

博士論文

Mobile Edge Computing を用いた
遠隔車両制御プラットフォーム

佐々木 健吾

論文要旨

2015年9月に国際連合で開催されたサミットの中で、「Sustainable Development Goals (SDGs)」と呼ばれる国際社会共通の開発目標が提唱された。本博士論文では、SDGsの開発目標の1つである「任意の場所へ普遍的なアクセスを提供する交通システムの確立」を究極の目標とする。

目標とする交通システムの実現には、自動車の自律制御による自動運転(以降、自律制御)は必須の技術である。しかし、自律制御には「自車センサによる情報収集の範囲」と「車両間の制御調停」に課題があるため、自律制御だけで目標とする交通システムを実現することは難しい。

我々は上記2つの自律制御に関する課題を解決するためには、広範囲の情報収集が可能な協調運転システムが必要であると考えている。協調運転システムでは、車両同士が連携してスムーズな道路交通を実現する。しかし、現在、多く議論されている協調運転システムでは、1つの交差点のような局所的な交通制御(局所交通制御)を実現することが目的であり、渋滞情報のような広域情報を考慮することが難しい。我々は、広域情報を考慮した上で、局所交通制御を行わない限り、目標となる交通システムは実現できないと考えており、「超広範囲の協調運転システム」の構築が必要であると考えている。ここで、「超広範囲」とは東京都全体や日本全体といった規模を示す。

この「超広範囲の協調運転システム」を構築するために、我々は「Mobile/Multi-access Edge Computing (MEC)」と「遠隔制御」が必要であると考えている。「MEC」は第5世代移動体通信網にて「超低遅延通信」を実現するための主要な技術として検討されており、基地局のようなネットワーク装置に「エッジサーバ」と呼ばれる汎用的な計算装置を配置し、そのエッジサーバ上で分散処理を行う技術である。エッジサーバは、クラウドと比較してユーザ端末と近い位置から情報処理を実現できるため、ユーザ端末—計算装置間の通信遅延を抑制することができる。「遠隔制御」は、クラウドやエッジサーバのような外部装置が車両からセンサ情報を収集・解析し、車両に対して制御情報を提供することで車両の制御を実現する。遠隔制御を用いることで、制御モデルが集中制御となるため、車両間の制御の調停を容易に解決することができる。現在、クラウドから遠隔制御を行うシステム(クラウド型遠隔制御システム)や、エッジサーバから遠隔制御を行うシステム(MEC型遠隔制御システム)は提案されている。しかし、クラウド型遠隔制御システムでは「通信遅延」が、MEC型遠隔制御システムでは「エッジサーバがセンサ情報を収集できる範囲(以降、収集範囲)」が大きな課題となっている。以上の「MEC」と「遠隔制御」を用いて、我々は超広範囲の協調運転システムを実現することを目標とする。しかし、そこには、大き

く3つの課題「プラットフォームに関する課題」、「プラットフォームの評価に関する課題」、「実通信環境への適応に関する課題」が存在する。

プラットフォームに関する課題とは、超広範囲の協調運転システムを実現するためのプラットフォームが存在しないことである。従来手法では「通信遅延」と「収集範囲」のトレードオフが課題となることに加えて、自律制御が考慮されていないため、超広範囲の協調運転システムを実現することは困難である。加えて、従来手法では、超広範囲の協調運転システムを実現するためのネットワークアーキテクチャも提案されていない。2018年時点で、通信事業者はエッジサーバの具体的な配置を決定しておらず、移動体通信網内には、「基地局」や「局舎」、「ゲートウェイ」等の様々なエッジサーバを配置可能なネットワーク装置が存在する。エッジサーバの配置は「車両—エッジサーバ間の通信遅延」、「収集範囲」および、「プラットフォーム構築に必要なエッジサーバ数（以降、エッジサーバ数）」に影響を与える。以上のことを考慮できる超広範囲の協調運転システムのためのプラットフォームを提案する必要がある。

プラットフォームの評価に関する課題とは、プラットフォームを評価する環境が存在しないことである。超広範囲の協調運転システムを実現するためには、提案するプラットフォームで安定した遠隔制御が実現できることを示す必要があり、そのためには、実車評価と同等の特性を持つテストベッドで評価を行う必要がある。

実通信環境への適応に関する課題とは、提案プラットフォームを実通信環境に適応できるか評価する環境が存在しないことである。現在、日本国内には様々な通信キャリアが存在するものの、遠隔制御を想定した通信経路の特性について調査を行った過去の研究は存在しない。また、上記テストベッドを構築、評価を行い、提案するプラットフォームが安定して動作する通信条件を示せたとしても、実通信環境に適応した場合、多くの時間帯で制御を自律制御に頼っているのは、超広範囲の協調運転システムを実現することはできない。そこで、キャリアの通信特性を網羅的に調査し、提案するプラットフォームが実通信環境下で、広い収集範囲と低通信遅延を両立した遠隔制御を実現できるか調査する必要がある。

本博士論文では、超広範囲の協調運転システムの実現するために「インフラ型車両制御プラットフォーム」を提案し、上記3つの課題を「プラットフォーム提案」、「テストベッド構築・評価」および、「実通信環境の測定・評価」によって解決する。プラットフォーム提案では、多層エッジサーバによる遠隔制御と自動車による自律制御が連携するプラットフォームを提案する。提案

プラットフォームは移動体通信網内に多層のエッジサーバを配置する「ネットワークアーキテクチャ」および、多層エッジサーバによる遠隔制御と自動車による自律制御を連携させた「車両制御アーキテクチャ」によって構成される。本論文の多層エッジサーバは Upper Edge Server (UpES) と Lower Edge Server (LoES) の 2 層のエッジサーバで構成され、UpES は LoES より上位(車両から離れた位置)のネットワークレイヤに配置されるため、UpES は広範囲のセンサ情報を利用した処理を、LoES は低通信遅延環境下での処理を実現することができる。UpES/LoES の配置は、「基地局」、「局舎」、「ゲートウェイ」、「クラウド」の間で調整可能であり、配置の変更は「車両—エッジサーバ間の通信遅延」、「収集範囲」、「エッジサーバ数」に影響を与える。これら 2 層のエッジサーバによる遠隔制御と自動車による自律制御を通信状態に基づいて切り換えるプラットフォームを構築する。

テストベッド構築・評価では、提案プラットフォームが安定して車両を制御できるか評価するために、2 層のエッジサーバから遠隔制御可能な実車の 1/10 スケールのマイクロカーによるテストベッドを構築し、2 種類の評価「基本特性評価」と「実通信特性評価」を行う。基本特性評価では、エッジサーバとマイクロカーの間に「定常遅延」、「バーストパケットロス」を加え、遠隔制御の通信負荷に対する特性を評価する。基本特性評価の結果、定常遅延が 100ms 以下で、バーストパケットロスの期間が 400ms 以下なら、大きな走行軌跡のずれが発生することなくエッジサーバから遠隔制御を実現できることを示す。実通信特性評価では、携帯端末とクラウドサービスの間で終日通信データを計測し、計測データの中で遅延の変化が激しい時間帯の通信遅延をテストベッド上で再現し、遠隔制御の評価を行う。実通信特性評価の結果、クラウドに配置された UpES と基地局に配置された LoES を切換えながら遠隔制御を行うことで、通信遅延の無い理想環境と同等の安定制御を実現できることを示す。

実通信環境の測定・評価では、まず、実通信環境を考慮した評価を行うために、様々なキャリア-クラウドサービス間の実通信データを 24 時間測定する。計測対象のクラウドサービスは 3 つ、キャリアは 11 である。通信計測ができれば、その実通信データを用いた提案プラットフォームの車両制御シミュレーションを行う。車両制御シミュレーションでは、UpES, LoES の配置を「基地局」、「局舎」、「ゲートウェイ」および、「クラウド」の間で変更し、「UpES による遠隔制御」、「LoES による遠隔制御」および、「自動車による自律制御」の制御比率(各制御が車両制御を担当した時間の割合)を導く。シミュレーションの結果、「UpES の配置を調整することで LoES-UpES

間の制御比率と収容台数のトレードオフを調整できること」,「LoES の配置を調整することで遠隔制御と自律制御の制御比率とエッジサーバ数のトレードオフを調整できること」 および,「UpES をクラウドに, LoES を局舎に配置することで, エッジサーバ配置のコストを抑制しつつ, 広範囲の遠隔制御と低遅延な遠隔制御を両立する提案プラットフォームを実現できること」を示す.

以上より,

- 超広範囲の協調運転システムを実現可能なプラットフォームを提案する
- テストベッドによる評価から, 提案プラットフォームが安全に動作する条件を確立する
- シミュレーションによる評価から, 提案プラットフォームを安全に動作する条件を満たしつつ, 多くの時間帯で広収集範囲のエッジサーバから遠隔制御できることを示す

ことによって, 超広範囲の協調運転システムの実現可能性を示す.

ABSTRACT

Cooperative driving systems, which achieves smooth road traffic by a collaboration of vehicles, have caught much attention recently. In the cooperative driving systems, local traffic control, such as collision avoidance in one intersection, is often achieved, while global traffic control, such as the navigation system considering congestion information, cannot be achieved. We consider that smooth road traffic can be realized by both local and global traffic control.

In this paper, our ultimate future is “**ultra-wide range cooperative driving system**”. The “ultra-wide range” means the scale of the whole prefecture or country. We consider that the ultra-wide range cooperative driving system is achieved by the collaboration of local and global traffic control.

On the other hand, to achieve stable ultra-wide range cooperative driving system, self-autonomous vehicles that execute “autonomous control” are required. In this thesis, autonomous control is defined as a control system by which vehicles collect peripheral information using attached sensors and determine the control of themselves. However, only self-autonomous vehicles cannot achieve cooperative driving system. The first problem is the information collectible range by sensors. Sensors attached to vehicles cannot collect a wide range of information. The second problem is control mediation among vehicles. The self-autonomous vehicles can cause “deadlock” because the self-autonomous vehicles execute stable control from the limited sensor information. This problem is a fundamental problem of the distributed control systems, and it is difficult that only self-autonomous vehicles can solve this problem. We consider “5th Generation cellular mobile communication (5G) and Mobile/Multi-access Edge Computing (MEC) ” and “remote control” can solve above two problems and achieve the ultra-wide range cooperative driving system.

5G is the next generation cellular mobile communication and aims at “enhanced Mobile Broadband (eMBB) / massive Machine Type Communication (mMTC) / Ultra-Reliable and Low Latency Communications (URLLC)”. MEC has caught much attention to achieve URLLC in 5G. MEC installs computational nodes referred to as edge servers in the mobile network. The edge server can process the data from vehicles with low latency compared with the cloud.

In this thesis, the remote control is defined as a control system by which external equipment, such as the cloud and edge servers, collects and analyzes sensor information from multiple vehicles and

controls them. Different from autonomous control, the remote control can achieve central control and easily solve the mediation of control among vehicles.

To achieve the ultra-wide range cooperative driving system using 5G/MEC and the remote control, however, there are three challenges, i.e. “problem about the remote control platform”, “problem about the platform evaluation” and “problem about the platform adaptation to the actual communication environment”.

The first challenge is that there is no platform to achieve the ultra-wide range cooperative driving system. The remote control system using the cloud (cloud-based remote control) or edge-server (MEC-based remote control) have been proposed. However, the cloud-based and MEC-based remote controls have the problem about the communication latency and the range edge servers can collect sensor information from vehicles (sensor range), respectively. Therefore, the previous control systems cannot achieve the ultra-wide range cooperative driving system. Furthermore, previous systems cannot consider autonomous control. As the previous mentioned, although only autonomous control by the self-autonomous vehicles cannot achieve cooperative driving, the control architecture with the combination of the autonomous and remote control is required to achieve stable cooperative driving systems. At last, there is no network architecture to achieve the ultra-wide range cooperative driving system. In the mobile network, there are some candidates to deploy edge-servers, such as the base station, the central office, the gateway and so on. The locations of the edge-servers affect “the latency between vehicle and edge-server”, “sensor range” and “the number of edge-servers to construct the platform (the number of ESs)”.

The second challenge is that there is no evaluation environment for the platform, even if we can propose the platform. We need to show that the proposed platform can achieve the safety remote control and evaluate the proposed platform using the test-bed which has equivalent features with the evaluation using real vehicles.

The third challenge is that there is no environment for evaluating whether the proposed platform is adaptable to the actual communication environment. Although there are various Mobile Network Operator (MNO) and Mobile Virtual Network Operator (MVNO) in Japan, no previous work evaluates these carrier communication features for the remote control. Therefore, we need to investigate these

carrier communication features. Furthermore, even if we show the proposed platform can safely control vehicles using autonomous control for many hours in an actual communication environment, the proposed platform cannot achieve the ultra-wide range cooperative driving system. Therefore, we need to investigate the ratio of control between autonomous and remote control using the proposed platform.

In this thesis, to achieve the ultra-wide range cooperative driving system, we propose “**Infrastructure based vehicle control platform**” and solve the above three challenges by “proposal of platform”, “construction of the test-bed and evaluation of the proposed platform using the test-bed”, “measurement of actual communication environment and evaluation of the proposed system using the actual communication data”, respectively.

First, we propose the platform coordinated remote control by multi-layered edge servers and autonomous control by vehicles. The proposed platform is composed of “the network architecture” which can deploy multi-layered edge servers and “the vehicle control architecture” which coordinates remote and autonomous control. In this thesis, multi-layered edge-servers are composed of Upper Edge Server (UpES) and Lower Edge Server (LoES), and UpES is deployed upper (farther from vehicles) layer than LoES. Therefore, UpES is susceptible to communication problems but has a small number of ESs and wide sensor range. On the other hand, LoES can execute low latency processing but has a large number of ESs and small sensor range. Furthermore, the proposed platform can move the location of UpES and LoES among the base station, central office, gateway, and cloud in the mobile network. The location of edge-servers affects “the latency between the vehicle and edge-server”, “sensor range” and “the number of ESs”. We propose the platform which switches the control among the autonomous control or the remote control by UpES or LoES based on communication condition.

Second, we construct the test-bed which can control 1/10 scale micro-car from multi-layered edge servers and evaluate whether our platform can control vehicles safely using the test-bed. We evaluate an essential feature and an actual communication feature of the remote control. The essential feature evaluation adds major communication load, i.e. “constant latency”, “burst packet loss” and “random packet loss”, to the test-bed and evaluate the feature of the proposed system against their communication load. From the essential feature evaluation, we show the following.

- If the constant latency is less than 100 ms, the cloud can achieve stable remote control.

- If the interval of burst packet loss is less than 400 ms, the cloud can achieve stable remote control.

In the actual communication feature evaluation, we measure the actual network latency of the Internet and select an hour that has a sharp fluctuating latency. We reproduce the selected latency data in the test-bed and evaluate the remote control. From the actual communication feature evaluation, we show that the remote control cooperated with UpES deployed at the cloud and LoES deployed at the base station can achieve stable control equivalent to the control in an ideal environment that has almost no communication load.

Third, to evaluate whether the proposed platform can adapt the actual communication environment, we measure the actual latency data between various cloud services and various carriers, and simulate the vehicle control of the proposed platform using the measured actual latency. The vehicle control simulation calculates control ratio among autonomous control, remote control by UpES and LoES, while the locations of UpES and LoES move. From the vehicle control simulation, we show the following.

- Our proposed system can control vehicles more than 90% from the UpES and suppress the autonomous control ratio to less than 0.6% by deploying the UpES and LoES at the cloud and the central office, respectively.
- The deployment of the UpES can adjust the trade-off between the sensor range and control ratio between the UpES and LoES.
- The deployment of the LoES can adjust the trade-off between the number of ESs and control ratio between autonomous and remote control.
- The autonomous control is a requirement regardless of the deployment of the UpES and LoES because unexpected burst packet loss cannot be prevented.
- Our platform, which deploys UpES to the cloud and LoES to the central office, can achieve the maximum sensor range and low latency communication while suppressing the number of ESs.

We propose the platform to achieve the ultra-wide range cooperative driving system and establish the condition which proposed platform achieves stable control from evaluation using test-bed. Furthermore,

from the simulation evaluation, we show that the proposed platform can achieve stable remote control by the edge server which can collect sensor information from a wide range in many time zones. From the above, we show the feasibility of the ultra-wide range cooperative driving system.

謝辞

本研究において、学生時代からの懇切な御指導、並びに御助言をいただき、格別のご配慮と暖かい激励を賜りました東京大学教授 中尾彰宏先生に心から深謝し、厚く御礼申し上げます。そして、本研究の遂行において、御議論、並びに貴重なご意見をいただきました東京大学特任講師 桐葉佳彰先生および山本周先生に深謝し、心から御礼申し上げます。また、本研究をまとめるにあたり、数々の貴重なご意見やご指導を頂きました東京大学教授 越塚登先生および暦本純一先生に深く感謝いたします。

本研究は、私が株式会社豊田中央研究所、及び、東京大学在学中における研究業務の一環として実施したものであり、多くの方々の御指導、御支援をいただきました。長年にわたり、ほぼ一貫して同一分野の研究に従事できましたのは、一重に上司の理解の賜物であります。まず、本研究の共同研究者であり直接の上司として御指導、御鞭撻を賜りました株式会社豊田中央研究所 走行環境理解研究室 室長 牧戸知史殿に心から御礼申し上げます。次に、本研究の3章の共同研究者で、貴重な御指導、御鞭撻を賜りました株式会社豊田中央研究所 走行環境理解研究室 鈴木有也研究員に深く感謝いたします。また、同一分野の研究テーマを継続して担当させてくれました、株式会社豊田中央研究所 システム・エレクトロニクス1部 部長 内藤貴志殿、同社 データアナリティクス領域 領域リーダー 小島祥子殿、同社 戦略研究企画・推進室 室長 寺寫立太殿に深く感謝いたします。

さらに、本研究の遂行にあたり、日頃から熱心な討論をさせていただき、有意義なご指導や多大なご協力をいただきました株式会社豊田中央研究所 和田錦一研究員、及び、東京大学大学院生 安藤翔伍氏、岩井貴充氏、雨宮宏一郎氏に感謝いたします。

最後に、私の研究活動を温かく見守り、励まし、常に心の支えとなってくれました妻 美保と長男 陸人に感謝いたします。また、これまで育て、見守ってくれました父 篤男、母 三喜子に感謝いたします。さらに、学位取得に関して、いつも気にかけて、励ましてくれました妻の父 保夫、妻の母 久子に感謝いたします。

2019年
佐々木健吾

目次

第 1 章 序論	9
1.1 現在の自動車産業および, 都市環境について	9
1.2 自動車の自律制御による自動運転について	10
1.3 協調運転について	11
1.4 第 5 世代移動体通信, Mobile/Multi-access Edge Computing について	12
1.5 遠隔制御について	12
1.6 5G/MEC と遠隔制御による協調運転システムの課題	13
1.6.1 プラットフォームに関する課題	13
1.6.2 プラットフォームの評価に関する課題	15
1.6.3 プラットフォームの実通信環境への適応に関する課題	15
1.7 提案手法	16
1.7.1 プラットフォーム提案	16
1.7.2 テストベッドの構築・評価	18
1.7.3 実通信環境の測定・評価	18
1.8 貢献	18
1.9 本論文の構成	20
第 2 章 関連研究	21
2.1 はじめに	21
2.2 自動車とクラウドの連携に関する研究	22
2.3 クラウド型遠隔制御に関する研究	23

2.4	自動車とエッジサーバの連携に関する研究	24
2.5	MEC 型遠隔制御に関する研究	25
2.6	まとめ	25
第 3 章	インフラ型車両制御プラットフォーム	26
3.1	はじめに	26
3.2	アーキテクチャ	27
3.2.1	UpES による遠隔制御	29
3.2.2	LoES による遠隔制御	29
3.2.3	自動車による自律制御	30
3.3	提案プラットフォームの動作	30
3.4	車両制御アルゴリズムについて	31
3.5	車両制御の切手法	33
3.6	提案プラットフォームを配置するネットワークトポロジ	33
3.6.1	エッジサーバ数/収容台数	35
3.6.2	通信遅延	36
3.6.3	移動に伴う車両の基地局間ハンドオーバについて	37
3.7	まとめ	37
第 4 章	遠隔車両制御テストベッド構築および, 評価	38
4.1	はじめに	38
4.2	マイクロカーを用いたテストベッド環境	39
4.2.1	Micro-car prototype	39
4.2.2	FLARE	41
4.2.3	WAN エミュレータ	43
4.3	基本特性評価	44
4.3.1	評価環境	44
4.3.2	車両走行軌跡の評価方法について	45
4.3.3	測定条件	46

4.3.4	測定結果：UpES 上での車両の走行制御 (定常遅延)	49
4.3.5	測定結果：UpES 上での車両の走行制御 (ランダムパケットロス)	50
4.3.6	測定結果：UpES 上での車両の走行制御 (バーストパケットロス)	51
4.3.7	評価のまとめ	52
4.3.8	安定制御に必要な切替閾値について	52
4.4	実通信特性評価	55
4.4.1	実通信遅延計測方法	55
4.4.2	評価方法	56
4.4.3	評価結果	56
4.5	まとめ	58
第 5 章	日本国内の通信遅延計測	62
5.1	はじめに	62
5.2	測定するネットワークの構造について	63
5.3	調査	64
5.3.1	調査方法	65
5.3.2	調査対象	65
5.4	調査結果	66
5.4.1	ネットワークトポロジー	66
5.4.2	パケットロス	69
5.4.3	通信遅延	70
5.5	各ネットワーク装置上でのロケーションパラメータについて	74
5.6	まとめ	75
第 6 章	実トラフィックを用いた提案プラットフォームのシミュレーション評価	77
6.1	はじめに	77
6.2	エッジサーバ配置のトレードオフ	79
6.3	シミュレーション評価	80
6.4	シミュレーション結果 (全体):制御比率, 収容台数, エッジサーバ数	81

6.4.1	超広範囲の協調運転の実現可能性について	82
6.4.2	UpES 配置の影響	84
6.4.3	LoES 配置の影響	84
6.5	シミュレーション結果 (キャリア別): 制御比率	85
6.5.1	UpES の配置効果 (LoES=基地局)	86
6.5.2	LoES の配置効果 (UpES=クラウド)	89
6.6	考察:最適なエッジサーバ配置について	92
6.6.1	UpES の配置	92
6.6.2	LoES の配置	92
6.7	まとめ	93
第 7 章	結論	95
7.1	まとめ	95
7.1.1	プラットフォーム提案	95
7.1.2	テストベッド構築・評価	96
7.1.3	実通信環境の測定/評価	97
7.2	今後の課題	98
7.2.1	多層エッジサーバモデルの拡張	99
7.2.2	5G 環境での評価	99
7.2.3	CORD 上での実装	99
7.2.4	ローカル 5G との連携	99
	参考文献	101

図一覧

1.1	インフラ型車両制御プラットフォーム	17
3.1	提案プラットフォームのアーキテクチャ	28
3.2	提案システム	31
3.3	状態遷移図	32
3.4	提案プラットフォームが動作するネットワークトポロジ	34
4.1	テストベッドの構成	40
4.2	Micro-car prototype	41
4.3	MPT のコース	42
4.4	FLARE アーキテクチャ	43
4.5	Click での MAC 実装例	44
4.6	基本特性評価環境	45
4.7	「ずれ」の計算方法	46
4.8	走行コースと標準軌跡	47
4.9	コースアウト	48
4.10	定常遅延の変化に対する「ずれ」の変化 (CDF)	49
4.11	エッジサーバでの走行制御 (定常遅延 100ms) による軌跡	50
4.12	エッジサーバでの走行制御 (定常遅延 150ms) による軌跡	51
4.13	ランダムパケットロス率の変化による走行軌跡のずれの変化	52
4.14	エッジサーバでの走行制御 (ランダムパケットロス 30%) による軌跡	53
4.15	バーストパケットロス率の変化による走行軌跡のずれの変化	54
4.16	計測環境	55

4.17	計測結果	57
4.18	実通信特性評価環境	58
4.19	CDF (ずれ)	59
4.20	例. 車両軌跡	60
4.21	UpES, LoES 間の制御切換の様子	61
5.1	移動体通信網の構造	64
5.2	ネットワークトポロジー	68
5.3	パケットロスの比較	69
5.4	最低遅延	70
5.5	遅延の中央値	71
5.6	遅延データ (D)	72
5.7	遅延データ (FB)	73
5.8	遅延データ (K)	74
5.9	遅延の比較 (ping, traceroute)	75
6.1	エッジサーバの配置とトレードオフ	79
6.2	制御比率 (平均) Up:UpES 制御, Lo:LoES 制御, V:自律制御	82
6.3	制御比率 (最小) Up:UpES 制御, Lo:LoES 制御, V:自律制御	83
6.4	UpES の収容台数	83
6.5	エッジサーバ数	84
6.6	制御比率 (UpES=クラウド, LoES=基地局)	85
6.7	制御比率 (UpES=ゲートウェイ, LoES=基地局)	86
6.8	制御比率 (UpES=局舎, LoES=基地局)	87
6.9	FB - 東京大学の制御比率 (LoES=基地局)	88
6.10	制御比率 (UpES=クラウド, LoES=無し)	89
6.11	制御比率 (UpES=クラウド, LoES=ゲートウェイ)	90
6.12	制御比率 (UpES=クラウド, LoES=局舎)	91
6.13	FB - 東京大学の制御比率 (UpES=クラウド)	91

6.14 LoES の配置変更時の自律制御比率とエッジサーバ数 (UpES=クラウド)	92
---	----

表一覧

2.1	VCC アプリケーションと情報の寿命	22
3.1	エッジサーバ数と収容台数	35
3.2	ロケーションパラメータ	36
4.1	測定パラメータ	47
4.2	計測パラメータ	56
4.3	測定パラメータ	58
5.1	クラウドサービス一覧	66
5.2	キャリア一覧	66
5.3	ロケーションパラメータ α_{up}, α_{lo}	76
6.1	ロケーションパラメータ α_{up}, α_{lo} , エッジサーバ数, 収容台数について	80
6.2	シミュレーションパラメータ	81

第 1 章

序論

1.1 現在の自動車産業および、都市環境について

2015 年 9 月に国際連合で開催されたサミットの中で、「Sustainable Development Goals（持続可能な開発目標、以降 SDGs）[1]」と呼ばれる国際社会共通の開発目標が提唱された。SDGs の中では、貧困の解決や衛生問題、エネルギー問題、環境汚染問題等、持続可能な社会を実現するために必要な 17 の開発目標とその開発目標のための具体的なターゲットが設定されている。

17 の目標のうちの 1 つ（目標 11）は「まちづくり」に関する開発目標で、その中には持続可能な社会を実現するための 10 のターゲットが設定されている。そのうちの 1 つであるターゲット 11.7 を以下に示す。

2030 年までに、女性・子ども、高齢者および、障害者を含め、人々に安全で包摂的かつ利用が容易な緑地や公共スペースへの普遍的アクセスを提供する。

このことは、持続可能な社会を実現するためには、安定した交通システムの確立が必要不可欠であることを示している。

SDGs の目標として掲げられていることから、現在の交通システムが万人に普遍的アクセスを提供しているとは言い難い。地方は公共交通インフラが乏しく、自家用車を所有していない場合、様々な場所へのアクセスが困難になる。そのため、車の運転ができない高齢者や子どもは公共スペース等へのアクセスが困難になる可能性が高い。一方で、都市部では公共交通は充実しているものの、高齢者にとっては公共交通の乗り場までのアクセスが困難である場合が存在する。地方と

比較すると、自家用車を所有していない人も多く、足の悪い高齢者にとっては地方以上に自由な移動が困難である可能性が高い。また、都市部では、許容量を超える自家用車やバス等によって渋滞問題が発生しており、効率的な交通が維持できているとは言い難い。渋滞中に発生する不要な排気ガスは、SDGs で提唱される環境問題 (目標 13) にもつながっている。

本博士論文では、持続可能な社会を実現するための「任意の場所へ普遍的アクセスを提供する交通システムの確立」を究極の目標とする。

1.2 自動車の自律制御による自動運転について

目標となる交通システムを実現するためには自動車の自律制御による自動運転 (以降、自律制御) は必須の技術である。本論文では、自律制御を自動運転車が自身に取り付けられたセンサ (Lidar や高精度カメラ等) を用いて周辺環境をセンシングし、それらの情報を用いて自身を制御するシステムと定義する。自律制御については、トヨタ [2] のような自動車企業だけでなく、google[3] や Apple[4] のような IT 企業、パナソニック [5] や三菱電機 [6] のような電機メーカー等、様々な企業・大学が研究開発を行っており、その注目度の高さが伺える。

一方で、我々は自律制御だけでは目標となる「任意の場所へ普遍的アクセスを提供する交通システム」を実現することはできないと考えている。第一の課題は、「センサによる情報収集の範囲」である。各車両に取り付けられたセンサで収集可能な情報の範囲は狭く、狭い路地や交差点等では安全な制御に必要となる情報を収集することができない。車々間通信 (以降、V2X)[7] を用いて、周辺車両と情報共有を行ったとしても、渋滞等を考慮できる広い範囲のセンサ情報の収集は実現できない。第二の課題は、「車両間の制御の調停」である。自律制御の自動運転車は収集したセンサ情報から「自身の判断」で安全な制御を行う必要がある。そのため、車が密集する交差点では、車両同士が限られたセンサ情報に基づいて安全を考慮することで互いに停止する「デッドロック」を引き起こす可能性がある。この制御調停の課題は、分散制御の根本的な課題であるため、たとえ広範囲の情報共有を実現できたとしても、完全に解決することは難しい。

我々は、これら 2 つの自律制御の大きな課題

1. センサによる情報収集の範囲
2. 車両間の制御の調停

を解決し、目標となる交通システムを構築するためには、自律制御に加えて、広範囲の情報が収集可能な「協調運転システム」を構築する必要があると考えている。

1.3 協調運転について

本論文では、協調運転システムを都市部において「複数車両に対して」超低遅延通信による位置取得・調停・制御を行うシステムと定義する。

現在、多く議論されている協調運転システムは、V2Xのような直接通信と車両自身による分散制御によって制御の調停を行い、「スムーズな交差点の通過」や「スムーズな高速道路の分合流」を実現する [8, 9]。制御対象となる空間は狭いものの、局所的な交通制御は実現可能である。

一方で、上記の協調運転システムでは、渋滞や交通規制のような広域情報を考慮していない。我々は、広域情報を考慮した上で、局所的な交通制御を行わない限り、目標となる交通システムは実現できないと考えている。言い換えると、「高速道路の分合流」や「交差点通過」のような局所的な交通制御と「渋滞制御」や「災害時交通の管理」のような全体的な交通制御を同時に行うことで、初めて任意の場所への普遍的アクセスを提供できると考えている。

我々は、広域情報を考慮して車両制御を行うためには、「車両自身による分散制御モデル」よりも「センターによる集中制御モデル」のほうが適していると考えている。ここで、集中制御モデルとは制御対象の情報を一箇所に集約・解析を行い、各制御対象に与える入力を決定するモデルである。分散制御モデルでは広範囲になるほど制御の調停が難しくなるため、我々は、車両の位置や速度といった情報を一箇所に集約して制御を行う集中制御モデルを用いることで、全体交通制御と局所交通制御の両立を実現できると考えている。

以上より、我々は究極の目標である「任意の場所へ普遍的アクセスを提供する交通システム」を確立するためには、集中制御モデルを用いた「超広範囲の協調運転システム」が必要であると考えている。ここで、「超広範囲」とは東京都全体や日本全体といった規模を示す。

我々はこの超広範囲の協調運転システムを実現するためには

1. 第5世代移動体通信, Mobile/Multi-access Edge Computing
2. 遠隔制御

が必要であると考えている。

1.4 第5世代移動体通信, Mobile/Multi-access Edge Computing について

超広範囲で情報共有を実現するために, 次世代の移動体通信網である第5世代移動通信システム (以降, 5G)[10, 11] は必須の技術である. 5G では, 3つの目標

- 高速大容量データ伝送を実現する「eMBB (enhanced Mobile Broadband)」
- 大量端末の同時接続を実現する「mMTC (massive Machine Type Communications)」
- 高信頼かつ低遅延通信を実現する「URLLC(Ultra-Reliable and Low Latency Communications)」

を提示している. その中で, 自動車が5Gに最も期待する通信要件はURLLCである. 国際電気通信連合 (ITU[12]) が進める5Gの標準化「IMT-2020[13]」の中でも, 自動車の自動運転が最も低遅延・高信頼を必要とするユースケースとして定義されている.

