東京大学
情報理工学系研究科 創造情報学専攻
修士論文

協調型自動運転のための地理的特性を考慮したネットワーク分析・可視化

Geographic-Aware Network Analysis and Visualization for Cooperative Automated Driving

神原 滉一
Koichi Kambara

指導教員　塚田 学 准教授

2022 年 1 月

概要

近年，協調型の自動運転が交通安全や交通流の効率化につながるとして注目されている．協調型自動運転とは，自動運転車が周囲の車両や道路に設置した路側機と通信を行い，自車の車載センサでは認識できなかった情報やタスクを共有するシステムのことである．協調型自動運転の重要な要件の一つは，すべての車両が適切なタイミングで適切なメッセージを受信することであり，そのためには，一定以上の通信性能の担保が必要である．事前に通信性能を把握しておくことによって，自動運転車は適切な認知方法の選択や，経路計画を行うことができる．そこで，本研究では，協調型自動運転のための地理的特性を考慮した通信性能分析，可視化システムを提案する．このシステムでは，車両が一度通過した場所の通信性能を分析，可視化し，クラウドに保存することによって，次に同じ場所を通過する車が，事前に通信性能を参照し，移動ルート決定ができるようになる．本提案システムを自動運転のためのオープンソースソフトウェアである Autoware とダイナミックマッププラットフォームである DM2.0PF を用いて実装し，インドアでのテストと柏の葉キャンパステストコース，愛知県春日井市石尾台地区にて，アウトドア実験，評価を行った．実環境に近い，住宅街での実験の結果，クラウドに保存した通信性能は，動的物体や電波干渉などに影響を受けるため，必ずしも未来の通信性能と一致するわけではないが，ある程度参考にすることができることを確認できた．

Abstract

In recent years, cooperative automated driving has been attracting attention to improve traffic safety and streamline traffic flow. Cooperative automated driving is a system in which an automated vehicle communicates with surrounding vehicles and roadside units installed on the road to share information and tasks that the vehicle's onboard sensors cannot recognize. One of the critical requirements for cooperative automated driving is that all vehicles receive appropriate messages at appropriate times. For this purpose, measurement and visualization of network performance are essential. Knowing the communication performance in advance allows an automated vehicle to select appropriate cognitive methods and plan its route. Therefore, we propose a communication performance analysis and visualization system that considers geographical characteristics for cooperative automated driving. The system stores the communication performance of a location once passed by a vehicle in the cloud so that the next vehicle that passes by the same location can see the communication performance in advance. We implemented the proposed system using Autoware, an open source software for automated driving, and DM2.0PF, a dynamic map platform. We conducted indoor tests and outdoor experiments and evaluations at the Kashiwanoha Campus test course and the Ishiodai area in Kasugai, Aichi Prefecture. The results of the experiments in a residential area, which is close to the real environment, showed that the communication performance stored in the cloud does not necessarily match the future communication performance since it is affected by dynamic objects and radio interference, but it can be used as a reference to some extent.

目次

第 1 章　序論 1
1.1　背景 1
1.2　目的 2
1.3　構成 3

第 2 章　協調型自動運転の現状と課題 4
2.1　背景知識 4
2.2　関連研究 8
2.3　現状の課題 14

第 3 章　システムアーキテクチャ 16
3.1　要求事項および本研究の設計方針 16
3.2　システムアーキテクチャ 17

第 4 章　実装 20
4.1　DM2.0PF との統合 20
4.2　ネットワーク分析 21
4.3　ネットワーク可視化 23

第 5 章　実験・評価 26
5.1　実験概要 26
5.2　インドアテスト 28
5.3　アウトドアテスト 31
5.4　まとめ 42

第 6 章　結論 45
6.1　まとめ 45
6.2　今後の課題 45

発表文献と研究活動 47

参考文献 48

第 1 章

序論

1.1 背景

自動車は私達の日常生活において移動手段や輸送手段として，欠かせないものとなっている．平日における交通手段のうち，自動車を利用する人の割合が三大都市圏の場合，約 32% で，地方都市圏の場合約 61% に達する [1]．一方で，自動車にはいくつか問題もあり，その一つが交通事故である．警察庁交通局のデータによると，令和 3 年中に発生した交通事故（人身事故）件数は 305196 件で，2636 人の人が亡くなっている [2]．そのような背景から，運転操作からの解放や交通事故の防止，燃費向上のための近接距離での隊列走行といった安全・安心・快適・環境対策を目的として，自動運転技術に対する関心が高まっている [3]．例えば，千葉県柏市柏の葉地区では柏 ITS 推進協議会の主導のもと 2019 年 11 月 1 日から、柏の葉キャンパス駅から東京大学柏キャンパスの間で自動運転バスの長期営業運行実証実験が行われている [4]．本実証実験では，レベル 2 の自動運転が実現されており，走行領域は限定的で，運転手体も人であるが，部分的な運転自動化が行われている．また，Honda の SENSING Elite が搭載された LEGEND は，レベル 3 の自動運転であり，高速道路や渋滞時などの一定の条件下において，システムがドライバーに代わり，運転操作を行うことができる [5]．自動運転のレベル分けは図 1.1 のように定義されている．

上記で説明した自動運転の事例は，自車両に搭載されたセンサ情報のみを使用する自律型の自動運転である．自律型の自動運転では，交差点などの死角に存在する車両や歩行者を認知することができないという問題点が存在する．現状だと，自律型の自動運転で，死角から車両や歩行者が飛び出してくる可能性のある場所では，一時停止するのが一般的となっている．これに対して，協調型の自動運転では，周囲の車両や，道路上に設置された路側機と自車両が通信を行う．リアルタイムで CAM（Cooperative Awareness Message） [7] や CPM（Cooperative Perception Message） [8, 9] のような，自車両の位置や認識した物体の情報を共有するため，自車のセンサでは得られない情報を得ることができる．

2015 年 9 月に発売されたトヨタの ITS（Intelligent Transport Systems：高度道路交通システム） Connect は，ITS 周波数向け 760MHz 帯を利用した車両と車両，車両と道路をつなぐ無線システムである [10]．既に市販車として一般向けに発売されており，緊急車両の存在を

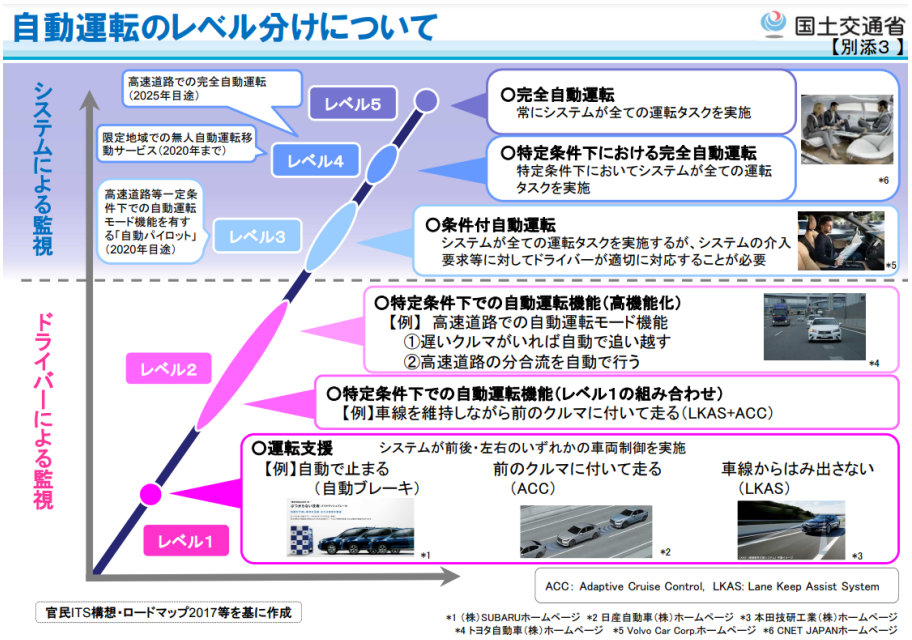

図 1.1. 自動運転レベル分け [6]

通知したり，対応交差点で赤信号の注意喚起，信号待ち時間の目安表示，右折時の対抗車両や歩行者の存在の検知による注意喚起を行うことができる．

移動体である自動運転車両の通信状況は，通信先との距離やネットワークの干渉，混雑等により常に変動しており，協調認知，遠隔運転のために十分な通信性能を維持できないことがある．通信性能が低いと，パケットロスによって，死角の先の情報が得られないことや，遠隔運転での遅延によって操作が間に合わないなど，危険が存在する．事前に通信性能を把握することによって，通信性能の悪い場所を通らない経路計画や，通信性能が悪いため，協調型の自動運転ではなく，自律型の自動運転に切り替えることで安全を確保することができるようになる．

協調型の自動運転のためだけではなく，自動運転車の通信性能は非常に重要な要素となっている．例えば，自動運転車の緊急時対応手段として，遠隔運転による，操作支援が期待されている [11]．遠隔地の人間の知能と局所的な自律制御の組み合わせにより，安全で効率的なモビリティの開発を加速させることが可能である．一方で，遠隔支援を期待通りに動作させるためには，通信性能や通信密度等を含む ODD（Operational Design Domain）を適切に設定することが重要である．

1.2　目的

本研究の目的は，自動運転のための通信性能の分析と可視化を行うことで，1.1 節で述べた課題を解決し，協調型自動運転車や遠隔運転者の走行を支援することである．そのために，地理的特性を考慮した通信性能の分析，可視化を行うシステムを提案し，自動運転用のオープン

ソースソフトウェアである Autoware [12] を使用して実装した．実装したシステムを，インドアでのテストで動作確認を行い，東京大学柏の葉キャンパステストコースと，愛知県春日井市石尾台地区にて，アウトドアで車両を走らせながら，通信性能を測定し，評価を行った．

1.3 構成

本論文の構成は次のようになっている．2 章では協調型自動運転の現状と関連研究，課題について述べる．3 章では，課題を解決するためのシステムアーキテクチャについて述べる．4 章では，実装方法について述べる．5 章では，実験した結果とその評価について述べる．6 章では，本研究のまとめと今後の課題を述べる．

第 2 章

協調型自動運転の現状と課題

本章では，協調型自動運転の現状と課題について述べる．まず，2.1 節で協調型自動運転に関する要素技術や背景知識を紹介する．次に，2.2 節で関連研究の紹介をする．最後に，2.3 節で現状の課題について述べる．

2.1 背景知識

本節では，本研究の背景知識について述べる．まず，自動運転と自動運転のオープンソースソフトウェアである Autoware を紹介する．次に，協調型 ITS と用いられる技術について紹介する．その後，ダイナミックマップについて紹介する．最後に，協調型自動運転で用いられるエッジコンピューティングについて説明する．

2.1.1 自動運転

一般的な自律型の自動運転車がどのように自車両の周囲の状況を認識しているか説明する．自動運転車には，以下のようなセンシングデバイスが搭載されている．

カメラ
: カメラから得た映像から，AI を用いて，信号の色の認識を行う．また，自己位置推定や，CNN と組み合わせて周囲の物体の検知や分類に用いられることもある．

LiDAR（Light Detection and Ranging）
: LiDAR とはレーザー光を照射して、その反射光の情報をもとに対象物までの距離や対象物の形などを計測するセンサである．点群地図とのマッチングによる自己位置推定や，周囲の物体認識に用いられる．

IMU（Inertial Measurement Unit / **慣性計測装置**）
: IMU とは速度センサにより並進運動，ジャイロセンサにより回転運動を計測するセンサである [13]．LiDAR と合わせて，自己位置推定を行うことで，自己位置推定の精度を向上させることができる．

GNSS（Global Navigation Satellite System / **全球測位衛星システム**）

GNSS とは米国の GPS、日本の準天頂衛星（QZSS），欧州連合の Galileo 等の衛星測位システムの総称である [14]．GNSS を用いることで，初期の自己位置推定を自動で行うことができる．また，LiDAR のみでは，同じようなビルが並んでいたり，特徴の少ない場所での自己位置推定が難しいが，GNSS を合わせて，自己位置推定を行うことで自己位置推定の精度を向上させることができる．

また，自動運転には，以下の 2 つの地図データが用いられる．

点群地図

点群地図とは，LiDAR が検知した周囲の物体や地形から反射してきた一つ一つのパルスを点として表現した地図データのことである．LiDAR を用いた SLAM ベースの自己位置推定を行う場合，事前に作成した点群地図と LiDAR から得られるリアルタイムなセンサデータをマッチングすることによって自己位置推定を行う．

ベクターマップ

走行可能領域や信号や標識、速度制限、優先道路などの交通ルールや注意が必要なエリアのような情報をまとめた地図データである．停止線の位置や車線の情報等を使用して走行経路の生成を行う際に用いられる．

上記のセンサと地図データから，自車両の自己位置推定を行い，周囲の物体の検知を行う．得られた情報とベクターマップを利用して経路計画を行い，車両を制御することによって自動運転が実現されている．

自動運転のオープンソースソフトウェアには，日本で開発された Autoware [12] や中国の Baido が主導している Apollo [15] がある．本研究では，日本の多くの会社や大学で利用されている [16]，Autoware を用いて実験を行う．Autoware のシステムアーキテクチャを図 2.1 に示す．Autoware は Linux と ROS をベースに開発されており，自動運転に必要な，自己位置推定，物体認識，経路計画，車両制御などの機能を備えている．また，標準の可視化ツールとして RViz [17] が使用されている．

2.1.2 協調型 ITS

協調型 ITS（Cooperative Intelligent Transport Systems）とは，図 2.2 のように，車載器，路側機，中央サーバ，歩行者端末が，道路交通を安全・安心・快適にするために情報とタスクを共有するシステムである [18]．車両同士で通信する車車間通信（Vehicle to Vehicle: V2V）や車両と道路に設置した路側機（Road side Unit: RSU）で通信する路車間通信（Vehicle to roadside Infrastructure: V2I），車両と歩行者端末間での通信（Vehicle to Pedestrian: V2P），車とネットワークとの接続（Vehicle to Network: V2N）車と様々なものを接続する通信（Vehicle to Everything: V2X）によって情報が送受信される．

送受信されるメッセージとして CAM，CPM，DENM（Decentralized Environmental Notification Message）[19] などがある．いずれも ETSI（欧州電気通信標準化機構 : European

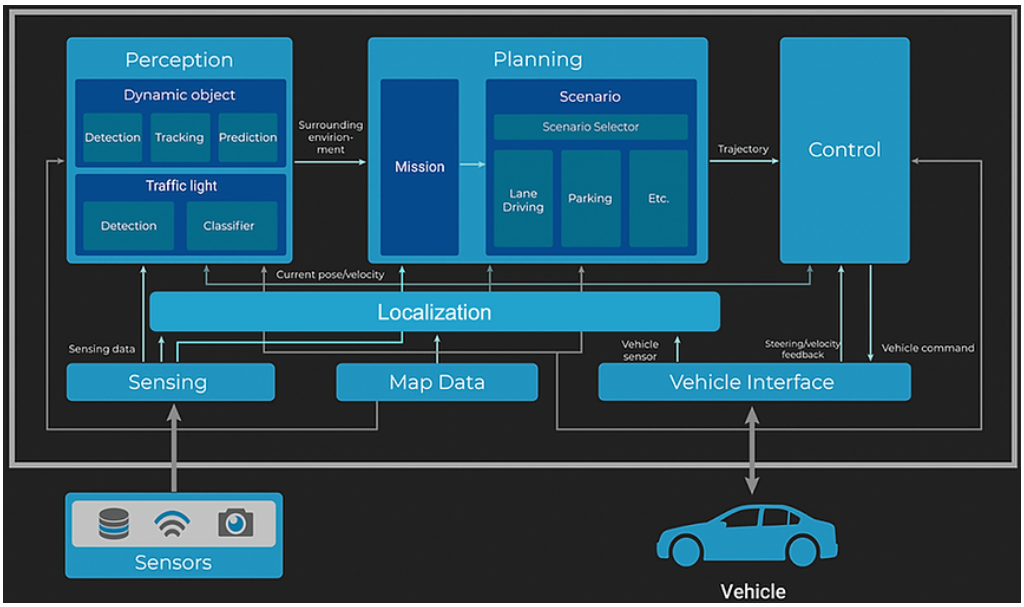

図 2.1. Autoware のシステムアーキテクチャ [12]

