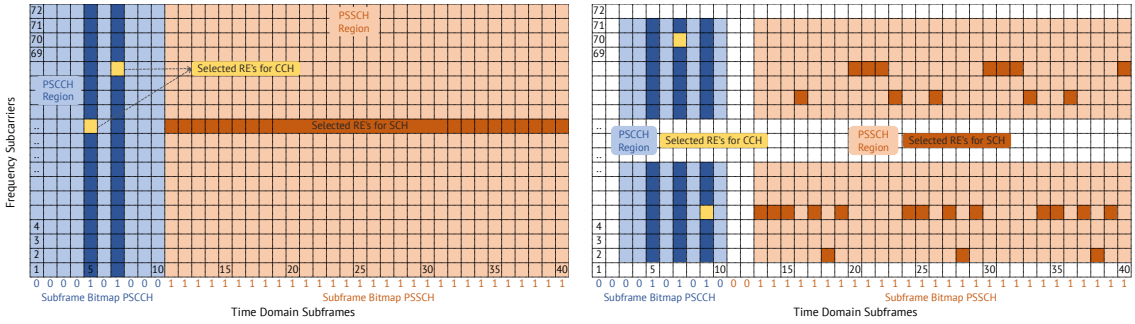
Mode 2 Also referred to as the *out-of-coverage* scenario, such a control is not possible. The UE uses resources which are pre-configured.However, the term out-of-coverage has to be interpreted carefully. It does not mean that there is no coverage at all. It rather means that there is no coverage on the frequency used for ProSe direct communication, although the UE might be in coverage on a different carrier for cellular traffic.

Mode 3 It also referred to as *partial-coverage* scenario. The UE out-of-coverage uses the pre-configured values, whereas the UE in coverage gets its resources from the eNB. A careful coordination between the network and the pre-configured values is necessary in

order to enable communication and to limit the interferences to UEs at the cell boundary near an out-of-coverage UE.

2.2.1.3 The Concept of Resource Pools (RPs)

Since the sidelink transmissions are scheduled to operate side by side with the UL transmissions, new measures for resource allocation and transmission scheduling are required. This is achieved by means of RPs; a set of resources assigned to the SL operation. It consists of the sub-frames and resource blocks within. Separate RPs are defined for transmission (Tx RPs) and reception (Rx RPs). For every Tx RPs, there must be an associated Rx RPs. The physical resources (sub-frames and resource blocks) associated with a given pool are partitioned into a sequence of repeating 'hyperframes' known as PSCCH periods, also referred to as the Scheduling Assignment (SA) period or Sidelink Control (SC) period. Within a PSCCH period there are separate sub-frame pools and resource block pools for control (PSCCH) and data (PSSCH).The PSCCH carries SCI messages, which describe the dynamic transmission properties of the PSSCH that follow it. The receiving UE searches all configured PSCCH resource pools for SCI transmissions of interest to it. A UE can be a member of more than one SL communications group.



(a) Simple Configuration with no hopping **(b)** A more complex Configuration with hopping

Figure 2.10: The concept of RPs

Figure 2.10 illustrates the concept of RPs by means of two arbitrary configurations. The X-axis denotes the time domain sub-frames (in ms) for the given period (here assumed to be 40). The Y-axis denotes the number of sub-carriers which depend on the considered bandwidth (5 MHz). The sub-frames are divided into two separate regions for the control and user data as depicted by the light blue and magenta colors respectively. The actual sub-frames used by a given UE to transmit the control and user data (dark blue) is determined by the sub-frame bitmap. There are separate sub-frame bitmaps for both control and user data where a 1 indicates the current sub-frame is used for SL transmission and 0 indicates otherwise. Within an SL sub-frame, a set of Resource Elements (REs) are selected for the actual control and data transmission as indicated by the yellow and dark magenta colors respectively. 2.10a shows a simple configuration where the the first 10 sub-frames denoting the control region and the subsequent 30 sub-frames denoting the shared region respectively. Both the control and shared regions occupy the entire spectrum of sub-carriers. As indicated by the Control Channel (CCH) bitmap, sub-frames 5 and 7 have been selected for the PSCCH transmission. This is because every PSCCH transmission is done twice on two separate sub-carriers (may also do so within a single sub-frame). In the case of 2.10a, the two PSCCH transmissions are scheduled on two different sub-frames

Table 2.1: RP Configuration Parameters

Data Type	Parameter (3GPP notation)	Description
Control	prb_num	indicates the number of Physical Resource Blocks (PRBs) in a sub-pool
	prb_start	indicates the starting PRB of the SL resources
	prb_end	indicates the end PRB of the SL resources
	offsetIndicator	indicates the offset of PSCCH region in terms of sub-frames from period start
	subframeBitmap	The subframe bitmap for the CCH
Shared	n_{PSSCH}	The number of PRBs available for data transmission
	Time Resource Index (I_{TRP})	subframe bitmap for \gls{ac_sch}
	Resource Indication Value (RIV)	resource index value that corresponds to the current allocated REs
	frequencyHopping & hoppingBits	Frequency hopping flag and the corresponding hopping sequence

(5 and 7) at different sub-carrier frequencies. The Shared Channel (SCH) transmissions occur immediately after the last CCH region within all the remaining sub-frames (Hence the sub-frame bitmap is represented by all 1's).

2.10b shows an example resource allocation for a slightly complex scenario with 2 *sub-pool* regions within the CCH where PSCCH transmissions are allowed. Similarly, SCH region is also divided into 2 contiguous blocks instead of one big block. Additionally, two changes can be observed. 1) the first 2 sub-frames in the SCH region are not allocated for SCH transmission as indicated by the SCH bitmap, 2) Instead of having fixed REs for data transmission, the concept of *frequency resource hopping* is applied in order to *jump* over multiple sub-carriers in the time domain.

The RPs can be configured using a set of parameters as outlined in Table 2.1.If the UEs are in coverage of the eNB, then these parameters are signalled to the UEs by eNBs by means of a Downlink Control Information (DCI) message. In case of out of coverage mode, the transmitting UE selects the REs by itself according to rules aimed at minimizing the collision risk.

2.2.2 Rel.14 C-V2X standard

The LTE D2D standard is proposed keeping in mind the emergency public communications and proximity based advertisements using conventional UEs, i.e., smartphones, whose positions are usually assumed to be semi-static. However, V2X links are highly dynamic with higher channel uncertainties. Secondly, the node density is also comparatively higher especially in urban areas. Hence, to this end 3GPP introduced a few fundamental modifications to the PC5 interface to meet the more stringent latency and reliability requirements associated with the vehicular use cases. They are

i. Using additional DMRS symbols (4 instead of 3) to handle the higher Doppler corresponding to relative speeds of up to 500 km/h and at high frequency (5.9 GHz ITS band)

ii. Using a new resource scheduling assignment of UL resources where the control data and the shared data are transmitted in a single subframe over adjacent PRBs

iii. For out of coverage resource scheduling assignment, a sensing with semi-persistent transmission based mechanism was introduced. Since V2V traffic is mostly periodic in nature, this property is utilized to sense congestion on a resource and estimate

future congestion on that resource. Based on this estimation, resources were booked. This technique optimizes the use of the channel by enhancing resource separation between transmitters that are using overlapping resources.

iv. Use of one blind retransmission for increasing the packet reception probability at high speeds

The C-V2X standard is envisioned to support multiple modes of communication including V2P, Vehicle-to-Network (V2N) in addition to the standard vehicular links (V2V & V2I). The network assistance provides an added advantage of providing cellular backend connectivity to the existing RSUs, thereby increasing the range of vehicular use cases.

Rel.14 introduces two additional modes, namely Mode 3 and Mode 4 that are specifically designed for V2V communication scenarios and differ from the D2D Modes 1 and 2 in terms of resource assignment and scheduling [70]. In Mode 3, the eNB is in charge of assigning PRBs to the vehicles in its coverage as well as scheduling their transmissions in time. In Mode 4, the vehicles autonomously select resources for transmitting using a scheme called SPSS.

In this thesis, we limit our discussion to Rel.14 C-V2X. In this regard, extensive link and system level simulations were conducted for both the technologies and their performance was compared. The link level simulation framework for C-V2X id developed in Python using standard computation libraries and extensive simulation campaigns are conducted for different channel models. For ITS-G5, existing open source implementation in Matlab was used. Since the MAC schemes of both the technologies are well established, existing implementation in Matlab was used for system level analysis with various additions. The next chapters deal with a more detailed explanation of the PHY and MAC layers of both the technologies as well as providing implementation details of the developed simulator and finally conclude with the results.

3 *pycv2x* - The link level simulation framework for C-V2X

Built as an over the top component, C-V2X inherits the PHY layer operations from the underlying LTE standard. To fully detail the PHY layer of LTE would constitute an entire book in itserlf. Therefore, in this chapter, we briefly summarize the PHY layer of LTE and the operations that are specific for C-V2X.

The design of the LTE PHY layer is heavily influenced by the societal requirements of the 20th century such as very high data rates, high mobility, dense connectivity and very low latencies. The LTE standard achieves these goals using a variety of spectral efficiency techniques such as OFDM, multiple antenna (MIMO) in addition to high rate coding (Turbo Coding) and modulation techniques. It supports multiple access in both Frequency Division Duplexing (FDD) and Time Domain Duplexing (TDD) configurations with varying supported bandwidths ranging from 1.4 MHz to 20 MHz.

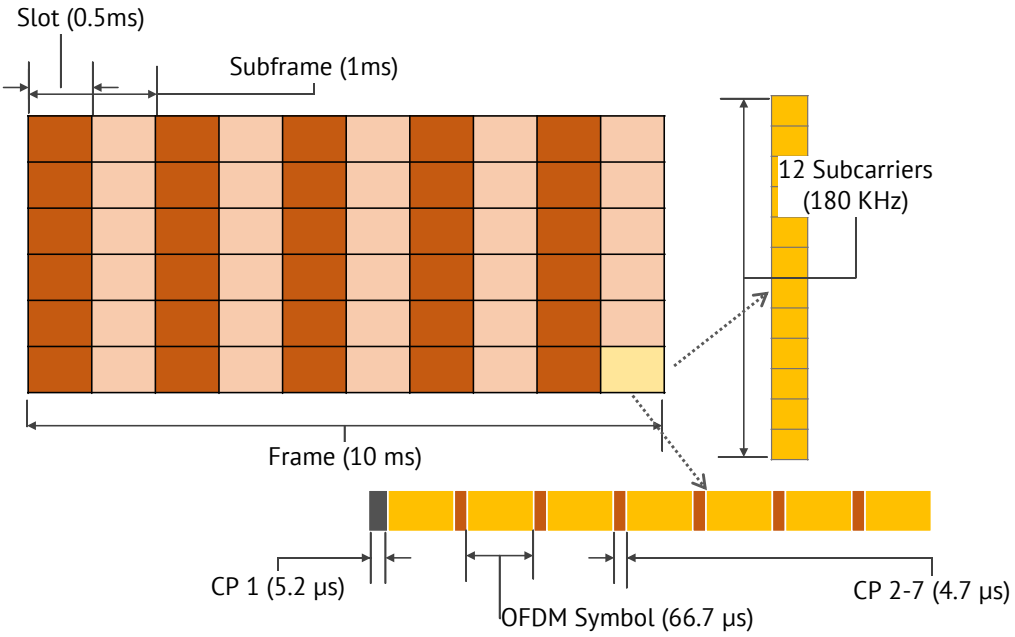


Figure 3.1: LTE Frame Structure for 1.4 MHz bandwidth

The basic time unit T_s in LTE is $1/30720000 \approx 32.6$ ns. In time domain, the transmissions in LTE occur in synchronized blocks called frames. A frame is 10 ms ($30700 * T_s$) in time domain and consists of 10 subframes of length 1 ms ($30720 * T_s$) each. Each subframe in inturn divided into 2 slots of 0.5 ms ($15360 * T_s$) each with each slot consisting of 6-7 OFDM symbols depending on the type of cyclic prefix used. In the frequency domain, the available bandwidth is divided into chunks of blocks called REs / sub-carrier (SC) with each RE spanning over a frequency of 15 kHz. Further 12 such REs are grouped together

Table 3.1: Supported bandwidths and configurations for LTE

Bandwidth (MHz)	1.4	3	5	10	15	20
No. of PRBs	6	15	25	50	75	100
Sampling Frequency (MHz)	1.92	3.84	7.68	15.36	23.04	30.72
Fast Fourier Transform (FFT) size	128	256	512	1024	1536	2048

to form a PRB with each PRB having a bandwidth of 180 kHz. A PRB is the smallest unit of resources that can be allocated to an user. Depending upon the assigned bandwidth, different number of PRBs are available for transmission as shown in Table 3.1.

The useful symbol time, denoted by T_u is $2048 * T_s \approx 66.7\,\mu s$. For the normal Cyclic Prefix (CP), the length of CP for the first symbol is $T_{cp} = 160 * T_s \approx 5.2\,\mu s$. The subsequent CPs has a length $T_{cp} = 144 * T_s \approx 4.7\,\mu s$. The reason for the different length is to make the overall slot length as an integer multiple of T_s. The extended CP is of the length $T_{cp-e} = 512 * T_s \approx 16.7\,\mu s$. The former is used for high data rate scenarios with low mobility whereas the latter is used in special cases like multi-cell broadcast and in very large cells.

For multiplexing in the downlink, LTE employs Orthogonal Frequency Division Multiple Access (OFDMA) which is a parallel transmission scheme, where a high rate serial data stream is split into a set of several low-rate sub-streams, each of which is modulated on a separate SC. Unlike Frequency Division Multiplexing (FDM) where multiple SCs are non overlapping and are separated by a guard band, OFDM systems use closely spaced SCs with overlapping spectra thereby increasing the spectral efficiency. This is achieved by choosing SC frequencies with a constant SC spacing Δ_f, calculated as the reciprocal of symbol time $1/T_s$ to make them orthogonal to each other and eliminate cross talk. Each SC is modulated individually but will be transmitted simultaneously in a superimposed and parallal form. The modulation is accomplished by means of a Inverse Fast Fourier Transform (IFFT) operation that converts the frequency domain complex symbols into a time domain signal. The IFFT operations takes N symbols as input at a time and applies sinusoidal basis functions where the amplitude and phase of any given sinusoid at a particular SC is determined by the value of the complex symbol at that SC. The output from IFFT is the summation of all N sinusoids that make up a single OFDM symbol and can be represented as

$$x(n) = \frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} X(k) e^{j2\pi \frac{kn}{K}} \tag{3.1}$$

Demodulating an OFDM waveform is done by means of an FFT that converts the time domain OFDM symbol into a frequency domain complex valued waveform and is represented as

$$x(k) = \sum_{n=0}^{K-1} x[n] e^{-j2\pi \frac{nk}{K}} \tag{3.2}$$

However, one of the main disadvantage of OFDMA is the large signal PAPR. It is seen that the OFDM symbol is a combination of all SCs. SC voltages can add in-phase at some points within the symbol, resulting in very high instantaneous peak power - much higher than the average power. A high PAPR drives dynamic range requirements for A/D and D/A converters. Even more importantly, it also reduces efficiency of the transmitter RF power amplifier. Due to this relative loss of efficiency, it is not well suited for power constrained

UE terminals. Therefore, SC-FDMA has been chosen as the transmission scheme in uplink since it combines the low PAPR techniques used in single-carrier transmission systems with the multipath resilience and flexible frequency allocation offered by OFDMA.

Along with the data signals, various control signals are also multiplexed together and transmitted in both downlink and uplink. The first of these signals are called synchronization signals (Primary Synchronization Signal (PSS) and Secondary Synchronization Signal (SSS)) that are transmitted in the downlink and used for cell search and identification by the UE in order to synchronize with the eNB. The second set of control signal is called a Reference signal that is transmitted in both uplink and downlink and is used for channel estimation. The uplink Reference signal is also used for synchronization to the UE by the eNB.

Finally, another key aspect of LTE is the concept of multiple antenna techniques which are used to increase coverage and capacity. Adding more antennas to a radio system gives the possibility of performance improvements because the radiated signals will take different physical paths. There are three main types of multiple antenna techniques. The first makes direct use of path diversity in which one radiated path may be subject to fading loss and another may not. The second uses beamsteering by controlling the phase relationships of the electrical signals radiated at the antennas to physically steer transmitted energy. The third type employs spatial separation (the path differences introduced by separating the antennas) through the use of spatial multiplexing or beamforming, also known as MIMO techniques.

3.1 Key aspects of C-V2X PHY

The PHY layer of the C-V2X is same as the LTE uplink and uses SC-FDMA as the access technique.The individual sub-carriers are modulating using one of the two modulation schemes namely - QPSK and 16-QAM. The key difference is the use of 4 DMRS instead of 3 that are used in uplink. This increases the performance of channel estimation and makes it suitable for high mobility scenarios. Secondly, the bandwidth of 10 MHz is the most commonly used configuration and hence we limit our discussion to this bandwidth only. There are a total of 50 PRBs out of which 2 are used for transmitting the SCI. Hence, only 48 out of 50 PRBs are available for coded SL data transmission. The effective coding rate is then calculated as

$$Effective\ Coding\ Rate = \frac{TBS}{N_{PRB} * 12 * Q_m * N_{sym}} \tag{3.3}$$

where TBS is the transport block size which is specified in 3GPP, N_{PRB} is the total number of PRBs available for SL data (48), Q_m is the modulation order (bits/symbol) for the selected MCS and N_{sym} is the total number of usable data symbols. Out of 14 OFDM symbols per subframe, 4 are used for DMRS and the remaining 10 are used for carrying data. However, before SC-FDMA modulation, the last OFDM symbol is set to 0 in accordance with 3GPP specification. Therefore, the total useful symbols per subframe becomes 9. The supported MCS are outlined in Table 3.2.

A second distinguishing factor for C-V2X is the user multiplexing scheme. In contrast to SL D2D, both the control and shared data transmissions of C-V2X occur in the same subframe. Additionally, they can also be transmitted in adjacent or non-adjacent PRBs. Figure 3.2

Table 3.2: C-V2X MCS

MCS Index	Modulation	Transport Block Size	Effective Coding Rate
0	QPSK	1320	0.127
1	QPSK	1736	0.167
2	QPSK	2152	0.207
3	QPSK	2792	0.269
4	QPSK	3496	0.337
5	QPSK	4264	0.411
6	QPSK	4968	0.479
7	QPSK	5992	0.577
8	QPSK	6712	0.647
9	QPSK	7480	0.721
10	QPSK	8504	0.820
11	16QAM	8504	0.410
12	16QAM	9528	0.459
13	16QAM	11064	0.533
14	16QAM	12216	0.589
15	16QAM	13536	0.652
16	16QAM	14688	0.708
17	16QAM	15840	0.763
18	16QAM	17568	0.857
19	16QAM	19080	0.920
20	16QAM	20616	0.994

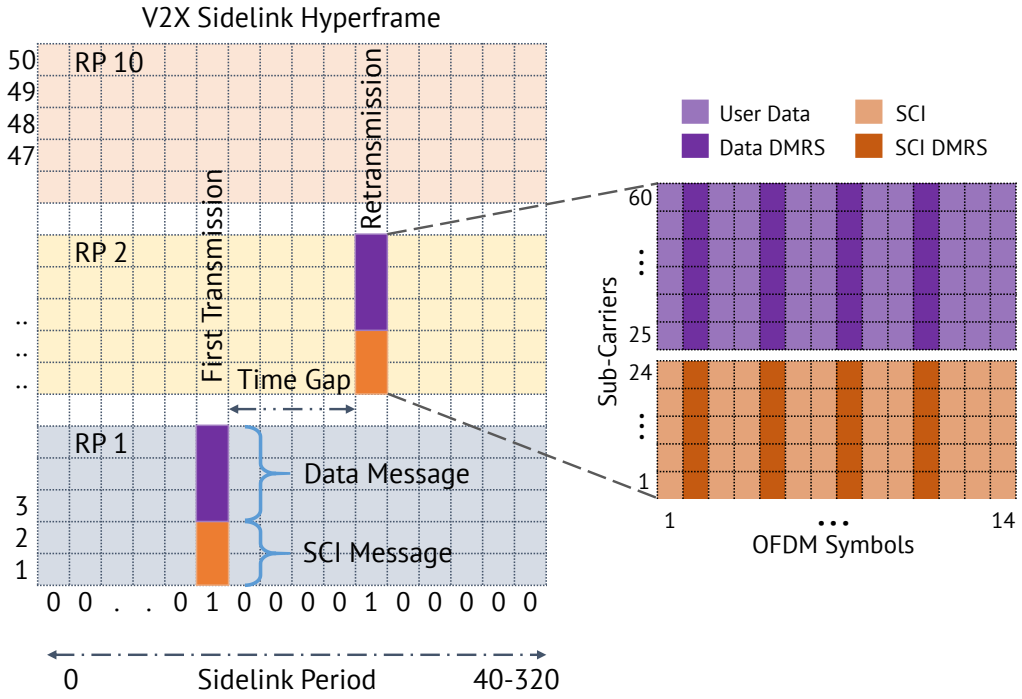


Figure 3.2: C-V2X Hyperframe Structure

illustrates an example sidelink hyperframe for a bandwidth of 10 MHz and an arbitrary PSSCH period (in ms) . Within a PSSCH period, the actual sidelink transmissions can be found on any two subframes (for first transmission and retransmission) given by the

subframe bitmap. For the considered bandwidth of 10 MHz, there are 50 PRBs that are divided into 10 sub-pools (RP1, RP2, ..., RP10) each consisting of 5 contiguous PRBs. A UE can use one or multiple sub-pools for transmission as specified by higher layer messages. For retransmission, the UE can use the same set of sub-pools as the first transmission and use different sub-pools for the subsequent retransmission. In our example, the UE uses RP1 for the first transmission and RP2 for the retransmission.

The SCI message always spans 2 PRBs which is succeeded by the data message. For the given example, a data message spanning over 3 PRBs is assumed. The structure of data and control messages is also illustrated in Figure 3.2. In line with the LTE specification, each PRB consists of 12 SCs in the frequency domain and 14 OFDM symbols in the time domain. Symbols [2, 5, 8, 11] are used for transmitting DMRS that are used for frequency correction and channel estimation. The remaining 10 symbols are used to carry the actual data.

3.2 Baseband Signal Processing

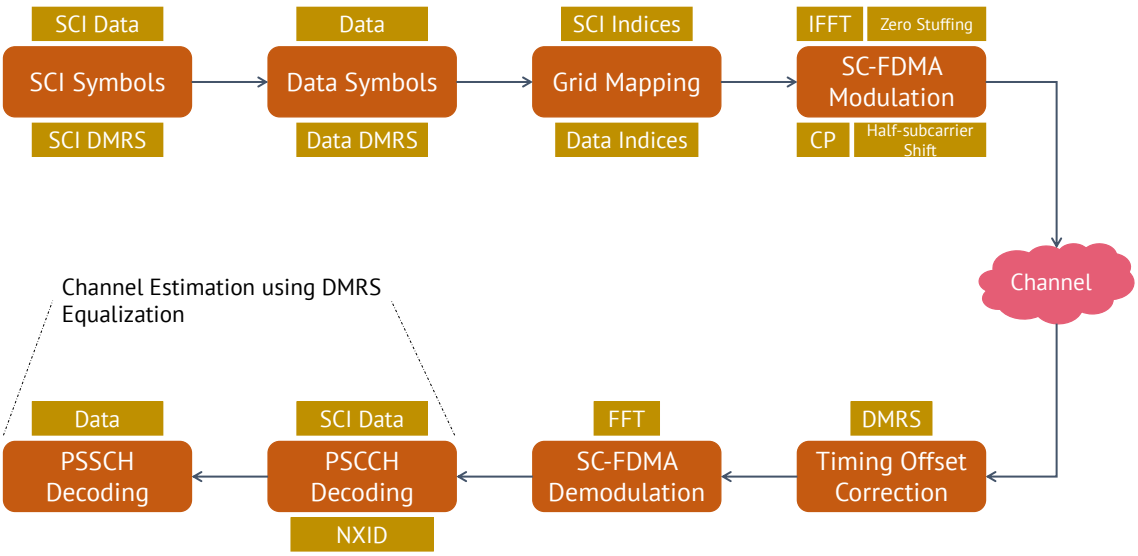


Figure 3.3: C-V2X Tx-Rx Chain

Baseband signal processing encompasses all the operations necessary for encoding and decoding of the binary data bits into/from transmittable waveforms. Figure 3.3 illustrates the operations involved as a flow diagram. The necessary data and control symbols are generated (more details below) and are mapped to the time-frequency grid which is then modulated using SC-FDMA to crrate the time domain waveform which is then passed throuch the channel. The receiver operations consist of timing offset correction to determine the start of the frame and to extract the received time domain waveform. This is then followed by SC-FDMA demodulation to get the retrieve the time-frequency grid. Channel estimation and equalization is performed individually for both the control and data parts starting with the control channel. Finally, the data bits are decoded and recovered.

3.2.1 SCI Operations

The SCI message is generated and transmitted using PSCCH. The uncoded SCI message ia a Format1 32 bit long binary sequence that consists of control information as outlined in Table 3.3. Bits $1-3$ are used for setting the message priority. Bits 4-8 are used to indicate the resource reservation that is used to decide whether to keep the existing SL resource or to select new resources. Both these fields are provided by higher layers. The Frequency resource location of the initial transmission and retransmission is equal to the Resource Indication Value (RIV) corresponding to a starting sub-channel index (n_{subCH}^{start}) and the total number of sub-channels (L_{subCH}) allocated for the data transmission and is calculated as

$$RIV = \begin{cases} N_{subCH}\,(L_{subCH}-1)+n_{subCH}^{start} & (L_{subCH}-1) \leq \lfloor N_{subCH}/2 \rfloor \\ N_{subCH}\,(N_{subCH}-L_{subCH}+1)+\left(N_{subCH}-1-n_{subCH}^{start}\right) & otherwise \end{cases} \tag{3.4}$$

where N_{subCH} is the total number of subchannels available for SL transmission. The number of bits occupied is then calculated as a function of N_{subCH} as follows

$$n_{bits}^{RIV} = \lceil \log_2(N_{subCH}(N_{subCH}+1)/2) \rceil \tag{3.5}$$

Table 3.3: SCI Message Content

Bit Locations	Num Bits	Content
1 - 3	3	Message Priority
4 - 8	4	Resource Reservation
9 - (14 - 22)	Variable	Frequency Resource Location
(15 - 23) - (18 - 26)	4	Time Gap between 1st transmission and retransmission
(19 - 2) - (23 - 31)	5	Modulation and Coding Scheme
(24 - 31)	1	Retransmission Index
(25 - 32) - 32	Variable	Reserved

Based on Equation 3.5, the number of bits for the RIV message can range between 1 to 13 bits. In practice, for a bandwidth 10 MHz, the available 50 PRBs are divided into 10 subchannels with a size 5. Therefore, the number of bits required for RIV in this case is calculated to be 6. The next 4 bits after RIV are used for encoding the time gap between the first transmission instance and the retransmission for the same SL message. This is followed by a 5-bit binary sequence indicating the MCS used. The next bit is used for indicating the type of current SL transmission - 0 for transmission and 1 for retransmission. The remaining bits are reserved for future use and are set to 0.

The binary SCI message is then converted into a code word using a series of operations as outlined in Figure 3.4. The operations can be logically seperated into transport and PHY layers respectively. At first, the binary SCI message is appended with a 16-bit CRC. This 16-bit CRC reminder is also referred to as V2X Scrambling Identity (NXID) and is used as an initial condition to scramble the data code word later. This is followed with a $1/3$ rate convolutional encoding using three octal polynomials $G_0 = 133, G_1 = 171$ and $G_2 = 165$. Because the code is tail-biting, output is three times the length of the input, thereby resulting in 144 bit codeword. The codework is then spread to match the number of available bits for PSCCH resouces through a process called *Rate Matching*. Finally, the resulting codeword is multiplexed and interleaved to get the final output. PSCCH processing including bit scrambling followed by mapping the bits to QPSK symbols. This is finally followed by transform precoding where the symbols are mapped to antenna ports.

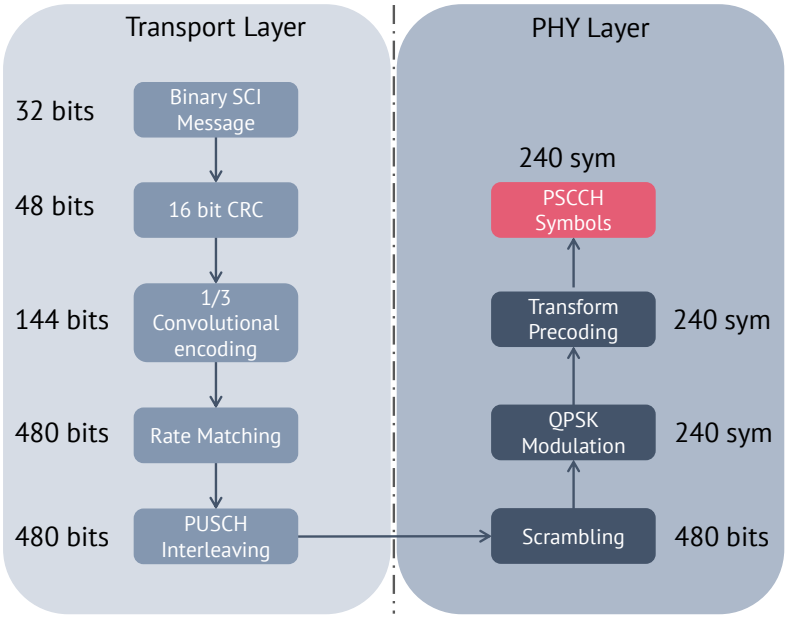


Figure 3.4: SCI Message Operations

3.2.2 Data Operations

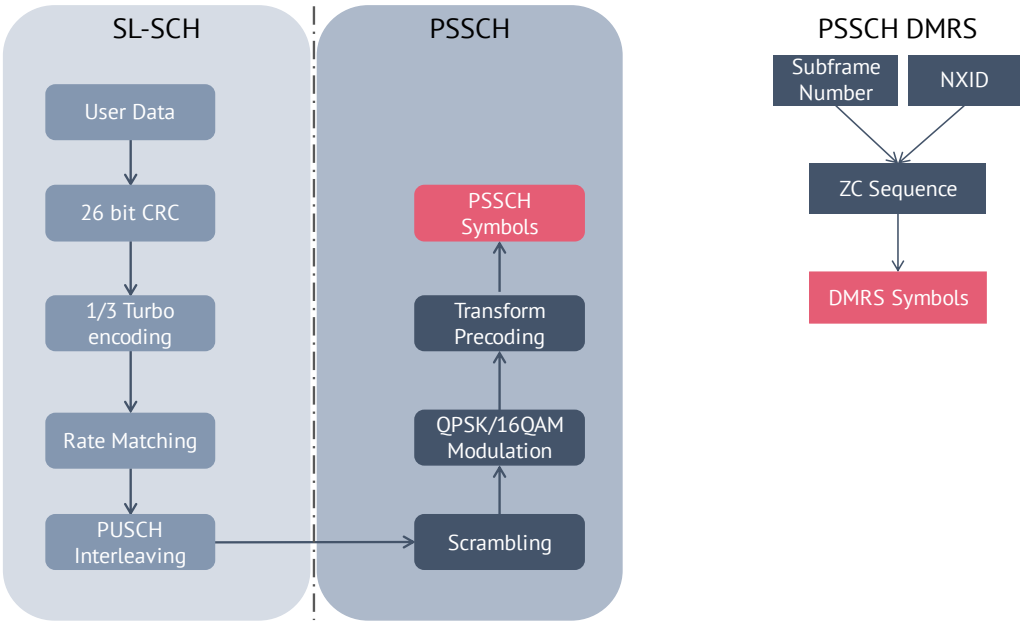


Figure 3.5: Data Operations

The data operations are similar to the SCI operations but differ with respect to generation of DMRS. They are outlined in the Figure 3.5. The binary data message is appended with a 24-bit CRC in contrast to the 16-bit CRC used for SCI. This is followed by a turbo encoding operation with two rate 1/3 Recursive Systematic Convolutional (RSC) encoders that are seperated by an internal interleaver. The LTE standard predefined list of values for the interleaver and hence only a finite number of message sizes are supported by the LTE variant of turbo encoder. The rate matching for turbo coded shaed data channel

consists of interleaving the three bit streams, followed by the collection of bits and the generation of a circular buffer as defined in TS36.212. After multiplexing and interleaving the rate matched output, the SL-SCH codeword is passed on to PHY layer for PSSCH processing.

The PSSCH processing starts with bit level scrambling using a gold sequence generator initialized using the integer value if NXID generated during processing the SCI message. The receiver must decode the SCI message first in order to retrieve the NXID which is then used to descramble and decode the PSSCH. This effectively means that the receiver can decode the SL transmission if and only if it decodes the SCI message. This is followed with modulating the scrambled bit sequence using one of QPSK or 16QAM mapping. Finally, the modulated symbols are transform precoded to map them to their respective antenna ports to obtain the final PSSCH symbols.

3.2.3 DMRS for PSCCH and PSSCH

DMRS are used to perform synchronization and channel estimation at the receiver end and subsequently support data demodulation. It is transmitted alongside the PSCCH and PSSCH symbols (at symbol locations 2, 5, 8 ,11) as shown in Figure 3.2. The DMRS symbols are sequences $r_{u,v}^{(\alpha)}$ that are obtained by a cyclic shift of a base sequence $r_{u,v}(n)$ according to

$$r_{u,v}^{(\alpha)} = e^{j\alpha n} \cdot \bar{r}_{u,v}(n), 0 \leq n \leq M_{sc}^{RS} \tag{3.6}$$

where $M_{sc}^{RS} = mN_{sc}^{RB}$ is the length of the DMRS sequence, m is the number of PRBs and N_{sc}^{RB} is the number of SCs within one PRBs (12 in case of LTE). The base sequence for PSCCH is the same for all SCI transmissions and is given as

$$\begin{aligned} bs_{pscch} = \{ &-3, +1, +3, -3, +1, -1, -3, +3, -3, +3, -1, -1, -1, -1, +1, -3, -3, -3, +1, \\ &-3, -3, -3, +1, -3\} \end{aligned} \tag{3.7}$$

The sequence is QPSK modulated to get the PSCCH DMRS for one symbol. The same sequence is repeated across all the 4 OFDM symbols in one SL subframe. The specification also allows for cyclic shift of the symbols with values selected randomly from [0,3,3,9] in order to multiplex different UEs. Figure 3.6 shows the PSCCH DMRS sequence for different cyclic shifts.

For the case of PSSCH, the DMRS uses a different base sequence which is defined as the cyclic extension of the Zadoff-Chu (ZC) Sequence and is given as

$$\bar{r}_{u,v}(n) = x_q\left(n \mod N_{ZC}^{RS}\right), 0 \leq n \leq M_{sc}^{RS} \tag{3.8}$$

with x_q given by

$$x_q(m) = e^{-j\frac{\pi q m(m+1)}{N_{ZC}^{RS}}}, 0 \leq m \leq N_{ZC}^{RS} - 1 \tag{3.9}$$

where $x_q(m)$ is the q_{th} root of ZC sequence and N_{ZC}^{RS} is the length of ZC sequence that is given by the largest prime number such that $N_{ZC}^{RS} < M_{sc}^{RS} < 3N_{sc}^{RB}$, the base sequence is defined as the computer generated Constant Amplitude Zero AutoCorrelation (CAZAC) sequence.

$$\bar{r}_{u,v}(n) = e^{j\varphi(n)\pi/4}, 0 \leq n \leq M_{sc}^{RS} \tag{3.10}$$

The transmitting node can select a base sequence from a set of groups each differentiated with a hopping sequence that depends on the current subframe number and NXID as

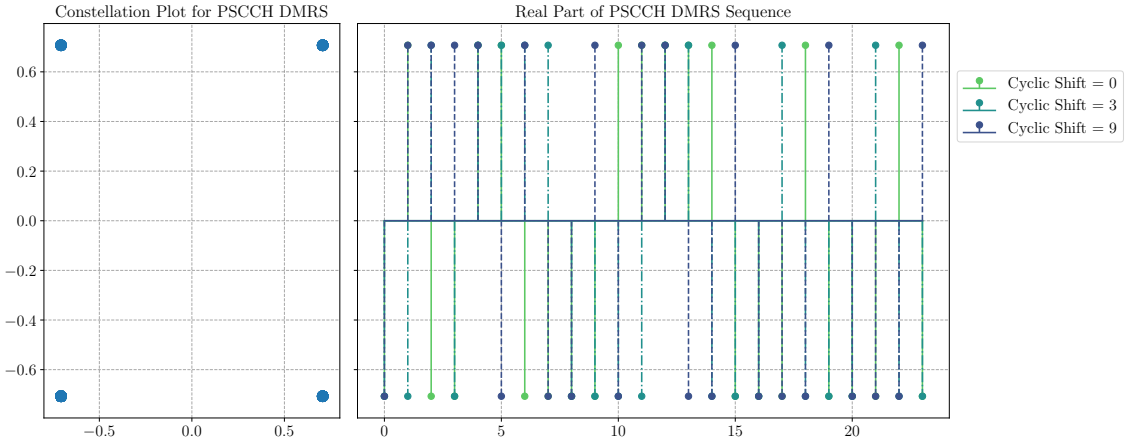


Figure 3.6: PSCCH DMRS sequence

shown in Figure 3.5 and illustrated in Figure 3.7. In this way, the DMRS sequences are randomized for different vehicles thereby reducing inter-cell interference.

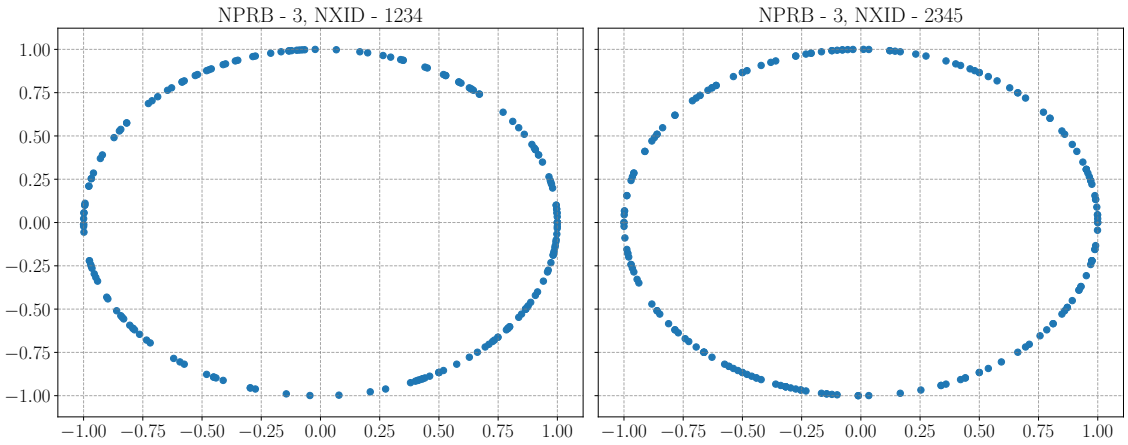


Figure 3.7: PSSCH DMRS Constellation Plot

3.2.4 SC-FDMA Operations

As explained in the introduction, C-V2X employs SC-FDMA instead of OFDMA due to low PAPR. In fact, both the techniques employ a similar operation chain as shown in Figure 3.8 with Serial to parallal (S/P), Parallal to serial (P/S), CP insertion and removal, SC mapping / demapping. Discrete Fourier Transformation (DFT) / Inverse Discrete Fourier Transformation (IDFT). However, SC-FDMA differs with an additional DFT block is used before the subcarrier mapping. This is the reason why SC-FDMA is a single carrier system, unlike OFDM where it employs DFT spread orthogonal frequency multiplexing and frequency domain equalization.

For U users and P SCs, the SC allocation for each user is $M = \frac{P}{U}$. For the u_{th} user, the input bit stream is converted into constellation symbols and grouped into blocks of data

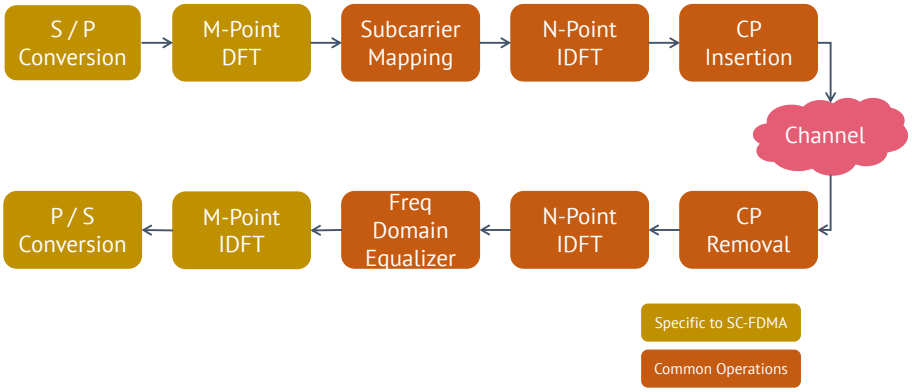


Figure 3.8: SC-FDMA Tx-Rx operations. Blocks in dark color highlight common operations for OFDMA and SC-FDMA

$s(u)$ with size M after which it is transformed with a M-point DFT as follows

$$s_{(u)}(k) = \sum_{m=0}^{M-1} s_{(u)}(m) e^{-j2mok/M} \; k = 0, 1, ..., M-1 \tag{3.11}$$

The DMRS symbols are generated as explained in subsection 3.2.3 and inserted into their respective locations. Subsequently, $s_{(u)}$ is mapped to P subcarriers as

$$X_{(u)} = D_{(u)} \cdot S_{(u)} \; u = 1, 2, ..., U \tag{3.12}$$

Finally, after the SC mapping, the time domain waveform is obtained by a P -point IDFT at sample time m as

$$x_{(u)(m)} = \frac{1}{P} \sum_{k=0}^{P-1} X_{(u)}(k) e^{j2\pi nk/p} \tag{3.13}$$

The time domain waveform is now passed through a channel and noise is added.

3.2.5 Packet Detection, Timing & Frequency Offset Correction

Due to the delay induced by the channel, it becomes imperative that the start of the packet be detected. Hence, packet detection is performed using DMRSin order to find an approximate estimate of the start of the an incoming packet. This is achieved by means of an delay and correlate algorithm proposed by Schmidl and Cox [160] that defines two consecutive sliding windows $P(n)$ and $R(n)$ to calculate the received energy and form a decision variable $M(n)$ as a ratio of total energy contained inside the two windows. The correlation windows at time n are defined as

$$P(n) = \sum_{m=0}^{L-1} (r^*(n+m) \cdot r(n+m+L)) \quad \text{and} \tag{3.14}$$

$$R(n) = \sum_{m=0}^{L-1} |r(n+m+L)|^2 \tag{3.15}$$

where r_n are the received symbols and L is the DMRS interval. The decision variable $M(n)$ is given as

$$M(n) = \frac{|P(n)|^2}{(R(n))^2} \tag{3.16}$$

When the received signal consists of only noise, the output $P(n)$ of the delayed cross correlation is a zero mean random variable. Once the start of the packet is detected, the value of M_n jumps quickly to its maximum value which in turn gives a good estimate of the location of the packet edge.

Besides timing delays, the receiver carrier frequency is also shifted due to inherent non-idealities in the radio front ends of both the transmitter and the receiver. This causes linear phase shifts in the received signal given as $\epsilon = \frac{f_e}{\Delta f}$ with Δf being the SC frequency spacing. It is shown in [160] that the correlation presented in Equation A.11 can also be used to estimate the Coarse Frequency Offset (CFO) as follows

$$\hat{\epsilon} = \frac{1}{2\pi}\frac{N}{L}\angle P(n) \tag{3.17}$$

The estimated $\hat{\epsilon}$ is further refined using DMRS. Due to the presence of pilot symbols whose phase is known, this results in a more fine tuned frequency offset estimate $\hat{\epsilon}_f$. Finally, the received samples are multiplied by $\exp(-j2\pi\hat{\epsilon}_f n/N)$ to remove the frequency offset [115].

3.2.6 Channel Estimation & Equalization

A plethora of algorithms exists for estimating the channel in OFDM systems with varying levels of complexity [77, 220]. However, it can be said that Least Squares (LS) estimator forms the basis for all the other algorithms for pilot assisted channel estimation. The LS estimate of channel coefficient on k^{th} SC $\hat{H}_k(n)$, given the received sample $y_k(n)$ and the transmitted sample $x_k(n)$ is given as

$$\hat{H}_k(n) = \frac{y_k(n)}{x_k(n)} = H_k(n) + w_k(n) \tag{3.18}$$

where $H_k(n)$ denotes the true channel coefficients and $w_k(n)$ denotes the noise. Hence, it can be seen that LS estimator allows us to calculate channel coefficients with a simple division operation without relying on any knowledge about statistics of the channel. This also leads to its inherent problem of high mean-squared error. In case of C-V2X, after extracting the SCI symbol locations, channel estimation is performed where the receiver extracts the DMRS symbols from their known location in the PSCCH time-frequency grid and divides them with their expected value

$$\tilde{H}_{(i,k)} = \frac{Y_{(i,k)}}{X_{(i,k)}} = H_{(i,k)} + N_{(i,k)} \tag{3.19}$$

where $\tilde{H}_{(i,k)}$ is the LS channel estimate at pilot location (i,k), $Y_{(i,k)}$ and $X_{(i,k)}$ are the received and sent pilot symbols at (i,k) and $N_{(i,k)}$ is the noise at (i,k). It can be seen that the calculated LS estimate is noisy and hence in order to minimize the effect of noise, a 2D averaging is performed with a chosen window size. Hence, averaging the instantaneous channel estimates over the window, we have

$$\tilde{H}^{AVG}(i,k) = \frac{1}{|S|}\sum_{m\in S}\tilde{H}_{(i,k)}(m) \approx H_{(i,k)} \tag{3.20}$$

where S is the set of pilots in the 2D window and $|S|$ is the number of pilots in S. The LS estimates and the averaged estimates contain the same data, apart from additive noise. Simply taking the difference between the two estimates results in a noise level value for the LS channel estimates at pilot symbol locations. This knowledge of noise can be useful to increase the performance of some receivers especially using soft demodulation techniques.

Finally, the averaged LS estimates are interpolated across the whole time-frequency grid to get the complete channel matrix $H(t)$ for the received subframe. Equalization is performed by multiplying the received grid $Y(t)$ with the complex conjugate of $H(t)$

$$Y^{eq}(t) = Y(t) * H(t)^* \tag{3.21}$$

The equalized symbols are decoded and if an SCI message is present, the NXID which is the integer value of the 16-bit CRC along with the RIV is recovered. Using these values the receiver then proceeds with decoding the data. if no SCI is found, then the same process is repeated for next cyclic shift and subsequently to the next subchannel.

With the help of RIV value, the receiver extracts the PSSCH symbols from the received grid. The same channel estimation, equalization operations used for SCI decoding are repeated and finally the data is recovered. Recall that the SCI message also has a information indicating the retransmission index and the time gap which is used by the receiver to know if the current SL transmission is the first instance or the second. If the current transmission instance is the first, the receiver stores the decoded soft bits in a buffer to be soft combined with the retransmission that occurs after the time gap. However, the SCI still needs to be blind decoded in this case since the retransmission may use a different set of resources for PSSCH.

3.3 pycv2x Simulator

pycv2x is a python package for simulating and evaluating the link level performance of 3GPP Release 14 C-V2X. It is structurally similar to the LTE System Toolbox from Matlab. However, it is not as extensive as the LTE System Toolbox in the sense that pycv2x only consideres LTE-Sidelink Modes 3 and 4 and does not support Uplink, Downlink and Sidelink Modes 1 and 2. Its main features are

1. Complete Transport and Physical Channel processing for Control, User data and Broadcast
2. Support for Blind Re-transmissions of User data
3. V2X resource pool creation and configuration
4. Multi-path Fading channel models with predefined 3GPP delay profiles (Single Input Single Output (SISO) and Single Input Multiple Output (SIMO))
5. The provided scripts support both single and multi-processing
6. 3GPP Reference Tests as defined in [186]

3.3.1 Organization and Structure

As shown in Figure 3.9, the simulation framework is organized into 8 sub-packages that logically seperate the overall functionality in terms of their location in the Open Systems Interconnect (OSI) stack. In this section we briefly summarize the purpose of each sub-package

Channel Models contains all the functions necessary to realize and apply fading channels (with predefined delay profiles as per 3GPP) and to add AWGN

Channel Processing contains the complete channel operations for Control (SCI), data(SL-SCH) and broadcast(SL-BCH). This includes creating binary messages for broadcast, and control information as well as parsing the received binary messages to extract information.

Config Files contains all the configuration files like Hybrid Automatic Repeat Request (HARQ) and other higher layer simulation parameters

MAC contains all the functions for MAC layer operations such as channel coding, code block segmentation / concatenation, CRC encoding and decoding etc.

Mapping tables contains the look up tables for parameters such as MCS, block lengths etc. from 3GPP

PHY contains all the functions required to process PHY layer operations such as modulation, SC-FDMA, channel estimation and equalization, frame offset calculation, creating PSS and SSS etc.

Utils contain some utility functions to indirectly assist in simulations

Scripts contain the python scripts that build the complete pipeline for link level simulation with and without the use of retransmissions.

In order to check the correctness of implementation, 3GPP has specified some reference tests that give an average value of SNR in order to achieve a BLER target. [186]. The tests cover all the three channels - control, data and broadcast and are outlined in Table 3.4. The results of the simulations for the specified reference channels are illustrated in Figure 3.10. It can be seen that the developed simulator performs on par with the minimum performance requirements of all the reference channels with the exception of PSSCH test 2 where there is a slight offset of 3 dB. This is due to the high Doppler (2700 Hz) associated with that reference channel.