この5Gの中で, URLLCを実現するために有用な技術として Mobile/Multi-Access Edge Computing (MEC) [14, 15] に注目が集まっている. MECは, 移動体通信網の基地局のようなネットワーク装置上に「エッジサーバ」と呼ばれる汎用的な計算装置を配置し, そのエッジサーバ上で分散処理を行う技術である. クラウドと比較してユーザ端末と近い位置から情報処理を実現できるため, ユーザ端末-計算装置間の通信遅延を抑制することができる. 低遅延化技術として自動車業界からの注目も高く, 2017年にはMECを車両情報の処理に応用するためのコンソーシアムである Automotive Edge Computing Consortium (AECC)[16]も立ち上げられた.

我々は, 5G/MECを用いれば超広範囲の情報共有を低遅延に実現できると考えており, 広範囲の協調運転システム実現するための通信基盤として5G/MEC環境を想定する.

1.5 遠隔制御について

車両間の制御の調停は, 車両同士が互いに判断を行う分散制御モデルでは難しい. そこで, 我々は外部から車両を制御する「遠隔制御」に基づいた協調運転システムの構築を行う. 本論文では, 遠隔制御をクラウドやエッジサーバのような外部装置が車両からセンサ情報を収集・解析し, 遠隔から制御情報の提供を行うシステムと定義する. 遠隔制御を用いてシステムを構築することで, 制御モデルが分散制御ではなく集中制御となるため, 車両間の制御の調停を解決することができる.

すでに、クラウドを用いた遠隔車両制御（以降、クラウド型遠隔制御）システム [17, 18] が提案されている。クラウド型遠隔制御システムではクラウドがセンサ情報を広範囲に存在する車両から収集し、センサ情報の解析を行い、車両に対して制御情報の提供を行う。広範囲に存在する車両からセンサ情報を収集するほど、有効な情報を車両に提供できる一方で、車両-クラウド間の通信遅延低減が大きな課題となる。

クラウド型遠隔制御システムの通信遅延に関する問題を解決するために、MEC によるエッジサーバを用いた遠隔車両制御（以降、MEC 型遠隔制御）システムも提案されている [19, 20]。MEC 型遠隔制御システムでは、クラウド型遠隔制御システムと比較して車両に近い位置から情報処理を実現できるため、低遅延な車両制御を実現できる。しかし、クラウド型遠隔制御のように広範囲から車両のセンサ情報を収集することはできない。超広範囲の協調運転システム実現のためには、広範囲から車両のセンサ情報を収集できる装置が、広範囲の車両状態を把握したうえで、車両に対して指示を与えることが望ましい。

1.6 5G/MEC と遠隔制御による協調運転システムの課題

5G/MEC と遠隔制御を用いて、超広範囲の協調運転システムを実現するためには、大きく 3 つの課題が存在する。以下にそれらの課題を示す。

1. プラットフォームに関する課題
2. プラットフォームの評価に関する課題
3. 実通信環境への適応に関する課題

1.6.1 プラットフォームに関する課題

第一の課題は、超広範囲の協調運転システムを実現するためのプラットフォームが存在しないことである。

1.5 節で述べたように、すでにクラウド型、MEC 型の遠隔制御システムは提案されている。しかし、これら 2 つの制御システムの間には「エッジサーバがセンサ情報を収集できる範囲」と「車両-エッジサーバ間の通信遅延」のトレードオフが存在する。クラウドやエッジサーバ単体では、こ

れらを同時に実現することができないため、従来システムでは超広範囲な協調運転システムを実現することはできない。

また、従来システムの中では、自律制御が考慮されていない。1.2節で述べたとおり、自律制御だけでは協調運転システムを実現することは難しい。しかし、安全な超広範囲の協調運転システムを構築するために、遠隔制御と自律制御を組み合わせた車両制御アーキテクチャは必須である。

加えて、超広範囲の協調運転システムを実現するためのネットワークアーキテクチャも提案されていない。2018年時点で通信事業者からエッジサーバの具体的な配置計画は発表されておらず、移動体通信網内には、「基地局」、「局舎」、「ゲートウェイ」等様々なエッジサーバを配置可能なネットワーク装置が存在する。エッジサーバは配置されるネットワーク装置によって、下記の3つの特性が変化する。

- 車両-エッジサーバ間の通信遅延
- 収容台数
- エッジサーバ数

車両-エッジサーバ間の通信遅延はエッジサーバの配置が車両に近いほど短く、遠いほど長くなる。収容台数はエッジサーバがセンサ情報を収集できる車両の台数を示し、エッジサーバがセンサ情報を収集可能な範囲に影響を与える。例えば、基地局であれば、収容台数は基地局と無線通信を介して接続する車両の台数であり、複数台の基地局を集約する局舎¹であれば、局舎に接続された基地局と通信を行う車両の台数が収容台数となる。このことはエッジサーバの配置が車両から近いほど収容台数が小さく、遠いほど大きくなることを示す。言い換えれば、収容台数が大きいほど広範囲の車両のセンサ情報を収集できることを示す。エッジサーバ数は超広範囲の協調運転システム実現時に配置する必要があるエッジサーバの数を示し、設備投資コストに大きな影響を与える。例えば、基地局に配置する場合、エッジサーバ数は日本国内に数万と必要になる一方で、クラウドであればエッジサーバ数は数台である。すなわち、エッジサーバ数は、エッジサーバの配置が車両から近いほど多く、遠いほど少なくなり、エッジサーバ数が少ないほど設備投資コストを抑制することができる。

¹詳細は 3.6 節

以上で述べられた点を考慮できる超広範囲の協調運転システム実現のためのプラットフォームを提案する必要がある。

1.6.2 プラットフォームの評価に関する課題

第二の課題は、プラットフォームを提案できたとしても評価する環境が存在しないことである。提案するプラットフォーム上での協調運転システムを評価するためには、実車評価と同等の特性を持つテストベッドで評価を行う必要がある。本来であれば、協調運転システムの評価には実車を用いた性能評価を行うことが望ましい。しかし、実車を用いた評価環境の構築は難しく、また実車による評価では試行錯誤を繰り返す評価が難しい。その一方で、シミュレーションベースによる評価の場合、実車との差が大きく、実際に提案するプラットフォームを社会実装するとき大きな齟齬が生じる可能性がある。

超広範囲の協調運転システムを実現するためには、提案するプラットフォームは常に安定した車両制御を継続する必要がある。プラットフォームの安定制御を示すためには、テストベッドを構築し、様々な通信負荷をテストベッドに加えても、テストベッドが安定して制御を継続できることを示す必要がある。

以上より、テストベッド環境を構築し、テストベッド環境上で通信負荷が増大した時の安定制御に関する評価を行う必要がある。

1.6.3 プラットフォームの実通信環境への適応に関する課題

第三の課題は、提案するプラットフォームを実通信環境に適応可能か評価する環境が存在しないことである。

現在、日本国内には主要な Mobile Network Operator (MNO)²が3社存在することに加え、様々な Mobile Virtual Network Operator (MVNO)³が台頭している。しかし、これらキャリア⁴の遠隔制御を想定した通信経路の特性について調査を行った過去の研究は存在しない。そのため、キャリアの通信特性を網羅的に調査し評価する必要がある。

また、テストベッドによる評価から、提案するプラットフォームの安定制御を示せたとしても、

²物理的な移動体通信網を自社で保有する通信事業者

³物理的な移動体通信網を MNO から借りてサービスを提供する通信事業者

⁴本論文では MNO, MVNO の総称をキャリアとする

実通信環境に適応した場合,多くの時間帯で制御を自律制御に頼ってはいは,超広範囲の協調運転システムを実現することはできない.提案するプラットフォームの実通信環境への適応を評価するには,実通信環境下で遠隔制御を行える時間帯,自律制御が必要になる時間帯を調査する必要がある.

以上より,様々なキャリアの通信環境を把握した上で,提案するプラットフォームの実通信環境下での動作を評価する必要がある.

1.7 提案手法

本博士論文では,超広範囲の協調運転システム実現のために,自律制御と遠隔制御を連携させた「インフラ型車両制御プラットフォーム [21, 22, 23, 24]」を提案する.

図 1.1 は提案プラットフォームの概要を示す.本論文では,前節で定義された3つの課題を図 1.1 中の3つの項目で解決することで,超広範囲の協調運転システムの実現可能性を示す.

1.7.1 プラットフォーム提案

多層のエッジサーバによる遠隔制御と自動車による自律制御が連携する車両制御プラットフォームを提案する.

図 1.1 の 1 に提案プラットフォームの全体像を示す.提案プラットフォームは多層のエッジサーバ⁵を配置する「ネットワークアーキテクチャ」と,その多層エッジサーバによる遠隔制御と自動車による自律制御を連携させた「車両制御アーキテクチャ」によって構成される.

エッジサーバ間の遠隔制御の切換および,遠隔/自律制御の切換は通信環境に応じて行い,通信環境が安定(通信遅延が小さく,パケットロスが無い)している限り,上位のエッジサーバ(図 1.1 ではクラウド)から制御を行う.一方で,通信が不安定となった場合には,下位のエッジサーバ(図 1.1 では基地局)による遠隔制御および,自律制御に切り換えることで安定した制御を維持する.可能な限り広範囲の情報を収集可能な上位から制御を行い,非常時のみ下位から制御を行うことで,大きな収容台数と低遅延の両立を実現する.

本論文の多層エッジサーバは Upper Edge Sever (UpES) と Lower Edge Server (LoES) の 2 層で構成され,UpES は LoES より上位のネットワークレイヤに配置される.そのため,UpES は LoES

⁵本論文では,クラウドもエッジサーバに含める.

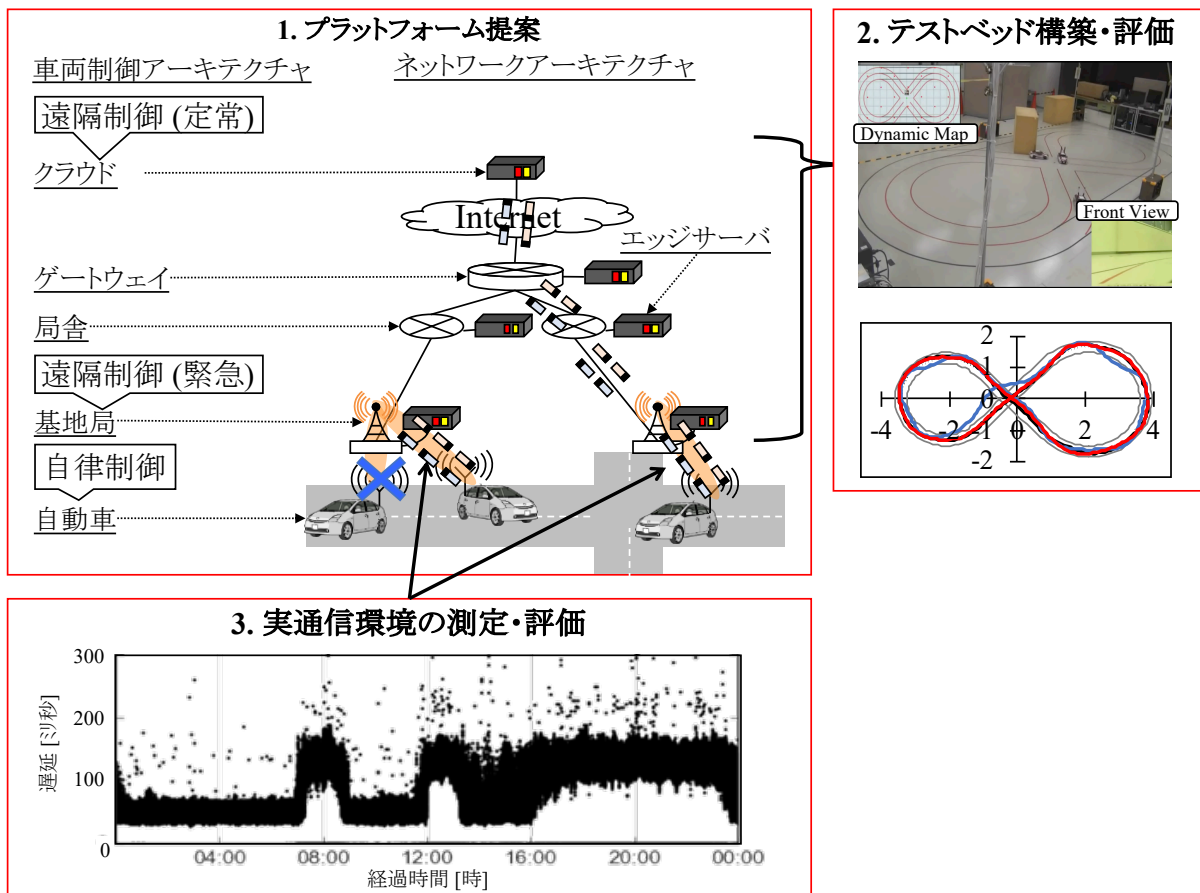


図 1.1: インフラ型車両制御プラットフォーム

と比較してエッジサーバ数が少なく、収容台数が大きいものの、ネットワーク障害の影響を受けやすい。

本論文では、提案プラットフォームを適応する環境に 5G/MEC を想定しており、UpES/LoES の配置先は、「基地局」, 「局舎」, 「ゲートウェイ⁶」, 「クラウド」の間で調整できる。UpES/LoES の配置は「車両-エッジサーバ間の通信遅延」, 「収容台数」, 「エッジサーバ数」に影響を与え、これらの値は統計情報や実通信測定の結果から導く。

これら 2 層のエッジサーバによる遠隔制御と自動車による自律制御が通信状態に基づいて車両制御を切り換えるプラットフォームを構築する。

⁶LTE における Packet data network gateway を想定。

1.7.2 テストベッドの構築・評価

提案プラットフォームを評価するために、本論文では、複数のエッジサーバからマイクロカーの遠隔制御を実現できるテストベッドを構築 [22, 25, 26, 24] する。また、構築したテストベッドを用いて、提案プラットフォームによる車両制御の評価を行う。

テストベッドを用いた評価は「基本特性評価」と「実通信特性評価」の2種類を行う。基本特性評価では、クラウドとマイクロカーの間に基本的なネットワーク障害を加えて、クラウドから車両の走行制御を行う。ネットワーク障害と走行軌跡の変化から協調運転システムが成立する通信の閾値（以降、切換閾値）を導く。実通信特性評価では、導いた切換閾値と実計測した通信遅延データを用いて、走行制御精度の評価を行い、通信の状態に応じて UpES, LoES 間制御を切り換えることによって、車両制御の精度を下げることなく遠隔制御を実現できることを示す。

1.7.3 実通信環境の測定・評価

実通信環境を考慮した評価を行うために、3つのクラウドサービスに対して、11のキャリアの通信端末で実遅延データを測定し、パケットロスと通信遅延について評価する [21, 24, 23]。

通信データが計測できたら、それらを用いて、提案プラットフォームの車両制御シミュレーションを行う。シミュレーションでは、UpES, LoES の配置を変更することで遅延を変化させて、UpES・LoES による遠隔制御、自動車の自律制御の制御比率（各制御手法が車両制御を担当した時間の割合）を導く。また、UpES, LoES の配置を変更し、収容台数と制御比率、エッジサーバ数の関係について評価する。更に、UpES, LoES の最適配置を考察し、エッジサーバ数を抑制しつつ、最大の収容台数と低遅延を両立する提案プラットフォームを実現できることを示す。

1.8 貢献

本博士論文における最大の貢献は「通信遅延/収容台数/エッジサーバ数のトレードオフを解決可能な多層エッジサーバを用いた車両制御プラットフォームの確立」である。

主要な貢献を以下に示す。

- エッジサーバを多層に配置するネットワークアーキテクチャの提案 (3 章)

- 提案ネットワークアーキテクチャで動作する車両制御アーキテクチャの提案および、協調運転システムへの応用 (3 章)
- 協調運転システムが安定動作するための通信遅延・連続パケットロスに関する閾値の導出 (4 章)
- テストベッドを用いた実通信環境化での遠隔制御の安定動作の確認 (4 章)
- 車両制御を目的とした日本国内の通信環境の網羅的な測定・評価 (5 章)
- 網羅的に計測した遅延データを用いた国内通信環境化での制御比率の導出 (6 章)
- エッジサーバの配置変更による、制御比率と収容台数および、エッジサーバ数の間にあるトレードオフの解決 (6 章)
- 最適なエッジサーバ配置の提案 (6 章)

3 章の中で、協調運転実現のために車両制御アーキテクチャ/ネットワークアーキテクチャの提案を行う。エッジサーバを多層に配置するネットワークアーキテクチャおよび、多層のエッジサーバから遠隔制御を行う車両制御アーキテクチャは本論文が最初の提案であり、2018 年時点では本論文以外で確認されていない。

4 章の中では、テストベッドを用いた提案プラットフォームの通信遅延に対する基本特性評価により、遠隔制御が安定動作するための「車両-エッジサーバ間遅延」と「制御データの連続パケットロス数」に関する閾値を決定する。遠隔制御を行うエッジサーバと車両の間の遅延が閾値を超えた場合、下位のエッジサーバによる遠隔制御もしくは車両による自律制御に切り換えることで安定した車両制御を実現できることを示す。また、パケットロスに対する評価から、制御データの連続パケットロスが閾値を超えたタイミングで遠隔制御を自律制御に切り換えることで安定制御を実現できることを示す。最後に、クラウドに配置された UpES と基地局に配置された LoES の間に実通信遅延を加えて、導いた閾値を用いて切換制御を行うことで、遅延の影響を受けることなく安定して遠隔制御が実現できることを示す。

5 章では、国内キャリアの測定から、「MNO の 3 社はパケットロスを発生させにくい安定したネットワークを持つこと」、「キャリアによっては通信負荷が集中する時間帯で、大きく遅延が増加すること」を示す。

6章では計測した遅延を用いたシミュレーション評価を行う。シミュレーションの結果、以下の4つを示す。第一に制御比率について、平均的な通信経路では、「UpES を最大の収容台数を持つクラウドに配置しても90%以上の時間帯で制御が可能であること」、「LoES を局舎・基地局に配置することで自律制御の制御比率を0.6%にまで抑制できること」から超広範囲の協調運転システムの実現可能性を示す。第二に、エッジサーバの配置について「UpES の配置を調整することでLoES-UpES 間の制御比率と収容台数のトレードオフを調整できること」、「LoES の配置を調整することで遠隔制御と自律制御の制御比率とエッジサーバ数のトレードオフを調整できること」を示す。第三に、自律制御の必要性について「UpES, LoES の配置を調整しても、突発的なバーストパケットロスには抑制できないため、自律制御が必須であること」を示す。最後に、UpES と LoES の最適配置について「UpES をクラウドに、LoES を局舎に配置することで、エッジサーバ配置のコストを抑制しつつ、最大の収容台数と低遅延を両立する提案プラットフォームを実現できること」を示す。

以上より、超広範囲の協調運転システムを実現可能なプラットフォームを提案し、テストベッドによる評価から、提案プラットフォームが安全に動作する条件を確立する。また、シミュレーションによる評価から、提案プラットフォームを安全に動作する条件を満たしつつ、多くの時間帯で最大の収容台数を持つエッジサーバから遠隔制御できることを示すことで、超広範囲の協調運転システムの実現可能性を示す。

1.9 本論文の構成

本論文は全7章で構成される。2章では、車両の遠隔制御に関連する研究の調査結果をまとめる。3章では、提案するインフラ型車両制御プラットフォームについて述べる。4章では、提案プラットフォームを評価するために構築したテストベッド環境について述べる。5章では、提案プラットフォーム評価のために実トラフィックの計測を行い、6章にて、その計測データを用いたシミュレーション評価を行う。7章にて本博士論文のまとめを行う。

第 2 章

関連研究

2.1 はじめに

本論文は, 多層にエッジサーバを配置するネットワークアーキテクチャと多層エッジサーバによる遠隔制御を提案した最初の論文である. その一方で, クラウドもしくはエッジサーバと自動車を連携させる研究は既に行われているため, 本章ではそれらを関連研究としてまとめる.

自動車とクラウドの連携に関する研究では, ライドシェアや交通予測, 車両の遠隔診断など比較的にリアルタイム性の低い研究が多い. また, 数は少ないながらクラウド型遠隔制御に関する研究も行われており, それらの研究では通信遅延の影響からクラウド型遠隔制御は困難であることが示されている.

自動車とエッジサーバの連携に関する研究では, MEC の局所性を生かすために, 基地局相当の車から最も近いネットワーク装置に配置されたエッジサーバとクラウドが連携するシステムを構築する. 我々の提案プラットフォームとは異なり, エッジサーバの配置は固定であり, 様々な配置の検証は行われていない. また, 低遅延性を生かした MEC 型遠隔制御の研究も存在するが, クラウドからの制御は考慮しておらず, 我々の目標とする広範囲の協調運転は実現できない構成となっている.

本章は全 6 節で構成されている. 2.2 節では自動車とクラウドの連携に関する研究を, 2.3 節ではクラウド型自動運転に関する研究を紹介する. 続けて, 2.4 節では自動車とエッジサーバの連携に関する研究を, 2.5 節では MEC 型自動運転に関する研究を紹介する. 最後に, 2.6 節でまとめる.

表 2.1: VCC アプリケーションと情報の寿命

	アプリケーション タイプ	コンテンツ タイプ	情報の 寿命
ライドシェア [29, 30]	エンタテイメント	マルチメディア情報共有	10 分
交通予測 [31, 32]	運転改善	渋滞情報	30 分
		ナビゲーション	30 分
災害時交通 [33]	アクティブセーフティ	危険道路警告	10 秒
		工事区域警告	復旧まで
遠隔診断 [34]	公共サービス	危険車両警告	10 分

2.2 自動車とクラウドの連携に関する研究

自動車とクラウドの連携によって、交通安全やインフォテインメントを実現する Vehicular Cloud Computing(VCC)[27, 28] については様々な研究が行われている。

Dimitrijevic 等 [29], Amey 等 [30] は VCC を用いたライドシェア¹に関する研究を行っている。車両の現在位置と目的地, 歩行者の目的地をクラウドで収集し, 車両と歩行者のマッチングを行う。双方にリアルタイムな情報を提供することで効率的なライドシェアを実現する。

Ancona 等 [31], Stanica 等 [32] は VCC を用いた交通予測に関する研究を行っている。車両は位置や速度といった GPS から得られる情報を周期的にクラウドに対してアップロードし, クラウドは収集した情報を解析し, 渋滞情報等の交通情報を車両に対して提供する。交通情報の提供にはクラウドから移動体通信網を経由して直接通信を行う場合と, V2X を介して間接的にクラウドから情報を受け取る場合の 2 パターンが存在する。

Zubaida 等 [33] はクラウドを用いた災害時交通の管理に関する研究を行っている。V2X を用いて通過できた道の情報を収集, クラウド上に反映し, 各車両の走行経路を決定する。

Lionel 等 [34] はクラウドを用いた車両の遠隔診断に関する研究を行っている。各車両は Car Area Network² (CAN) 上の各装置の診断メッセージであるダイアグメッセージをクラウドに送信する。クラウドでは収集したダイアグメッセージを用いて異常検知を行い, 車両に異常が発見されたら, その車両に対して警告メッセージを送信する。

表 2.1 に上記アプリケーションとそれぞれで取り扱う情報の寿命をまとめる。各アプリケーション

¹WEB サービスを通じて, 目的地が同じ複数人が 1 台の乗用車に同乗するシステム

²車両内の電子部品同士の通信で使用されるネットワーク

ンのタイプ, コンテンツタイプ, 情報の寿命に関しては [35] 中で定義されている. 情報の寿命は許容できる遅延の閾値と考えることもできる. Lee 等 [36] によると V2X が対象とする協調運転や安全関係のサービスの遅延要件は全て 100ms 以下である. それと比較して表 2.1 を見ると, VCC で対象となるアプリケーションは, ある程度遅延を許容することができるアプリケーションが多いことが分かる.

2.3 クラウド型遠隔制御に関する研究

数は少ないもののクラウド型遠隔制御システムも提案されている. これらでは, クラウドがセンサ情報を広範に存在する車両から収集し, センサ情報の解析を行い, 車両に対して制御情報の提供を行う. 広範囲の車両を同時に制御できる一方で, 車両-クラウド間の通信遅延が大きな課題となる.

Suzuki 等 [17] はクラウド上のデータ管理機構に複数の車両の位置・速度等のデータを集約し, クラウド上のアプリケーションがそのデータ管理機構の情報を用いて車群を集中制御する「クラウド型自動運転」を提案している. Local Dynami Map³[37] (以降, LDM) を用いたデータ管理機構を構築し, シミュレーションによって 1000 台規模の車両に対して衝突制御を実現している. ただし, シミュレーションの中では通信遅延を考慮していない.

Kumar 等 [38] はクラウド上に高精度の地図データを収集し, クラウドから車両制御を支援するシステムの提案を行っている. しかし, この中ではネットワークの遅延について考慮されておらず, 実車を用いた評価はローカルネットワーク上で行われている.