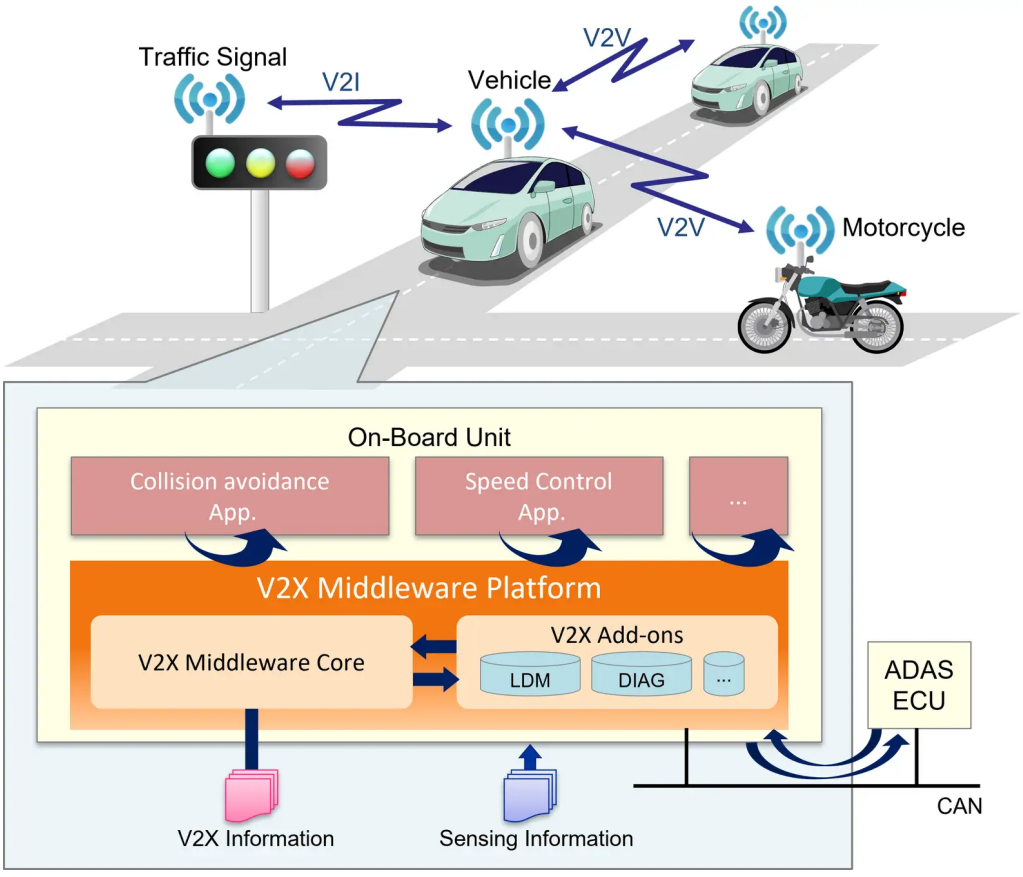

図 2.2. 協調型 ITS [18]

Telecommunications Standards Institute）によって標準仕様が定められている．

CAM
: 道路の利用者である，車両，RSU，歩行者などの間で自分の情報を伝えるためのメッセージ．送信元の位置情報，進行方向，速度などが含まれる．

CPM
: 車両または，RSU に設置された LiDAR やカメラが検出した周囲の物体の情報を共有するためのメッセージ．検出した物体の位置情報などが含まれる．

DENM
: 道路の利用者に対して検知されたイベントを通知するメッセージ．事故や故障などによる車線上への停止，悪天候や路面凍結などを示す．

2.1.3 ダイナミックマップ

動的情報と道路地図を扱う国際標準として，協調型 ITS での利用を目指したローカルダイナミックマップ（LDM：Local Dynamic Map）の研究が進んでいる [20, 21]．LDM は CAM，DENM，DPM などを受け取り，周辺の物体情報を提供するためのデータベースとしての役割を持つ．欧州の SAFESPOT [22] では，LDM で取り扱う情報は，情報の更新頻度に応じて，動的情報，準動的情報，準静的情報，静的情報の 4 種類に分けられる．

- 動的情報: 位置情報，進行方向，信号情報など更新頻度が非常に高い情報
- 準動的情報: 渋滞，道路工事，路面情報など更新頻度がやや高い情報
- 準静的情報: 標識，ランドマーク，路側設備など更新頻度がやや低い情報
- 静的情報: 道路地図など更新頻度が低い情報

実際に, 日本で開発されたダイナミックマッププラットフォームとして DM2.0PF（Dynamic Map 2.0 Platform）が挙げられる [23]．こちらはダイナミックマップ 2.0 コンソーシアム [24] で，欧州の LDM を参考に開発されたものであり，LDM にはなかったクラウドでのデータ収集が想定されている．DM2.0PF の構成を図 2.3 に示す．クラウド，エッジ，車両にまたがる分散システムとなっており，データを保管するためのデータベースと他のノードに情報を共有するための通信部分で構成されている．車両からエッジの通信はエニキャストによって行われ，エッジから車両への通信はマルチキャストで行われる．エッジからクラウドへはインターネットを介して，情報を送受信し，クラウドにデータが収集される．エッジサーバーの主な役割として，クラウドよりも車両に近い場所でリアルタイムなデータ処理を行い，車両と情報を共有している．クラウドの役割としては，広域の大規模データを扱い，応答時間にリアルタイム性を求めない処理やエッジ間の連携を行っている．

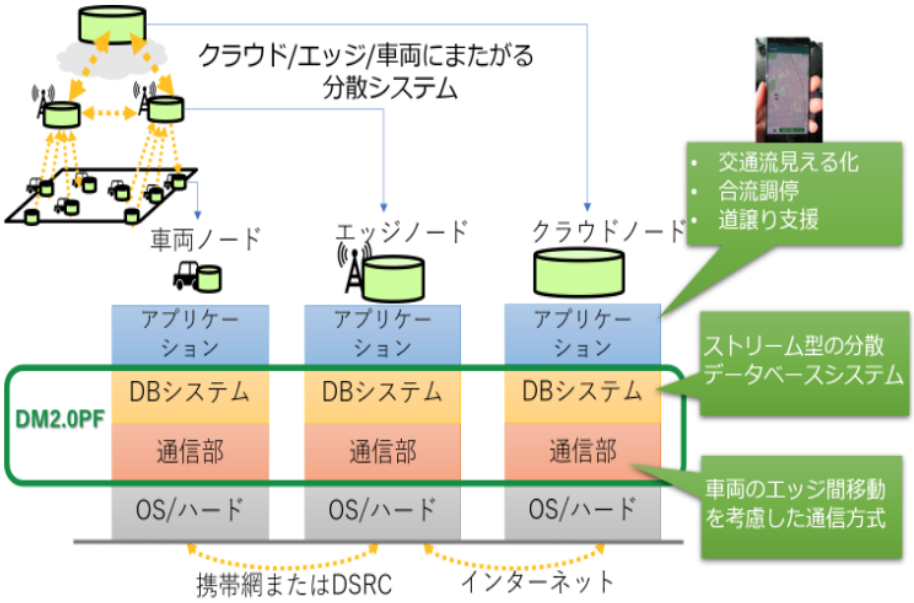

図 2.3. DM2.0PF の構成図 [24]

2.1.4　マルチアクセスエッジコンピューティング

協調型自動運転を実現する上で，重要な要素の一つが通信遅延である．例えば，50km で走行中の車両の位置情報を CPM によって，RSU に共有したときに，0.5 秒の遅延が発生した場合，受信側では車両が走行している位置情報と CPM にて受け取った位置情報とで約 7m の差分が生じてしまうという課題が存在する．そのため，通信遅延の削減可能なマルチアクセスエッジコンピューティング [25] によって，クラウドを介さないエッジ処理を行うことでリアルタイム処理が可能となる．マルチアクセスコンピューティングの構成図を図 2.4 に示す．RSU で処理させることによって，ネットワークリソースだけでなく，ストレージやコンピューティングリソースの面でも効率的になることが期待される．

2.2　関連研究

本節では，本研究の関連研究について説明する．まず，2.2.1 節で，DM2.0PF の関連研究について紹介する．次に 2.2.2 節で Autoware を使って RSU から CAM を受信する研究，2.2.3 節で車両の遠隔操作の関連研究を紹介する．最後に 2.2.4 節，2.2.5 節でフィールドの地理的特性を考慮した，通信分析ツールを紹介する．

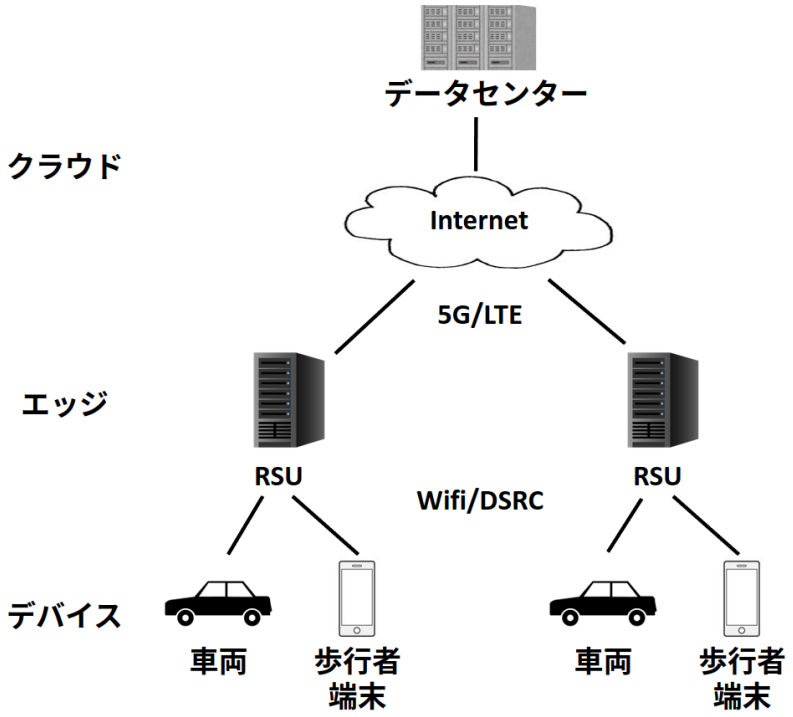

図 2.4. マルチアクセスコンピューティングのアーキテクチャ

2.2.1 DM2.0PF

DM2.0PF を使用した研究について 3 つ紹介する．DynamicMap 2.0: A Traffic Data Management Platform Leveraging Clouds, Edges and Embedded Systems [23] では，DM2.0PF によって，RSU と車両間で道路地図，静的情報，動的情報，予測情報を共有する．実際にシミュレーションにて，DM2.0PF を適用した例を図 2.5 に示す．本図は，T 字路の合流地点での交通流を最適化するためのシミュレーションであり，接近してくる車両に対して，アシストメッセージを送信する．そのメッセージに従って，各車両は加速，減速，車線変更を行う．また，緑の車のために，青の車が車線変更をして，緑の車のスペースを確保している．

ダイナミックマップ 2.0 を活用した自動運転車両に対するインフラ情報配信システム [26] では，車両とインフラとの情報共有を行っている．具体的には，信号機にて，信号機の灯火状態や信号機に取り付けられた，LiDAR，カメラから得られた物標情報の配信を行う．自動運転車がこれを受け取り，自動運転制御に活用することによって，高度な自動運転システムを構築することが可能となる．

また，愛知県春日井市高蔵寺を実証フィールドとして，「ゆっくり自動運転プロジェクト」の実証実験と連携し，ダイナミックマップのサービスを地域住民へ提供している [27]．具体的には，高齢者等の移動困難車を主な対象とした，カーシェアリングサービスや，人だけでなく，商品の配達を行うサービスなどが挙げられる．

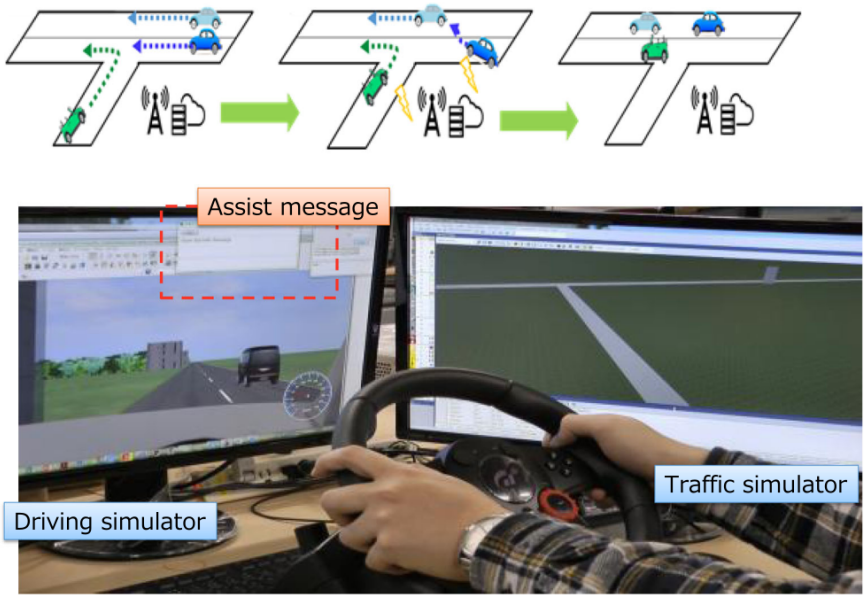

図 2.5. DM2.0PF による交差点調停 [23]

2.2.2　AutoC2X [28] と AutoMCM [29]

AutoC2X では，Autoware に OpenC2X を拡張，統合することで，自車の車載センサでは得られない情報を受信した．具体的には，RSU で認識した周辺の物体を座標変換し，CAM を生成する．生成した CAM を車両に送信することによって，受信した車両では，図 2.6 のように，死角にある物体を認識することができる．図 2.6 の下部の赤い線が LiDAR が検知した波形であり，付近の緑色のバウンディングボックスが認識した物体である．これに対して，上部の緑色のバウンディングボックスは赤い波形が届かない位置（LiDAR の認識範囲外）であるが，RSU から送信されてきた CAM によって協調認知が行われている．これに加えて，AutoMCM では，MCM（Maneuver Coordination Message）を使用して，車線変更時に適切な車両に減速の指示を行い、スムーズな走行を行うことができるかの検証を行った．MCM を使用した場合，車線変更時に適切に走行調停を行うことができるため，急激な減速等がなくなり，目的地までの到達時間が短く，快適な走行をすることができた．また，パケットロスを変化させたときに目的地までの到着時間の変化が図 2.7 のようになった．この図は横軸が，パケットロスを表しており，縦軸が到着時間の割合を，それぞれの色が到着時間を表している．例えば濃い青で示された領域は，目的地までの到着時間が 29 秒以下であった場合である．この図をみると，MCM による走行調停には，パケットロスの割合が大きく関係しており，パケットロスが 60% を超えると到着時間はほとんど 31 秒以上となっていることがわかる．このように協調型自動運転では，通信性能が重要な要素の一つである．

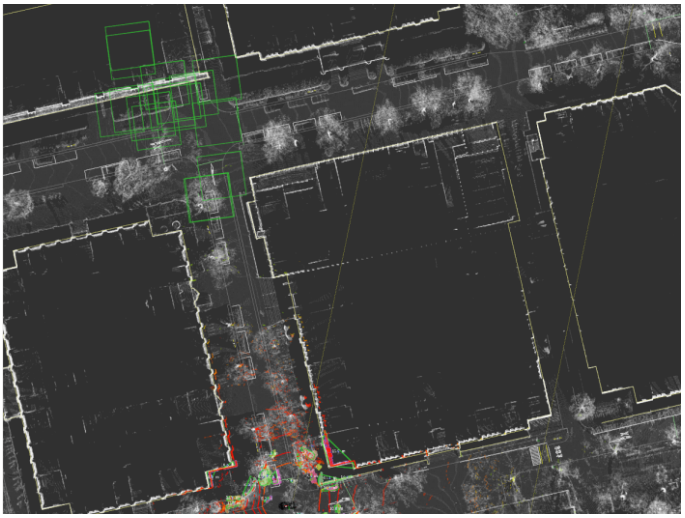

図 2.6. AutoC2X の動作画面 [28]

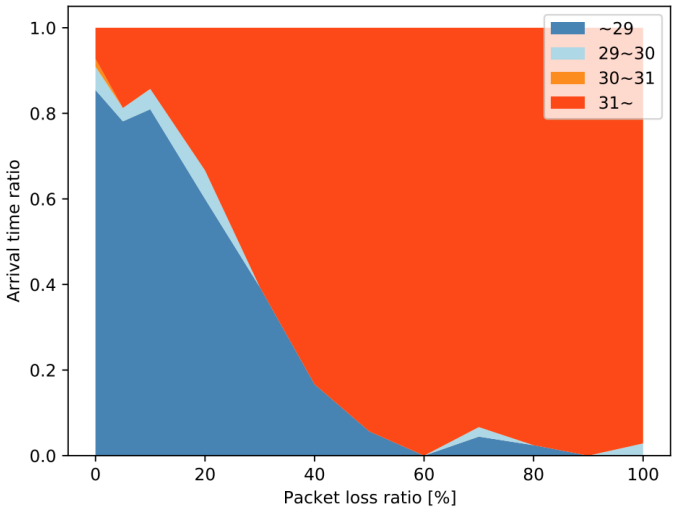

図 2.7. AutoMCM（パケットロスを変化させたときの目的地までの到着時間） [29]

2.2.3 遠隔運転環境の操作性向上のための遅延の見える化機能 [30]

自動運転車の緊急時対応手段として，遠隔地にいるオペレーターが車両を操作する遠隔運転機能が期待されている [11]．遠隔運転の操作性の低下要因は数多く存在し，中でも遅延時間は遠隔運転の操作性に大きな影響を与える要因として挙げられる [31]．特に，遅延時間が 700ms 以上のとき，遠隔運転の操作性が大きく損なわれることが明らかとなっており [32]，170ms 以下のとき，遅延時間が遠隔操作の操作性に与える影響が小さいことが示されている [33]．文献 [30] では，遠隔操作卓と遠隔運転車を無線ネットワークで接続し，通信遅延を可視化し，遅延の可視化機能によって遠隔運転の操作性が向上することを示した．