Table 3.4: Performance Tests from 3GPP

Test Name	Channel	Referece Configuration	Bandwidth (MHz)	Propagation Condition	BLER Target (%)	SNR Target (dB)
Minimum Performance	PSCCH	CC.8	10 MHz	EVA1500	1	2.6
Minimum Performance	PSSCH	CD.8	10 MHz	EVA2700	10	5.4
Minimum Performance	PSSCH	CD.9	20 MHz	EVA180	10	14.2
Minimum Performance	PSBCH	CP.2	20 MHz	EVA180	1	2.5
Soft Buffer Test	PSSCH	CD.11	20 MHz	AWGN	5	8.0

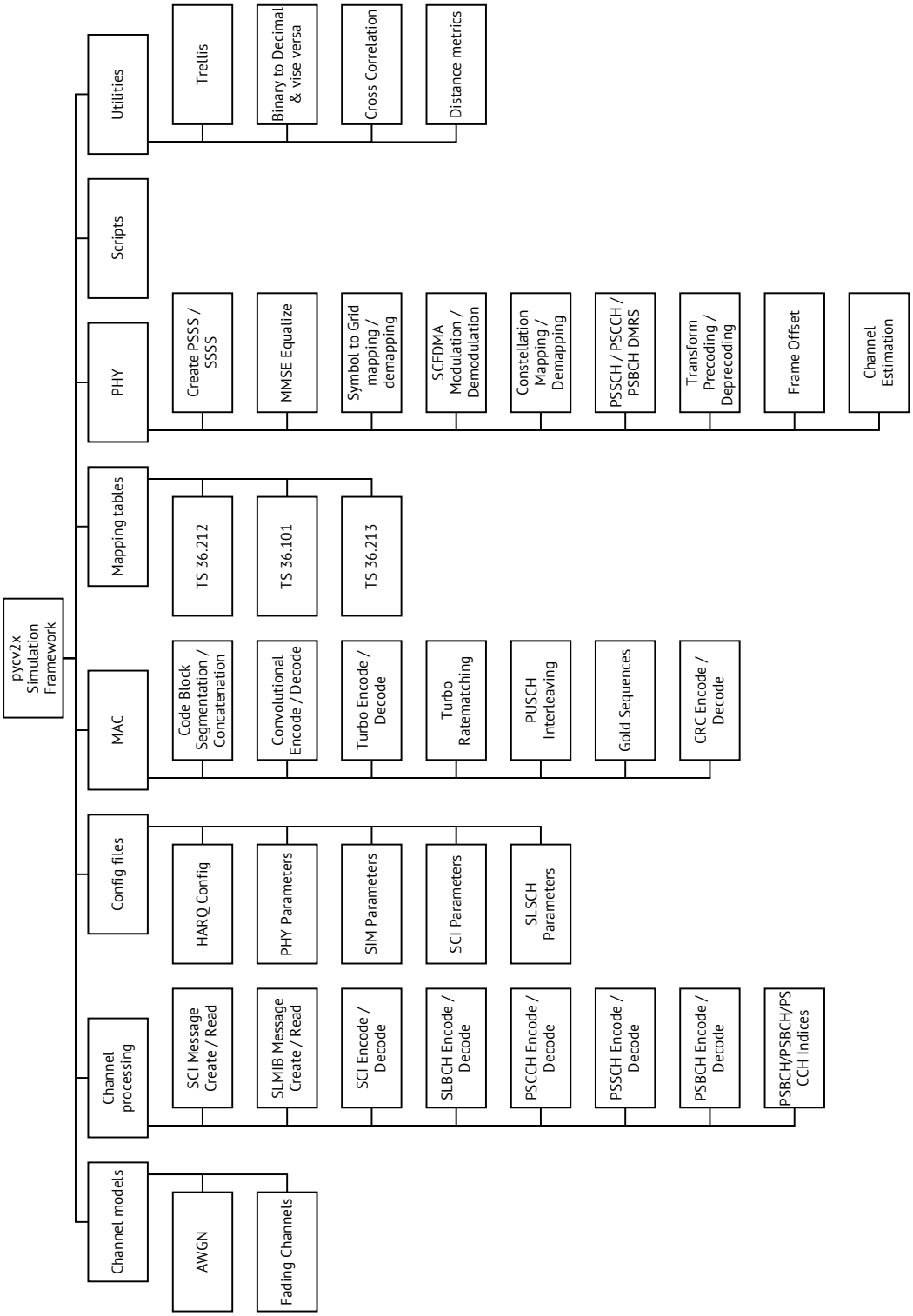


Figure 3.9: The structure of pycv2x simulation framework

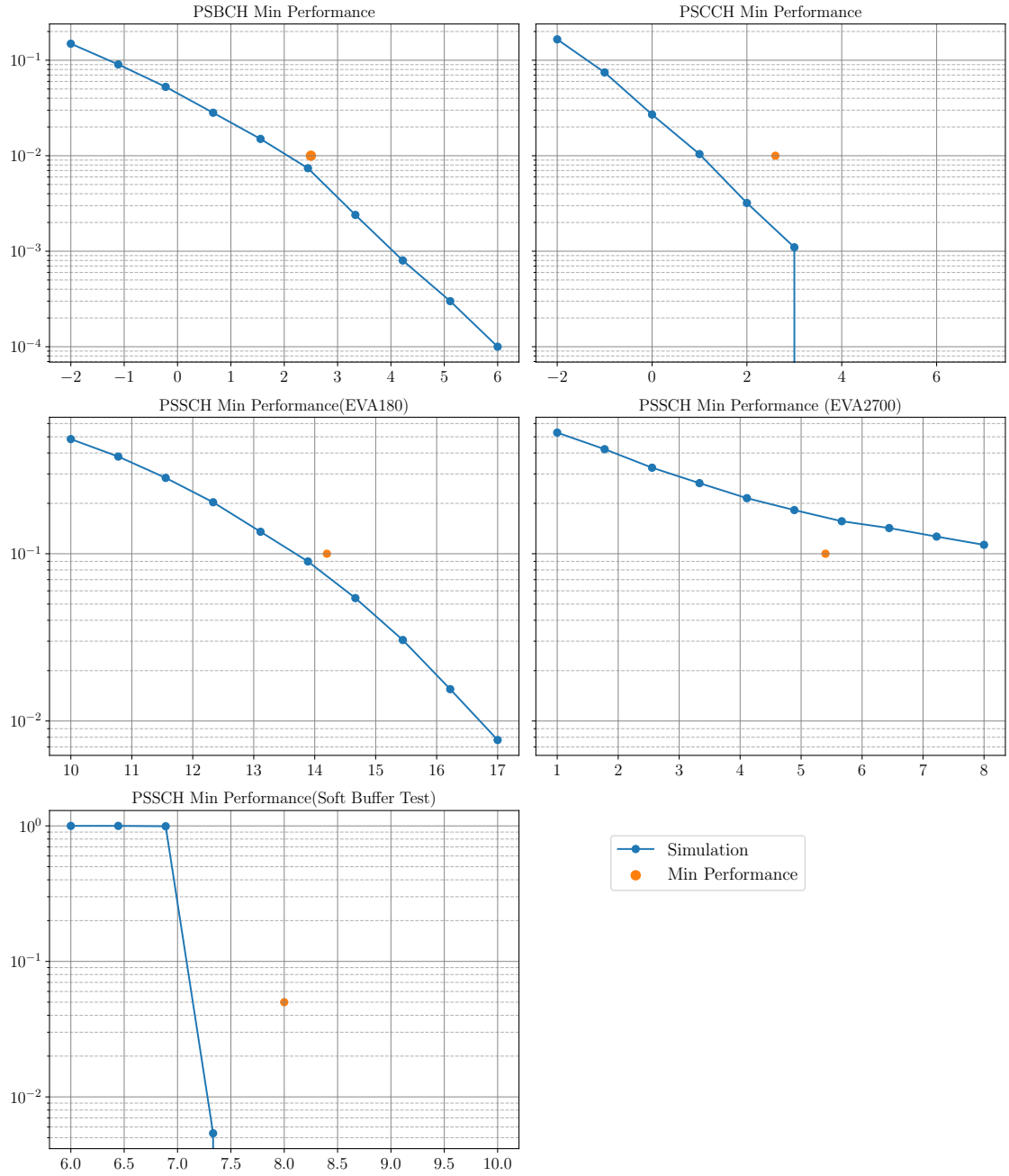


Figure 3.10: C-V2X performance over specified reference channels

3.4 Link Level Performance

Link level simulations model a single link by considering the whole transmit / receive chain of the RAT at the PHY layer. These are useful to understand the relationship between SNR and the PERs. They work by constructing the Transmit-Receive operations in complete detail (as per specifications) and by using abstraction to model the channel behavior. Most commonly used channel model is the AWGN channel model that mimics the effects of many random processes that occur in nature whose summation will tend to have a Gaussian or Normal distribution. This channel model is too simplistic in the sense that it assumes that the channel is linear and time invariant, and sometimes also frequency non-selective. However, it provides a basis to understand the best case performance of the considered RAT as well as get an idea about the achievable system capacity.

In reality, the signal propagates from the transmitter to the receiver via several propagation paths which add up at the receiver. As a result, the received signal experiences fading, variation of signal power with time, and signal echoes with different delays. The contributions of the various paths, i.e., amplitudes and phases, and their respective delays define the impulse response [119]. In order to design robust wireless communication technologies, it is crucial to understand and characterize the wireless channel [50, 40, 74]. In contrast to cellular scenarios, the channel modeling approach for V2V communication must account for the dynamic mobile environments, including highway, suburban and rural scenarios. Moreover, other factors such as vehicle speed, multiple scattering, rapid obstruction of multipath components, *keyhole* and *pinhole* effects needs to be considered as well [118, 122, 123].

In order to have a benchmark for performance comparison, we used ITS-G5 as the candidate since it the only other mature V2X technology that has well documented link level results. The PHY operation chain for ITS-G5 is outlined in the appendix.

3.4.1 Performance comparison over AWGN Channel

Table 3.5: 802.11p - MCS & Data rates

Modulation	Coding Rate	Coded bits per OFDM symbol	Data bits per OFDM symbol	Data Rate (Mbit/s)
Binary Phase Shift Keying (BPSK)	1/2	48	24	3
BPSK	3/4	48	36	4.5
QPSK	1/2	96	48	6
QPSK	3/4	96	72	9
16-QAM	1/2	192	96	12
16-QAM	3/4	192	144	18
64-QAM	2/3	288	192	24
64-QAM	3/4	288	216	27

The continuous-time AWGN is a random channel whose output is a real random process

$$Y(t) = X(t) + N(t)$$

where $X(t)$ and $Y(t)$ are the real input and output waveforms and $N(t)$ is the noise, independent of time with a zero mean and a standard deviation of σ^2.

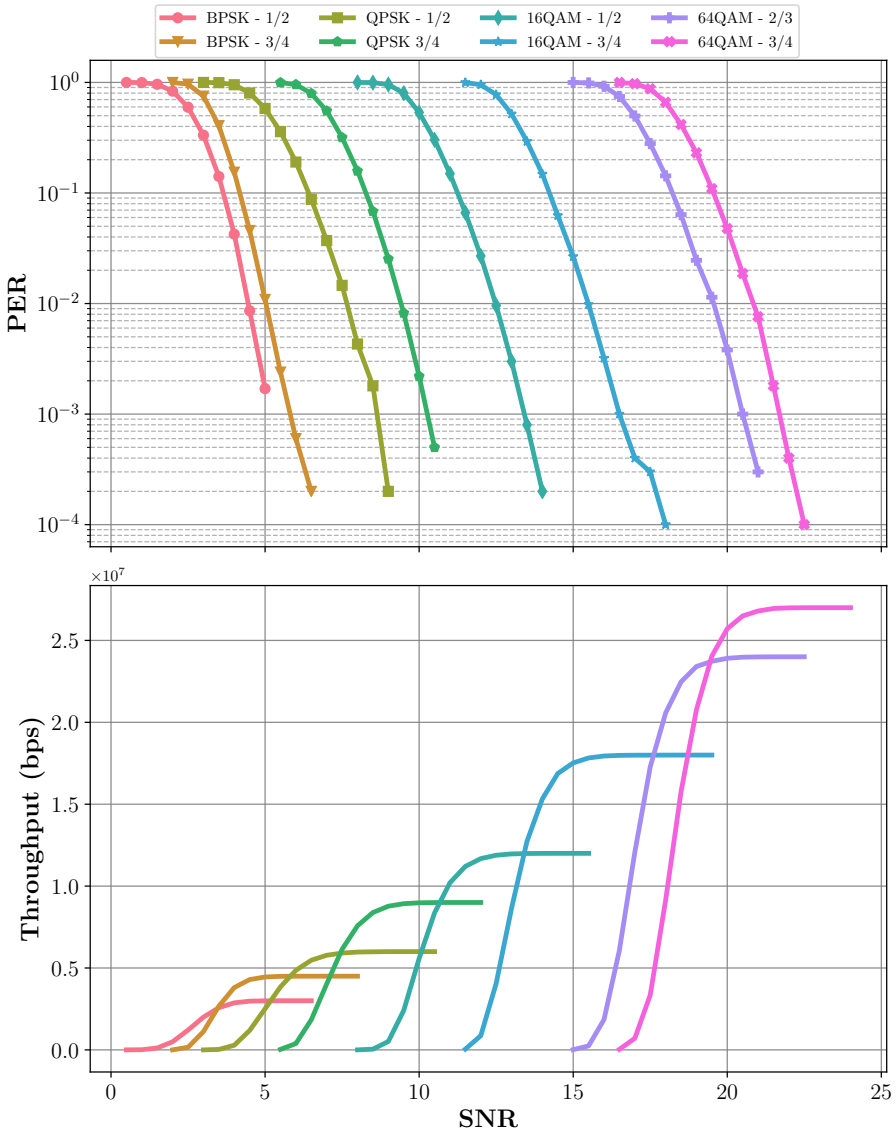


Figure 3.11: AWGN Performance of ITS-G5 PHY

Figure 3.11 shows the AWGN performance of the PHY layer of ITS-G5 for all the supported MCS schemes. For a considered 90% PER, all the schemes satisfy the condition for SNR values $\leq$ 20 dB. Moreover, it can also be seen that there is a wider gap between different modulation schemes whereas the gap is not smaller for different coding rates of the same modulation scheme. The maximum supported throughput for any given MCS is given in Table A.1.

Figure 3.12 and Figure 3.13 shows the performance of C-V2X over AWGN channel without and with the use of a blind retransmission respectively. The solid lines represent the QPSK

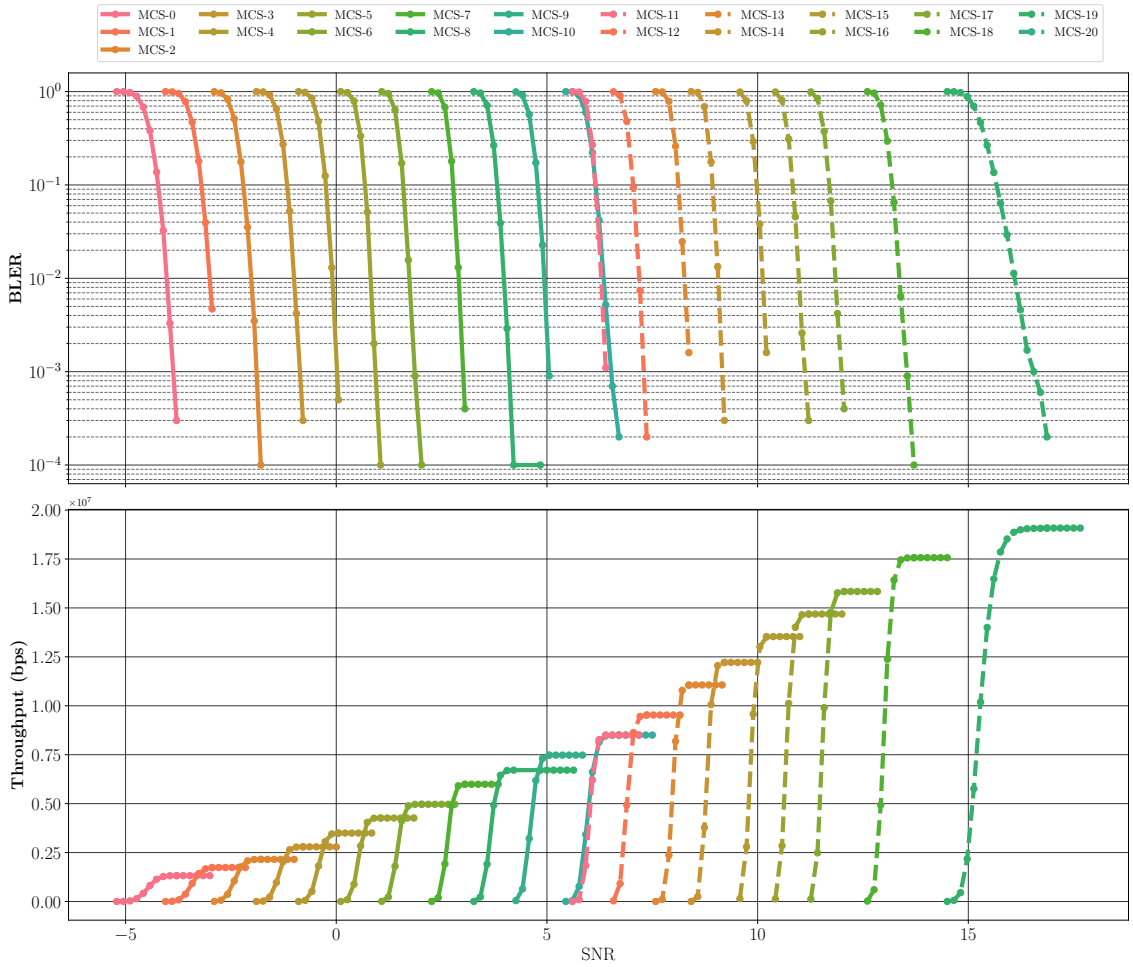


Figure 3.12: AWGN performance for C-V2X without blind retransmission

schemes (MCS0 - MCS10) wheres the dotted lines represent 16QAM schemes (MCS11 - MCS20). For the case of no blind retransmission, it can be seen that all the schemes reach a target BLER of 10% between SNR range of −4 dB to 17 dB. Secondly, there is a gap between 2-3 dB between successive MCS schemes with the exception of MCS20 where there is a larger gap with MCS19. Additionally, it can also be noticed that MCS10 (QPSK) and MCS11 (16QAM) have a similar performance due to the same Transport Block Size (TBS).

With the use of one blind retransmission, the performance improves significantly and all of the schemes achieve a target BLER value of 10% between SNR range of −7 dB to 8 dB. The performance improvement is more pronounced for higher coding rates which is showy by the decreasing gap between successive MCS with increasing coding rate. Notice that the performance gain with retransmission is 3 dB for MCS0 whereas it is almost equal to 5 dB for MCS10. On the other hand, for 16QAM schemes, the gain for MCS11 is 4 dB whereas for MCS20, it is almost 10 dB. Another aspect that can be noticed in the case of blind retransmission is the increased gap between MCS10 and MCS11 which is also due to the better gain for higher coding schemes compared to lower coding schemes.

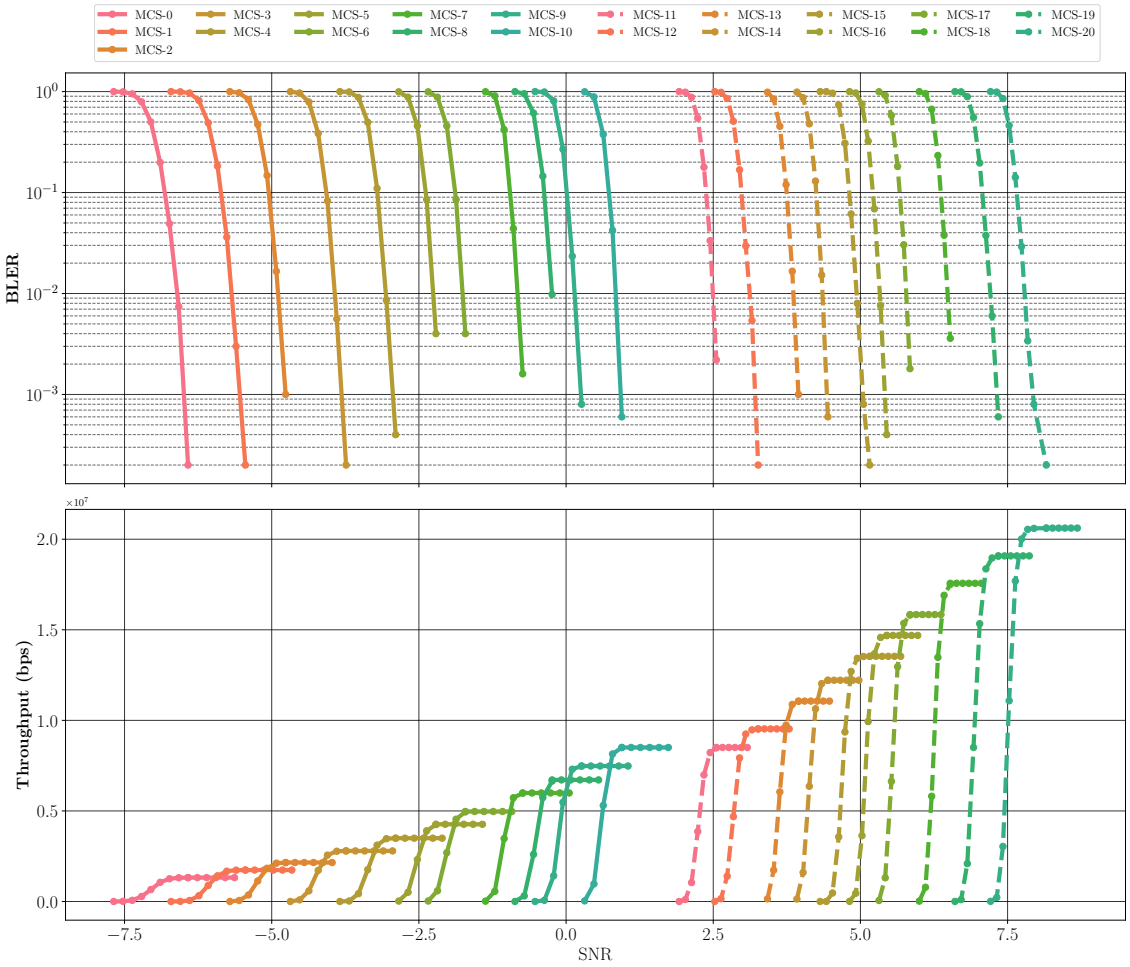


Figure 3.13: AWGN performance for C-V2X with blind retransmission

3.4.2 Fading Channels

Though numerous approaches exist for modeling the channels, we limit our discussion to tapped delay line models [18] that is viewed as sufficiently accurate, and hence is widely used for analysis, simulations and system design. Recall that the channel impulse response of a multi path fading channel in time-domain can be represented by discrete number of impulses as follows

$$h(t) = \sum_{i=1}^{N} a_i \delta(t - \tau_i) \tag{3.22}$$

where a_n and τ_n are the amplitude gain and time delay for the nth multi-path component. This model represents a tapped delay line channel with N taps where the selection of N and the delay values depend upon what is considered a significant level. This significant level is inturn calculated by means of empirical and real world measurement campaigns. For the purpose of simulation, statistical channel models are derived by using the average delay profile measurements as done in [180]. On the other hand, delay/Doppler spread measurements were done by works such as [35, 161, 3] using channel sounding techniques as presented in [1]. Campaigns such as [134, 133, 135] led to the development of Geometry based stochastic channel models [98]. After analysis, we shortlist two classes of channel

models for our link level analyis - the ones recommended by ITU[147] and the ones based on measurements by DSRC Tiger team [3].

In order to have a meaningful comparison, we only considered the MCSs that are common to both the technologies, i.e., QPSK and 16QAM with coding rates 1/2 and 3/4 respectively. For ITS-G5, this corrseponds to MCS schemes 2,3,4,5. The closest counterparts in C-V2X are 7, 10, 14 and 18 respectively.

3.4.2.1 ITU Channel Models

The specification ITU-R M.1225 [147] specifies three different test environments: Indoor office, outdoor-to-indoor pedestrian and vehicular-high antenna. For the vehicular test environment, low (Channel A) and medium (Channel B) delay spreads have been defined with 6 channel taps and an Root Mean Square (RMS) delay spread of 370 ns and 4000 ns respectively. The Channel A model was extended to also support higher bandwidths by adding more paths and delays to the existing model. These models are outlined in Table 3.6.

Table 3.6: ITU SISO Channel Models

Model	Path Delays (ns)	Path Gains (dB)
ITU - VA	[0, 310, 710, 1090, 1730, 2510]	[0, -1, -0, -10, -15, -20]
ITU - VB	[0, 300, 8900, 12900, 17100, 20000]	[-2.5, 0, -12.8, -10, -25.2, -16]
ITU-EVA	[0, 30, 150, 310, 370, 710, 1090, 1730, 2510]	[0, -1.5, -1.4, -3.6, -0.6, -9.1, -7, -12, -16.9]

The ITU channels models (VA, VB and EVA) also support multiple antennae by means of MIMO correlation matrices. Here, we consider a 1X2 configuration with 1 transmit and 2 receive antennas with the correlation matrix as outlined in Table 3.7

Table 3.7: ITU MIMO Correlation Matrices

	One antenna	Two antennas	Four antennas
UE Correlation	$R_{UE} = 1$	$R_{UE} = \begin{pmatrix} 1 & \beta \\ \beta^* & 1 \end{pmatrix}$	$R_{UE} = \begin{pmatrix} 1 & \beta^{\frac{1}{9}} & \beta^{\frac{4}{9}} & \beta \\ \beta^{\frac{1}{9}*} & 1 & \beta^{\frac{1}{9}} & \beta^{\frac{4}{9}} \\ \beta^{\frac{4}{9}*} & \beta^{\frac{1}{9}*} & 1 & \beta^{\frac{1}{9}} \\ \beta^* & \beta^{\frac{4}{9}*} & \beta^{\frac{1}{9}*} & 1 \end{pmatrix}$

Figure 3.14 show the performance comparison between both the PHY technologies for the ITU channel models as defined in Table 3.6 and QPSK schemes. The 16QAM schemes are outlined in Figure 3.15. For the case of C-V2X, both the cases of with and without blind retransmission are shown. Based on the results, the following observations can be made

For the ITU-VA channel in SISO configuration, both the technologies exhibit a very similar performance (especially for 16QAM schemes). However, C-V2X has a slight edge over ITS-G5 with a gain between 1 dB to 2 dB with the exception of 16QAM-3/4 where ITS-G5 performs better than C-V2X. It can also be seen that the C-V2X gain is more pronounced for QPSK-3/4 than it is for QPSK-1/2. Additionally, C-V2X schemes show a more steeper degradation of BLER at high SNRs compared to ITS-G5. When retransmissions are enabled for C-V2X, one can clearly notice a performance gain of 7 dB to 9 dB with 3/4 coding

schemes showing maximal exploitation of the retransmission. In SIMO configuration, it can be noticed that, for a reference BLER of 10 %, C-V2X shows a slightly better performance (2 dB to 3 dB) compared to ITS-G5 (with the exception of 16QAM-3/4), highlighting the superiority of the channel estimation scheme employed in C-V2X. On the other hand, 16QAM-3/4 scheme in ITS-G5 outperforms C-V2X by 1.3 dB.

The ITU-VB is an extremely dynamic scenario with delay spread of 20 µs that is almost 4 times the CP length of C-V2X. This results in a very high Inter-Symbol Interference (ISI) and therefore the reason why both the technologies show worse performance. For the SISO configuration, it can be seen that none of the schemes (both QPSK and 16QAM) for both the technologies achieve a target BLER of 10 %. However, with retransmissions enabled for C-V2X, both the QPSK schemes achieve the target BLER at SNR values of 6 dB and 7.5 dB respectively. On the other hand, even with retransmissions enabled, the 16QAM schemes do not achieve the target BLER. For the SIMO configuration, the performance of all the schemes gets better. It can be seen that both the QPSK schemes of C-V2X achieve a target BLER (with and without blind retransmission) whereas only the QPSK-1/2 scheme of ITS-G5 passes this BLER requirement. On the other hand,the BLER requirement for 16QAM schemes are only satisfied in C-V2X using one blind retransmission.

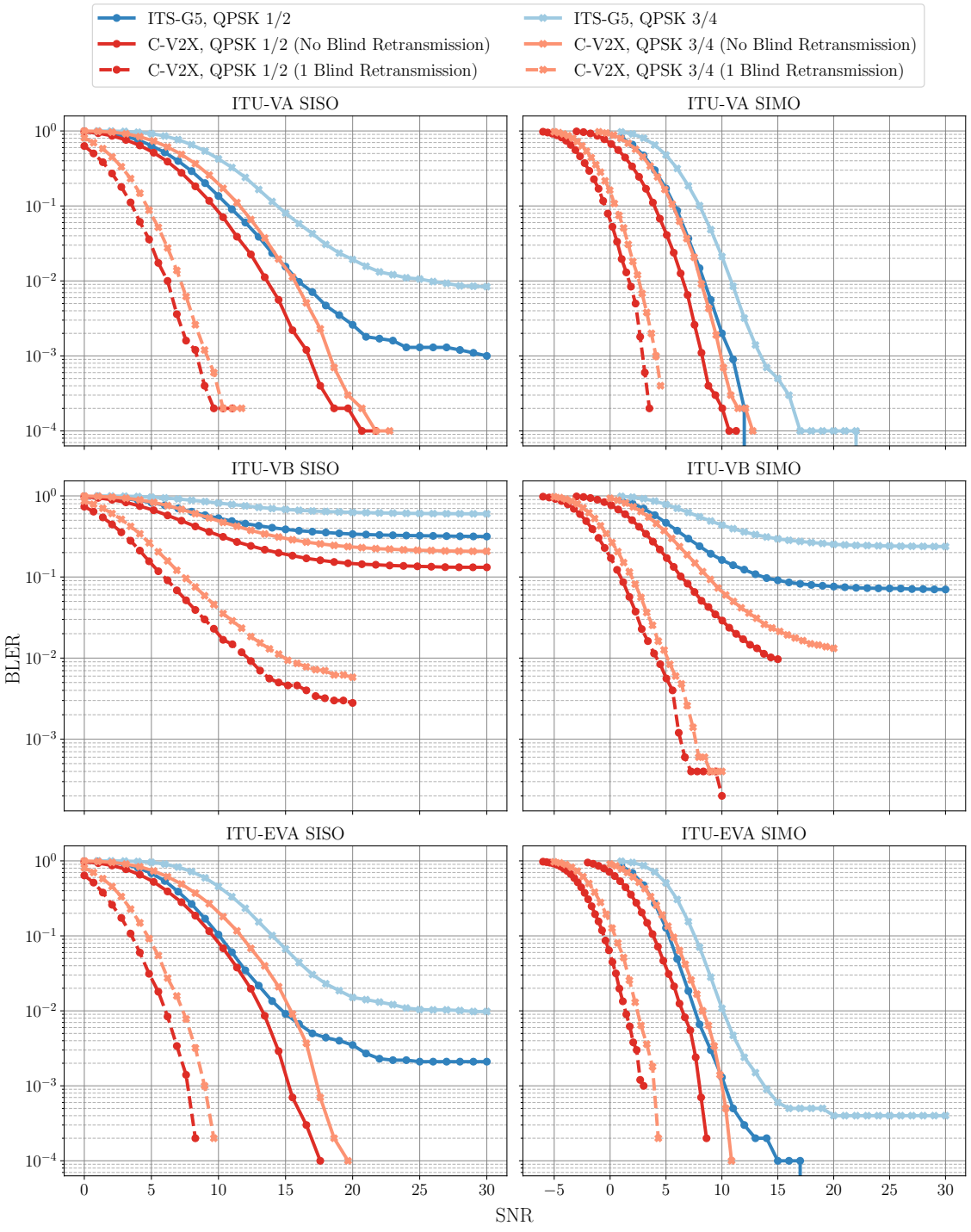


Figure 3.14: Performance Comparison for QPSK Schemes over ITU Channel Models

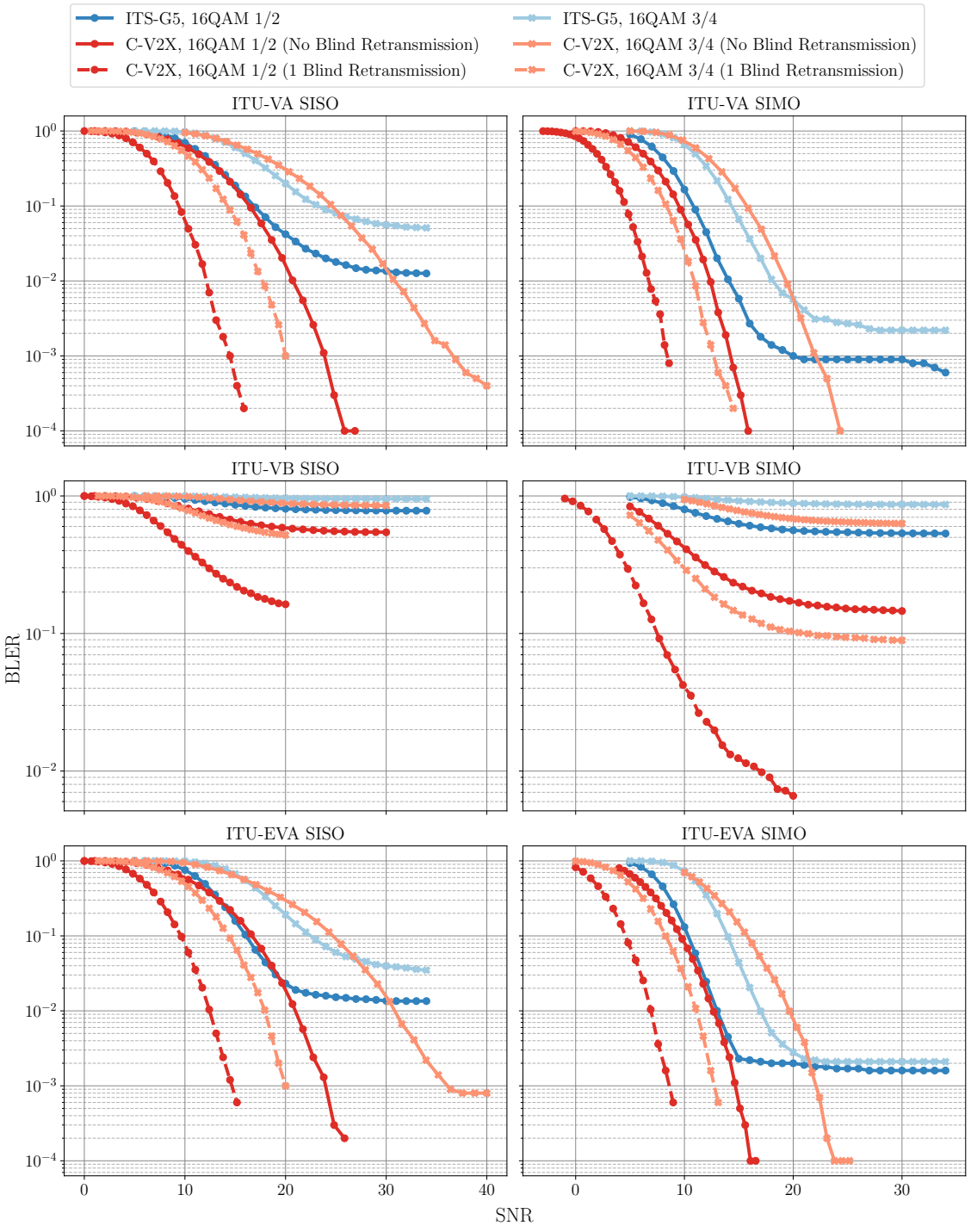


Figure 3.15: Performance Comparison for 16QAM Schemes over ITU Channel Models

ITU-EVA is basically an extension of ITU-VA with additional paths and delays. Therefore the performance of both the technologies is similar to that of ITU-VA model. A key difference that can be noted here is that the higher coding schemes in ITS-G5 (QPSK-3/4 and 16QAM) show a slight deterioration in performance compared to that of ITU-VA channel.

3.4.2.2 DSRC V2V Channel Models

During 2007-2010, a total of 35 field trial campaigns were conducted on public roads in US, Germany, Austria, Italy and Australia totaling over 1100 kilometres and covering a wide variety of physical environments [3]. These campaigns were scenario based and demonstrated different V2I and V2V scenarios such as Intersection Movement Assist (IMA), Do Not Pass Warning (DNPW), Emergency Electronic Brake Light (EEBL) and driving across an RSU. For each test location, multiple repetitions of a scenario were run transmitting messages at an aggregate of 400 packets/s. For the purpose of measurements, vehicles mounted with Cohda wireless MKI 802.11p DSRC units with single antenna were used. The channel sounding data captured during the field trials were analysed to obtain delay and Doppler spread characteristics. Using these statistics, a total of 5 channel models were proposed for different scenarios and are outlined in Table 3.8.

Table 3.8: V2V Channel Models

Scenario	Path Delays (ns)	Path Gains (dB)	Doppler Shift (Hz)
Rural LOS	[0, 83, 183]	[0, -14, -17]	[0, 492, -295]
Urban Approaching LOS	[0, 117, 183, 333]	[0, -8, -10, -15]	[0, 236, -157, 492]
Urban NLOS	[0, 267, 400, 533]	[0, -3, -5, -10]	[0, 295, -98, 591]
Highway LOS	[0, 100, 167, 500]	[0, -10, -15, -20]	[0, 689, -492, 886]
Highway NLOS	[0, 200, 433, 700]	[0, -2, -5 -7]	[0, 689, -492, 886]

Figure 3.16 show the performance comparison between both the PHY technologies for the V2V channel models as defined in Table 3.8 and QPSK schemes. The 16QAM schemes are outlined in Figure 3.17. For the case of C-V2X, both the cases of with and without blind retransmission are shown. Based on the results, the following observations can be made

For the R-LOS channel in SISO configuration, C-V2X clearly shows a superior performance with a gain of 1 dB to 2 dB for QPSK schemes. In fact, it can be seen that QPSK-3/4 scheme in ITS-G5 has a similar performance to that of QPSK-1/2 scheme in C-V2X. On the other hand, 16QAM schemes show a mixed result with C-V2X showing better performance with a gain of 1.2 dB for 16QAM-1/2 scheme and ITS-G5 showing better performance for 16QAM-3/4 with a gain of 1.2 dB. However, with retransmissions enabled, C-V2X outperforms ITS-G5 for all the schemes with a gain ranging between 5 dB to 7 dB.

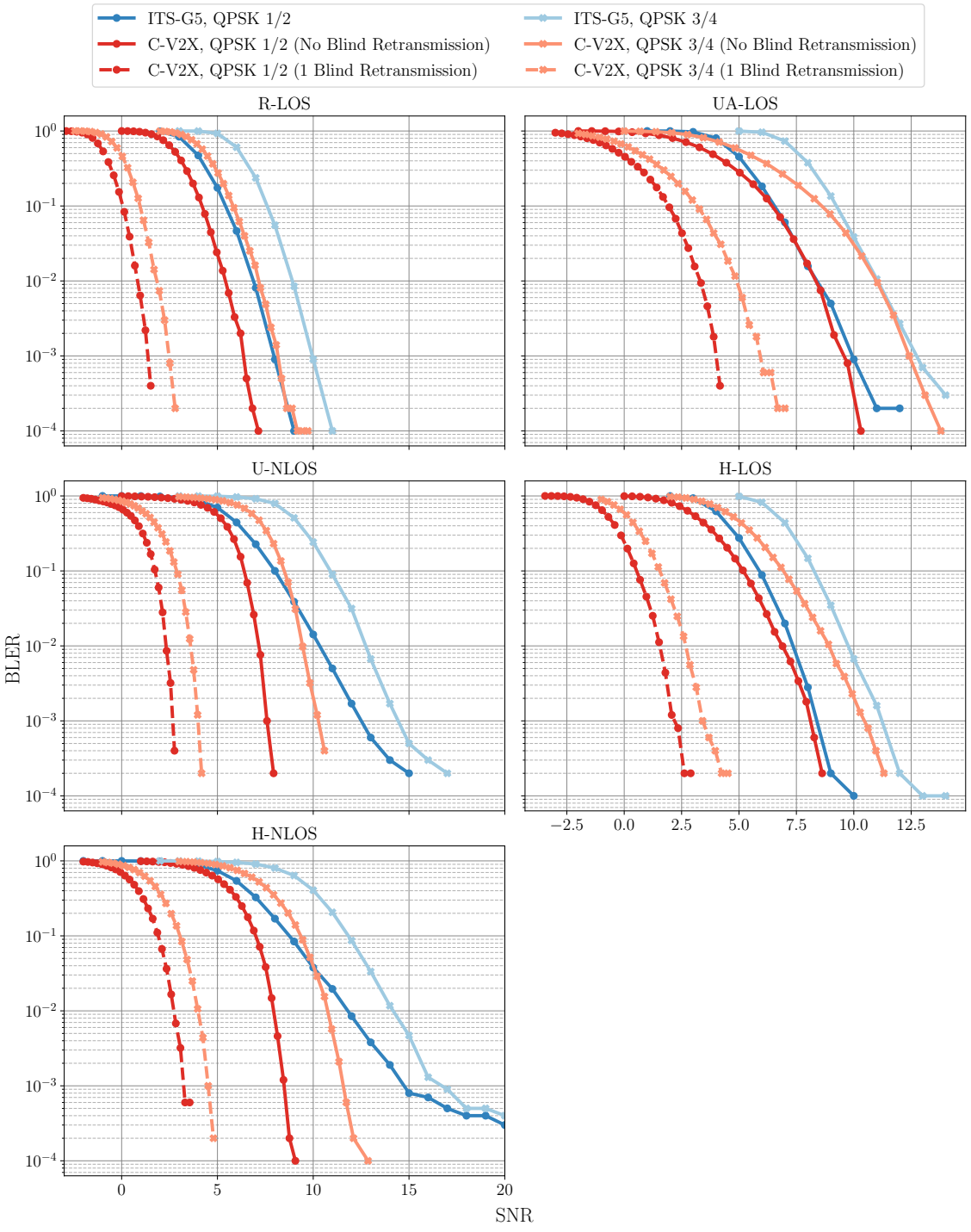


Figure 3.16: Performance Comparison for QPSK Schemes over V2V Channel Models

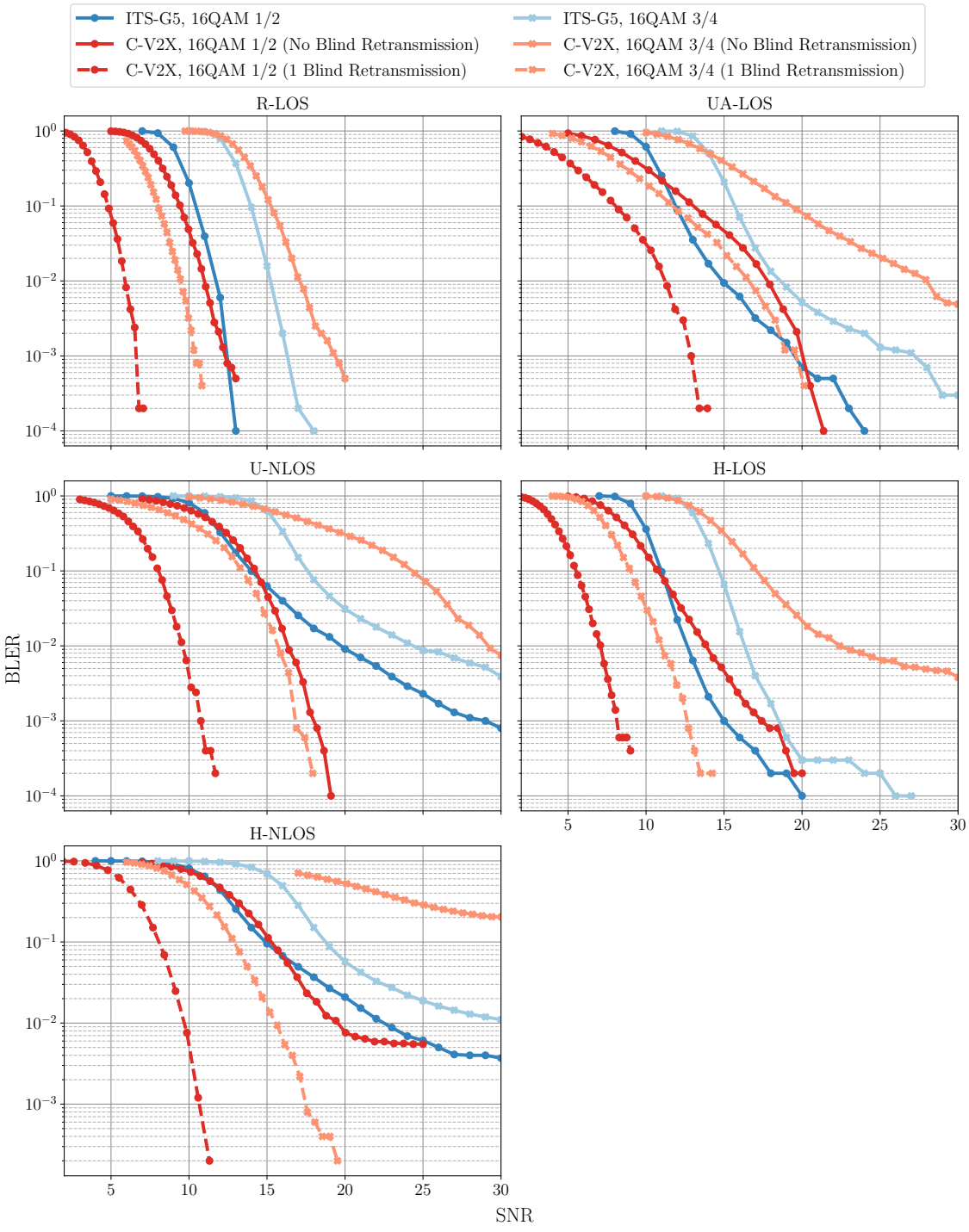


Figure 3.17: Performance Comparison for 16QAM Schemes over V2V Channel Models

For the UA-LOS channel in SISO configuration, the performance of both the technologies also very similar. A key takeaway here is that C-V2X shows a low BLER at low SNRs for QPSK schemes but as the SNR increases, the performance of both the technologies converges. However, for 16QAM schemes, even though C-V2X exhibits low BLER at lower SNRs, ITS-G5 compensates this effect via higher degradation in terms of BLER as the SNR increases. This also results in ITS-G5 showing a superior performance than C-V2X with a gain of 1.1 dB and 3.5 dB for 16QAM-1/2 and 16QAM-3/4 respectively. With the use of retransmissions, all the C-V2X schemes outperform ITS-G5 with gains ranging between 3 dB to 6 dB.

For the U-NLOS channel model, C-V2X, the QPSK schemes for C-V2X show a substantial performance superiority over ITS-G5. For a target BLER of 10%, the gain ov C-V2X is 1.6 dB and 2.4 dB for QPSK-1/2 and QPSK-3/4 respectively. Additionally, the performance gap between C-V2X and ITS-G5 increases monotonically as the SNR values increase with C-V2X showing a higher degradation in BLER. When it comes to 16QAM schemes, even though ITS-G5 has a slight performance edge over C-V2X at a target BLER of 10%, C-V2X shows a significant performance improvement and overtakes ITS-G5 at increasing SNR values. However, the same cannot be said for 16QAM-3/4 where ITS-G5 beats C-V2X in terms of performance over the entire SNR spectrum. With the use of retransmissions, all the C-V2X schemes outperform ITS-G5 with gains ranging between 4 dB to 8 dB.

For the H-LOS channel model, the performance of the two technologies follows closely their trend in UA-LOS model. At lower SNRs, C-V2X shows lower BLER for QPSK schemes and as the SNR increases, the performance of both the technologies converges. Unlike UA-LOS, C-V2X shows a slight edge over ITS-G5 with a gain of 1 dB to 3 dB across the entire SNR range. The 16QAM-1/2 scheme in C-V2X starts good but is overtaken by ITS-G5 as the SNR increases. On the other hand, ITS-G5 excels in comparison to C-V2X for 16QAM-3/4 across the entire SNR spectrum. With the use of retransmissions, all the C-V2X schemes outperform ITS-G5 with gains ranging between 5 dB to 7 dB.

For the H-NLOS channel model, the performance of both the schemes follow similar trend as U-NLOS model.For a target BLER of 10%, the QPSK schemes in C-V2X show a performance superiority of 1.8 dB and 2.5 dB over their ITS-G5 counterparts respectively. This performance gap widens further as the SNR increases. For the 16QAM-1/2 scheme, the performance of both the technologies looks similar with C-V2X showing a steeper descent in terms of BLER at mid SNR ranges and converging at high SNRs. However, the 16QAM-3/4 scheme in ITS-G5 performs way better than C-V2X with the latter failing to even achieve the target BLER value. With the use of retransmissions, all the C-V2X schemes outperform ITS-G5 with gains ranging between 5 dB to 9 dB.