Lei 等 [18], Ruilin 等 [39] は LTE 環境を介してドライバーが車両の遠隔操作を行うシステムを提案している. これらのシステムでは, 車両は設置されたカメラから映像を取得しクラウドに対して送信し, クラウドから制御情報を受信する. クラウド側にはドライバーが存在し, ドライバーは映像を見ながら, ハンドルやアクセルを操作することで車両の制御を実現する. これらの研究では, LTE の通信遅延が遠隔操作システムの実現を妨げることを示している. このことは, 移動体通信網の遅延が大きい時には, クラウドからの遠隔制御が成立しないことを示す.

以上より, クラウドと車両に関する研究は, リアルタイム性の低いアプリケーションに焦点を当

³地形や道路のような静的情報に加えて, 車両の現在位置や車速, 渋滞情報のような動的情報の管理も行う地図

てていることが多く、クラウド型遠隔制御は移動体通信網の遅延が大きい時には成立しないことを確認した。

2.4 自動車とエッジサーバの連携に関する研究

ETSI のホワイトペーパー [14] の中で、MEC の重要な応用事例として自動車の自動運転が示されていることから、車と MEC の連携は注目されており、すでに様々な研究が行われている [40, 41]。

Zhang 等 [42] は、クラウドベースの V2X アプリケーションの処理をエッジサーバへオフロードするフレームワークに関する研究を行っている。クラウドからエッジサーバへの処理のオフロードの研究は多く提案されているが、車の場合、同一のエッジサーバにアクセスできる時間が短いため、移動速度とエッジサーバのリソースを考慮して、クラウドとエッジサーバの間で処理の割当を行う。

Tang 等 [43] はエッジコンピューティングベースのスマート駐車システムの提案を行っている。クラウドとエッジサーバ⁴の 2 層アーキテクチャで構成され、エッジサーバは局所情報の処理を担当し、各地域で車両から駐車要求が送信された際に、車両に対して駐車場の割当を行う。クラウドは全体情報の管理を担当し、エッジサーバから送られてくる情報に基づいて駐車場全体の利用率を管理する。駐車場が全て利用されている地域で、車から駐車要求が送信された場合には、クラウドから車両に対して最適な駐車場への経路を提案する。

Shinkuma 等 [44] はエッジサーバとクラウドを連携させてリアルタイムな渋滞予測を行うシステムを提案している。システムの基本構成は、車両のセンサ情報をクラウドで収集し、クラウドは解析結果である交通渋滞の結果を返す。しかし、エッジサーバとクラウド間の通信で大きな遅延が発生する場合には、エッジサーバ上でデータ解析を行い、車両とクラウド双方に解析結果を返すシステムとなっている。

以上のシステムでは、計算装置の配置がクラウドと車両から最寄りのネットワーク装置に固定されている。それに対して我々が提案する多層エッジサーバアーキテクチャでは、クラウドと基地局以外の様々な配置を検討する。また、これらの研究は車両制御に焦点を当てておらず、エンターテイメントサービスや危険警告等のアプリケーションに限られている。

⁴論文中では fog node と表記

2.5 MEC 型遠隔制御に関する研究

数は少ないものの、MEC 型遠隔制御に関する研究も存在する。

Ito 等 [19] は基地局に配置されたエッジサーバから車両制御を行う手法を提案しシミュレーションによる評価を行っている。この中では局所制御のみを対象としており、1つのエッジサーバから衝突回避を行うシーンのみの検証を行っている。

Nokia[20] は 5G 回線を利用した遠隔制御のデモを行った。基地局間のハンドオーバに焦点を当てており、基地局に配置されたエッジサーバから車両を遠隔制御することが想定されている。

これらの MEC 型遠隔制御では、クラウド型遠隔制御と比較して車両に近い位置から情報処理を実現できるため、低遅延な車両制御を実現できる。しかし、クラウド型遠隔制御のように広範囲から車両のセンサ情報を収集することはできない。また、収容台数やエッジサーバ数に影響を与えるエッジサーバの配置を考慮していない。超広範囲の協調運転システム実現のためには、エッジサーバを可能な限り広範囲のセンサ情報を収集できるネットワーク装置に配置し、広範囲の車両状態を把握したうえで、車両制御を行うことが望ましい。

2.6 まとめ

現在提案されている遠隔制御システムは、クラウド型遠隔制御システムもしくは MEC 型遠隔制御システムであり、それらが連携して遠隔制御を行うシステムは存在していない。超広範囲の協調運転を実現するとき、クラウド型遠隔制御では、車両ークラウド間の通信遅延が大きな問題となる。一方で、MEC 型遠隔制御では、収容台数が大きな問題となる。

我々の提案プラットフォームでは多層のエッジサーバによる遠隔制御と自動車の自律制御を組み合わせることで、可能な限り上位に存在する収容台数の大きい装置から制御を行い、通信環境に応じて下位の装置に切り換えることで超広範囲の協調運転システムを実現する。

次章では、超広範囲の協調運転システムを実現するためのインフラ型車両制御プラットフォームについて提案する。

第 3 章

インフラ型車両制御プラットフォーム

3.1 はじめに

本章では, 超広範囲の協調運転システムを実現するために, 多層エッジサーバと自動車が連携して車両制御を実現するインフラ型車両制御プラットフォームを提案する.

提案プラットフォームは多層にエッジサーバを配置する「ネットワークアーキテクチャ」と, その多層エッジサーバによる遠隔制御と自動車による自律制御を連携させた「車両制御アーキテクチャ」によって構成される. 多層に配置されるエッジサーバは, 車両から離れた位置に存在するエッジサーバほど大きな収容台数を持ち, 広範囲のセンサ情報を収集可能であるものの, 通信環境が不安定である. その一方で, 車両から近い位置に存在するエッジサーバほど通信遅延が小さく, 通信環境は安定するものの, 収容台数が小さい. 本論文のネットワークアーキテクチャは, 車両から離れた位置に存在する Upper Edge Server (UpES) と車両の近隣に存在する Lower Edge Server (LoES) の 2 層のエッジサーバによって構成される. また, 車両は車両-LoES 間の通信環境が不安定な場合, 自律制御を行う. 車両の自律制御は, 通信を行うことなく, 車両に取り付けられたセンサ情報のみを使用して制御を行う. このことから, 自律制御は通信遅延 0 秒, 収容台数 1 台の遠隔制御と考えることができ, 自動車も一番下位のエッジサーバと考えることができる. これらの多層エッジサーバと自動車が連携して車両の制御を行うプラットフォームを提案する.

また, 2 層のエッジサーバによる遠隔制御と自動車の自律制御を, 通信の状況に応じて切り換える手法も提案する. 超広範囲の協調運転システムを実現するためには, 広範囲のセンサ情報を収集し, それらに基づいて車両制御を行う必要があるため, 可能な限り上位のエッジサーバから制御を

行う必要がある。言い換えれば、可能な限り車両（最下位のエッジサーバ）の自律制御は避ける必要がある。提案する切手法では、可能な限り上位のエッジサーバから制御を行い、通信が不安定な状況でのみ下位のエッジサーバへの切換を実行する。

最後に、エッジサーバを配置するネットワークトポロジについて検討を行う。本論文ではエッジサーバは移動体通信網内のネットワーク装置に配置されることを想定している。配置先の候補としては「基地局」、「局舎」、「ゲートウェイ」、「クラウド¹」が考えられる。各ネットワーク装置ではエッジサーバ数、収容台数および、車両からの通信時間が異なるため、エッジサーバ数と収容台数を国内の統計情報から見積もり、車両-ネットワーク装置間の通信時間を遅延モデルから導く。本章の主要な貢献は、以下である。

- エッジサーバを多層に配置するネットワークアーキテクチャの提案
 - ネットワーク装置の配置変更による、エッジサーバ数、収容台数および、通信遅延の導出
- 提案ネットワークアーキテクチャで動作する車両制御アーキテクチャの提案および、協調運転システムへの応用
 - 車両制御の切手法の提案

本章は全7節で構成されている。3.2節にて提案プラットフォームのアーキテクチャについて説明する。3.3節では提案プラットフォームの動作について説明し、3.4節では車両制御アルゴリズムを、3.5節では車両制御の切手法を示す。その後、3.6節にて提案プラットフォームを配置するネットワークトポロジについて説明し、3.7節にてまとめる。

3.2 アーキテクチャ

図 3.1 に提案プラットフォームのアーキテクチャを示す。提案プラットフォームのアーキテクチャはネットワークアーキテクチャと車両制御アーキテクチャによって構成され、ネットワークアーキテクチャは2層のエッジサーバと車両で構成される。2層のエッジサーバは UpES と LoES であり、「車両-LoES」と「LoES-UpES」がいくつかのネットワーク装置を経由して接続されており、車両-LoES 間には無線通信が存在する。本論文は移動体通信を想定しているため、無線通信

¹本論文では、移動体通信網の最上位のネットワーク装置としてクラウドを想定している

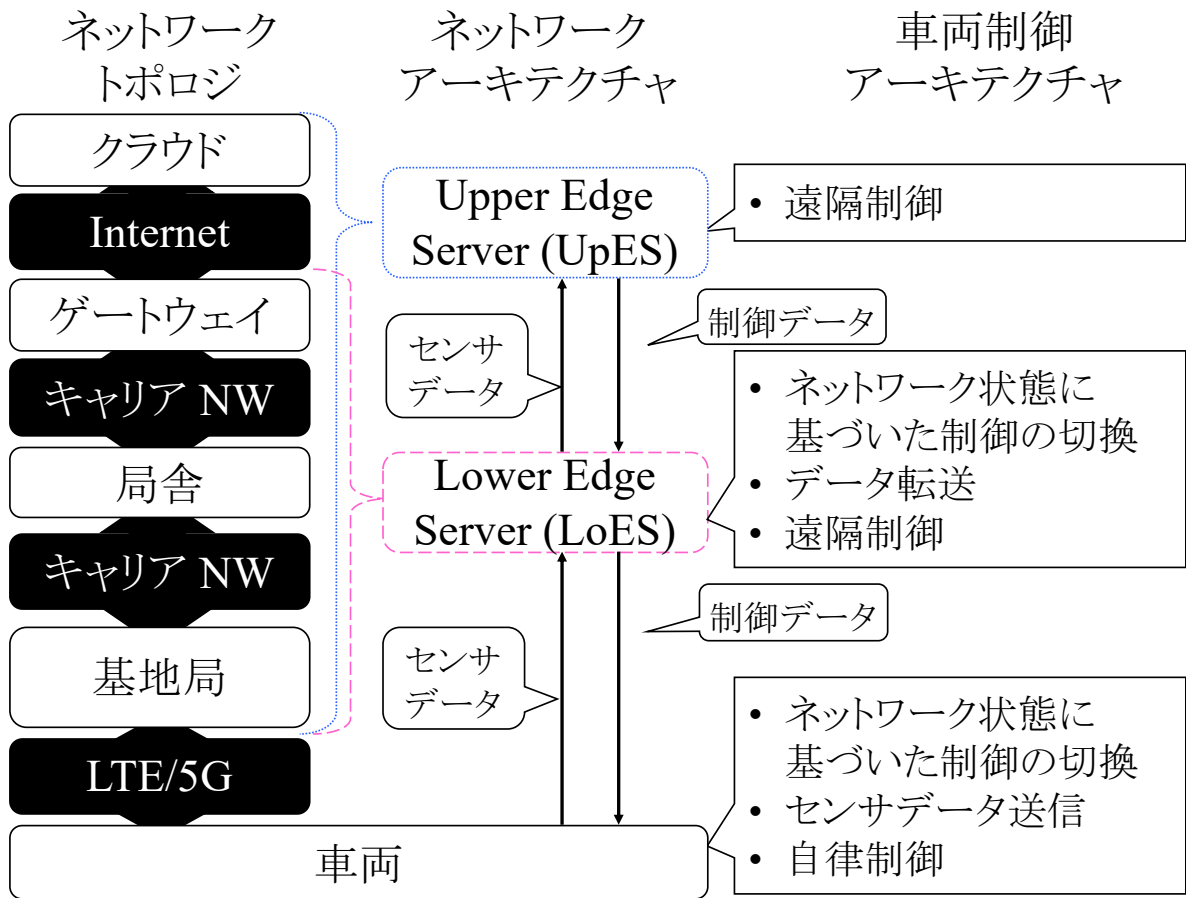


図 3.1: 提案プラットフォームのアーキテクチャ

には 5G を想定している. UpES と LoES はクラウド, ゲートウェイ, 局舎, 基地局のいずれかに配置され, UpES は LoES より上位のネットワーク装置に配置される.

車両はセンサデータを多層エッジサーバに対して送信し, 制御データを受信することで遠隔制御を行う. その一方で, 車両はネットワークの状態も同時に監視し, 通信障害を検知したら制御の切替を行い, 自律制御へ移行する. LoES は車両と UpES から送信されるセンサデータ/制御データを転送する一方で, ネットワークの状態も監視し, 通信障害を検知したら LoES が遠隔制御を行い, 車両に対して制御データを送信する. UpES は車両からセンサデータを受信し, 制御データを送信する.

以下で, UpES, LoES, 車両による制御について詳細な説明を行う.

3.2.1 UpES による遠隔制御

UpES は LoES より上位のネットワーク装置から遠隔制御を行うエッジサーバである。クラウドやゲートウェイのようなネットワークを集約する装置上に配置されることを想定しているため、UpES は大きな収容台数を持ち、広範囲のセンサデータを生かした遠隔制御を実現できる。その一方で、UpES では、遠隔制御が成立しなくなるような大きな通信遅延やバーストパケットロス²が発生する可能性がある。このような、UpES からの遠隔制御が成立しなくなる状況では、LoES による遠隔制御や自動車による自律制御に移行する。

本提案プラットフォームでは、UpES の配置を基地局、局舎、ゲートウェイ、クラウドのいずれかと考えており、上位のネットワーク装置に配置するほど、収容台数と共に通信遅延も増加する。

3.2.2 LoES による遠隔制御

LoES-UpES 間の通信環境が不安定である場合には、UpES より下位のエッジサーバである LoES から遠隔制御を行う。UpES と比較して、LoES は収容台数が小さくなるものの、低遅延で安定した遠隔制御を実現できる。

LoES-UpES 間の通信環境が安定している場合、LoES はセンサデータおよび、制御データの転送を行う。データの転送と同時に、LoES-UpES 間通信の監視も行い、通信環境が不安定になった場合には、UpES から LoES へ遠隔制御の切替を行い、LoES が遠隔制御を実行する。

本論文のシステムでは、LoES の配置を基地局、局舎、ゲートウェイのいずれかと考えており、下位の装置に配置するほど、通信遅延と共に収容台数が減少する。

LoES の必要性について 本提案アーキテクチャでは、自動車が自律制御による自動運転を実行できることを仮定しているため、通信が安定しているときは UpES から制御し、非常時のみ自律制御を行うことで、超広範囲の協調運転が実現できる可能性がある。

しかし、デッドロックを防ぎスムーズな交通を実現するためには、可能な限り複数台の車両情報を使用して制御を行う必要がある。UpES と自動車の 2 層による車両制御アーキテクチャでは、UpES を車両から遠い位置に配置すると、自律制御による制御の時間が長くなる。長時間の自律制御は、デッドロックの発生確率を増加させ、協調運転が成立しなくなる可能性を増加させる。それ

²連続したパケットロス

に対して、LoES を自動車-UpES 間に配置する場合、通信が安定しているときは UpES が広範囲のセンサ情報を用いて遠隔制御し、通信が不安定になっても LoES に移行して遠隔制御を持続することで、可能な限り複数台の車両情報を使用した制御を実現できるので、デッドロックの発生確率を最小限に抑制することができる。

本件については、6章で詳細に検証する。

3.2.3 自動車による自律制御

遠隔制御は無線通信の突発的な遮断や大きな通信遅延によって、突然成立しなくなる可能性がある。そのため、情報収集から制御に至るまでの間に通信が存在せず安定した制御が期待できる自律制御は、協調運転システムにおいて必須の技術である。

提案プラットフォームでは、車両は自身に取り付けられたセンサで収集したセンサデータを周期的にエッジサーバに対して送信し、エッジサーバから制御データを受信することで遠隔制御を実行する。その一方で、常に通信状態を監視し、通信が不安定になったら制御の切換を行い、自車センサで収集した情報に基づいて自律制御を実行する。

自律制御は、車両が1台である場合、通信障害が存在しないため安定した制御が期待できる。しかし、自律制御を行う車両同士の協調運転では車両間の調停に起因するデッドロックが発生する可能性がある。そのため、協調運転を実現するためには複数車両のセンサデータを収集する遠隔制御が必要になる。

3.3 提案プラットフォームの動作

図 3.2 に提案プラットフォームの動作の様子を示す。遠隔制御を行う場合、車両から LoES に対して「センサデータ」が送信されると、UpES もしくは LoES がそれを処理し、車両に対して「制御データ」を返信する。ここで、センサデータは「車両 ID」と「位置」、「向き」および、「速度」で構成される車両の状態を表すデータで、制御データは「車両 ID」と「ステアリング量」、「アクセル量」および、「ブレーキ量」で構成される車両制御を行うデータである。車両は制御データを受信すると、それに基づいて、ハンドル・アクセル・ブレーキの調整を行う。車両は周期的にセンサデータを送信し、車両が制御データを受信することで遠隔制御を実現する。

その一方で、車両はセンサデータと制御データの送受信から、車両-LoES 間遅延 (l_{V2L}) と連続パ

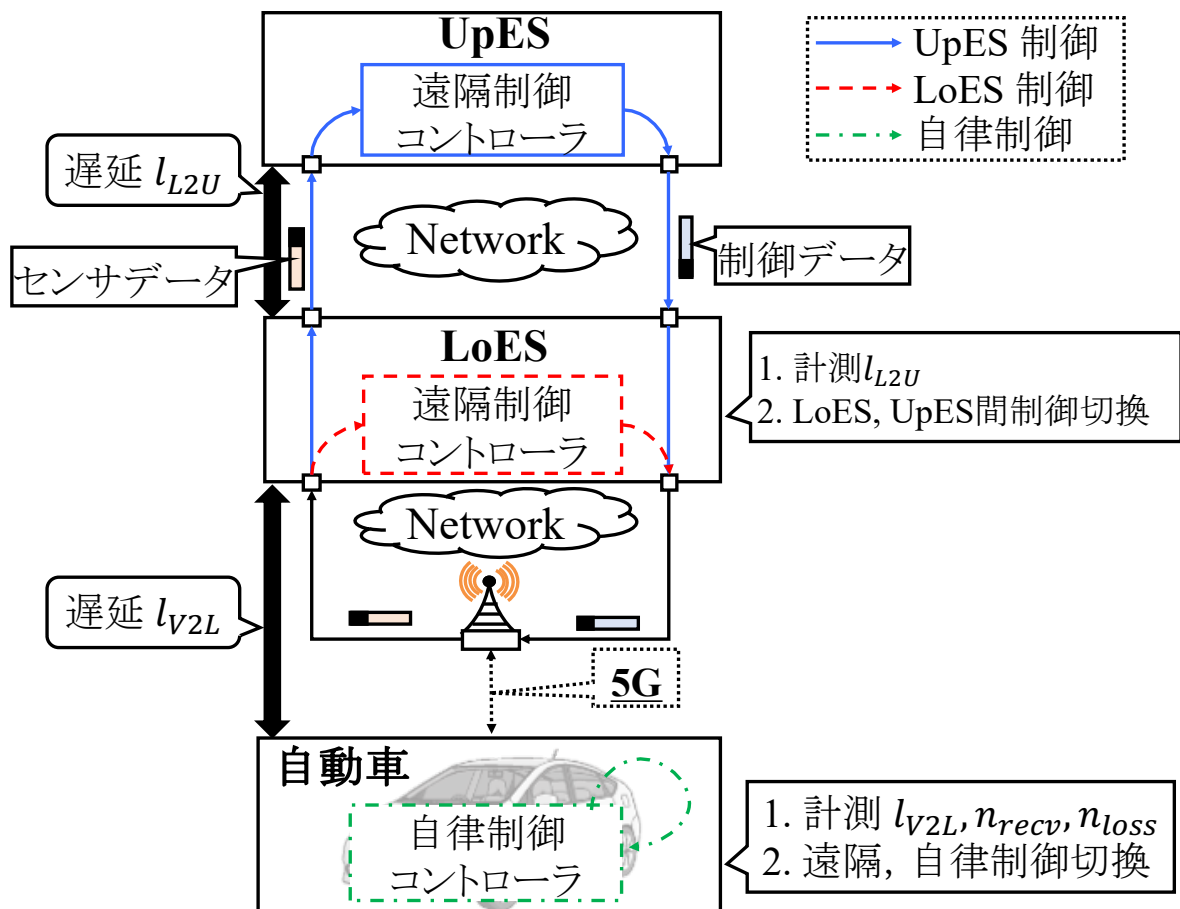


図 3.2: 提案システム

ケット受信数 (n_{recv}) と連続パケットロス数 (n_{loss}) を計測しており、それらに基づいて遠隔制御と自律制御の切換を行う。また LoES も LoES-UpES 間遅延 (l_{L2U}) と車両-LoES 間遅延 (l_{V2L}) に基づいて UpES/LoES 間の制御の切換を行っている。

3.4 車両制御アルゴリズムについて

本論文では、遠隔制御および、自律制御のアルゴリズムに Proportional-Integral-Differential 制御 (PID 制御) アルゴリズムを用いる [45].

PID 制御は、入力値の制御を出力値と目標値との偏差、その偏差の積分および、その偏差の微分の 3 つの要素によって行う方法で、出力値を素早く安定的に目標値に近づけることができる。出力

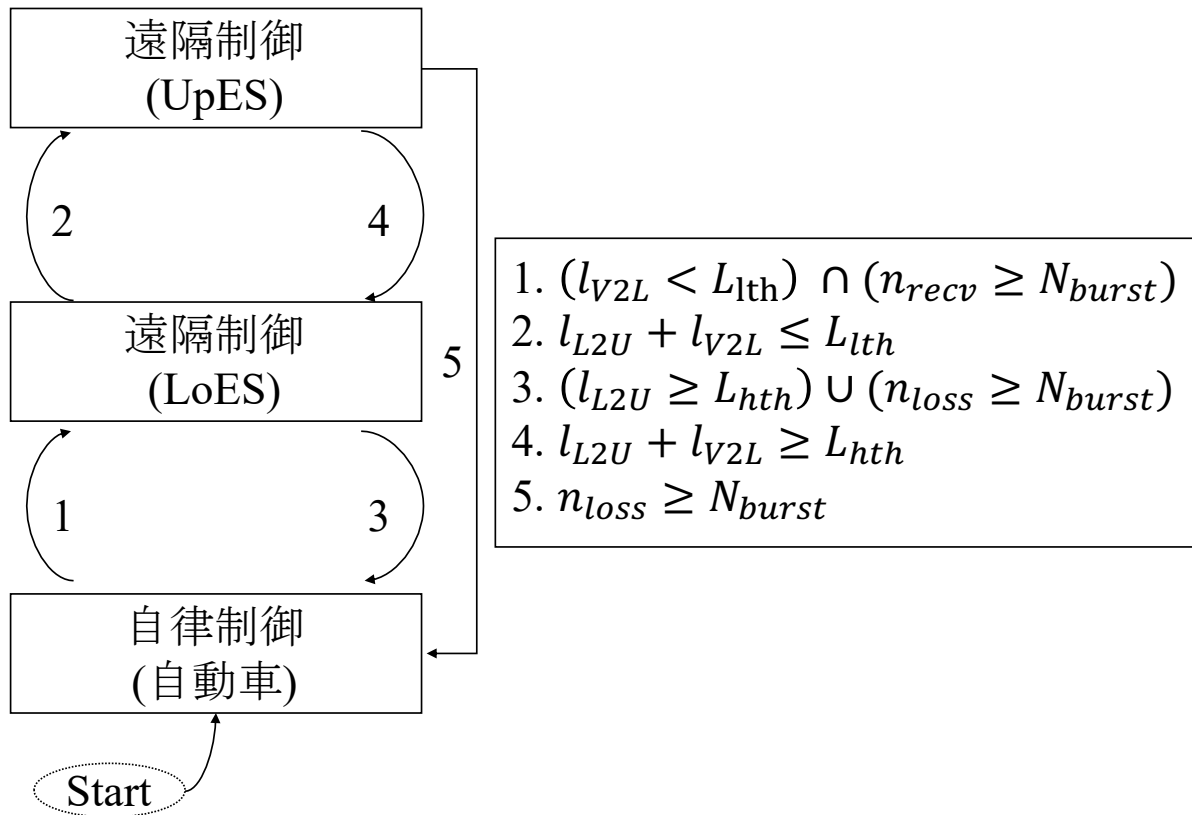


図 3.3: 状態遷移図

を $y(t)$, 目標値を $r(t)$, 偏差を $e(t)$ とすると入力値 $u(t)$ の更新式は

$$e(t) = r(t) - y(t) \quad (3.1)$$

$$u(t+1) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (3.2)$$

となる。このとき, K_p, K_i, K_d はそれぞれ比例項, 積分項, 微分項の重みである。

本論文の PID 制御では, UpES/LoES 上で動作し, 車両からセンサデータを受信すると, エッジサーバ内の車両制御機能に転送する。車両制御機能では, 現在の車両の「スピード」と「向き」を出力, 「アクセル/ブレーキ」と「舵角」を入力として, 上記更新式を用いて, 制御データを生成し, 車両へ送信する。

3.5 車両制御の切手法

図 3.3 に提案プラットフォームによる車両制御切換の状態遷移図を示す. 3つの閾値はそれぞれ

L_{hth} ネットワーク上位の制御装置が下位の制御装置に切り換える通信遅延閾値

L_{lth} ネットワーク下位の制御装置が上位の制御装置に切り換える通信遅延閾値

N_{burst} 遠隔制御から自律制御に切り換えるためのバーストパケットロス閾値

である.

最初, 制御は自律制御から開始する. 連続パケット受信数 (n_{recv}) が切換閾値 N_{burst} より多く, パケットロスなく連続で受信し, 車両-LoES 間遅延 (l_{V2L}) が切換閾値 (L_{lth}) 以下ならば, 安定して遠隔制御できると判断し, LoES による遠隔制御に遷移する (図 3.3 の 1).

LoES が遠隔制御をする場合, LoES-UpES 間遅延 (l_{L2U}) と車両-LoES 間遅延 (l_{V2L}) の合計が切換閾値 (L_{lth}) 以下ならば, 安定した低遅延を持つと判断し, UpES に制御を遷移する (図 3.3 の 2). その一方で, LoES 制御中に, 連続したパケットロス数 (n_{loss}) が切換閾値 N_{burst} より多く発生する, もしくは, 車両-LoES 間遅延 (l_{V2L}) が切換閾値 L_{hth} を超える場合, 安定した遠隔制御は実現できないと判断し, 車両による自律制御に遷移する (図 3.3 の 3).

UpES が遠隔制御する場合, LoES-UpES 間遅延 (l_{L2U}) と車両-LoES 間遅延 (l_{V2L}) の合計が切換閾値 (L_{hth}) 以上ならば, 不安定な遅延を持つと判断し LoES に制御を遷移する. (図 3.3 の 4). また, UpES 制御中に, 連続したパケットロス (n_{loss}) が切換閾値 N_{burst} より多く発生する場合は, 安定した遠隔制御は実現できないと判断し, 車両による自律制御に遷移する (図 3.3 の 5).

図 3.3 の状態遷移を用いて安定した車両制御を実現するためには, 3つの切換閾値 L_{hth} , L_{lth} , N_{burst} を適切に決定する必要がある. これらの値は 4 章のプロトタイプを用いた評価によって決定され, それぞれ, $L_{hth} = 100$, $L_{lth} = 70$, $N_{burst} = 3$ である.

次節では, 提案する車両制御アーキテクチャが動作するネットワークトポロジについて述べる.

3.6 提案プラットフォームを配置するネットワークトポロジ

図 3.4 は提案プラットフォームが動作するネットワークトポロジを示す.