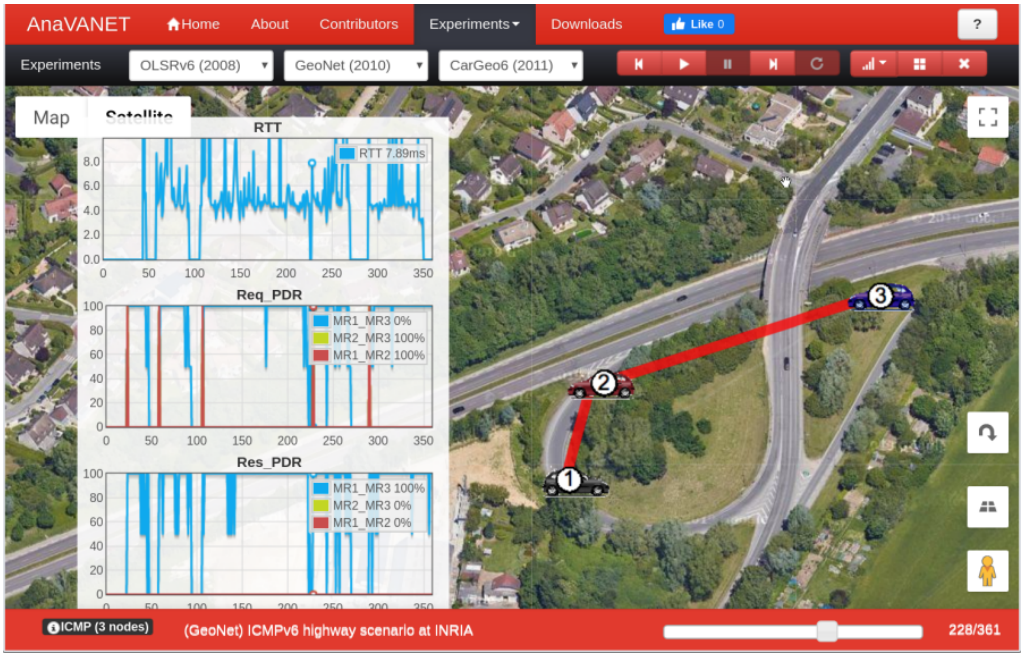

図 2.8. Anavanet の動作画面 [36]

2.2.4　AnaVanat

V2X 通信のフィールド実験を行った研究では通信性能の分析に ping や iperf などのツールを採用している場合もある [34, 35]．しかし，ping や iperf などのツールにはフィールドの地理的特性を反映していない，V2X 通信で使われるプロトコルの特性を反映していないなどの問題がある．そこで，地理的特性を考慮した V2X 通信分析ツールとして AnaVANET（Analyzer for Vehicular Ad-hoc Networks） [36] がある．このシステムは V2X の通信プロトコルである GeoNetworking の解析を可能としており，GPS ログと TCP DUMP ログを入力として gnuplot によるグラフ化や Google マップによる可視化をしている（図 2.8）．GeoNetworking とは，図 2.9 のように送信元車両から宛先の車両に情報を送信する際に，直接は通信できない場所にいるときでも，他の車両を中継してマルチホップすることで，宛先まで情報を届ける通信方式である．Anavanet は GeoNetworking に対応しているが，データをすべて収集した後に，テストデータの処理を行うという性質上，iperf や ping に存在していたリアルタイム性という利点を失っている．

2.2.5　V2X 協調型運転システムにおける通信性能のリアルタイム可視化ツール

Anavanet ではリアルタイム性が失われていたが，V2X 協調型運転システムにおける通信性能のリアルタイム可視化ツール [38] ではリアルタイム性を残しつつ，フィールドの地理的特

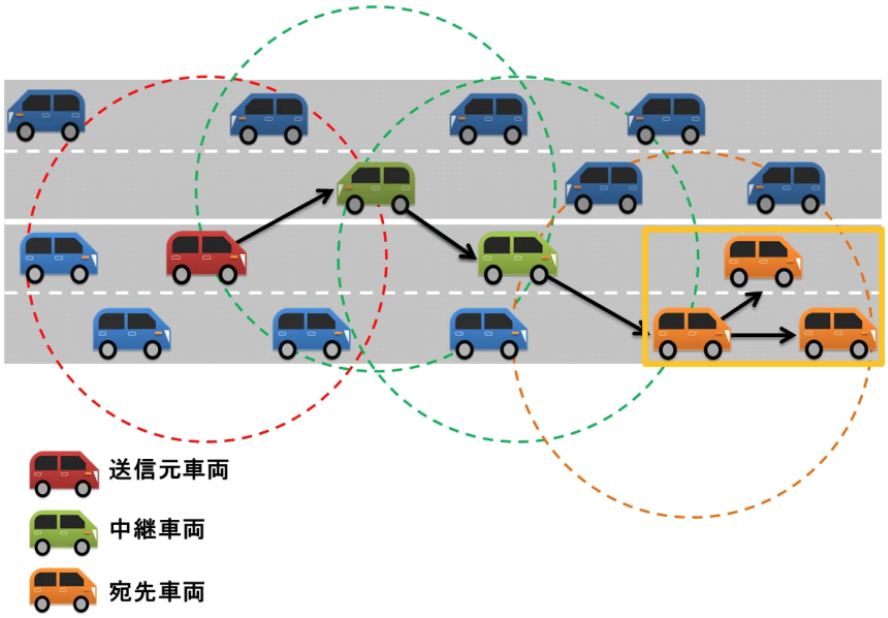

図 2.9. GeoNetworking [37]

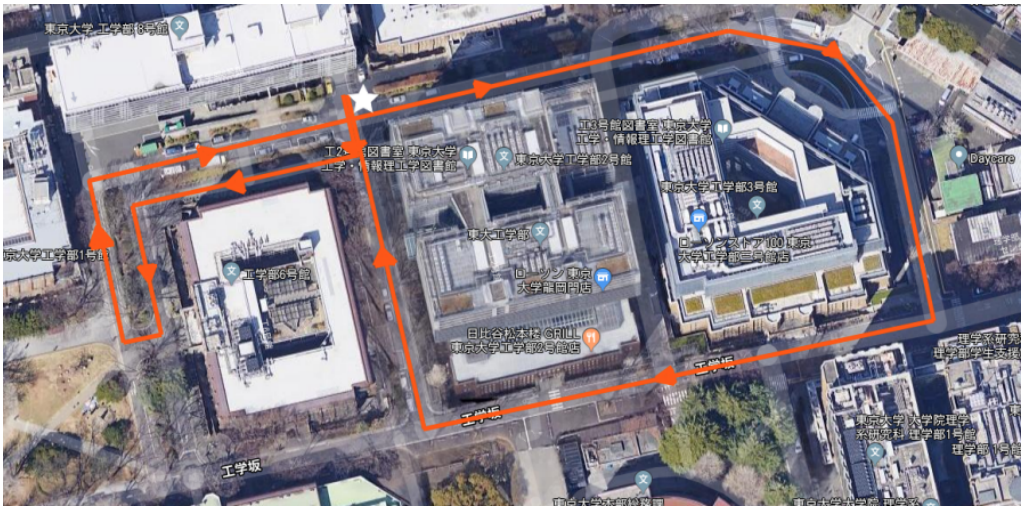

図 2.10. 移動経路 [38]

性を反映した通信性能可視化をすることができる．図 2.10 の白い星の位置に RSU を設置し赤い経路に沿って車両を動かしたときの，RSU から車両への PDR を図 2.11 が示している．青い地点が PDR が高く，赤に近づくにつれて PDR が低くなっている．このツールでは，送信パケット数を動的に変更しているため，パケットロスを算出するためには両者の接続が必要となっており，接続が切れている場所（図 2.11）の右下の位置では，可視化が行えていない．また，1 秒間の通信性能を測定して，各地点にプロットをしているという性質上，プロット間の通信性能がわからないという問題がある．

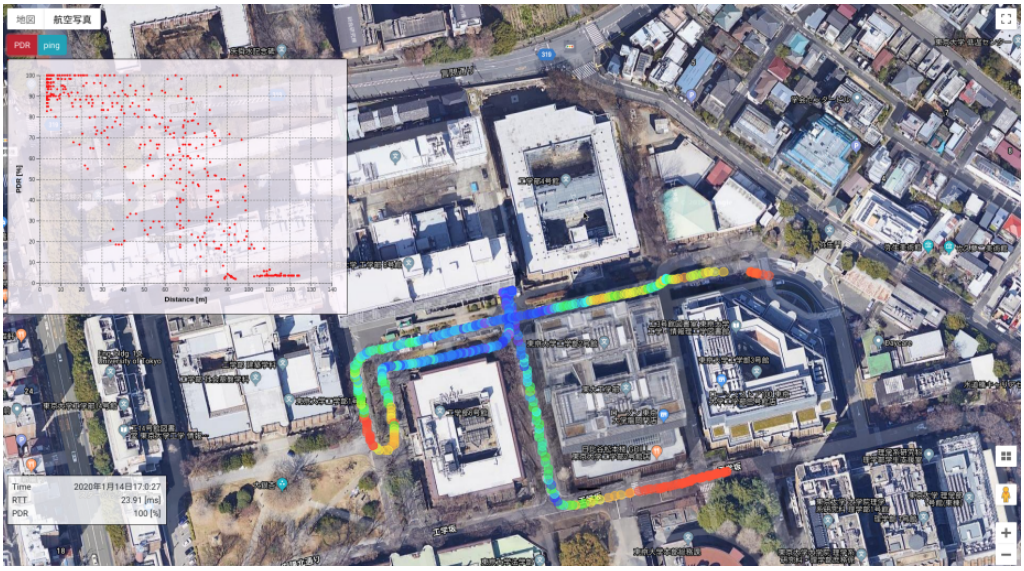

図 2.11. V2X 協調型運転システムにおける通信性能のリアルタイム可視化ツール [38]

2.3　現状の課題

現在の協調型自動運転の課題を延べる．協調型自動運転では，車両が移動するという特性上，通信の混雑，電波干渉や通信先との距離，障害物などによって通信が安定するとは限らない．また，通信が安定しない場所では，交差点の先などの自車両のセンサだけでは死角となり，認識できない場所の情報が適切なタイミングで手に入らないという問題がある．自律型の自動運転では，死角から歩行者や自動車が飛び出してくる可能性のある状況では一時停止するのが一般的となっている．一方で協調型の自動運転では，通信によって死角をなくすことができるが，パケットロスによって検知された物体の情報が届かなかったり，遅延が発生して，物体の位置が大きくずれていた場合に衝突の危険がある．そこで，通信性能が悪いことによって，死角の先の情報が手に入っていないのか，そもそも死角の先に物体が存在しないのかを把握する必要がある．自動運転車の経路上の通信性能を事前に把握することができれば，通信性能に応じて，協調型と自律型の自動運転を切り替えたり，そもそも通信性能が悪い場所を通らない経路計画などが可能になるため，協調型自動運転の安全性が向上する．

2.2 章で説明した ping や iperf，Anavanet，V2X 協調型運転システムにおける通信性能のリアルタイム可視化ツールでは，以下のような課題があるため，これらを考慮した協調型自動運転のための通信性能分析，可視化手法を提案する．

課題 1: 地理的特性サポート

通信性能の分析に一般的に使用されるツールである ping や iperf が挙げられる．しかし，ping や iperf では，フィールドの地理的特性を反映していないという問題点がある．自動運転車は，移動物体であるという特性上，遮蔽物や斜面などの地理的

特性を反映した通信性能分析を行う必要がある．

課題 2: GeoNetworking 解析

課題 1 と同様に ping や iperf は，V2X の通信プロトコルである GeoNetworking に対応していない．協調型自動運転の通信性能分析，可視化のためには GeoNetworking の対応が不可欠である．

課題 3: リアルタイム性

Anavanet では，フィールドの地理的特性の考慮と，GeoNetworking への対応が可能になった．一方で，データをすべて集めた後に，通信性能の分析を行うという性質上，iperf や ping に存在していたリアルタイム性という利点を失っている．自動運転車を走らせながら，通信性能が確認できるように，通信性能の分析を行う手法を提案する必要がある．

課題 4: 様々なシナリオサポート

限られた場所やシナリオだけでなく，どんな場所，どんなシナリオでも動作するシステムが求められる．V2X 協調型運転システムにおける通信性能のリアルタイム可視化ツールでは，RSU と車両の間の接続が切れた場所で，パケットロスの分析ができないという問題があった．

課題 5: 通信性能の共有

関連研究 [36, 38] では，通信性能の分析と可視化までしか行っていない．分析した通信性能を他車両に共有できるようすることで，次に同じ場所を通る車両が自律型と協調型の切り替えや，走行経路の変更を行うことができる．しかし，協調型自動運転車両の通信性能を分析した後，他車両に共有する研究は存在しない．

第 3 章

システムアーキテクチャ

本章では，2.3 節で述べた課題を解決するためのシステムのアーキテクチャを説明する．まず，3.1 節でシステムの要求事項をあげる．次に 3.2 節で提案したシステムアーキテクチャを説明する．

3.1 要求事項および本研究の設計方針

協調型自動運転車の通信性能（NP: Network Performance）を分析，可視化する上での要求事項と要求に対する設計方針を以下に示す．

3.1.1 協調型自動運転のための NP 分析と可視化

ping や iperf では，対応できない，協調型自動運転に特化した NP 分析，可視化システムを作成する．具体的には，遮蔽物や斜面などの地理的特性を反映する．NP を計測する際に，常に車両の位置情報と紐づけて保存する．これによって，どの地点でどの程度通信ができるのかのデータが取得できるため，地理的特性を考慮した NP 分析，可視化が可能となる．

また，協調型自動運転の NP 分析，可視化のためには V2X の通信プロトコルである，GeoNetworking によるマルチホップ対応が不可欠である．マルチホップに対応させるため，通信経路とその経路ごとの NP 分析，可視化を行うことができるように設計をする．具体的には，流れてくるパケットをパケットキャプチャでキャプチャすることによって，それぞれの経路間の NP を計測する．例えば，送信元の車両 A が中継車両 B，C を経由して，宛先車両 D へと GeoNetworking でマルチホップして通信した場合，AB，BC，CD 間の NP を計測することができる．

一般的に，Autoware を用いた自動運転の実験では，実験時に rosbag ファイルを保存しておき，実験後に rosbag ファイルを再生することによって，いつでも実験時の状況を再現することが可能となる．NP 分析した結果も ROS メッセージとして，パブリッシュすることによって，rosbag ファイルに保存することができるようになり，NP に関するデータも実験後に再現できる．また，ROS メッセージとしてパブリッシュすることによって，Autoware の標準可視

化ツールである RViz 上に通信性能を可視化することができる．

3.1.2 汎用性

NP 分析，可視化のシステムを作成するにあたって，汎用的に利用できるように設計する．限られた場所やシナリオだけでなく，どんな場所，どんなシナリオでも動作するシステムを作成する．V2X 協調型運転システムにおける通信性能のリアルタイム可視化ツールでは，RSU と車両間の両者の接続が切れた場合，通信性能の可視化ができなかった．本システムでは，接続が切れた場合や，パケットロスが 100% の場合でも，持続的に分析，可視化可能なものとする．通信性能を分析する際に，接続が切れても，送信側でパケットを流し続けるようにする．また，受信側で，接続が切れていても，パケットが 1 つも届かなかったとみなすようにする．

NP 分析，可視化システムの使用用途として，実験後の NP 分析だけでなく，実験中にも，NP が確認できるように，リアルタイムで動作するシステムとする．リアルタイムで，パケットロスを計算するためには，送信パケット数と受信パケット数を把握することが必要である．しかし，送信パケットが動的に変化する場合，受信側で送信パケット数を把握することが難しい．そこで，パケットロスをリアルタイムで計測するため，送信側で送信する単位時間あたりのパケット数を固定する．

また，NP の分析，可視化は双方向に行うものとする．RSU から車両への NP だけでなく，同時に車両から RSU の NP も分析，共有する．RSU，車両の両方で，NP 分析用の ROS ノードを作成する．

3.1.3 クラウドでの NP マップの管理，共有

分析した NP を他車両に共有することによって，次に同じ場所を通る車両が NP を事前に参照できるようにする．自動運転車は走行中に，NP を分析しながら，ダイナミックマップと付随して，その位置でどの程度通信できるのかをクラウドへアップロードする．クラウドで，NP の情報を集約，管理することで，NP マップ（位置情報と合わせて，NP の情報を集約した地図データ）を生成する．これによって，作成した NP マップを自動運転車はいつでもクラウドから参照することができる．NP マップの生成と共有に関する部分が本研究の新規性である．

3.2 システムアーキテクチャ

要求事項を踏まえて，提案するシステムアーキテクチャを図 3.1 に示す．重要な要素として，RSU と車両間の NP をクラウドに収集し，NP マップを生成する．これによって，次に同じ場所を通る車両は，過去の NP を参照し，自律型と協調型の切り替えや，経路計画を行うことができるようになる．