Table 3.9: SNR Thresholds (in dB) for a target BLER of 10% for ITU Channel Models

Channel Model	MCS	C-V2X	C-V2X (with retransmission)	ITS-G5	C-V2X Gain	C-V2X Gain (with retransmission)
ITU-VA SISO	QPSK-1/2	9.69	3.61	10.79	1.1	7.18
	QPSK-3/4	11.63	4.69	14.42	2.79	9.73
	16QAM-1/2	16.44	9.43	16.89	0.45	7.46
	16QAM-3/4	24.64	14.25	23.27	-1.37	9.02
ITU-VA SIMO	QPSK-1/2	3.98	-0.44	5.84	1.86	6.28
	QPSK-3/4	5.64	0.48	8.02	2.38	7.54
	16QAM-1/2	9.51	4.6	10.86	1.35	6.26
	16QAM-3/4	15.76	8.38	14.40	-1.36	6.02
ITU-VB SISO	QPSK-1/2	NA	5.98	NA	NA	inf
	QPSK-3/4	NA	7.47	NA	NA	inf
	16QAM-1/2	NA	36.71	NA	NA	inf
	16QAM-3/4	NA	NA	NA	NA	NA
ITU-VB SIMO	QPSK-1/2	6.37	0.97	13.73	7.36	12.76
	QPSK-3/4	8.76	1.96	NA	inf	inf
	16QAM-1/2	NA	7.52	NA	NA	inf
	16QAM-3/4	NA	21.19	NA	NA	inf
ITU-EVA SISO	QPSK-1/2	9.65	3.55	10.9	1.25	7.35
	QPSK-3/4	11.73	4.73	14.04	2.31	9.31
	16QAM-1/2	16.7	9.62	16.09	-0.61	6.47
	16QAM-3/4	24.75	14.35	22.52	-2.23	8.17
ITU-EVA SIMO	QPSK-1/2	3.87	-0.55	5.35	1.48	5.9
	QPSK-3/4	5.63	0.47	7.65	2.02	7.18
	16QAM-1/2	9.66	4.62	10.43	0.77	5.81
	16QAM-3/4	15.78	8.27	13.97	-1.81	5.7

3.4.3 Summary & Conclusions

In this section, the link level performance was compared for both ITS-G5 and C-V2X by means of extensive simulation campaigns. To realize this, the PHY level processing chain was explained in detail for both the technologies. For ITS-G5, existing open source implementation in Matlab was used. For C-V2X, the complete PHY level chain was built in Python using standard computational libraries such as Numpy, Pandas etc. The resulting link level simulator was validated by running the reference channel tests as specified by TS36.101 and the results show that the simulator performs in line as the specification.

In the next step, both the technologies were simulated for various channel models starting with AWGN. For fading channels, both the models from ITU and the DSRC tiger team were used. Additionally, SIMO simulations were also carried out for ITU models by means of correlation matrices. All the results were plotted in terms of SNR - BLER graphs. The results show that,in single transmission scheme, C-V2X outperforms ITS-G5 in almost all of the considered channel models with some exceptions with 16QAM and higher coding schemes. With one blind retransmission enabled, C-V2X exhibits a gain of atleast 6 dB and in some cases, reaching as high as 10 dB over ITS-G5.

The respective gains were also summarized and outlined in Table 3.9 for ITU channel models and Table 3.10 for DSRC models. *Inf* denotes that the gain cannot be calculated since one of the technology could not achieve the target BLER of 10 %. Color codes were used to

Table 3.10: SNR Thresholds (in dB) for a target BLER of 10% for DSRC V2V Channel Models

Channel Model	MCS	C-V2X	C-V2X (with retransmission)	ITS-G5	C-V2X Gain	C-V2X Gain (with retransmission)
R-LOS	QPSK-1/2	4.21	0.07	5.58	1.37	5.51
	QPSK-3/4	5.83	0.98	7.75	1.92	6.77
	16QAM-1/2	9.43	4.82	10.63	1.2	5.81
	16QAM-3/4	15.27	8.03	13.99	-1.28	5.96
UA-LOS	QPSK-1/2	6.48	1.93	6.68	0.2	4.75
	QPSK-3/4	8.64	3.18	9.37	0.73	6.19
	16QAM-1/2	13.07	8.06	11.94	-1.13	3.88
	16QAM-3/4	19.32	11.72	15.79	-3.53	4.07
U-NLOS	QPSK-1/2	6.43	1.74	8.00	1.57	6.26
	QPSK-3/4	8.52	2.88	10.93	2.41	8.05
	16QAM-1/2	14.26	8.04	13.99	-0.27	5.95
	16QAM-3/4	24.32	13.42	17.68	-6.64	4.26
H-LOS	QPSK-1/2	5.19	0.56	5.94	0.75	5.36
	QPSK-3/4	6.94	1.56	8.42	1.48	6.86
	16QAM-1/2	10.76	5.52	10.99	0.23	5.47
	16QAM-3/4	17.11	9.01	14.80	-2.31	5.79
H-NLOS	QPSK-1/2	7.02	1.92	8.81	1.79	6.89
	QPSK-3/4	9.36	3.05	11.89	2.53	8.84
	16QAM-1/2	15.30	8.14	14.92	-0.38	6.78
	16QAM-3/4	NA	12.90	18.80	-inf	5.9

show the superiority of C-V2X over ITS-G5. Green indicates better performance whereas orange indicates worse performance of C-V2X in relation with ITS-G5. The summary table validates the superiority of C-V2X over ITS-G5 especially with QPSK schemes which are the most commonly used MCS for safety messages.

4 System Level Analysis of ITS-G5 and C-V2X

While the link level simulations focus on a single link and a detailed PHY layer model, system level simulations focus more on higher layer issues such as resource allocation and scheduling, multiple access, mobility management, network planning and optimization etc [120, 143, 197]. They do so by means of abstracting the link level model (to reduce computational complexity) and by placing a sufficiently large number of base stations and UEs to obtain average system performance [38]. For the case of adhoc networking protocols such as the ones used for V2X communication, system level simulations boils down to the analysis of MAC layer since it is in charge of all the system level functions such as resource allocation and channel access. The purpose of this chapter is as follows

1. Present the MAC layer architectures of both ITS-G5 and C-V2X RATs
2. Perform a system level simulation with multiple nodes to evaluate the MAC performance of both the RATs

The MAC layer and the Logical Link Control (LLC) layer together constitute the data link layer (Layer 2 in OSI model). MAC layer is responsible for a number of functions including addressing and channel access controlling mechanism. For multiple nodes in a network to communicate through shared medium, MAC sublayer provides channel access controlling mechanism known as multiple access protocol. The multiple access protocol design for vehicular networks must be carefully designed keeping the following criteria in mind [13]

1. The number of nodes participating in the network is very dynamic and cannot be restricted.
2. Vehicular networks are required to be operated in decentralized manner which restricts the use of centralized multiple access schemes such as Frequency Division Multiple Access (FDMA), TDMA or Code Division Multiple Access (CDMA) etc. This in turn effects the scalability of solutions that can be applied due to the lack of central coordination that has the global knowledge about all nodes in a network
3. Complex propagation conditions that arise due to environment (obstructions due to buildings, streets and other vehicles), scattering (due to low antenna heights) and high doppers (due to high relative mobility). This in turn has effects when using sensing based channel access schemes as well as reliability of the transmission. This problem is further aggravated by the fact that no feedback channels can be implemented in broadcast scenarios. The only way out is to use blind retransmissions which further congests the network.

4.1 ITS-G5

As already seen from previous chapters, ITS-G5 inherits its PHY and MAC layer functionalities from DSRC which inturn is based on the IEEE 802.11a standard. However, the biggest challenge for reusing the 802.11 standard for vehicular communications are the MAC delays associated with connection setup times. Typical IEEE 802.11a links are expected to be quasi-stable, long-term connections with time intensive procedures such as synchronization, authentication and association. Secondly, a device can only be associated with only one BSS at any given time. Given the short term and highly dynamic nature of the vehicular links, these procedures are infeasible. Hence, 802.11p changed the MAC layer by introducing a new mode of operation called WAVE allowing devices to operate without being part of a BSS. The stations are expected to transmit and receive frames on a CCH using a predetermined set of PHY layer parameters, due to which security and other confidentiality measures that are typically taken care of by the MAC layer, cannot be used. Thereby, this mode enables non-sensitive information to be exchanged without any association between proximate devices [86, 175].

The WAVE mode was designed around the assumption that a single CCH is used to exchange some basic set of information. However, the vehicles should also be able to communicate on all the remaining allocated channels in the ITS band, called the SCHs. Since, this would result in vehicles missing broadcast messages (that are typically transmitted in CCH) due to using a random channel at any given time, some co-ordination is necessary to enable all the vehicles to switch to CCH for a defined period of time (t_c). After this default time, the vehicles are free to tune to any other channel of their choice for another time period (t_s), after which the cycle repeats. By default, both the values of t_c and t_s are set to 50 ms unless overruled by higher layers. To account for synchronization errors, the first 4 ms of the 50 ms interval serve as guard interval and are left ununsed. These MAC layer enhancements were standardized as IEEE 1609.4 and was published in 2006.

The multiple access protocol used is directly derived from IEEE 802.11a where the access to the channel is governed by a so called DCF that employs CSMA/CA as the underlying protocol. The DCF is used for contention resolution among different nodes whereas EDCA from IEEE 802.11e is used for contention resolution among different ACs within a single node.

4.1.1 DCF

The DCF provides basic rules for channel access by multiple nodes and preventing/minimizing packet collisions in the process. It does so by means of CSMA/CA where the Carrier Sense (CS) part prevents the node from transmitting when it senses the channel to be busy and the CA part avoids collisions when they are most likely to occur by deferring the transmission. Specifically, the sequence of operations performed assuming the node starts from an idle state are shown in Figure 4.1

1. Upon reception of a new packet from higher layers, the MAC initiates CS. If the channel is idle for Arbitration Inter-Frame Spacing (AIFS), the transmission may start immediately. If the channel is sensed as busy, then the node enters contention phase, which is divided into a countdown and a freeze state

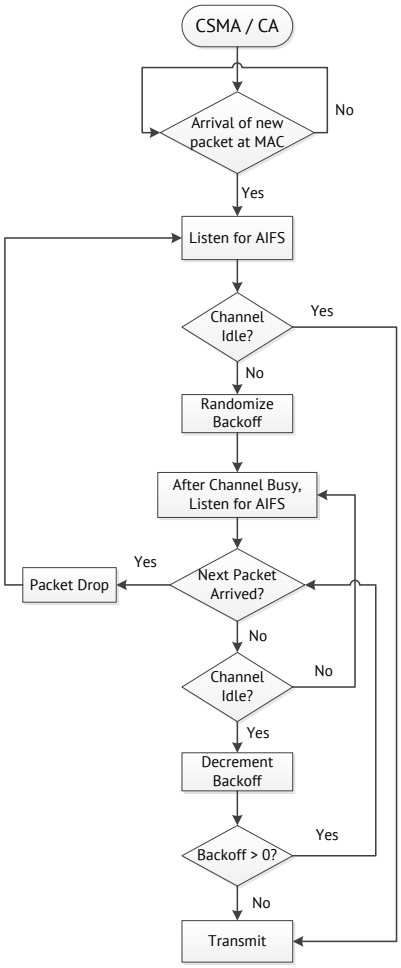


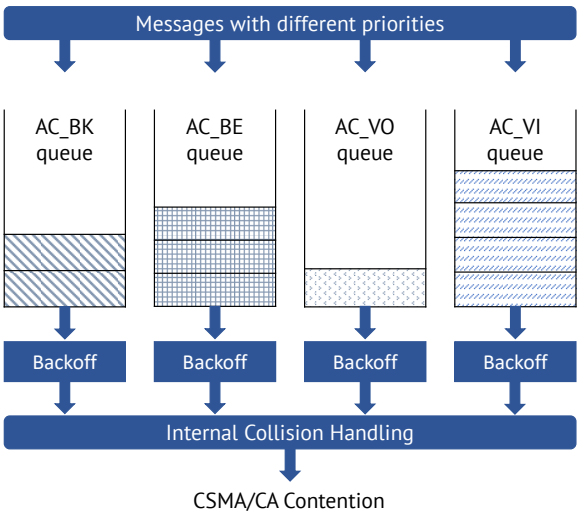
Figure 4.1: CSMA/CA Protocol Sequence

2. The node draws a Backoff Counter (BC), which is a uniformly distributed Random Variable (RV) drawn from the interval $[0, CW_{min}]$ where CW_{min} is the minimum (initial) length of the Contention Window (CW) defined for its AC
3. When the channel turns idle, the node waits for one AIFS. If the channel is sensed idle, the BC is decremented (countdown). If the channel is sensed as busy during countdown, the BC is frozen and this step is repeated again. In the meantime, if a new packet has arrived at MAC, the current packet is dropped (Packet error) and the process is restarted from Step 1.
4. When BC value reaches 0, the node starts transmitting the message. Transmission may also happen if the node selects 0 as BC value in Step 2.

4.1.2 EDCA

EDCA has been introduced in IEEE 802.11e in order to enable service differentiation for packets of different flows. This is motivated by the fact, that there exists different kinds of messages each with different traffic patterns and Quality of Service (QoS) requirements. It does so by means of classifying them into different ACs and two controllable parameters - 1. Using different AIFS for each AC so that nodes with longer AIFS have less chance

of accessing the medium than the nodes with shorter AIFS 2. Using different contention windows for different ACs so that nodes with smaller CWs access the channel faster and more frequently than the nodes with larger CWs



Access Categories and Parameters

AC	CW_{min}	CW_{max}	AIFSN
AC_BK	15	1023	9
AC_BE	15	1023	6
AC_VO	7	15	3
AC_VI	3	7	2

Figure 4.2: IEEE 802.11p EDCA Mechanism and the associated ACs

Figure 4.2 shows the EDCA functional methodology along with the different ACs used for IEEE 802.11p. Arriving messages are put in their respective queue where each queus acts as an virtual node and has its own countdown and freeze states. For a given time step, the message whose BC equals 0 wins the contention and is then scheduled to be transmitted where it has to contend with other stations as explained in subsection 4.1.1. If two or more messages win contention at the same time, then the message with the highest priority is scheduled and the one with lower priority is dropped. Due to the internal nature of this contention, virtual collisions and packet drops occur at each node. Figure 4.2 also shows the different ACs that are used in IEEE 802.11p and their associated parameters. Four types of traffic are supported namely BK (Background), BE (Best Effort), VO (Voice) and VI (Video) respectively. Arbitration Inter-Frame Spacing Number (AIFSN) is a configurable parameter that is used to derive the AIFS for a given AC as follows

$$AIFS_{AC_i} = t_{SIFS} + AIFSN_{AC_i} * t_{slot}$$

where t_{SIFS} is the Short Inter-Frame Space and t_{slot} is the slot time whose values are derived from the PHY specifications. For IEEE 802.11p, the value of t_{SIFS} is 32 µs and t_{slot} is 13 µs respectively [86]. $AIFSN_{AC_i}$ is dependent on the selected AC whose values are given in the table in Figure 4.2.

4.2 C-V2X

Multiple access in C-V2X is controlled by means of resource allocation to the nodes in both time and frequency domain whilst maintaining as much orthogonality as possible. C-V2X supports both centralized and decentralized channel access schemes by means of Mode 3 and Mode 4 respectively. In Mode 3, the eNB is in charge of allocating and scheduling resources to the nodes whereas Mode 4 supports autonomous resource selection by the participating nodes. In general, Mode 3 is expected to outperform Mode 4 due its global

knowledge about the participating nodes [195, 30], mode 4 remains the only option during out-of-coverage scenarios. Here, we focus only on Mode 4 resource allocation scheme for our analysis.

As outlined in Figure 3.2, V2X transmissions in C-V2X occur in contiguous Resource Blocks (RBs) where each RB spans over 12 SCs in frequency domain totaling 180 kHz and 1 ms that correponds to 14 OFDM symbols in time domain. These set of contiguous RBs are also referred to as subchannels whose size and number are pre-set by the network according to some rules given in [189]. Each subchannel can carry at most one data packet, although one data packet can span over multiple subchannels. For each data packet, 2 RBs are required to transmit the control data and the rest are used for the data (also referred to as Transport Block (TB)). The number of subchannels allocated to carry one data packet, also referred to as Beaconing Resources (BRs) [16], depends on the kind of allocation (adjacent or nonadjacent RBs for control and data), the subchannel size, the TBS and the MCS used. So, in principle, each data packet in C-V2X is sent on one single BR that is also used as the basic unit for resource allocation.

The resource allocation scheme in Mode 4 is called Semi Persistant Subchannel Selection (SPSS) and basically consists of two operations [90, 16] 1. Sensing part where the node tries to senses the channel and maps out the free and occupied BRs. 2. Resource Selection part that selects/reuses the BR for the subsequent transmissions. The selected resources are persisted for some message periods and hence this process is also referred to Semi-Persistent Scheduling.

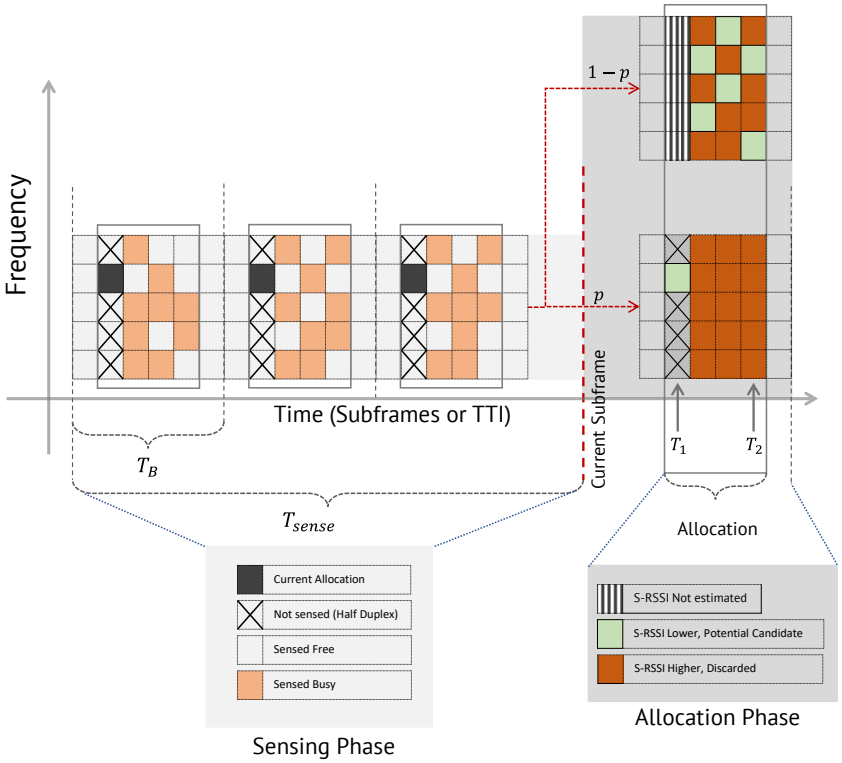


Figure 4.3: Sensing Scheme for C-V2X SPSS. Here $T_B = 6$, $T_1 = 1$ and $T_2 = 5$ [16]

4.2.1 Sensing

The sensing operation is done as part of the PHY layer operations where the node tries to decode the control messages (SCI messages) and measures the Received Signal Strength Indicator (RSSI) called Sidelink RSSI ($S - RSSI_{BR_i}$) in each BR. If the measured $S - RSSI_{BR_i}$ is greater than a threshold P_{th}, then this BR is marked as busy and if $S - RSSI_{BR_i} < P_{th}$, the BR is marked as free. The current BR that the node is using is discarded from sensing due to half duplex limitations, i.e., a node cannot sense during transmitting. The duration for which the node keeps the Reference Signal Received Power (RSRP) history of the BRs are is called the sensing window/interval (T_{sense}). The standard [188] specifies a T_{sense} of 1 s corresponding to 1000 subframes.

Given the sensing information over T_{sense}, the node focuses on the BRs in the next message period (T_B) that lay in the interval T_1 and T_2 subframes where T_1 and T_2 are parameters that define the start and end of the resource allocation subframes. The node then calculates the average S-RSSI of BRs that are marked as free within the interval $[T_1, T_2]$ and sorts them in ascending order. It then selects a portion n_R of these BRs with the lowest value, such that

$$n_R = \lceil R_{sel} * R \rceil$$

where R_{sel} is a parameter that denotes the portion of BRs that should be passed to the MAC layer and R is the number of BRs in one T_m. If the number of candidate BRs are less than n_R, due to a high number of BRs sensed as busy, then P_{th} is decreased by 3 dB and the previous step is repeated until the required number of candidate BRs are reached. These n_R BRs are then passed to the MAC

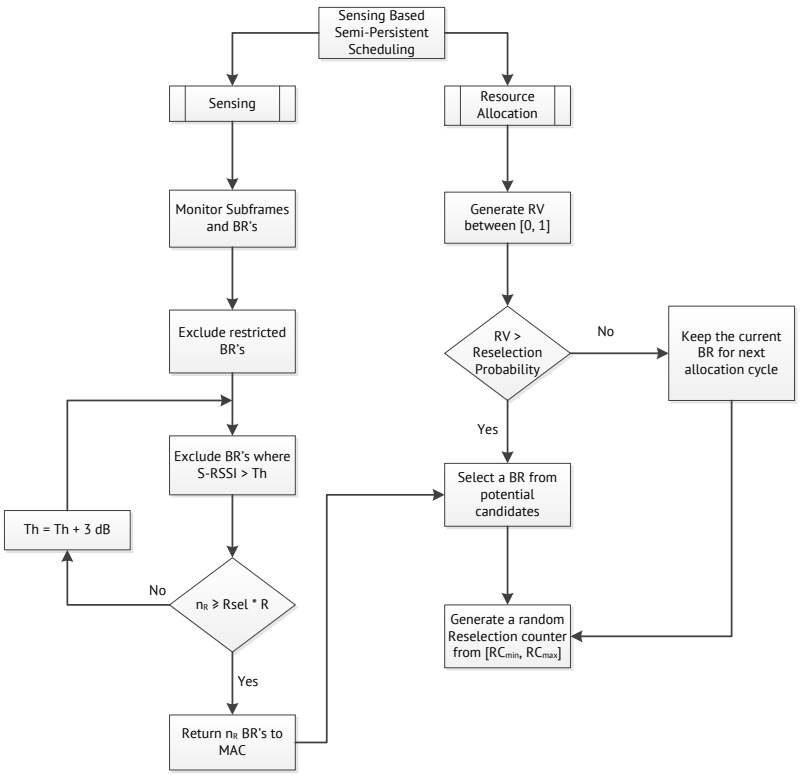


Figure 4.4: SPSS Flowchart Sequence

4.2.2 Resource Selection

Once the MAC layer receives the n_R BRs from the sensing operation, a decision is made as to whether to choose a different BR or to keep the current allocation. This additional step is used to mitigate the effect of two UEs selecting the same resource in the previous attempt and is done by means of generating a RV between $[0, 1]$ and comparing it with the parameter p_k that is defined at higher layers. If the generated $RV \geq p_k$, then a new BR is randomly selected from the candidate BRs. Otherwise, the UE continues to use the same BR(persistence). After selecting the resource, another RV called Resource Reselection Counter (RRC) is generated from an interval defined as $[RC_{min}, RC_{max}]$ that defines the number of times that the selected BR is used for transmission without reselection. With each transmission instance, this RRC is decremented and a new selection procedure is instantiated once it's value becomes 0.

4.3 Simulation Modeling

The system level performance can be basically evaluated by means of analytical methods or by means of simulations. In this work, we adopt the latter case wherein, by means of extensive simulations, we analyze and compare the MAC level performance of both the V2X technologies. In this regard, a distinction needs to be made between simulating vehicular networks to the traditional cellular networks due to the following points [175]

1. Traditional network simulations at the system level needs to consider a detailed model of the underlying RAN and the functions of the eNB. For vehicular networks, this is not required due to the decentralized operation where the vehicles communicate with each other in adhoc mode. Even though C-V2X supports eNB support in Mode 3, we do not consider this in our work. It also helps to keep the simulator complexity low compared to traditional networks.

2. Due to the dynamic topology, the simple mobility models (e.g., random waypoint [96]) that are typically used in traditional network simulators can no longer be used [214]. A more detailed and realistic traffic modeling needs to be done on a microscopic level and should also consider factors such as lane change, car following models, intersection models etc. Additionally, models describing the driver's behavior also needs to be considered so as to model real life events such as non-reactions to traffic bulletins. Secondly, when evaluating real-world V2X applications, it can be noticed that that not only the node mobility influences the network traffic, but also the network traffic influences the node mobility. The best example for this is dynamic route selection where the vehicles would be informed about any traffic congestion ahead so that they can take an alternative route, which in turn influences the network traffic.

3. Key Performance Indicators (KPIs) and metrics such as PER, latency might be insufficient to describe the performance of a certain V2X application. In this regard, traffic specific metrics such as CO_2 emissions and travel time needs to be considered as well.

In principle, any simulation can always be written from scratch, implementing all the aspects described above in a programming language of choice. In case of vehicular networking simulations, detailed traffic models also needs to be developed alongside network

models. However, it brings with it a big burden on the developer in terms of implementation complexity and the efforts needed to maintain it. Moreover, the correct execution of the simulation depends on developer centric aspects like correct implementation, seeding, avoiding initialization bias and the collection of statistically valid results. Additionally, if the simulation is not well documented, it hinders reproducibility and comparability of the results [137, 95]. On the other hand, reusing existing validated simulation frameworks avoid the problem of implementation complexity, but increases integration complexity especially if the existing framework does not follow modular development. However, this offers the flexibility to choose the best from both worlds (networking and traffic modeling communities) and integrate them together to get the desired simulation framework. This is the approach that we follow in this work.

Available Network and Traffic Simulation Frameworks

There exists many simulation frameworks for both network and traffic modeling both for academic and commercial purposes and there are many works in both fields that try to survey and outline the pros and cons of each framework. Here, we try to only look at the most commonly used frameworks in the form of a tabular survey without getting into too many details.

Network simulator is used to create and simulate new models on an arbitrary network by specifying both the behavior of the network nodes and the communication channels. It provides a virtual environment to model the network based on specific criteria and analyzing its performance under different scenarios [171]. The most common network simulators that were developed follow a discrete-event simulation approach simulation happens around an ordered queue of scheduled events. Each event is scheduled for a specific simulation time at which it will be triggered, giving simulation a chance to react to the event.

Table 4.1: An overview of different Traffic Modeling Frameworks

Criteria	SUMO	Paramics	Treiber's Tool	Aimsun	SimTraffic	TRAFVU	Vissim
Availability	Open Source	Commercial	Open Source	Commercial	Commercial	Commercial	Commercial
OS Portability	Windows, Linux & Mac OSX	Windows	Windows, Linux & Mac OSX	Windows	Windows	Windows	Windows
Documentation	Good	Good	Good	Not available freely	Good	Good	Good
Creating Network Flows	Manual Import from OSM Automatic network generation	Automatic network generation	Predefined scenarios using JSON	Manually using GUI	Predefined Scenarios	Predefined Scenarios	Graphical Modeling & Editing Import networks
Creating Vehicle patterns	Flow definitions and turning ratios OD matrices Random route generator	OD matrices	Statistical Distribution	OD Matrices Random route	Automatic trip generation using TripGen	NA	Trip generation using statistical distributions
GUI quality	2D with texture customization	2D and 3D with fine texture customization	2D	2D and 3D (with reasonable quality)	2D (3D available on request)	2D	2D and 3D
Network Scalability	Yes.	Yes but slower 3D simulation	No	Only with full version	Only with full version	Only with full version	Only with full version
Simulation Output (Data and Files)	Network states Loop Detectors Emissions Traffic Lights Vehicle states	Queuing patterns Demand troughs and peaks Speed and Density Journey times Others	NA	20 different viewstyles of statistical information about traffic and events		Graphical Player only	Data collection points Signal control Travel times Delay segments Nodes, Vehicles and Edges
CPU Usage	5-17%	Constant 50%	NA	25-40%	Constant 50%	Constant 50%	NA
Vehicle types	Various vehicles and custom definitions	Various vehicles and custom definitions	Cars and Trucks only	Various vehicles and custom definitions	Various vehicles and custom definitions	Various vehicles and custom definitions	Various vehicles and custom definitions
Public transport and Stops	Yes	Yes (Full version)	No	Yes	NA	Yes (Full version)	Yes
Simulation of pedestrians	Yes	Yes	No	Yes	Yes	NA	Yes
Manual configuration of traffic lights	Yes	NA	No	NA	Yes	NA	Yes

Such a reaction will typically change the state of the simulation and/or trigger new events to be scheduled in turn. After an event has been processed, the simulation will move on to processing the next event in the queue, advancing the current *simulation time* to that of the next event. Depending upon the desired level of granularity, an event might represent anything from a message received by a vehicle, to an electromagnetic wave being picked up by an antenna [175]. Table outlines the features and comparison of different network simulators that are currently in use today.

We have seen that mobility modeling plays a direct and significant role while evaluating the performance of a given V2X RAT. In order to address this issue, ETSI recommended *Manhattan-grid* model that constrained the node movement to move along a grid of possible paths by assuming a downtown Manhattan scenario [51]. However, this also proved to be very simplistic and provided vastly differing results from real world. Further efforts included recording and using real-world mobility traces by equipping vehicles with GPS devices [214, 91] and using these traces in simulations by simply dropping the nodes in the positions given by the GPS coordinates at any time instance. Even though this approach resulted in a more realistic mobility simulation, its use was limited since only a few traces are available and those available do not completely reflect all the vehicles on the road. Additionally, the trace-driven approach is also unsuitable when evaluating V2X applications involving dynamic route selection as it invalidates the pre-computed trace. With the advances in transportation and traffic science, better mobility models were developed that are classified as macroscopic, mesoscopic and microscopic models according to the granularity with which traffic flows are examined. These mobility models promises to lead to the same results as having mobility traces of all vehicles within a network depending upon the level of realism with which they are modeled. Examples of such mobility modeling frameworks include SUMO [104], Vissim [112] etc. that model vehicular mobility at a microscopic level by considering details such as driver imperfections and car following / lane changing models etc. A detailed comparison between the currently available mobility simulators can be found in the table.

Co-simulation of Network and Traffic Simulators by Coupling

Evaluating V2X applications such as dynamic route selection will result in situations where the network and road traffic simulators influence each other's behavior alternatively. Hence, using pre-computed traces (also by means of mobility simulators) would result in its invalidation. In such cases, this loop between road traffic simulation and network traffic simulation needs to be closed. This is achieved by means of *bi-directionally coupled simulation* frameworks that run both the road and network traffic simulators concurrently in alternating phases

1. While the network simulation is running, it sends parameters changes to the road traffic simulation, altering driver behavior or road attributes, and influencing vehicle's routing decisions
2. At regular intervals controlled by the network simulator, the road traffic simulation performs traffic computations that are based on these new parameters and sends vehicle movement updates to the network simulation

Examples of such integrated simulation frameworks include TraNS [114], Vehicles in Network Simulation (VEINS) [176], iTetris [152] etc. However, it is also possible to use other

network simulators and combine them with any available road traffic simulation framework via a unified interface known as Traffic Control Interface (TraCI) [199]. In this work, we adopt this approach and use existing network simulator developed in Matlab for evaluating the system level performance of both the technologies

4.4 Scenarios

The selection of the simulation scenario has a significant impact on the evaluation of a particular V2X application. Depending upon the vehicular networking protocol and application being studied, one has to carefully select and parameterize the scenario configuration. As an example, a protocol that shows a very good performance on a single lane street with sparsely distributed vehicles might fail completely on a 6-lane highway during rush hour due to interference and congestion of the channel. In order to generalize the performance of the selected protocol, it is important that the evaluation be carried out for different road networks and traffic scenarios. For the purpose of evaluating the system level performance, we do not need to consider bidirectionally coupled traffic-network simulation since we are only interested in capacity analysis and edge performance rather than traffic optimization. Hence road networks with predefined traffic scenarios should suffice for our analysis. In this section, we outline the different road networks used along with the corresponding traffic models. These scenarios are mainly based on the 3GPP recommendations for evaluating V2X applications [185]

Scenario 1 - ETSI Highway

This is the simplest scenario where cars can drive in east-west direction on multiple lanes. If needed, lanes can be combined to support bi-directional traffic. Cars can be spawned from either end of the road network with a given flow rate or using a Poisson distribution. Depending upon the level of detail required, different flow rates for different lanes, different vehicle types (e.g., cars, trucks), different car following/lane changing models as well as overtaking maneuvers needs to be considered - thus complicating what looks a simplistic scenario. In a way, this models the typical *highway* scenario and is applicable for many V2X applications and would help in investigating the reliability/availability of the V2X application under high speeds/mobility.

For this scenario, we constructed a highway traffic model as illustrated in Figure 4.5. The traffic is modeled inline with ETSI specifications in [185]. A road length of 2 km is considered with 3 lanes in each direction (Figure 4.5) with an inter-lane spacing of 4 m. All the vehicles are assumed to have the same acceleration, deceleration and speed characteristics. Moreover, all the vehicles also maintain a constant headway space to the vehicles in front and rear. This is the most simplistic scenario and is considered to be the benchmark for performance comparison. The total number of vehicles on the highway is then calculated as

$$n_v = \frac{l_h}{(l_v + h_v))} * n_l \tag{4.1}$$

Where l_h is the length of the considered road section, l_v is the length of the vehicle, h_v is the inter-vehicle distance that is calculated as 2.5 s * s_v where s_v is the absolute vehicle

Table 4.2: Simulation Parameters for ETSI Highway

Parameter	C-V2X	ITS-G5
Road	Length = 2 km, Width = 4m. Number of Lanes (per direction) = 3	
Traffic	3 Traffic Types 1. Density (veh/km) = [70, 124, 247] 2. Speed (km/h) = [250, 140, 70]	
Channel Models	ITU-EVA (SIMO), H-LOS, H-NLOS (SISO)	
Packet Size	300 bytes (600 bytes with retransmission)	300 bytes
MCS Schemes	7, 10, 14, 18	2, 3, 4, 5
SINR Thresholds	ITU-EVA (No retrans) = [3.87, 5.63, 9.66, 15.78] ITU-EVA (Retrans) = [-0.55, 0.47, 4.62, 8.27] H-LOS (No retrans) = [5.19, 6.94, 10.76, 17.11] H-LOS (Retrans) = [0.56, 1.56, 5.52, 9.01] H-NLOS (No retrans) = [7.02, 9.36, 15.30, NA] H-NLOS (Retrans) = [1.92, 3.05, 8.14, 12.9]	ITU-EVA = [5.35, 7.65, 10.43, 13.97] H-LOS = [5.94, 8.42, 10.99, 14.80] H-NLOS = [8.81, 11.89, 14.92, 18.8]

speed and n_l is the total number of lanes. Subsequently, the vehicle density ρ_v can be calculated as

$$\rho_v = n_v * \frac{l_h}{1000} \tag{4.2}$$

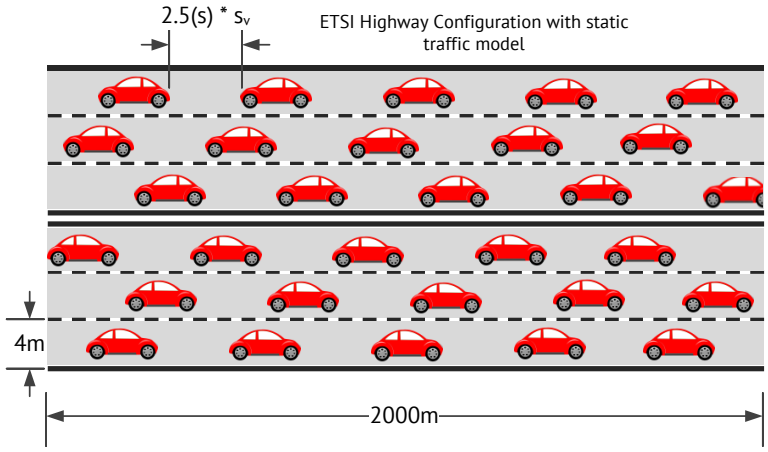


Figure 4.5: Highway Scenario as per ETSI specification

Scenario 2 - Manhattan Grid

Manhattan grid is one of the most frequently used scenarios to evaluate the performance of V2X protocols in dense urban scenarios. It models the urban vehicular traffic by using a grid topology with organized streets and intersections. The vehicles move either horizontally or vertically on the grid. At an intersection of a horizontal and a vertical street, the mobile node can turn left, right or go straight with certain probability. The probability of going straight is 0.5 and the probability of going left/right is set at 0.25. By means of creating dense vehicle distributions over multiple intersections, KPIs such as latency, interference, Packet Reception Ratio (PRR) etc. can be studied. Figure 4.6 illustrates this scenario as per ETSI specifications [185]. This road network can be parameterized by number and length of each grid in both x and y directions, number of lanes per grid n_l, lane width as well as the width of the sidewalk.

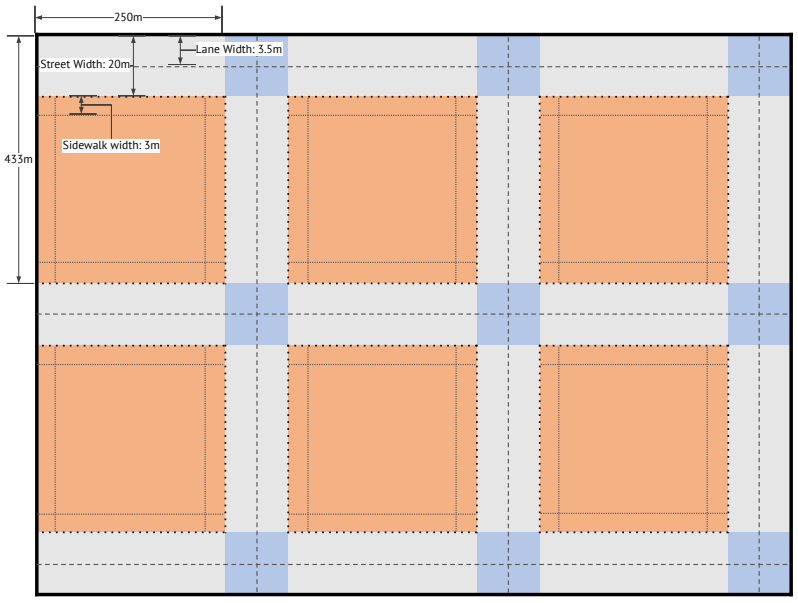
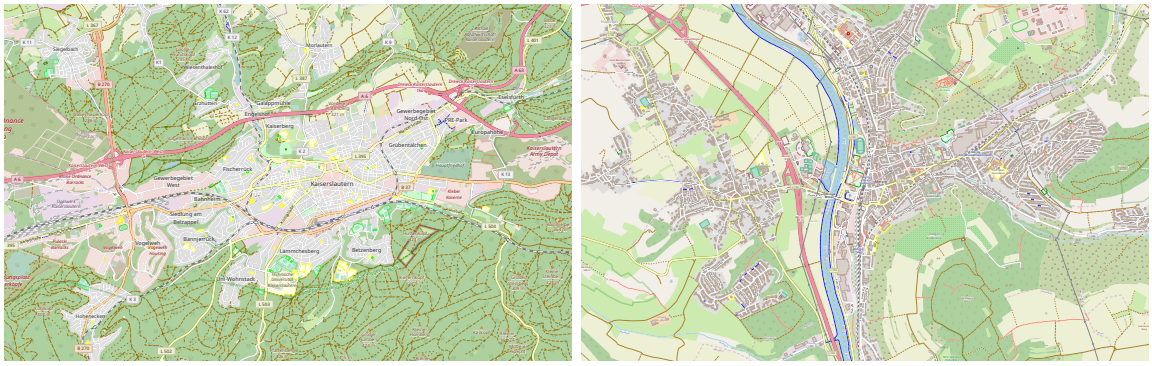


Figure 4.6: Illustration of Manhattan Grid

Table 4.3: Simulation Parameters for Manhattan Grid Model

Parameter	C-V2X	ITS-G5
Road	Manhattan Grid with Grid Size = 433 X 250 m Total Simulation Area = 1299 X 750 m (3 blocks) Number of Lanes (per direction) = 2, Lane width=3.5 m	
Traffic	3 Traffic Types 1. Speed (km/h) = [15, 60, 120] 2. Total vehicles in network = [2361, 591, 296]	
Channel Models	ITU-EVA (SIMO), UA-LOS, U-NLOS (SISO)	
Packet Size	300 bytes (600 bytes with retransmission)	300 bytes
MCS Schemes	7, 10, 14, 18	2, 3, 4, 5
SINR Thresholds	ITU-EVA (No retrans) = [3.87, 5.63, 9.66, 15.78] ITU-EVA (Retrans) = [-0.55, 0.47, 4.62, 8.27] UA-LOS (No retrans) = [6.48, 8.64, 13.07, 19.32] UA-LOS (Retrans) = [1.93, 3.18, 8.06, 11.72] U-NLOS (No retrans) = [6.43, 8.52, 14.26, 24.32] U-NLOS (Retrans) = [1.74, 2.88, 8.04, 13.42]	ITU-EVA = [5.35, 7.65, 10.43, 13.97] UA-LOS = [6.68, 9.37, 11.94, 15.79] U-NLOS = [8.0, 10.93, 13.99, 17.68]

As seen in Figure 4.6, ETSI specifies a block size of $433\,\mathrm{m} \times 250\,\mathrm{m}$ with bi-directional traffic and 2 lanes of width 3.5 m per direction. It also specifies a sidewalk (for pedestrians) with a width of 3 m. Therefore, the total width of the road section (with sidewalks) between each building is $4 \times 3.5\,\mathrm{m} + 2 \times 3\,\mathrm{m} = 20\,\mathrm{m}$. For generation of traffic, the same approach as used in highway scenario is assumed, i.e., constant inter-vehicle distance of 2.5 s * s_v where s_v ranges between 15 km/h to 120 km/h. For our simulation, we use the same grid parameters as per the ETSI specification. The traffic modeling, on the other hand is done in a more realistic way by keeping the average density of vehicles same as the the ETSI specification (using Equation 4.2), but with a non-uniform distribution of vehicles. Additionally, vehicles also accumulate at traffic junctions during red lights which leads to a situation where there is a high vehicle density at junctions. The simulation parameters are outlined in the Table 4.3



(a) Kaiserslautern Map (b) Merzig Map

Figure 4.7: The Maps of the selected scenarios (Source:OSM)

Scenario 3 - Real world maps for urban scenarios with realistic traffic

In order to achieve the highest level of realism for traffic modeling, it becomes necessary to use real world map information to prepare a scenario for an urban environment. Geographical data is provided by commercial map data sources such as Google or Here maps. Recently, crowd sourced data providers like OpenStreetMap (OSM) have been increasingly used by the research community. Depending upon the location, geographical data provided by OSM is even more accurate than that provided by commercial services. Another advantage of using OSM data is the availability of rich metadata corresponding to the region of interest that includes data about the buildings and their shapefiles, traffic light cycles, information about public transportation etc. This metadata can in turn be used to model effects such as shadowing due to buildings in a realistic way.

An important aspect of using real world scenarios is the modeling of realistic traffic. It is a very effort intensive process with an entire engineering discipline dealing with the topic. Even if real world maps are used for road network generation, it would be useless if the traffic is not modeled realistically. Traffic modeling frameworks like SUMO provide functionality (by means of driver models, vehicle profiles, randomness etc.) to model vehicular traffic realistically. But these functions still far short in capturing the mobility needs of people in a realistic way. An alternative approach is to use real world traffic data from sources such as traffic detectors placed at different road sections, real world mobility traces etc. and then using this data to recreate similar mobility patterns. In the latter case, we would still be left with the problem of invalidated traffic model when evaluating dynamic routing based V2X applications. But since we are dealing with the system level simulation without any special focus on a particular use case, precomputed / trace based simulation would suffice for our evaluation.

The next step is to find and select the available real world scenarios with realistic traffic models for the simulation. SUMO already provides such data sources and scenarios for different cities such as Bologna, Cologne, Luxembourg, Monaco etc. For our system level simulations, we selected Merzig and Kaiserslautern as reference scenarios since they capture the mobility patterns of small and medium sized cities respectively. The simulation parameters for both these scenarios are outlined in

4.5 Results and Conclusions

Figure 4.8 shows the PRR of both ITS-G5 and C-V2X for the highway scenario for the ITU-EVA channel for for different speeds (by varying the inter vehicular distance as explained in section 4.4) In general, it can be seen that C-V2X outperforms ITS-G5 for all considered scenarios. C-V2X exhibits a range gain of atleast 100 m for low coding schemes and almost 200 m for high coding schemes. It can be noted that there is a high PRR for higher speeds as compared to lower speeds which is due to the fact that there are few vehicles in the scenario due to the high inter-vehicular distance thereby resulting in lower interference. This is also validated by the CDF plot of the CBR (Figure 4.9) where the lower speed scenarios have a very high CBR approaching close to 1.0 for the case of C-V2X with retransmission.

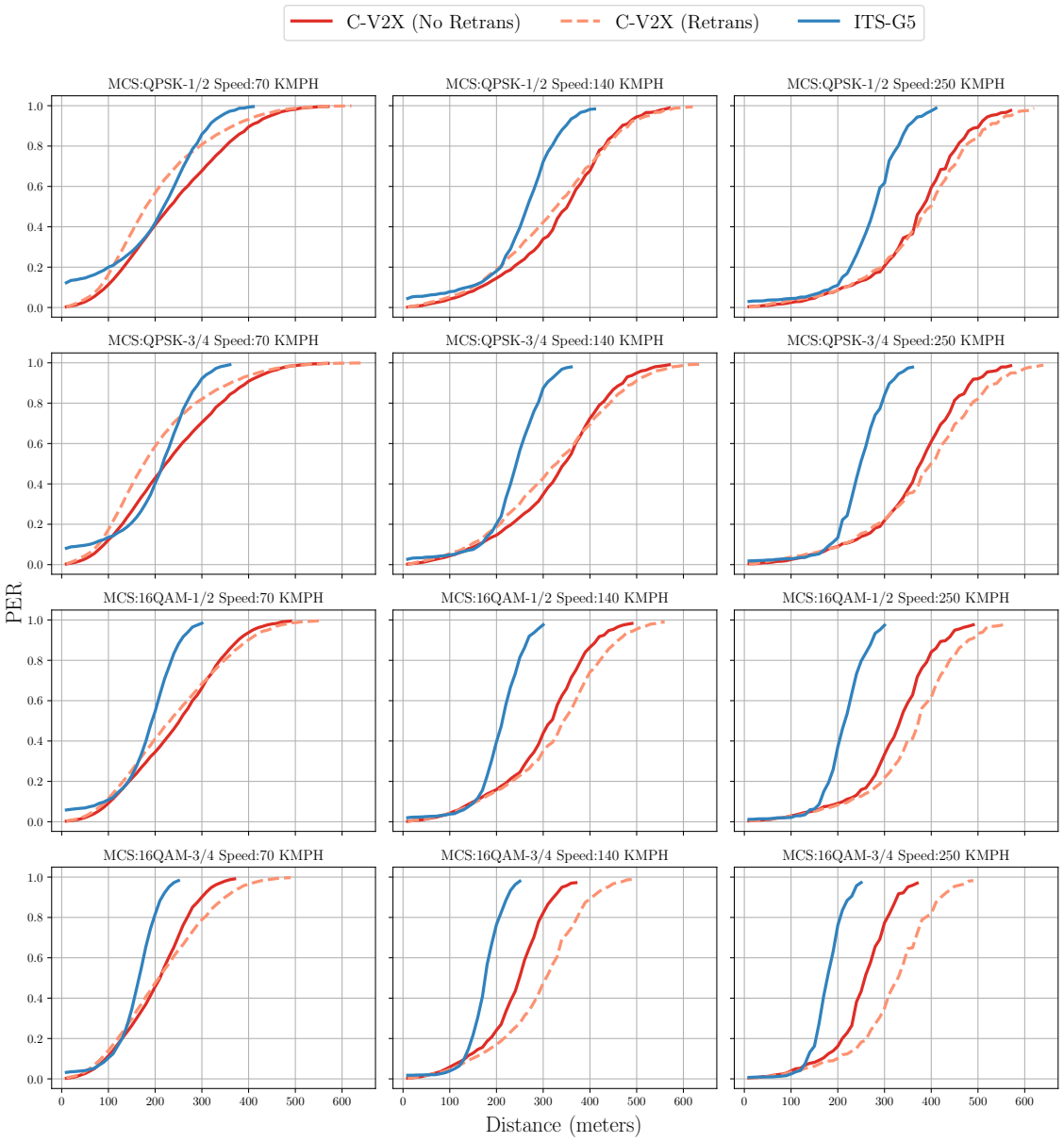


Figure 4.8: PER for ETSI Highway Scenario for ITU EVA Channel for different speeds

A second notable point is the variation in retransmission gain when using C-V2X.Due to the high CBR ast at low vehicular speeds, there is a high amount of interference thereby resulting in degraded overall PRR at lower MCS. However, retransmission gain is visible at higher MCS and lower vehicular density. In fact, the retransmission gain shows an increasing trend with the MCS order.