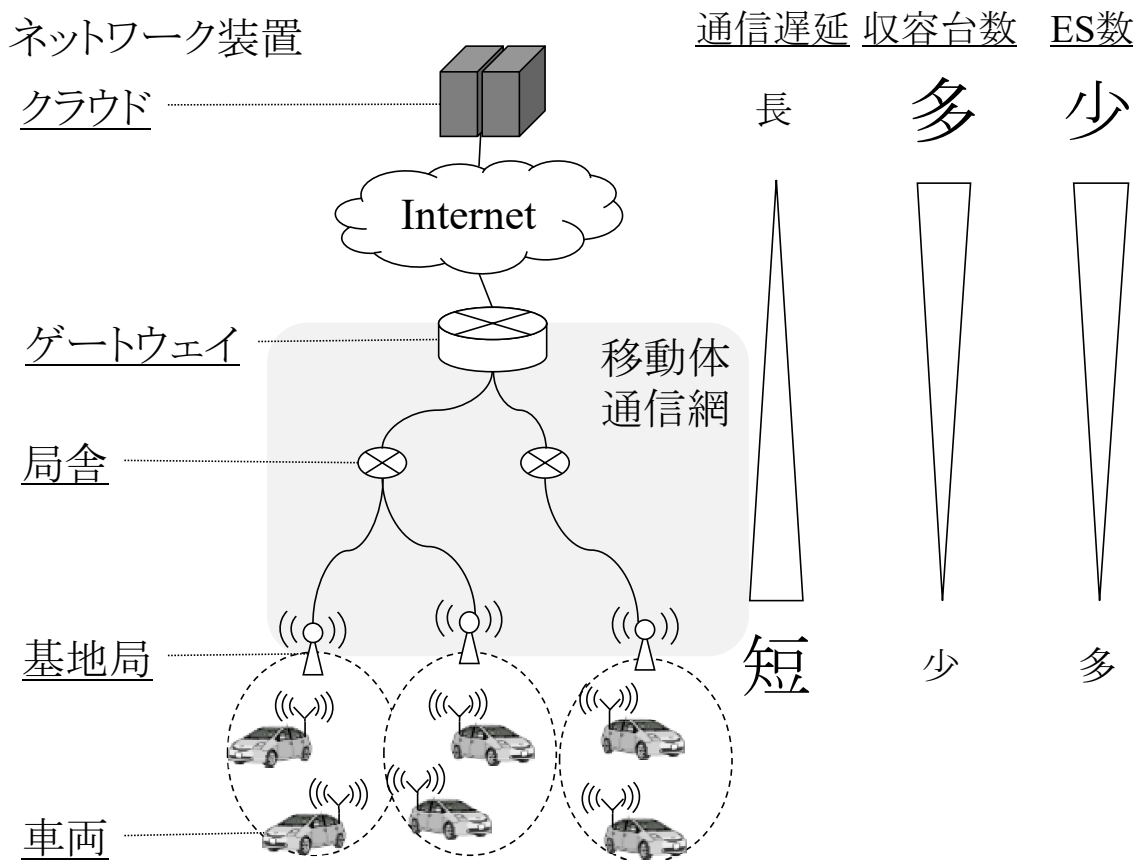


図 3.4: 提案プラットフォームが動作するネットワークトポロジ

本論文のネットワークは図 3.4 中の 5 つの要素, 「車両」, 「基地局」, 「局舎」, 「ゲートウェイ」, 「クラウド」によって構成され, 車両を除いた 4 層のネットワーク装置のいずれか 2 層に UpES/LoES が配置される. UpES と LoES の配置が変更されると「収容台数」と「エッジサーバ数 (図 3.4 中では ES 数と表記)」および, 「通信遅延」に変化が生じ, エッジサーバが上位のネットワーク装置に配置されるほど, 収容台数が増加し, 必要になるエッジサーバの数が減少する一方で, 遅延の大きさが増加する.

車両は 5G の無線通信を用いて基地局と接続される. 基地局は局舎と, 局舎はゲートウェイと接続しており, 基地局-ゲートウェイ間のネットワークがキャリアが管理する移動体通信網となる. この移動体通信網がインターネットを経由してクラウドと接続される.

以降, エッジサーバの配置変更に影響を与える「エッジサーバ数」, 「収容台数」および, 「通信遅延」について説明する.

3.6.1 エッジサーバ数/収容台数

本論文では, 各ネットワーク装置のエッジサーバ数および, 収容台数を日本の環境に基づいて見積もりを行う. 表 3.2 に国内のエッジサーバ数と収容台数の見積もり結果を示す.

表 3.1: エッジサーバ数と収容台数

ネットワーク装置	エッジサーバ数	収容台数
クラウド	1	3,415,090
ゲートウェイ	55	62,093
局舎	5,640	606
基地局	51,076	67

基地局の数は 2016 年 9 月の LTE の eNodeB の数に基づき 51,076 と仮定する [46]. 局舎の数に関する情報は日本国内で開示されている統計情報が存在しなかったため, アメリカの AT&T 社が公開する局舎数情報 [47] を AT&T の米国内シェアと日米の人口比率に基づいて変換し, 5,640 とする. ゲートウェイの数は国内の MNO と MVNO の数に基づき決定する. 日本国内には主要な MNO が 3 つ (NTT ドコモ (docomo)[48], Softbank[49], KDDI[50]) と 2015 年時点で 30,000 人以上のユーザ数を持つ MVNO が 25 社存在する [51]. MNO が 10, MVNO が 1 つゲートウェイを持つと仮定すると, 日本国内のゲートウェイの数は 55 となる. 最後に協調制御を行うクラウドは国内に 1 つと仮定する.

収容台数は各ネットワーク装置のエッジサーバ数で車両の台数を割った数となる. 車両の台数は 2016 年 8 月の登録車両台数 81,311,679 台 [52] が 1 日 1 時間 (4.2%) 動く と仮定し, $81,311,679 \times 0.042 = 3,415,090$ 台となる. 基地局にエッジサーバが配置される場合, 収容台数は無線通信で直接データを授受できる車両の台数となる. 局舎にエッジサーバが配置される場合, 収容台数は局舎に収容されている基地局と無線通信を行う車両の台数となる. ゲートウェイにエッジサーバが配置される場合, 収容台数はそのキャリアのネットワークに接続された車両の台数となる. クラウドにエッジサーバが配置される場合, 収容台数はインターネットを経由してクラウドに接続する日本全体の車両の台数となる.

3.6.2 通信遅延

本論文の5章では, 様々なキャリアの通信遅延を計測する. しかし, 我々が計測できる遅延は携帯端末-クラウド間の遅延だけであり, 携帯端末-局舎間や局舎-ゲートウェイ間のような遅延を計測することはできない. そこで, ネットワークの遅延モデルを定義し, 計測した遅延を変換することで各ネットワーク装置への通信遅延を導く. 時刻 t での LoES-UpES 間遅延モデルを lm_{L2U} , 車両-LoES 間遅延モデルを lm_{V2L} とすると,

$$\begin{aligned} lm_{L2U} &= (\alpha_{up} - \alpha_{lo})(l_{wire}(t)) \\ &= (\alpha_{up} - \alpha_{lo})(l_{mea}(t) - L_{wl}) \end{aligned} \quad (3.3)$$

$$lm_{V2L} = L_{wl} + \alpha_{lo}(l_{mea}(t) - L_{wl}) \quad (3.4)$$

このとき $l_{wire}(t)$ は時刻 t での有線遅延で, 計測遅延 $l_{mea}(t)$ から無線遅延 L_{wl} を引くことで導かれる. α_{up}, α_{lo} は UpES, LoES のロケーションパラメータで, $1.0 \sim 0.0$ の値をとり, $\alpha_{up} \geq \alpha_{lo}$ である. 各ネットワーク装置上での α_{up}, α_{lo} は5章の遅延計測の結果から決定され, 表 3.2 に示す. 本モデルでは無線遅延を定数で仮定し, ロケーションパラメータを用いて有線遅延の大きさを調整することによって自動車-LoES 間, LoES-UpES 間の通信遅延を導く.

3.3 節の車両-LoES 間遅延 l_{V2L} と LoES-UpES 間遅延 l_{L2U} は遅延モデル lm_{V2L}, lm_{L2U} を用いて表現され,

$$l_{V2L}(t) = EWMA(lm_{V2L}(t)) \quad (3.5)$$

$$l_{L2U}(t) = EWMA(lm_{L2U}(t)) \quad (3.6)$$

となる. $EWMA(X)$ は指数移動平均関数を示す.

表 3.2: ロケーションパラメータ

ネットワーク装置	ロケーションパラメータ (α_{lo}, α_{up})
クラウド	1
ゲートウェイ	0.862
局舎	0.431
基地局	0

3.6.3 移動に伴う車両の基地局間ハンドオーバについて

車両が移動しながら通信を継続するためには、移動先に応じて接続する基地局を切り換える「ハンドオーバ」が必要がある。ハンドオーバでは、どんなに短くても通信が遮断されるため、通信が不安定になる可能性が高い。その一方で、我々はハンドオーバに伴う通信の不安定は、今後改善していき、スムーズな遠隔制御を妨げる要因とはなりえないと考えている。

第一の理由が交差点への基地局の配備である。協調運転システムにおいて一番通信が重要になるのは、車両同士の軌跡が重なる交差点である。日本においては交差点の信号機は 2023 年度より 5 G 基地局の配置場所として利用可能になる [53]。現在、基地局の配置場所は飽和状態であり、信号機が配置場所として開放されることで全国の交差点に基地局が配置されるため、交差点でのハンドオーバを抑制することができる。

第二の理由はハンドオーバ技術の向上である。関連研究でも述べたが、Nokia 等によってハンドオーバをスムーズに行う研究 [20] が行われており、十分短い時間でハンドオーバを実現できる可能性は高い。

以上より、本論文では、交差点でハンドオーバが起こる可能性は十分低く、起きたとしてもハンドオーバ技術の向上によって通信が不安定になることを抑制できるため、基地局間のハンドオーバはスムーズに実現できる通信環境を想定している。

3.7 まとめ

本章では、多層エッジサーバと自動車が連携して車両制御を実現するプラットフォームを提案し、提案プラットフォームの動作および、車両制御の切手法について説明を行った。また、提案プラットフォームが動作するネットワークトポロジについても説明し、エッジサーバを配置するネットワーク装置について説明し、それらの「通信遅延」、「エッジサーバ数」、「収容台数」を導いた。

次章では、提案プラットフォームを評価するために、テストベッドを構築し、構築したテストベッドを用いて提案プラットフォームによる車両制御の安定性を評価する。

第 4 章

遠隔車両制御テストベッド構築および, 評価

4.1 はじめに

3 章で提案したプラットフォームを評価するために, 複数のエッジサーバからマイクロカーの遠隔制御を実現できるテストベッドを構築する. また, 構築したテストベッドを用いて提案プラットフォームが安定して車両を制御できるか評価する.

構築するテストベッドは実車の 1/10 スケールのマイクロカーを用いた評価環境で, 多層エッジサーバによる車両の遠隔制御をプロトタイプ実装し, 車両の走行軌跡から遠隔制御の精度を評価することができる.

テストベッドを用いた評価は 2 種類行う. 第一の評価は, 「基本特性評価」である. エッジサーバとマイクロカーの間に基本的なネットワーク障害を加えて, エッジサーバから車両の走行制御を行い, 走行軌跡を測定する. ネットワーク障害を加えていない場合と加えた場合との走行軌跡の差の大きさから, 3 章で示した制御を切り換えるための通信遅延とバーストパケットロスに関する閾値 (切換閾値) L_{hth} , L_{lth} , N_{burst} を決定する. 第二の評価は, 「実通信特性評価」である. 基本特性評価で導いた切換閾値と実計測した通信遅延データを用いて, 車両走行軌跡の精度の評価を行い, 通信の状態に応じて二層のエッジサーバ間で車両制御を切り換えることによって, 制御精度を下げることなく遠隔制御を実現できることを示す.

本章な主要な貢献は以下である.

- 協調運転システムが安定動作するための通信遅延・バーストパケットロスに関する閾値の導出

- テストベッドを用いた実通信環境化での遠隔制御の安定動作の確認

基本特性評価では、エッジサーバとマイクロカーの間に「定常遅延」, 「ランダムパケットロス」, 「バーストパケットロス」を加えて、遠隔制御を行い走行軌跡を測定することで、遠隔制御の基本的な通信負荷に対する特性を評価し、提案プラットフォームが安定して動作する3つの閾値を $L_{hth} = 100$, $L_{lth} = 70$, $N_{burst} = 3$ と決定する。また、実通信特性評価では、遅延の変化が激しい時間帯の実通信データを二層のエッジサーバ UpES と LoES の間に加え、導いた切替閾値に用いて UpES/LoES を切り換えながら遠隔制御を行うことで、理想環境と同等の安定制御を実現できることを示す。

本章は全5節で構成され、4.2節ではマイクロカーを用いたテストベッド環境について説明する。4.3節では基本的な通信負荷に対する遠隔制御の特性を、4.4節では実遅延を加えた時の走行軌跡のずれについて評価を行う。4.5節で本章をまとめる。

4.2 マイクロカーを用いたテストベッド環境

提案システムに対するテストベッドの構成を図4.1に示す。テストベッドの中で、自動車はマイクロカーを用いた評価環境である「Micro-car prototype[26, 54](以下, MPT)」によって実現される。UpES, LoES には東京大学が開発するプログラマブルノードである「FLARE[55]」を用い、LoES-UpES 間ネットワークの再現には WAN エミュレータ [56] を用いる。以下それぞれについて説明を行う。

4.2.1 Micro-car prototype

図4.2にMPTの動作の様子を示す。MPTは実車の1/10スケール(19cm × 42cm)のラジコンを改造したマイクロカーを外部から複数台同時に制御することができる遠隔制御システムのプロトタイプである。

マイクロカーは走行中、車速、向き、位置のセンシングを行い、センサデータ¹を周期的²に外部システムに対して送信し、外部システムから制御データ³を受信し、それに基づいて制御を行うことで遠隔制御を実現する。

¹車両 ID, 位置, 速度および、向き

²本論文では、センサデータの送信周期は 100ms である

³車両 ID, アクセル量, ブレーキ量および、ステアリング量

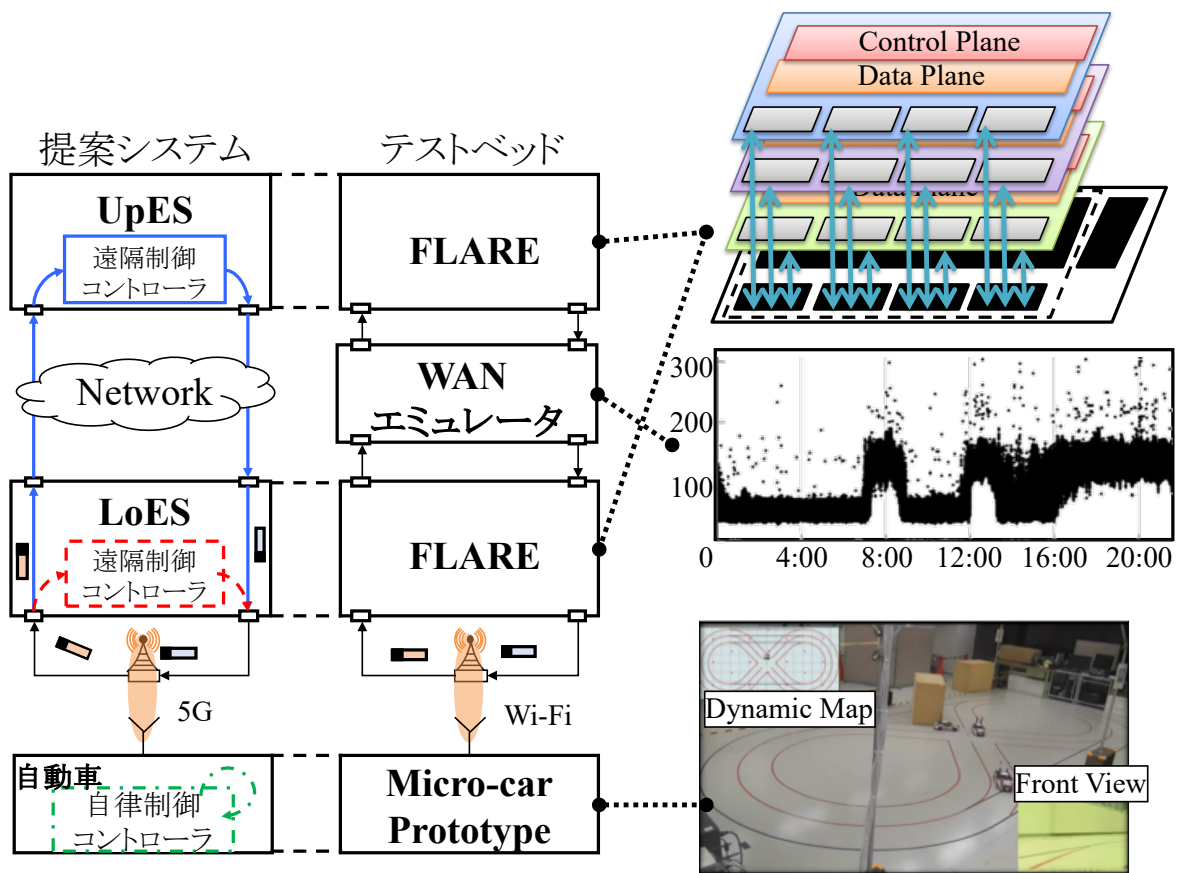


図 4.1: テストベッドの構成

図 4.3 にマイクロカーの走行コースを示す. 走行コースのサイズは $8\text{ m} \times 4\text{ m}$ で, 走行コースは 8 の字, 外周から選択できる.

本論文では, 点線で示す 8 の字コースを矢印の方向に走行して, 車両の遠隔制御の性能評価を行う. 8 の字コースは, 車両間のインタラクションが発生する交差点を含むことに加え, 車両の走行性能評価において必須である「直進/右カーブ/左カーブ」を含む. これらのことを考慮して 8 の字コースを評価コースとする.

位置のセンシングには超音波を用い, 位置精度誤差は約 2 cm である. MPT とエッジサーバの間の通信には, 5G の代替として Wi-Fi を用いる. Wi-Fi による通信遅延および, パケットロス率は十分小さい.

MPT からセンサデータを受信し, 制御データをフィードバックするのが, UpES, LoES を実現する FLARE である.

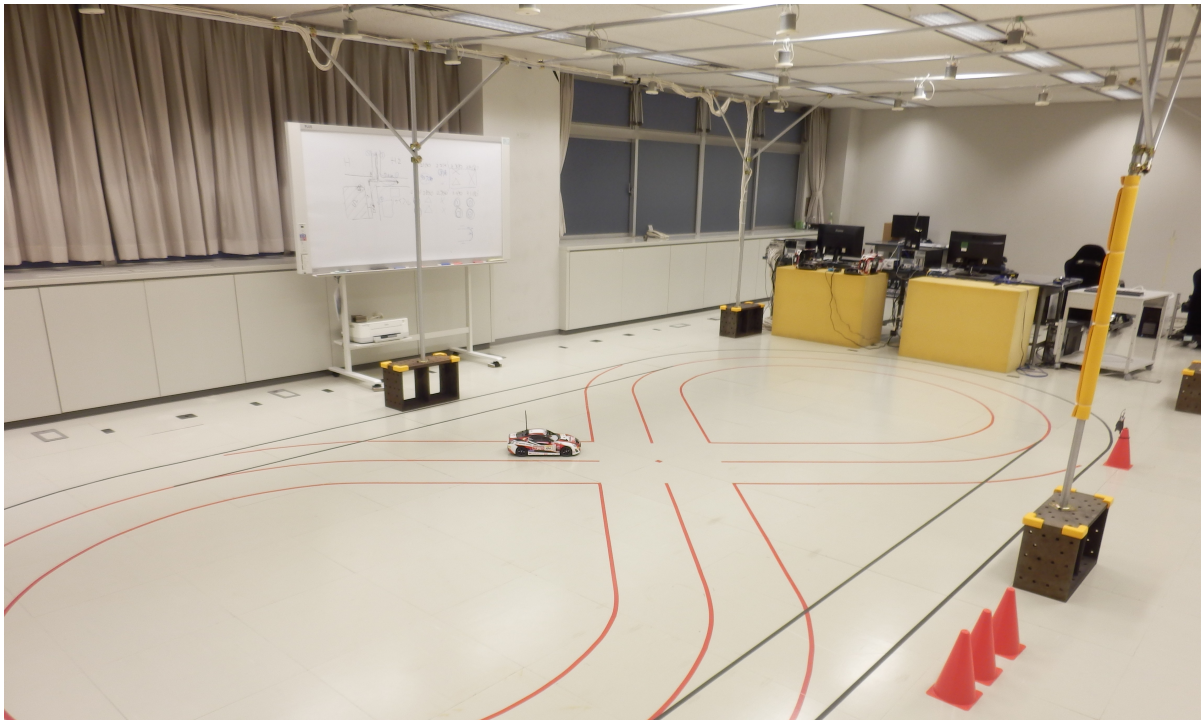


図 4.2: Micro-car prototype

4.2.2 FLARE

テストベッド内のエッジサーバにはプログラマブルノード FLARE[55]を使用する。FLARE は東京大学 中尾研究室が開発する Software Defined Networking (SDN)[57]/ Network Function Virtualization (NFV)[58] を実現可能なプログラマブルノードである。メニーコアアーキテクチャによるパケット処理のスケラビリティと後述するソフトウェアルータ Click Modular Router[59](以降, Click) を用いた柔軟なパケット処理を両立することができる。

図 4.4 に FLARE のアーキテクチャを示す。物理スイッチの上に Slice と呼ばれる仮想スイッチを複数構築することができる。Slice は FLARE を使用するユーザ毎に用意され、Control Plane (以降, C-Plane) と Data Plane (以降, D-Plane) によって構成される。D-Plane 上で Click が動作し、ユーザは Click 上にプログラムを実装することによって D-Plane を自由に拡張することができる。D-Plane には複数の CPU コアを割り当てることが可能であり、トラフィック量に応じてスケラブルに性能を調整できる。C-Plane はスライス毎に 1 つだけ CPU コアを割り当てられ、D-Plane の管理に使用される。

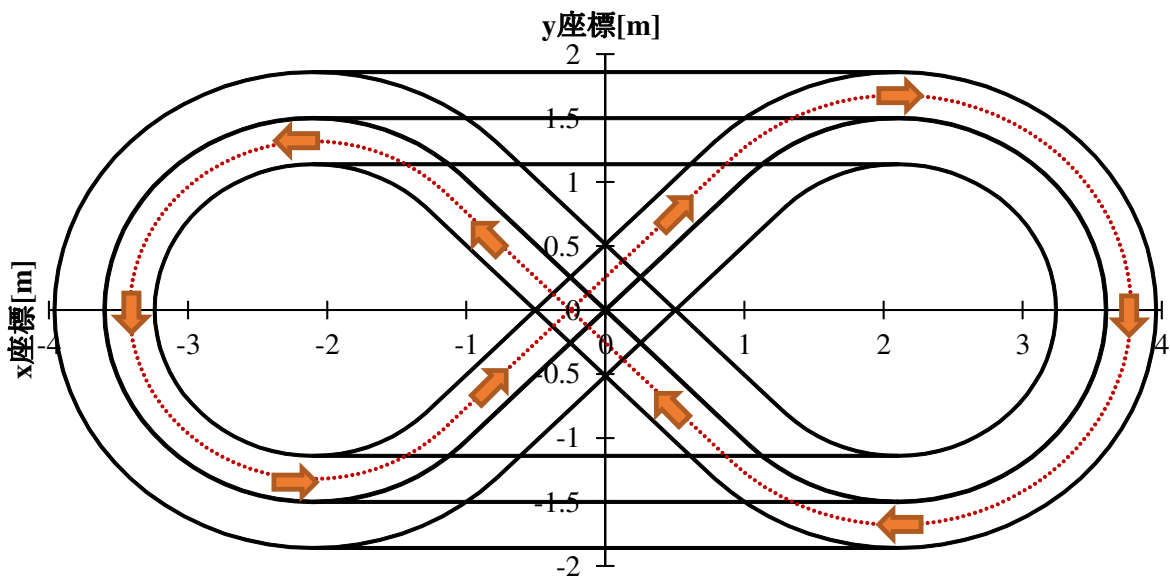


図 4.3: MPT のコース

Click Modular Router[59]

Click[59] は, MIT で開発された Unix/Linux 上で動作するモジュール形式のソフトウェアルータである. ハードウェアのルータや L2/L3 スイッチでは, パケットをストリームデータ⁴として処理する. それと同様に, Click では, パケットのストリームデータ処理を, 後述するエレメントをつなぎ合わせることによって, ソフトウェア上で実現できる構造となっている. 図 4.5 に, エレメントをつなぎ合わせることで実現した MAC 層の実装例を示す. エレメントとは, Ethernet や IP のような各種ヘッダの読み込み, ARP の処理, IP ルーティング等の単一のネットワーク機能を実現するモジュールである. このエレメントはあらかじめ用意されたもの以外にも, C/C++を用いて Click の利用者が自由に作成することが可能である. Click の利用者は既存のエレメントと組み合わせて, オリジナルのエレメントを実装/使用することで, 今までにない柔軟な処理が可能なネットワーク装置を容易に作成することができる.

本論文では, Click のオリジナルエレメントとして車両制御機能を実装し, FLARE の D-Plane で動作させることで遠隔制御を実現する.