RSU から車両への NP 分析とクラウドへのアップロードまでの動作が図 3.1 の左に示されている．具体的な手順を以下に示す．

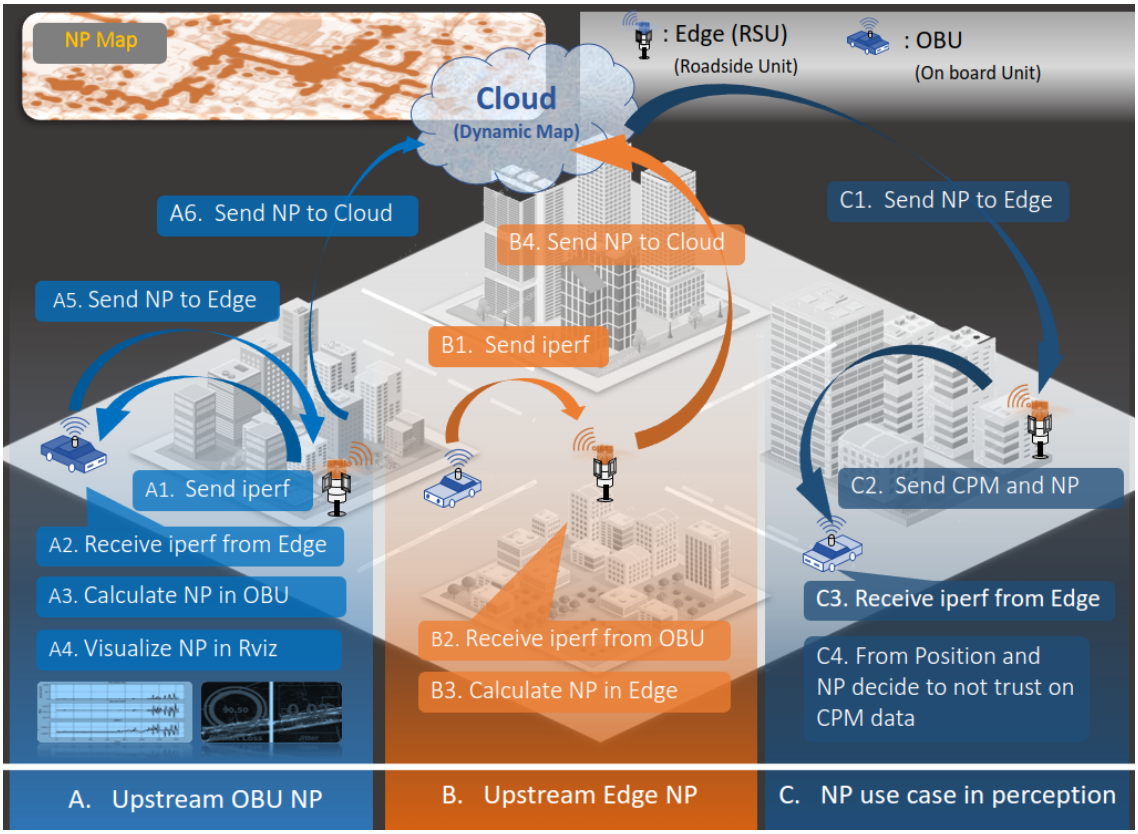

図 3.1. 提案するシステムアーキテクチャ

A1. RSU から車両に向けてパケットを送信
　iperf を使用して，単位時間あたり固定個のパケットを送信する．

A2. RSU からのパケットを車両で受信
　iperf の出力結果をそのまま使用するのではなく，パケットキャプチャを使用して，iperf のパケットをキャプチャすることによって GeoNetworking のマルチホップに対応する．

A3. NP 分析
　キャプチャしたパケットから，単位時間あたりの RSU から車両への NP を計算する．計算する NP は，パケットロス，帯域幅，Jitter とし，車両の位置情報と合わせて保存する．これによって地理的特性を考慮した NP 分析，可視化が可能となる．

A4. NP 可視化
　分析した NP を Autoware の標準可視化ツールである RViz 上にリアルタイムに可視化する．

A5. NP を RSU に送信
　計測した NP を RSU に送信し，RSU で受信した情報を近くの全車両にマルチキャストによって送信することによって各車両の NP を近くの全車両に共有する．

A6. NP をクラウドに送信

NP を RSU からクラウドにアップロードすることによって，より広域の情報を必要とする車両から参照可能となる．また，位置情報と合わせた NP の情報をクラウドで収集することによって，どの場所でどの程度 RSU と車両が通信できるかを示す NP マップを作成できる．

RSU 側で，車両から RSU への通信性能把握とクラウドへのアップロードまでの動作が図 3.1 の中央に示されている．具体的な手順を以下に示す．

B1. 車両から RSU に向けてパケットを送信
iperf を使用して，単位時間あたり固定個のパケットを送信する．で

B2. 車両からのパケットを RSU で受信
iperf の出力結果をそのまま使用するのではなく，パケットキャプチャを使用して，iperf のパケットをキャプチャすることによって GeoNetworking のマルチホップに対応する．

B3. 通信性能計測
キャプチャしたパケットから，単位時間あたりの RSU から車両への NP を計算する．計算する NP は，パケットロス，帯域幅，Jitter とする．

B4. NP を Cloud に送信
NP を RSU からクラウドにアップロードすることによって，より広域の情報を必要とする車両から参照可能となる．

クラウドに保存した NP を車両に共有する動作が図 3.1 の右に示されている．

C1. NP マップを Cloud から RSU に送信
RSU 付近の車両が必要とする場所の NP マップを Cloud から RSU に送信する．

C2. RSU から CPM と NP を車両へ送信
RSU に設置されたセンサから得た CPM と NP マップを車両に送信する．

C3. 車両で CPM と NP を受信
車両付近の NP マップと RSU が取得した CPM を受信する．ここで車両の現在位置の NP を把握するために付近の NP マップのデータから補間を行う．本研究では，線形補間で補間を行う．

C4. 認知方法の決定
現在位置の NP に応じて，CPM のデータが信頼できるかどうかを決定する．例えば，パケットロスが多い場合は CPM の情報がすべて受け取れていない可能性が高いので自律型の自動運転に切り替える．逆に，パケットロスが少ない場合は，CPM の情報を信頼することができる．

第 4 章

実装

本章では，提案したシステムアーキテクチャをどのようにして実装するかを説明する．まず，4.1 節で DM2.0PF との統合について説明する．次に，4.2 節でネットワーク分析の手法を説明する．最後に，4.3 節でネットワーク可視化の手法を説明する．

4.1 DM2.0PF との統合

提案したシステムでは，車両で計測した RSU から車両への NP を RSU やクラウドへ送信し，情報を共有する．また，他車両で計測した NP をクラウド，RSU を通して，車両で受信する設計となっている．そこで，本実験では，車両と RSU での情報共有，クラウドでのデータ収集のために DM2.0PF を使用する．得られた情報（本研究では NP）を車両の DM2.0PF に保存することで，付近の RSU，クラウドへと通信され，情報共有される．また，必要に応じて，車両側で RSU，クラウドの持つデータを参照することができる．

提案したシステムは 3.1 節で説明した要求事項を満たすため，ROS ベースで構築する．ROS では，出力のフォーマットとして ROS メッセージが一般的に使用されているが，DM2.0PF で扱うデータはストリームデータとなっている．そのため，提案したシステムと DM2.0PF を統合するためには，ROS メッセージをストリームデータに，あるいはストリームデータを ROS メッセージに変換する機能が必要になる．

実際に実装したシステムの構成図を図 4.1 に示す．灰色で示された部分が，既存のシステムを利用した部分で，色がついている部分が提案システムである．システムの動作中は車両と RSU の間で常時 iperf でパケットを流し続ける．車両，RSU の両方で NP Analyzer によって NP を計測し，ストリームデータに変換することで，DM2.0PF のデータベースにアップロードする．アップロードされた情報は，クラウド，エッジ，車両に DM2.0PF の通信機能によって共有される．車両側で他車両や RSU から，DM2.0PF を介して共有されたデータを参照する場合は，NP Extractor によって，DM2.0PF 内を流れるストリームデータを ROS メッセージに変更することで車両内の Autoware で扱うことができるようになる．車両では，自動運転用の処理端末と通信用の端末に分けている．自動運転用の処理端末では，GPU を搭載し，Autoware の処理や NP の可視化を行う．通信用の端末では，NP の計測や，DM2.0PF によ

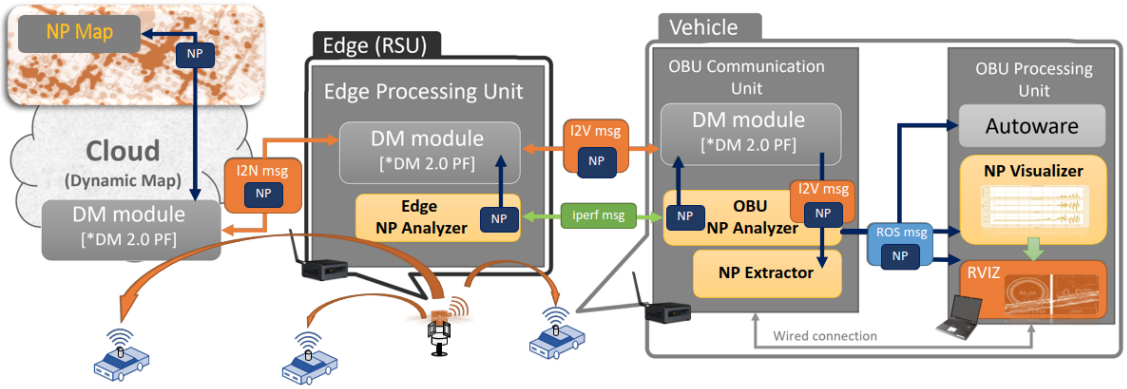

図 4.1. 提案システムと DM2.0PF の統合

る RSU，Cloud との通信を行う．2 台の端末は有線接続することによって ROS メッセージが共有されている．実験時は 2 台とも車載器として扱う．

より具体的な ROS ノードとメッセージの流れを図 4.2 に示す．オレンジ色で示したものが本研究で実装した ROS ノードである．車両の NP Analyzer では，双方向に iperf を送受信し，受信側で NP の計測，分析を行う．分析した NP をパブリッシュするために，NP をやり取りするための独自の ROS メッセージ NetworkPerformanceMsg を表 4.1 のように定義する．定義した，NetworkPerformanceMsg を利用して，トピック/network_performance/local に NP の情報と車両の位置情報を 1 秒に 1 度パブリッシュする．車両の現在位置は Autoware がパブリッシュしている ROS トピック/tf から取得する．/tf は，Autoware で Sensing して得たデータから自己位置推定を行った結果のデータであり，タイムスタンプ，車両の座標，向きなどの情報が含まれている．パブリッシュした/network_performance/local は RViz でサブスクライブされて，可視化を行う．また，ROS ノードである NP Converter で，サブスクライブし，ROS メッセージからストリームデータに変換して，DM2.0PF にアップロードする．このとき，DM2.0PF で，NP が扱えるように表 4.1 と同様にストリームデータの定義を行う．DM2.0PF にアップロードしたデータは，RSU，クラウドへと共有され，NP マップが生成される．同様に，クラウドや RSU が持つデータが車両に共有され，共有されたデータは NP Extractor で，ストリームデータから ROS メッセージに変換して，トピック/network_performance/dm にパブリッシュする．パブリッシュした/network_performance/dm を NP Visualizer でサブスクライブし，グラフで可視化する．

4.2　ネットワーク分析

ネットワーク分析するプログラム（NP Analyzer）は 3.1 節で述べた要求事項を満たすため ROS ノードで作成する．まず，wireshark [39] を使用して，iperf パケットのキャプチャを行う．iperf では，パケットサイズ 1.6Kbit のパケットを，1 秒間に 625 個送信しており，合計 1Mbit の帯域幅となる．wireshark で 1 秒間に受信したパケット数と送信パケット数を比較し

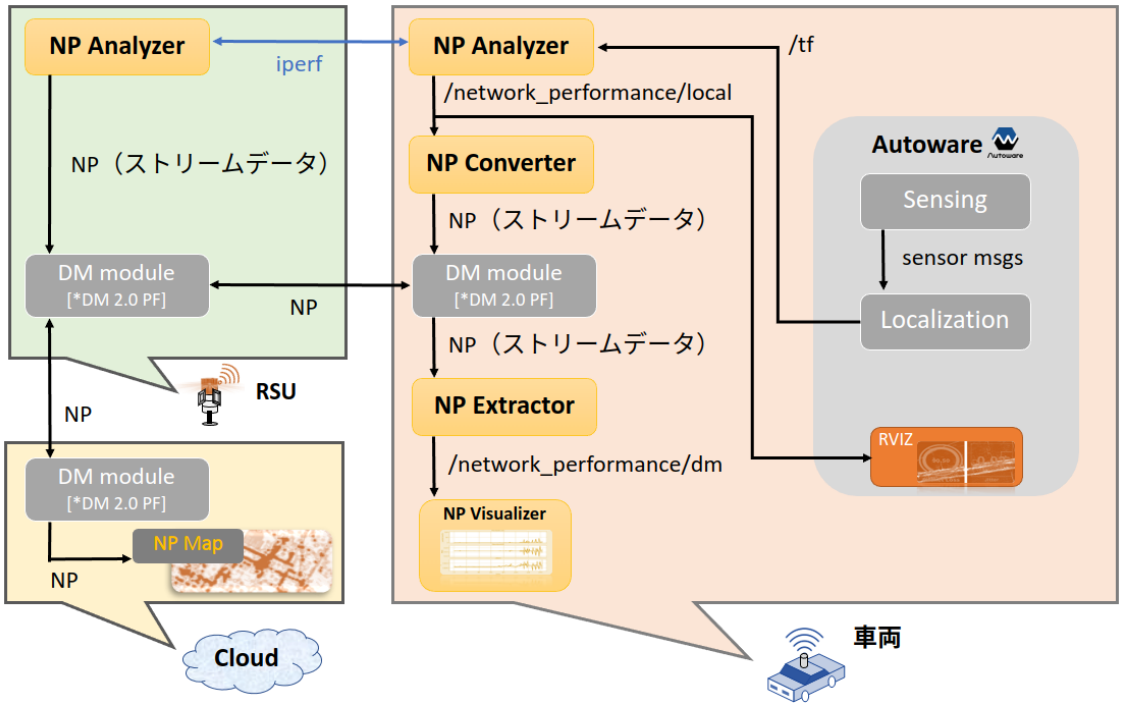

図 4.2. 提案システムの ROS ノードとメッセージの流れ

表 4.1. NetworkPerformanceMsg の定義

データ型	変数名	用途
string	id	MAC アドレス
string	src	パケットの送信 IP
string	dst	パケットの受信 IP
float32	x	位置座標 x
float32	y	位置座標 y
float32	z	位置座標 z
float32	packetloss	パケットロス
float32	bandwidth	帯域幅
float32	jitter	ジッター

て，1 秒間のパケットロスを算出する．また，受信したパケットのサイズとパケットの数から 1 秒間の帯域幅を算出する．Jitter は RFC3550 [40] で定義された以下の式 4.1，4.2 から算出する．

$$D(i,j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i) \tag{4.1}$$

$$J(i) = J(i-1) + (|D(i-1,1)| - J(i-1))/16 \tag{4.2}$$



図 4.3. PieChart [43]



図 4.4. Plotter2D [44]

図 4.5. JSK ROS Package

S はパケットのタイムスタンプ，R はパケットの到着時刻，D は送信側と受信側のパケット間隔の差，J は Jitter を表している．2 つのパケット i と j が届いたときに，パケット i と j の送信側と受信側のパケット間隔の差は $D(i,j)$ で表され，D を用いて Jitter を求めることができる．

4.3　ネットワーク可視化

本節ではネットワークの可視化手法を説明する．

4.3.1　JSK ROS Package

RViz 上に NP を可視化する方法として，東大 JSK が開発した JSK ROS Package [41] を使用する．JSK ROS Package は ROS 用の可視化ライブラリで，RViz のプラグインや rqt のプラグインが公開されている．RViz のプラグイン [42] の中でも PieChart と Plotter2D を使用して可視化する．PieChart と Plotter2D が実際に動作しているときの図を図 4.5 に示す．PieChart は円グラフとなっており，Plotter2D は曲線グラフとなっている．また，グラフだけでなく，数値に応じて色を変化させることも可能である．今回はパケットロスを PieChart で表示し，Jitter を Plotter2D で表示した．また，NP が良いときは青色で表示し，悪いときは赤色で表示した．色が変わるしきい値は，用途に応じて変更しやすいように，RViz から変更可能である．元々，JSK ROS Package は ROS1 で動作するものであったため，Autoware のバージョンに合わせて，ROS2 で動作するように変更した．NP Analyzer によってパブリッシュされた ROS トピック/network_performance/local をサブスクライブして NP 可視化を行う．

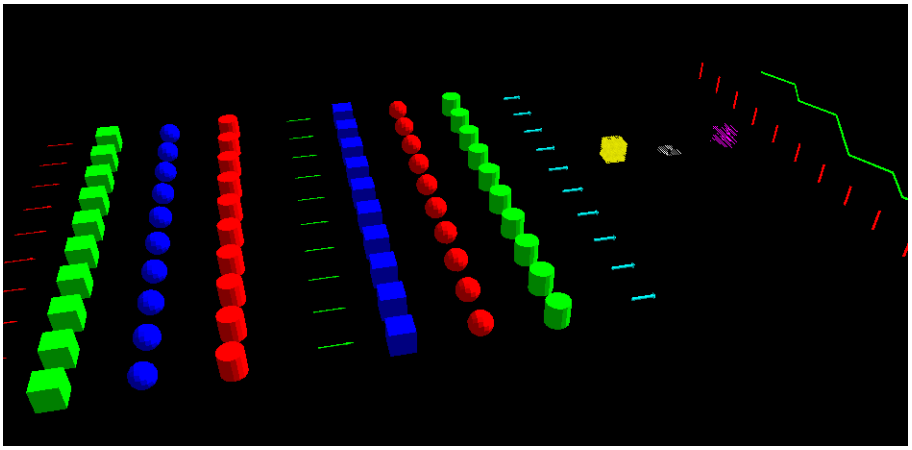

図 4.6. marker 可視化例 [46]

4.3.2　matplotlib

RSU から自車両, 自車両から RSU, 他車両と RSU 間の NP を matplotlib [45] を使用してグラフで可視化する．matplotlib を使用した可視化も ROS ノード（NP Visualyzer）として実装し，NP Extractor によってパブリッシュされた ROS トピック（/network_performance/dm）をサブスクライブして，可視化する．