The performance of both the technologies for HLOS channel is similar to the EVA channel as seen from Figure 4.10. C-V2X shows PRR of almost 100 % at distances less than 50 m whereas ITS-G5 approaches a maximum to 90 %. The retransmission gain of C-V2X is also visible at high MCS schemes with a range gain of almost 100 m. The CBR (Figure 4.11) also shows the same trend as that of the EVA channel. Finally, Figure 4.12 shows the

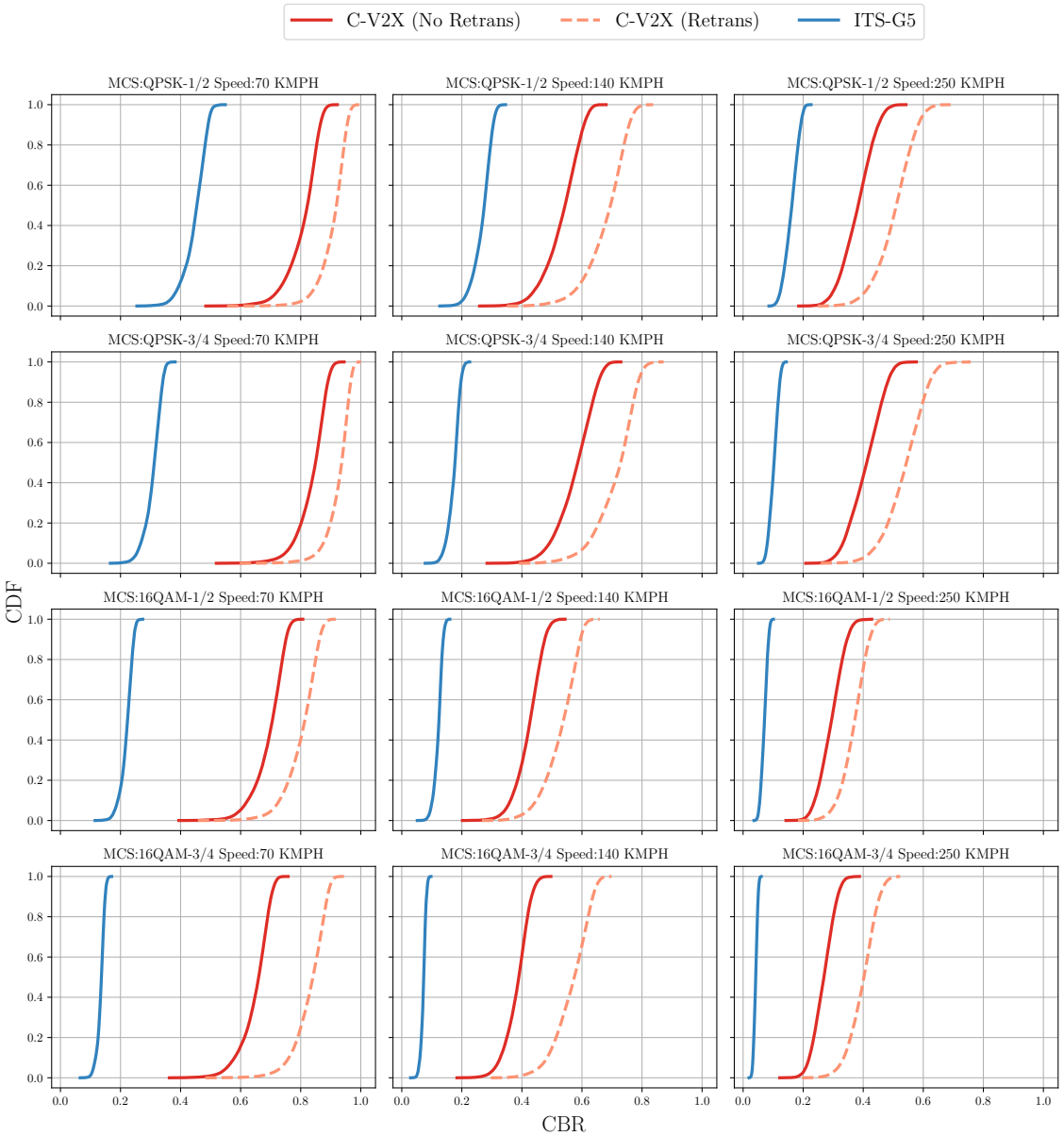


Figure 4.9: CDF of CBR for ETSI Highway Scenario for ITU EVA Channel for different speeds

performance comparison of both the technologies for HNLOS channel. In general, it can be seen that due to the shadowing effect, the range is decreased in general for all constellations.

This is also seen in the form of steeper degradation of PRR of all the schemes. A notable difference here is the absence of C-V2X results for QAM16 scheme where the SNR threshold is indeterminate from the link level simulations. When retransmissions are enabled, this brings huge benefit as seen from the large gap between ITS-G5 and C-V2X.

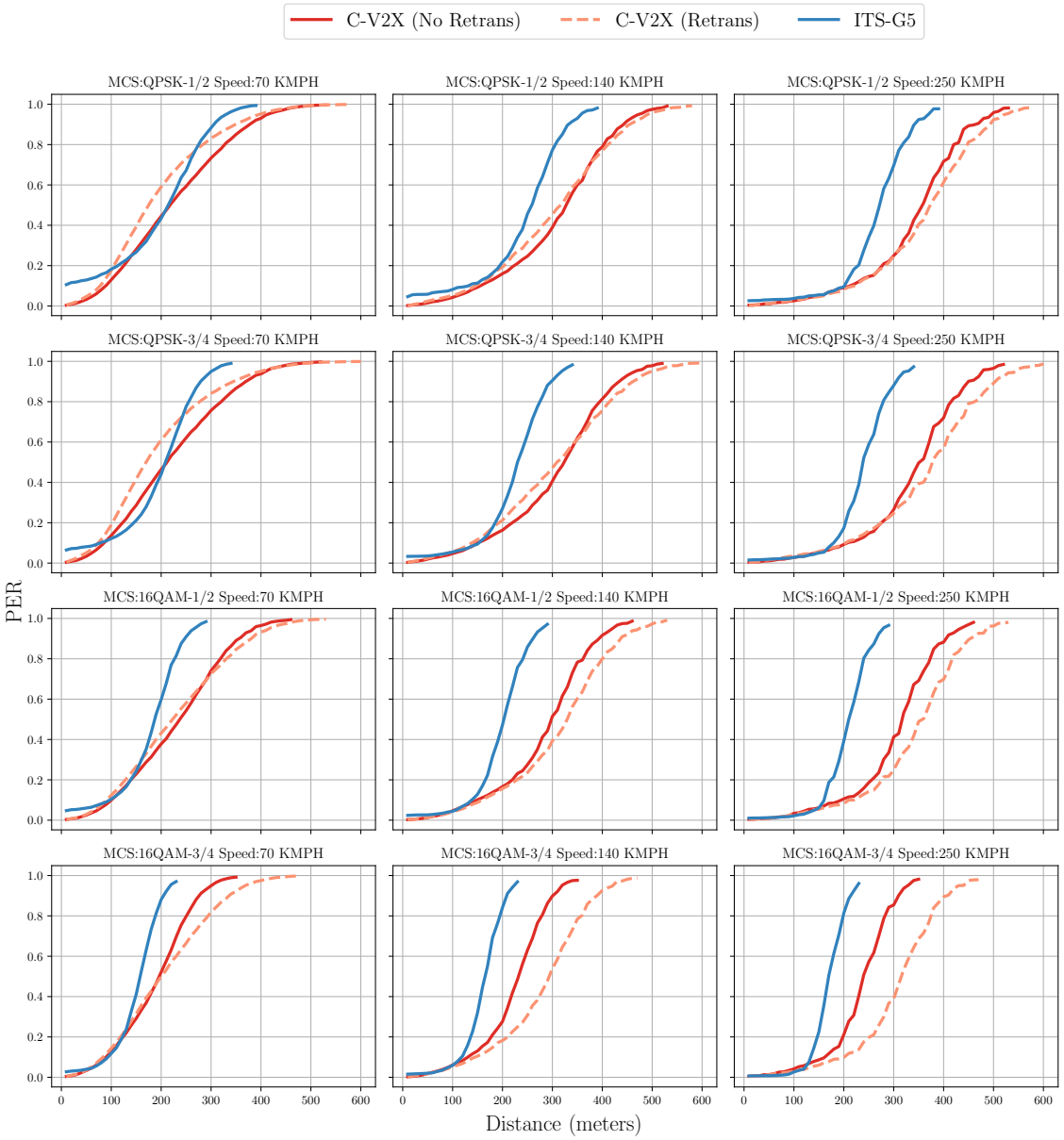


Figure 4.10: PER for ETSI Highway Scenario for HLOS Channel for different speeds

For the Manhattan grid scenario, it can be said in general that the performance of both the technologies exhibit similar behavior as seen in Figure 4.14. This applies to all the MCS schemes. The retransmission gain, as seen from previous results is only visible at high MCS schemes and at high distances where it improves the overall PRR. In Figure 4.15, the CDF of CBR of both the technologies is highlighted. It clearly shows the efficacy of SPSS over the traditional CSMA/CA with C-V2X doubling (or tripling) the channel usage as compared to ITS-G5.

As mentioned in the previous section, the two scenarios that were simulated as part of

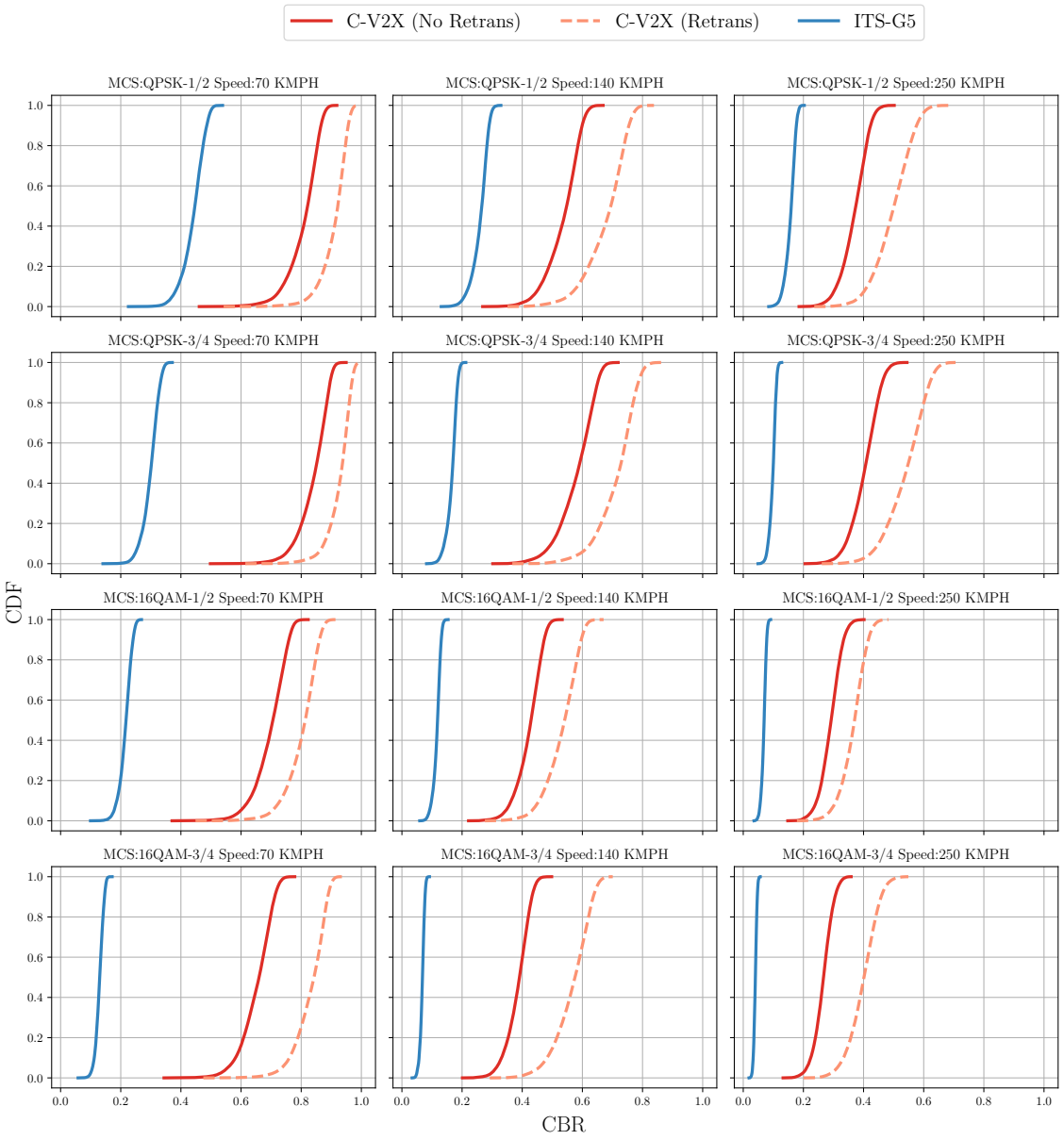


Figure 4.11: PER for ETSI Highway Scenario for HLOS Channel

real world scenarios are Kaiserslautern and Merzig corresponding to small city and rural area respectively. Figure 4.16 shows the PER for Kaiserslautern for all the three channel models. The higher range benefit of C-V2X can be clearly seen from these results with C-V2X delivering packets beyond 500 m distance whereas ITS-G5 shows no packet deliveries beyond 400 m. Additionally, the range of ITS-G5 degrades more with QPSK-3/4 and 16QAM-1/2 schemes compared with C-V2X. However, the performance of both the technologies remain quasi equal below a distance of 200 m. Beyong 200 m, the PER of ITS-G5 increases steeply in relation to C-V2X. Another notable point is that the retransmission gain in C-V2X is very low with lower MCS and increases with the MCS order. Figure 4.17 shows the CBR for the Kaiserslautern scenario. Inline with previous results, it can be seen that C-V2X has a higher channel utilization rate than ITS-G5 and this gap widens as the MCS order increases.

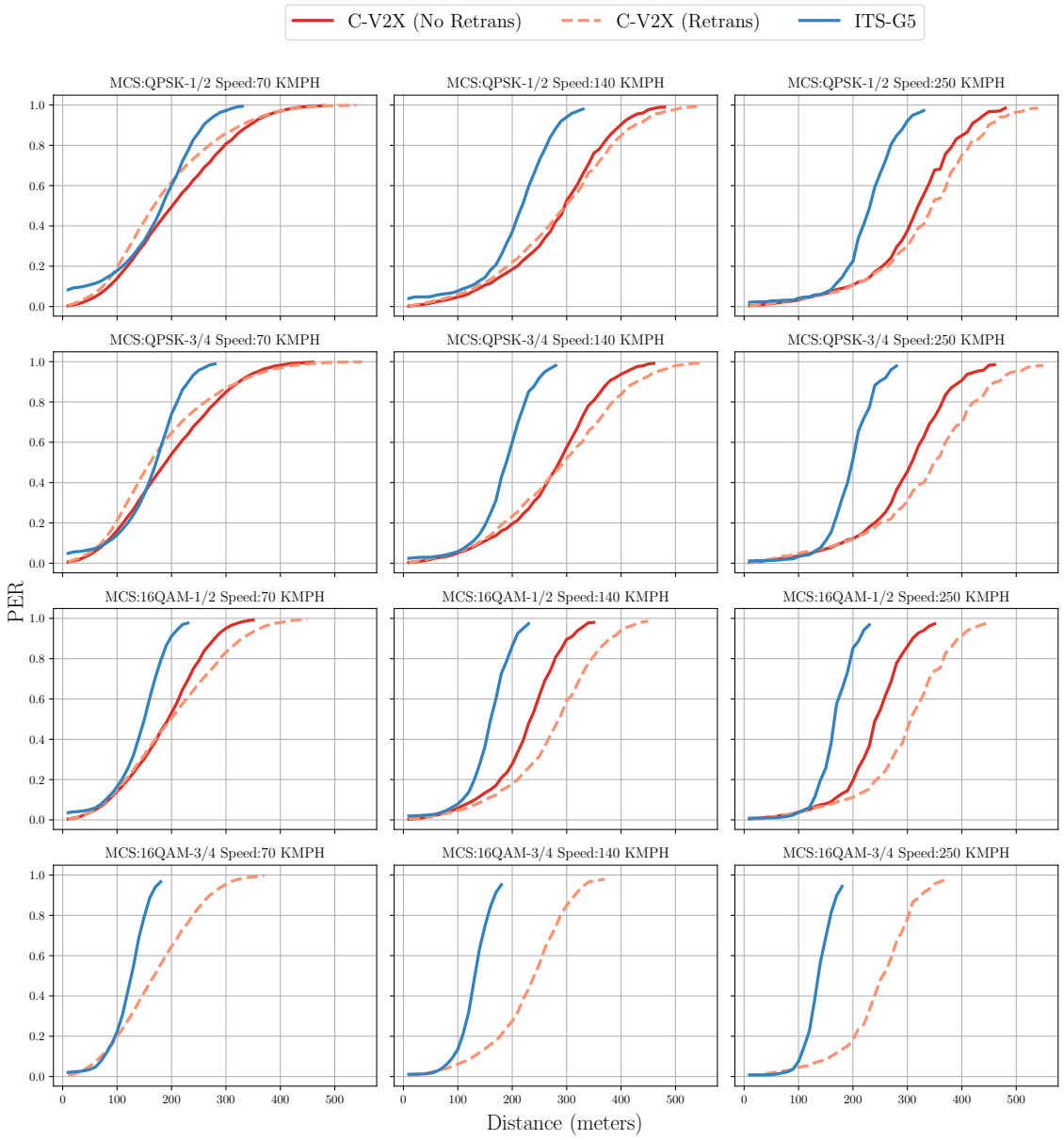


Figure 4.12: PER for ETSI Highway Scenario for HNLOS Channel for different speeds

4.6 Summary & Conclusions

In this chapter, the system level simulations were performed and comparisons were drawn for MAC schemes of both C-V2X and ITS-G5 namely, SPSS and CSMA/CA respectively. For this purpose, the bidirectional simulation approach was used where the mobility simulator (SUMO) was used in conjunction with the network simulator. The mobility simulator generates vehicular traffic traces and the network simulator calculates the performance of MAC schemes with the given vehicle positions. Multiple scenarios were considered for the overall analysis in order to generalize the performance of the MAC schemes. These scenarios include the ones specified by ETSI such as Highway scenario and the Manhattan grid scenario to custom real world maps. These include the cities of Kaiserslautern and Merzig which fall under the category of small city and rural areas respectively.

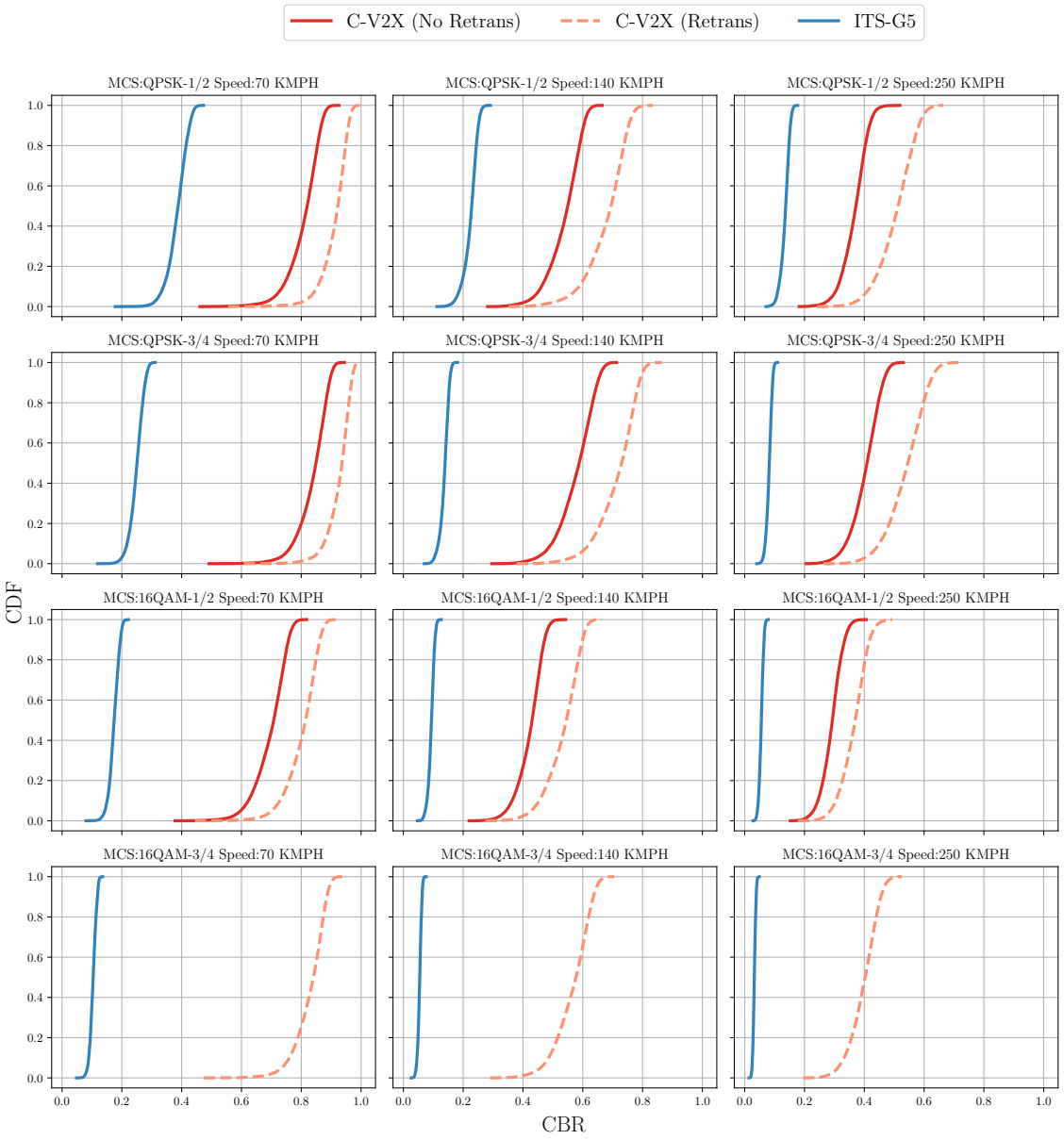


Figure 4.13: PER for ETSI Highway Scenario for HNLOS Channel

Overall, it can be seen that C-V2X has a higher range than ITS-G5 inline with the specification.This can be seen in the form of PER where ITS-G5 maxes out before C-V2X. In terms of distance, this translates to a range gain of almost 100 m for QPSK coding schemes and almost 200 m for 16QAM schemes. C-V2X also makes use of spectrum efficiently compared to ITS-G5 which can be seen in the form of higher CBR. The retransmission gain with C-V2X is significant at higher MCS, lower vehicular densities and higher speeds.

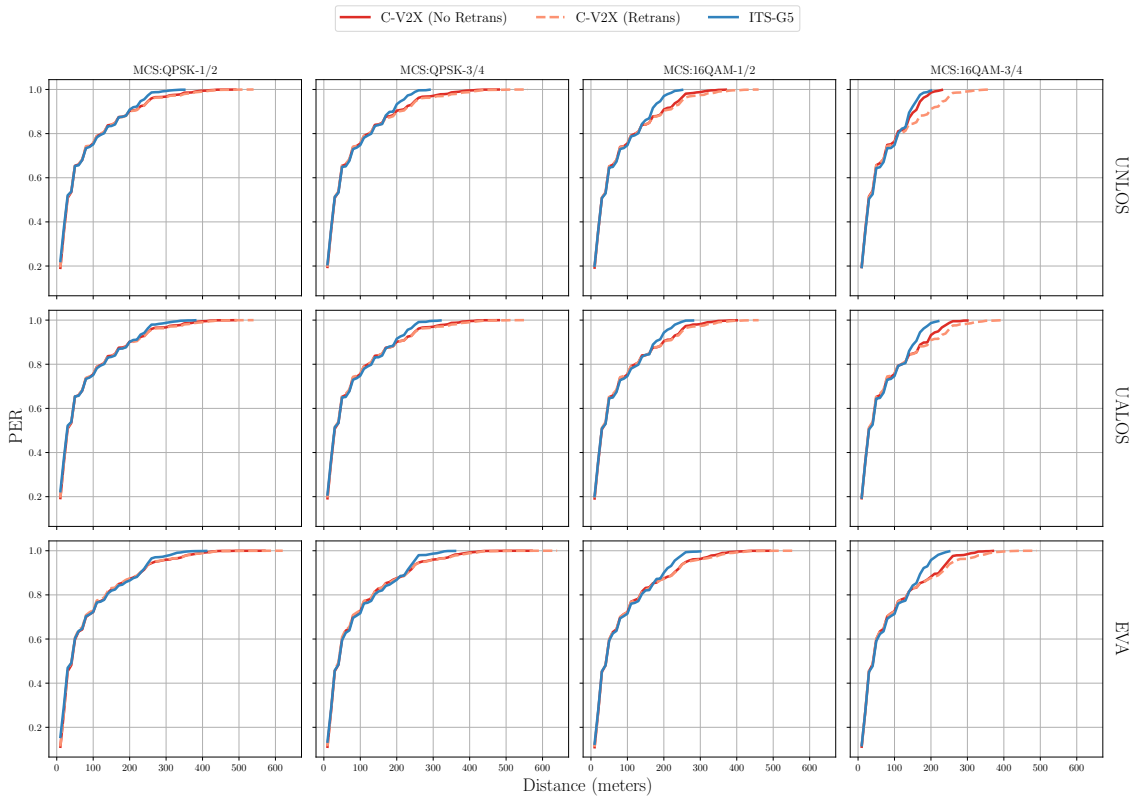


Figure 4.14: PER for Manhattan Grid for all channel models

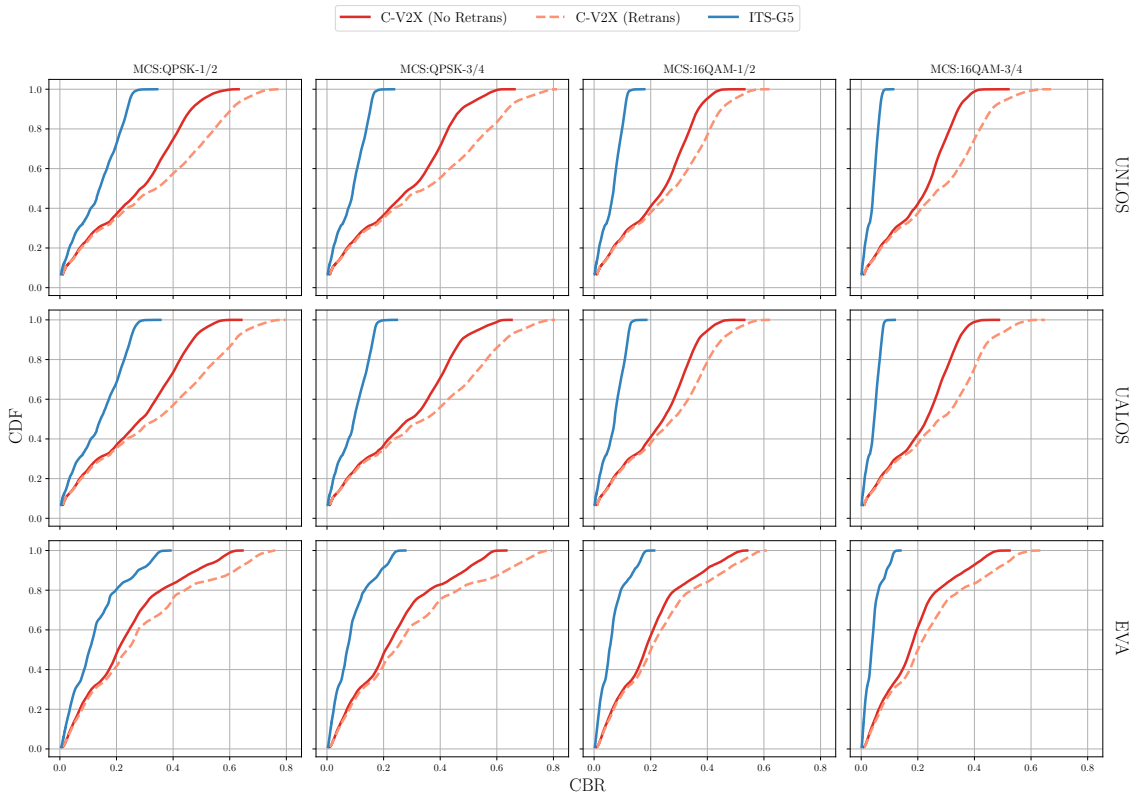


Figure 4.15: CBR for Manhattan Grid for all channel models

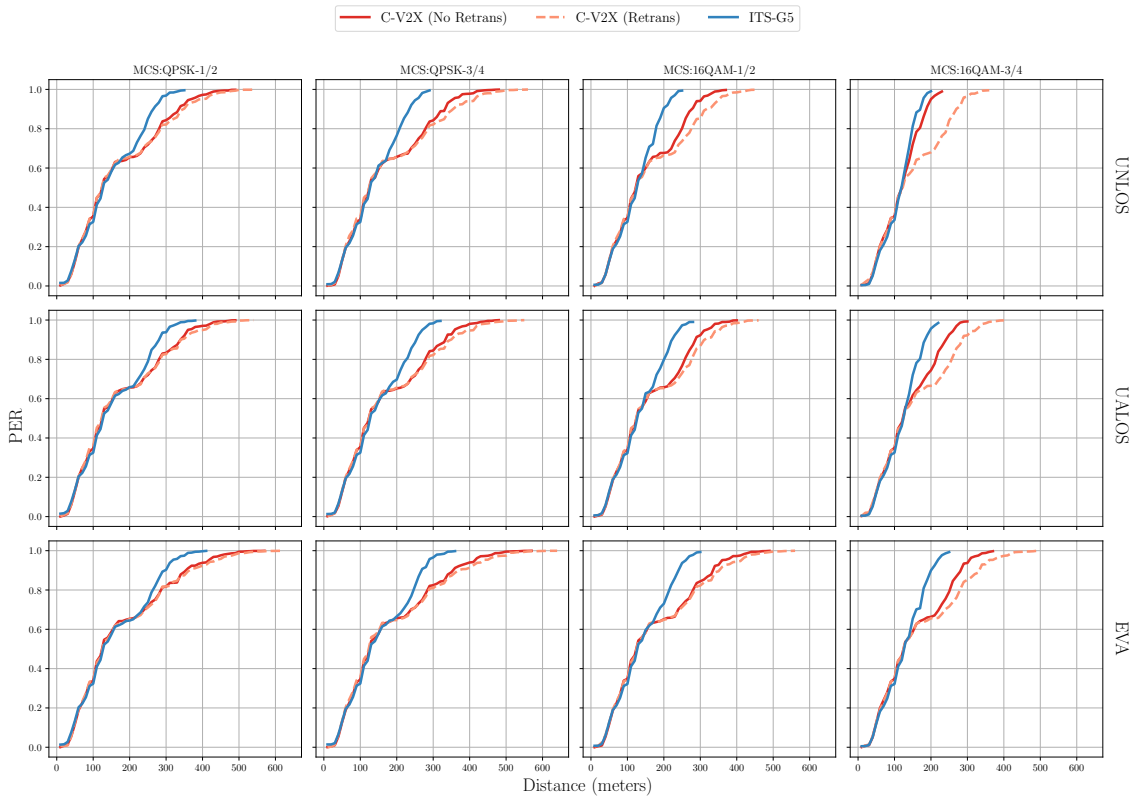


Figure 4.16: PER for Kaiserslautern for all channel models

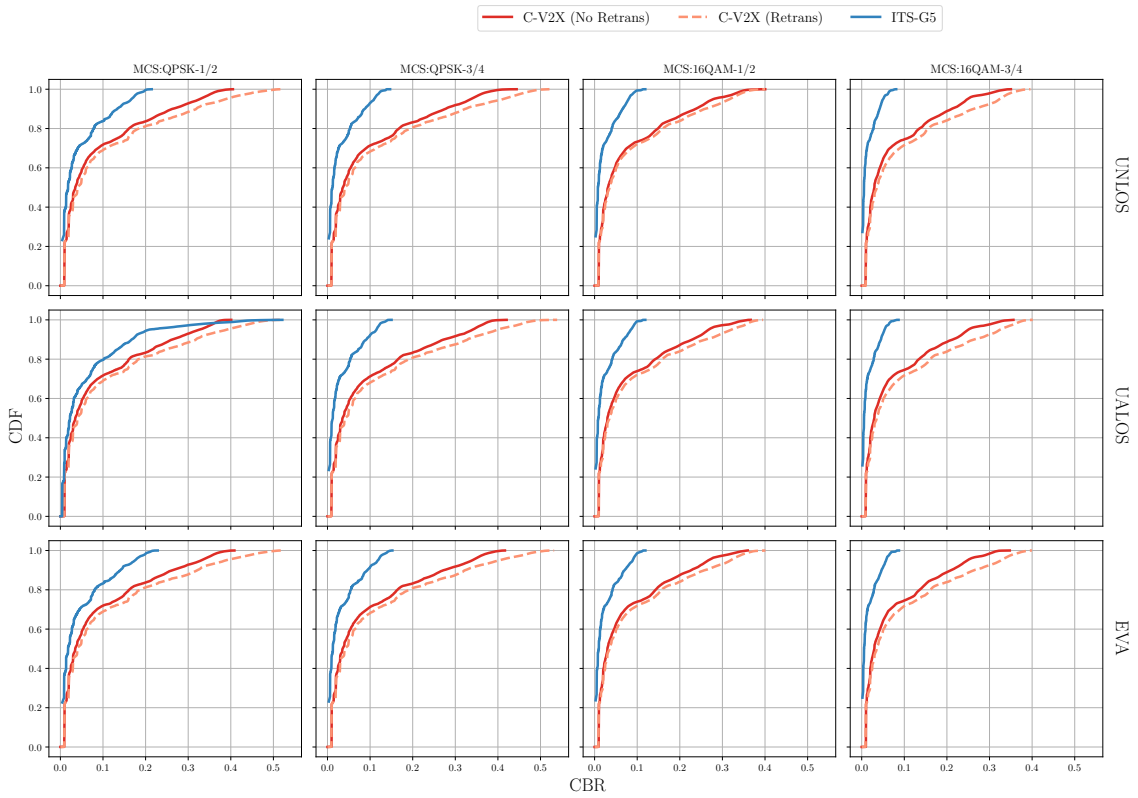


Figure 4.17: CBR for Kaiserslautern for all channel models

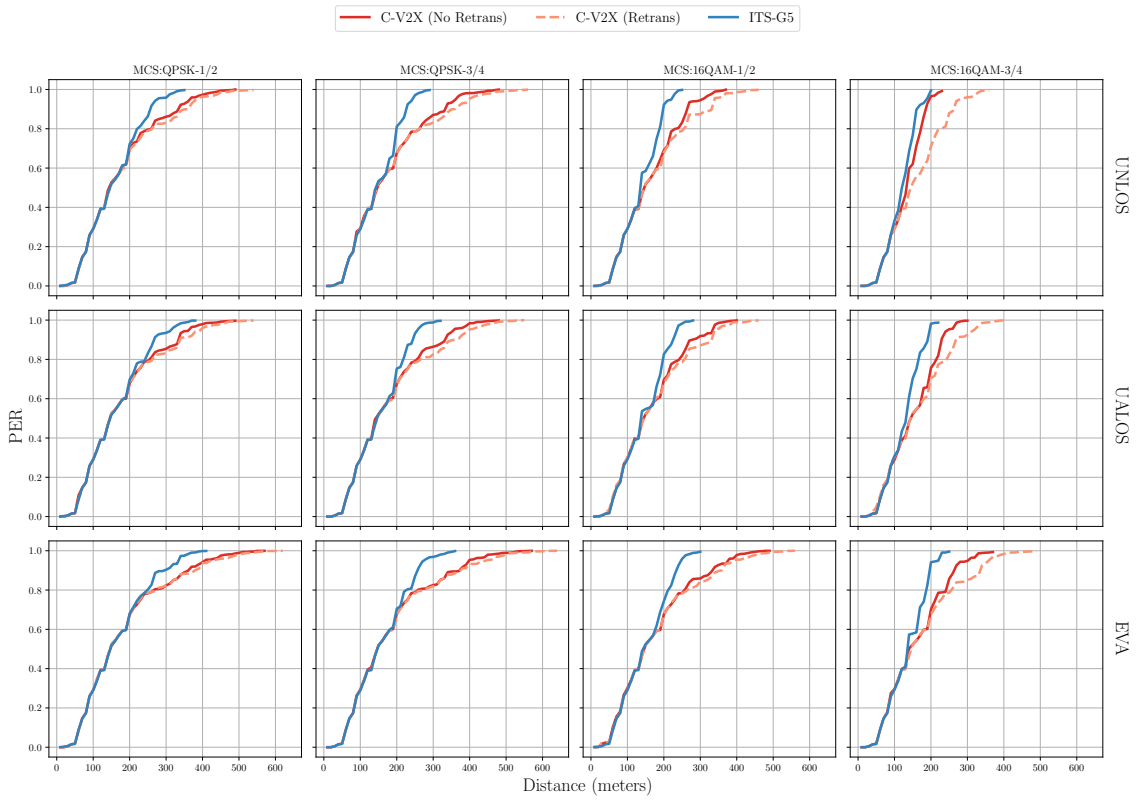


Figure 4.18: PER for Merzig for all channel models

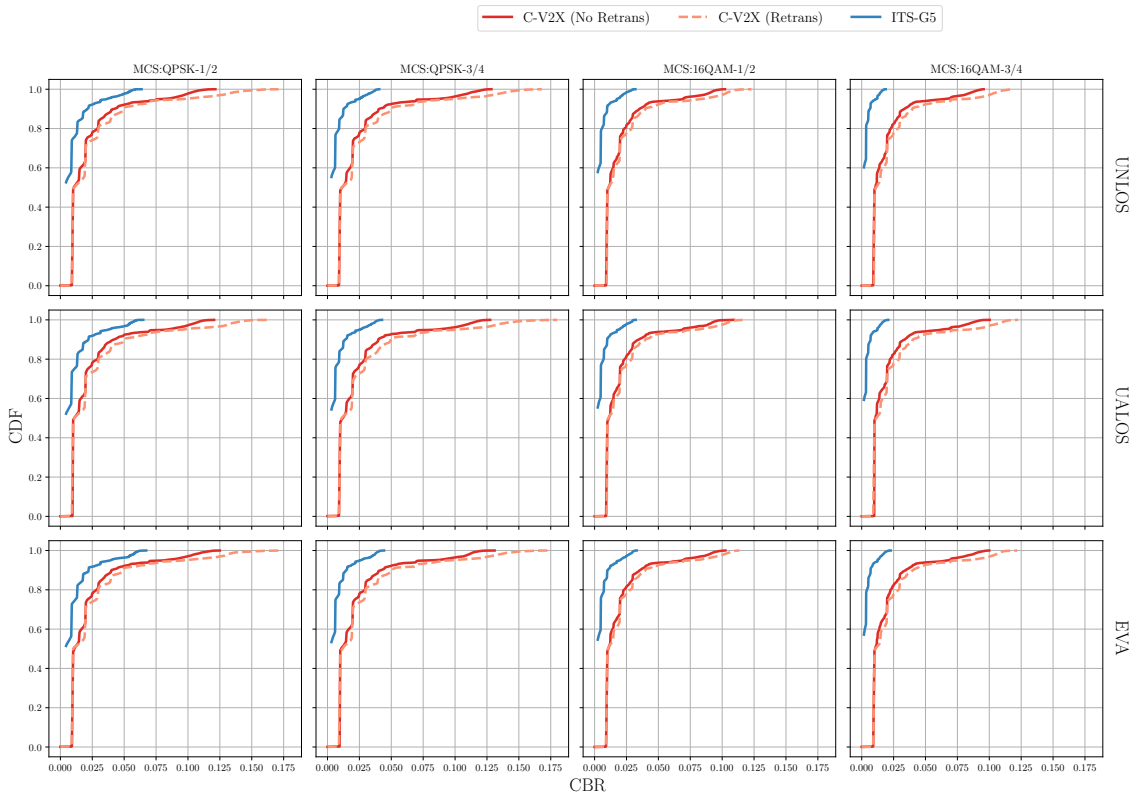


Figure 4.19: CBR for Merzig for all channel models

Part II

AI assisted PHY

5 Machine Learning for Wireless Communications

> Machine learning is the study of computer algorithms that allow computer programs to automatically improve through experience
>
> *(Tom M. Mitchell)*
>
> Artificial intelligence is the science and engineering of making computers behave in ways that, until recently, we thought required human intelligence
>
> *(Andrew Moore)*

Our society is at the cusp of a digital revolution, with a dramatic increase of both internet users and connected devices. The fifth generation (5G) mobile network, that is currently readied for roll out is envisaged to support diverse use cases that span a multitude of verticals such as Industry 4.0, IoT, V2X communication, smart grid monitoring etc. Meanwhile, the research community is already focusing on the development of beyond-5G solutions and the 2030 era. It is envisaged that AI and ML will play a pivotal role in both link and system level solutions of the future generation wireless networks.

Traditional Wireless Network optimization was done by means of acquired domain knowledge in the form of mathematical models / heuristics and designing solutions / algorithms based on this knowledge. This approach, though perfected over the years, will be insufficient when dealing with future hyper-connected society and anticipated services such as autonomous driving, eHealth, Industrial Communication etc. This can be due to *model deficit*, where no physics-based mathematical models exist for the problem due to insufficient domain knowledge or due to *algorithm deficit*, where even though a sound mathematical model is available, the existing algorithms optimized on the basis of this model are too complex to be implemented [168]. Moreover, due to static modelling on predefined scenarios, algorithms designed using this approach cannot react to dynamic data demands, connectivity problems and/or hardware failures. Self Organizing Networks (SONs) aim to address this problem by adding functions relating to network / resource reconfigurability, automating operations and management. However, they are only limited to specific RAN applications without providing a true end-to-end solution.

Based on the points mentioned above, it can be said that the future networks need to have intelligence that is spread throughout the infrastructure. In this regard, a promising solution comes in the form of AI/ML where, instead of relying on domain knowledge, a learning flow is used by means of using sufficiently large examples of desired behavior for the algorithm of interest. Such ML algorithms have been applied to various domains such as computer vision, natural language processing, social network filtering, drug design etc. where they have produced results comparable to and in some cases superior to human experts.

5.1 A brief introduction to AI & ML

Many works use the terms ML and AI interchangeably, a trend that has some historical significance. The term *artificial intelligence* came to inception in 1956 by a group of researchers, including Allen Newell and Herbert A. Simon. During the early decades, it attracted a lot of hype to the extent that many researchers concurred that human-level AI is just around the corner. However, this hype within industry and academia slowly faded away that resulted in the *AI Winter* in the 80's and 90's where the funding and interest in the field subsided considerably. This evidently led to organizations to seperate themselves with the term AI, which had become synonymous with unsubstantiated hype. Concurrently, topics such as big data, predictive analytics, machine learning have made big progress and started to find widespread use in different domains. Advances in ML through techniques such as deep learning, neural networks coupled with the availability of powerful cheap hardware (Graphical Processing Units (GPUs)) renewed the interest and facilitated a gearshift back to AI.

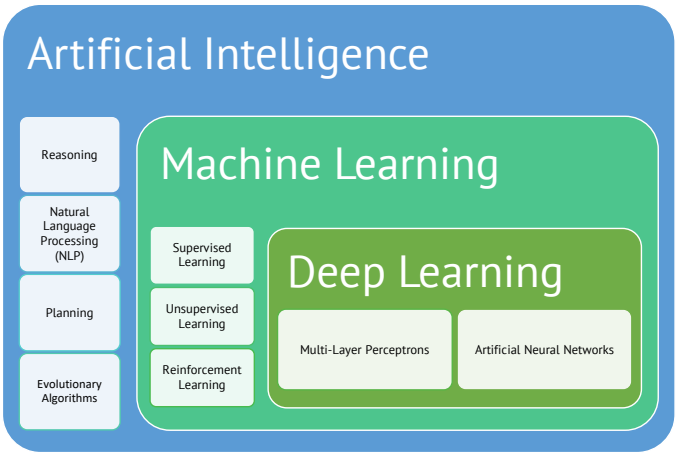


Figure 5.1: Relation between AI, ML and DL

In its most basic form, ML is a form of AI that enables a system to learn from data rather than through explicit programming. It uses a variety of algorithms that iteratively learn from data to improve, describe data, and predict outcomes. AI, on other hand is more broader in scope that essentially tries to mimic human behavior in terms of rationalizing and decision making. However, the term AI is aspirational, a moving target based on those capabilities that humans possess but which machines do not. The relation between both these concepts is clearly illustrated in Figure 5.1. In this thesis, we focus on the ML part where we try to investigate various techniques for use in wireless communication networks.

Mathematically, any ML algorithm learns the execution of a particular task T, maintaining a specific performance metric M, based on exploiting its experience E[92]. In contrast to classical engineering flow where an algorithm is developed and optimized using acquired domain knowledge and and an underlying mathematical description of the problem at hand, ML uses subsets of data to generate an algorithm. It essentially replaces the task of acquiring domain knowledge with the potentially easier task of collecting a sufficiently large number of examples of desired behaviour for the algorithm of interest. These examples constitute the training set and are fed to a learning algorithm to produce a trained model that carries out the desired task. Learning is made possible by the choice of a set of possible

states, also known as the hypothesis class, from which the learning algorithm makes a selection during training. Finally, learning algorithms are based on the optimization of a performance criterion that measures how well the selected model matches the available data.

Depending upon the problem being addressed, ML techniques can be broadly classified into four categories

Supervised Learning In supervised learning, the training set consists of pairs of input and desired output (labels), and the goal is that of learning a mapping between input and output space. When the label is continuous, it is a regression; when the data comes from a finite set of values, it known as classification

Unsupervised Learning Unsupervised learning is best suited when the problem requires a massive amount of data that is unlabeled. Unsupervised learning generally aims at discovering properties of the mechanism generating the data. Examples include Classification and dimensionality reduction techniques

Reinforcement Learning Reinforcement learning is a behavioral learning model. The algorithm receives feedback from the analysis of the data so the user is guided to the best outcome. Reinforcement learning differs from other types of supervised learning because the system is not trained with the sample data set. Rather, the system learns through trial and error. Therefore, a sequence of successful decisions will result in the process being *reinforced* because it best solves the problem at hand.

ANNs and DL DL is a specific method of machine learning that incorporates ANNs in successive layers in order to learn from data in an iterative manner. A neural network consists of three or more layers: an input layer, one or many hidden layers, and an output layer. Data is ingested through the input layer. Then the data is modified in the hidden layer and the output layers based on the weights applied to these nodes. The typical neural network may consist of thousands or even millions of simple processing nodes that are densely interconnected. The term deep learning is used when there are multiple hidden layers within a neural network. Using an iterative approach, a neural network continuously adjusts and makes inferences until a specific stopping point is reached.

Many ML algorithms for different functions have been proposed till date and the list is continually growing with each passing day. Some of the most commonly used models are grouped according to their function and are outlined in the Figure 5.2.

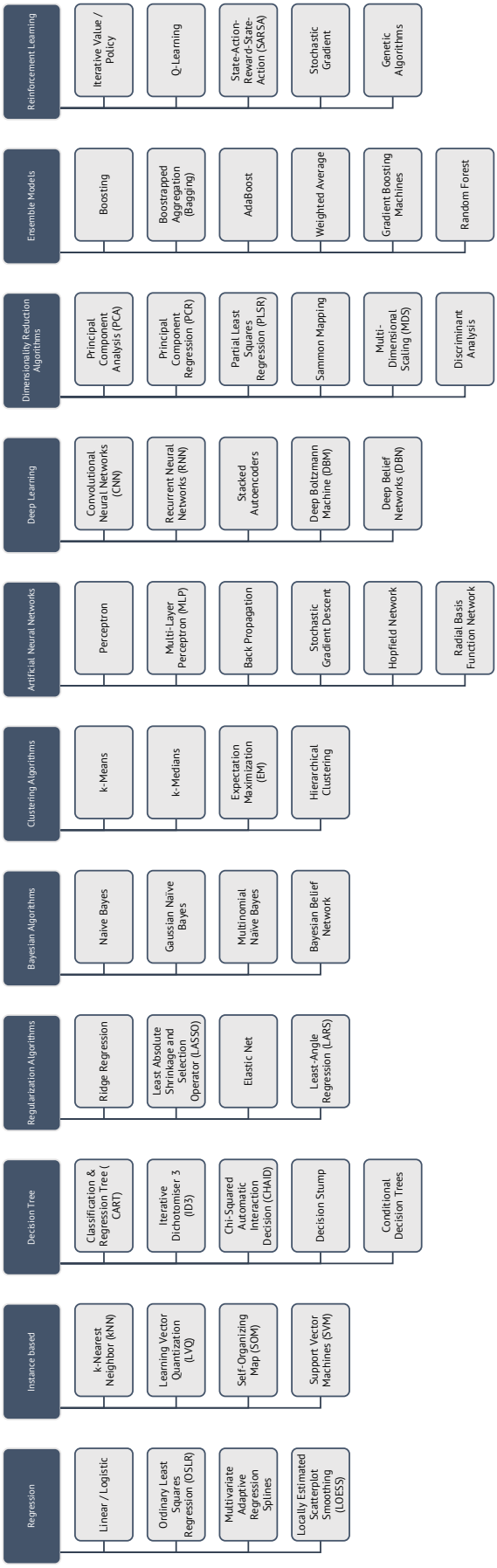


Figure 5.2: Taxonomy of ML techniques grouped by function

5.2 Related State of the Art

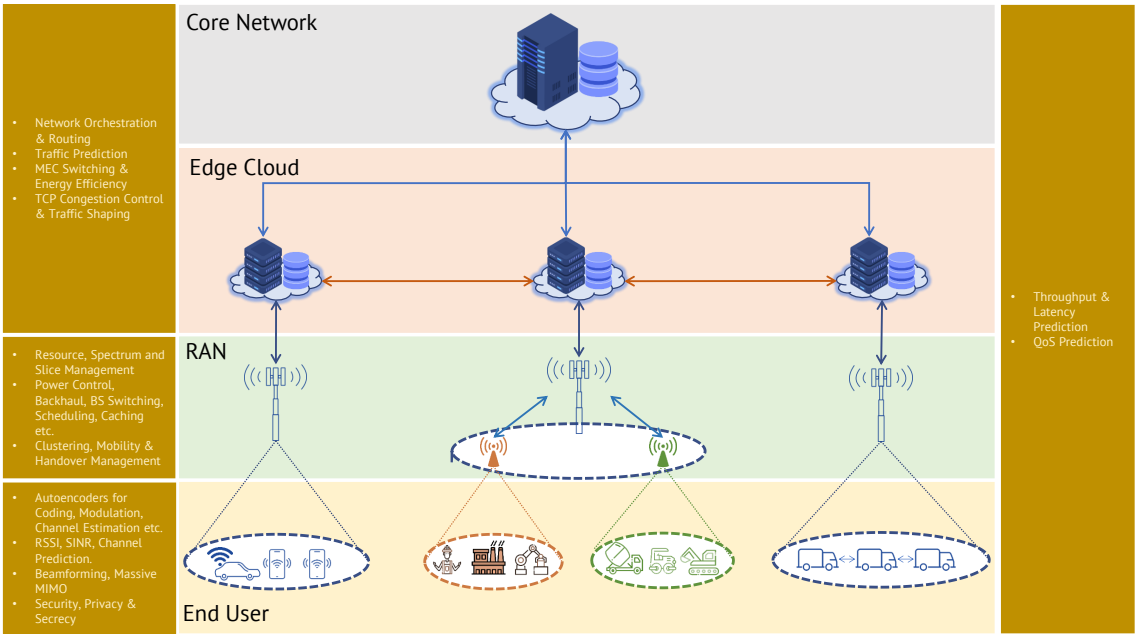


Figure 5.3: Applications of ML at different Network Layers

In line with other engineering fields, ML is rapidly converging with communication technology [155]. Combined with the advances in ML and the availability of powerful cheap hardware (GPUs), many publications were put forward that show the usecases and potential benefits of this convergence. Even though it is impossible to list all of the works due to their sheer volume, we make a comprehensive state of the art analysis by trying to cover some of the works that are spread across the whole network stack.