⁴無限に到来する時刻順のデータ

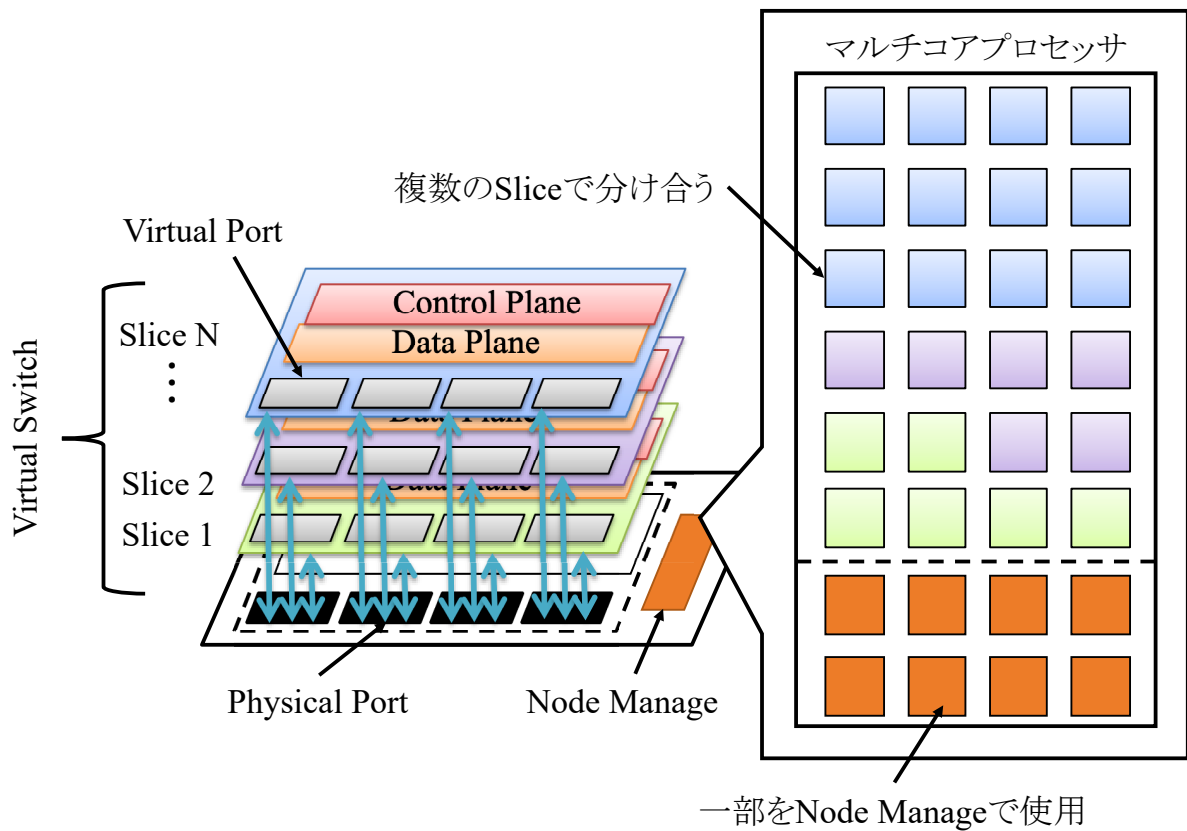


図 4.4: FLARE アーキテクチャ

FLARE 上に実装されるマイクロカーの遠隔制御アルゴリズムについて

3.5 節でも述べたが、本論文では、マイクロカーの遠隔制御アルゴリズムに Proportional-Integral-Differential 制御 (PID 制御) アルゴリズムを用いる。UpES/LoES は、センサデータを MPT より受信すると、Click 内の車両制御エレメントに転送し、車両制御エレメントで PID 制御を用いて、制御データを生成し、MPT へ送信する。

4.2.3 WAN エミュレータ

WAN エミュレータは、通信環境を模擬するために使用するソフトウェアもしくはハードウェアである。本プロトタイプでは、ソフトウェアエミュレータである tc-netem[56] を LoES-UpES 間に配置して、転送されるトラフィックに通信制御を加える。tc-netem では帯域制限や、パケットの定常遅延、パケットロスに加えて、時間に揺らぎを持った遅延、パケットの重複および、並び替えも実

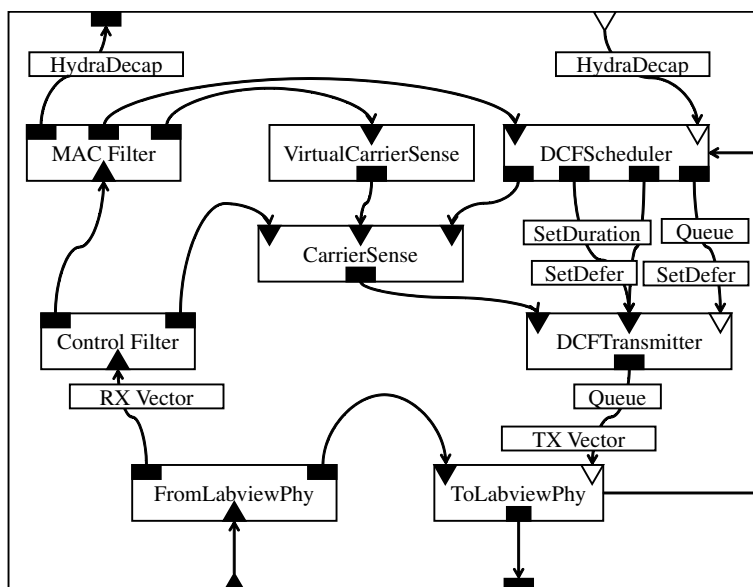


図 4.5: Click での MAC 実装例

現することができる。本論文の基本特性評価では、WAN エミュレータによって「通信遅延」、「ランダムパケットロス」、「バーストパケットロス」を加える。また、実通信特性評価では、python のスクリプトを用いて計測した通信遅延に合わせて WAN エミュレータの通信遅延を設定することで実測した通信遅延の再現を行う。

次節では、MPT を用いて、基本的な通信負荷が加わった際の走行制御の精度について評価を行う。

4.3 基本特性評価

基本特性評価では、UpES 相当の FLARE からマイクロカーの遠隔制御を行う際に、WAN エミュレータ上に「通信遅延」、「ランダムパケットロス」、「バーストパケットロス」を加えて、それらが車両の遠隔制御の精度に与える影響について評価する。また、基本特性評価の結果から制御切換に必要な切換閾値 L_{hth} , L_{lth} , N_{burst} を決定する。

4.3.1 評価環境

図 4.6 に基本特性評価環境を示す。基本特性評価では、LoES を実現する FLARE ではセンサーデータ・制御データの転送のみを行い、UpES が遠隔制御を行う。このとき、WAN エミュレータでは様々な通信負荷を加え、その有無がマイクロカーの走行軌跡に与える影響を比較評価する。

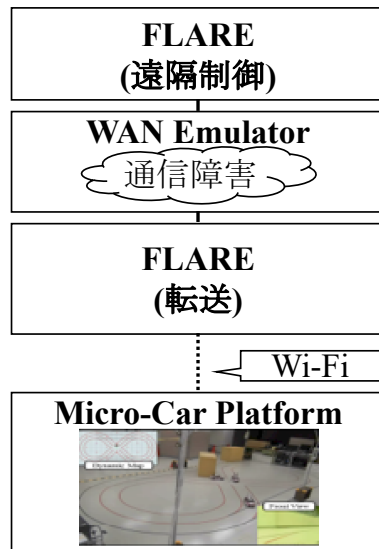


図 4.6: 基本特性評価環境

4.3.2 車両走行軌跡の評価方法について

測定した走行軌跡は標準軌跡からの「ずれ」を用いて評価する。

図 4.7 にずれの計算方法を示す。マイクロカーの位置から標準軌跡に向かって引いた垂線の長さを、その時点でのマイクロカーのずれとする。コース、マイクロカーともに 1/10 スケールであるため、マイクロカーで計測されるずれの大きさを 10 倍にすると実車相当のずれの大きさになる。

標準軌跡は、遠隔制御を WAN エミュレータからの通信負荷無しで行うときに、走行コースの中心からのずれが最も小さい軌跡である。走行コースの中心と測定した走行軌跡を比較する場合、PID 制御によるずれも含まれた結果になり、通信負荷の影響が分かりにくくなってしまう。対して、標準軌跡と測定した走行軌跡を比較する場合、通信負荷の影響によるずれだけを取り出すことができるため、本論文では測定した走行軌跡と標準軌跡の比較を行う。例として、図 4.8 に評価で使用する 8 の字コースとその標準軌跡を示す。

また、本論文では、測定した軌跡のずれの大きさが 0.09625m を超えた場合を「コースアウト」と定義する。この値は国土交通省の交通標準 [60] を参考に導いた値で、図 4.9 にコースアウトの導き方を示す。日本の標準的な乗用車 ([60] の small sized vehicle に相当) の車幅は、1.7m である。また、日本には、標準的な道路と路側帯が 4 種類 ([60] の Type1 から 4 に相当) 定義されており、道路幅、路側帯幅の最大値と最小値の中間の値を取るとそれぞれ 3.125m、0.5m となり、これらを合わ

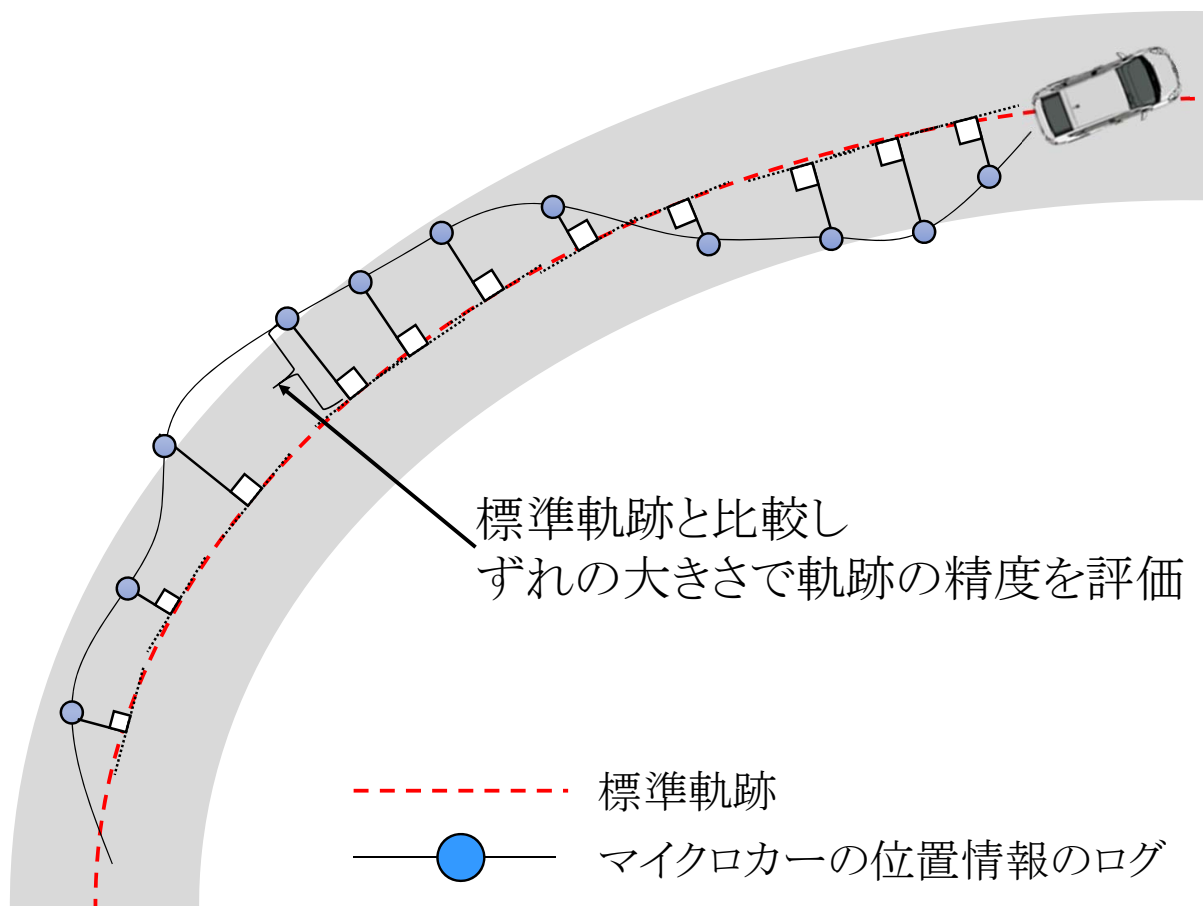


図 4.7: 「ずれ」の計算方法

せると 3.625m となる。これらの値に基づく場合、左右どちらかへ 0.9625m ずれるとコースアウトするため、MPT は実車の 1/10 スケールなので 0.09625m のずれを計測するとコースアウトとする。

4.3.3 測定条件

表 4.1 に測定パラメータを示す。1 台のマイクロカーが 8 の字コースを 6 周し、最初の 1 周を破棄した残りの 5 周の走行軌跡を使用して、ずれを評価する。マイクロカーの平均速度は、1[m/s] (実車換算 10[m/s]) で、100ms に 1 回、センサデータを送信する。WAN エミュレータには 3 種類の通信負荷「定常遅延」、「ランダムパケットロス」、「バーストパケットロス」が加えられる。以下でそれらについて説明する。

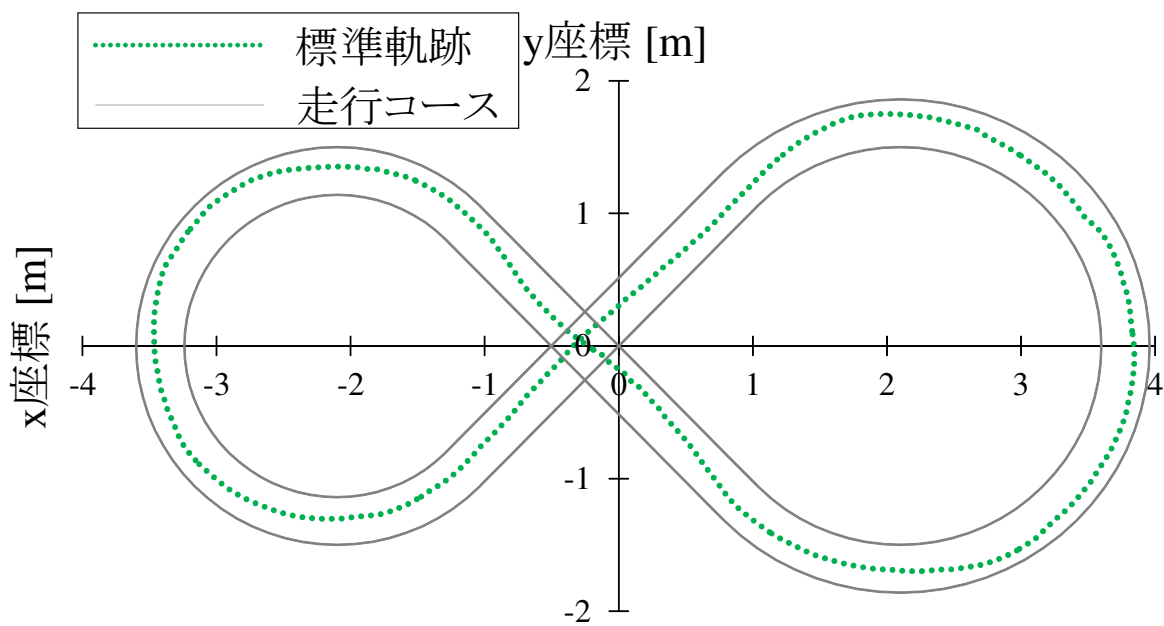


図 4.8: 走行コースと標準軌跡

定常遅延

定常遅延では LoES から UpES に転送されるセンサデータに対して均一な大きさの遅延を加える。遅延の大きさは 50ms-200ms まで 50ms 刻みで変化させ、車両の走行軌跡の測定を行う。

表 4.1: 測定パラメータ

計測条件	値
コース	8 の字
走行回数	5[周]
平均速度	1[m/s]
センサデータの送信周期	100[ms]
定常遅延	50, 100, 150, 200 [ms]
ランダムパケットロス	10, 30, 50[%]
バーストパケットロス	0, 100, ...600[ms]

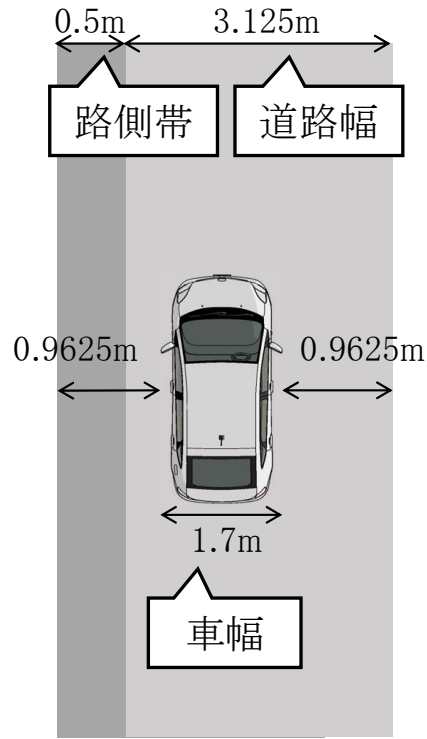


図 4.9: コースアウト

ランダムパケットロス

ランダムパケットロスでは, LoES から UpES に転送されるパケットをランダムにドロップする. パケットを破棄する確率を 10%, 30%, 50%と変化させ, 車両の走行軌跡の測定を行う.

バーストパケットロス

バーストパケットロスでは, 5 秒周期に一定期間 WAN エミュレータを通過するパケットを全てドロップする. ドロップする期間は 100ms~600ms まで 100ms 単位で変化させ, 車両の走行軌跡を測定する.

以降, WAN エミュレータを用いて, 定常遅延とランダムパケットロスおよび, バーストパケットロスを加えた場合の UpES による遠隔制御の測定結果について評価する.

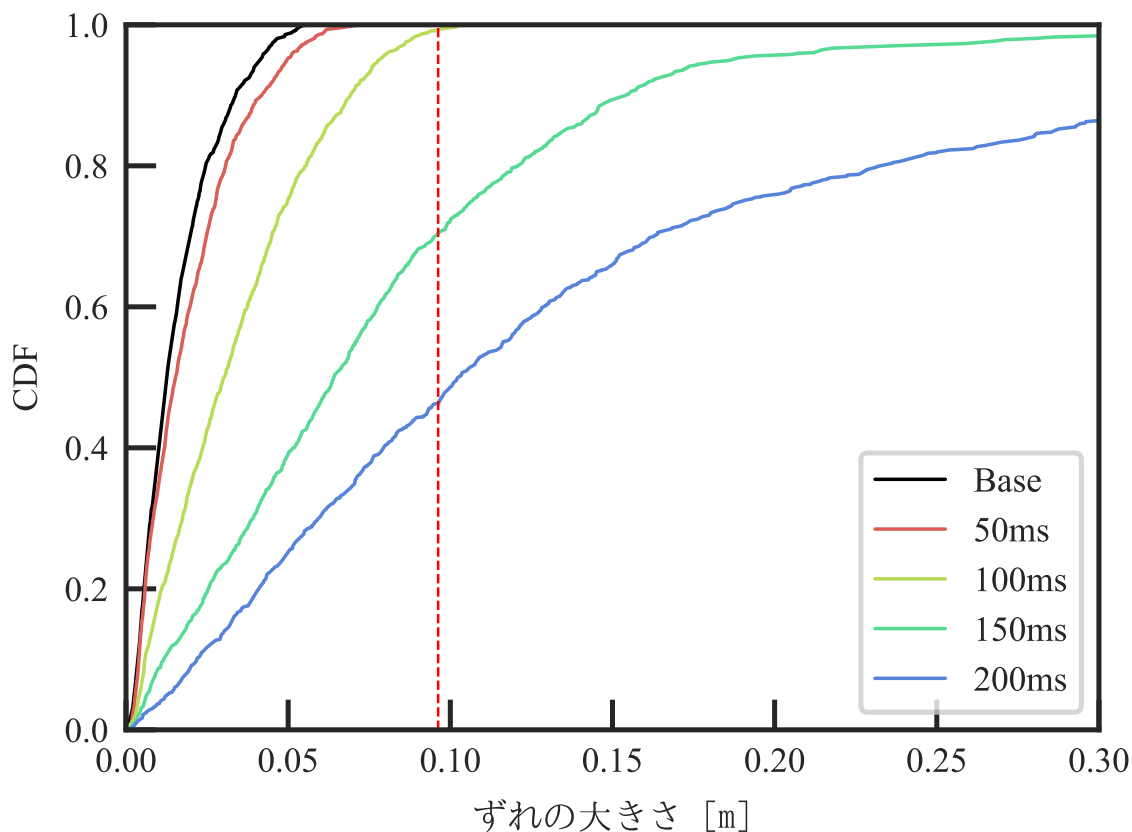


図 4.10: 定常遅延の変化に対する「ずれ」の変化 (CDF)

4.3.4 測定結果：UpES 上での車両の走行制御 (定常遅延)

図 4.10 は、WAN エミュレータに加える定常遅延の大きさの変化に対する走行軌跡のずれの変化を示す。横軸が走行軌跡のずれの大きさ [m] で、縦軸は CDF を示す。赤の点線は 0.09625m を示し赤線より右側がコースアウトした比率となる。黒線の「Base」は、WAN エミュレータからの通信負荷がない状態で評価した結果を示す。

図 4.10 より定常遅延が大きくなると、ずれが大きくなることが確認できる。また、定常遅延が 100ms 以下であれば、95%以上コースアウトすることなく安定制御が実現できるのに対して、定常遅延が 150ms を超えると大きな走行軌跡のずれが発生し、走行軌跡の約 35%がコースアウトした。図 4.11, 4.12 に定常遅延が 100ms, 150ms 時の走行軌跡をそれぞれ示す。縦軸と横軸は、二次元の空間座標になっており単位は [m] である。黒の点線はコースの中心線である。図 4.11 より定常遅延

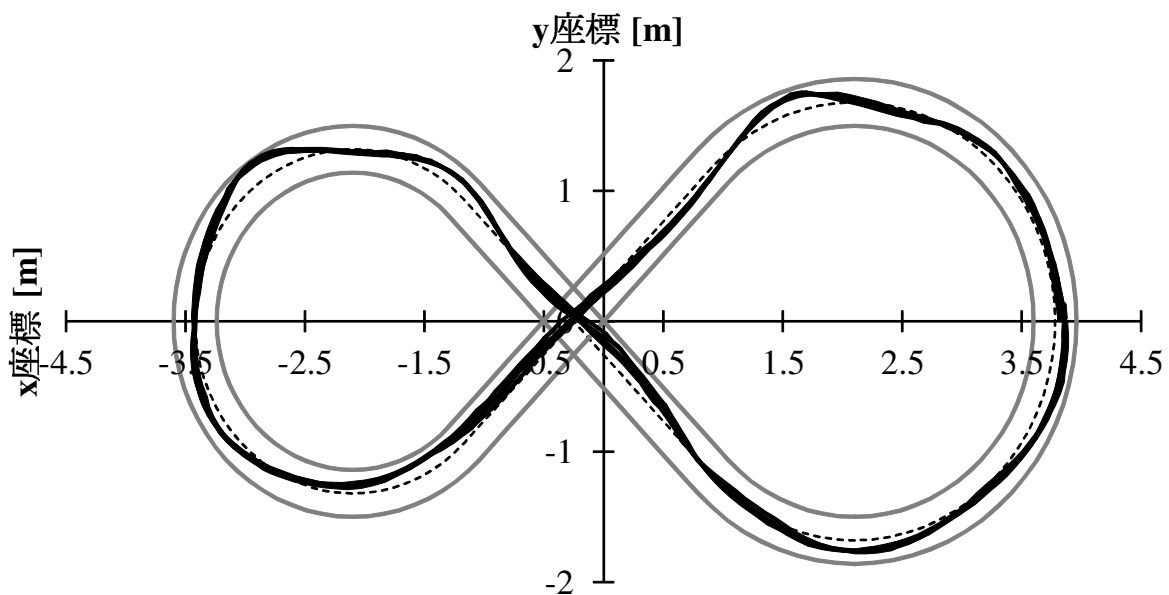


図 4.11: エッジサーバでの走行制御 (定常遅延 100ms) による軌跡

が 100ms の時は, 走行軌跡が常にコースに収まっていることが確認できる. それに対して, 図 4.12 では, 点線のコで示されたところで大きくコースから外れている. これは遅延によって遠隔制御の指示が遅れ, カーブ走行時の制御指示が残った状態で直線に入るためである. 150ms より定常遅延が大きい場合の走行軌跡をプロットすると, すべて同様にカーブ直後の直進で大きなずれが確認できた.

以上より, 定常遅延が 100ms 以下なら安定制御が実現できること, 定常遅延が 150ms 以上だと大きなずれが発生し, 走行軌跡の約 35% がコースアウトすることを確認した.

4.3.5 測定結果 : UpES 上での車両の走行制御 (ランダムパケットロス)

図 4.13 は, ランダムパケットロスの確率を変化させた際の走行軌跡のずれの変化を示す. 図 4.13 よりパケットロス率を上げていくと走行軌跡のずれが大きくなることが確認できる. ランダムパケットロスが 10% 以下だと, 95% 以上コースアウトすることなく安定した制御が実現できているものの, 30% を超えると走行軌跡のずれの最大値が急激に大きくなっている. これはランダムパケットロスのタイミングが重なって, バーストパケットロスとなっていることが原因である. 30% 以上のランダムパケットロスでは, 連続してパケットがドロップする確率が高くなり, 連続でパケットロスが発生したタイミングで大きな走行軌跡のずれが生じている. 図 4.14 にパケットロス率 30%

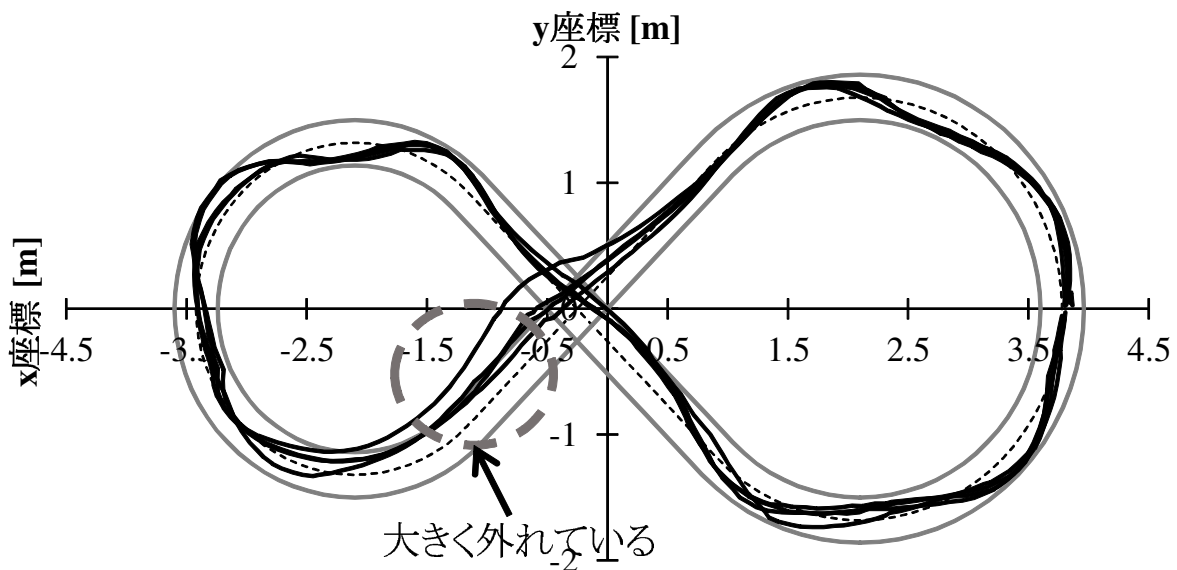


図 4.12: エッジサーバでの走行制御 (定常遅延 150ms) による軌跡

時の走行軌跡を示す。定常遅延時の走行軌跡と比較すると、定常遅延の走行軌跡ではカーブの直後に乱れるといった一定の規則性があったが、ランダムパケットロスには規則性が存在せず、バーストパケットロスが発生したタイミングで、安定した走行軌跡が突然大きく乱れることが確認できる。言い換えれば、バーストパケットロスさえ発生しなければ、30%がパケットロスする通信環境でも安定した制御を実現でき、10%以下のランダムパケットロスのようなバーストパケットロスが発生しにくい状況では安定した制御を実現できる。

以上より、ランダムパケットロス率が増加するとバーストパケットロスが発生する確率が高くなり制御が乱れやすくなる。その一方で、ランダムパケットロス率が10%以下であるバーストパケットロスが発生しにくい状況では安定した遠隔制御を実現できることが確認できた。

4.3.6 測定結果：UpES 上での車両の走行制御 (バーストパケットロス)

図 4.15 にバーストパケットロスを加えた際の走行軌跡のずれの変化を示す。図 4.15 を見ると、バーストパケットロスの時間が400ms までは95%以上の時間帯で、安定制御が実現できているのに対して、バーストパケットロスの時間が500ms で20%コースアウトすることが確認できる。このことは、遠隔制御のセンサの送信周期が100msであることを考慮すると、バーストパケットロスが4つを超えると安定した制御が維持できないことを示す。

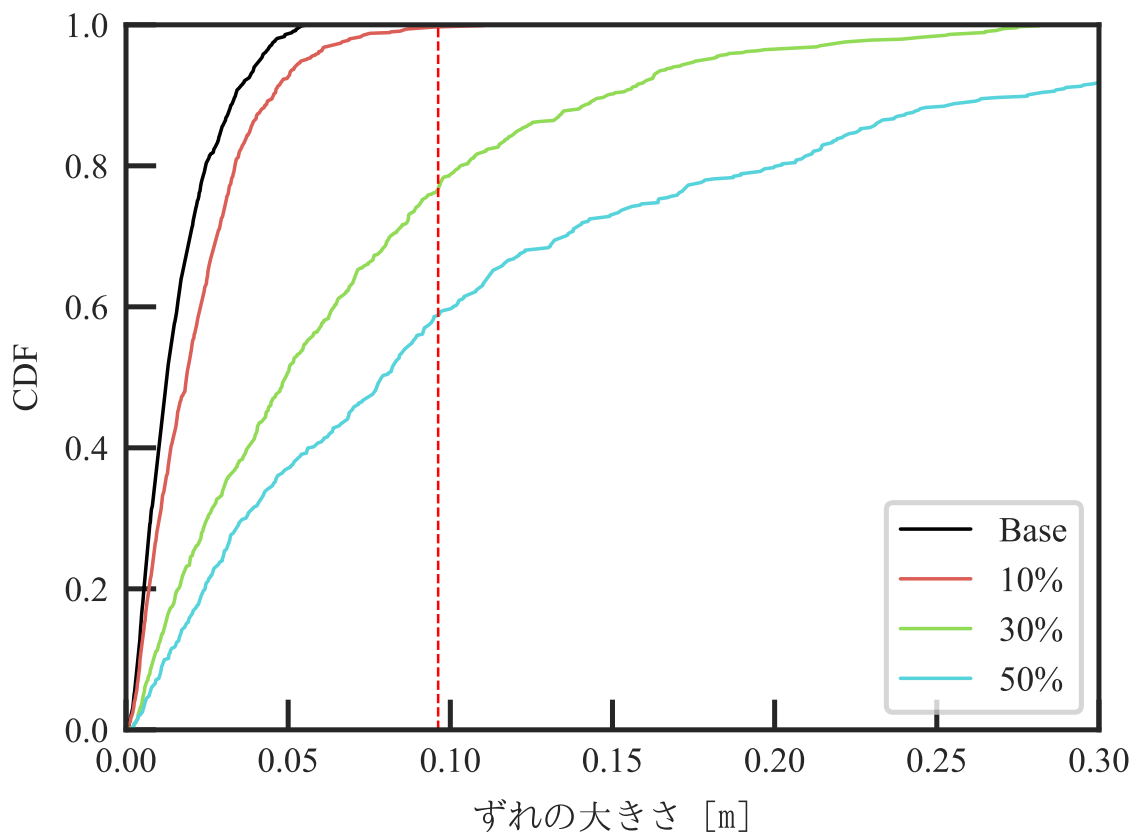


図 4.13: ランダムパケットロス率の変化による走行軌跡のずれの変化

4.3.7 評価のまとめ

定常遅延を加えた遠隔制御の評価より, 定常遅延が 100ms なら大きな走行軌跡のずれを発生させることなく UpES から車両の遠隔制御を実現できることが分かった. また, ランダムパケットロス, バーストパケットロスと 2 種類のパケットロスを変えた評価を行ったが, 両評価ともに制御が不安定になる原因はバーストパケットロスであった. バーストパケットロスの評価より, ドロップ期間が 400ms 以下なら安定した遠隔制御が実現できることが確認できた.