4.3.3　地図上に NP 可視化

2.2.5 節で説明した V2X 協調型運転システムにおける通信性能のリアルタイム可視化ツールの図 2.11 のように，NP を色分けしながら，点をプロットするようにして可視化する．パケットロスが良い状態だと青で表示し，悪くなるに連れて，緑，黄色，オレンジ，赤と変化させる．RViz 上に可視化するため，ROS ノードとして実装し，プロットには RViz 上にオブジェクトを描画するライブラリである marker [46] を使用する．marker を使用したときのオブジェクト描画の例を図 4.6 に示す．

4.3.4　データ補間

図 2.11 のように，単位時間あたりの通信性能を分析してその地点を通信性能に応じてプロットする．このとき，後から，通信性能を参照しようとすると，プロットとプロットの間の通信性能がわからないという問題がある．そこで，取得した NP のデータを補間することによって，プロット間の NP を参照できるようにする．補間の手法として，線形補間を実装する．線形補間とは，2 つのデータの間の値を一次関数で近似して求める手法で計算量が少ない点やオーバーシュート，アンダーシュートが生じない．一方で，接続がなめらかではないため，サンプル点以外の精度が落ちることが問題である．これを解消するには，サンプル点をよ

り細かくする必要がある．今回は計算量が少ない点から，コンピュータリソースを圧迫しないため，線形補間を使用する．

第 5 章

実験・評価

本章では，実験とその結果について述べる．NP 分析と可視化実験を実際に車両を動かしながら行った．5.1 節では行った実験の概要を述べる．5.2 節では，インドアで移動を制限した環境下で動作確認テストを行った結果と評価を述べる．5.3 節では，専用区画の実証フィールド（東京大学柏の葉キャンパステストコース），住宅街の公道（愛知県春日井市石尾台地区）でアウトドアテストを行った結果と評価を述べる．5.4 節で，本章のまとめを述べる．

5.1　実験概要

本研究では，NP 分析と可視化の評価のため，RSU と車両間の通信実験を行う．通信方式は，IEEE802.11g[47] の AP モードで，RSU を AP とした．また，2.4GHz のチャンネル 1（2401MHz-2423MHz）を使用した．実際に，使用した機材のスペック，ソフトウェアのバージョンを表 5.1 に示す．車載器として，自動運転用の GPUPC と，NP 分析のための NUC の 2 台を使用し，RSU では，NP 分析のための NUC を 1 台のみ使用する．

まず，インドアで，NP 分析，可視化が正常に動作するかの確認を行う．具体的には，tc コマンドを使用して擬似的にパケットロスを発生させたときの，NP 分析，可視化の数値が，擬似的に発生させたパケットロスと一致するかを確認する．

次にアウトドアで，実際に RSU を設置して，車載器を載せた車両を動かすことによって，位置情報を考慮した可視化を行う．東京大学柏の葉キャンパステストコースで，取得した位置情報とその場所の NP を補間し，可視化する．取得した NP のデータから，補間精度，動的物体による NP 変動の 2 点を評価する．

愛知県春日井市石尾台地区にて動作を確認する．取得した位置情報とその場所の NP を補間し，NP マップの有用性を評価する．柏の葉キャンパステストコースでは，動的物体の影響がほとんどないが，石尾台地区は，住宅地であり，他の車両や歩行者等が存在するため，動的物体を考慮した実環境の結果を得ることができる．

表 5.1. 使用した機材のスペックとソフトウェアのバージョン

項目	車載器		RSU
PC	G-Tune H5	Intel NUC11PAHi5	Intel NUC11PAHi5
CPU	Core i7-10875H 2.30GHz 16core	Core i5-1135G7 4.20 GHz 4core	Core i5-1135G7 4.20 GHz 4core
GPU	NVIDIA GeForce RTX 2070 SUPER	-	-
RAM	64GB	64GB	64GB
アンテナ	-	TP-Link Archer T3U Plus	TP-Link Archer T3U Plus
OS	Ubuntu20.04	Ubuntu18.04	Ubuntu18.04
NVIDIA Driver	510.108.03	-	-
CUDA	11.6	-	-
Autoware	Autoware.universe 最新バージョン (2023 年 1 月 13 日時点)	-	-
ROS	galactic	dashing	dashing
wireshark	-	2.6.10	2.6.10
iperf	-	2.0.20	2.0.10

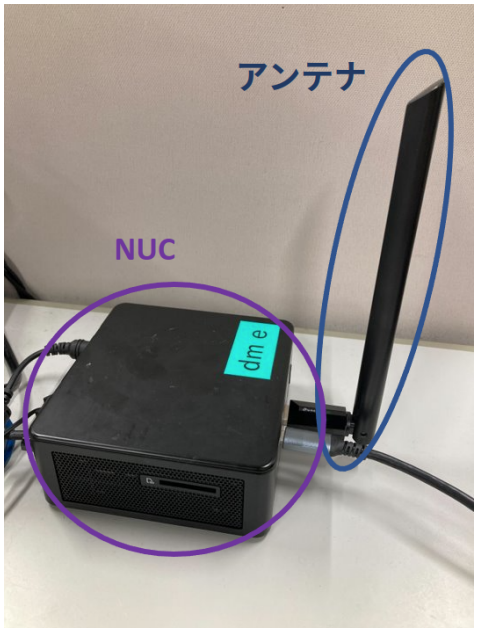

図 5.1. インドアテスト RSU

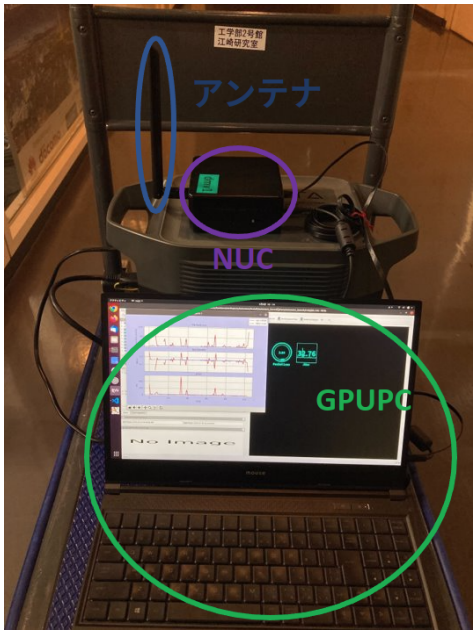

図 5.2. インドアテスト車載器

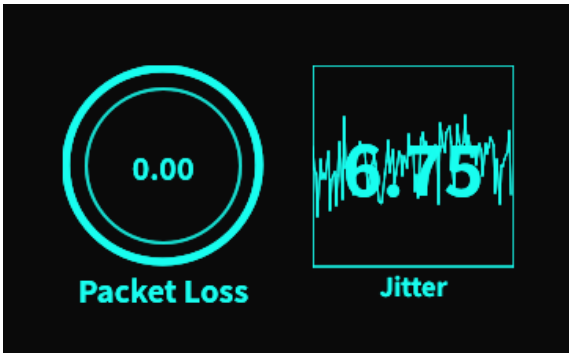

図 5.3. JSK ROS Package 可視化（NP が良いとき）

5.2　インドアテスト

室内で，作成した NP 分析と可視化のプログラムの動作確認を行った．RSU を図 5.1 のように仮想的に設置し，車載器を図 5.2 のように台車に載せて動かしながら，動作を確認する．

まず，NP 分析の動作を確認する．NP が非常に良い状況下（RSU と車載器を距離 1m 程度で，障害物等もない状態）で NP を出力すると，パケットロスは 0% から 0.16% の間で推移しており，Jitter は，約 6.3ms で安定していた．動作確認のため，tc コマンドを使用してパケットロスを 50% 発生させると，システムが分析したパケットロスは 48.4% から 52.1% の間で推移しており，Jitter は 9ms から 13ms の間で変動した．また，パケットロスを 90% にすると，システムが分析したパケットロスは，88.4% から 91.4% の間で推移しており，Jitter は 27ms から 43ms まで増加した．以上のことから，実装したソフトウェアでは，±2% 程度の精度で，パケットロスを分析することができた．

次に，JSK ROS Package を使用した可視化の動作を確認する．NP が非常に良い状況下で JSK ROS Package を使用して可視化した様子を図 5.3 に示す．同様に動作確認のため，tc コマンドを使用して疑似的にパケットロスを発生させる．パケットロスを 50% にしたときに NP を可視化した様子を図 5.4 に，パケットロスを 90% にしたときに NP を可視化した様子を図 5.5 に示す．tc コマンドで発生させたパケットロスとほとんど同様の数値を示しており，数値に応じて色わけができていることを確認した．

次に，matplotlib を使用した可視化の動作をを確認する．NP が非常に良い状態下で，NP を可視化した様子を図 5.6 に示す．同様に tc コマンドを使用して，パケットロスを擬似的に発生させる．RSU 側で tc コマンドを使用してパケットロスを 50% にしたときに NP を可視化した様子を図 5.7 に示す．また，車両側で tc コマンドを使用してパケットロスを 90% にしたときに NP を可視化した様子を図 5.8 に示す．tc コマンドで発生させたパケットロスと同様の数値でグラフが推移しており，車両側，RSU 側の両方で動作を確認した．

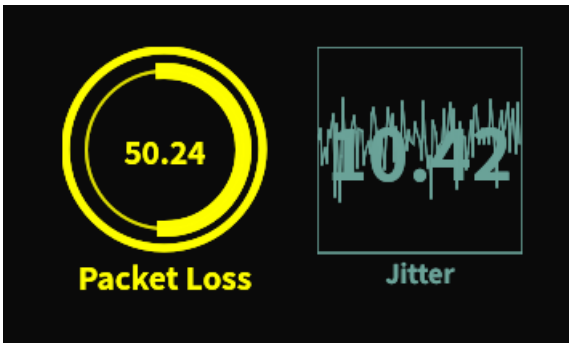

図 5.4. JSK ROS Package 可視化（パケットロス 50%）

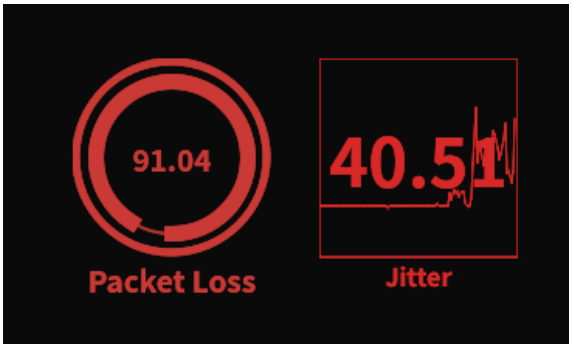

図 5.5. JSK ROS Package 可視化（パケットロス 90%）

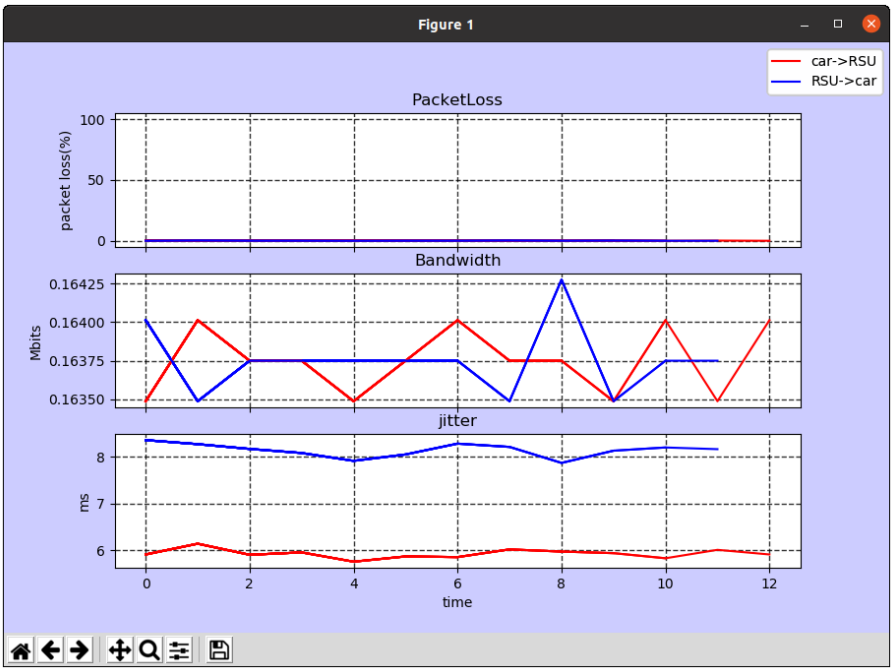

図 5.6. matplotlib 可視化（NP が良いとき）

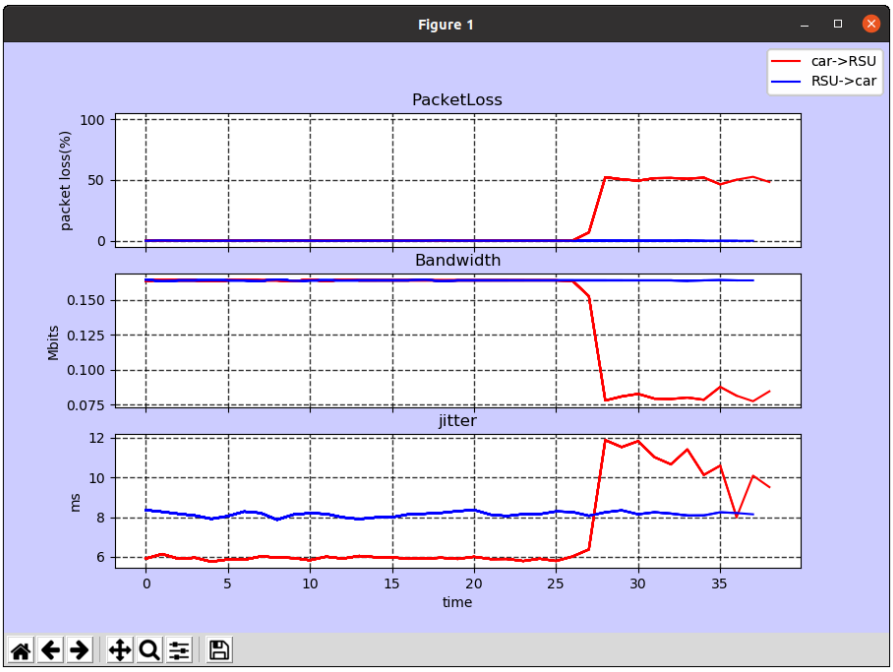

図 5.7. matplotlib 可視化（パケットロス 50%）

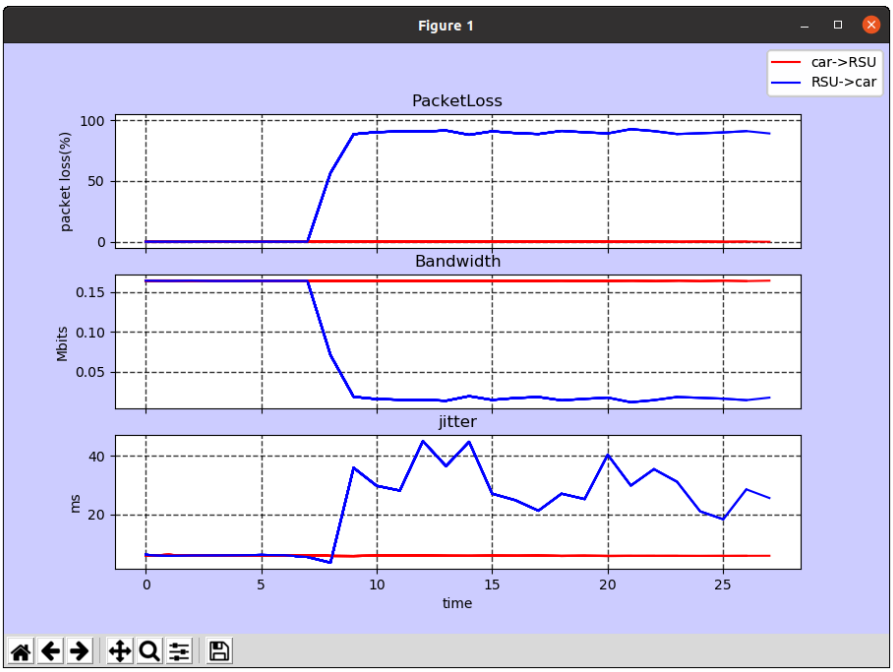

図 5.8. matplotlib 可視化（パケットロス 90%）



図 5.9. 柏の葉キャンパステストコース俯瞰図

5.3　アウトドアテスト

インドアテストで，NP 分析と可視化の動作確認を行った後，アウトドアで実際に車両を使用して実験を行った．まず，5.3.1 節に専用区画の実証フィールド（柏の葉キャンパステストコース）で実験した結果と評価を述べる．柏の葉キャンパステストコースは，大学内にある実証フィールドであり，他の車両が通らないため，動的物体の影響を受けない理想的な環境下での実験となる．次に，5.3.3 節で住宅街の公道（愛知県春日井市石尾台地区）で実験した結果と評価を述べる．愛知県春日井市石尾台地区は，住宅地となっており，他の車両や歩行者，自転車等も動的物体が存在する．また，住民が使用している電波も多く存在している実環境での実験となる．

5.3.1　テストコース実験

柏の葉キャンパステストコースで NP 分析，可視化実験を行った．テストコースの俯瞰図を図 5.9 に示す．RSU は図 5.9 の A の位置に図 5.10 のように，NUC とアンテナを設置する．このときアンテナの地面からの高さは約 155cm である．また，図 5.11 のように車に LiDAR（VLP-16），アンテナ，NUC，GPUPC（G-Tune H5）を取り付けて，車両を動かしながら，RSU と車両間のパケットロスを可視化を行ったときの結果を図 5.12 に示す．図 5.12 はパケットロスの値に応じて，パケットロスが低い場合，青で，高くなるにつれて緑，黄色，オレンジ，赤と色分けしながら一秒に一度，現在位置にプロットしたものである．