Survey papers are a good start for getting an overview and categories of problems that have been addressed using ML. Works [168, 32, 178] are surveys that provide a high level overview about ML, the available ML models along with a survey of different works that deal with ML based solutions across PHY, MAC and Network layers. [88] surveys problems and potential models for IoT devices whereas [54] extends the survey to topics such as social media analysis and Wireless Sensor Networks that are relevant for communication. Since DL is emerging as one of the most promising solutions, its applicability for use in wireless networks was surveyed in works such as [5, 117, 217] . ML algorithms that are especially suited for automotive usecases and communication scenarios are surveyed in [213, 183] whereas [9] presents survey results for robotic communication. By summarizing all these surveys, the problems that were addressed by most works were identified.

Figure 5.3 shows the topics that were identified as most prominently investigated and separated with respect to their location in the network stack. It can be seen that most works try to apply ML to solve localized problems within a single layer. With the exception of QoS prediction where the it is done across multiple layers. We start by listing out works starting with Layer 1 (PHY) followed with Layer 2 (Data Link and Network) and extending it to Layer 3 (Edge and Cloud).

The applications of ML/DL at the PHY layer include models for optimization of individual signal processing blocks or an end-to-end transmit-receive auto-encoding [132]. ML/DL

models were realized for different signal processing operations such as Channel Coding [157, 209, 62, 93], Channel estimation [158, 130, 174, 121, 110], MIMO Operations [43, 84, 94, 201], Beamforming [7, 34, 116] etc. An end-to-end communication system solely based on ML without the need for any signal processing blocks is realized in [49, 83, 12]. Physical Layer Security is another emerging topic that has been investigated in [71, 138, 206, 200] with many of them based on unsupervised clustering methods. Channel Prediction (link quality) / Coverage Maps is another important research topic that directly affects and QoS and has been investigated in [4, 11, 99, 105, 107, 173].

The most widespread use of AI/ML in the context of wireless communications can be found in the Data link & Network layer. For the problem of Mobility/Handover Management, solutions based on RNNs [203, 192, 193], Reinforcement Learning [210], Generalized and Fuzzy Q-learning [126, 194, 204], Supervised ANNs [6, 219, 166] are used. Algorithms based on deep reinforcement learning have showed promise for proactive resource allocation as demonstrated by [108, 33, 181, 164, 109] in contrast to the ones based on traditional DL models [48, 165, 182]. A few works/studies have investigated optimal power control strategies using distributed Q-learning [169, 156, 63, 14, 64], reinforcement learning [20] and Support Vector Machines (SVMs) [208]. ML has also found applications in the topic of efficient spectrum management with works such as [55, 218, 8, 177, 127].

Due to the availability of comprehensive end to end data, ML is highly suitable for optimizing core network operations [144, 196]. In [2, 191] and [202], the authors predict the future network traffic based on historical data using DL. All of them used RNNs that are found to perform very well for time-series prediction problems. The use of ML for routing optimization is studied in [106, 198, 170]. [106] and [198] evaluate the routing performance of different supervised learning algorithms such as Multi-Layer Perceptron (MLP), Decision Trees, Rule Learners, Basis Function Neural networks whereas [170] investigates the performance of clustering based schemes. Works such as [81, 82, 215] used DL networks for making a decision on computational offloading, a key topic for deploying Mobile Edge Clouds (MECs).

In order to completely exploit the power of AI, it is important to have an end-to-end integration of the AI framework into the wireless networks. Such an integrated AI framework monitors/collects data across the entire network stack and takes decisions with an aim of optimizing the network right from the core until the end user. The works that consider such an end-to-end optimization generally deal with QoS predictions that makes it possible to adjust application behavior accordingly. Throughput and latency predictions that serve as direct indicators of QoS are used as the baseline. QoS prediction was done using a variety of ML models such as LSTMs [145], Random Forests [154, 216, 151], Gradient boosting [68] using downlink throughput whereas [172] used uplink throughput. Other works such as [97, 173] used raw data rates for QoS prediction. The authors in [212, 31] used other contextual information such as geolocation, neighbour information etc. for a more accurate QoS prediction.

6 ML based Autoencoders

In this chapter, we look at the autoencoding problems starting with singal processing blocks such as channel coding and channel estimation

6.1 ML based Channel Coding

Channel Coding has been one of the central disciplines driving the success stories of current generation wireless systems and beyond. It is a process by which redundancy is added to the input data in order to counter the effects of channel. The goal of any channel coding scheme is to reduce the number of transmission errors which is also quantified as Bit Error Rate (BER), or sometimes loosely referred to as *error rate*. It is defined as the ratio of number of erroneous bits received to the total number of bits. Shannon showed that the channel capacity is typically much greater than the rate achieved by conventional methods and the reliability of the channel depends on its rate rather than SNR. He also derived the theoretical formula for how much information can be transmitted over a channel such that an arbitrarily low error probability can be achieved. This value, called the channel capacity C and in the case of discrete and memoryless channels can be given as

$$C = \max_{p(x)} I(X;Y) \tag{6.1}$$

For the case of AWGN channels with inputs $\pm A$ and noise variance σ^2, the capacity is given as

$$C = \frac{1}{2}\log_2\left(1 + \frac{P}{\sigma^2}\right) \tag{6.2}$$

For a continuous-time AWGN channel with bandwidth WHz, power constraint $\bar{P}$ W, and a noise with power spectral density $N_0/2$, the capacity becomes

$$C_{awgn} = W\log 2\left(1 + \frac{\bar{P}}{N_0 W}\right) \tag{6.3}$$

$$= W\log_2(1 + \text{SNR}) \text{ bits/s} \tag{6.4}$$

In the taxonomy of coding theory, there are basically two methods for channel coding 1. Error-detection coding where errors are detected by means of parity bits and a re-transmission is requested via the feedback channel, 2. Forward Error Correction (FEC), a technique where the transmission errors are detected and corrected without the presence of a feedback channel.

Automatic Repeat Request (ARQ) is a technique which uses error detection code, typically a CRC to detect the presence of errors. If an error is detected in a particular packet, the receiver requests for a re-transmission of the complete packet or just the erroneous part by means of transmitting a negative acknowledgement (NAK) on the feedback channel. Similarly, if no error is detected in the received data packet, the received data is declared

error free and the transmitter is notified by sending a positive acknowledgement (ACK) on the feedback channel. This process goes on until the receiver does not detect an error in the packet, or if the number of re-transmissions has reached a threshold, depending on the implementation.

On the other hand, Forward Error Correction (FEC) works by adding redundancy in the form of parity bits to the information bits. These parity bits are computed from the information bits using different coding schemes which can be broadly classified into two major classes - *block codes* and *convolutional codes*. In block codes one of the $M = 2^k$ messages, each representing a binary sequence of length k, is mapped to a binary sequence of length n, where $n > k$. The output binary sequence, also called as codeword is then transmitted over the communication channel by modulating it with a carrier wave. Block coding schemes are memoryless in the sense that the resulting codeword depends only on the current k information bits and is independent of all the codewords transmitted before. Examples of block codes include Reed-Solomon code [148] , BCH code [80, 25], Low-Density Parity-Check (LDPC) code [65] etc.

Convolutional codes are a class of codes that add dependence between successive blocks instead of treating each code block independently. Thus the current output block depends not only on the information bits in the current input block, but also on those of one or more previous input blocks. Unlike block codes that send message bits interlaced with parity bits, convolutional encoders send only the parity bits. The encoder uses a sliding window to calculate parity bits by combining various subsets of bits in the window. The size of the window, in bits, is referred to as *constraint length* denoted by L. Greater the window size, the higher the resilience of bit errors. But this comes at a cost of increased bandwidth due to coding overhead. Typically, the convolutional encoder performs a convolution of the input stream with the encoder's impulse response

$$y_i^j = \sum_{k=0}^{\infty} h_k^j x_{i-k} \tag{6.5}$$

where x is the input sequence, y^j is the sequence from output j and h^j is the impulse response for output j. A convolutional encoder is a discrete linear time-invariant system. Every output of an encoder can be described by its own transfer function, which is closely related to the generator polynomial G. The convolutional codes are decoded using a Viterbi algorithm whose goal is to find the sequence $\hat{s}$ that was transmitted with the highest likelihood, if the received sequence is r given as

$$\hat{s} = \max_a \Pr(\mathrm{r}|\mathrm{s}) \tag{6.6}$$

where the maximization is done over all possible transmit sequences s and is based on the trellis diagram representation. More details on the Viterbi alogirithm can be found in various works.

Turbo codes [21] are a class of codes that use a parallal concatenation of 2 RSC encoders seperated by an internal interleaver. They are the first practical codes to closely approach the channel capacity, a theoretical limitation of the number of bits that can be reliably transmitted over a given bandwidth. Turbo codes are mostly used for cellular and other applications where a reliable data transfer is required for latency-constrained communication in the presence of data-corrupting noise. However, the decoding algorithm for turbo codes is computationally intensive and thereby limiting its applicability in hand-held devices. In this paper, we study the feasibility of using DL architectures based on RNN

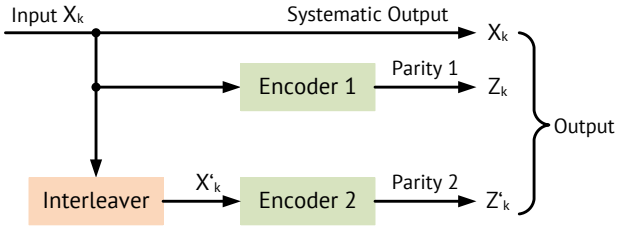


Figure 6.1: Turbo Encoder Architecture

for encoding and decoding of turbo codes. In this regard, we simulate and use data from various stages of the transmission chain (turbo encoder output, AWGN channel output, demodulator output) to train our proposed RNN architecture and compare its performance to the conventional turbo encoder/decoder algorithms.

6.1.1 Turbo Encoder Architecture in LTE

In general sense, a turbo encoder consists of two encoders (referred to as constituent encoders) separated by an interleaver. The encoders are normally identical and the interleaver is used to scramble the bits before being fed to the second encoder. Thus the encoder outputs are different from each other. LTE uses two 8-state identical RSC encoders that are concatenated in parallel and separated by an internal interleaver [187] as shown in Figure 6.1.

The transfer function of each constituent encoder is given as

$$G(D) = \left[1, \frac{g_1(D)}{g_0(D)}\right]$$

where $g_0(D) = 1 + D^2 + D^3$ and $g_1(D) = 1 + D + D^3$ For a given input bit stream $X_0, X_1, ..., X_{K-1}$ of length K, the output of the turbo encoder is given as

$$X_0, Z_0, Z'_0, X_1, Z_1, Z'_1, ..., X_{K-1}, Z_{K-1}, Z'_{K-1}$$

where

1. Bits $X_0, X_1, ..., X_{K-1}$ are the systematic bits as well as the input to the first constituent encoder and the internal interleaver
2. Bits $Z_0, Z_1, ..., Z_{K-1}$ and $Z'_0, Z'_1, ..., Z'_{K-1}$ are outputs from the first an second constituent encoders

As can be seen from the Fig. 6.1, bits $X'_0, X'_1, ...X'_{K-1}$ are outputs from the internal interleaver (and the input to the second constituent encoder) for which the relationship between input and output bits is given as

$$X'_i = X_{\pi(t)},\ i = 0, 1, ..., K-1$$

where the relationship between the output index i and the input index, $\pi(i)$ satisfies the following quadratic form

$$\pi(i) = (f_1 i + f_2 i^2)$$

where parameters f_1 and f_2 depend on the block size K. The valid block lengths along with their corresponding f_1, f_2 values are given in the 3GPP specification [187]. Each bit

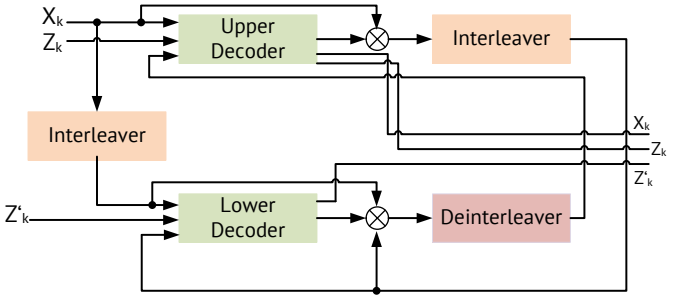


Figure 6.2: Turbo Decoder Architecture

stream is trellis terminated, by taking the tail bits from the shift register after encoding and padding them to the stream bits. This is done so as to reset the encoder state to zero after every encoding operation. Hence, for any given k bits, the output length of the turbo encoder is $3*k+3*m$ where m is the memory size in the shift registers. For LTE variant, the size of m is 4 and hence 4 tail bits are added to each stream totaling 12 bits.

At the receiver end, the turbo decoder consists of two single soft-in soft-out (SISO) decoders that work iteratively. As seen from Figure 6.2, the output of the upper decoder feeds into the lower decoder to form a turbo decoding iteration. Interleaver and deinterleaver blocks re-order data in this process. Two decoding algorithms based on Maximum A Posteriori (MAP), namely $LogMAP$ and $MaxLogMAP$ are used for the decoding process.

6.1.2 Problem Formulation, Data Generation & Preparation

In order to train the model, data was generated using the communications system toolbox in Matlab and the sequence of operations are highlighted in Figure 6.3

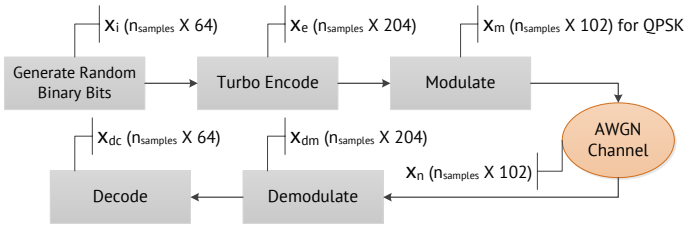


Figure 6.3: Simulation Method

A total of four autoencoding problems have been formulated - one for encoding and the remaining three for decoding as follows.

6.1.2.1 Turbo Encoding

A series of binary bits are given as input to the RNN whose goal is to encode them by adding redundancy and output the turbo encoded bits. 10000 packets of size 64 bits were generated i.e., size$(x_i) = (10000, 64)$ as input and we used the LTE variant of turbo encoder to encode the data. Because of the trellis termination, tail bits are added to each data stream thus creating the resulting encoded data x_e of size $(10000, 204)$. No AWGN noise is considered for this scheme.

6.1.2.2 Turbo Decoding

For the turbo decoding, three data generation approaches were considered

i. Reversing the x_i and x_e obtained from the previous step and feeding the encoded data x_e to the RNN to obtain the decoded bits x_i. This is the simplest case without considering any noise and modulation.

ii. Using the demodulated soft bits x_{dm} which is fed to the RNN in-order to obtain the decoded bits x_i. Data is generated for different SNRs in range [-2,2) totalling 11*10000 = 110000 samples.

iii. Using the noise affected data x_n which is of complex data type directly without demodulation and feeding it as input to the RNN to obtain the decoded bits x_i. Similar to the second approach, data is generated for different SNRs in the range [-2,2) totalling 110000 samples.

For each auto-encoding problem, we shuffle and split the input data into training and testing datasets in the ratio of 30%-70% respectively. No scaling/preprocessing has been used.

6.1.3 RNN Model, Training & Validation

ANNs can be broadly classified into two types - feed forward networks where the information moves only in the forward direction from input nodes, through the hidden nodes and to the output nodes. Examples of such ANN architectures include a simple perceptron, CNNs. These are suitable for learning unconnected inputs such as image / video processing. In this section, we are dealing with turbo encoding and decoding operations that exhibit temporal dependencies between input sequences. Hence, our discussion is limited to RNNs that are better suited to learning connected inputs. Unlike traditional feed forward networks, RNNs have an internal hidden state (memory) whose activation at each time is dependent on that of the previous time and are hence suitable to process connected input sequences. Formally, given a sequence $(x_1, x_2, ..., x_T)$, the RNN updates its recurrent hidden state h_t as follows

$$h_t = g(Wx_t + Uh_{t-1})$$

where g is a smooth, bounded non-linear function (e.g., sigmoid function), and W, U are parameters of the network.

Out of many available RNN, two variants stand out. The first is based on LSTM [79] and the second one is based on the more recent Gated Recurrent Unit (GRU) [37]. The key difference between them is that LSTM uses three gates (namely input, output and forget gates) whereas GRU uses two gates (reset and update gates). The flow of information is similar for both the architectures except that GRU does not use a memory unit and exposes the full hidden content without any control. The performance of GRU is almost on par with LSTM albeit at a lower computational complexity. It is because of these reasons that we select GRUs as the underlying units for our proposed RNN.

GRU

GRU was proposed in [37] to make each unit to adaptively capture dependencies of different time scales. The activation h_t of a GRU at time t is a linear interpolation between the previous activation h_{t-1} and the target activation $\tilde{h}_t$

$$h_t = (1 - z_t)h_{t-1} + z_t\tilde{h}_t$$

where the update gate z_t decides how much the unit updates it activation and is computed as

$$z_t = \tanh(Wx_t + U(r_t \odot h_{t-1}))^j$$

where $\odot$ is an elementwise multiplication and r_t is a set of reset gates that are computed similarly to the update gates as

$$r_t = \sigma(W_r x_t + U_r h_{t-1})^j$$

Activation Function

The activation used in all the models is a sigmoid function to induce non-linearity to the outputs and also due to its good binary classification capability. It is a special case of a logistic function and is defined by the formula

$$S(x) = \frac{1}{1 + e^{-x}}$$

Cost Funtion

The cost or loss function is used to evaluate the performance of a given network and update the weights accordingly. Due to the binary nature of our activation function, we selected binary cross-entropy as the loss function which can be calculated as

$$f(l) = -(y \log(p)) + (1 - y) \log(1 - p)$$

where y is the binary indicator (0 or 1) if the output value is equal to the true value and p is the predicted probability of an observation being either 0 or 1.

Optimizer

For all the models, Adaptive Moment Estimation (ADAM) optimizer was used. It computes the learning rates by calculating an exponentially decaying average of past gradients m_t in addition to past squared gradients v_t as follows[102]

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)g_t$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)g_t^2$$

These values are then used to update the weights according to following rule

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{v_t} + \epsilon} m_t$$

The model used in this work is similar to the one presented in [101] and can be seen in Table 6.2. It consists of two layers of bidirectional GRUs with each layer followed by a batch normalization layer. The output layer is a single fully connected sigmoid unit. The bi-directionality is intended to support recursion in both forward pass and backward pass through the received sequence. It has to be noted that the same model is used for all the autoencoding problems by modifying the input shape (Table 6.1)

Table 6.1: Input Shapes

Problem	Input Shape	Output Shape
Turbo Encoding	(N_s, 1, 64)	(N_s, 204)
Turbo Decoding - 1	(N_s, 1, 204)	(N_s, 64)
Turbo Decoding - 2	(N_s, 1, 204)	(N_s, 64)
Turbo Decoding - 3	(N_s, 2, 204 / m*)	(N_s, 64)

N_s is batch size
*m = 2,4,6 for QPSK, 16QAM and 64QAM

Table 6.2: Layers

Layer	Shape	Parameters
Input	Refer to 6.1	None
GRU - 1	(N_s,1,800)	2078400
Batch Normalization - 1	(N_s,1,800)	3200
GRU - 2	(N_s,800)	3844800
Batch Normalization - 2	(N_s,800)	3200
Dense	Refer to 6.1	Variable

Total Trainable Parameters ≈ 6,000,000

The training set is further split into model training and validation datasets in the ratio of 90%-10%. The model is trained on the training dataset on an Nvidea GPU for 30 epochs (an *epoch* is one pass over the entire dataset) and validated on the validation set. Finally, the model is applied on the testing dataset.

6.1.4 Results

For problem sets 1 and 2, the training, validation and testing accuracies for a single SNR point were used for performance evaluation since there is no noise added to the input data. For problem set 3 and 4, the testing Bit Error Rate (BER) (1 – testing accuracy) per SNR point was used.

Turbo Encoding and Decoding on data with no noise

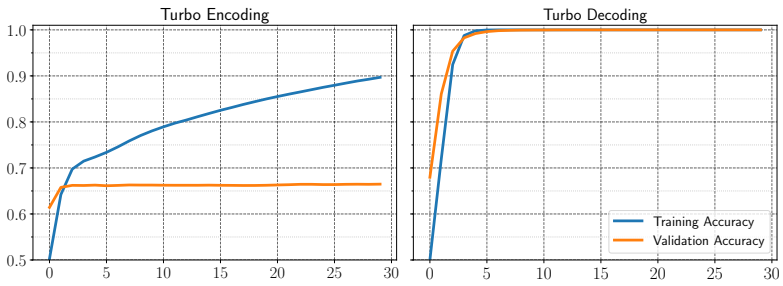
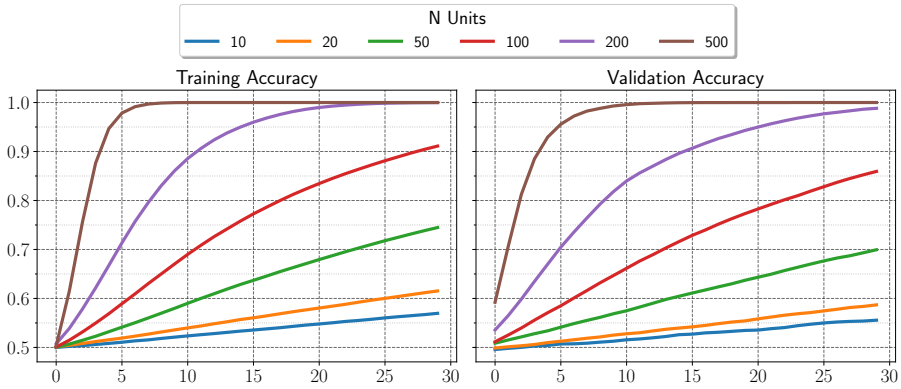
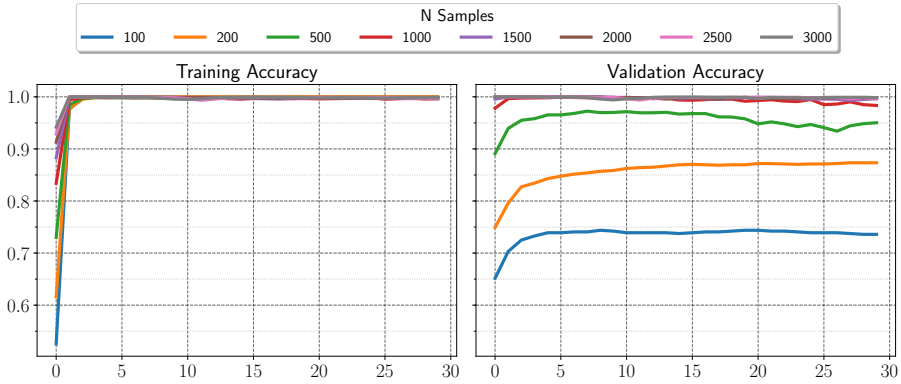


Figure 6.4: Training & Validation Accuracy - No Noise

Figure 6.4 shows the training accuracy of the model over number of epochs for the turbo encoding and decoding performance without any noise (Problem 1 and 2). In case of encoding, even though the model shows a high training accuracy, the validation accuracy stays constant at 67% after 3 epochs. On the contrary, the model performs well for decoding the turbo codes with training and validation accuracies approaching 100% after 6 epochs.



(a) Effect of Number of GRU Units on Accuracy



(b) Effect of Number of training samples on Accuracy

Figure 6.5: Model Selection & Validation

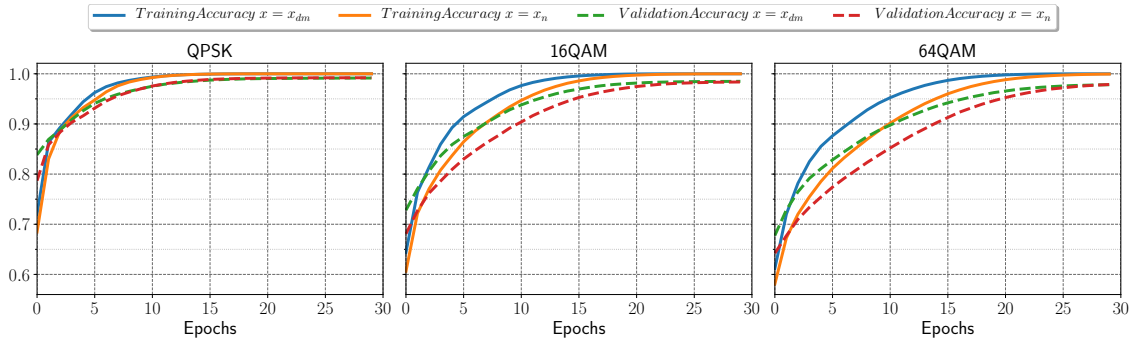
This shows us, that the proposed model is good for the decoding operation and not very well suited for encoding.

The testing accuracies for both the turbo encoding and decoding operations are consistent with the validation accuracies and are outlined in Table 6.3

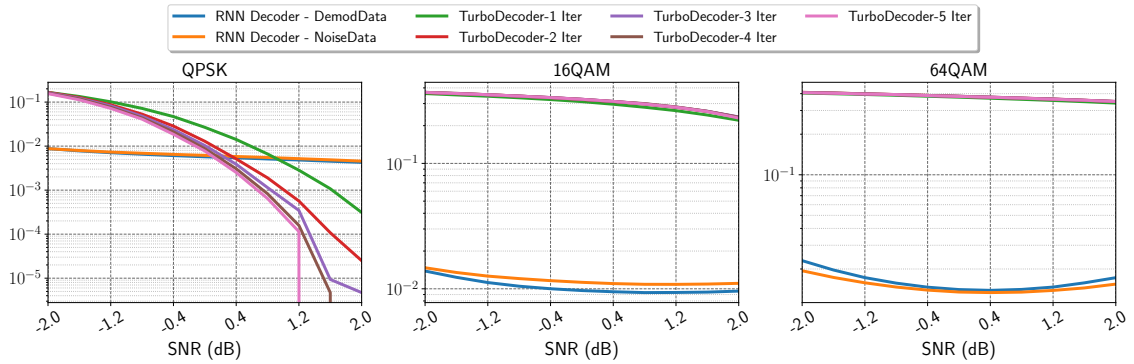
Table 6.3: Testing Accuracy - No Noise

Problem	Testing accuracy
Turbo Encoding	67%
Turbo Decoding - 1	100%

6.5a and 6.5b show the effect of number of GRU units and the number of samples on training and validation accuracies respectively. It can be seen, that if the number of GRU units is less than 200, the model fails to converge for the selected number of epochs. Hence, the choice of having 800 units is a safe assumption. Similarly, the choice of using 3000 samples for training the model also seems safe given that there is some jitter in accuracy for the number of samples$<$2000.



(a) Training and Validation Accuracies



(b) BER Curves

Figure 6.6: Decoding Performance - Problem 3 & 4

Turbo Decoding on Demodulated and Channel Output Data

For problems 3 and 4, the data from the demodulator and the channel are used respectively. 6.6a shows the evolution of training and validation accuracies with respect to the number of epochs for the three modulation schemes. It can be seen clearly that the model shows good training and validation performance on QPSK when compared to that of 16-QAM and 64-QAM. It can also be seen that the model performance is similar on both the demodulated and noise data which shows the ability of the model to understand the modulation structure thereby eliminating the need for demodulating the data beforehand. A careful look at the validation accuracies when using both the demodulated and noise data reveals, that the model fails to converge for the considered number of epochs.

6.6b shows the BER performance of the model on both the datasets (calculated as 1 − Testing Accuracy) for each given SNR. It can be seen that the RNN model outperforms the conventional turbo decoder (for all decoding iterations) for low SNRs (<0.4 dB). However, at higher SNRs, the turbo decoder's BER drops down exponentially while the drop is only linear for the RNN decoder.

6.2 ML based Channel Estimation

Channel Estimation refers to the process of characterizing the wireless channel in order to remove its distortion effects from the received signal. It forms an important component in OFDM systems that aims to mitigate the ISI by calculating the Channel State Information

(CSI). The channel estimation techniques can be broadly classified into two categories - blind and non-blind. The blind channel estimation techniques exploit the statistical behaviour of received signals to estimate the channel. Examples of such methods are the subspace-based algorithms using precoding methods where a linear block precoder is applied at the transmitter and the CSI is extracted by exploiting the covariance matrix of the received signals [66, 153, 142, 141, 167]. Other subspace algorithms include noise subspace method [207] and virtual carriers [36] based methods. Other blind methods exploit the cyclostationary characteristics of the OFDM signal via the CP [124, 76, 125, 205]. The advantage of using blind estimation algorithms is the increased bandwidth and throughput due to eliminating the pilot sequences. However, they also require large amount data for a reliable stochastic estimation and hence suffer from high computational complexity and severe performance degradation in fast fading channels.

In non-blind channel estimation, known symbols called pilots are transmitted at known positions in the OFDM waveform. These pilots can be arranged in a few frequency SCs in all OFDM symbols (comb configuration), or across all SCs in a few OFDM symbols (block configuration) or across few subcarriers on few OFDM symbols (2D grid configuration). At the receiver side, the channel is estimated by means of comparing the received pilot symbols with the transmitted pilot symbols. A plethora of algorithms exists for estimating the channel in OFDM systems with varying levels of complexity [77, 220]. However, for the purpose of simplification, we limit our discussion to the two most widely used pilot based channel estimation techniques - the LS estimator and the Minimum Mean Squared Error (MMSE) estimator.

If $X(t)$ is the time-domain transmitted waveform, the received waveform $Y(t)$ after being corrupted by channel effects is given as

$$Y(t) = X(t) \cdot H(t) + W(t) \tag{6.7}$$

where $H(t)$ is the complex channel gain and $W(t)$ is the the complex zero-mean Gaussian noise. The LS estimator minimizes the following cost function

$$\min_{\hat{H}} (Y - X\hat{H})^H (Y - X\hat{H}) \tag{6.8}$$

where $[.]^H$ is the Hermitian (conjugate) transpose operator. The LS estimation is then given as

$$\hat{H}(t) = \frac{Y(t)}{X(t)} = H(t) + W(t) \tag{6.9}$$

where $H(t)$ denotes the true channel coefficients and $W(t)$ denotes the noise. Hence, it can be seen that LS estimator allows us to calculate channel coefficients with a simple division operation without relying on any knowledge about statistics of the channel. This also leads to its inherent problem of high mean-squared error since it does not take into account the effect of noise on the signal.

MMSE estimator, on the other hand employs the second-order statistics of the channel, channel correlation function and the operating SNR for estimation. Let R_{gg}, R_{HH} and R_{YY} denote the auto-covariance matrix of g, H and Y respectively. The cross covariance between g and Y is denoted by R_{gY}. Furthermore, it is assumed that the channel vector H and the noise vector N are uncorrelated. We can then derive

$$R_{HH} = E\{HH^H\} = E\left\{(Fg)(Fg)^H\right\} = FR_{gg}F^H \tag{6.10}$$

$$R_{gY} = E\{gY^H\} = E\left\{g(XF_g + N)^H\right\} = R_{gg}F^H X^H \tag{6.11}$$

and,

$$R_{YY} = E\left\{YY^H\right\} = XFR_{gg}F^HX^H + \sigma_N^2 I_n \tag{6.12}$$

where σ_N^2 is the noise variance, given as $E\{|N|^2\}$, and I_N is the NXN identity matrix. Assuming R_{gg} (thus R_{HH}) and σ_N^2 are known at the receiver, the MMSE estimator of g is given as

$$\hat{g}_{MMSE} = R_{gY}R_{YY}^{-1}Y \tag{6.13}$$

Finally, the frequency domain MMSE estimator can be calculated as

$$\begin{aligned}\hat{h}_{MMSE} &= F\hat{g}_{MMSE} \\ &= F\left[(F^HX^H)^{-1}R_{gg}^{-1}\sigma_N^2 + XF\right]^{-1}Y \\ &= FR_{gg}\left[(F^HX^HXF)^{-1}\sigma_N^2 + R_{gg}\right]F^{-1}\hat{H}_{LS} \\ &= R_{HH}\left[R_{HH} + \sigma_N^2(XX^H)^{-1}\right]^{-1}\hat{H}_{LS}\end{aligned} \tag{6.14}$$

Hence, it can be seen that the MMSE estimator yields much better performance, especially under the low SNR scenarios. However, it is also computationally expensive, due to multiple matrix inversions and complex due to the knowledge required in the form of channel correlation and the noise variance.

6.2.1 Channel Estimation in C-V2X

Each sidelink subframe (1 ms) contains 14 OFDM symbols out of which 10 are used for carrying user data and the remaining 4 (at positions [2,5,8,11] with 0-based indexing) are used for carrying DMRS symbols. The DMRS symbols are sequences $r_{u,v}^{(\alpha)}$ that are obtained by a cyclic shift of a base sequence $r_{u,v}(n)$ according to

$$r_{u,v}^{(\alpha)} = e^{j\alpha n} \cdot \bar{r}_{u,v}(n), 0 \leq n \leq M_{sc}^{RS}$$

where $M_{sc}^{RS} = mN_{sc}^{RB}$ is the length of the DMRS sequence, m is the number of PRBs and N_{sc}^{RB} is the number of subcarriers within one PRB (12 in case of LTE). The base sequence itself is defined as the cyclic extension of the Zadoff-Chu Sequence and is given as

$$\bar{r}_{u,v}(n) = x_q\left(n \mod N_{ZC}^{RS}\right), 0 \leq n \leq M_{sc}^{RS}$$

$$x_q(m) = e^{-j\frac{\pi qm(m+1)}{N_{ZC}^{RS}}}, 0 \leq m \leq N_{ZC}^{RS} - 1$$

where $x_q(m)$ is the q_{th} root of Zadoff-Chu sequence and N_{ZC}^{RS} is the length of Zadoff-Chu sequence that is given by the largest prime number such that $N_{ZC}^{RS} < M_{sc}^{RS} < 3N_{sc}^{RB}$, the base sequence is defined as the computer generated CAZAC sequence.

$$\bar{r}_{u,v}(n) = e^{j\varphi(n)\pi/4}, 0 \leq n \leq M_{sc}^{RS}$$

The transmitting node can select a base sequence from a set of groups each differentiated with a hopping sequence that depends on the current subframe number and the V2X scrambling identity. In this way, the DMRS sequences are randomized for different vehicles thereby reducing inter-cell interference. The DMRS symbols along with the data symbols are multiplexed together, modulated by SC-FDMA and then passed through a channel. At the receiver, the received OFDM grid denoted as Y_t is given as

$$Y_t = H_tX_t + N_t,$$

where H_t is the channel frequency response and N_t is the AWGN for symbol t. As a first step in LS channel estimation, the receiver extracts the pilot symbols from their known location in the time-frequency grid and divides them with their expected value

$$\tilde{H}_{(i,k)} = \frac{Y_{(i,k)}}{X_{(i,k)}} = H_{(i,k)} + N_{(i,k)}$$

where $\tilde{H}_{(i,k)}$ is the LS channel estimate at pilot location (i,k), $Y_{(i,k)}$ and $X_{(i,k)}$ are the received and sent pilot symbols at (i,k) and $N_{(i,k)}$ is the noise at (i,k). It can be seen that the calculated LS estimate is noisy and hence in order to minimize the effect of noise, a 2D averaging is performed with a chosen window size. Hence, averaging the instantaneous channel estimates over the window, we have

$$\tilde{H}^{AVG}(i,k) = \frac{1}{|S|} \sum_{m \in S} \tilde{H}_{(i,k)}(m) \approx H_{(i,k)}$$

where S is the set of pilots in the 2D window and $|S|$ is the number of pilots in S. The LS estimates and the averaged estimates contain the same data, apart from additive noise. Simply taking the difference between the two estimates results in a noise level value for the LS channel estimates at pilot symbol locations. This knowledge of noise can be useful to increase the performance of some receivers especially using soft demodulation techniques.

Finally, the averaged LS estimates are interpolated across the whole time-frequency grid to get the complete channel matrix $H(t)$ for the received subframe. Equalization is performed by multiplying the received grid $Y(t)$ with the complex conjugate of $H(t)$

$$Y^{eq}(t) = Y(t) * H(t)^*$$

6.2.2 Problem Formulation, Data Generation and Preparation

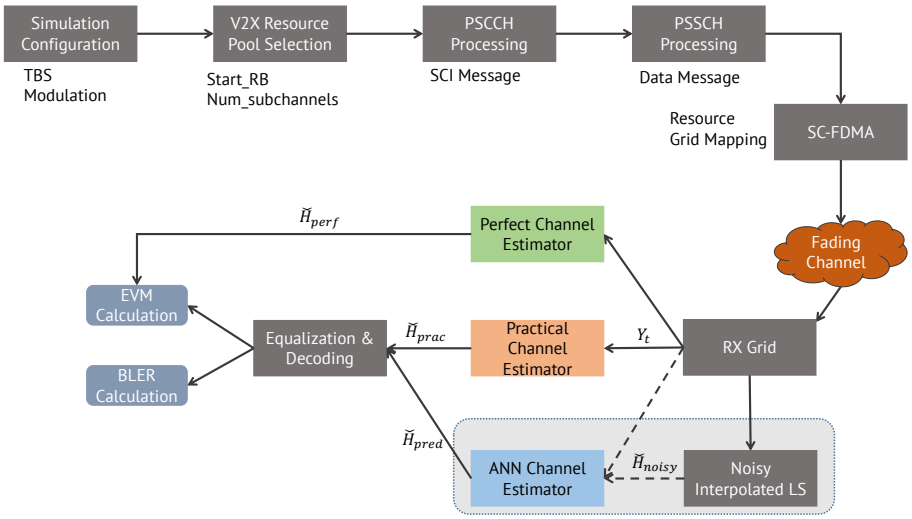


Figure 6.7: Simulation Method

The data is generated using the link level simulator whose workflow is shown in Figure 6.7. For the given set of parameters as outlined in Table 6.4, we generated a set of sidelink subframes. These subframes were then converted to a time domain waveform by employing

Table 6.4: Simulation Parameters

Parameter Group	Name	Value
High-level Parameters	Bandwidth	10 MHz
	NSLRB	48
	TBS	3496
	N_Subframes	500
	SNR Range	[-2, 5] dB
SCI Message	Modulation	QPSK
	Time Gap	1 subframe
Data Message	Modulation	QPSK
Channel	Delay Profile	EVA
	MIMO	1X2
	Speeds	[100,200,300,400] kmph

SC-FDMA and the waveform was passed through a multi-path fading channel (with *EVA* delay profile) to get the received grid Y_t. The receiver operations consist of subframe synchronization followed by perfect and practical channel estimation that produced channel matrices ($\hat{H}_{perf}$) and $\hat{H}_{prac}$ respectively. The noisy LS estimates were obtained by dividing the received DMRS with the transmitted DMRS symbols and this is linearly interpolated over each subframe to get $\hat{H}_{noisy}$.

For the training data, we generated a total of 500 subframe samples for SNR values ranging between $[-2, 5]$ dB hence totaling 5000 samples. This process was repeated for 4 different speeds (Table.6.4) bringing the total number of samples to 20000. Each sample is has a shape of $(576, 14)$ corresponding to 48 PRBs in frequency domain and 14 symbols in time domain. y also has a similar shape as X.

6.2.3 DL Architecture, Training & Validation

Due to the spatial dependency between the DMRS in the time-frequency resource grid, we adopt DL architecture based on CNN and compare their performance with respect to BLER and Error Vector Magnitude (EVM) to that of the perfect and practical channel estimators. As shown in Figure 6.8, the proposed model consists of 4 convolutional layers with different kernel sizes each of them followed by a batch normalization to minimize vanishing or exploding gradients. The final layer is a Dense layer followed by a reshape layer to reshape the data to have the same dimensions as the input data.

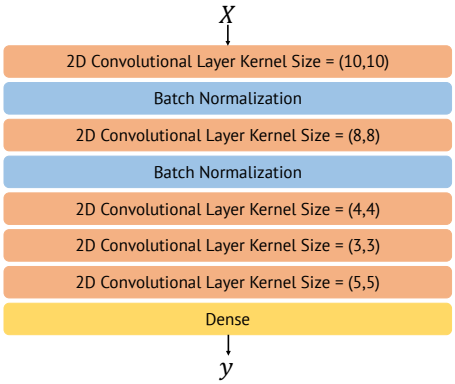


Figure 6.8: ANN Architecture

The input to the ANN is the noisy interpolated LS channel matrix $\hat{H}_{noisy}$ and the output is the estimated channel matrix $\hat{H}_{pred}$

$$\hat{H}_{pred} = f(\Phi; \hat{H}_{noisy})$$

where Φ is the set of parameters of the ANN

For training, we used 30 % of the samples. Finally, the trained model is used to output the $\hat{H}_{pred}$ for the whole sample set.

The loss function is the Mean Squared Error (MSE) between the estimated $\hat{H}_{est}$ and perfect channel matrix $\hat{H}_{perf}$ and is calculated as follows

$$MSE = \frac{1}{\|\tau\|} \sum_{h \in \tau} \left\| f(\Phi; \hat{H}_{pred})) - \hat{H}_{perf} \right\|^2$$

For optimizing the loss, ADAM optimizer was used. It computes the learning rates by calculating an exponentially decaying average of past gradients m_t in addition to past squared gradients v_t as follows[102]

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

These values are then used to update the weights according to following rule

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{v_t} + \epsilon} m_t$$

The network was trained for 20 epochs and then used to predict $\hat{H}_{pred}$.The predicted channel $\hat{H}_{pred}$ is then used for equalizing the received grid. The equalized grid is then subsequently decoded and compared to the input data bits to obtain the BLER. In order to quantify the performance of the practical and ANN based channel estimator, the EVM was chosen as the metric that is calculated as follows

$$EVM = \sqrt{\frac{\frac{1}{N}\sum_{k=1}^{N}(e_k)}{\frac{1}{N}\sum_{k=1}^{N}(I_k^2 + Q_k^2)}} * 100$$

where $e_k = (I_k - \hat{I}_k)^2 + (Q_k - \hat{Q}_k)^2$, $(I_k, Q_k)\&(\hat{I}_k, \hat{Q}_k)$ represent the In-phase component and the Quadrature phase component of the ideal and measured symbols respectively.

6.2.4 Results & Conclusions

Figure 6.9 shows the BLER performance comparison between the practical channel estimator and the ANN based channel estimator. It can be clearly seen that the ANN based channel estimation scheme performs on par with the LS scheme at low speeds and low SNRs. The real benefits of using ANN scheme become apparent at higher speeds and higher SNRs as the proposed scheme outperforms LS scheme by almost an order of magnitude. This is because, at higher speeds, the averaging and interpolation used in LS causes excessive information loss thereby resulting in pure noise. In contrast, ANN was better able to learn the quick channel variations in high speed scenarios. Figure 6.10 also shows the EVM performance between the ANN and the LS schemes. It can be seen that the EVM is almost identical for both the schemes at low speeds. At higher speeds and higher SNRs, ANN scheme shows lower EVM than the LS scheme due to the effectiveness of the channel estimation.

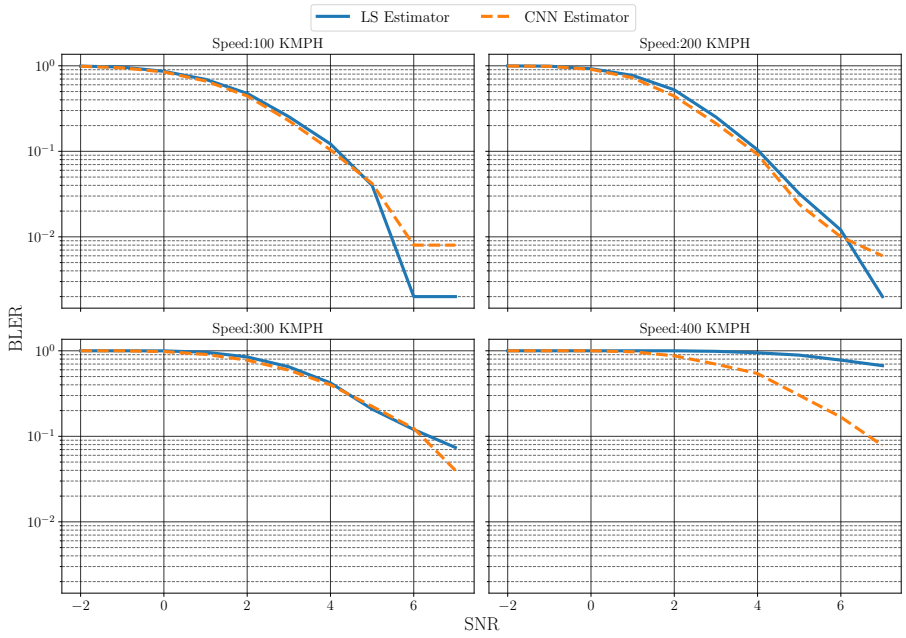


Figure 6.9: BLER Performance

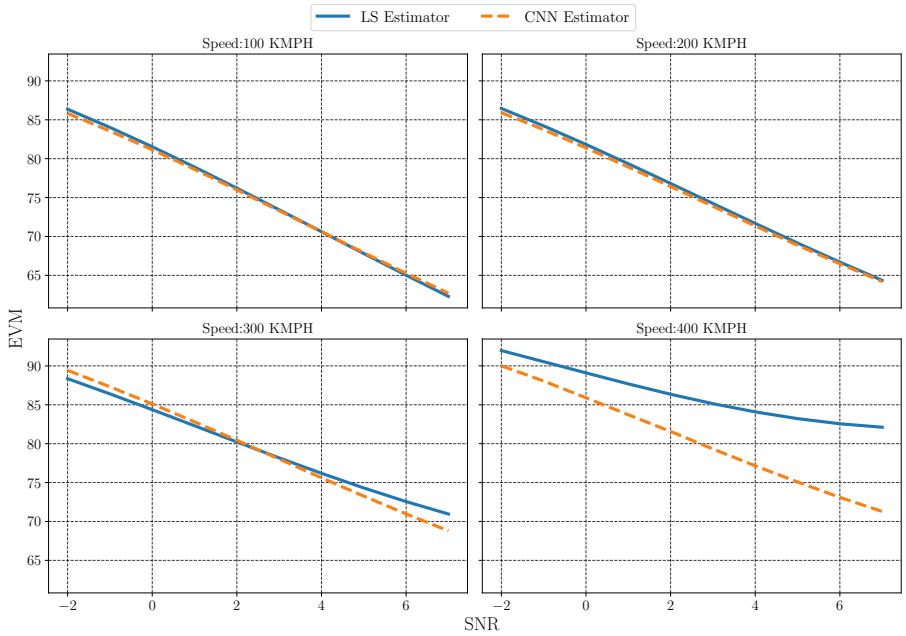


Figure 6.10: EVM Performance

7 Reliability Prediction using ML

Historically, the focus of cellular communications has been on human-centric communication with user throughput being the main optimization parameter. The 5G network additionally supports vertical domains such as vehicular and industrial communications etc. Contrary to human-centric communications, these vertical domains deal with communication between sensors, actuators, physical objects and other embedded controllers that not directly operated by humans. Moreover, the rate of communication is also typically orders of magnitude higher than that of traditional communications. The ability to support such diverse use cases (also referred to as URLLC use cases) forms the core design target from the very beginning for 5G networks.

In a typical wireless transmission, many physical parameters like noise, interference, distortion or bit synchronization errors influence the channel characteristics and hence the transmission quality. These parameters inject uncertainties in the wireless channel which in turn directly reduces the real-time or reliable transmission capabilities of the channel as a whole. These uncertainties are impossible to eliminate, however, it is possible to reduce the effect of these uncertainties and guarantee an acceptable reliable estimate for the success probability and the timeliness of message delivery.