4.3.8 安定制御に必要な切換閾値について

定常遅延を加えた評価結果より, ネットワーク上位の制御装置が下位の制御装置に切り換える通信遅延閾値 L_{hth} を 100ms とする. また, バーストパケットロスを加えた評価結果より, 遠隔制

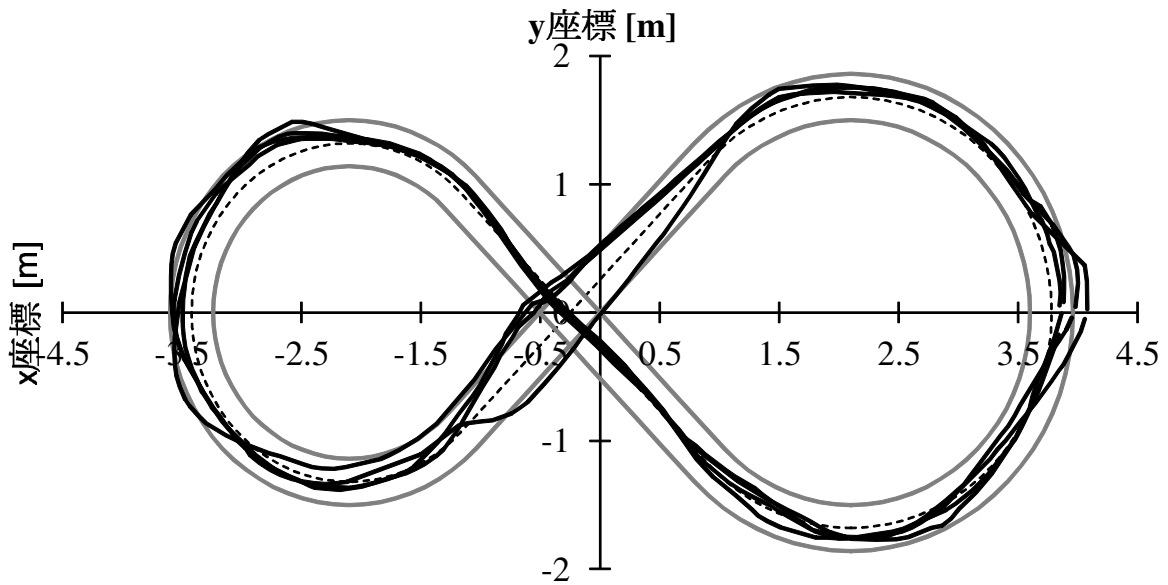


図 4.14: エッジサーバでの走行制御 (ランダムパケットロス 30%) による軌跡

御のセンサの送信周期が 100ms であることを考慮すると, N_{burst} を 3 に設定することで, 遠隔制御が大きく乱れる前に, 自律制御へと移行することができる。

ネットワーク下位の制御装置が上位の制御装置に切り換える通信遅延閾値 L_{lth} は, L_{hth} より小さい値なら安定した遠隔制御を実現できる。しかし, L_{hth} と L_{lth} があまり近い値だと, 制御の切り換えを頻発し, 結果として不安定な制御になる可能性がある。そこで, L_{lth} の値を L_{hth} より 30ms 小さい「70ms」とすることで, 制御装置の切り換えにヒステリシスを持たせる。

以上より決定した $L_{hth} = 100$, $L_{lth} = 70$, $N_{burst} = 3$ を用いて, 次節の実通信特性評価と 6 章のインフラ型車両制御システムのシミュレーション評価を行う。

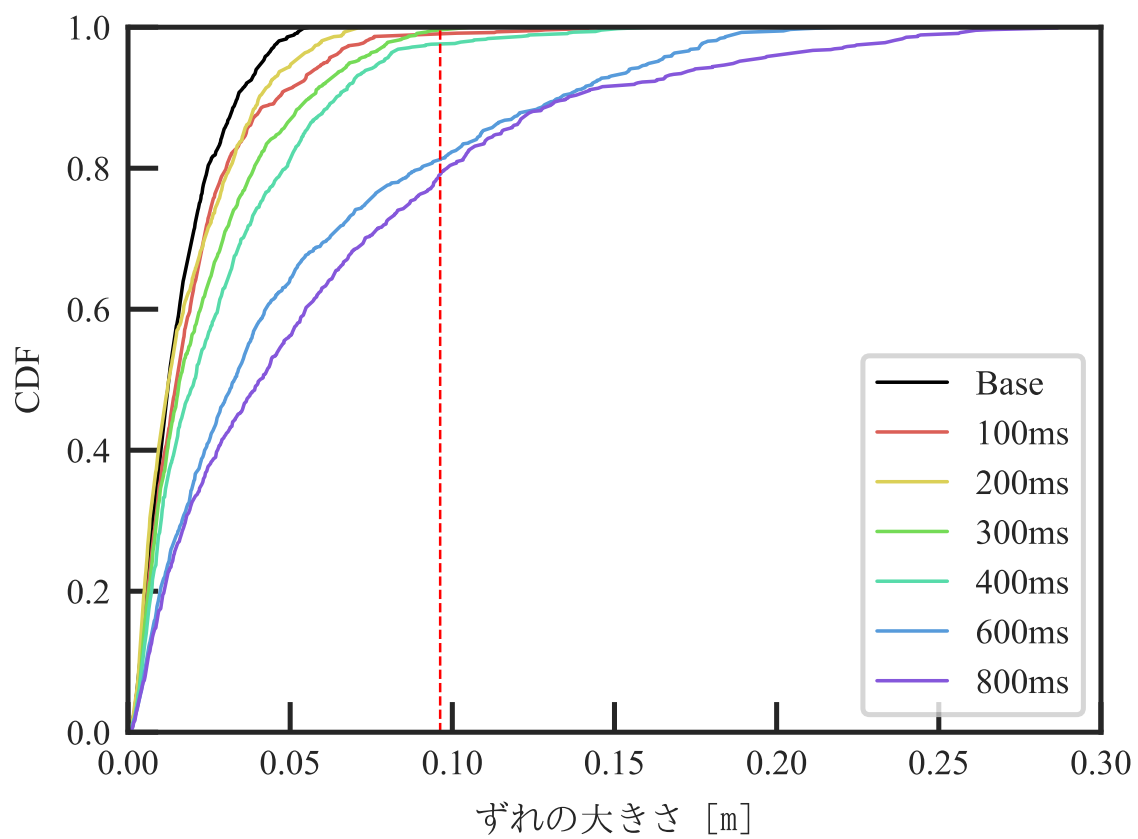


図 4.15: バーストパケットロス率の変化による走行軌跡のずれの変化

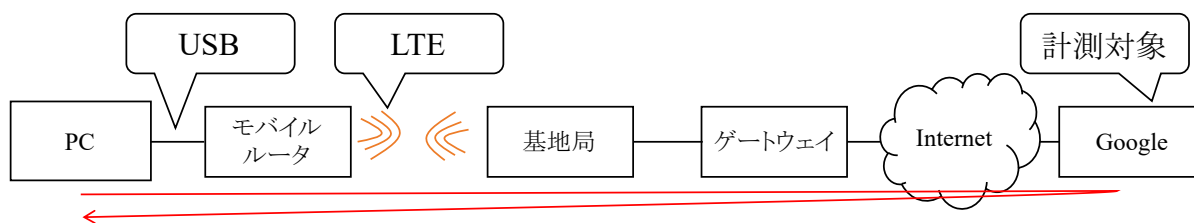


図 4.16: 計測環境

4.4 実通信特性評価

前節にて導いた切換閾値を用いて実通信遅延に対する遠隔制御の特性を評価する。

UpES をクラウドに、LoES を基地局に配置し、「UpES 制御」、「LoES 制御」、「UpES/LoES 切換制御」で比較を行う。このとき、UpES と LoES の間に存在する WAN エミュレータでは実通信遅延を再現を行う。再現を行う通信遅延は、携帯端末-クラウドサービス間の通信遅延を一日測定し、その中で遅延の変化が激しい時間帯の遅延とする。

最初に、通信遅延の計測方法について説明し、その後計測した遅延を用いた評価について述べる。

4.4.1 実通信遅延計測方法

本節では、インターネットの代表的な遅延を測定するために、Google Public DNS[61] に対して 100ms 周期で ping (5.3 節参照) を用いた遅延計測を行い、遅延の大きさの変化を計測する。Google は代表的なクラウドサービス事業者であり日本で大規模なサービス展開を行っているため、Google のサービスに対して遅延計測を行うことで、超広範囲の協調運転システムの運用を想定した通信計測が可能になる。Google Public DNS は Google が管理する OpenDNS で、日本国内にサーバが存在し、ping に対して応答を返すため、本節では Google Public DNS に対して遅延計測を行う。

遅延計測環境を図 4.16 に、計測パラメータを表 4.2 に示す。PC とモバイルルータは USB で接続されている。モバイルルータには楽天モバイル [62] の SIM が搭載されており、モバイルルータを介してインターネットにアクセスすることができる。PC から Google Public DNS (8.8.8.8) に対して ping を 100ms 周期で送信し、ラウンドトリップタイムを遅延として計測する。計測期間は 2016 年 11 月 1 日の 0:00~24:00 である。

図 4.17 に遅延計測の結果を示す。横軸が経過時間を、縦軸が遅延の大きさ (ms) を示す。平均遅延は 86.7ms であり、パケットロス率は 0.3% である。図 4.17 を見ると、30-40ms の幅で遅延が細か

表 4.2: 計測パラメータ

パラメータ名	値
送信先	Google Public DNS (8.8.8.8)
使用回線	楽天モバイル
送信周期	100ms
計測期間	2016/11/1 0:00 - 24:00

く変動することが確認できる。また、時間帯によって大きく平均遅延も変動しており、昼の 12:00 から 13:30 頃や夕方など多くの人がインターネットを使用する時間帯では平均遅延が大きく増加する。加えて、突発的に 200ms 以上の大きな遅延が発生している。ただし、遅延が 200ms 以上のパケットとパケットロスを合わせても全体の 0.4% であるため、ランダムなパケットロスであれば遠隔制御に大きな影響は与えない。

4.4.2 評価方法

WAN エミュレータ上で、計測遅延を再現し、遠隔制御の精度について性能評価を行う。3 つの制御手法「UpES 制御」、「LoES 制御」、「UpES/LoES 切替制御」でマイクロカーの走行制御を行い、走行軌跡のずれを比較する。図 4.18 に評価環境を示す。「UpES 制御」は図 4.18 の上位の FLARE から、「LoES 制御」は下位の FLARE から遠隔制御を行い、「UpES/LoES 切替制御」は両 FLARE を通信状況に応じて切り換えて遠隔制御を行う。本評価では、UpES がクラウドに、LoES が基地局に配置されたネットワーク環境を想定している。

表 4.3 に測定パラメータを示す。マイクロカーは 8 の字コースを計測遅延を加えた状態で 6 分間走行する。評価には計測した 24 時間の遅延の中で、通信遅延の変化の大きく切替が頻繁に起こる時間帯 (13:29-13:35) の通信遅延を使用する (図 4.17)。車両の走行速度は 1m/s (実車換算時速 36km) で、マイクロカーからのセンサデータの送信周期は 100ms である。切替閾値 L_{hth} , L_{lth} は 4.3 節より 100ms, 70ms とする。

4.4.3 評価結果

図 4.19 に各制御手法による軌跡のずれの大きさを示す。横軸がずれの大きさを、縦軸が CDF を示し、黒、青、赤線がそれぞれ「LoES 制御」、「UpES 制御」、「UpES/LoES 切替制御」を示す。

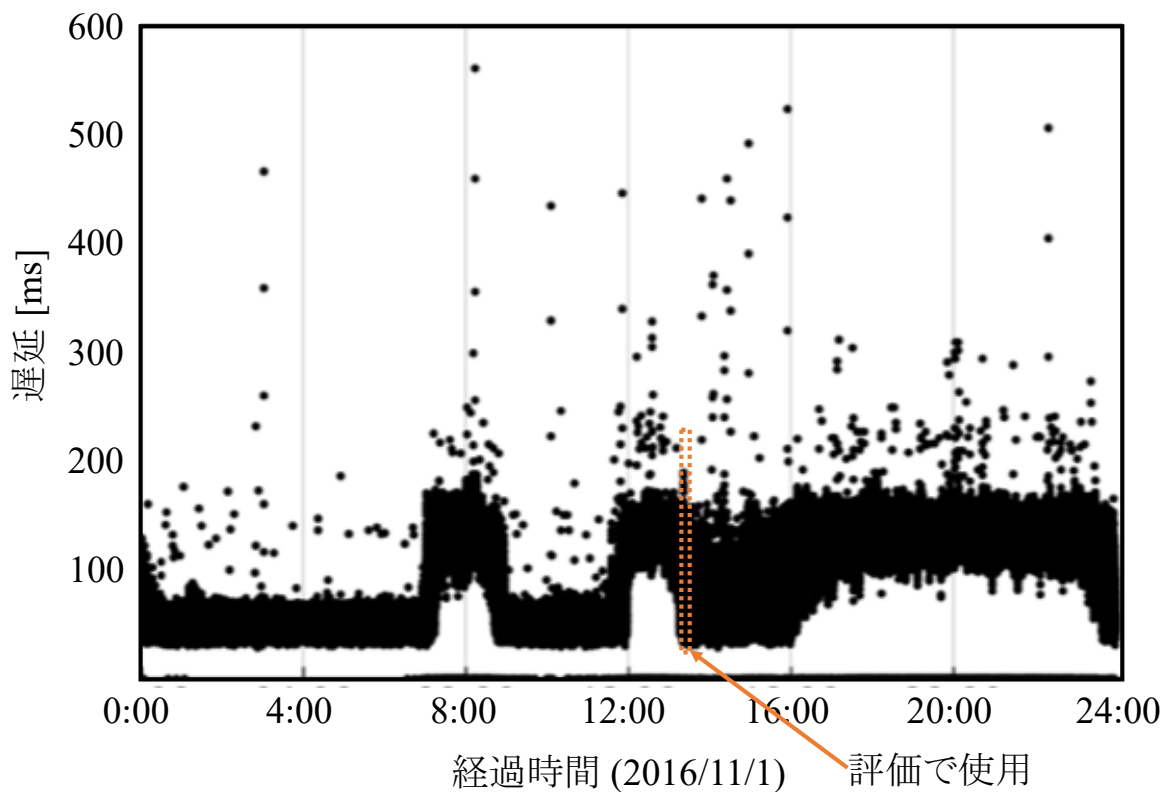


図 4.17: 計測結果

LoES は基地局に配置され、計測遅延の影響なく制御を行っているため、コースアウトすることなく安定した遠隔制御を実現している。それに対して、UpES 制御では、約 40%がコースアウトしている。このことは、インターネットの遅延の影響により、UpES だけでは遠隔制御が成立しないことを示している。一方で、UpES/LoES 切替制御では、46%の制御を UpES に任せている (図 4.21) のにも拘らず、LoES よる制御とほぼ同等の制御性能を持つ。

図 4.20 はマイクロカーの走行軌跡の例を示し、縦、横軸は各測定のマイクロカーの位置を示す。黒、青、赤線がそれぞれ「LoES 制御」、「UpES 制御」、「UpES/LoES 切替制御」を示す。図 4.20 から UpES 制御だと大きくコースから外れ、軌跡が安定しないのに対し、LoES 制御と UpES/LoES 切替制御は同等の遠隔制御が実現できていることが確認できる。

図 4.21 に「UpES/LoES 切替制御」実行時の、UpES/LoES 間の制御比率を示す。横軸は経過時間 (2016/11/1 13:29–13:35) を縦軸は遅延の大きさを示し、黒い点が計測遅延である。赤と青の線が計測遅延の EWMA を示し、青線の時間帯で UpES 制御を、赤線の時間帯で LoES 制御を行う。UpES

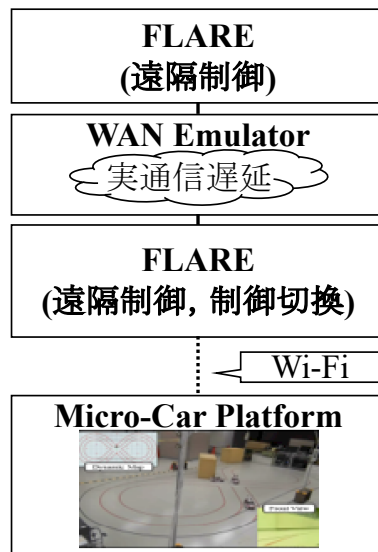


図 4.18: 実通信特性評価環境

表 4.3: 測定パラメータ

測定条件	値
コース	8 の字
走行時間	6 分間
平均速度	1[m/s]
センサデータの送信周期	100[ms]
L_{hth}	100[ms]
L_{lth}	70 [ms]

と LoES の制御の比率は 46%:54%である。図 4.21 より、遅延の変動が激しい時間帯でも、その変化に対応して UpES-LoES 間で遠隔制御を切り換えられていることが確認できる。

以上より、UpES, LoES を切り換えながら遠隔制御を行うことで、LoES と同等の安定制御を実現できることを示した。

4.5 まとめ

本章では、提案プラットフォームを評価するために、テストベッド環境を構築し、遠隔制御の成立性について評価を行った。

テストベッドは実車の 1/10 スケールのマイクロカーで構築される MPT と、UpES および、LoES

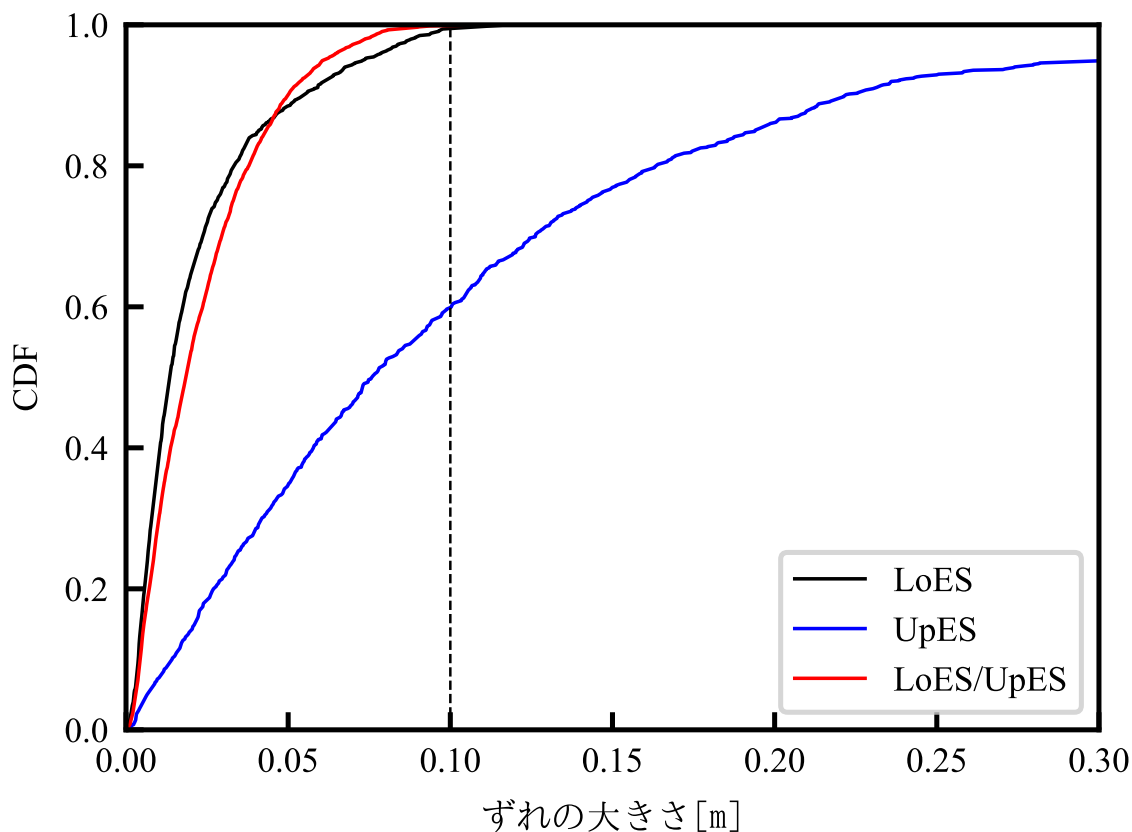


図 4.19: CDF (ずれ)

を実現する FLARE, UpES と LoES の間で実通信遅延を再現する WAN エミュレータで構成されている。

テストベッドを用いて基本特性評価と実通信特性評価の 2 種類を行った。基本特性評価では、WAN エミュレータ上で「定常遅延」、「ランダムパケットロス」、「バーストパケットロス」を加え、遠隔制御の基本的な通信負荷に対する特性を評価し、提案プラットフォームが安定して制御を切り換えるための通信に関する 3 つの閾値を $L_{hth} = 100$, $L_{lth} = 70$, $N_{burst} = 3$ と決定した。実通信特性評価では、導いた閾値に基づいて UpES/LoES を切換えながら遠隔制御を行うことで、LoES と同等の安定制御を実現できることを示した。

以上より、安定して遠隔制御・自律制御を切換可能な閾値を決定し、その閾値を用いれば安定した遠隔制御が実現できることを確認した。

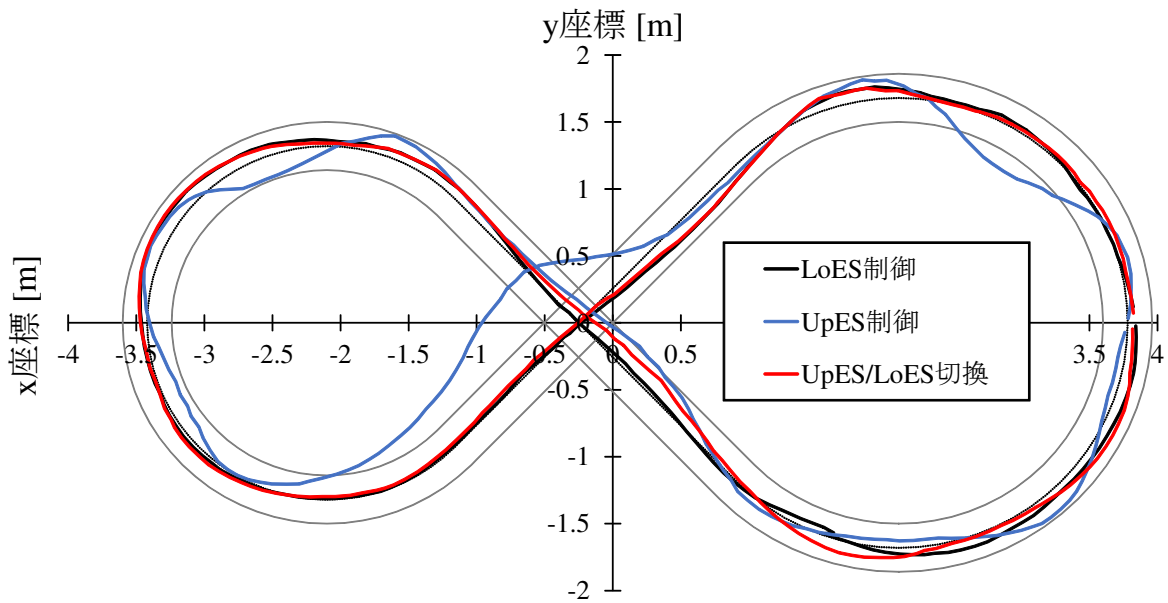


図 4.20: 例. 車両軌跡

一方で、本章の実験は提案プラットフォームで安定制御が実現できることを確認する事が目的であるため、遅延の変動が激しい時間帯での短時間の評価しか行わなかった。超広範囲の協調運転を実現するためには、1つのキャリアだけでなく複数のキャリアの通信遅延を測定し、長時間の車両制御の比率を知る必要がある。もし、図 4.21 で示すように、全時間帯の 54%の時間帯で LoES 制御や自律制御が必要となる場合、超広範囲の協調運転の実現は困難である。

複数のキャリアの通信環境の測定を 5 章で、それらを用いた長時間の車両制御のシミュレーション評価を 6 章で行う。

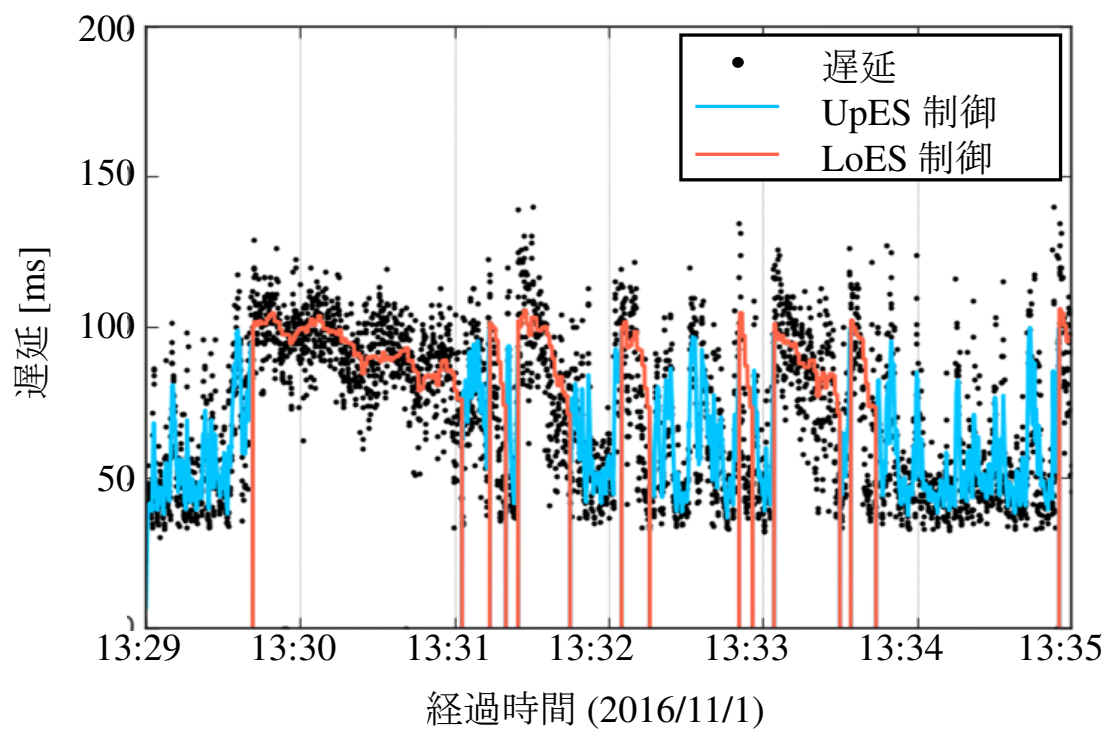


図 4.21: UpES, LoES 間の制御切換の様子

第 5 章

日本国内の通信遅延計測

5.1 はじめに

提案するインフラ型車両制御プラットフォームは、動作環境として 5G による移動体通信網を想定している。5G は 2020 年にサービスの開始を予定しており、試験環境を除くと 5G サービスを利用した提案プラットフォームの評価は実現できない。

4 章では、提案プラットフォームで安定制御が実現できることを確認するために 1 つのキャリア (楽天モバイル) と 1 つのクラウドサービス (Google Public DNS) の間で通信遅延を計測し、プロトタイプを用いて、遅延の変動が激しい時間帯での車両制御評価を行った。しかし、超広範囲の協調運転システムを実現するためには、1 つのキャリアとクラウドサービスだけでなく、複数のキャリアとクラウドサービス通信環境を測定し、提案プラットフォームを用いた長時間の車両制御について評価する必要がある。

一方で、キャリアの遅延特性について、ごく短い時間の遅延を計測した例 [63] は存在するが、長時間の遅延の変動を計測したデータや文献は公開されていない。日本国内には主要な MNO が 3 社存在することに加え、様々な MVNO が台頭してきている。これらキャリアの遅延特性を把握することなく提案プラットフォームの実現可能性を議論することは難しい。

本章では、提案プラットフォームを評価するために、複数のキャリア-クラウドサービス間の通信遅延、パケットロスおよび、ネットワークトポロジについて 24 時間の計測を行う。通信計測には 5.3 節で説明を行う 2 つのネットワークコマンド「ping」、 「traceroute」を用い、ping によって通信遅延とパケットロスを、traceroute によってネットワークトポロジを測定する。計測対象となる

クラウドサービスは Amazon, Google, 東京大学の 3 つで, それらに対して 11 キャリアの通信端末を用いて通信計測を行う.

本章の主要な貢献は以下である.