次に RSU の位置を図 5.9 の B の位置に図 5.13 のように NUC とアンテナを設置する．このときのアンテナの高さは，約 2m である．車両を動かして，パケットロスを可視化したときの結果を図 5.14 に示す．図 5.12，5.14 から，RSU と車両の距離が近いほど，パケットロスが



図 5.10. 柏の葉キャンパス RSU（A）

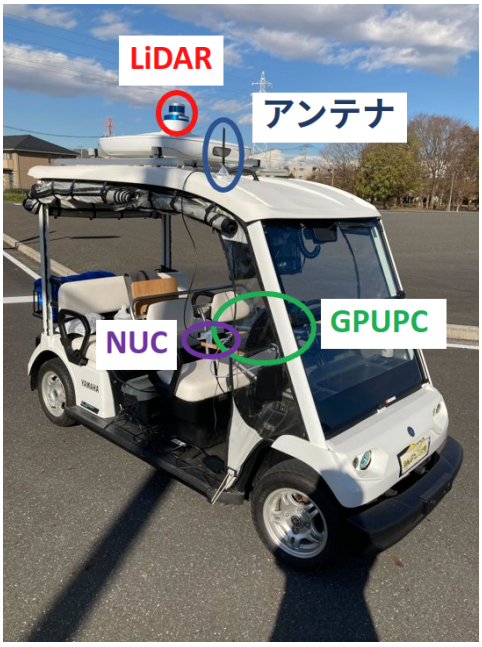

図 5.11. 柏の葉キャンパス車両

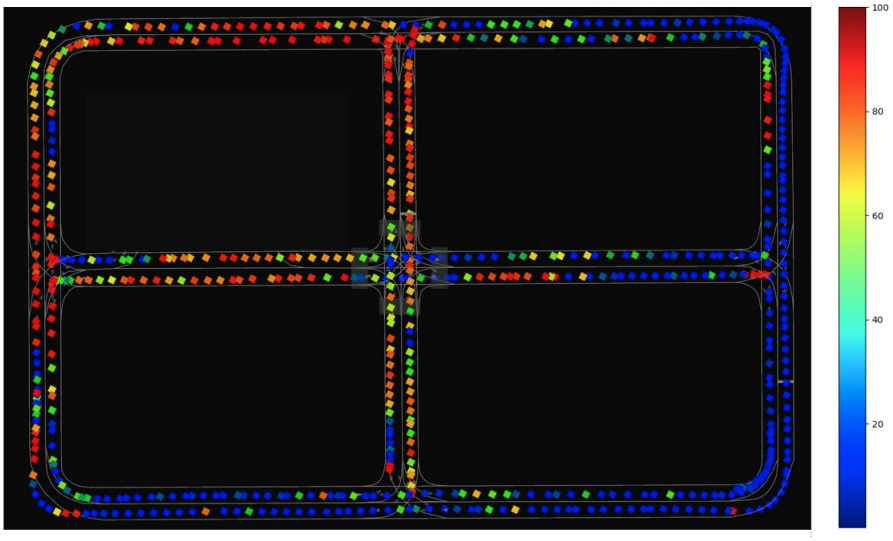

図 5.12. 柏の葉キャンパステストコース可視化結果 A

小さい傾向があることがわかる．一方で，距離が遠い位置でも，距離の近い位置よりもパケットロスが小さくなっている地点もある．原因としては，電波干渉が考えられるため，周波数帯域を 2.4GHz から 5GHz に変更し，実験を実施した結果図 5.15 になった．このときの RSU の位置は図 5.9 の A の位置とした．2.4GHz から 5GHz に変更するとパケットロスがほとんどなくなっており，2.4GHz では電波干渉が起きていたと考えられる．

実際に VIAVI のスペクトラムアナライザ [48] を使用して，テストコースを 1 周して他の電波の計測を行った．周波数ごとの電波強度は図 5.16 となっており，使用していた帯域



図 5.13. 柏の葉キャンパス RSU（B）

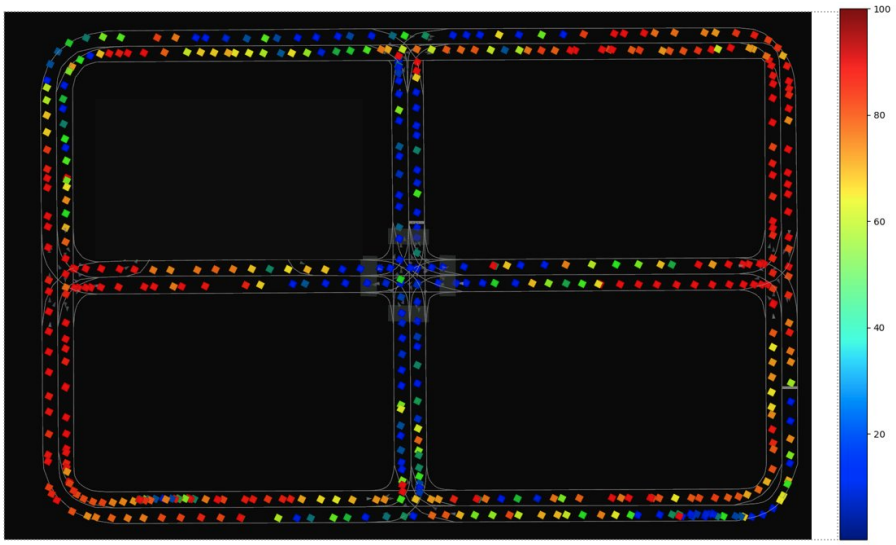

図 5.14. 柏の葉キャンパステストコース可視化結果 B

（2400MHz から 2423MHz の間）で全体的に-70dBm 程度の電波が存在していたことがわかる．また，テストコースを 1 周したときの電波の分布は図 5.17 となっており，テストコースでは，周波数帯や場所に応じて，飛んでいる電波の電波強度が異なっていたが，多くの場合-69dBm から-83dBm 程度の強さであった．また，最大で-64dBm，最小で-86dBm の電波強度が計測されており，これらが実験時の電波を阻害していたことがわかる．飛んでいた電波は，東京大学柏の葉キャンパス内にあるテストコースであるため，学内ネットワークである可能性や，付近が住宅地であるため，住宅地から来ていた電波である可能性が考えられる．

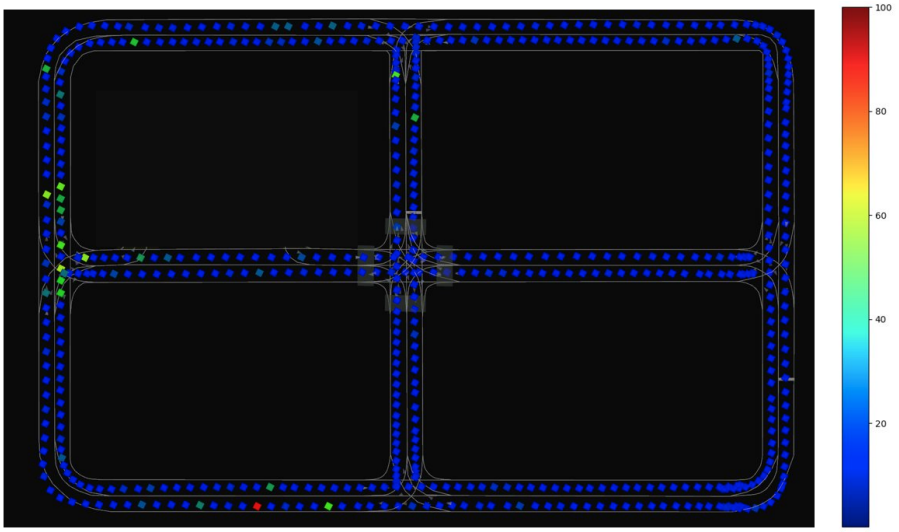

図 5.15. 柏の葉キャンパステストコース可視化結果（5GHz）

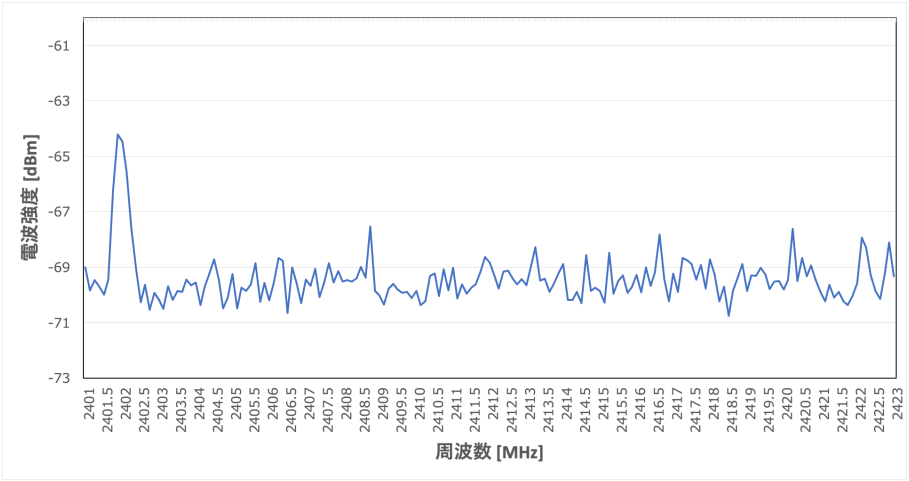

図 5.16. 周波数ごとの電波強度

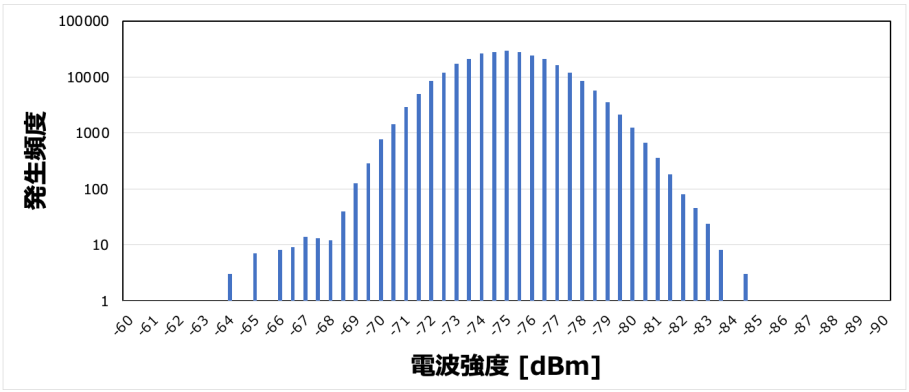

図 5.17. 電波強度分布

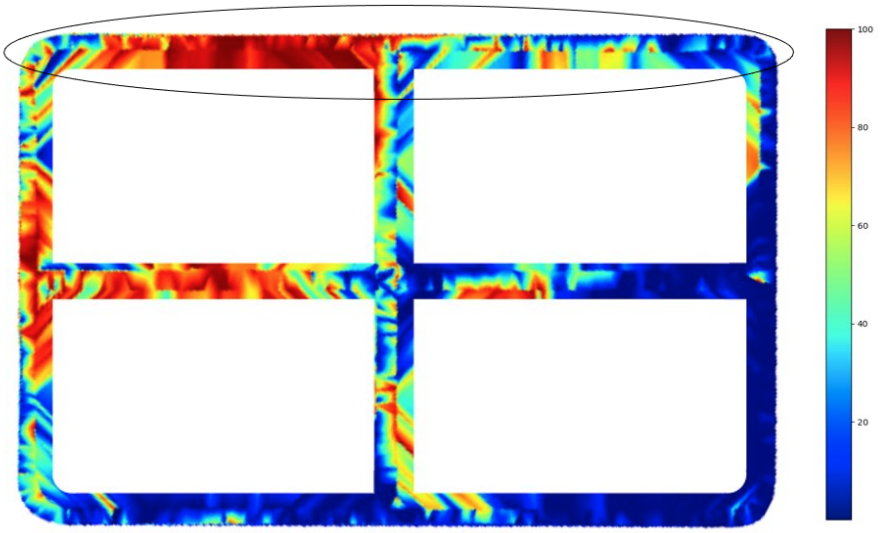

図 5.18. 柏の葉キャンパステストコース補間結果

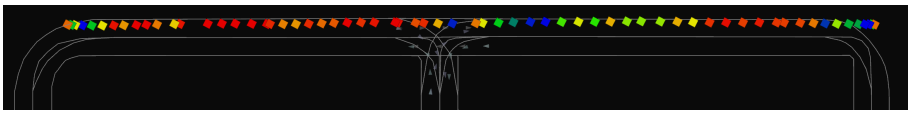

図 5.19. 補間精度計測用データ

次に，柏の葉キャンパステストコースで取得したデータを使用して，補間精度，NP マップの動的物体による影響を評価する．

補間精度評価

補間精度評価では，車両が通過した場所を補間した場合の補間精度評価と，付近を走行したデータから通過していない場所の補間精度評価の 2 通りを評価する．まず，車両が通過した場所を補間した場合の補間精度評価をする．図 5.12 のデータを使用し，テストコース全体のパケットロスを線形補間すると図 5.18 のようになった．本図の中で，パケットロスの変動の大きい，一番上の横向きの車線（黒い楕円で囲われている地点）を使用して，補間精度の評価を行う．黒い楕円で囲われている範囲を左から右へ走行したときのデータを図 5.19 に示す．このデータを補間用データと，精度評価用データに分けて，補間用データのみを使用して補間を行った後，精度評価用データを使用して補間精度の評価を行う．また，補間用のデータの間隔を切り替えて補間を行うことで，NP 補間のために必要なデータ間隔を明らかにする．データ間隔を 4m,8m,16m として，線形補間を行った結果を図 5.20 に示す． 本図は，精度評価用データ（実データ）を黒色でプロットし，それぞれデータ間隔 4m，8m，12m のデータに線形補間を適用し，実データと線形補間で得られたデータを比較したものである．また，実データと線形補間で得られたデータの差

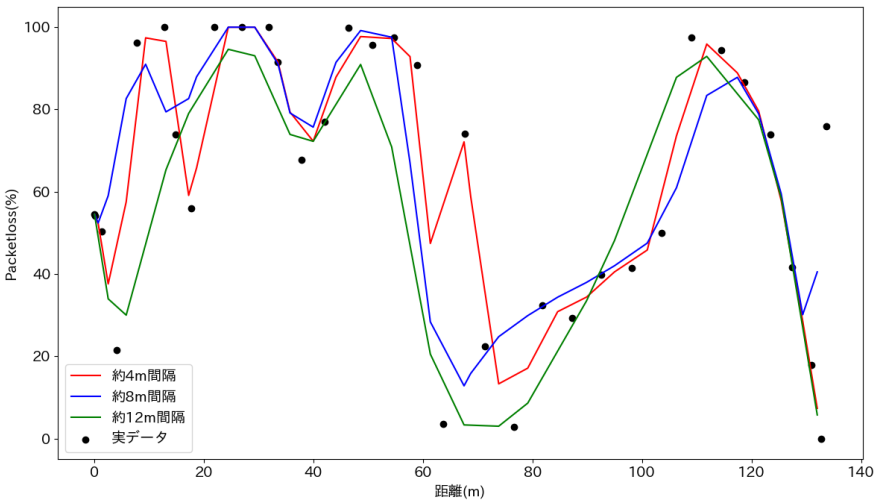

図 5.20. 補間間隔ごとの線形補間結果

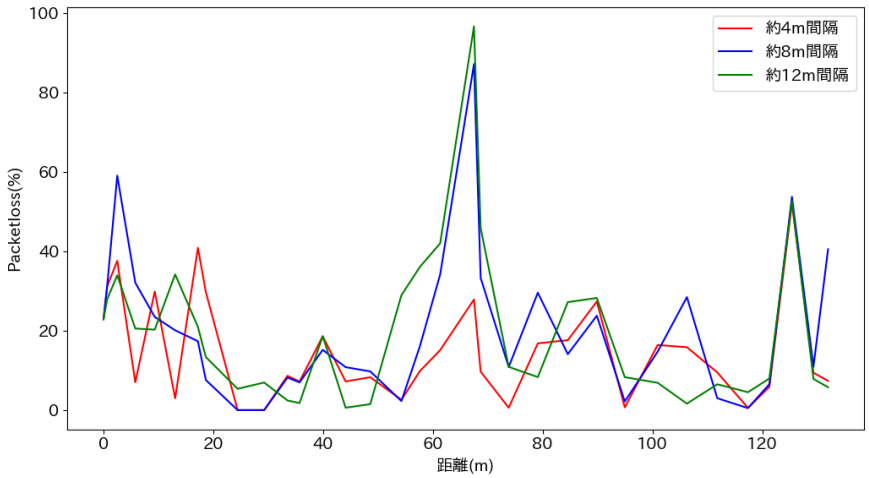

図 5.21. 補間間隔ごとの線形補間精度

をグラフにしたものを図 5.21 に示す．4m 間隔のデータでは，実データと線形補間で得たデータの差が最も小さく，平均 15% 程度であった．多くの場合 30% 以下で収まっているが，最悪のケースでは 52% となる瞬間もあった．8m 間隔のデータでは，平均 20.5% で平均では，4m 間隔のデータと大きな差はないが，最悪のケースが 87.1% であり，大きく補間精度が落ちる瞬間があることが確認できた．また，12m 間隔も同様に平均では，19.9% で，4m，8m の場合とほとんど変わらないが，最悪のケースが 96.7% となり，補間精度が悪くなる瞬間があった．また，補間精度が悪くなる瞬間を確認すると，図 5.21 から，65m 付近で補間精度が落ちていることがわかる．この瞬間の実データを図 5.20 から確認すると，パケットロスが大きく変動している瞬間であることがわかる．以上のことにより，線形補間では，平均