7.1 Related State of the Art

The performance of wireless communication is inherently a stochastic variable. This randomness is primarily due to two reasons 1. the time, frequency & space invariance of the wireless channel, and 2. the way in which the channel is modeled mathematically - by employing simplifications in order to obtain tractable stochastic approximations of the underlying mathematical model. Subsequently, many techniques were used to increase the reliability of the wireless transmission. Early efforts started by simply allowing a fade margin (margin of error) in case of any unexpected signal distortion. This is then followed by more advanced techniques such as frequency diversity (e.g., transmitting same data at multiple frequencies). spatial diversity (e.g., transmitting same data over multiple antennae) and time diversity (e.g., transmitting same data at multiple time slots). later enhancements include link adaptation techniques such as adaptive MCS where the signal and protocol parameters are adapted to the conditions of the radio link as measured at the receiver and transmitted to the base station by means of a feedback channel. These techniques, however resulted in a trade off decision to be made in terms of latency versus reliability. On the other hand, increasing reliability required higher computational resources (for channel coding, parity etc.) and retransmissions thereby inducing additional latency at both the transmitter and the receiver. This is not acceptable for URLLC applications with hard real time constraints. On the other hand, ultra-low latencies mandate the use of short packets which in turn severely limits the channel coding gain [19], thus violating the ultra-high reliability requirement.

From an another end, significant research efforts were also invested in estimating and tracking the channel accurately - a critical requirement for coherent decoders . Starting with GSM, wireless networks increasingly relied in Pilot Assisted Transmission (PAT) techniques [28] where a group of known symbols called pilots are multiplexed with data symbols which is then exploited by the receiver for channel estimation. Since, the pilot symbols are transmitted in non-contiguous blocks, an additional mechanism is needed in order to interpolate and predict the chanell coefficients between successive pilot transmssions. The most widely used and simplest channel estimator is a linear adaptive filter where the channel coefficients are updated based on the Least Mean Squares (LMS) algorithm [75]. Later works such as [111, 67, 211, 129] have been developed in order to improve the performance of the LMS algorithm especially in time varying environments and majority of these works were deicated to using an optimum window size (predication window) to improve the tracking performance. However, the gain is heavily dependent on th accurate estimation of the optimal prediction window [128].

7.2 Reliability

Reliability in wireless communication is defined and addressed differently at different layers of the communication stack since reliability as an attribute should be defined by the application itself. For some applications like data transfer, reliability is about data integrity and all the information sent by the transmitter must be accurately received at the receiver. For other applications such as audio and video, it is less about data integrity and more about tolerable distortion at the application layer which is a convoluted function of error rates, error burstiness, delay, error concealment techniques, etc. traditionally each layer of the communication stack addresses reliability at different time scales to fix errors that are not correctable, observable, or too costly to correct at the lower layers. In wireless systems, however, independent decisions at each layer often lead to an unreliable or inefficient communication environment. Therefore, some degree of cross-layer coordination/optimization has been proposed by numerous research papers and adopted by some systems (especially between the PHY and MAC layers).

The PHY layer in a digital communication system is responsible for bit-level transmission/reception of signals between the nodes. It has to endure that the transmitted bits are reliably reconstructed at an intended receiver. Some of the metrics that characterize reliability at the Physical layer include the SINR, BER, Symbol Error Rate (SER), PER and outage probability [72]. A reliable communication in this context is defined as having an arbitrarily small error probability P_b, and the maximum data rate at which reliable communication is possible is defined as the capacity C of the channel. Achievable capacity for reliable communications can simply be written for AWGN channels as

$$C_{awgn} = B \log \left(1 + \frac{P_{rx}}{\alpha_1^2 + \alpha_2^2} \right) \tag{7.1}$$

where B is the communication bandwidth, P_{rx} is the received power of the signal, α_1^2 is the variance of error/interference terms which are assumed to be Gaussian independent of the noise term), α_2^2 is the noise variance given by $B.N_0$ with N_0 being the noise spectral density, and $\frac{P_{rx}}{\alpha_1^2+\alpha_2^2}$ is the SINR

As the channel capacity in Equation 7.1 increases, reliable communications become possible at higher data rates. In order to increase the capacity, the bandwidth B can be increased (e.g., through scheduling algorithms), average interference power (α_1^2) can be decreased (e.g., through interference cancellation techniques), or the received power P_{rx} can be increased (e.g., through power control algorithms)

At the MAC layer, reliability is traditionally defined from the data integrity point of view and packets erroneously received from the physical layer are dropped. Thus, a critical metric at this layer is the Packet Drop Ratio (PDR). Collision free channel access and coded/uncoded packet retransmissions are the main mechanisms employed at this layer to improve the PDR. Methods like ARQ and HARQ can be used to provide higher levels of reliability. These schemes exploit the availability of a reliable feedback link in order to provide an indication of correct receipt of packet data. Error-detection bits are appended to the packet in order to detect whether the packet is correctly received. Polynomial codes such as CRC are examples of error-detecting codes.

At the routing/network layer, reliability traditionally targets end-to-end connectivity and maintenance of sufficiently high-quality communication path under dynamic network conditions. Network conditions might vary as a result of node or link failures, mobility, changes in wireless channel quality, changes in traffic demand, etc. Depending on the particular scenario, few of these network dynamics become the dominant characteristics and routing protocols can be customized accordingly with various notions of reliability. For instance, many works on routing in wireless networks in the context of Mobile Adhoc NETworks (MANETs) have mainly focused on developing protocols that can work in high-mobility scenarios. With links forming and tearing up quite fast route discovery and packet losses due to lack of connectivity are the main reliability issues that have been investigated. Therefore, routing protocols in MANET scenarios have been evaluated principally with respect to their overhead versus PDR mainly under deterministic coverage models [139, 27].

When wireless nodes are quasi-stationary or stationary (e.g., vehicles as moving base stations), other aspects, such as losses due to unreliable wireless channel conditions and to congestion, network stability, delay, and network capacity, surface as critical objectives. Some of the notable developments to increase the reliability of the routing layer range from devising new routing metrics [103] to developing better protocols that utilize techniques such as multipath diversity, opportunistic routing, back-pressure algorithms, cooperative communications, erasure and network coding. Many of these methods take advantage of the broadcast medium and cross-layer optimization with the PHY and MAC layers being important aspects.

7.3 Availability Indication

In order to have a sufficient cross-domain analysis & optimization, it is important to redefine reliability as an attribute. In line with the Mobile and Wireless communications Enablers for Twenty-twenty (2020) Information Society (METIS) project, the reliability and timeliness have been captured together with a single definition for reliability as the deadline until successful decoding of the packet. To this end. METIS also defines a class of use cases called ultra-reliable machine-type communication (uMTC) that refers to services that provide very high reliability and often very short latencies. Further definitions such

as service availability and failure were also defined and adapted for the case of wireless transmission. This presents the opportunity to distinguish reliability at a link and system level. Link reliability is the ability of the radio link to transmit and receive certain amount of data successfully within a predefined deadline. System reliability is th ability of the system to accurately indicate the absence of link reliability to the application, and at the same time, to ensure the presence of link reliability as often as possible when required by the application.

Hence, in this context, we introduce a concept for Ultra-Reliable Communication (URC) based on a novel metric referred to as *Availability* which indicates the presence or absence of link reliability. URC applications such as road safety use cases require very high and predictable success rates within low deadlines (e.g.,100 m sec). Due to the sensitive nature of these applications, it is of paramount importance to warn the application about the absence of link reliability according to application specific requirements. The current generation of mobile communication systems are typically not designed to provide reliability at all times and in every reception scenario, as this would result in an overdesigned system with a very inefficient air interface in terms of data rates and power consumption. Such an approach would harm the acceptance of URC services and restrict their usage. In the following, we define the URC concept in a universal manner, i.e., without relying on details of any air interface specification or signaling scheme. This approach is motivated by the fact that a transport- agnostic definition from the applications point of view is required in order to allow URC services to be deployed in a wide range of scenarios. From this perspective, the implementation details related to the wireless communication are not included in the proposed URC concept.

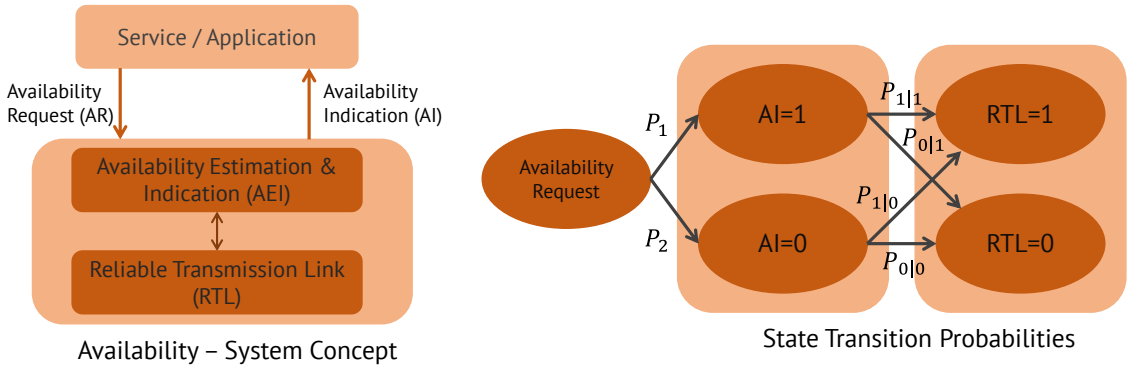


Figure 7.1: System concept and state space for Availability Indication

Figure 7.1 shows the system concept for URLLC based on two entities 1. Reliable Transmission Link (RTL) that is optimized to transmit packets successfully and within a predefined deadline 2. Availability Estimation and Indication (AEI) mechanism that is able to reliably predict the availability of the RTL under given conditions. In addition, a novel link control indicated called AI signals the outcome of the AEI to the application. In this context, the application requests an RTL by sending an Availability Request (AR) to the AEI. The AEI is designed to indicate to the application the availability of the RTL for the forthcoming transmissions given the AR and the historical data (e.g., SINR, ACK/NACK statistics etc.). The indication is done by means of the AI that is typically a binary value, i.e., RTL available (AI = 1) or unavailable (AI = 0). After indicating the RTL availability, the application will be able to use it by transmitting data packets over the RTL.

Figure 7.2 describes a possible implementation of the URC system concept. According to

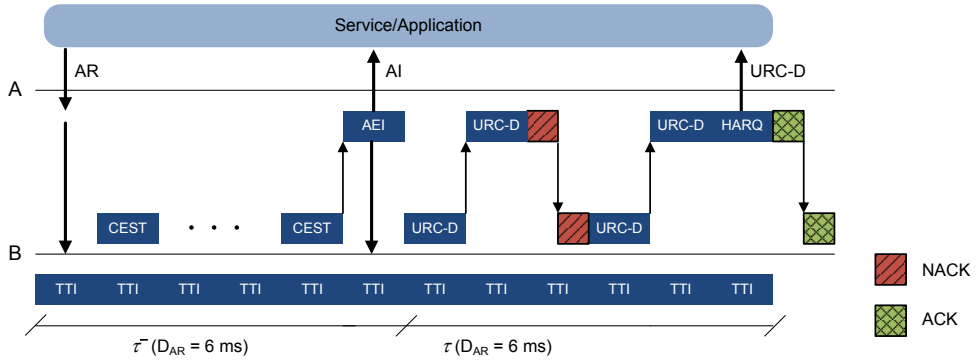


Figure 7.2: Example Timing illustration of URC Concept

this implementation, an application requests in time slot τ^- the use of an RTL for the transmission of data from "B" to "A". This could be the case of a road safety application awaiting information from nearby traffic participants in order to alert the driver or to take control of the vehicle so that a collision could be avoided. The request is being done by means of an AR, which is first generated in "A" and then forwarded to "B". Please note that the AR could alternatively be generated by the application in B. In response to the AR, "B" starts the transmission of a pilot signal for channel estimation, which is used by the AEI in "A" to predict the availability of the RTL in the following time slot τ according to the maximum delay that is tolerated by the application. In the case of Figure 7.2, each time slot extends across six Transmission Time Intervals (TTIs), i.e.,($\Gamma = 6ms$).

Based on this prediction, the AI will be signaled to the application requesting the RTL. In case of RTL availability (AI=1), the application informs both "A" and "B" about the start of the RTL usage. After that, the application starts sending the URC data (URC-D) from "B" to "A" through the RTL. Here a processing delay of 1 ms is assumed. "A" will then try to decode the URC-D. If successful, an ACK is sent; if not successful, a NACK is sent instead. Figure 7.2 shows the case where the first decoding attempt is not successful, and a NACK is sent from "A", which triggers "B" to send a retransmission. After receiving the retransmission, the HARQ recombining process in "A" decodes successfully the data and delivers it to the application.

As seen in the example, the AEI is in charge of predicting the availability of the RTL in future time slots based on the application requirements carried by the AR, and also on channel information such as SINR measurements. The AR comprises information about the application requirements that might impact the availability of the RTL, such as the data packet size or the maximum delay until successful reception. For example, in the case of a road safety application for intersection collision warning, data packets of approximately 200 B have to be delivered successfully to other traffic participants within 100 ms. The computation of the AI in automotive scenarios can also benefit significantly from information regarding the variability of the channel in the time domain, especially in wireless communication systems that allow the use of retransmissions and HARQ. Fast varying channels generally result in a higher time diversity that improves the performance gain of retransmissions and HARQ techniques. In this sense, the vehicle speed can be used to estimate the time variability of the channel and predict the availability of the RTL in a more precise manner. Furthermore, the AEI mechanism can also use related vehicular sensor and context information for the computation of the AI, such as route and trajectory information given by the navigation system of the vehicle in combination with pre-recorded

radio maps, which might contain, among others, information related to the SINR or the packet error rate.

7.3.1 Mathematical Interpretation

In the following, we formulate mathematically the concept by adopting a simple time-slotted model, in which each time slot, τ, corresponds to the time interval $[t, t + \mathrm{D_{AR}})$, where $\mathrm{D_{AR}}$ is the maximum delay tolerated by the application. Under this definition we define

$$\mathrm{RTL}(\tau) = \begin{cases} 1, & \text{transmission is successful} \\ & \text{for time slot } \tau; \\ 0, & \text{transmission is not successful} \\ & \text{for time slot } \tau. \end{cases}$$

For the availability indication, a simple binary signaling format per time slot is used, where

$$\mathrm{AI}(\tau) = \begin{cases} 1, & \text{AI indicates the availability of RTL} \\ & \text{for time slot } \tau; \\ 0, & \text{AI indicates the non-availability of RTL} \\ & \text{for time slot } \tau. \end{cases}$$

Figure 7.1 describes the URC state transition probabilities divided into two steps. The essence of the URC concept is that an application should rely on the wireless communication only in those instances in which the link reliability is guaranteed with a certain probability. In other words, once the AI indicates the availability of a RTL for time slot τ to the application, the probability of transmission success for the time slot τ must be above a certain value, $\mathrm{P_{UR}}$, according to the application requirements. We refer to this criteria as the ultra-reliable requirement, which can be understood as the conditional probability

$$\mathrm{P_{1|1}} = \Pr(\mathrm{RTL}(\tau) = 1 | \mathrm{AI}(\tau) = 1), \tag{7.2}$$

where $\mathrm{P_{1|1}} \geq \mathrm{P_{UR}}$. In this context, the goal of the URC system design is to maximize the availability of the RTL under the previously defined requirement:

$$\begin{aligned} \max \quad & \mathrm{P_1} = \Pr(\mathrm{AI}(\tau) = 1) \\ \text{s.t.} \quad & \mathrm{P_{1|1}} \geq \mathrm{P_{1|1}}. \end{aligned} \tag{7.3}$$

From Figure 7.1 and according to Bayes' rule, we have

$$\begin{aligned} \mathrm{P_1} &= \Pr(\mathrm{AI}(\tau) = 1) \\ &= \frac{\Pr(\mathrm{RTL}(\tau) = 1)\Pr(\mathrm{AI}(\tau) = 1 | \mathrm{RTL}(\tau) = 1)}{\mathrm{P_{1|1}}}. \end{aligned} \tag{7.4}$$

It can be concluded that in order to optimize the URC concept, there are in general two possibilities: either by improving the transmission scheme ($\Pr(\mathrm{RTL}(\tau) = 1)$) or by improving the estimation of the AI ($\Pr(\mathrm{AI}(\tau) = 1 | \mathrm{RTL}(\tau) = 1)$). The first possibility can be achieved by means of more robust modulation and coding techniques, whereas the second possibility can be accomplished through more accurate channel estimation and prediction techniques. On the other hand, it is interesting to note that the two probabilities $\mathrm{P_{0|1}}$ and

$P_{1|0}$ in Figure 7.1 correspond to the Type I and Type II errors[100] in statistical probability analysis:

$$P_{0|1} = Pr(RTL(\tau) = 0|AI(\tau) = 1) \quad (7.5)$$
$$P_{1|0} = Pr(RTL(\tau) = 1|AI(\tau) = 0). \quad (7.6)$$

We now explain the URC concept using a simple example based on the predicted SINR. Let us assume that the SINR for time slot τ is $\Gamma(\tau)$. We define that the transmission is successful for time slot τ if the actual SINR is larger than or equal to a given threshold Γ_l ($\Gamma(\tau) \geq \Gamma_l$), which is given by the selected modulation and coding scheme including the use of retransmissions. On the other hand, the AEI signals to the application AI=1 if the predicted SINR for time slot τ, $\Gamma_p(\tau)$, is larger than or equal to the threshold Γ_l ($\Gamma_p(\tau) \geq \Gamma_l$). Under these definitions, we can formulate the optimization problem of the URC concept as

$$\begin{aligned} \max \quad & P_1 = Pr(\Gamma(\tau) \geq \Gamma_l) \\ \text{s.t.} \quad & P_{1|1} = Pr(\Gamma(\tau) \geq \Gamma_l|\Gamma_p(\tau) \geq \Gamma_l), \end{aligned}$$

whereas the type I and Type II error can be formulated as follows

$$P_{0|1} = Pr(\Gamma(\tau) < \Gamma_l|\Gamma_p(\tau) \geq \Gamma_l)$$
$$P_{1|0} = Pr(\Gamma(\tau) \geq \Gamma_l|\Gamma_p(\tau) < \Gamma_l).$$

In this example, it would be possible to optimize the URC concept by decreasing Γ_l by means of more robust modulation and coding schemes, or by improving the computation of $\Gamma_p(\tau)$ in order to predict more accurately the availability of the RTL.

7.3.2 Problem Formulation, Data Generation & Preparation

In order to have an apriori link availability warning, we turn our attention to the problem of channel tracking and prediction. In this regard, we exploit the fault tolerance of the URLLC in order to predict the availability of RTL for a given time window. To this end, we frame the availability prediction as a time-series problem and use a RNN based on LSTM units for prediction. To demonstrate the performance of our proposed ANN architecture, we consider an LTE system and simulate multiple subframes passing through a fading channel and use the post equalized SINRs as training data. As a baseline, we consider a naive persistence model that takes the minimum SINR of the previous n transmissions and compares this to the current SINR value to make a decision about RTL. We would also like to emphasize that this approach could also be extended to 5G systems without any loss of generality.

In order to demonstrate the feasibility of our approach, we consider an example application in an underlying LTE network and use an SINR based threshold inorder to calculate the availability $P_{1|1}$. We assume a packet size of 7500 B and a maximum delay of 5 ms (prediction window). This corresponds to 12 000 bits of data per TTI of 1 ms. For the assumed bandwidth of 5 MHz, this corresponds to a transport block length of 12 576 bits using an MCS of QPSK with coding rate of 0.75 [188]. Then, we used the following strategy (also outlined in Figure 7.3) in order to calculate the predicted and actual SINR.

1. Create a LTE transmit resource grid with the data and reference symbols.

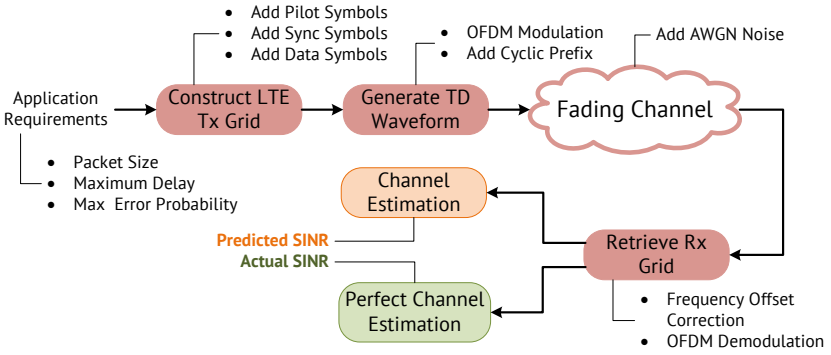


Figure 7.3: Data Generation for Training

Table 7.1: Simulation Parameters

Entity	Parameters	Value
Application Requirements	Packet Size	7500 bytes
	Max Delay	5 ms
LTE eNodeB	BandWidth	5 MHz
	PRBs	25
	Centre Freq	2.0 GHz
Channel	Type	Multipath Fading Channel [185] [186]
	Vehicle Speeds (Km/h)	[10,30,50,70,100,130]
Channel Estimation	Time Window	7
	Freq Window	19
	Interpolation	Cubic

2. Apply OFDM modulation and IFFT to obtain a Time Domain (TD) signal.
3. Pass the TD signal through a multipath fading channel [186] with varying dopplers.
4. At the receiver's end, apply frequency offset correction, FFT and OFDM demodulation to retrieve the received grid.
5. Estimate the channel using the methodology described in [185] to obtain the predicted SINR
6. Use perfect channel estimator in order to calculate the actual received SINR

Overall, for SNR ranges of (1,...,10) and various Doppler frequencies corresponding to vehicle speeds at a centre frequency of 2 GHz as shown in Table 7.1, we simulated a total of 10000 TTIs correponding to 2000 packets where each packet spans over 5 TTIs. In total we have $10 * 6 * 10000 = 600000$ TTIs (subframes).

RTL Model For the given transmission constellation, the minimum SINR required to ensure a successful transmission is 5.5 dB [89] and all 5 transmissions need to be successfully received inorder to guarantee the successful delivery of the complete packet. Assume Γ_i to be the received SINR of the i-th TTI, the condition for a successful transmission for time slot τ ($\mathrm{RTL}(\tau) = 1$) is

$$\min_{1 \leq i \leq 5} \Gamma_i(\tau) \geq 5.5.$$

AEI Model The AI mechanism determines AI by comparing the predicted value of the SINR for time slot τ, $\Gamma_p(\tau)$, with the threshold.

$$\mathrm{AI} = \begin{cases} 1, & \Gamma_p(\tau) \geq 5.5 \\ 0, & \text{otherwise.} \end{cases}$$

7.3.3 Model Architecture, Training & Evaluation

ANNs have emerged as one of the most powerful class of estimators that are used for both classification and regression problems where they progressively learn tasks by considering examples. However, traditional ANNs are unable to exhibit the dynamic temporal behavior of a time sequence. This can be accomplished by RNNs) which are a special class of ANNs that maintain their own internal state (memory) in order to process temporal sequence of inputs. An example of such a RNNs is the LSTM [79] that has set accuracy records in multiple applications domains.

An LSTM unit is composed of a cell, an input gate, an output gate and a forget gate and is responsible for remembering values over arbitrary time intervals (either long or short intervals and hence the name). The input gate controls the extent to which a new value flows into the cell, the forget gate controls the extent to which a value remains in the cell and the output gate controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit [79]. LSTMs are already being used in mobile networking for the purpose of predicting the channel [184], network traffic [15], spectrum occupancy [78] etc.

The availability prediction can be formulated as a time series forecasting problem where the ANN should predict the SINRs for the timeslot τ^+ based on the SINR values estimated by the channel estimation mechanism for the previous timeslot τ. Since each timeslot is 5 TTIs, we use the former 5 TTIs to predict the SINRs for the subsequent 5 TTIs. Assume that our input data is a series in the form $[X_1, X_2, X_3, ..., X_n]$ of n samples with a shape of $(n, 1)$. The sequence is then reshaped into the form $(n/5, 5)$ to represent $n/5$ individual packets each spanning over 5 TTIs.

Subsequently, the time series forecasting is converted into a supervised learning problem by creating multiple shifted copies of the timeseries so as to create 5 previous timeslots as training data (X) and 5 subsequent timeslots as training labels (y) in one sample as shown in Equation 7.7.

$$X = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \\ x_2 & x_3 & x_4 & x_5 & x_6 \\ . & . & . & . & . \\ . & . & . & . & . \\ x_{n-9} & x_{n-8} & x_{n-7} & x_{n-6} & x_{n-5} \end{bmatrix} \quad y = \begin{bmatrix} x_6 & x_6 & x_8 & x_9 & x_{10} \\ x_7 & x_8 & x_9 & x_{10} & x_{11} \\ . & . & . & . & . \\ . & . & . & . & . \\ x_{n-4} & x_{n-3} & x_{n-2} & x_{n-1} & x_n \end{bmatrix} \tag{7.7}$$

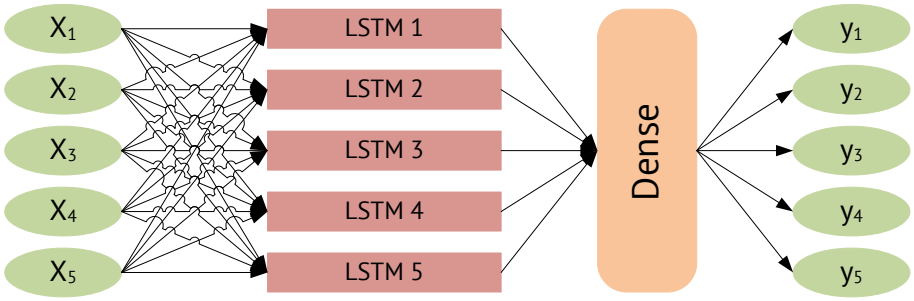
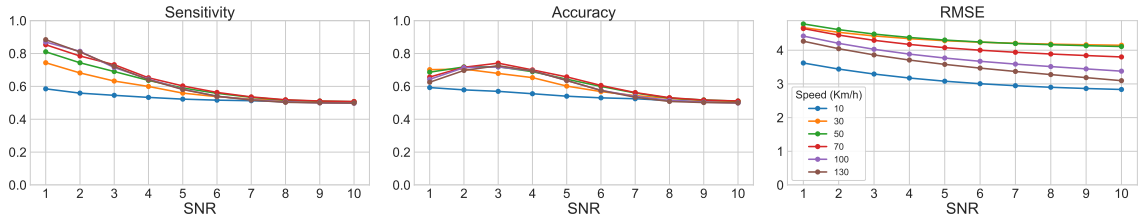


Figure 7.4: LSTM Architecture



(a) Sensitivity **(b)** Accuracy **(c)** RMSE

Figure 7.5: Naive Persistence based Prediction

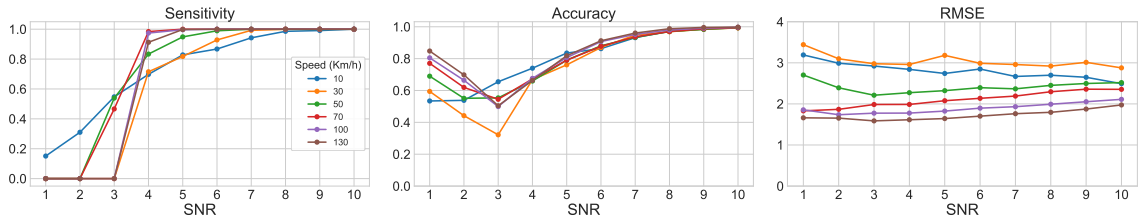
Once the data is formulated as a supervised regression problem, it is scaled fall in range[-1,1] due to the tanh activation function of the LSTM units. The data is then split into training (X_{train}) and testing (X_{test}) datasets in a ratio of 10% - 90% with respect to the total samples of X.

As shown in Fig 7.4, the LSTM used in this work is a simple model with an input layer, a 5-unit LSTM layer followed by a dense layer with 5 outputs totalling 170 trainable parameters. For prediction on X_{test}, a small modification had to be made. Since, in real world, an ACK/NACK is sent only after the successful transmission of the complete packet, the transmitter would not have access to the intermediate SINR predictions at the receiver. It has to solely rely on the 5 SINR predictions of the previous packet. Hence the input data is sampled to be continuous in such a way that it results in no repetition of data as shown in Equation 7.7.

The network is trained for 100 epochs on X_{train} using an ADAM optimizer and evaluated on X_{test}. The results are discussed in the next section.

7.3.4 Results and Summary

Figure 7.6 shows the results when LSTM predictions are compared with the actual received SINRs from perfect channel estimation. It can be seen that, irrespective of the Doppler frequency, it is possible to satisfy the ultra-reliable requirement at SNR $>$ 5 dB. Quite expectedly, LSTM based prediction outperforms the Naive Persistence model (Figure 7.5) in terms of all the three metrics. The accuracy of LSTM prediction follows a similar signature as the channel estimator. However, the Root Mean Square Error (RMSE), though low does not exhibit a predictable pattern. On the other hand, high doppler frequencies seem to have a positive effect on the estimator sensitivity which is difficult to explain given



(a) Sensitivity (b) Accuracy (c) RMSE

Figure 7.6: LSTM based Prediction

the lack of clarity in understanding how ANNs train. With more detailed analysis, this may become clearer and is left as an open topic as of now.

Overall, it can be seen that the LSTM models can be suitable for network availability prediction even at high vehicle speeds. The work presented in this paper only scratches the top of immense possibilities that ML offers especially at the PHY layer. As a future work, the following analysis can be carried out

1. For each SNR and Doppler frequency, a single LSTM model is trained which is rather inefficient and does not offer the possibility to completely exploit their full potential. As a possible extension, a more generic model can be trained by framing the prediction as a multivariate LSTM forecasting problem with SNR and doppler as extra features. Alternatively, multiple LSTM networks (for different SNRs and dopplers) can be stacked over each other to compile one single model for all predictions.
2. Adding multiple LSTM and dense layers and experimenting with number of epochs, neurons etc.
3. The current work considers only the fading channel as explained in [187, 185] along with a static packet size of 7500 B and a maximum delay of 5 ms. These parameters can also be made variables inorder to investigate their effects on prediction quality.
4. The data used for training the ANN and the subsequent analysis is simulated data. It would also be interesting to see if the LSTM performs the same in real experiments.

8 ML based Obstacle Detection using UWB RADAR

Short Range wireless devices are becoming more and more popular for ubiquitous sensor and actuator connectivity in industrial communication scenarios. Real-time ranging & localization is an important requirement for *Industry 4.0* scenarios such as Industrial Machine type communication, V2V communications etc. that rely on the relative proximity of nodes to each other. Applications such as Autonomous Guided Vehicles (AGVs), Autonomous Train Pairing (ATP) [159] etc. additionally require that the devices scan the environment and detect any potential humans/obstacles and stop if necessary to avoid a collision. Obstacle detection is typically done by means of two classes of sensor systems - passive and active systems [46]. Passive systems use the available natural forms of energy (e.g.,visible light) whereas active systems provide their own energy for detection. Examples of passive systems include include vision systems such as visible light and infrared cameras while examples of active systems include RADAR, Laser Detecion and Ranging (LADAR), Light Imaging Detection And Ranging (LIDAR) and SOund Navigation and Ranging (SONAR) etc.

Vision systems use video sensors with a sufficiently high resolution both horizontally and vertically. The detection is generally done by means of three main steps; object segmentation, classification and tracking. Segmentation involves extracting features of the objects from the background including depth and motion information (stereoscopy and stereo matching). From the obtained features, models are constructed that identify the patterns from the features and classify them accordingly. Finally, tracking mechanism keeps track of the classified object across multiple frames and uses different algorithms to predict its motion trajectory. Vision based systems provide advantages such as ease of use, no interference in addition to their low price. However, their performance is severely affected by lightning conditions and shading. Infrared vision systems have been proposed as an alternative due to the fact that all heated objects emit infrared radiation that can be registered with an infrared camera. Since some classes of objects (pedestrians, vehicles) have a specific infrared signature (pedestrian's head, body, legs respective vehicle's wheel and engine), positive object identification can be made based on the received energy. However, infrared sensors are sensitive to weather conditions and infrared is still a new domain for obstacle detection applications.

Active systems emits radiation in a series of pulses from an antenna. When the energy reaches the target, some of the energy is reflected back toward the sensor and this radiation is detected, measured, and timed. The time required for the energy to travel to the target and return back to the sensor determines the distance or range to the target. In case of RADAR, radio waves are used as radiation sources whereas systems much as LIDAR and LADAR use light/laser pulses. On the other hand SONAR uses sound waves as radiation source and since the speed of sound varies depending on the pressure and temperature of the medium, it is not well suited for ground based applications.

Sensor fusion is the process of combining inputs from different sensors so as to increase the detection accuracy. It also helps leverage the advantages offered by using one particular technique to compensate the limitations of another. Sensor fusion can be done both at the high level or low level. In high level fusion, the detection ans processing of objects is done at the sensor level whereas in low level fusion, all the raw sensor data is centrally accumulated and the processed further to detect objects. Both the approaches have their respective merits with low level fusion offering advantages such as better detection due to availability of big data pool whereas high level fusion is more efficient in terms of computational load and communication resources [60].

In this section, we investigate the performance of different ML algorithms for classifying the severity of obstacles using the received signal from a pulsed UWB RADAR. In the following, we describe the experiment setup including data generation, construction of hypothesis, selection and validation of ML models.

8.1 Problem Statement, System Model and Experiment Setup

A signal is characterized as UWB signal if it has a fractional bandwidth (η) $\geq$ 25 % of its center frequency or 500 MHz when the frequency is above 6 GHz. The formula for calculating η is given as [149]

$$\eta = \frac{f_h - f_l}{f_h + f_l} * 100$$

where f_h and f_l represent the highest and lowers frequencies which are 10 dB below the maximum respectively. The advances in UWB technology allowed the development of low-cost RADAR systems with multiple fields of applications. UWB uses very short pulses with pulse widths in tha range of nanoseconds to a few tens of picoseconds, thereby spanning over several GHz of bandwidth and offering resolutions in the range of few cm. They also offer high sensitivity in the sense that they can even detect variations as small as a heart beat or breathing rate and have extensive applications in the fields of Medical, Building, Surveillance, Security and Monitoring. It is also fit for obstacle/collision detection applications as evident from 24 GHz short-range RADAR that uses a combination of UWB RADAR and a conventional Doppler RADAR to measure vehicle speeds and detect obstacles within a range of 10 cm to 30 m.

In conventional RADAR, the range resolution Δd of a target detection is given by

$$\Delta d = \frac{c}{2 * BW} = \frac{\tau * c}{2}$$

where c is the speed of light, BW is the bandwidth of the RADAR pulse in frequency domain and τ is the width of the RADAR pulse in time domain. It can be seen that due to the large bandwidth offered by UWB (and conversely, short τ), it exhibits very high range resolution and hence, it is highly suitable for short range ranging, detection and localization applications. During operation, the pulses emitted by the UWB radar impinge upon different objects and part of the electromagnetic energy if reflected back to the UWB radar. Therefore, the time delay Δt between the transmitted and received signals is obtained which is then used to calculate the distance to the object as follows

$$d = \frac{\Delta t * c}{2}$$

8.1.1 Motivation

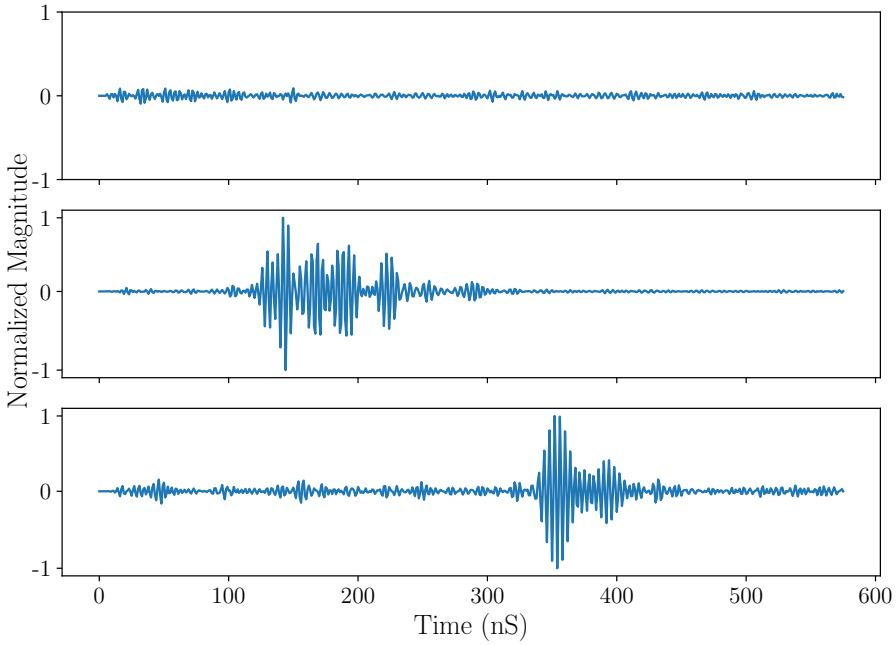
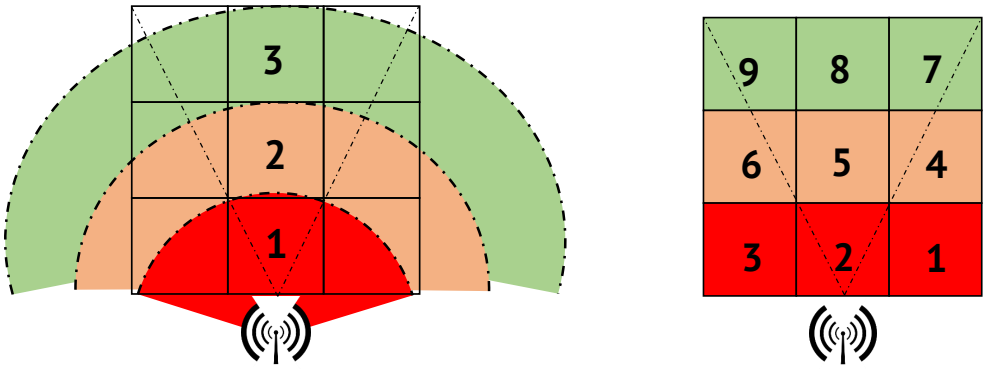


Figure 8.1: UWB signals in time domain without and with a person. The first picture shows an empty environment with no obstacles whereas the second and third figures correspond to the received waveforms with a person standing directly in front.Notice the delay in reflected pulse as the person moves further

Figure 8.1 shows the UWB captured UWB waveforms that are passed through a second order difference filter in order to remove the noise. The horizontal axis indicates time in ns relative to the arrival time of the direct path and the vertical axis indicates response magnitude and polarity. The advantages offered by UWB in terms of obstacle detection and ranging are clearly visible in the figures. From the figures, it is evident that in theory, it is possible to detect an object and calculate the distance to it simultneously, which serves the purpose of obstacle detection. However, the following problems arise

1. When there are multiple objects in the environment, the received waveform is a superimposed version of the reflected waves from all these targets. In this case, additional processing is necessary to resolve these multiple targets.
2. Not all the detected objects need to be considered as obstacles. Some of these objects may be part of the detection device and hence can omitted as potential obstacles. Additionally, a subset of these obstacles might be closer to the RADAR in terms of distance than real obstacles. This would result in the received waveforms registering *peaks* corresponding to these objects which are otherwise never deemed to be obstacles. Therefore resolving these different objects from the received waveforms becomes a complex process and requires that the radio silhouette of all the objects considered safe be modeled and filtered out.
3. Even the radio silhouette of the same object can be different depending upon different propagation environments. This is especially true for moving objects (such as humans) where the reflected signal's profile is dependend on the order of reflections form the moving object.



(a) Multi-class Labeling with 4 labels

(b) Grid Labeling with 10 labels

Figure 8.2: Data Labeling

A careful look at these problems shows that it is important to understand the reflected waveform profile (radio silhouette) of the obstacle of interest so as to filter it out from the rest of the received waveform and calculate the distance to it. This is an impractical approach and also requires to model both the kinematics of the moving obstacles and the radio environment in detail. Another practical way is by defining this as a classification problem as using the ML algorithms that have proven to tackle such kind of problems with relative ease.

8.1.2 Construction of Hypothesis

The hypothesis of obstacle presence/absence can be constructed in multiple ways depending upon the desired detection resolution. The simplest scheme is a binary classification scheme where there are only two labels - 0 and 1. 0 indicates the absence of the obstacle of interest and 1 indicates the presence of obstacle of interest.

$$H = \begin{cases} 0, & \text{If there is no person} \\ 1, & \text{If there is a person} \end{cases}$$

Though, this method gives us information about if a person is present or not within the sensing range of the UWB RADAR, it does not give us any information regarding the likelihood of collision. Moreover, it also fails to exploit the time domain resolution offered by UWB RADARs to determine the distance of the target. In order to do so, we need to define some more labels and the likelihood of collision associated with that label. The second approach is a 4-class Radial classification where we divide the area in front of the UWB transmitting antenna radially into multiple regions (labels) differentiated by their radii as shows in 8.2a. The hypothesis can be constructed as follows

$$H = \begin{cases} 0, & \text{If there is no person} \\ 1, & \text{High Risk person} \\ 2, & \text{Medium Risk person} \\ 3, & \text{Low Risk person} \end{cases}$$

This hypothesis is suitable for UWB RADARs that sense radially with omnidirectional antennas.Finally, we can also adapt a grid based approach for multi-class labeling as shown in 8.2b. This is particulary useful when we want to know the distance as well as direction

of the target of interest. In this case, each grid corresponds to an independent label for the training data. So for a $3X3$ grid, there are 10 labels including label 0 for no person.

$$\mathrm{H} = \begin{cases} 0, & \text{If there is no person} \\ 1, & \text{Person in Grid 1} \\ .. & .... \\ 9, & \text{Person in Grid 9} \end{cases}$$

8.1.3 Data Collection and Labelling

For the purpose of data collection, a UWB RADAR namely, PulsON P440 module from Time Domain systems has been used [140]. It supports multimode operation in the sense that it can be used as a ranging RADAR (with Time of Flight (ToF) ranging), monostatic or bistatic RADAR. Here, we used the module in monostatic RADAR mode where the UWB pulses are sent and received using the same module in a time-slotted approach. For the purpose of generalization, the measurement campaign was conducted for both indoor and outdoor scenario where the raw received UWB were collected and labeled according to the schemes shown in Figure 8.2. For the radial labeling scheme the measurement area consisted of a 3 concentric half circles with a step radius of 1 m between each concentric circle and for the 10-class grid labeling, we considered a 3X3m grid (Figure 8.2).

Inline with the guidelines specified in ML literature, independent training and testing datasets were recorded during the data collection phase. This resulted in training data (X), training labels (y), testing data (X_{test}) and testing labels (y_{test}) respectively. All the data sets were preprocessed by normalizing them around zero center as follows

$$X_n = \sum_{i=1}^{N} \left(\frac{X_i - \frac{1}{N}\sum_{i=1}^{N} X_i}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_i - \mu)^2}} \right)$$

The datasets X is further split in the ratio of 10 % to 90 % into training (X_{train}) and validation (X_{valid}) sets respectively with y_{train} and y_{valid} as the corresponding labels. This set is used for selecting the best estimator.

8.1.4 Model Selection, Training & Evaluation

The hypothesis is constructed as a supervised learning problem and there are many algorithms designed for this purpose. The list of evaluated estimators along with their hyper-parameters is outlined in Table 8.1. Hyper-parameters are the tunable parameters of the estimator (e.g., regularization parameter for linear models, number of neighbors for knearest neighbors etc and their selection visibly effects the model's performance.

In order to select the optimal parameters, we used exhaustive search over the grid space (Table 8.1) using X_{train} and K-Fold cross validation. For each instantiation of the ML classifier with a given hyper-parameter set, we divide X_{train} into 5-folds (5 was chosen randomly) and preserving the ratio of labels in each fold. Hence, each training operation lasts 5 iterations where for each iteration, the model is tested on each fold by training on the other four folds. For the performance comparison (scoring), we used classification accuracy which is the number of correct predictions made divided by the total number of predictions made, multiplied by 100 to turn it into a percentage.

Table 8.1: List of estimators and hyper-parameter grid

Name	Underlying Model	Parameters	Values
Logistic Regression	Linear	Regularization Parameter (C)	[0.001, 0.01, 0.1, 1, 10, 100, 1000]
		Solver	[lbfgs, sag, newton-cg]
Perceptron	Linear	Regularization Parameter (Alpha)	[0.0001, 0.001, 0.01, 0.1, 1]
K-Nearest neighbors	Nearest neighbor	Number of neighbors to consider (N)	[1, 2, 3, ..., 30]
Linear SVC	Support Vector Machines	Regularization Parameter (C)	[0.001, 0.01, 0.1, 1, 10, 100, 1000]
Decision Tree	Tree Based	Splitting Quality Measure	[gini, entropy]
		Max_features to consider for splitting	[auto, sqrt, log2]
Random Forest Classifier and Extra Trees Classifier	Tree Based Ensemble	Number of estimators (n)	[16, 32, 64, 128, 256]
		Splitting Quality Measure	[gini, entropy]
		Max_features to consider for splitting	[auto, sqrt, log2]
Gradient Boosting Classifier	Tree Based	Number of estimators (n)	[16, 32, 64, 128, 256]
		Learning Rate	[0.2, 0.5, 0.8, 1.0]

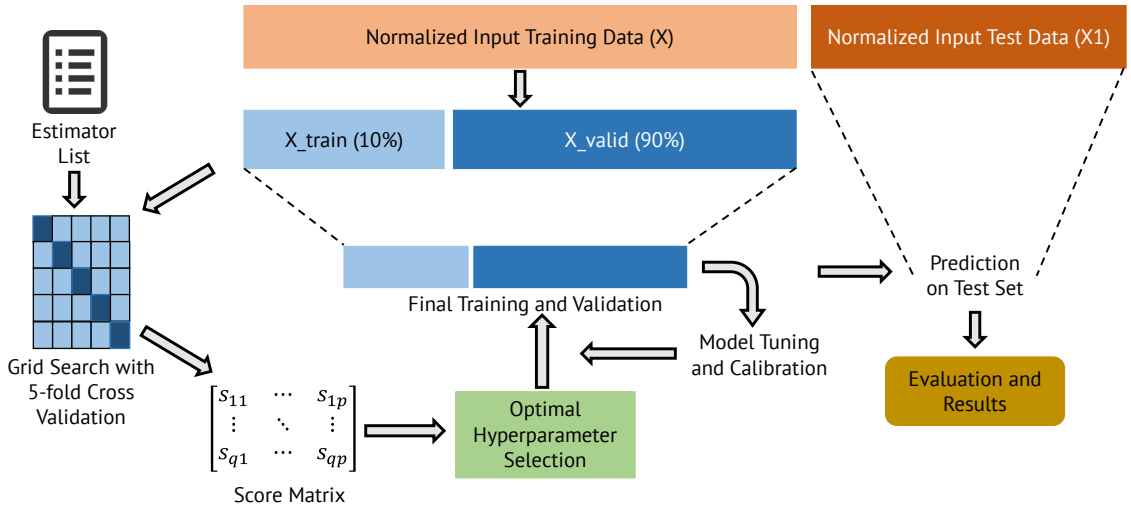


Figure 8.3: Model Selection Method

We used the following strategy to select the optimal hyperparameters. For every instantiation of the estimator, there are $k = 5$ scores corresponding to each fold and as associated scores array s. We selected the minimum s_{min} resulting in an array $(S_{min,1}, S_{min,2}, ..., S_{min,n})$ where n is the number of the estimator's parameters in Table 8.1. The optimal estimator's hyper-parameters set is then selected by picking out the indices $(argmax)$ of the maximum value S_{max} from S.

The selected model is once again trained on the whole training data X_{train} (since we omitted one fold per training operation during grid search). The trained model is then used for prediction on the validation set (X_{valid}). Finally, the model is used to predict the labels on the test set X_1 so that we have the validation and testing performance for performance comparison.

8.1.5 Results & Conclusions

Figure 8.4 shows the accuracy scores of all the considered estimators for the exhaustive grid search.The top figure shows the results for indoor scenario and the bottom figure shows the results for outdoor scenario. In general, it can be seen that the all of the methods perform very well with accuracies for all hyperparameter combinations exceeding 95 % for

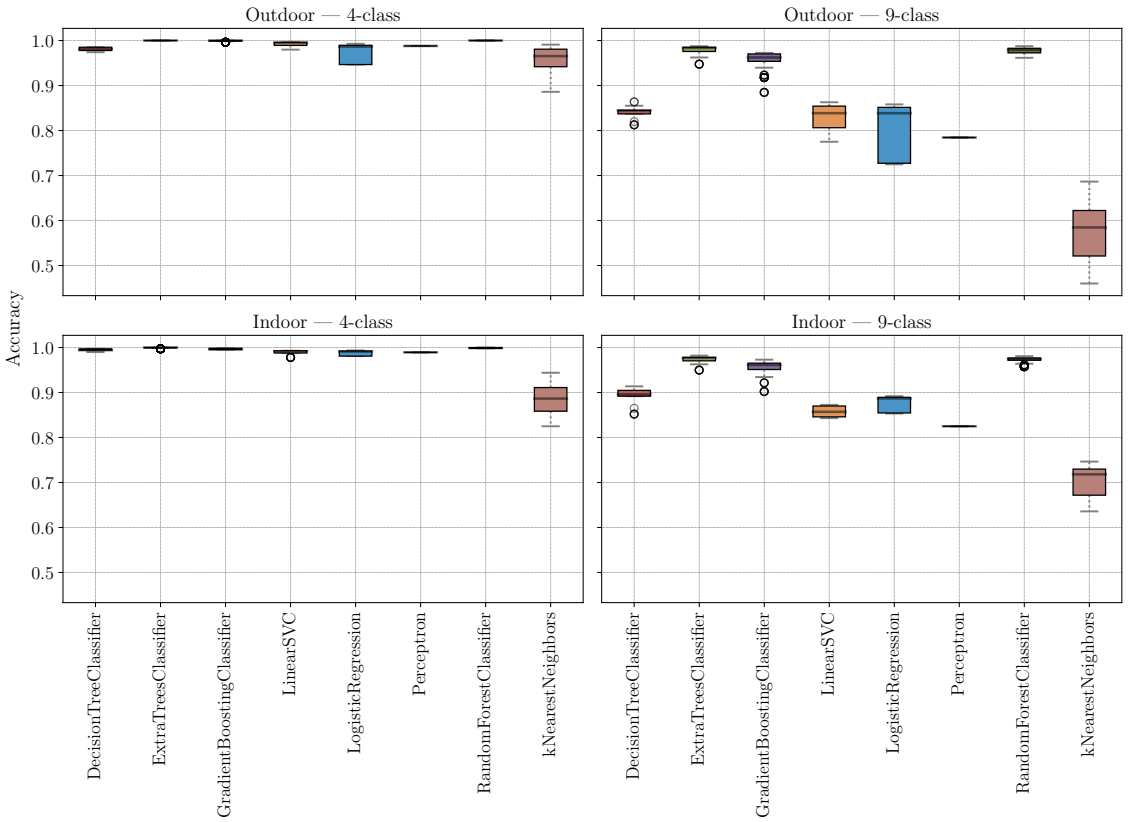


Figure 8.4: Accuracy Scores for Grid Search for all the considered estimators

the simple 4-class classification. However, for grid classification with 9 labels, the scores drop significantly with the exception of Random Forest Classifier that still manages to score close to 95 %.