- 車両制御を目的とした日本国内の通信環境の網羅的な測定・評価

- 計測データに基づく, エッジサーバの配置と通信遅延の関係性の導出

traceroute による評価から, 大規模なクラウドサービスは, インターネットを経由することなく, キャリアと直接接続するパスを持つことを示す. また, ping を用いた評価から, 「キャリア毎に異なる通信経路の特徴をもつこと」, 「MNO の 3 社はパケットロスが発生させにくい安定したネットワークを持つこと」を示す. 最後に, ping と traceroute の結果から, 3 章で定義した各ネットワーク装置 (基地局, 局舎, ゲートウェイ, クラウド) 上での UpES, LoES のロケーションパラメータ α_{up}, α_{lo} を決定する.

本章は, 全 6 節で構成されている. 5.2 節では測定するネットワークの構造について説明する. 5.3 節では測定方法と対象について述べ, 5.4 節で測定結果についてまとめる. 5.5 節ではロケーションパラメータの値を決定する. 5.6 節で本章をまとめる.

5.2 測定するネットワークの構造について

計測を行うネットワークの構造を図 5.1 に示す. 通信端末からクラウドに向けて通信を開始する場合, まず基地局を経由して無線通信回線に転送される. 2018 年現在, 日本において全国展開されている無線通信回線網は docomo, SoftBank, KDDI の 3 社であり, 今後楽天が参入する予定である [64]. 無線通信回線網のパケットは各 MNO, MVNO の Packet data network Gateway (ゲートウェイ) に転送される. このゲートウェイは無線通信回線網で使用される専用プロトコルをインターネットで通信できるプロトコル (TCP/IP プロトコル) に変換する. ゲートウェイがパケットを転送し MNO, MVNO のネットワークを抜けると, インターネット区間を介してクラウドに接続される. このとき, MNO, MVNO が管理する各ネットワークを自律システム (Autonomous System (AS)) と定義し, 各クラウドと AS は一般的には Internet eXchange (IX) と呼ばれる相互接続ポイントを経由してクラウドに接続される.

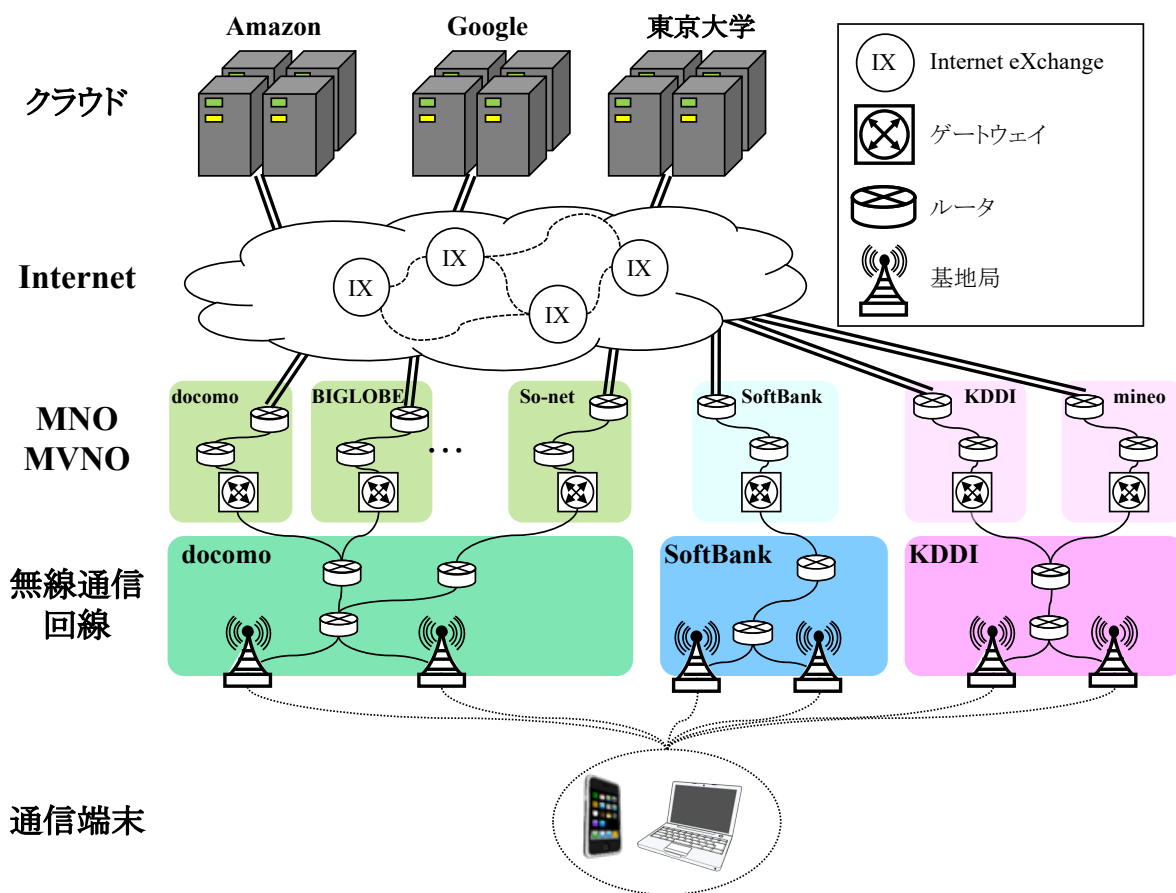


図 5.1: 移動体通信網の構造

本章では, traceroute によるネットワークポロジの測定を行う. ただし, 通信端末-ゲートウェイ間は専用プロトコルによって隠蔽されているため, 観測できるネットワークポロジはゲートウェイ-クラウド間のネットワークである.

5.3 調査

3つのクラウドサービスを計測対象として, 11 キャリアの SIM を挿入した携帯端末を用いて網羅的に (合計 33 経路) 調査を実行する. 1つのキャリア-クラウドサービス経路につき 24 時間の定点調査を行う. 調査項目は通信遅延, パケットロスおよび, ネットワークポロジである.

5.3.1 調査方法

本報告書では通信遅延、パケットロスの調査に ping を、ネットワークトポロジの調査に traceroute を用いる。

ping

ping は IP ネットワーク上で計測対象へ到達できるかを確認するためのツールで、送信元から計測対象までの往復遅延時間を測定できる。提案プラットフォームでは、車両が 100ms 周期でセンサデータを送信し、制御データを受信することで、遠隔制御を実現する。そこで、本計測でも 100ms 周期で遅延測定を行う。

traceroute

traceroute は IP ネットワークにて計測対象までのパケットの経路を調査するツールで、送信元から計測対象に至るまでに通過するネットワーク装置の IP、通信遅延を知ることができる。本計測では、traceroute を用いて 1 分に 1 回経路情報を入手し、入手した経路情報の IP アドレスから WhoIS[65] コマンドを用いて IP を所有する AS を特定、AS 間の接続関係を示すネットワークトポロジを作成する。また、本計測での traceroute の 1 ホップ目で計測されるネットワーク装置がゲートウェイの次の装置であることから、traceroute の 1 ホップ目の遅延をゲートウェイまでの遅延とし、端末-ゲートウェイ間遅延の計測を行う。

5.3.2 調査対象

3つのクラウドサービスと 11 のキャリアを使用して調査を行う。表 5.1 に計測対象であるクラウドサービス一覧を示す。本論文では、代表的なクラウド事業者である Amazon と Google のクラウドサービス、物理的な位置が把握できている東京大学のサーバをクラウドサービスとして使用する。表 5.2 にキャリア一覧を示す。日本の代表的な MNO である docomo, SoftBank, KDDI に加え、MVNO 8 社の回線で調査を行った。以降は略称を使用して、各キャリアを表記する。

表 5.1: クラウドサービス一覧

クラウドサービス	IP
Amazon AWS[66]	13.113.254.104
Google Public DNS[61]	8.8.8.8
東京大学サーバ	東京大学 IP

表 5.2: キャリア一覧

キャリア名称	略称	回線	通信端末
(株) NTT ドコモ [48]	D	D	Huawei HW-02G
KDDI (株) [50]	K	K	Huawei 501 HW
ソフトバンク (株) [49]	SB	SB	Huawei W01
日本通信 (株) (b-mobile) [67]	BM	D	NEC MR04LN
ビッグローブ (株) [68]	BL	D	NEC MR04LN
フリービット (株) [69]	FB	D	NEC MR04LN
(株) インターネットイニシアティブ [70]	IJ	D	NEC MR04LN
(株) インターリンク [71]	IL	D	NEC MR04LN
NTT コミュニケーションズ (株) [72]	OCN	D	NEC MR04LN
ソニーモバイルコミュニケーションズ (株) [73]	SN	D	NEC MR04LN
(株) ケイ・オプティコム [74]	MN	K	NEC MR04LN

5.4 調査結果

ping と traceroute による 24 時間計測の調査結果を示す. 初めに traceroute の測定結果から作成したネットワークトポロジについて説明する. 次に, ping による遅延計測から各計測経路でのパケットロス, 最低遅延および, 遅延の中央値を導き, traceroute と ping の結果の関連を推測する. 最後に, ping と traceroute の 1hop 目の遅延から, 端末-クラウド間遅延に対する端末-ゲートウェイ間遅延の比率を導く.

5.4.1 ネットワークトポロジ

図 5.2 に traceroute より導いたネットワークトポロジを示す. 赤色のノードがクラウドサービス (SINET[75] は, 東京大学のサーバが存在する AS を示す) を, 緑色のノードが IX を, 青のノードが各キャリア (濃い青が MNO, 薄い青が MVNO) を示す. 青と緑の間色である OCN と IJ は

MVNO であると同時に IX としての役割も果たしている。黒色の小さいノードは traceroute に応答しなかった所属不明のノード (以降, Unknown ノード) である。また矢印は AS 間の接続関係を示しており, 赤色矢印はキャリアとクラウドが IX 無しで直接接続する経路を示す。青色の矢印は Unknown ノードを経由してキャリアとクラウドが接続している経路を示しており, Unknown ノードが IX ではなくキャリアもしくはクラウドに所属するネットワーク装置である場合, キャリアとクラウドが直接接続される経路となる。

図 5.2 を見るとキャリアが SINET に接続するためには IX を必ず経由する必要がある (IIIJ と OCN は IX としての役割を果たしている)。その一方で, 大規模にクラウドサービスを展開している Amazon や Google は多くの MNO, MVNO と直接接続できることが確認できる。特に Google は多くのキャリアと直接接続が可能であり, 今回計測を行った 11 のキャリアのうち少なくとも 6 つの経路で直接接続が実現できている。このことは, 大規模なクラウドサービスを使用することでインターネットが原因となる通信障害を抑制できることを示している。

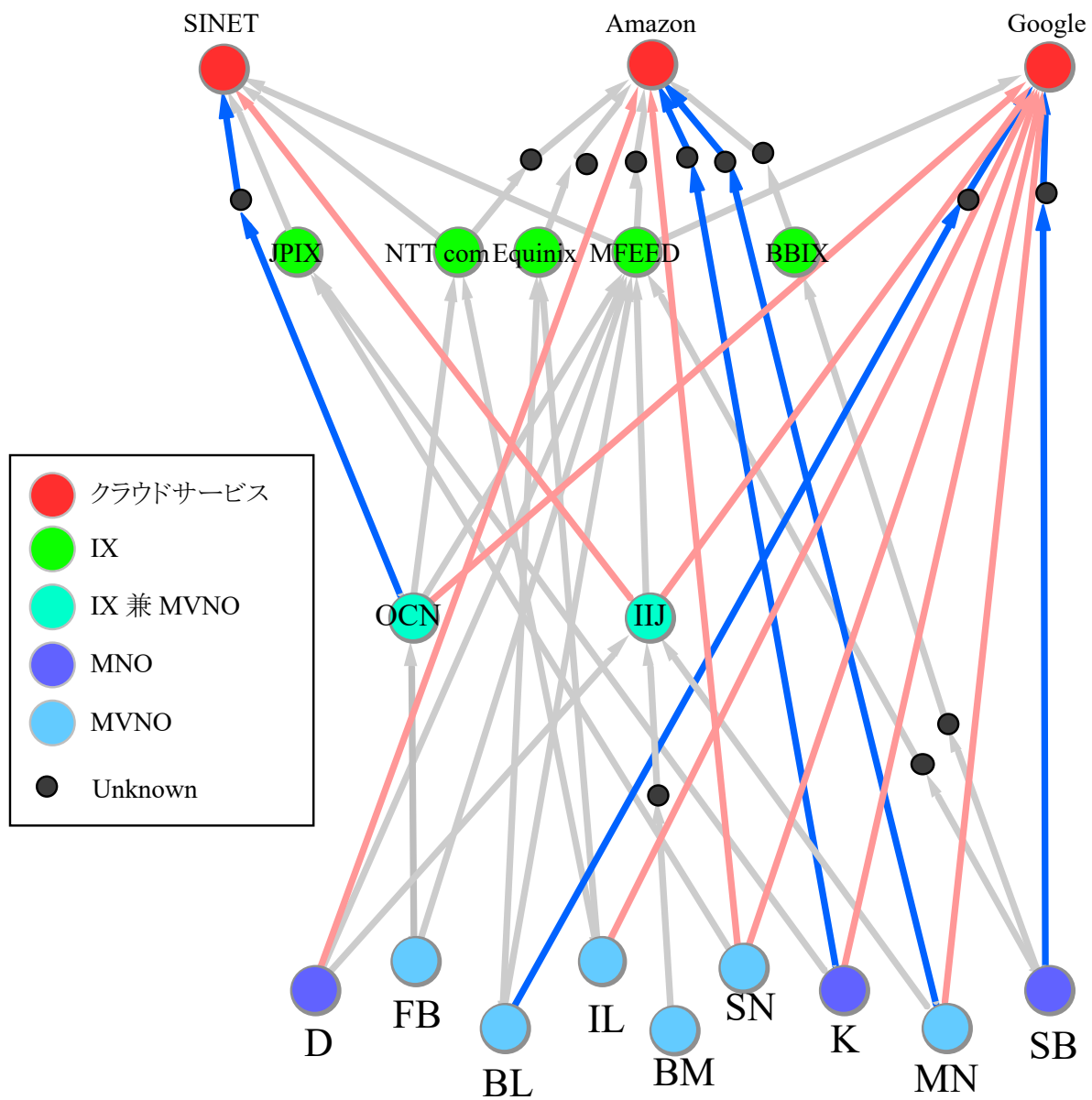


図 5.2: ネットワークトポロジー

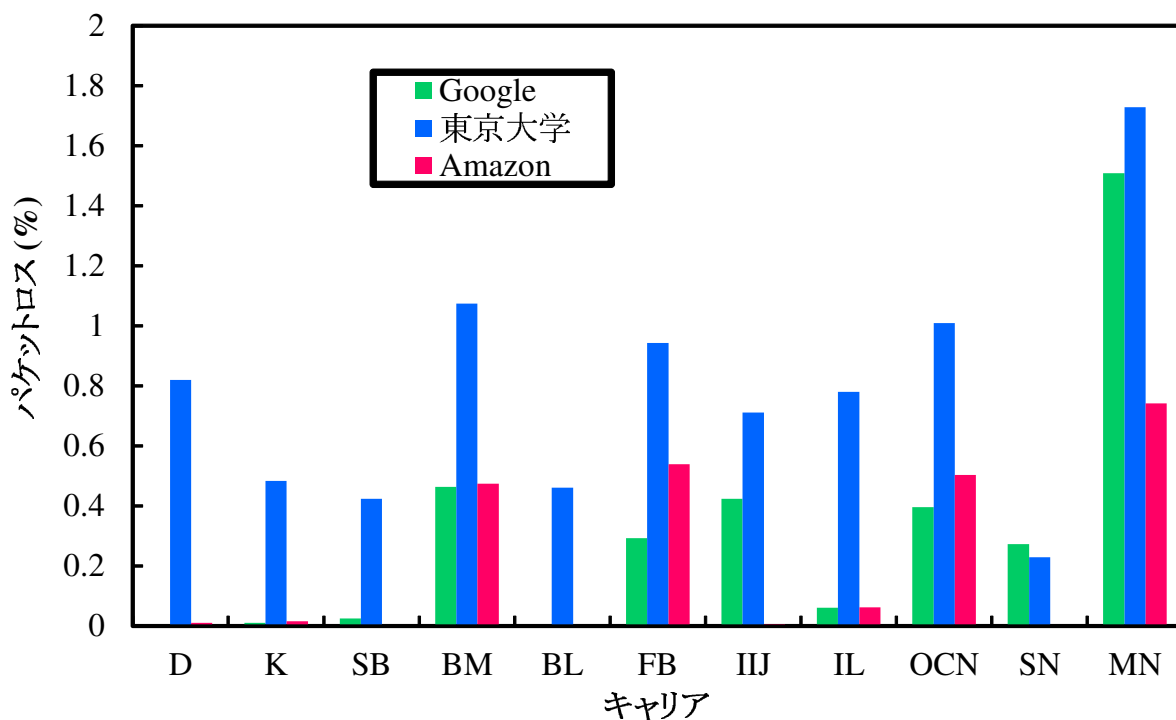


図 5.3: パケットロスの比較

5.4.2 パケットロス

図 5.3 に ping より導いたキャリア-クラウドサービス間のパケットロスの計測結果を示す。横軸がキャリアの名称を、縦軸がパケットロスの大きさ (%) を示す。各キャリアから Google, 東京大学, Amazon に対して ping を行った際のパケットロスを緑, 青, 赤で示す。

図 5.3 を見ると、キャリアから東京大学への通信経路にて多くのパケットロスが発生していることが確認できる。この事は東京大学のサーバが接続されている学術情報ネットワーク SINET が原因となりパケットロスが発生していることを示す。その一方で Google や Amazon への計測では、MNO の 3 社がほぼパケットロスを発生させていないのに対して、MVNO だとパケットロスを発生させることが多い。キャリアによってパケットロスの発生に差があるため、Google や Amazon への計測によるパケットロスはキャリアネットワークが原因だと考えることができ、「MNO はパケットロスを発生させにくい安定したネットワークを持つこと」、「一部の MVNO キャリアがパケットロスを発生させやすい通信経路を持つこと」が確認できる。

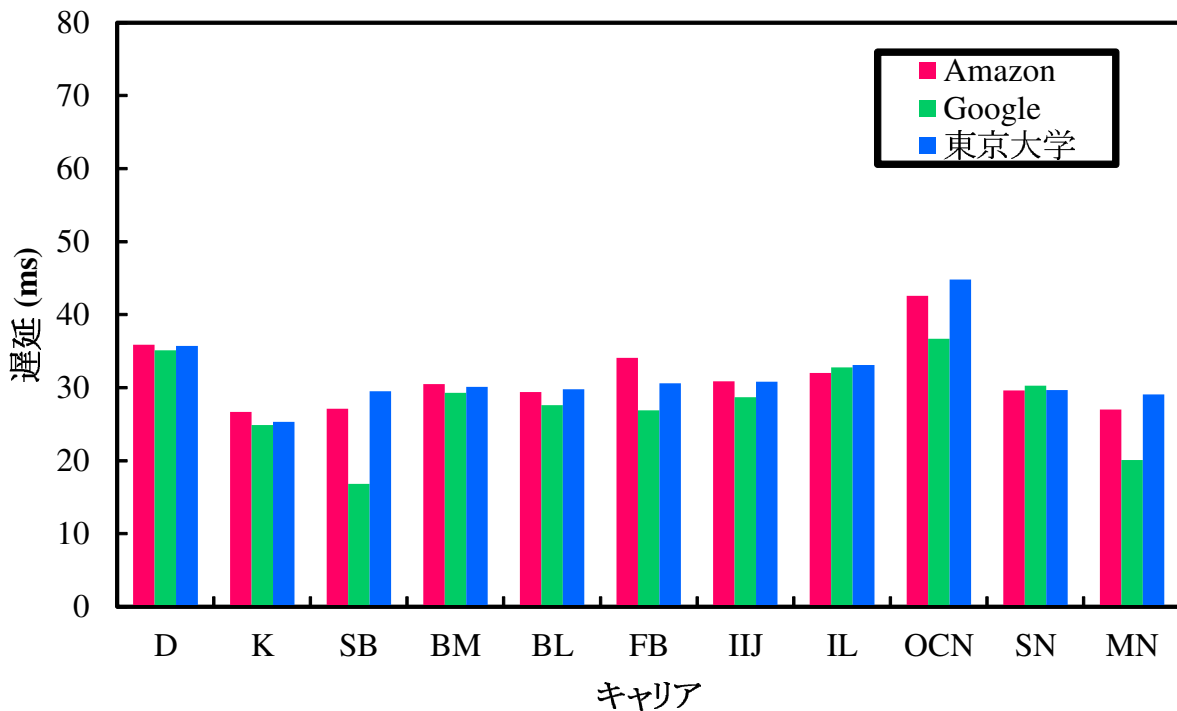


図 5.4: 最低遅延

5.4.3 通信遅延

図 5.4 に ping によるキャリア-クラウドサービス間測定での最低遅延の値を示す。横軸がキャリアを、縦軸が最低遅延の大きさ (ms) を示す。

図 5.4 を見ると、11 キャリアのうち 9 キャリアで Google が最も小さい最低遅延を持つことが確認できる。24 時間定点計測した中での最低遅延はその通信経路の下限の遅延と考えることができるため、Google は多くのキャリアと直接接続されることにより、Google にアクセスするための低遅延な通信経路を構築していると考えられる。図 5.5 の遅延の中央値を見ても、11 キャリアのうち 7 キャリアで Google が最小の遅延の中央値を持つことが確認できる。以上の結果より、Google が多くの通信経路で最も小さい最低遅延、遅延の中央値を持つことが確認できる。これは図 5.2 で示したように、大規模なクラウドサービスは多くのキャリアとインターネットを経由せずに直接接続する通信経路を持つことで低遅延なネットワークを構築しているためである。

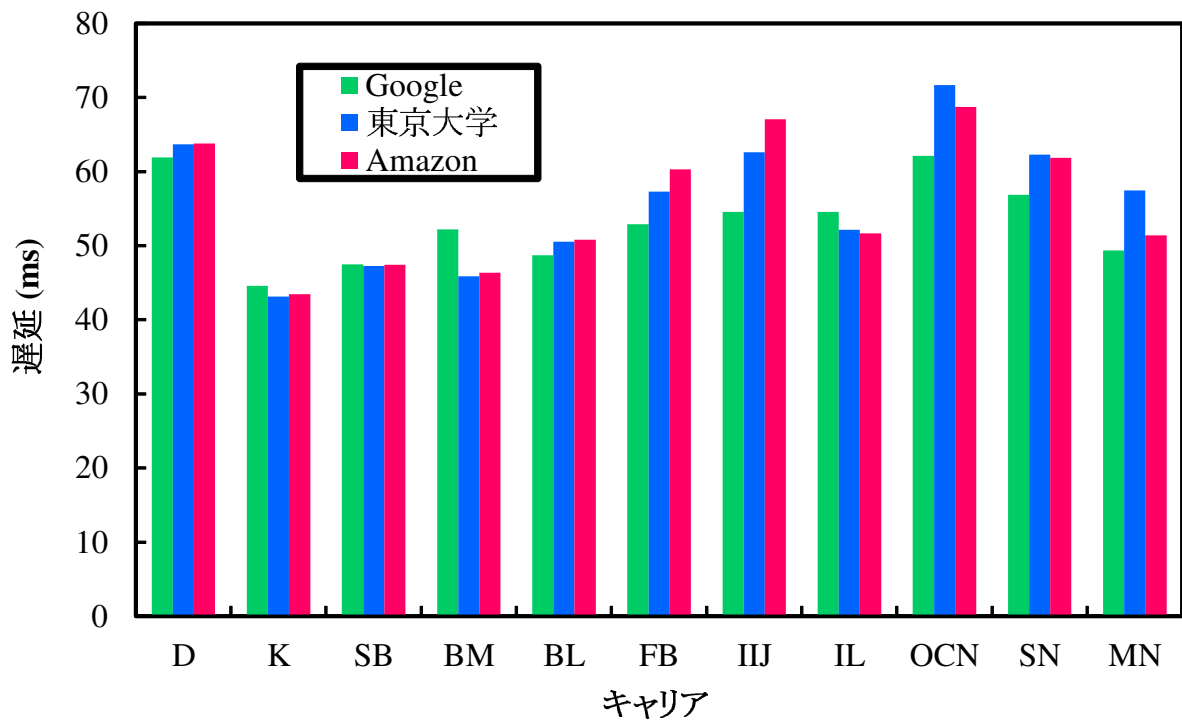


図 5.5: 遅延の中央値

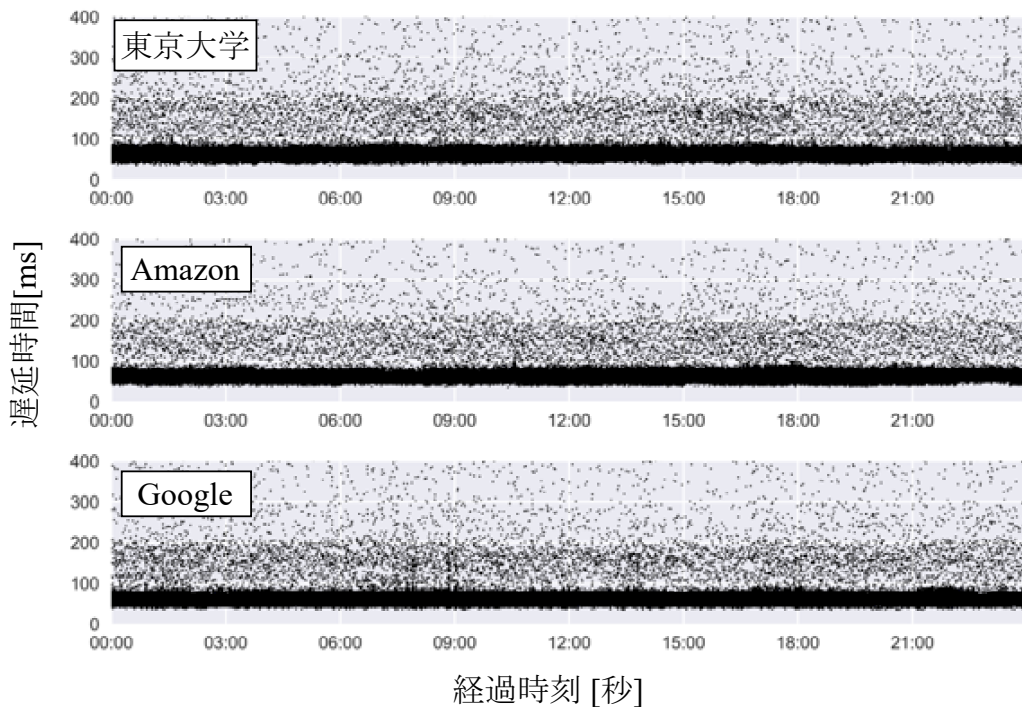


図 5.6: 遅延データ (D)

考察：特徴的な遅延について 図 5.6 に ping を用いた D-クラウドサービス間の時系列遅延計測データを示す. 横軸が経過時間 (秒) を, 縦軸が遅延 (ms) を示す.

図 5.6 を見ると, D の通信計測では経過時間によらず一定の通信遅延の分布を持つことから D が安定した通信を実現できることが分かる. また, 図 5.6 の分布では多くの通信が 60ms 前後の通信遅延を持ち, 大きい通信遅延を持つ通信の頻度は徐々に減少していくことが確認できる. その一方で, 経過時間によって通信遅延の分布が変化したり, 遅延の大きさの頻度分布に偏りが生じたりするキャリアが存在する. 以下では, それらの結果とともに理由を考察する.

FB 図 5.7 に ping を用いた FB-クラウドサービス間の時系列遅延計測データを示す.

図 5.7 を見ると, 12 時過ぎの時間帯と 18 時頃から 24 時手前までの時間帯で, 遅延が大きくなっていることが確認できる. この現象は FB の無線通信回線である D では発生しなかったことから, 無線通信回線が原因ではなく, FB 内のネットワークで負荷が集中する時間帯のパケットを処理しきれなかったことが原因であると考えられる. 同様の現象は, FB 以外にも IJ, MN, OCN, SN で見られた.