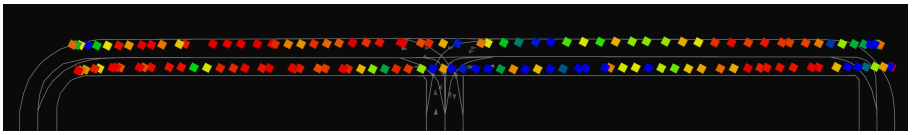

図 5.22. 補間前データ

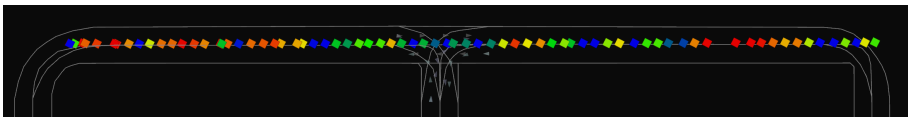

図 5.23. 補間精度比較用データ

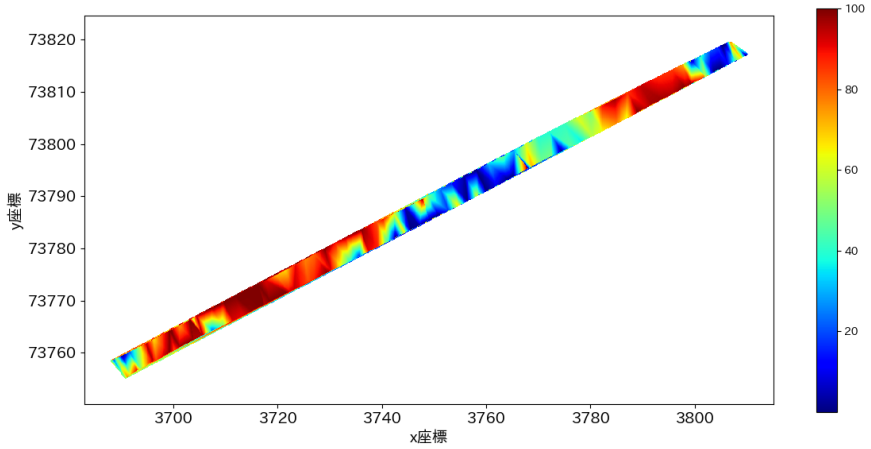

図 5.24. 線形補間結果

すると悪くない補間精度であったが，パケットロスが大きく変動する瞬間に補間精度が落ちる．また，これを防ぐためには，データの間隔を小さくする必要があることがわかる．

次に，付近を走行したデータから通過していない場所の補間精度評価をする．図 5.22 のデータを使用して，2 つの走行経路の間の中央線の NP を補間する．補間精度評価のために図 5.23 のデータを使用する．

図 5.22 を使用して線形補間した結果を 5.24 に示す．また，図 5.23 のデータと比較した結果を図 5.25 に示す． 図 5.25 を見ると，線形補間した結果と実データでは，似たような推移をしている点もあるが，補間結果が実データと大きく異なっている点もあることがわかる．例えば，x 座標が 3700 のときのように急激なパケットロスの変動が起きると，補間結果と大きく異なっている．また，x 座標が 3720 から 3740 の間では補間結果はパケットロスが 50% から 90% を超えていることもあり，通信性能が非常に悪い地点であるように見える．一方で実データでは，それほど悪くなく，0% から 60% 程度で推移している．補間前の元データを確認すると，図

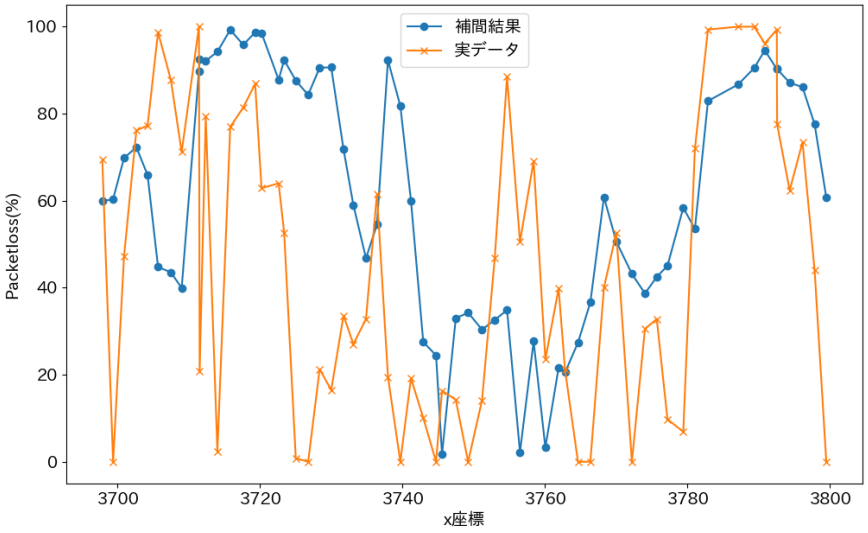

図 5.25. 線形補間評価

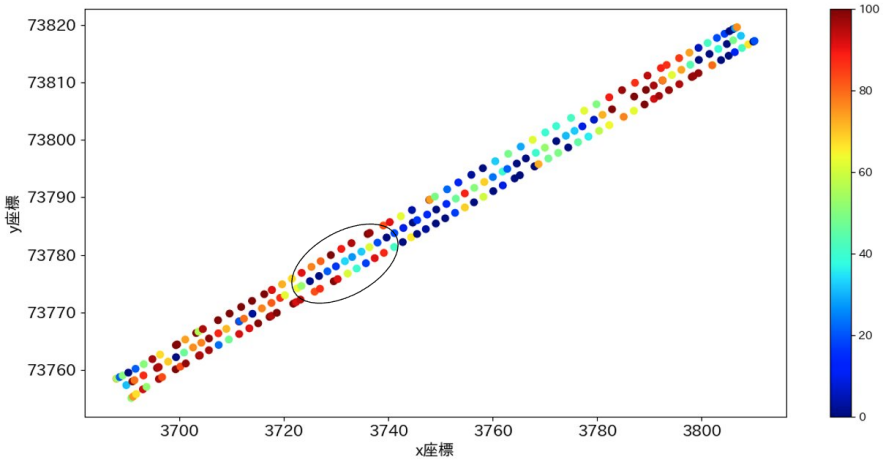

図 5.26. 補間元データ

5.26 の x 座標が 3720 から 3740（黒い楕円）の間で，左車線と右車線を走行したときのパケットロスと中央線を走行したときのパケットロスが大きく異なっている．x 座標が 3720 から 3740 の間で補間精度が落ちてしまったと考えられる．結果として，線形補間では，急激なパケットロスの変動や，ある地点のパケットロスが周囲のパケットロスと異なる場合に，補間精度が低下した．

動的物体による影響評価

動的物体による NP の影響を評価するために，図 5.27 のように，RSU を設置し，地点 A から地点 B に向けて，オレンジの経路を車両が走行する．このとき，赤で示した位置に図 5.28 のようにトラックを設置した状態としていない状態でパケッ



図 5.27. 動的物体による影響実験



図 5.28. トラック画像

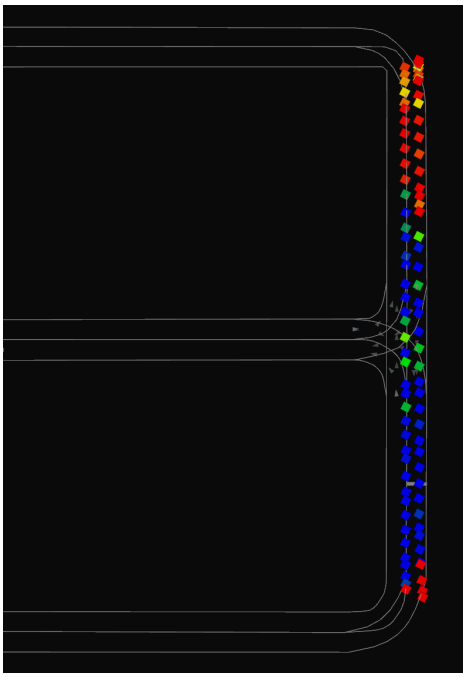

図 5.29. トラックなしの場合のパケットロス

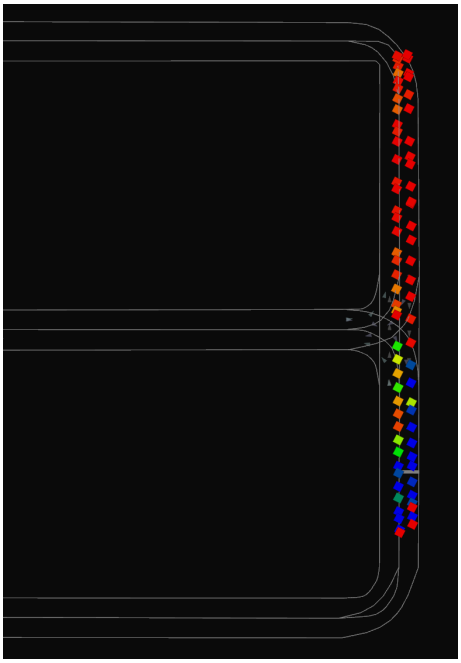

図 5.30. トラックありの場合のパケットロス

トロスを比較する．トラックを設置していないときのパケットロスを図 5.29 に，設置しているときのパケットロスを図 5.30 に示す． また，2 つのシナリオのパケットロスを比較したグラフを図 5.31 に示す．図 5.31 は横軸を地点 A からの距離，縦軸をパケットロスとして，トラックありの場合とトラックなしの場合をそれぞれ折れ線グラフで表示したものである．トラックの死角となっている範囲は地点 A からの距離が 19.8m から 49.9m の間である．この図を見ると，トラックありの場合，

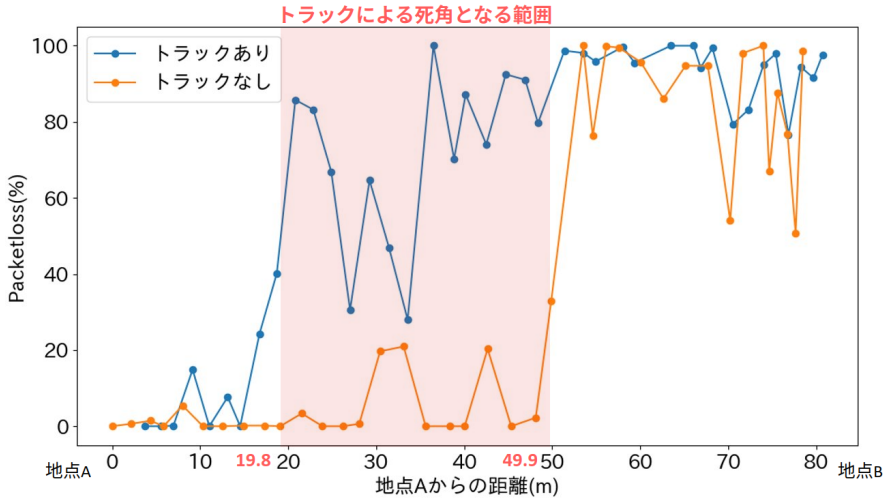

図 5.31. パケットロス比較（トラックありとトラックなし）



図 5.32. 石尾台地区経路と RSU 設置位置

地点 A からの距離，15m から 50m くらいの間で，トラックありのほうが，パケットロスが大きくなっており，トラックの影響を受けていることがわかる．

5.3.2　住宅街実験

愛知県春日井市石尾台地区で NP 分析，可視化実験を行った．石尾台地区の俯瞰図を図 5.32 に示す．石尾台地区での実験では，図 5.33 のように，実際に DM2.0 コンソーシアムで使用されている RSU の中に NUC を設置し，アンテナを外に出すようにして RSU を設置した．アンテナの高さは約 3m 程度である．また，図 5.34 のように，LiDAR（HDL-64E）の設置され



図 5.33. 石尾台 RSU



図 5.34. 石尾台車両

た車両に，アンテナを設置して実験を行った．RSU の設置位置と，車両の走行経路を図 5.32 の通りである．

このときの RSU と車両間のパケットロスを可視化した様子を図 5.35 に示す．また，パケットロスを線形補間した結果を図 5.36 に示す．交差点に対して横方向は広く通信できているが，縦方向のパケットロスが比較的大きくなっていることがわかる．これは，RSU の設置位置からみて，図 5.37 のように，縦方向は建物に阻害されて見通しが悪くなっていることが原因だと考えられる．

石尾台地区は，住宅街であり実環境のデータを得ることができた．そこで，このデータを使用して，NP マップの有用性の評価を行う．具体的には，一度取得したデータから補間を行い生成した NP マップが，その後通る車両にとって，どの程度参考になるのかを評価する．図 5.38 の地点 A から B に向かって走行し，線形補間を行って作成した NP マップに対して，その 3 分後，5 分後に同じ経路を通る車両の NP と比較を行う．

まず，生成した NP マップと 3 分後に走行した車両の NP の比較を図 5.39 に示す．地点 A からの距離が 0m から 30m，90m から 120m の範囲では，NP マップと 3 分後のデータはほとんど一致している．一方で，30m から 90m の間では，NP マップと比較して，3 分後に得られたデータでは，パケットロスが多く発生した．原因として，3 分後に得たデータでは，進行方向（RSU と車両の間）に別の車両が存在したため，その影響で電波が阻害されたことが考えられる．また，住宅街であるため，他のネットワークとの干渉の可能性も挙げられる．

次に，生成した NP マップと 5 分後に走行した車両の NP の比較を図 5.40 に示す．こちらは，ほとんど一致しており，5 分後のデータについては NP マップを参照することで，事前に通信性能を予測することができたと言える．以上のことから，生成した NP マップは動的物体

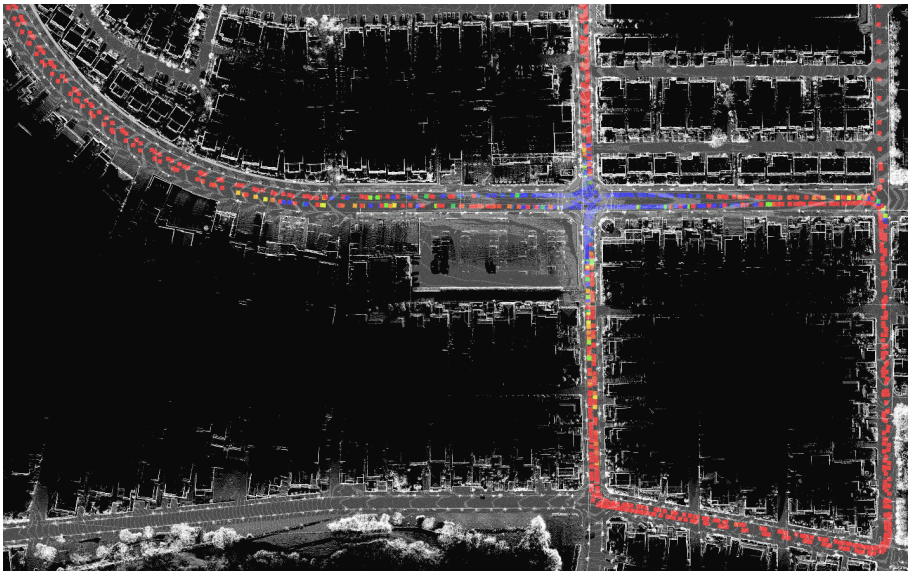

図 5.35. 石尾台地区可視化結果

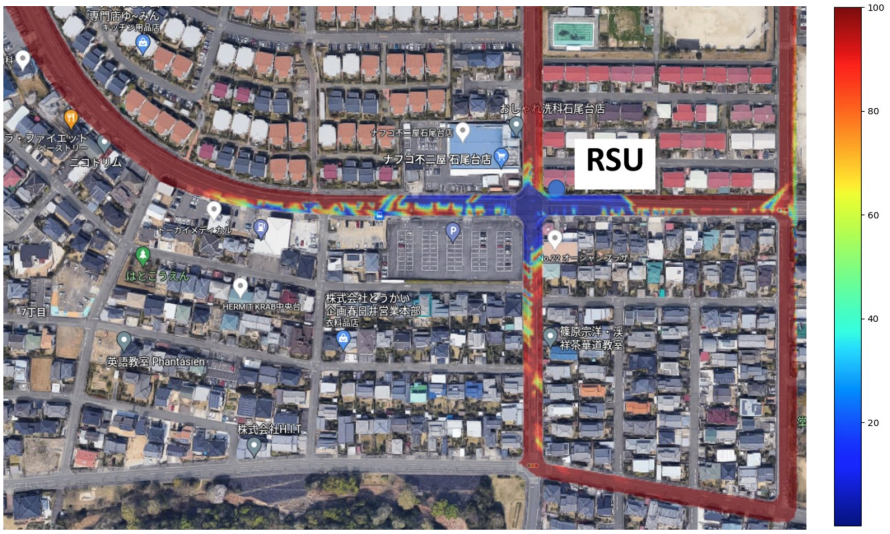

図 5.36. 石尾台地区線形補間結果

や，電波干渉などに影響を受けるため，必ずしも未来の通信性能と一致するわけではないが，ある程度参考にできると考えられる．

5.4　まとめ

協調型自動運転のための通信性能分析，可視化システムを提案，実装し，インドア，テストコース，住宅地で評価を行った．システムの動作を確認し，NP マップを生成することができた．NP の補間では，線形補間を適用した．平均的には，それほど大きな誤差はなかったが，