Figure 8.5 shows the validation and testing accuracy of all the estimators selected with the optimal hyperparameters. Evidently, the simple 4-class approach shows higher accuracies than 9-class labeling. This is due to the ambiguity of the radar to distinguish similarly spaced targets that are on the left to those targets to the right. This problem can be mitigated using multiple directional antennae which does at better job at distinguishing the directionality of the obstacle compared to the omnidirectional antenna.It can also be seen that the indoor scenario shows a relatively higher degradation of the validation and testing scores. This can be attributed to the rich multipath environment in the indoor scenario in which case the model fails to generalize well.

Based on the overall performance of all the scenarios and labeling types, Random Forest Classifier does the best job at the prediction task, followed by linear models such as Perceptron and Linear SVC. However, k-Nearest Neighbors shows the most consistent performance with little difference between validation and testing accuracies thereby leaving room for further improvement by further training.

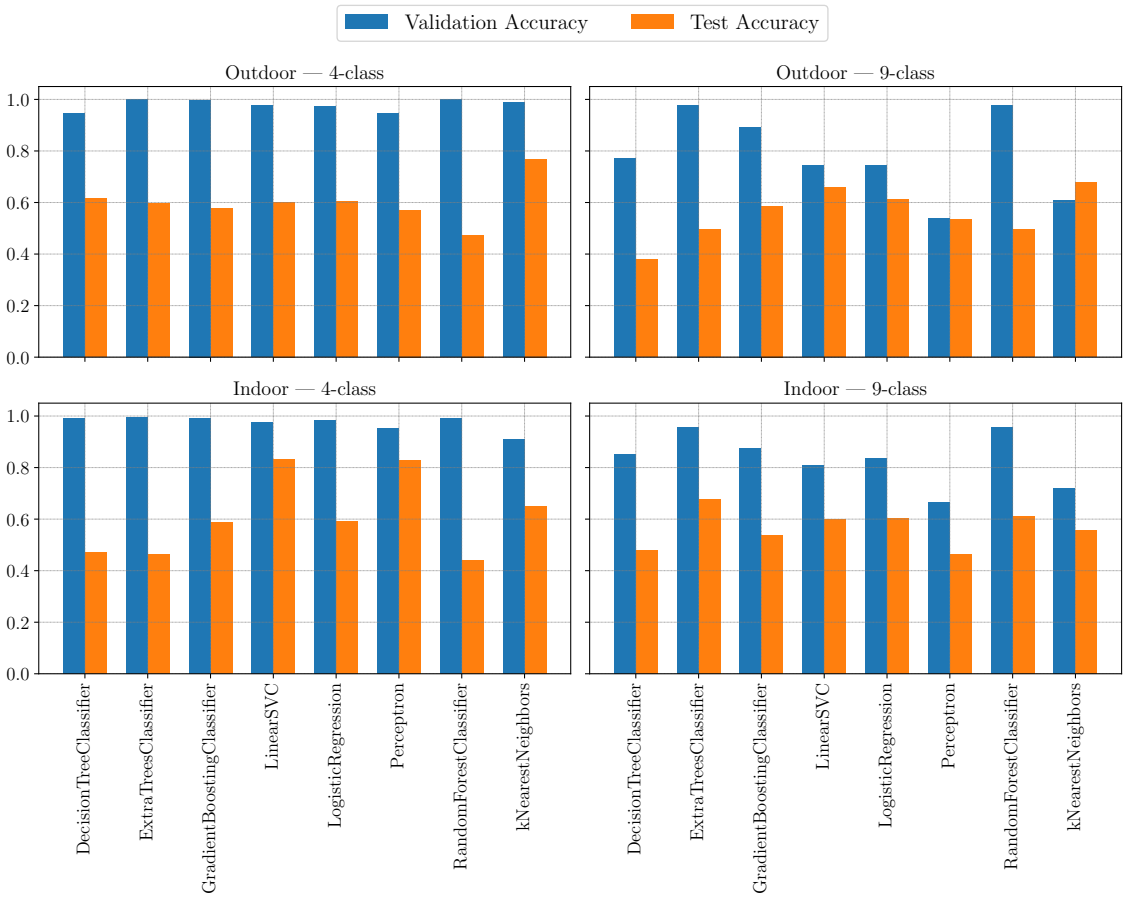


Figure 8.5: Validation and Testing Accuracy Scores for all the considered estimators

9 Summary and Conclusions

V2X communications will play a crucial role in the 21st century in increasing the overall traffic safety and efficiency. By exchanging information between vehicles and other traffic entities, unexpected events such as sudden/harsh braking can be communicated in real-time and accidents can be avoided. Moreover, with a global knowledge of multi-modal traffic entities on the roads, it becomes possible to manage the traffic more efficiently. The primary enabler for realizing such connected mobility is the underlying communication technology. The choice of communication technology must take into account the stringent automotive requirements owing to the safety criticality of the applications. It should be able to guarantee very high levels of reliability and low latencies as well be available under diverse propagation conditions including high mobility scenarios. In this regard, two technologies exist that aim to address the performance requirements for connected mobility - one based on the IEEE Wi-Fi standard named ITS-G5 and the other based on the 3GPP standard named C-V2X. A detailed explanation of both these technologies is given in chapter 2

Before deploying any technology in the real world, it is important to understand its performance and theoretical bounds. This is typically done using analytical methods, simulations and hardware demonstrations. In this thesis, we deal with the simulative approach where the performance of both the technologies is extensively compared at both the link and system level. Since ITS-G5 is a matured technology, many simulation frameworks exists already for evaluating its performance. On the other hand, C-V2X is relatively new technology (first introduced in 2015) and therefore not many frameworks exist. However, C-V2X is based on the well established 3GPP standard that can be realized by making modifications to the existing simulation frameworks. Therefore, a new link level simulation framework is developed in Python for evaluating the performance of C-V2X that supports all the different sidelink logical, transport and physical channels. The implementation details can be found in chapter 3

The choice of channel model plays a crucial part in the link level simulation since it replicates, to some extent, the propagation conditions of the real world as perceived by the receiver. There are basically two types of propagation models - physical and non-physical models. Non-physical models are based on the statistical description of the channel using non-physical parameters such as signal correlation whereas physical models take into account the locations, electromagnetic properties of scatterers or physical description of rays [41]. In this thesis, we use the Tapped Delay Line (TDL) models where the power delay profile is described by a limited number of paths and each path is characterized by a relative amplitude, delay and a Doppler spectrum. For vehicular environments, two model families exist - one from the ITU and the second from the DSRC Tiger team. Both the model families were derived after extensive field trials and are used frequently for evaluating performance of various RATs for vehicular environments.

The link level simulation results were conducted for both the AWGN model and the TDL models and the results can be found in chapter 3.The results show that,in single transmis-

sion scheme, C-V2X outperforms ITS-G5 in almost all of the considered channel models with some exceptions with 16QAM and higher coding schemes. With one blind retransmission enabled, C-V2X exhibits a gain of atleast 6 dB and in some cases, reaching as high as 10 dB over ITS-G5. The superior performance of C-V2X can be attributed to two factors - 1. The use of turbo coding that is better than the convolutional encoding scheme used in ITS-G5 and 2. The channel estimation mechanism where the pilots are more evenly distributed across the time-frequency grid compared to ITS-G5 where the pilots are only present in the preamble. This effectively translates to better averaging performance using the LS estimator.

chapter 4 describes the MAC level schemes of both the technologies in detail and also presents system level performance where the MAC schemes were evaluated in terms of overall system capacity. It is usually done by means of deploying multiple nodes and using the link to system level mapping tables obtained from the link level simulations in order to define a threshold SINR that is used for calculating the overall PER. For this purpose, the system level simulator named V2X simulator, an open source implementation in Matlab developed by university of Bologna was used [29]. The two MAC schemes namely CSMA/CA used in ITS-G5 and SPSS used in C-V2X were evaluated for different scenarios namely highway, manhattan grid and real world maps. Using the bidirectionally coupled approach where the network and road traffic simulators (SUMO) are used in a time-slotted approach, vehicular traffic was generated for all these scenarios for the simulation. The network simulator then calculates the overall PER using the vehicle locations and macroscopic channel models. The result show that C-V2X has a higher range than ITS-G5 inline with the specification.This can be seen in the form of PER where ITS-G5 maxes out before C-V2X. In terms of distance, this translates to a range gain of almost 100 m for QPSK coding schemes and almost 200 m for 16QAM schemes. C-V2X also makes use of spectrum efficiently compared to ITS-G5 which can be seen in the form of higher CBR. The retransmission gain with C-V2X is significant at higher MCS, lower vehicular densities and higher speeds.

Traditional Wireless Network optimization was done by means of acquired domain knowledge in the form of mathematical models / heuristics and designing solutions / algorithms based on this knowledge. This approach, though perfected over the years, will be insufficient when dealing with future hyper-connected society and anticipated services such as autonomous driving, eHealth, Industrial Communication etc. This can be due to *model deficit*, where no physics-based mathematical models exist for the problem due to insufficient domain knowledge or due to *algorithm deficit*, where even though a sound mathematical model is available, the existing algorithms optimized on the basis of this model are too complex to be implemented [168]. Moreover, due to static modelling on predefined scenarios, algorithms designed using this approach cannot react to dynamic data demands, connectivity problems and/or hardware failures. SONs aim to address this problem by adding functions relating to network / resource reconfigurability, automating operations and management. However, they are only limited to specific RAN applications without providing a true end-to-end solution.

It can be said that the future networks need to have intelligence that is spread throughout the infrastructure. In this regard, a promising solution comes in the form of AI/ML where, instead of relying on domain knowledge, a learning flow is used by means of using sufficiently large examples of desired behavior for the algorithm of interest. In this regard, chapter 5 presented the comprehensive state of the art analysis of the current applications of ML at different layers of the network protocol stack. After motivating the user to the

benefits of using ML at the PHY layer, the subsequent chapters deals with demonstrating the ability of various ML models in performing different signal processing tasks.

chapter 6 presented the use of ML as autoencoders that can learn the end-to-end behavior of traditional signal processing blocks by using sufficiently large number of samples. For this purpose, various DL models were presented and used for operations of channel coding and channel estimation in C-V2X. Specifically, RNNs were used for decoding of turbo codes due to their ability to learn temporal dependencies and CNNs were used for channel estimation due to their spatial dependency learning characteristic. The results show the with little training effors, these DL models can completely learn the behavior and can be used in place of the legacy blocks. For the case of turbo decoding, the results show that the RNN model outperforms the conventional turbo decoder (for all decoding iterations) for low SNRs ($<$0.4 dB). However, at higher SNRs, the turbo decoder's BER drops down exponentially while the drop is only linear for the RNN decoder. For the case of channel estimation, it can be clearly seen that the ANN based channel estimation scheme performs on par with the LS scheme at low speeds and low SNRs. The real benefits of using ANN scheme become apparent at higher speeds and higher SNRs as the proposed scheme outperforms LS scheme by almost an order of magnitude. This is because, at higher speeds, the averaging and interpolation used in LS causes excessive information loss thereby resulting in pure noise. In contrast, ANN was better able to learn the quick channel variations in high speed scenarios.

chapter 7 introduced the concept of reliability for wireless communication by means of a novel metric called *Availability* that determines the presence/absence of link reliability. This approach is motivated by the fact that a transport- agnostic definition from the applications point of view is required in order to allow URC services to be deployed in a wide range of scenarios. In order to have an apriori link availability warning, we turn our attention to the problem of channel tracking and prediction. In this regard, we exploit the fault tolerance of the URLLC in order to predict the availability of RTL for a given time window. To this end, we frame the availability prediction as a time-series problem and use a RNN based on LSTM units for prediction. The results show that, irrespective of the Doppler frequency, it is possible to satisfy the ultra-reliable requirement at SNR $>$ 5 dB. Quite expectedly, LSTM based prediction outperforms the Naive Persistence model (Figure 7.5) in terms of all the considered metrics.

Finally, the ability of ML based algorithms to detect the presence/absence of obstacles using time domain UWB waveforms has been demonstrated in chapter 8. By using the time domain P440 UWB module in monstatic radar mode, data was collected for both indoor and outdoor environments with and without human obstacles. Two kinds of classification problems were formulated with 4 and 9 classes respectively depending upon the position of the human obstacle. Different supervised learning algorithms ranging from linear models to more complicated emsemble models were selected and evaluated on the training data. The completew ML pipeline including data preparation, hyperparamater optimization were performed to get the best possible parameters for the considered estimators. The results show that tree based methods achieve classification accuracies close to 95 % on the raw UWB waveforms. With further data, these estimators can be made better and can be suitably applied to various scenarios where human detection is necessary.

A PHY Layer of ITS-G5

A.1 ITS-G5

In this section, the PHY layer of ITS-G5 and the functions associated with it are described in detail. Similar to 802.11a, the PHY layer of 802.11p is divided into two sub-layers namely Physical Layer Convergence Protocol (PLCP) and Physical Medium Access (PMD). PMD is the lowest layer that interfaces directly with the wireless medium and provides functions for modulating & demodulating the data, multiplexing the modulated symbols using OFDM and creating a time domain waveform by taking the IFFT of the multiplexed data. The PLCP acts as an interface between the MAC layer and PMD interacting with both of them by means of primitives. Its function is to act as a convergence process that transforms the MAC Protocol Data Unit (MPDU) (also referred to as PLCP Service Data Unit (PSDU) at PHY) arriving from the MAC layer to compose an PPDU frame.

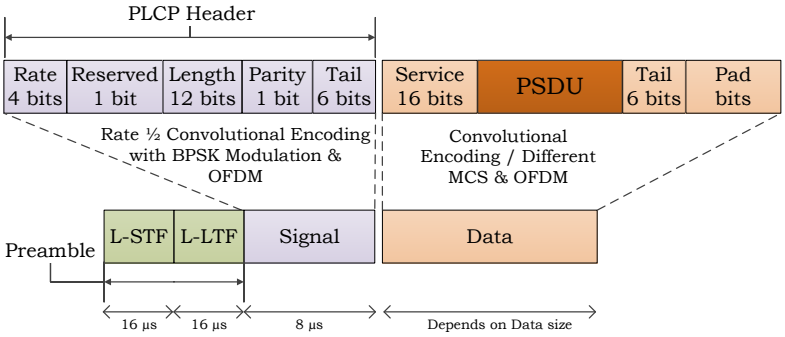


Figure A.1: PPDU Frame Structure

Figure A.1 shows the operations performed on the PSDU by the PLCP upon it's reception from the MAC layer. The PSDU is first prepended with *service* field and appended with tail and pad bits to form the *data* field of PPDU frame. Concurrently, a PLCP header is added that contains packet information such as the coding rate used and length of the PSDU. In the next step, the binary bits in the PSDU + header are converted into symbols by means of channel coding and modulation. The header is encoded using a rate 1/2 convolutional encoder and is BPSK modulated to get the symbols in the signal field. The PSDU is convolutional encoded and modulated with the desired MCS (Refer to Table A.1) to obtain the symbols in the data field. A PLCP preamble is then prepended to the header and data symbols that consists of 10 short symbols (Legacy Short Training Field (L-STF)) and 2 long symbols (Legacy Long Training Field (L-LTF)). L-STF is used for packet detection, coarse frequency correction and automatic gain control whereas L-LTF is used for fine frequency correction, symbol timing offset correction and pilot based channel estimation.

Table A.1: 802.11p - MCS & Data rates

Modulation	Coding Rate	Coded bits per OFDM symbol	Data bits per OFDM symbol	Data Rate (Mbit/s)
BPSK	1/2	48	24	3
BPSK	3/4	48	36	4.5
QPSK	1/2	96	48	6
QPSK	3/4	96	72	9
16-QAM	1/2	192	96	12
16-QAM	3/4	192	144	18
64-QAM	2/3	288	192	24
64-QAM	3/4	288	216	27

A.1.1 Baseband Signal Processing

This section details the operations that are necessary to convert the binary user data into symbols that can be readily transmitted over the wireless channel using 802.11p.

A.1.1.1 Transmitter Operations

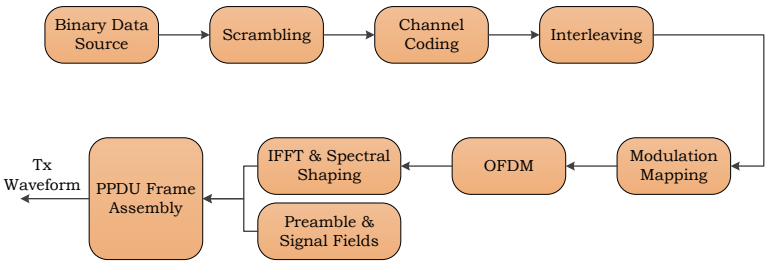


Figure A.2: Tx Operations for 802.11p

The user data in the form of binary bits is first scrambled in order to randomize the data pattern. A scrambler transforms the input binary sequence into another sequence by means of addition or multiplication with a pseudo random binary sequence. It has many uses such as reduced burst error, energy dispersal by means of bit redistribution in case the input binary sequence contains a long continuous sequence of 0s or 1s. In case of OFDM, this translates to reduction in PAPR. The scrambler in 802.11p uses the generator polynomial $S(x)$ defined as follows

$$S(x) = x^7 + x^4 + 1 \tag{A.1}$$

Channel Coding & Modulation Mapping Channel coding, also known as Forward Error Control Coding (FECC), is a process of detecting and correcting bit errors in digital communication systems by means of adding extra bits (parity bits) to the original data stream [56]. This introduces redundancy to the input data stream in a controlled manner that is used to combat the channel effects. The scrambled data output from the previous step is encoded using a convolutional encoder with a base coding rate of 1/2. This means that at every time index, the encoder takes on bit as input and outputs 2 bits as output. The

associated trellis structure for the convolutional encoder is represednted by the generator polynomials $g_0 = 133_8 = [01011011]_2$ and $g_1 = 171_8 = [01111001]_2$.

Apart from the base coding rate of 1/2, 802.11p also supports coding rates of 2/3 and 3/4 for the available MCS schemes (Table A.1). This is achieved by means of puncturing that periodically deletes certain code bits in the sequence encoded with base coding rate. The bits are deleted according to the given puncture vectors. Puncture vector is a pattern of 1s and 0s, with 0 indicating stolen bits, which is deleted from the base code. The standard defines following puncture vectors p0 = [1110] and p1 = [110110], in order to achieve rates 2/3 and 3/4 respectively.

An interleaver is a hardware device that takes symbols from an fixed alphabet as the input and produces the identical symbols at the output in a different temporal order. The classical use for interleaving is to disperse sequences of bits in a bit stream so as to minimize the effect of burst errors introduced in transmission [10]. The 802.11p PHY specified the use of block interleavers which are implemented by writing bits in to the matrix row by row and then reading them column by column. The standard also defines two permutations for interleaving. The first permutation ensures that adjacent bits are modulated onto nonadjacent subcarriers and the second permutation ensures that the adjacent bits are mapped alternatively onto less and more significant bits of the constellation. The block size is always equal to number of code bits in a single OFDM symbol, which depends on the size of the signal constellation.

Let k denote the index of the bit before first permutation, i be the index of the bit after first permutation an before second permutation and j be the index of the bit after second permutation. The first permutation can then be defined as

$$i = \frac{N_{CBPS}}{16}k \mod 16 + \lfloor \frac{k}{16} \rfloor, \quad k = 0, 1, 2, ...N_{CBPS} - 1 \tag{A.2}$$

where N_{CBPS} is the number of coded bits in one OFDM symbol. The second permutation defined intra-column permutations, which depend on the block size and is given as

$$j = s\lfloor \frac{i}{s} \rfloor + (i + N_{CBPS} - \lfloor \frac{16.i}{N_{CBPS}} \rfloor) \mod (s), \quad i = 0, 1, 2, ...N_{CBPS} - 1 \tag{A.3}$$

where s is the parameter dependent on N_{CBPS} and is calculated as

$$s = \max(\frac{N_{CBPS}}{2}, 1) \tag{A.4}$$

802.11p supports 4 different modulation schemes - BPSK, QPSK, 16QAM and 64QAM with varying data rates as outlined in Table A.1. The coded bits to symbol mapping is performed according to Gray-coded constellation mappings. To achieve equal average symbol power, the symbols are multiplied with a normalization factor k_m.

OFDM & IFFT The IEEE 802.11 specification specifies the arrangement of 64 SCs, with symbol duration of 8 µs. Out of these 48 SCs are used for actual data transmission, 4 SCs are used for transmitting pilot symbols and the remaining 12 SCs are used as null carriers carrying no information. The modulated data is divided into groups of 48 complex symbols which are indexed logically from 0 to 47. These are then mapped to SC indices -26 to 26, skipping SCs -21, -7, 0, 7 and 21. SC indices -21, -7, 7, 21 are used for mapping the 4 pilot symbols whereas SC is used as a null Direct Current (DC) SC in order to simplify

DC conversion for the receivers. The subcarriers -32 to -27 and 28 to 32 are set to zero resulting in a guard band that considerably contributes in reducing the out of band power. The pilot symbols are constructed by taking the values $1, 1, 1, -1$ and multiplying them with the second element of sequence p_n that controls the polarity of the pilot subcarriers. The sequence p_n is the cyclic extension of the 127 elements sequence and can be generated by the scrambler when the all ones initial state is used, and by replacing all 1s with -1s and all 0s with 1s and is given as

$$\begin{aligned} p_n = \{ &+1,+1,+1,+1,-1,-1,-1,+1,-1,-1,-1,-1,+1,+1,-1,+1,-1,-1,+1,+1, \\ &-1,+1,+1,-1,+1,+1,+1,+1,+1,+1,-1,+1,+1,+1,-1,+1,+1,-1,-1,+1, \\ &+1,+1,-1,+1,-1,-1,-1,+1,-1,+1,-1,-1,+1,-1,-1,+1,+1,+1,+1,+1, \\ &-1,-1,+1,+1,-1,-1,+1,-1,+1,-1,+1,+1,-1,-1,-1,+1,+1,-1,-1,-1, \\ &-1,+1,-1,-1,+1,-1,+1,+1,+1,+1,-1,+1,-1,+1,-1,+1,-1,-1,-1,-1, \\ &-1,+1,-1,+1,+1,-1,+1,-1,+1,+1,+1,-1,-1,+1,-1,-1,-1,+1,+1,+1, \\ &-1,-1,-1,-1,-1,-1,-1\} \end{aligned} \tag{A.5}$$

In the next step, the complex data stream needs to be transformed into a analog signal waveform $x(t)$ that is suitable for transmission over a channel. This is accomplished by an IFFT operation that converts the frequency domain complex symbols into a time domain signal. The IFFT operations takes $N = 64$ symbols as input at a time and applies sinusoidal basis functions. The output from IFFT is the summation of all 64 sinusoids that make up a single OFDM symbol. Thereafter, each OFDM symbol is prefixed with the last $N_{cp} = 16$ samples of itself. Called CP, it serves as a guard time and eliminate the ISI caused by the previous symbol. This also has the added advantage of converting the linear convolution of the frequency selective multipath channel to a circular one.

Preamble Generation & PPDU Frame Assembly As we have already seen in section A.1, a PLCP specific preamble that spans 12 OFDM out of which 10 are short training symbols (L-STF) and 2 are long training symbols (L-LTF). Seven out of the 10 L-STF are used for packet detection, automatic gain control setting and diversity selection. The remaining three are responsible for coarse frequency offset estimation and timing synchronization. L-STF is composed of 12 SCs and is generated using the sequence S_{stf} as shown in Equation A.6 and illustrated in Figure A.3. Note the leading (6) and trailing (5) zeros that are added to the sequence in order to have a sequence length equal to $N_{sc} = 64$

$$\begin{aligned} S_{stf} = P_w\{&0,0,0,0,0,0,0,0,1+j,0,0,0,-1-j,0,0,0,1+j,0,0,0,-1-j,0,0,0,-1-j, \\ &0,0,0,1+j,0,0,0,0,0,0,0,-1-j,0,0,0,-1-j,0,0,0,1+j,0,0,0,1+j,0, \\ &0,0,1+j,0,0,0,1+j,0,0,0,0,0,0,0\} \end{aligned} \tag{A.6}$$

where P_w is the normalization factor whose value typically is set to $\sqrt{\frac{13}{6}}$, which utilizes 12 out of 52 subcarriers. The resulting sequence is then converted into a time domain waveform using IFFT operation as represented in Equation A.7. However, no CP is prepended in this case. The resulting OFDM waveform is illustrated in Figure A.4

$$R_{short}(t) = \sum_{k=-N_{ST}/2}^{N_{ST}/2} S_{stf} e^{-j2\pi k \Delta_F t} \tag{A.7}$$

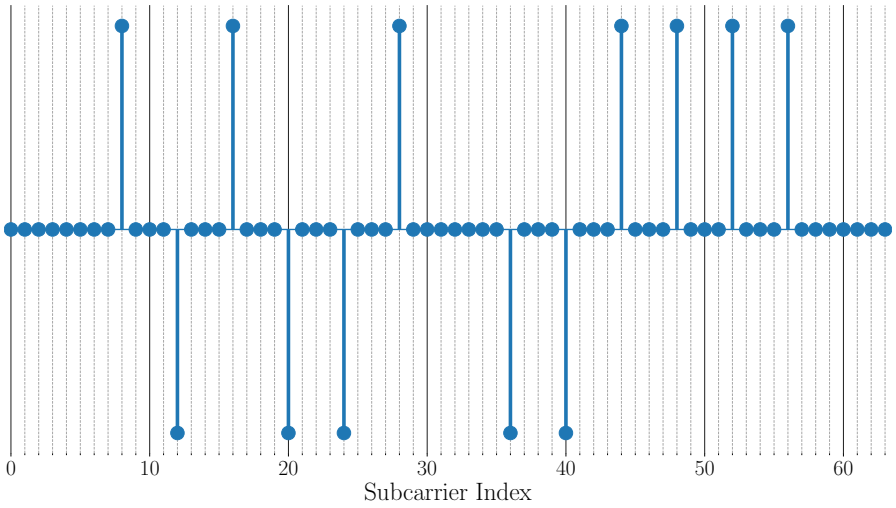


Figure A.3: L-STF in Frequency Domain

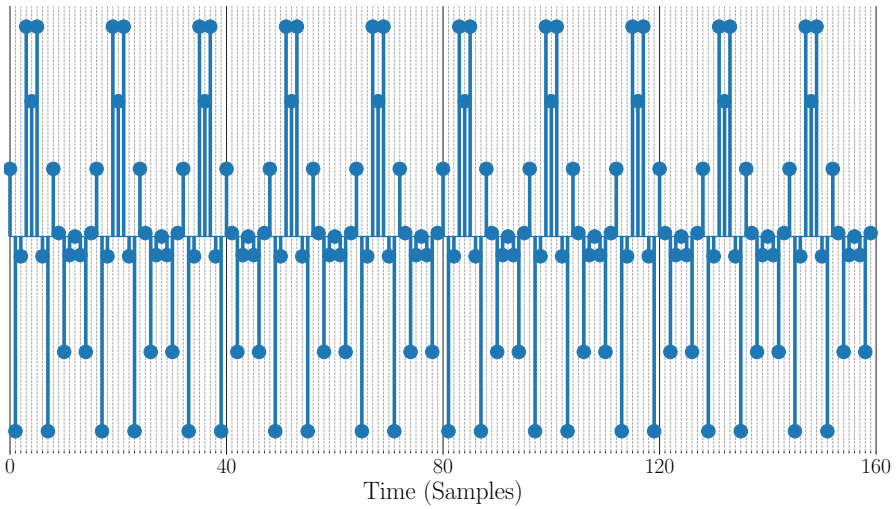


Figure A.4: L-STF in Time Domain

The L-LTF is used for fine frequency offset correction and channel estimation. It is composed of 2 long OFDM symbols and is constructed by repeating the sequence given in Equation A.8 two times. Leading and trailing zeros are also added to the sequence to make its length equal to $N_{sc} = 64$

$$\begin{aligned} S_{ltf} = \{&0,0,0,0,0,0,+1,+1,-1,-1,+1,+1,-1,+1,-1,+1,+1,+1,+1,+1,+1,\\ &-1,-1,+1,+1,-1,+1,-1,+1,+1,+1,+1,0,+1,-1,-1,+1,+1,-1,+1,\\ &-1,+1,-1,-1,-1,-1,-1,+1,+1,-1,-1,+1,-1,+1,-1,+1,+1,+1,+1,\\ &0,0,0,0,0\} \end{aligned} \tag{A.8}$$

The generated sequence is OFDM modulated by applying IFFT as shown in Equation A.9

$$R_{long}(t) = \sum_{k=-N_{ST}/2}^{N_{ST}/2} S_{ltf} e^{-j2\pi k \Delta_F (t-t_g)} \tag{A.9}$$

where t_g is the 3.2 µs guard interval that is used to avoid interference between L-STF and L-LTF. After OFDM modulation, a CP is prepended to the sequence only at the beginning and hence it is double the length of the standard data OFDM CP.

The L-LTF sequence in frequence and time domains is illustrated graphically in Figure A.5 and Figure A.6 respectively

Figure A.5: L-LTF in Frequency Domain

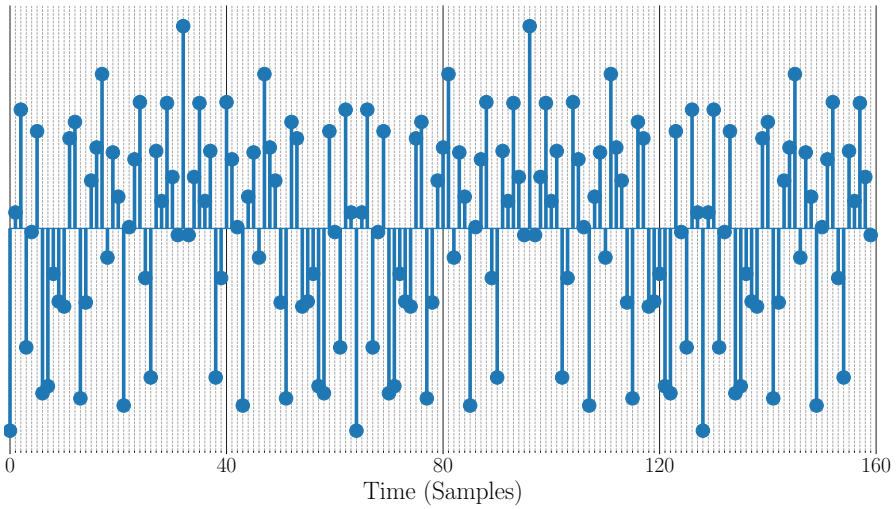


Figure A.6: L-LTF in Time Domain

The PLCP preamble is then followed by a Signal field, that consists of information about the transmission configuration used by the receiver to successfully decode the data packet. The first 4 bits contain the rate information that specifies the MCS used. Bits 6 to 17 contain information about the length of the PSDU data in octets. Bits 5 and 18 are reserved for future use while bits 19-24 constitute the tail bits that are used to terminate the convolutional encoder used later. The binary message bits are then convolutionally encoded using a base coding rate of 1/2, interleaved and mapped to symbols using BPSK which makes them very robust to transmission errors. Finally, they are OFDM modulated by means of IFFT, attached with CP to get the time domain OFDM symbols.

In the final step, the boundaries of each OFDM symbol are set via multiplication with a windowing function, which is defined as a rectangular pulse of duration T_{sym} where $T_{sym} = T_{FFT} + T_{CP}$.The windowing function can be represented in discrete time as

$$W[k] = \left\{ \begin{array}{ll} 1/2, & k = 1 \\ 1, & 1 \leq k \leq K \\ 1/2, & k = K + 1 \end{array} \right\} \tag{A.10}$$

where K is the number of samples in one OFDM symbol. Note that this effectively creates a small overlap between consecutive OFDM symbols, thereby smoothening transitions between them. Finally, the PLCP preamble, signal and data OFDM symbols are appended in the same order to create the final PPDU time domain frame that is ready for transmission.

A.1.1.2 Receiver Operations

The 802.11p receiver performs the inverse operations of the transmitter in order to decode the received waveform and retrieve the message. The processing steps are outlined in Figure A.7

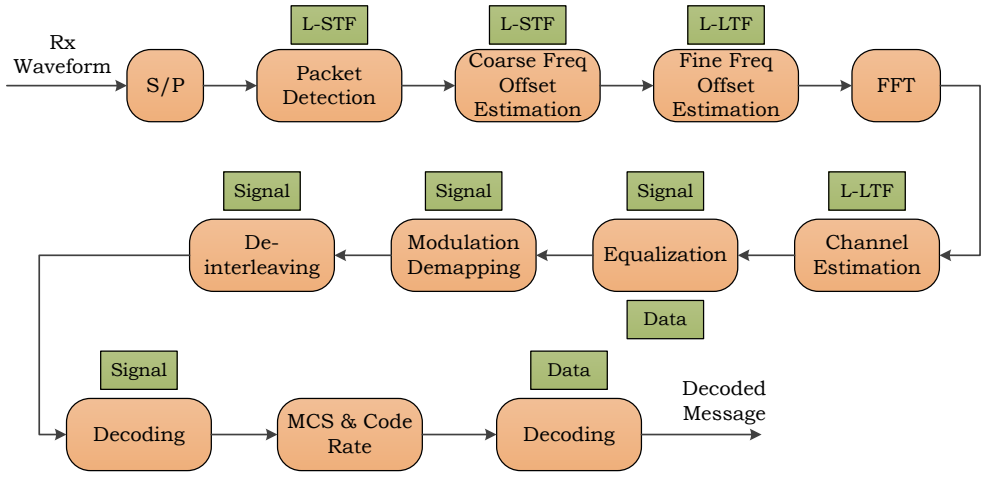


Figure A.7: Rx Operations for 802.11p

Packet Detection and CFO Estimation Being a Random Access Network, the receivers in an 802.11 network are not synchronized and hence does not know exactly when the packet starts. Hence, packet detection is performed in order to find an approximate estimate of the start of the preamble of an incoming packet. This is achieved by means of an delay and correlate algorithm proposed by Schmidl and Cox [160] that defines two consecutive sliding windows $P(n)$ and $R(n)$ to calculate the received energy and form a decision variable $M(n)$ as a ratio of total energy contained inside the two windows. The correlation windows at

time n are defined as

$$P(n) = \sum_{m=0}^{L-1} (r^*(n+m) \cdot r(n+m+L)) \quad \text{and} \tag{A.11}$$

$$R(n) = \sum_{m=0}^{L-1} |r(n+m+L)|^2 \tag{A.12}$$

where r_n are the received symbols and $L = 16$ is one L-STF interval. The decision variable $M(n)$ is given as

$$M(n) = \frac{|P(n)|^2}{(R(n))^2} \tag{A.13}$$

When the received signal consists of only noise, the output $P(n)$ of the delayed cross correlation is a zero mean random variable. Once the start of the packet is detected, the value of M_n jumps quickly to its maximum value which in turn gives a good estimate of the location of the packet edge.

Besides timing delays, the receiver carrier frequency is also shifted due to inherent non-idealities in the radio front ends of both the transmitter and the receiver. This causes linear phase shifts in the received signal given as $\epsilon = \frac{f_e}{\Delta f}$ with Δf being the SC frequency spacing. It is shown in [160] that the correlation presented in Equation A.11 can also be used to estimate the CFO as follows

$$\hat{\epsilon} = \frac{1}{2\pi} \frac{N}{L} \angle P(n) \tag{A.14}$$

The estimated $\hat{\epsilon}$ is further refined using L-LTF. Due to the presence of pilot symbols whose phase is known, this results in a more fine tuned frequency offset estimate $\hat{\epsilon}_f$. Finally, the received samples are multiplied by $\exp(-j2\pi\hat{\epsilon}_f n/N)$ to remove the frequency offset [115].

After correcting the timing and frequency offsets, the waveform is demodulated by means of an FFT that converts the time domain OFDM symbol into a frequency domain complex valued waveform and is represented as

$$S[k] = \sum_{n=0}^{K-1} s[n] e^{-j2\pi \frac{nk}{K}} \tag{A.15}$$

Channel Estimation & Equalization The 802.11p standard aims to characterize the channel by using the training symbols that are transmitted at the start of the OFDM waveform. In principle, it is possible to perform channel estimation on either of L-STF and L-LTF since both these sequences are known beforehand. However, the estimation process requires the presence of non-zero symbols since it essentially consists of a division operation (and division by zero valued complex symbols leads to some practical implementation issues). The L-STF contains only 12 non-zero SCs, thereby allowing us to only estimate 12 out of the available 52 SCs. Hence, for this reason channel estimation is performed on L-LTF that consists of 52 non-zero complex symbols.

A plethora of algorithms exists for estimating the channel in OFDM systems with varying levels of complexity [77, 220]. Some of them are outlined in the table. However, it can be said that LS estimator forms the basis for all the other algorithms for pilot assisted channel

estimation. The LS estimate of channel coefficient on k^{th} SC $\hat{H}_k(n)$, given the received sample $y_k(n)$ and the transmitted sample $x_k(n)$ is given as

$$\hat{H}_k(n) = \frac{y_k(n)}{x_k(n)} = H_k(n) + w_k(n) \tag{A.16}$$

where $H_k(n)$ denotes the true channel coefficients and $w_k(n)$ denotes the noise. Hence, it can be seen that LS estimator allows us to calculate channel coefficients with a simple division operation without relying on any knowledge about statistics of the channel. This also leads to its inherent problem of high mean-squared error.

The 802.11 standard specifies LS estimation as the base technique with some minor modifications. The first one is an averaged LS estimation technique where the repetitive structure of the L-LTF is exploited in order to calculate LS estimates for the first and second symbol individually ($\hat{H}_k(1)$ and $\hat{H}_k(2)$). The final channel estimate for the K^{th} SC used for subsequence symbols is the average of these two estimates

$$\hat{H}_k = \frac{1}{2}(\hat{H}_k(1) + \hat{H}_k(2)) \tag{A.17}$$

The second enhancement includes frequency smoothing that involves applying a moving average filter in order to derive a low-rank approximation of the LS estimates [190]. This is based on the fact that the channel impulse response is short compared with the OFDM symbol length and hence much of the energy of the channel impulse response (after frequency domain sampling) is contained in, or near the first L taps where L denotes the channel delay spread. Hence, to improve the estimation performance, only the taps with significant energy are considered and the low energy taps are approximated by zero.

The third enhancement includes tracking the highly dynamic V2V channel by means of pilot symbols that are located in SCs -21, -7, 7 and 21 respectively [57]. It is shown by a large body of research literature that the vehicular channel changes drastically over the course of a single packet transmission. Thereby, the channel estimate that is performed at the beginning of the packet (via preamble) is not adequate to combat these effects. Hence, the pilot indices in the data symbols are used as a feedback mechanism in order to compensate (to a limited extend) for these intra-packet channel effects. First, the LS estimates for each of the 4 pilot symbols are calculated and linearly interpolated across all th 64 SCs to obtain a channel estimate matrix H_{upd}. Then the overall channel estimate is updated so as to track the channel as follows

$$\hat{H}_k = \left(1 - \frac{1}{\alpha}\right)\hat{H}_{k-1} + \frac{1}{\alpha}\left(H_{upd}\right) \tag{A.18}$$

where $\hat{H}_{k-1}$ is the initial channel estimate and α is a memory parameter that controls the historical effect on current channel estimate.

After estimating H_k , the channel effects are mitigated by using an equalizer with coefficients that are inversely proportional to H_k, i.e., $E_k = \frac{1}{H_k}$. This simple strategy is known as Zero-Forcing (ZF) equalization and can be represented in discrete time domain as

$$\hat{y}_k(n) = E_k y_k(n) = \frac{1}{H_k}(H_k y_k(n) + w_k(n)) = \frac{H_k y_k(n)}{H_k} + \frac{w_k(n)}{H_k} = x_k(n) + \tilde{w}_k(n) \tag{A.19}$$

The disadvantage of ZF equalization is that for small values of $H_k(n)$, the magnitude of the corresponding ZF filter coefficient becomes very large this effectively amplifies the noise

by a large factor. Furthermore, the channel may have zeros in its frequency response that cannot be inverted. In order to address these problems, a linear MMSE estimator can be used that keeps track of the noise and signal variance over time in order to reduce noise amplification at low SNRs. However, the main drawback of MMSE estimator is its increased complexity. The 802.11 standard allows for both the equalization techniques.

Modulation Demapping & Decoding The equalized symbols are then mapped into binary bits by means of a soft demapping which outputs Log-Likelihood Ratios (LLRs). The posteriori LLR of c_i given an observation r is defined as

$$\Lambda(c_i|r) = \log \frac{P(c_i = 1|r)}{P(c_i = 0|r)} \tag{A.20}$$

The sign of $\Lambda(c_i|r)$ contains information about the value of the bit (i.e., 0 or 1). The magnitude of the LLR expresses the reliability of the associated decision. Higher values indicates higher certainty and vice versa.

The soft LLR bits are first de-interleaved via two permutations that are basically the inverse of the permutations defined in A.1.1.1. Let j denote the index of the original received bit before first permutation and i the index after the first and before the second permutation. The first permutation is then defined by the rule

$$i = s \cdot \lfloor \frac{j}{s} \rfloor + j + (\lfloor \frac{16 \cdot j}{N_{CBPS}} \rfloor) \mod (s)\ j = 0, 1, ..., N_{CBPS} - 1 \tag{A.21}$$

and the second permutation as

$$k = 16 \cdot i - (N_{CBPS} - 1) \lfloor \frac{16 \cdot i}{N_{CBPS}} \rfloor,\ i = 0, 1, ..., N_{CBPS} - 1 \tag{A.22}$$

where k denotes the final bit index that is fed as n an input the decoder. The 802.11 standard recommends the Viterbi algorithm for decoding whose goal is to find the sequence $\hat{s}$ that was transmitted with the highest likelihood, if the received sequence is r given as

$$\hat{s} = \max_a \Pr(\mathrm{r}|\mathrm{s}) \tag{A.23}$$

where the maximization is done over all possible transmit sequences s and is based on the trellis diagram representation. More details on the Viterbi alogirithm can be found in various works. Finally, the decoded data bits are descrambled using the same scrambler as defined in A.1.1.1 to get the user data.

Bibliography

[1] Guillermo Acosta-marum, Guillermo Acosta-marum, and Mary Ann Ingram. Doubly selective vehicle-to-vehicle channel measurements and modeling at 5.9 GHz. *PROC. INT. SYMP. WIRELESS PERSONAL MULTIMEDIA COMMUN*, 2006.

[2] Imad Alawe, Adlen Ksentini, Yassine Hadjadj-Aoul, and Philippe Bertin. Improving Traffic Forecasting for 5G Core Network Scalability: A Machine Learning Approach. *IEEE Network*, 2018.

[3] Paul Alexander, David Haley, and Alex Grant. Cooperative intelligent transport systems: 5.9-GHz field trials. *Proceedings of the IEEE*, 99(7):1213–1235, jul 2011.

[4] Mohamed I. Alhajri, Nazar T. Ali, and Raed M. Shubair. Classification of Indoor Environments for IoT Applications: A Machine Learning Approach. *IEEE Antennas and Wireless Propagation Letters*, 2018.

[5] Samad Ali, Walid Saad, Nandana Rajatheva, Kapseok Chang, Daniel Steinbach, Benjamin Sliwa, Christian Wietfeld, Kai Mei, Hamid Shiri, Hans-Jürgen Zepernick, Thi My Chinh Chu, Ijaz Ahmad, Jyrki Huusko, Jaakko Suutala, Shubhangi Bhadauria, Vimal Bhatia, Rangeet Mitra, Saidhiraj Amuru, Robert Abbas, Baohua Shao, Michele Capobianco, Guanghui Yu, Maelick Claes, Teemu Karvonen, Mingzhe Chen, Maksym Girnyk, and Hassan Malik. 6G White Paper on Machine Learning in Wireless Communication Networks. apr 2020.

[6] Zoraze Ali, Nicola Baldo, Josep Mangues-Bafalluy, and Lorenza Giupponi. Machine learning based handover management for improved QoE in LTE. In *Proceedings of the NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, 2016.

[7] Ahmed Alkhateeb, Sam Alex, Paul Varkey, Ying Li, Qi Qu, Djordje Tujkovic Facebook, and Djordje Tujkovic. Deep Learning Coordinated Beamforming for Highly-Mobile Millimeter Wave Systems. Technical report, 2019.

[8] Ghassan Alnwaimi, Seiamak Vahid, and Klaus Moessner. Dynamic heterogeneous learning games for opportunistic access in LTE-based macro/femtocell deployments. *IEEE Transactions on Wireless Communications*, 14(4):2294–2308, apr 2015.

[9] S. H. Alsamhi, Ou Ma, and Mohd Samar Ansari. Survey on artificial intelligence based techniques for emerging robotic communication, nov 2019.

[10] Kenneth Skilling Andrews, Chris Heegard, and Dexter Kozen. A Theory of Interleavers, 1997.

[11] Miguel Angel Gutierrez-Estevez, Renato L.G. Cavalcante, and Slawomir Stanczak. Nonparametric Radio Maps Reconstruction Via Elastic Net Regularization with Multi-Kernels. In *IEEE Workshop on Signal Processing Advances in Wireless Communications, SPAWC*, 2018.

[12] Faycal Ait Aoudia and Jakob Hoydis. End-to-End Learning of Communications Systems Without a Channel Model. In *Conference Record - Asilomar Conference on Signals, Systems and Computers*, 2019.

[13] Raul Aquino Santos, Arthur Edwards, and Victor Rangel Licea. *Wireless technologies in vehicular ad hoc networks: Present and future challenges*. IGI Global, 2012.

[14] Alia Asheralieva and Yoshikazu Miyanaga. An Autonomous Learning-Based Algorithm for Joint Channel and Power Level Selection by D2D Pairs in Heterogeneous Cellular Networks. *IEEE Transactions on Communications*, 2016.

[15] Abdelhadi Azzouni and Guy Pujolle. A Long Short-Term Memory Recurrent Neural Network Framework for Network Traffic Matrix Prediction. may 2017.

[16] Alessandro Bazzi, Giammarco Cecchini, Alberto Zanella, and Barbara M. Masini. Study of the Impact of PHY and MAC Parameters in 3GPP C-V2V Mode 4. *IEEE Access*, 6:71685–71698, 2018.

[17] Lucia Lo Bello. Novel trends in automotive networks: A perspective on Ethernet and the IEEE Audio Video Bridging. In *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pages 1–8. IEEE, sep 2014.

[18] P Bello. Characterization of randomly time-variant linear channels. *IEEE transactions on Communications Systems*, 11(4):360–393, 1963.

[19] Mehdi Bennis, Mérouane Debbah, and H. Vincent Poor. Ultra-Reliable and Low-Latency Wireless Communication: Tail, Risk and Scale. jan 2018.

[20] Mehdi Bennis, Samir M. Perlaza, Pol Blasco, Zhu Han, and H. Vincent Poor. Self-organization in small cell networks: A reinforcement learning approach. *IEEE Transactions on Wireless Communications*, 2013.

[21] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1. In *Proceedings of ICC '93 - IEEE International Conference on Communications*, volume 2, pages 1064–1070. IEEE.

[22] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1. In *Proceedings of ICC '93 - IEEE International Conference on Communications*, volume 2, pages 1064–1070. IEEE, 1993.

[23] Ken Bisson. SAE AS6802 Deterministic Ethernet Network Solution, 2011.

[24] R.C. Bose and D.K. Ray-Chaudhuri. On a class of error correcting binary group codes. *Information and Control*, 3(1):68–79, mar 1960.

[25] R.C. Bose and D.K. Ray-Chaudhuri. On a class of error correcting binary group codes. *Information and Control*, 3(1):68–79, mar 1960.

[26] Robert Boys. Safe-by-Wire: The Leading Edge in Vehicle Airbag Control. Technical report, mar 2004.

[27] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols, 1998.

[28] J.K. Cavers. An analysis of pilot symbol assisted modulation for Rayleigh fading channels (mobile radio). *IEEE Transactions on Vehicular Technology*, 40(4):686–693, 1991.

[29] Giammarco Cecchini, Alessandro Bazzi, Barbara M Masini, and Alberto Zanella. Ltev2vsim: An lte-v2v simulator for the investigation of resource allocation for cooperative awareness. In *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pages 80–85. IEEE, 2017.

[30] Giammarco Cecchini, Alessandro Bazzi, Barbara M. Masini, and Alberto Zanella. Performance comparison between IEEE 802.11p and LTE-V2V in-coverage and out-of-coverage for cooperative awareness. In *IEEE Vehicular Networking Conference, VNC*, volume 2018-Janua, pages 109–114. IEEE Computer Society, jan 2018.