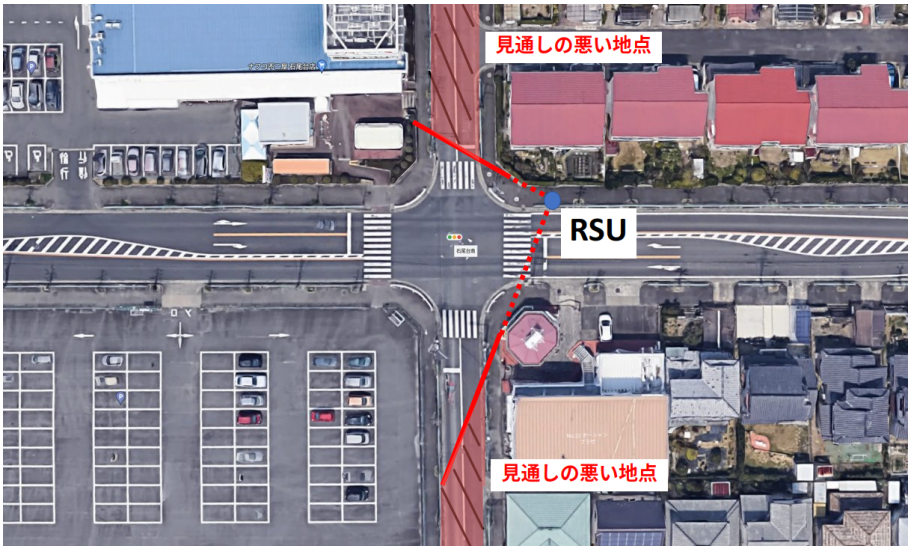

図 5.37. 石尾台地区 RSU から見通しの悪い地点



図 5.38. 石尾台地区 NP マップ有用性評価用経路

急激なパケットロスの変化に対応できていないため，補間精度を上げるためには，データの間隔をより小さくするか，急激なパケットロスの変化に対応できる別の手法を取り入れる必要がある．また，実際に住宅地を走行したデータから，NP マップは，電波干渉や，動的物体の影響を受けるため，必ずしも未来の通信性能と一致するわけではない．しかし，場合によってはほとんど一致している場合もあり，その後同じ場所を走行する車両にとって十分参考にすることができる精度であることが確認できた．

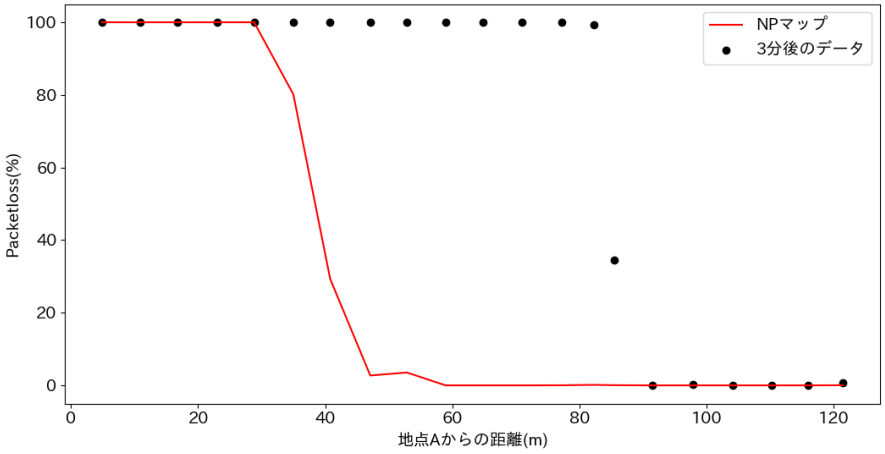

図 5.39. 石尾台地区 NP マップと 3 分後の NP の比較

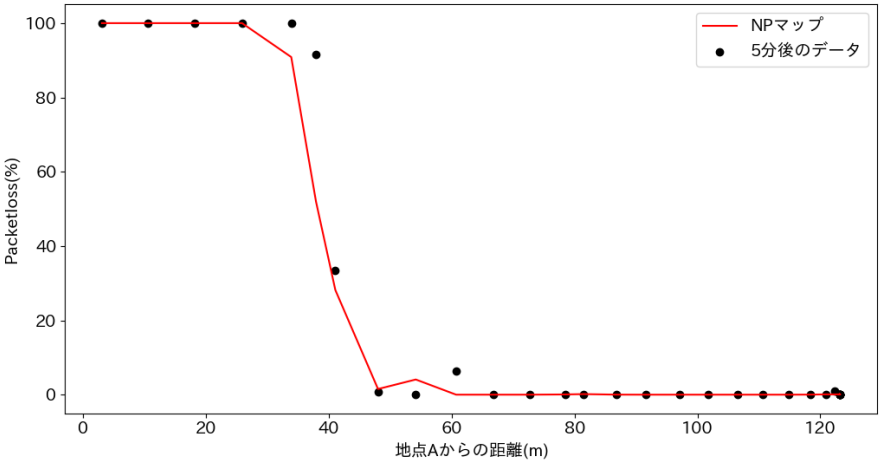

図 5.40. 石尾台地区 NP マップと 5 分後の NP の比較

第 6 章

結論

6.1　まとめ

協調型自動運転車は，V2X 通信によってリアルタイムで情報やタスクを共有し，安全で効率的な自動運転を行う．しかし，移動体である車両の通信性能は，通信先との距離やネットワークの干渉，混雑等により常に変動しており，協調認知，遠隔操作のために十分な通信性能を維持できないことがある．通信性能が低いと，パケットロスによって死角の先の情報が手に入らないことや，遠隔運転での遅延などによって操作が間に合わないなどの危険が存在する．事前に通信性能を把握することで，協調型自動運転や，遠隔運転で，適切な認知方法の切り替えや経路計画ができるようになるが，現状では，適切な通信性能を分析，可視化を行うシステムが存在しない．そこで本研究では，要求事項を満たす，通信性能の分析，可視化システムを作成した．作成したシステムの動作と精度を評価するため，移動を制限した環境下での実験，専用区画の実証フィールドでの実験，住宅街の公道での実験を行った．これらの実験により，実装システムが問題なく動作できることを確認した．通信性能の補間では，線形補間を適用したところ，平均的には大きな誤差はなかったが，急激にパケットロスが変動する場所では，部分的に大きな誤差が発生した．これを避けるためには，データの間隔をより小さくするか，急激なパケットロスの変動に対応できる補間手法に切り替える必要がある．また，生成した NP マップは，電波干渉や動的物体の影響を受けるため，必ずしも未来の通信性能と一致するわけではないことの知見が得られた．

6.2　今後の課題

今後の課題としては，実際に生成した NP マップを実際にどのように役立てるかを考える必要がある．具体的には，NP マップの値がどの程度のときに自律型と協調型を切り替えるのがよいのか，遠隔運転の場合は，NP マップがどのくらい悪い数値の場所は通行を避けるべきなのか等を考える必要がある．

また，本研究では，NP マップの生成にそれほど多くのデータを使用しなかったが，同じ場所を 10 回通過して外れ値を取り除くなどのデータの分析，加工をすることで，NP マップの

精度を向上させることができると考えられる.

発表文献と研究活動

(1) Koichi Kambara, Ehsan Javanmardi, Jin Nakazato, Yousuke Watanabe, Kenya Sato, Hiroaki Takada, Manabu Tsukada. Towards Cooperative Automated Driving: Geographic-Aware Network Analysis and Visualization tool. AINTEC2022.

(2) 神原滉一, Ehsan Javanmardi, 中里仁, 山田俊也, 渡辺陽介, 高田広章, 佐藤健哉, 塚田学, 協調型自動運転のための地理的特性を考慮したネットワーク補間, 電子情報通信学会 ITS 研究会, 2023.

参考文献

[1] 全国の都市における人の動きとその変化. https://www.mlit.go.jp/report/press/content/001573782.pdf, 2022.

[2] 令和 3 年における交通事故の発生状況等について. https://www.npa.go.jp/bureau/traffic/bunseki/nenkan/040303R03nenkan.pdf, 2022.

[3] 須田義大, 大口敬, 中野公彦, 大石岳史, 小野晋太郎, 吉田秀範, 杉町敏之. 自動運転システムの社会実装に関する課題と展望. 生産研究, No. 68(2), pp. 95–98, 2016.

[4] 柏の葉地区における自動運転バス長期営業運行実証実験. https://www.sip-adus.go.jp/evt/workshop2020/file/ra/04RA_03J_Shimono.pdf, 2020.

[5] Honda sensing elite press information. https://www.honda.co.jp/factbook/auto/LEGEND/202103/202003_LEGEND.pdf, 2022.

[6] 自動運転のレベル分けについて. https://www.mlit.go.jp/common/001226541.pdf, 2017.

[7] Intelligent transport systems: Vehicular communications; basic set of applications; part 2: Specification of cooperative awareness basic service. *ETSI EN 302 637-2*, Vol. 1.4.1, , 2019.

[8] Intelligent transport systems; cooperative perception services (cps). *ETSI TS 103 324*, Vol. 0.0.14, , 2019.

[9] Intelligent transport systems (its); vehicular communications; basic set of applications; analysis of the collective perception service (cps). *ETSI TR 103 562*, Vol. 2.1.1, , 2019.

[10] Toyota its connect. https://toyota.jp/technology/safety/itsconnect/, 2022.

[11] 佐藤雄大, 樫原俊太郎, 大岸智彦. 遠隔運転の遅延時間が操作性に与える影響に関する検討. 電子情報通信学会論文誌, Vol. 120, No. RCS-89, pp. 105–110, 2020.

[12] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada. An open approach to autonomous vehicles. *IEEE Micro*, Vol. 35, No. 6, pp. 60–68, 2015.

[13] 慣性計測ユニット（imu）. https://www.epson.jp/prod/sensing_system/imu/, 2022.

[14] Gnss とは. https://www.gsi.go.jp/denshi/denshi_aboutGNSS.html.

[15] Apollo: Open source autonomous driving. https://github.com/apolloauto/

apollo, 2017.

[16] 埼玉工大の自動運転機能後付けバスとその安全性向上に関する研究. https://www.sit.ac.jp/media/2020_01_wang_zhi.pdf, 2020.

[17] Ros2 documentation. http://wiki.ros.org/ja/rviz.

[18] 塚田学. 自動走行を支える協調型 its と車々間・路車間ネットワークの展望. 車載テクノロジー, Vol. 6, No. 11, pp. 1–7, 2019.

[19] intelligent transport systems(its); vehicular communications; basic set of applications; part 3: Specifications of decentralized environmental notification basic service. *ETSI EN 302 637-3*, Vol. 1.2.2, , 2014.

[20] Intelligent transport systems—extension of map database specification for local dynamic map for applications for cooperative its. *ISO/TS*, 2013.

[21] Intelligent transport systems (its); vehicular communications; basic set of applications; local dynamic map (ldm). *ETSI EN 302 895*, Vol. 1.1.1, , 2014.

[22] Ldm api and usage reference. http://www.safespot-eu.org/documents/SF_D7.3.1_Annex2_LDM_API_and_Usage_Reference_v0.7.pdf, 2020.

[23] Yousuke Watanabe, Kenya Sato, and Hiroaki Takada. Dynamicmap 2.0: A traffic data management platform leveraging clouds, edges and embedded systems. *International Journal of Intelligent Transportation Systems Research*, pp. 77–89, 2020.

[24] ダイナミックマップ 2.0 コンソーシアム：sip-adus の先のダイナミックマップを狙ったコンソーシアム型共同研究. https://www.nces.i.nagoya-u.ac.jp/dm2/, 2020.

[25] Etsi. multi-access edge computing - standards for mec. https://www.etsi.org/technologies/multi-access-edge-computing, 2020.

[26] 今健人, 渡辺仁. ダイナミックマップ 2.0 を活用した自動運転車両に対するインフラ情報配信システム. *YAMAHA MOTOR TECHNICSL REVIEW*, pp. 104–108, 2022.

[27] 先進モビリティサービスのための情報通信プラットフォームに関するコンソーシアム. http://nces.i.nagoya-u.ac.jp/ddm2/images/admobi-dm2-call_for_members_2022112816.pdf, 2022.

[28] 大井貴晴. 路車間通信を利用した自動走行のための協調認識システムの提案と評価. Master's thesis, 修士論文, 東京大学大学院 情報理工学系研究科, 2020.

[29] 水谷将也. 協調型自動運転のための機能抽象化した走行調停サービス. Master's thesis, 修士論文, 東京大学大学院 情報理工学系研究科, 2021.

[30] 佐藤雄大, 樫原俊太郎, 大岸智彦. 遠隔運転環境の操作性向上のための遅延の見える化機能. 電子情報通信学会論文誌 B, Vol. J105-B, No. 3, pp. 270–282, 2022.

[31] I.S. MacKenzie and C. Ware. Lag as a determinant of human performance in interactive system. INTERACT' 93 and CHI' 93 conference on Human factors in computing systems,, pp. 488–493, 1993.

[32] J. Davis. The effects of time lag on driving performance and a possible mitigation. *IEEE Trans Robotics*, Vol. 26, No. 3, pp. 590–593, 2010.

[33] L.H. Frank. Effects of visual display and motion system delays on operator performance and uneasiness in a driving simulator. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, Vol. 30, No. 2, pp. 201–217, 1988.

[34] A Napolitano, F Civerchia, K Kondepu, L Valcarenghi, and P Castoldi. Implementation of a mec-based vulnerable road user warning system. AEIT International Conference of Electrical and Electronic Technologies for Automotive（AEIT AUTOMOTIVE）, pp. 1–6, 2019.

[35] M. Kutila, P. Pyykonen, Q. Huang, W. Deng, W. Lei, and E. Pollakis. C-v2x supported automated driving. IEEE International Conference on Communications Workshops（ICC Workshops）, pp. 1–5, 2019.

[36] Manabu Tsukada, Jos'e Santa, Satoshi Matsuura, Thierry Ernst, and Kazutoshi Fujikawa. On the experimental evaluation of vehicular networks: Issues, requirements and methodology applied to a real use case. *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, Vol. 1, No. 1, pp. 1–14, 2014.

[37] 多田正範, 松本江里加, 島田秀輝, 佐藤健哉. 位置情報利用プロトコル（geonetworking）による車両情報配信の効率化. *DICOMO*, pp. 286–293, 2014.

[38] 伊藤彰秀. V2X 協調型運転システムにおける通信性能のリアルタイム可視化ツール, 2020.

[39] Wireshark. `https://www.wireshark.org/`.

[40] Rtp: A transport protocol for real-time applications. `https://www.rfc-editor.org/rfc/rfc3550`, 2003.

[41] jsk-ros-pkg. `https://github.com/jsk-ros-pkg`, 2022.

[42] jsk_rviz_plugins. `https://jsk-visualization.readthedocs.io/en/latest/jsk_rviz_plugins/`.

[43] Piechart. `https://jsk-visualization.readthedocs.io/en/latest/jsk_rviz_plugins/plugins/pie_chart.html`.

[44] Plotter2d. `https://jsk-visualization.readthedocs.io/en/latest/jsk_rviz_plugins/plugins/plotter_2d.html`.

[45] Matplotlib: Visualization with python. `https://matplotlib.org/`.

[46] rviz/displaytypes/marker. `http://wiki.ros.org/rviz/DisplayTypes/Marker`.

[47] Ieee sa. `https://standards.ieee.org/`.

[48] 光スペクトラムアナライザ（osa）. `https://www.viavisolutions.com/ja-jp/products/optical-spectrum-analyzers-osa`.

謝辞

本論文の執筆にあたり多くの方々にご指導とご協力をいただきました．ここに心より感謝申し上げます．

指導教官として研究の進め方や方向性を温かくご指導くださり，様々な勉強の機会を提供してくださった塚田学准教授に深く感謝いたします．

合宿やバーベキューなど，研究室が合同で活動する場において，鋭いご指摘をくださった江崎浩教授と，落合秀也准教授に感謝いたします．

協調型 ITS のグループにおいて，研究の方向性がなかなかまとまらない中で，様々なアイデアの提案や丁寧にご指導してくださった Ehsan Javanmardi 助教授，中里仁助教授に感謝いたします．

また，グループの中で共に励まし合い，アドバイスをくれた辻尾康平氏，廣瀬稔彦氏，Penfei Lin 氏，Ye Tao 氏，浅部佑氏，Vishal Chauhan 氏，Nicholaus D Yosodipuro 氏，松本和人氏，Yuze JIANG 氏に感謝いたします．

DM2.0 コンソーシアムにおいて，お忙しい中で毎月ミーティングを行い，様々な視点からの有益なご指摘をくださった名古屋大学の高田広章教授，渡辺陽介准教授，同志社大学の佐藤健哉教授に感謝いたします．

また，DM2.0PF について，動かし方から必要な知識等，様々な助言をいただいたり，実験に協力してくださった山田峻也助教授，徳井竜也氏，草山真一氏，川田福和氏に感謝いたします．

実験や出張，機材の購入など研究生活の様々な面でバックアップをしていただいた岩井愛映子秘書，高橋富美秘書に感謝いたします．

長い学生生活の中で，遠い場所から温かく見守り，勉強に集中できるように精神面，経済面等を支えていただいた，両親，祖父母に感謝いたします．

最後に学生生活をあらゆる面で支えてくださった，友人，先生方，すべての皆様に深く感謝いたします．

2023 年 1 月 20 日

神原 滉一