[31] Liang Chen, Fenfang Xie, Zibin Zheng, and Yaoming Wu. Predicting Quality of Service via Leveraging Location Information. *Complexity*, 2019.

[32] Mingzhe Chen, Ursula Challita, Walid Saad, Changchuan Yin, and Mérouane Debbah. Machine Learning for Wireless Networks with Artificial Intelligence: A Tutorial on Neural Networks. oct 2017.

[33] Xianfu Chen, Celimuge Wu, Tao Chen, Honggang Zhang, Zhi Liu, Yan Zhang, and Mehdi Bennis. Age of Information Aware Radio Resource Management in Vehicular Networks: A Proactive Deep Reinforcement Learning Perspective. *IEEE Transactions on Wireless Communications*, 2020.

[34] Yu Chen, Kai Zhao, Jing ya Zhao, Qing hua Zhu, and Yong Liu. Deep Learning Based Antenna Muting and Beamforming Optimization in Distributed Massive MIMO Systems. In *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, volume 278, pages 18–30. Springer Verlag, feb 2019.

[35] Lin Cheng, Benjamin Henty, Reginald Cooper, Daniel D. Stancil, and Fan Bai. Multi-path propagation measurements for vehicular networks at 5.9 GHz. In *IEEE Wireless Communications and Networking Conference, WCNC*, pages 1239–1244, 2008.

[36] Chengyang Li and S. Roy. Subspace based blind channel estimation for OFDM by exploiting virtual carrier. In *GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No.01CH37270)*, pages 295–299. IEEE.

[37] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. jun 2014.

[38] Sangjin Cho, Seungyeob Chae, Minjoong Rim, and Chung G. Kang. System level simulation for 5G cellular communication systems. In *International Conference on Ubiquitous and Future Networks, ICUFN*, pages 296–299. IEEE Computer Society, jul 2017.

[39] C. Ciocan. The Domestic Digital Bus system (D2B)-a maximum of control convenience in audio video. *IEEE Transactions on Consumer Electronics*, 36(3):619–622, 1990.

[40] R. H. Clarke. A Statistical Theory of Mobile Radio Reception. *Bell System Technical Journal*, 47(6):957–1000, 1968.

[41] Jean-Marc Conrat and Patrice Pajusco. A versatile propagation channel simulator for mimo link level simulation. *EURASIP Journal on Wireless Communications and Networking*, 2007:1–13, 2007.

[42] Ronald W. Cox. Local Area Network Technology Applied to Automotive Electronic Communications. *IEEE Transactions on Industrial Electronics*, IE-32(4):327–333, nov 1985.

[43] Dan Deng, Xingwang Li, Ming Zhao, Khaled M. Rabie, and Rupak Kharel. Deep learning-based secure mimo communications with imperfect CSI for heterogeneous networks. *Sensors (Switzerland)*, 20(6), mar 2020.

[44] Amit. Dhir. *The digital consumer technology handbook : a comprehensive guide to devices, standards, future directions, and programmable logic solutions*. Newnes, 2004.

[45] M Dinkel and D Fengler. Unified communication in heterogeneous automotive control systems. *Transport*, 2006.

[46] Anca Discant, Alexandrina Rogozan, Corneliu Rusu, and Abdelaziz Bensrhair. Sensors for obstacle detection - A survey. In *ISSE 2007 - 30th International Spring Seminar on Electronics Technology 2007; Conference Proceedings: Emerging Technologies for Electronics Packaging*, pages 100–105. Institute of Electrical and Electronics Engineers Inc., 2007.

[47] V. H.Mac Donald. Advanced Mobile Phone Service: The Cellular Concept. *Bell System Technical Journal*, 58(1):15–41, jan 1979.

[48] Rui Dong, Changyang She, Wibowo Hardjawana, Yonghui Li, and Branka Vucetic. Deep Learning for Hybrid 5G Services in Mobile Edge Computing Systems: Learn from a Digital Twin. *IEEE Transactions on Wireless Communications*, 2019.

[49] Sebastian Dorner, Sebastian Cammerer, Jakob Hoydis, and Stephan Ten Brink. Deep Learning Based Communication over the Air. *IEEE Journal on Selected Topics in Signal Processing*, 2018.

[50] G.D. Durgin. Space-Time Wireless Channels. *Space-Time Wireless Channels*, pages 1–19, 2002.

[51] ETSI. European Telecommunications Standards Institute Universal Mobile Telecommunications System (UMTS); Selection procedures for the choice of radio transmission technologies of the UMTS (UMTS 30.03 version 3.1.0) Universal Mobile Telecommunications System. Technical report, 1997.

[52] ETSI. ETSI EN 302 637-2 v1.3.1. Vehicular Communications ; Basic Set of Applications ; Part 2 : Specification of Cooperative. *History*, 2010.

[53] ETSI. ETSI EN 302 637-3 Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service. *Etsi*, 2010.

[54] Zubair Md Fadlullah, Fengxiao Tang, Bomin Mao, Nei Kato, Osamu Akashi, Takeru Inoue, and Kimihiro Mizutani. State-of-the-Art Deep Learning: Evolving Machine Intelligence Toward Tomorrow's Intelligent Network Traffic Control Systems. *IEEE Communications Surveys and Tutorials*, 19(4):2432–2455, oct 2017.

[55] Chaoqiong Fan, Bin Li, Chenglin Zhao, Weisi Guo, and Ying Chang Liang. Learning-Based Spectrum Sharing and Spatial Reuse in mm-Wave Ultradense Networks. *IEEE Transactions on Vehicular Technology*, 2018.

[56] Saleh Faruque. Introduction to Channel Coding. pages 1–16. Springer, Cham, 2016.

[57] Joseph A. Fernandez, Daniel D. Stancil, and Fan Bai. Dynamic channel equalization for IEEE 802.11p waveforms in the vehicle-to-vehicle channel. In *2010 48th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2010*, 2010.

[58] Andreas Festag. Cooperative intelligent transport systems standards in Europe. *IEEE Communications Magazine*, 2014.

[59] Andreas Festag. Standards for vehicular communication-from IEEE 802.11p to 5G. *Elektrotechnik und Informationstechnik*, 2015.

[60] Nikos Floudas, Aris Polychronopoulos, Olivier Aycard, Julien Burlet, and Malte Ahrholdt. High level sensor data fusion approaches for object recognition in road environment. In *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 136–141. Institute of Electrical and Electronics Engineers Inc., 2007.

[61] G. Forney. Convolutional codes I: Algebraic structure. *IEEE Transactions on Information Theory*, 16(6):720–738, nov 1970.

[62] Rick Fritschek, Rafael F. Schaefer, and Gerhard Wunder. Deep Learning for Channel Coding via Neural Mutual Information Estimation. In *IEEE Workshop on Signal Processing Advances in Wireless Communications, SPAWC*, 2019.

[63] Ana Galindo-Serrano and Lorenza Giupponi. Distributed Q-learning for aggregated interference control in cognitive radio networks. *IEEE Transactions on Vehicular Technology*, 2010.

[64] Ana Galindo-Serrano, Lorenza Giupponi, and Gunther Auer. Distributed learning in multiuser OFDMA femtocell networks. In *IEEE Vehicular Technology Conference*, 2011.

[65] R. Gallager. Low-density parity-check codes. *IEEE Transactions on Information Theory*, 8(1):21–28, jan 1962.

[66] Feifei Gao and Arumugam Nallanathan. Blind Channel Estimation for OFDM Systems via a Generalized Precoding. *IEEE Transactions on Vehicular Technology*, 56(3):1155–1164, may 2007.

[67] S. Gazor. Prediction in LMS-type adaptive algorithms for smoothly time varying environments. *IEEE Transactions on Signal Processing*, 47(6):1735–1739, jun 1999.

[68] Amir Ghasemi. Predictive Modeling of LTE User Throughput Via Crowd-Sourced Mobile Spectrum Data. In *2018 IEEE International Symposium on Dynamic Spectrum Access Networks, DySPAN 2018*, 2019.

[69] M.J.E. Golay. Notes on Digital Coding. *Proceedings of the IRE*, 37(6):657–657, jun 1949.

[70] Manuel Gonzalez-Martin, Miguel Sepulcre, Rafael Molina-Masegosa, and Javier Gozalvez. Analytical Models of the Performance of C-V2X Mode 4 Vehicular Communications. jul 2018.

[71] Nikhil Gulati, Rachel Greenstadt, Kapil R. Dandekar, and John M. Walsh. GMM based semi-supervised learning for channel-based authentication scheme. In *IEEE Vehicular Technology Conference*, 2013.

[72] Ismail. Guvenc. *Reliable communications for short-range wireless systems*. Cambridge University Press, 2011.

[73] R. W. Hamming. Error Detecting and Error Correcting Codes. *Bell System Technical Journal*, 29(2):147–160, apr 1950.

[74] K. V.S. Hari. Channel models for wireless communication systems. *International Series in Operations Research and Management Science*, 158:47–64, 2011.

[75] Simon S. Haykin and Simon. *Adaptive filter theory*. Prentice Hall, 1996.

[76] R.W. Heath and G.B. Giannakis. Exploiting input cyclostationarity for blind channel identification in OFDM systems. *IEEE Transactions on Signal Processing*, 47(3):848–856, mar 1999.

[77] Mattias Hermansson and Viktor Skoda. Evaluating channel estimation methods for 802.11p systems, 2011.

[78] Jorge Luis Hernandez Villapol. *Spectrum Analysis and Prediction Using Long Short-Term Memory Neural Networks (LSTMs) and Cognitive Radios*. PhD thesis, 2017.

[79] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, nov 1997.

[80] Alexis Hocquenghem. Codes correcteurs d'erreurs. *Chiffres*, 2(2):147–56, 1959.

[81] Liang Huang, Suzhi Bi, and Ying-Jun Angela Zhang. Deep Reinforcement Learning for Online Computation Offloading in Wireless Powered Mobile-Edge Computing Networks. *IEEE Transactions on Mobile Computing*, pages 1–1, aug 2018.

[82] Liang Huang, Xu Feng, Cheng Zhang, Liping Qian, and Yuan Wu. Deep reinforcement learning-based joint task offloading and bandwidth allocation for multi-user mobile edge computing. *Digital Communications and Networks*, 2019.

[83] Lingchen Huang, Huazi Zhang, Rong Li, Yiqun Ge, and Jun Wang. AI Coding: Learning to Construct Error Correction Codes. *IEEE Transactions on Communications*, 2020.

[84] Yu Di Huang, Paul Pu Liang, Qianqian Zhang, and Ying Chang Liang. A machine learning approach to MIMO communications. In *IEEE International Conference on Communications*, volume 2018-May. Institute of Electrical and Electronics Engineers Inc., jul 2018.

[85] R Hueber. DRIVE: Dedicated Road Infrastructure for Vehicle Safety in Europe. *Publication of: Prevention Routiere Internationale*, 1990.

[86] IEEE Computer Society. LAN/MAN Standards Committee., Institute of Electrical and Electronics Engineers., and IEEE-SA Standards Board. *IEEE standard for information technology : telecommunications and information exchange between systems : local and metropolitan area networks–specific requirements. Part 11, Wireless LAN medium access control (MAC) and physical layer (PHY) specifications.* Institute of Electrical and Electronics Engineers, 2012.

[87] Toshiaki Ishii, Nobutake Tsuyuno, Toshiya Sato, and Mitsuhiro Masuda. Corrosion studies of copper and aluminum interconnects exposed to automotive oils. *IEEE Transactions on Components and Packaging Technologies*, 2006.

[88] Jithin Jagannath, Nicholas Polosky, Anu Jagannath, Francesco Restuccia, and Tommaso Melodia. Machine Learning for Wireless Communications in the Internet of Things: A Comprehensive Survey. *Ad Hoc Networks*, 93, jan 2019.

[89] Marcel Jar and Gerhard Fettweis. Throughput maximization for LTE uplink via resource allocation. In *2012 International Symposium on Wireless Communication Systems (ISWCS)*, pages 146–150. IEEE, aug 2012.

[90] Yongseok Jeon, Seungho Kuk, and Hyogon Kim. Reducing Message Collisions in Sensing-Based Semi-Persistent Scheduling (SPS) by Using Reselection Lookaheads in Cellular V2X. *Sensors*, 18(12):4388, dec 2018.

[91] Jorjeta G. Jetcheva, Yih Chun Hu, Santashil PalChaudhuri, Amit Kumar Saha, and David B. Johnson. Design and evaluation of a metropolitan area multitier wireless ad hoc network architecture. In *Proceedings - 5th IEEE Workshop on Mobile Computing Systems and Applications, WMCSA 2003*, 2003.

[92] Chunxiao Jiang, Haijun Zhang, Yong Ren, Zhu Han, Kwang-Cheng Chen, and Lajos Hanzo. Machine Learning Paradigms for Next-Generation Wireless Networks. *IEEE Wireless Communications*, 24(2):98–105, apr 2017.

[93] Yihan Jiang, Hyeji Kim, Himanshu Asnani, Sreeram Kannan, Sewoong Oh, and Pramod Viswanath. Turbo Autoencoder: Deep learning based channel codes for point-to-point communication channels. nov 2019.

[94] Y Jide, H Q Ngo, Jide Yuan, Hien Quoc Ngo, and Michail Matthaiou. Machine Learning-Based Channel Estimation in Massive MIMO with Channel Aging. 2019.

[95] Stefan Joerer, Falko Dressler, and Christoph Sommer. Comparing apples and oranges?: Trends in IVC simulations. In *VANET'12 - Proceedings of the 9th ACM International Workshop on VehiculAr Inter-NETworking, Systems, and Applications*, 2012.

[96] David B Johnson and David A Maltz. DSR : The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. *Computer Science Department, Carnegie Mellon University, Addison-Wesley*, 1996.

[97] Florian Jomrich, Alexander Herzberger, Tobias Meuser, Björn Richerzhagen, Ralf Steinmetz, and Cornelius Wille. Cellular bandwidth prediction for highly automated driving evaluation of machine learning approaches based on real-world data. In *VEHITS 2018 - Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems*, 2018.

[98] Johan Karedal, Fredrik Tufvesson, Nicolai Czink, Alexander Paier, Charlotte Dumard, Thomas Zemen, Christoph F. Mecklenbräuker, and Andreas F. Molisch. A geometry-based stochastic MIMO model for vehicle-to-vehicle communications. *IEEE Transactions on Wireless Communications*, 8(7):3646–3657, jul 2009.

[99] Martin Kasparick, Renato L.G. Cavalcante, Stefan Valentin, Sławomir Stańczak, and Masahiro Yukawa. Kernel-Based Adaptive Online Reconstruction of Coverage Maps With Side Information. *IEEE Transactions on Vehicular Technology*, 2016.

[100] SM Kay. *Fundamentals of statistical signal processing*. Prentice Hall PTR, 1993.

[101] Hyeji Kim, Yihan Jiang, Ranvir Rana, Sreeram Kannan, Sewoong Oh, and Pramod Viswanath. Communication Algorithms via Deep Learning. may 2018.

[102] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. dec 2014.

[103] Can Emre Koksal and Hari Balakrishnan. Quality-aware routing metrics for time-varying wireless mesh networks. *IEEE Journal on Selected Areas in Communications*, 24(11):1984–1994, 2006.

[104] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent Development and Applications of {SUMO - Simulation of Urban MObility}. *International Journal On Advances in Systems and Measurements*, 2012.

[105] Adita Kulkarni, Anand Seetharam, Arti Ramesh, and J. Dinal Herath. DeepChannel: Wireless channel quality prediction using deep learning. *IEEE Transactions on Vehicular Technology*, 2020.

[106] Wei Kuang Lai, Mei Tso Lin, and Yu Hsuan Yang. A Machine learning system for routing decision-making in urban vehicular ad hoc networks. *International Journal of Distributed Sensor Networks*, 2015.

[107] Ron Levie, Cagkan Yapar, Gitta Kutyniok, and Giuseppe Caire. Pathloss Prediction using Deep Learning with Applications to Cellular Optimization and Efficient D2D Link Scheduling. 2020.

[108] Jing Li, Xing Zhang, Jiaxin Zhang, Jie Wu, Qi Sun, and Yuxuan Xie. Deep Reinforcement Learning-Based Mobility-Aware Robust Proactive Resource Allocation in Heterogeneous Networks. *IEEE Transactions on Cognitive Communications and Networking*, 2020.

[109] Le Liang, Hao Ye, Guanding Yu, and Geoffrey Ye Li. Deep-Learning-Based Wireless Resource Allocation with Application to Vehicular Networks. *Proceedings of the IEEE*, 2020.

[110] Yong Liao, Yuanxiao Hua, Xuewu Dai, Haimei Yao, and Xinyi Yang. ChanEstNet: A Deep Learning Based Channel Estimation for High-Speed Scenarios. In *IEEE International Conference on Communications*, 2019.

[111] L. Lindbom, M. Sternad, and A. Ahlen. Tracking of time-varying mobile radio channels .1. The Wiener LMS algorithm. *IEEE Transactions on Communications*, 49(12):2207–2217, 2001.

[112] Nicholas E. Lownes and Randy B. Machemehl. VISSIM: A multi-parameter sensitivity analysis. In *Proceedings - Winter Simulation Conference*, 2006.

[113] R. W. Lucky. Automatic Equalization for Digital Communication. *Bell System Technical Journal*, 44(4):547–588, 1965.

[114] A. Lezama Lugo and J.-P. Hubaux. TraNS: realistic joint traffic and network simulator for VANETs. *ACM SIGMOBILE Mobile Computing and Communications Review*, 2008.

[115] Georg Maier, Alexander Paier, and Christoph F. Mecklenbrauker. Packet detection and frequency synchronization with antenna diversity for IEEE 802.11p based on real-world measurements. In *2011 International ITG Workshop on Smart Antennas*, pages 1–7. IEEE, feb 2011.

[116] Taras Maksymyuk, Juraj Gazda, Oleh Yaremko, and Denys Nevinskiy. Deep learning based massive MIMO beamforming for 5G mobile network. In *Proceedings of the 2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS-SWS 2018*, 2018.

[117] Qian Mao, Fei Hu, and Qi Hao. Deep learning for intelligent wireless networks: A comprehensive survey, oct 2018.

[118] DW Matolak, Q Wu of the 5th European Conference on ..., and Undefined 2011. Channel models for V2V communications: A comparison of different approaches. *ieeexplore.ieee.org*.

[119] Christoph F. Mecklenbraüker, Andreas F. Molisch, Johan Karedal, Fredrik Tufvesson, Alexander Paier, Laura Bernadó, Thomas Zemen, Oliver Klemp, and Nicolai Czink. Vehicular channel characterization and its implications for wireless system design and performance. *Proceedings of the IEEE*, 99(7):1189–1212, 2011.

[120] Christian Mehlführer, Josep Colom Ikuno, Michal Šimko, Stefan Schwarz, Martin Wrulich, and Markus Rupp. The Vienna LTE simulators - Enabling reproducibility in wireless communications research. *EURASIP Journal on Advances in Signal Processing*, 2011(1):29, dec 2011.

[121] Kai Mei, Jun Liu, Xiaochen Zhang, and Jibo Wei. Machine Learning Based Channel Estimation: A Computational Approach for Universal Channel Conditions. nov 2019.

[122] Andreas F. Molisch, Fredrik Tufvesson, Johan Karedal, and Christoph Mecklenbrauker. Propagation aspects of vehicle-to-vehicle communications - An overview. In *RWS 2009 IEEE Radio and Wireless Symposium, Proceedings*, pages 179–182, 2009.

[123] Andreas F. Molisch, Fredrik Tufvesson, Johan Karedal, and Christoph F. Mecklenbräuker. A survey on vehicle-to-vehicle propagation channels. *IEEE Wireless Communications*, 16(6):12–22, dec 2009.

[124] E. Moulines, P. Duhamel, J.-F. Cardoso, and S. Mayrargue. Subspace methods for the blind identification of multichannel FIR filters. *IEEE Transactions on Signal Processing*, 43(2):516–525, 1995.

[125] B. Muquet and M. de Courville. Blind and semi-blind channel identification methods using second order statistics for OFDM systems. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, pages 2745–2748 vol.5. IEEE, 1999.

[126] Stephen S. Mwanje, Lars Christoph Schmelz, and Andreas Mitschele-Thiel. Cognitive Cellular Networks: A Q-Learning Framework for Self-Organizing Networks. *IEEE Transactions on Network and Service Management*, 2016.

[127] Woongsoo Na, Byungjun Bae, Sukhee Cho, and Nayeon Kim. Deep-learning based adaptive beam management technique for mobile high-speed 5G mmWave Networks. In *IEEE International Conference on Consumer Electronics - Berlin, ICCE-Berlin*, 2019.

[128] Qassim Nasir. Wireless Channel Tracking Based on Optimum Predictive LMS. *Wireless Personal Communications*, 48(4):511–519, mar 2009.

[129] Qassim Nasir and S. Faisal Ali Shah. Predictive LMS for Mobile Channel Tracking. *Journal of Applied Sciences*, 5(2):337–340, feb 2005.

[130] Shuai Nie and Ian F. Akyildiz. Deep kernel learning-based channel estimation in ultra-massive MIMO communications at 0.06-10 THz. In *2019 IEEE Globecom Workshops, GC Wkshps 2019 - Proceedings*, 2019.

[131] T. Nolte, H. Hansson, and L. lo Bello. Automotive Communications - Past, Current and Future. *2005 IEEE Conference on Emerging Technologies and Factory Automation*, 1:985–992, 2005.

[132] Timothy O'Shea and Jakob Hoydis. An Introduction to Deep Learning for the Physical Layer. In *IEEE Transactions on Cognitive Communications and Networking*, 2017.

[133] Alexander Paier, Johan Karedal, Nicolai Czink, Helmut Hofstetter, Charlotte Dumard, Thomas Zemen, Fredrik Tufvesson, Christoph F. Mecklenbräuker, and Andreas F. Molisch. First results from car-to-car and car-to-infrastructure radio channel measurements at 5.2 GHZ. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, 2007.

[134] Alexander Paier, Johan Karedal, Nicolai Czink, Helmut Hofstetter, Charlotte Dumard, Thomas Zemen, Fredrik Tufvesson, Andreas F. Molisch, and Christoph F. Mecklenbräuker. Car-to-car radio channel measurements at 5 GHz: Pathloss, power-delay profile, and delay-Doppler spectrum. In *Proceedings of 4th IEEE Internatilonal Symposium on Wireless Communication Systems 2007, ISWCS*, pages 224–228, 2007.

[135] Alexander Paier, Thomas Zemen, Laura Bernadó, Gerald Matz, Johan Karedal, Nicolai Czink, Charlotte Dumard, Fredrik Tufvesson, Andreas F. Molisch, and Christoph F. Mecklenbräuker. Non-WSSUS vehicular channel characterization in highway and urban scenarios at 5.2 GHZ using the local scattering function. In *2008 International ITG Workshop on Smart Antennas, WSA 2008*, pages 9–15, 2008.

[136] Natesan Palanivel and Tingsu Chen. Wiring harness reduction in automotive using Li-Fi technology. In *Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2017*, 2018.

[137] Krzysztof Pawlikowski, Hae Duck Joshua Jeong, and Jong Suk Ruth Lee. On credibility of simulation studies of telecommunication networks. *IEEE Communications Magazine*, 2002.

[138] Chengcheng Pei, Ning Zhang, Xuemin Sherman Shen, and Jon W. Mark. Channel-based physical layer authentication. In *2014 IEEE Global Communications Conference, GLOBECOM 2014*, 2014.

[139] Charles E. Perkins, Elizabeth M. Royer, Samir R. Das, and Mahesh K. Marina. Performance comparison of two on-demand routing protocols for ad hoc networks. *IEEE Personal Communications*, 8(1):16–28, 2001.

[140] A. Petroff. A practical, high performance Ultra-Wideband radar platform. In *2012 IEEE Radar Conference*, pages 0880–0884. IEEE, may 2012.

[141] A. Petropulu and Ruifeng Zhang. Blind channel estimation for OFDM systems. In *Proceedings of 2002 IEEE 10th Digital Signal Processing Workshop, 2002 and the 2nd Signal Processing Education Workshop.*, pages 366–370. IEEE.

[142] A. Petropulu, R. Zhang, and R. Lin. Blind OFDM Channel Estimation Through Simple Linear Precoding. *IEEE Transactions on Wireless Communications*, 3(2):647–655, mar 2004.

[143] Giuseppe Piro, Luigi Alfredo Grieco, Gennaro Boggia, Francesco Capozzi, and Pietro Camarda. Simulating LTE cellular systems: An open-source framework. *IEEE Transactions on Vehicular Technology*, 60(2):498–513, feb 2011.

[144] Michele Polese, Rittwik Jana, Velin Kounev, Ke Zhang, Supratim Deb, and Michele Zorzi. Machine Learning at the Edge: A Data-Driven Architecture with Applications to 5G Cellular Networks. *IEEE Transactions on Mobile Computing*, 2020.

[145] Darijo Raca, Ahmed H. Zahran, Cormac J. Sreenan, Rakesh K. Sinha, Emir Halepovic, Rittwik Jana, and Vijay Gopalakrishnan. On Leveraging Machine and Deep Learning for Throughput Prediction in Cellular Networks: Design, Performance, and Challenges. *IEEE Communications Magazine*, 2020.

[146] P. Ramakrishna. Rao. *Communication systems.* McGraw Hill Education (India), 2013.

[147] ITUR Recommendation. Guidelines for evaluation of radio transmission technologies for IMT-2000. *ITU-R M. 1225*, 1997.

[148] I. S. Reed and G. Solomon. Polynomial Codes Over Certain Finite Fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, jun 1960.

[149] J Reed. *Introduction to ultra wideband communication systems.* 2005.

[150] AH Reeves. The Past Present and Future of PCM. *IEEE Spectrum*, (May):58–63, 1965.

[151] Janne Riihijarvi and Petri Mahonen. Machine Learning for Performance Prediction in Mobile Cellular Networks. *IEEE Computational Intelligence Magazine*, 2018.

[152] Michele Rondinone, Julen Maneros, Daniel Krajzewicz, Ramon Bauza, Pasquale Cataldi, Fatma Hrizi, Javier Gozalvez, Vineet Kumar, Matthias Röckl, Lan Lin, Oscar Lazaro, Jérémie Leguay, Jérôme Härri, Sendoa Vaz, Yoann Lopez, Miguel Sepulcre, Michelle Wetterwald, Robbin Blokpoel, and Fabio Cartolano. ITETRIS: A modular simulation platform for the large scale evaluation of cooperative ITS applications. *Simulation Modelling Practice and Theory*, 2013.

[153] Ruifeng Zhang. Blind OFDM channel estimation through linear precoding: a subspace approach. In *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers, 2002.*, volume 1, pages 631–633. IEEE.

[154] Alassane Samba, Yann Busnel, Alberto Blanc, Philippe Dooze, and Gwendal Simon. Instantaneous throughput prediction in cellular networks: Which information is needed? In *Proceedings of the IM 2017 - 2017 IFIP/IEEE International Symposium on Integrated Network and Service Management*, pages 624–627. Institute of Electrical and Electronics Engineers Inc., jul 2017.

[155] Wojciech Samek, Slawomir Stanczak, and Thomas Wiegand. THE CONVERGENCE OF MACHINE LEARNING AND COMMUNICATIONS. Technical report.

[156] Tachporn Sanguanpuak, Sudarshan Guruacharya, Nandana Rajatheva, Mehdi Bennis, and Matti Latva-Aho. Multi-Operator Spectrum Sharing for Small Cell Networks: A Matching Game Perspective, 2017.

[157] Raja Sattiraju, Andreas Weinand, and Hans D. Schotten. Performance Analysis of Deep Learning based on Recurrent Neural Networks for Channel Coding. In *International Symposium on Advanced Networks and Telecommunication Systems, ANTS*, volume 2018-Decem. IEEE Computer Society, jul 2018.

[158] Raja Sattiraju, Andreas Weinand, and Hans D. Schotten. Channel Estimation in C-V2X using Deep Learning. In *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–5. IEEE, dec 2019.

[159] SBDist. D1 - System Requirements Specification. Technical report, 2016.

[160] T.M. Schmidl and D.C. Cox. Robust frequency and timing synchronization for OFDM. *IEEE Transactions on Communications*, 45(12):1613–1621, 1997.

[161] Indranil Sen and David W. Matolak. Vehicle-vehicle channel models for the 5-GHz band. *IEEE Transactions on Intelligent Transportation Systems*, 9(2):235–245, jun 2008.

[162] C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(4):623–656, oct 1948.

[163] C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3):379–423, jul 1948.

[164] Sandeepika Sharma and Brahmjit Singh. Cooperative Reinforcement Learning Based Adaptive Resource Allocation in V2V Communication. In *2019 6th International Conference on Signal Processing and Integrated Networks, SPIN 2019*, 2019.

[165] Sandeepika Sharma and Brahmjit Singh. Context aware autonomous resource selection and Q-learning based power control strategy for enhanced cooperative awareness in LTE-V2V communication. *Wireless Networks*, 2020.

[166] Cong Shen, Cem Tekin, and Mihaela Van Der Schaar. A Non-Stochastic Learning Approach to Energy Efficient Mobility Management. *IEEE Journal on Selected Areas in Communications*, 2016.

[167] Shengli Zhou, B. Muquet, and G.B. Giannakis. Subspace-based (semi-) blind channel estimation for block precoded space-time OFDM. *IEEE Transactions on Signal Processing*, 50(5):1215–1228, may 2002.

[168] Osvaldo Simeone. A Very Brief Introduction to Machine Learning with Applications to Communication Systems. *IEEE Transactions on Cognitive Communications and Networking*, 4(4):648–664, dec 2018.

[169] Meryem Simsek, Mehdi Bennis, and Ismail Güvenç. Learning Based Frequency- and Time-Domain Inter-Cell Interference Coordination in HetNets. *IEEE Transactions on Vehicular Technology*, 2015.

[170] Kuldeep Singh and Jaspreet Kaur. Machine Learning based Link Cost Estimation for Routing Optimization in Wireless Sensor Networks. Technical Report 1, 2017.

[171] Amanjot Singh Toor and A K Jain. Bulletin of Electrical Engineering and Informatics A Survey on Wireless Network Simulators. 6(1):62–69, 2017.

[172] Benjamin Sliwa, Thomas Liebig, Robert Falkenberg, Johannes Pillmann, and Christian Wietfeld. Efficient machine-type communication using multi-metric context-awareness for cars used as mobile sensors in upcoming 5g networks. In *IEEE Vehicular Technology Conference*, 2018.

[173] Benjamin Sliwa and Christian Wietfeld. Empirical analysis of client-based network quality prediction in vehicular multi-MNO networks. In *IEEE Vehicular Technology Conference*, 2019.

[174] Mehran Soltani, Vahid Pourahmadi, Ali Mirzaei, and Hamid Sheikhzadeh. Deep Learning-Based Channel Estimation. *IEEE Communications Letters*, 2019.

[175] Christoph Sommer and Falko Dressler. *Vehicular networking*. Cambridge University Press, jan 2014.

[176] Christoph Sommer, Reinhard German, and Falko Dressler. Bidirectionally coupled network and road simulation for improved IVC analysis. *IEEE Transactions on Mobile Computing*, 2011.

[177] Manikantan Srinivasan, Vijeth J. Kotagi, and C. Siva Ram Murthy. A Q-Learning Framework for User QoE Enhanced Self-Organizing Spectrally Efficient Network Using a Novel Inter-Operator Proximal Spectrum Sharing. *IEEE Journal on Selected Areas in Communications*, 2016.

[178] Yaohua Sun, Mugen Peng, Yangcheng Zhou, Yuzhe Huang, and Shiwen Mao. Application of Machine Learning in Wireless Networks: Key Techniques and Open Issues. *IEEE Communications Surveys and Tutorials*, 21(4):302–3108, oct 2019.

[179] Kinji Taguchi, Takanobu Shimada, Jun Yoshimoto, Tetsuya Kuwabara, and Yasuhiro Akasofu. High-strength aluminum wires for low-voltage automotive engine wiring harnesses. *SEI Technical Review*, 2017.

[180] Ian Tan, Wanbin Tang, Ken Laberteaux, and Ahmad Bahai. Measurement and analysis of wireless channel impairments in DSRC vehicular communications. In *IEEE International Conference on Communications*, pages 4882–4888, 2008.

[181] Le Thanh Tan and Rose Qingyang Hu. Mobility-aware edge caching and computing in vehicle networks: A deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, 2018.

[182] Sahrish Khan Tayyaba, Hasan Ali Khattak, Ahmad Almogren, Munam Ali Shah, Ikram Ud Din, Ibrahim Alkhalifa, and Mohsen Guizani. 5G vehicular network resource management for improving radio access through machine learning. *IEEE Access*, 2020.

[183] Wang Tong, Azhar Hussain, Wang Xi Bo, and Sabita Maharjan. Artificial Intelligence for Vehicle-To-Everything: A Survey. *IEEE Access*, 7:10823–10843, 2019.

[184] X Tong, S Sun International Conference On Signal Information, , and undefined 2017. Long Short-Term Memory Network for Wireless Channel Prediction. *Springer*.

[185] TSGR. TR 138 913 - V14.3.0 - 5G; Study on scenarios and requirements for next generation access technologies (3GPP TR 38.913 version 14.3.0 Release 14). Technical report, 2017.

[186] TSGR. TS 136 101 - V14.5.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception (3GPP TS 36.101 version 14.5.0 Release 14). Technical report, 2017.

[187] TSGR. TS 136 212 - V14.2.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding (3GPP TS 36.212 version 14.2.0 Release 14). Technical report, 2017.

[188] TSGR. TS 136 213 - V15.2.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (3GPP TS 36.213 version 15.2.0 Release 15). Technical report, 2018.

[189] TSGR. TS 136 331 - V15.3.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification (3GPP TS 36.331 version 15.3.0 Release 15). Technical report, 2018.

[190] J.-J. van de Beek, O. Edfors, M. Sandell, S.K. Wilson, and P.O. Borjesson. On channel estimation in OFDM systems. In *1995 IEEE 45th Vehicular Technology Conference. Countdown to the Wireless Twenty-First Century*, volume 2, pages 815–819. IEEE.

[191] R. Vinayakumar, K. P. Soman, and Prabaharan Poornachandran. Applying deep learning approaches for network traffic prediction. In *2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017*, 2017.

[192] Chao Wang, Ruida Zhou, Jing Yang, and Cong Shen. A Cascading Bandit Approach to Efficient Mobility Management in Ultra-Dense Networks. In *IEEE International Workshop on Machine Learning for Signal Processing, MLSP*, volume 2019-Octob. IEEE Computer Society, oct 2019.

[193] Chujie Wang, Zhifeng Zhao, Qi Sun, and Honggang Zhang. Deep Learning-Based Intelligent Dual Connectivity for Mobility Management in Dense Network. In *IEEE Vehicular Technology Conference*, 2018.

[194] Jingrong Wang, Kaiyang Liu, Minming Ni, and Jianping Pan. Learning Based Mobility Management under Uncertainties for Mobile Edge Computing. In *2018 IEEE Global Communications Conference, GLOBECOM 2018 - Proceedings*, 2018.

[195] Min Wang, Martin Winbjork, Zhang Zhang, Ricardo Blasco, Hieu Do, Stefano Sorrentino, Marco Belleschi, and Yunpeng Zang. Comparison of LTE and DSRC-Based Connectivity for Intelligent Transportation Systems. In *IEEE Vehicular Technology Conference*, volume 2017-June. Institute of Electrical and Electronics Engineers Inc., nov 2017.

[196] Xiaofei Wang, Yiwen Han, Victor C.M. Leung, Dusit Niyato, Xueqiang Yan, and Xu Chen. Convergence of Edge Computing and Deep Learning: A Comprehensive Survey, 2020.

[197] Ying Wang, Jing Xu, and Lisi Jiang. Challenges of system-level simulations and performance evaluation for 5G wireless networks. *IEEE Access*, 2:1553–1561, 2014.

[198] Yona Wang, Margaret Martonosi, and Li Shiuan Peh. A supervised learning approach for routing optimizations in wireless sensor networks. In *REALMAN 2006 - Proceedings of Second International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality*, 2006.

[199] Axel Wegener, Michat Piórkowski, Maxim Raya, Horst Hellbrück, Stefan Fischer, and Jean Pierre Hubaux. TraCI: An interface for coupling road traffic and network simulators. In *Proceedings of the 11th Communications and Networking Simulation Symposium, CNS'08*, 2008.

[200] Andreas Weinand, Raja Sattiraju, Michael Karrenbauer, and Hans D. Schotten. Supervised learning for physical layer based message authentication in URLLC scenarios. In *IEEE Vehicular Technology Conference*, 2019.

[201] Chao Kai Wen, Wan Ting Shih, and Shi Jin. Deep learning for massive MIMO CSI feedback. *IEEE Wireless Communications Letters*, 7(5):748–751, oct 2018.

[202] Yong Wen and Guangxi Zhu. Prediction for non-Gaussian self-similar traffic with neural network. In *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, volume 1, pages 4224–4228, 2006.

[203] Dilranjan S. Wickramasuriya, Calvin A. Perumalla, Kemal Davaslioglu, and Richard D. Gitlin. Base station prediction and proactive mobility management in virtual cells using recurrent neural networks. In *2017 IEEE 18th Wireless and Microwave Technology Conference, WAMICON 2017*, 2017.

[204] Jin Wu, Jing Liu, Zhangpeng Huang, and Shuqiang Zheng. Dynamic fuzzy Q-learning for handover parameters optimization in 5G multi-tier networks. In *2015 International Conference on Wireless Communications and Signal Processing, WCSP 2015*, 2015.

[205] Xiangyang Zhuang, Zhi Ding, and A.L. Swindlehurtst. A statistical subspace method for blind channel identification in OFDM communications. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, volume 5, pages 2493–2496. IEEE.

[206] Liang Xiao, Tianhua Chen, Guoan Han, Weihua Zhuang, and Limin Sun. Game Theoretic Study on Channel-Based Authentication in MIMO Systems. *IEEE Transactions on Vehicular Technology*, 2017.

[207] Xiaodong Cai and A.N. Akansu. A subspace method for blind channel identification in OFDM systems. In *2000 IEEE International Conference on Communications. ICC 2000. Global Convergence Through Communications. Conference Record*, volume 2, pages 929–933. IEEE.

[208] Lei Xu and Arumugam Nallanathan. Energy-Efficient Chance-Constrained Resource Allocation for Multicast Cognitive OFDM Network. *IEEE Journal on Selected Areas in Communications*, 2016.

[209] Shugong Xu. Channel Coding with Deep Learning. In *Machine Learning for Future Wireless Communications*, pages 265–285. Wiley, feb 2020.

[210] Yue Xu, Wenjun Xu, Zhi Wang, Jiaru Lin, and Shuguang Cui. Deep Reinforcement Learning Based Mobility Load Balancing under Multiple Behavior Policies. In *IEEE International Conference on Communications*, 2019.

[211] Yisheng Xue and Xuelong Zhu. Optimum wireless channel tracking based on second-order LMS algorithm. *Electronics Letters*, 37(15):989, 2001.

[212] Charlie Chen Yui Yang, Michael Ketcham, David Lu, Huang Lee, Raj Savoor, and David Kinsey. Performance monitoring with predictive QoS analysis of LTE backhaul. In *Proceedings - 2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2011*, 2011.

[213] Hao Ye, Le Liang, Geoffrey Ye Li, Joonbeom Kim, Lu Lu, and May Wu. Machine Learning for Vehicular Networks: Recent Advances and Application Examples. *IEEE Vehicular Technology Magazine*, 13(2):94–101, jun 2018.

[214] Jungkeun Yoon, Mingyan Liu, and Brian Noble. Random waypoint considered harmful. In *Proceedings - IEEE INFOCOM*, 2003.

[215] Shuai Yu, Xin Wang, and Rami Langar. Computation offloading for mobile edge computing: A deep learning approach. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, 2018.

[216] Chaoqun Yue, Ruofan Jin, Kyoungwon Suh, Yanyuan Qin, Bing Wang, and Wei Wei. LinkForecast: Cellular Link Bandwidth Prediction in LTE Networks. *IEEE Transactions on Mobile Computing*, 2018.

[217] Chaoyun Zhang, Paul Patras, and Hamed Haddadi. Deep Learning in Mobile and Wireless Networking: A Survey. *IEEE Communications Surveys and Tutorials*, 21(3):2224–2287, 2019.

[218] Yi Zhang, Wee Peng Tay, Kwok Hung Li, Moez Esseghir, and Dominique Gaïti. Learning temporal-spatial spectrum reuse. In *IEEE Transactions on Communications*, 2016.

[219] Yiming Zhou, Cong Shen, and Mihaela Van Der Schaar. A Non-stationary online learning approach to mobility management. In *IEEE Transactions on Wireless Communications*, 2019.

[220] Zijun Zhao, Xiang Cheng, Miaowen Wen, Bingli Jiao, and Cheng-Xiang Wang. Channel Estimation Schemes for IEEE 802.11p Standard. *IEEE Intelligent Transportation Systems Magazine*, 5(4):38–49, 2013.

Acronyms

3GPP	Third Generation Partnership Project
3GPP2	Third Generation Partnership Project-2
ADAS	Advanced Driver Assistance Systems
AEI	Availability Estimation and Indication
AGV	Autonomous Guided Vehicle
AI	Artificial Intelligence
AM	Amplitude Modulation
AN	Access Network
AoA	Angle of Arrival
AR	Availability Request
ARIB	Association of Radio Industries and Businesses
ARQ	Automatic Repeat Request
ASTM	American Society for Testing and Materials
AT&T	American Telephone and Telegraph Company
ATP	Autonomous Train Pairing
AWGN	Additive While Gaussian Noise
BCH	Bose Chaudhuri Hocquenghem
BER	Bit Error Rate
BMW	Bayerische Motoren Werke AG
BPSK	Binary Phase Shift Keying
BSM	Basic Safety Message
BSS	Basic Service Set
CA	Carrier Aggregation
CAM	Cooperative Awareness message
CAN	Controller Area Network
CAZAC	Constant Amplitude Zero AutoCorrelation
CCH	Control Channel
CFO	Coarse Frequency Offset
C-ITS	Cooperative - Intelligent Transportation Systems
CN	Core Network
CoMP	Coordinated MultiPoint
COTS	Commercial Off The Shelf
CP	Cyclic Prefix
CPFSK	Continous-Phase Frequency-Shift Keying
CRC	Cyclic Redundancy Check

CS	Circuit Switched
CSI	Channel State Information
CSMA	Carrier Sense Multiple Access
CSMA/BA	Carrier Sense Multiple Access with Bitwise Arbitration
C-V2X	Cellular Vehicle-to-everything
D2B	Domestic Digital Bus
D2D	Device-to-Device
DC	Direct Current
DCI	Downlink Control Information
DENM	Decentralized Environment Notification Message
DFT	Discrete Fourier Transformation
DL	DownLink
DMRS	Demodulation Reference Symbols
DNPW	Do Not Pass Warning
DQPSK	Differential Quadrature Phase Shift Keying
DSI	Distributed Systems Interface
DSRC	Dedicated Short Range Communications
ECU	Electronic Control Unit
EEBL	Emergency Electronic Brake Light
EFC	Electronic Fee Collection
eNB	eNodeB
EPC	Evolved Packet Core
EU	European Union
E-UTRAN	Evolved Universal Terrestrial Radio Access Network)
FCC	Federal Communications Commission
FDM	Frequency Division Multiplexing
FEC	Forward Error Correction
FECC	Forward Error Control Coding
FFT	Fast Fourier Transform
FM	Frequency Modulation
FTDMA	Flexible Time Division Multiple Access
GFSK	Gaussian Frequency Shift Keying
GM	General Motors
GOF	Glass Optical Fiber
GPS	Global Positioning System
GSM	Global System for Mobile communications
HARQ	Hybrid Automatic Repeat Request
I2C	Inter-Integrated Circuit

IBM	International Business Machine
IC	Integrated Circuit
IDB	ITS Data Bus
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IFFT	Inverse Fast Fourier Transform
IMA	Intersection Movement Assist
IoT	Internet of Things
IPv6	Internet Protocol version 6
ISDN	Integrated Services Digital Network
ISI	Inter-Symbol Interference
ISM	Industrial, Scientific and Medical
ISO	International Organization for Standardization
ITS	Intelligent Transportation Systems
ITU	International Telecommunication Union
LAN	Local Area Network
LDPC	Low-Density Parity-Check
LED	Light Emitting Diode
LIN	Local Interconnect Network
LLC	Logical Link Control
LLR	Log-Likelihood Ratio
L-LTF	Legacy Long Training Field
LMS	Least Mean Squares
LS	Least Squares
L-STF	Legacy Short Training Field
LTE	Long Term Evolution
LTE-A	LTE-Advanced
MAC	Media Access Control Layer
MANET	Mobile Adhoc NETwork
MCS	Modulation & Coding Scheme
METIS	Mobile and Wireless communications Enablers for Twenty-twenty (2020) Information Society
MI	Motorola Interconnect
MIMO	Multiple Input Multiple Output
ML	Machine Learning
MML	Mobile Multimedia Link
MMSE	Minimum Mean Squared Error
MOST	Media Oriented Systems Transport
MPDU	MAC Protocol Data Unit
MPE	Mean Percentage Error
MSE	Mean Squared Error
MSK	Minimum Shift Keying
MTS	Mobile Telephone Service

NRZ	Non Return to Zero
NTT	Nippon Telegraph and Telephone
NXID	V2X Scrambling Identity
OBD-II	On Board Diagnostics-II
OCB	Outside the Context of BSS
OEM	Original Equipment Manufacturer
OFDM	Orthogonal Frequency Division Multiplexing
OPEN	One-Pair Ether-Net
OQPSK	Offset QPSK
OSI	Open Systems Interconnect
PAM	Pulse Amplitude Modulation
PAN	Personal Area Network
PAPR	Peak to Average Power Ratio
PAT	Pilot Assisted Transmission
PCM	Pulse Coded Modulation
PDR	Packet Drop Ratio
PER	Packet Error Rate
PHY	Physical Layer
PLANET	Philips Lite Automotive NETwork
PLCP	Physical Layer Convergence Protocol
PMD	Physical Medium Access
PMF	Probability Mass Function
PoE	Power over Ethernet
POF	Polymeric Optical Fiber
PPDU	PLCP Protocol Data Unit
PRB	Physical Resource Block
ProSe	Proximity Services
PS	Packet Switched
PSBCH	Physical SideLink Broadcast Channel
PSDU	PLCP Service Data Unit
PSK	Phase Shift Keying
PSCCH	Physical Sidelink Control Channel
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RAN	Radio Access Network
RAT	Radio Access Technology
RE	Resource Element
RIV	Resource Indication Value
RMS	Root Mean Square
RP	Resource Pool

RSC	Recursive Systematic Convolutional
RSSI	Received Signal Strength Indicator
RSU	Road Side Unit
RTL	Reliable Transmission Link
RTPGE	Reduced Twister Pair Gigabit Ethernet
RTTI	Real-Time Traffic Information
SA	Scheduling Assignment
SAE	Society of Automotive Engineers
SBCCH	Sidelink Broadcast Control Channel
SC	sub-carrier
SC-FDMA	Single-Carrier Frequency Division Multiple Access
SCH	Shared Channel
SCI	Sidelink Control Information
SER	Symbol Error Rate
SIG	Special Interest Group
SINR	Signal to Interference plus Noise Ratio
SL	SideLink
SL-BCH	Sidelink Broadcast Channel
SL-SCH	Sidelink Shared Channel
SMSC	Standard Microsystems Corporation
SNR	Signal to Noise Ratio
SoC	System on Chip
SPI	Serial Peripheral Interface
SPSS	Semi Persistant Subchannel Selection
STCH	Sidelink Traffic Channel
TCP	Transmission Control Protocol
TCP/IP	Transport Control Protocol/Internet Protocol
TDM	Time Division Multiplexing
TDMA	Time Division Multiple Access
TDoA	Time Difference of Arrival
TMC	Traffic Message Channel
ToF	Time of Flight
TSN	Time Sensitive Networking
TTCAN	Time Triggered Controller Area Network
TW-ToF	Two Way-Time of Flight
UART	Universal Asynchronous Receiver Transmitter
UDP	User Datagram Protocol
UE	User Equipment
UL	Uplink
uMTC	ultra-reliable machine-type communication
UMTS	Universal Mobile Telecommunications System
URLLC	Ultra-Reliable Low-Latency Communications

USB	Universal Serial Bus
UWB	Ultra-Wide Band
V2I	Vehicle-to-Infrastructure
V2N	Vehicle-to-Network
V2P	Vehicle-to-Pedestrian
V2V	Vehicle-to-Vehicle
V2X	Vehicular-to-Everything
VAN	Vehicle Area Network
VLC	Visible Light Communication
VLSI	Very Large Scale Integration
WAVE	Wireless Access in Vehicular Environments
WSMP	Wave Short Message Protcol
ZC	Zadoff-Chu
ZF	Zero-Forcing

Curriculum Vitae

Profile Raja Sattiraju

Employment History

04/2022– Qualcomm Germany GmbH

- Network Optimization using ML based techniques

03/2012-03/2022 University of Kaiserslautern-Landau (RPTU)

- Funded projects (national): SBDist, ConVeX, AI4Mobile, METIS2020, CoMoRa
- Lecturer for master course Vehicular Networking (2015-2022) & supervisor of multiple Bachelor's and Master's theses.

Education

10/2008–03/2012 University of Kaiserslautern-Landau (RPTU), prev. University of Kaiserslautern (TUK)

- Program: Commercial Vehicle Technology
- Degree: Masters (M. Sc.)
- Master thesis: Code Instrumentation and Timing Analysis of Automotive Embedded Systems

10/2004–06/2008 Bachelors in Mechanical Engineering / Jawaharlal Nehru Technological University, Hyderabad