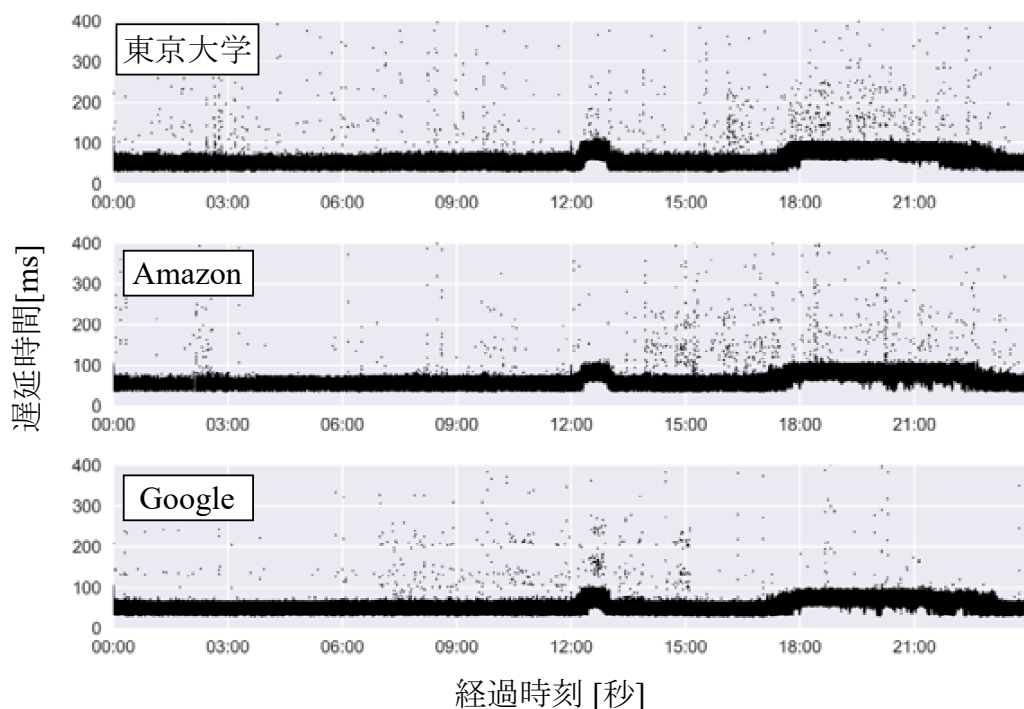


図 5.7: 遅延データ (FB)

K 図 5.8 に ping を用いた K-クラウドサービス間の時系列遅延計測データを示す。図 7 を見ると、多くの通信が 40ms 程度の通信遅延であるが、大きい通信遅延を持つ通信が 100ms 周辺と 200ms 周辺に偏って存在することが確認できる。この現象は、通信負荷に基づいてパケットの転送経路を切り換えることによって発生すると考えられる。同様の現象は、K 以外にも SB, MN, BL, IL, OCN で確認できる。

転送経路の切換箇所は遅延分布より推測できる。D では遅延分布の偏りが発生していないのに拘らず、D の無線通信回線を用いる BL, IL, OCN では遅延分布の偏りが発生している。このことから、これら 3 キャリアについては各キャリア内のネットワークで転送経路の切換を行っていることが推測される。一方で、K と K の無線通信回線を用いる MN では同様の遅延分布の偏りを持つ。このことから、これら 2 キャリアについては K の無線通信回線内で経路の切換を行っていることが推測される。

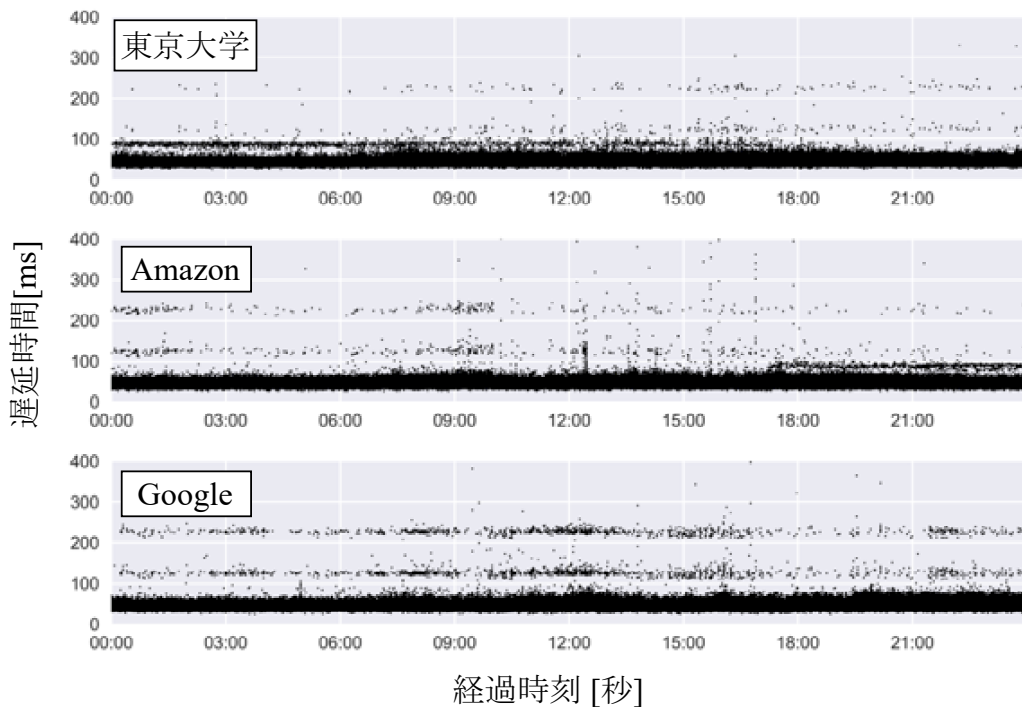


図 5.8: 遅延データ (K)

5.5 各ネットワーク装置上でのロケーションパラメータについて

計測結果から, 3 章で示した各ネットワーク装置上での UpES, LoES のロケーションパラメータ α_{up}, α_{lo} を決定する.

図 5.9 に ping による遅延の中央値と, traceroute の 1 ホップ目の遅延の中央値を示す (traceroute の 1 ホップ目が計測できたデータ¹だけ表示). 横軸がキャリア名, クラウドサービス名を, 縦軸が遅延の大きさ (ms) を示す.

移動体通信網では端末-ゲートウェイ間のネットワークは暗号化・カプセル化されて通信が行われている. そのため, traceroute では「端末-局舎間」および、「端末-ゲートウェイ間」の通信遅延を観測することはできず, traceroute の 1 ホップ目はゲートウェイの次のネットワーク装置との遅延となる. そこで, 我々は traceroute の 1 ホップ目の遅延を端末-ゲートウェイ間の遅延と考えロケーションパラメータを決定する. ping の中央値と traceroute の 1 ホップ目の中央値の比率は「0.862」となるため, 我々はこの 0.862 をゲートウェイのロケーションパラメータとする. また, 本

¹traceroute は拒否されて通信遅延が計測できないことがある

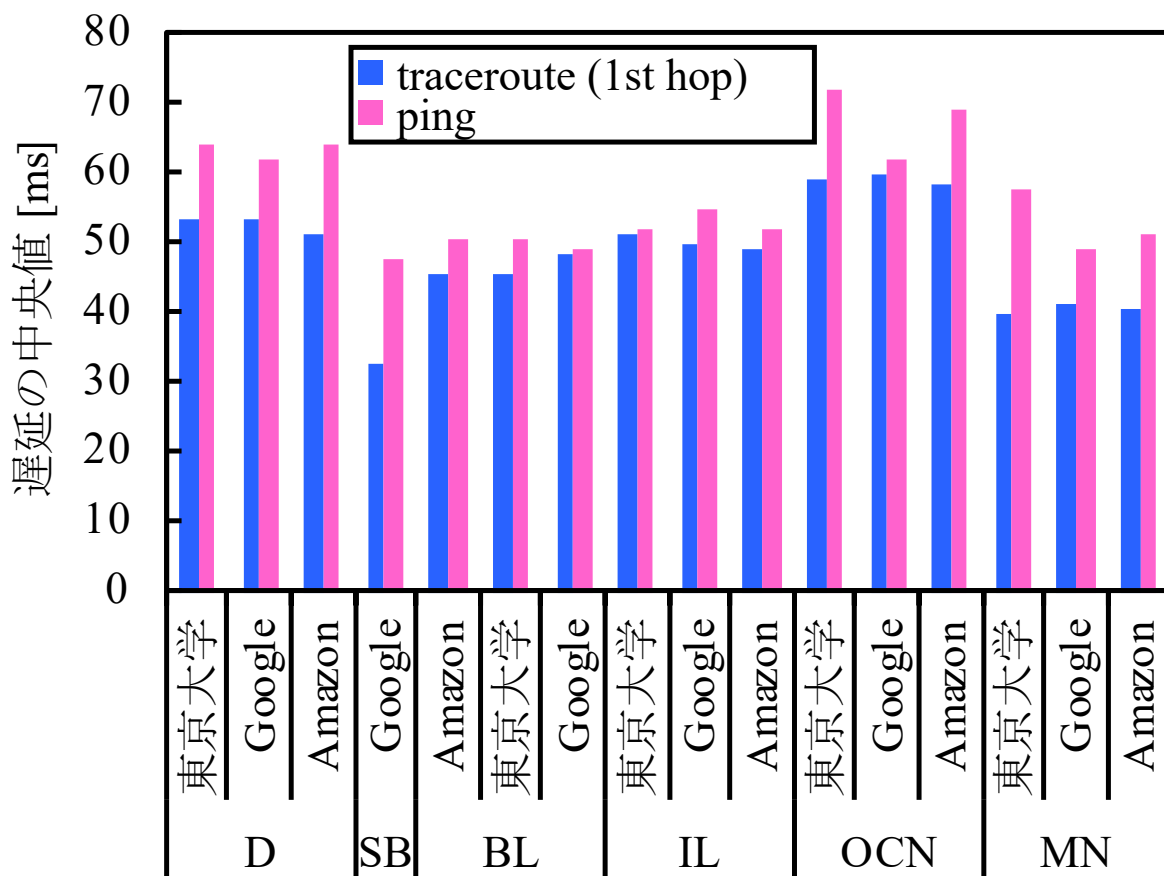


図 5.9: 遅延の比較 (ping, traceroute)

論文では局舎の配置は基地局とゲートウェイの中間の位置にあると仮定し、0.862 の半分の値である「0.431」を局舎のロケーションパラメータとする。表 5.3 に各ネットワーク装置のロケーションパラメータをまとめる。

5.6 まとめ

本章では、3つのクラウドサービスと 11 のキャリアの通信端末を用いて合計 33 の通信経路に対して 24 時間の通信環境計測を行った。

traceroute を用いた測定より、大規模なクラウドサービスはインターネットを経由することなく、キャリアと直接接続するパスを持つ。特に、Google は多くのキャリアと直接接続されているため、Google にアクセスするための低遅延ネットワークが構築されていることを示した。

表 5.3: ロケーションパラメータ α_{up}, α_{lo}

装置名	α_{up}, α_{lo}
基地局	0
局舎	0.431
ゲートウェイ	0.862
クラウド	1.0

ping を用いた測定より, MNO3 社はパケットロスを発生させにくい安定したネットワークを持つことを示した. また, キャリア-クラウド間の遅延計測結果の中に 2 つの特徴的な遅延計測結果が存在し, 「時間帯によって通信遅延の大きさが変化する通信経路」, 「通信遅延の分布に偏りが存在する通信経路」を示した.

最後に, ping と traceroute の結果から, 移動体通信網内のゲートウェイと局舎のロケーションパラメータを推定した.

次章では, 本測定結果を用いた提案プラットフォームのシミュレーション評価を行う.

第 6 章

実トラフィックを用いた提案プラットフォームのシミュレーション評価

6.1 はじめに

本論文では、超広範囲の協調運転システムを実現するために、インフラ型車両制御プラットフォームを提案している。我々は、3章でプラットフォームとそれを配置するネットワークトポロジの提案を行った。また、4章では、マイクロカーによるテストベッドを用いた車両制御の評価を行い、安定して遠隔制御・自律制御の切換を実現できる提案プラットフォームの切換閾値を導き、実通信遅延環境でも安定した制御が実現できることを示した。

一方で、5章で示したように、国内には様々なキャリアが存在するため、それらのネットワーク特性も考慮し、提案プラットフォームの評価を行う必要がある。4章のテストベッド評価は提案プラットフォームを用いた安定制御の確認が目的であったため、短時間の通信測定結果を用いた評価しか行っていない。安定制御可能な切換閾値を実通信環境で適応して、多くの時間帯で自律制御が必要となる場合、超広範囲の協調運転システムを実現することはできない。超広範囲の協調運転システムの実現可能性を示すためには、1つのキャリアだけではなく複数のキャリアの通信遅延データを用いて、長時間の車両制御について評価する必要がある。

また、4章では、あらかじめ UpES をクラウド、LoES を基地局に決めて検討を行ってきたが、UpES、LoES の配置についても評価を行う必要がある。2018 年時点で、キャリアからエッジサーバの具体的な配置計画は発表されていない。我々は、移動体通信網内の UpES、LoES の配置候補として「基

地局」,「局舎」,「ゲートウェイ」,「クラウド」を想定している. 3.6 節で示したように, UpES, LoES の配置が異なる場合,「車両-エッジサーバ間の通信遅延」と「収容台数」,「必要なエッジサーバの数」が異なる.

そこで,本章では,5 章で計測した通信遅延を用いて提案プラットフォームのシミュレーション評価を行う. シミュレーションでは, UpES と LoES の配置を「クラウド」,「ゲートウェイ」,「局舎」,「基地局」の間で変化させ,計測した全通信経路にて「UpES による遠隔制御(以降, UpES 制御)」,「LoES による遠隔制御(以降, LoES 制御)」,「自動車による自律制御(以降,自律制御)」の制御比率(24 時間中で,各制御手法が車両制御を担当した時間の割合)を評価する. また,シミュレーションと 3.6 節の見積もりから,エッジサーバ配置のトレードオフを検証し,提案プラットフォームにおける最適なエッジサーバの配置を考察する.

本章の主要な貢献は,下記である.

- 網羅的に計測した遅延データを用いた国内通信環境での制御比率の導出
- エッジサーバの配置変更による,制御比率と収容台数および,エッジサーバ数の間にあるトレードオフの解決
- 最適なエッジサーバ配置の提案

シミュレーションの結果,第一に制御比率について,平均的な通信経路では,「UpES を最大の収容台数を持つクラウドに配置しても 90%以上の時間帯で制御が可能であること」,「LoES を局舎・基地局に配置することで自律制御の制御比率を 0.6%にまで抑制できること」を示す. 第二に,エッジサーバの配置について「UpES の配置を調整することで LoES-UpES 間の制御比率と収容台数のトレードオフを調整できること」,「LoES の配置を調整することで遠隔制御と自律制御の制御比率とエッジサーバ数のトレードオフを調整できること」を示す. 第三に,自律制御の必要性について「UpES, LoES の配置を調整しても,突発的なバーストパケットロスには抑制できないため,自律制御が必須であること」を示す. 最後に, UpES と LoES の最適配置について「UpES をクラウドに, LoES を局舎に配置することで,エッジサーバ配置のコストを抑制しつつ,最大の収容台数と低遅延を両立する提案プラットフォームを実現できること」を示す.

本章は全 7 節で構成されている. 6.2 節では前章までで導いたシミュレーションで用いるパラメータをまとめるとともに,エッジサーバの配置に関するトレードオフを示す. 6.3 節ではシミュ

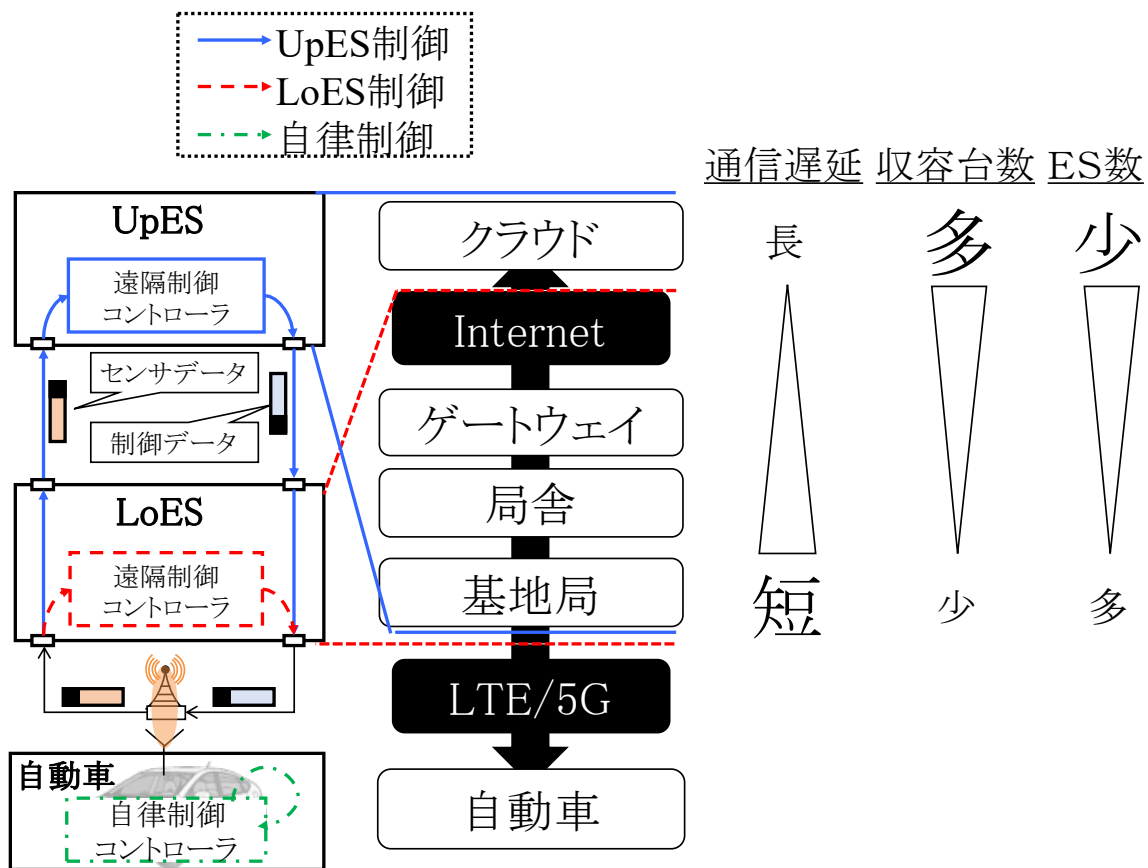


図 6.1: エッジサーバの配置とトレードオフ

レーション評価方法を示し, 6.4 および, 6.5 節でシミュレーション結果を評価し, 6.6 節で最適なエッジサーバ配置について考察する. 6.7 節で本章をまとめる.

6.2 エッジサーバ配置のトレードオフ

図 6.1 に UpES, LoES の配置と各ネットワーク装置の特徴を示す.

UpES はクラウド, ゲートウェイ, 局舎および, 基地局を, LoES はゲートウェイ, 局舎および, 基地局を配置先の候補とする. ただし, UpES は LoES の上位に配置される.

エッジサーバが配置されるネットワーク装置は, 自動車-ネットワーク装置間の「通信遅延」, 各ネットワーク装置がセンサ情報を収集可能な範囲に大きな影響を与える「収容台数」, システム構築のコストに大きな影響を与える「エッジサーバ数 (図 6.1 中では, ES 数と表記)」について異なる特徴を持つ. ネットワークの上位の装置ほど, 自動車から離れるため通信遅延は大きくなる. そ

の一方で, 収容台数が多くなるため, 広範囲の車両から情報収集が可能になり, 必要なエッジサーバ数が少なくなるため, 設備コストが少ない. 反対に, 下位のネットワーク装置ほど, 車両に近い
ため, 通信遅延は小さく安定する. しかし, 収容台数は少なくなり狭い範囲でしか協調運転が実現できず, 多くのエッジサーバ数, ひいては設備コストを必要とする.

表 6.1 に各ネットワーク装置でのロケーションパラメータ, エッジサーバ数, 収容台数を示す. ロケーションパラメータは 3.6 節で定義された遅延モデルで使用されるパラメータで, 5 章で計測した通信端末-クラウドサービス間遅延を通信端末と各ネットワーク装置間の遅延に変換するために使用される. 各装置のロケーションパラメータの値は 5.5 節の考察より決定された. ロケーションパラメータが小さいほど, 遅延が小さくなる. エッジサーバ数および, 収容台数は 3.6 節で見積もられた値である. 表 6.1 より, 「上位のネットワーク装置ほど, 通信遅延は大きくなるものの, 収容台数は多くなり, 必要なエッジサーバ数が少なくなること」, 「下位のネットワーク装置ほど, 通信遅延は小さくなるものの, 収容台数は少なくなり, 必要なエッジサーバ数も多くなること」が確認できる.

次節以降では, 本節で示したロケーションパラメータ, エッジサーバ数, 収容台数を用いてシミュレーション評価および, 考察を行う.

表 6.1: ロケーションパラメータ α_{up} , α_{lo} , エッジサーバ数, 収容台数について

装置名	α_{up}, α_{lo}	エッジサーバ数	収容台数
クラウド	1.0	1	3,415,090
ゲートウェイ	0.862	55	62,093
局舎	0.431	5,640	606
基地局	0	51,076	67

6.3 シミュレーション評価

提案プラットフォームの制御切替についてシミュレーション評価を行い, 「UpES による遠隔制御」, 「LoES による遠隔制御」および, 「自律制御」の制御比率を評価する. 評価には 5 章で計測した実遅延データを用い, UpES と LoES 配置は基地局からクラウドの間で変更する.

シミュレーションのパラメータを表 6.2 に示す. L_{hth} , L_{lth} , N_{burst} は 3 章で定義され, 4 章で導いた提案プラットフォームによって安定して制御を切り換えるための閾値である. 各閾値はそれぞれ

表 6.2: シミュレーションパラメータ

パラメータ名	値
遅延閾値 L_{hth}	100 [ms]
遅延閾値 L_{lth}	70 [ms]
バーストパケットロス閾値 N_{burst}	3
UpES の配置	クラウド, ゲートウェイ, 局舎, 基地局
LoES の配置	ゲートウェイ, 局舎, 基地局, 無
ロケーションパラメータ α_{lo}, α_{up}	1.0...0.0
無線遅延 L_{wl}	10 [ms]
遅延とパケットロスの境界	200[ms]

れ, ネットワーク上位の制御装置から下位の制御装置への切換に使用する遅延閾値 L_{hth} , 下位の制御装置から上位の制御装置への切換に使用する遅延閾値 L_{lth} , 遠隔制御と自律制御の切換に使用するバーストパケットロス閾値 N_{burst} である.

UpES の配置はクラウド, ゲートウェイ, 局舎, 基地局のいずれかである. LoES の配置はゲートウェイ, 局舎, 基地局に加え, LoES を配置しない場合の評価も行う. ロケーションパラメータ α_{lo}, α_{up} と無線遅延 L_{wl} は 3.6 で定義された遅延モデルのパラメータで, α_{lo}, α_{up} は表 6.1 に示された値をエッジサーバの配置に応じて変更し, L_{wl} は 5G 環境を想定し 10ms とする.

また, 本シミュレーションでは遅延データ中のパケットロス以外にも, UpES 制御中は $l_{V2L} + l_{L2U}$ が, LoES 制御中は l_{V2L} が 200ms を超えた場合, パケットロスと判定する.

6.4 シミュレーション結果 (全体): 制御比率, 収容台数, エッジサーバ数

図 6.2, 6.3 に UpES, LoES の配置を変更した際の制御比率の変化を, 図 6.4 に UpES の収容台数を, 図 6.5 にエッジサーバ数の変化を示す. 各図の縦軸, 横軸は UpES, LoES の配置を示し, CL がクラウド, GW がゲートウェイ, CO が局舎, BS が基地局への配置を示す. 図 6.2 は全通信経路でのシミュレーション結果の平均制御比率を示す. 図 6.3 は全通信経路の中で UpES 制御の制御比率が最小となる最も信頼性の低い通信経路のシミュレーション結果を示す. Up, Lo, V はそれぞれ UpES 制御, LoES 制御, 自律制御を示す. グレーになっている領域は, LoES 制御無しで自律制御と UpES 制御だけを用いてシミュレーションを行った結果を示す.

UpES 配置	CL	Up: 93.1% Lo: 6.3% V: 0.5%	Up: 92.8% Lo: 6.6% V: 0.6%	Up: 94.1% Lo: 3.0% V: 2.9%	Up: 92.8% V: 7.2%
	GW	Up: 97.3% Lo: 2.2% V: 0.5%	Up: 97.3% Lo: 2.2% V: 0.5%	Up: 97.1% V: 2.9%	
	CO	Up: 99.5% Lo: 0.0% V: 0.5%	Up: 99.5% V: 0.5%		
	BS	Up: 99.5% V: 0.5%			
		BS	CO	GW	CL
		LoES 配置			

図 6.2: 制御比率 (平均) Up:UpES 制御, Lo:LoES 制御, V:自律制御

6.4.1 超広範囲の協調運転の実現可能性について

図 6.2, 6.3, 6.4 より, UpES をクラウドに, LoES を局舎・基地局に配置することで, 「低遅延」かつ「広範囲」の遠隔制御を実現できることが確認できる。

まず, 図 6.2 より, 平均的な通信経路では, UpES の配置に拘らず, 90%以上の時間帯で遠隔制御が実現可能であることが確認できる。このことは, 移動体通信網でも定常時なら十分 UpES 制御が実現可能であることを示す。また, UpES がクラウドに配置され LoES が存在しない場合, 自律制御の制御比率は 7.2%となる一方で, LoES を局舎より下位に配置することで自律制御の制御比率が 0.6%になることが確認できる。このことは UpES までの通信が不安定なときでも, LoES を使用することで安定した遠隔制御を実現できることを示す。

次に, 図 6.3 は, 通信の信頼性が最も低い通信経路でのシミュレーション結果である。UpES を上位のネットワーク装置に配置した場合, UpES の制御比率が低くなるため, 広範囲の情報を使用して制御できる時間が短くなることが分かる。その一方で, UpES をクラウドに配置していたとしても, LoES を局舎より下位に配置していれば自律制御の制御比率を 0.8%以下に抑制することができる。このことは, 信頼性の低い通信経路では, UpES による広範囲の遠隔制御の効果は小さくなるものの, LoES が低遅延処理を実現することで, 安定した遠隔制御を実現できることを示す。

UpES 配置	CL	Up: 72.2% Lo: 27.8% V: 0.0%	Up: 61.1% Lo: 38.1% V: 0.8%	Up: 72.3% Lo: 4.4% V: 23.2%	Up: 64.3% V: 35.7%
	GW	Up: 77.8% Lo: 22.2% V: 0.0%	Up: 77.0% Lo: 22.9% V: 0.1%	Up: 76.7% V: 23.3%	
	CO	Up: 97.7% Lo: 0.0% V: 2.3%	Up: 97.7% V: 2.3%		
	BS	Up: 97.7% V: 2.3%			
		BS	CO	GW	CL
	LoES 配置				

図 6.3: 制御比率 (最小) Up:UpES 制御, Lo:LoES 制御, V:自律制御

最後に, 図 6.4 は, UpES の配置が収容台数に大きな影響を与えることが示している. 特に, クラウドに配置された UpES は提案プラットフォームの中で最大の収容台数を持ち, 平均的な通信経路では 90%以上の時間帯で UpES 制御を行うことができる.

超広範囲の協調運転を実現するためには, 「低遅延」かつ「大きな収容台数」を持つ遠隔制御システムを構築する必要がある. このシミュレーション結果は, 提案プラットフォームを使用することで「低遅延」と「大きな収容台数」の両立ができることを示しており, ひいては超広範囲の

UpES 配置	CL	3,415,090	3,415,090	3,415,090	3,415,090
	GW	62,093	62,093	62,093	
	CO	606	606		
	BS	67			
		BS	CO	GW	CL
	LoES 配置				

図 6.4: UpES の収容台数

UpES 配置	CL	51,071	5,641	56	1
	GW	51,131	5,695	55	
	CO	56,716	5,640		
	BS	51,076			
	BS	CO	GW	CL	
LoES 配置					

図 6.5: エッジサーバ数

協調運転に必要な要件を満たすことを示す。

6.4.2 UpES 配置の影響

また、図 6.2, 6.3, 6.4 より、UpES の配置が車両に近づくほど、LoES による制御比率が抑制され、UpES による制御比率が増加するものの、収容台数が小さくなることが確認できる。このことは、UpES の配置を調整することで UpES と LoES 間の制御比率と収容台数のトレードオフを調整できることを示す。

6.4.3 LoES 配置の影響

図 6.5 を見ると、LoES が車両から離れた位置に配置されるほど、エッジサーバ数が少なくなり、ひいてはコストを抑制できることが確認できる。その一方で、図 6.2 を見ると、UpES をクラウドに配置し、LoES を局舎に配置すると自律制御の比率が 0.6%であるのに対し、LoES をゲートウェイに配置すると自律制御の比率が 2.9%にまで増加する。また、UpES をクラウドに配置し、LoES を配置しない場合は、自律制御の比率が 7.2%にまで増加する。これらのことは、LoES の配置を調整することで、遠隔制御と自律制御の制御比率とエッジサーバ数のトレードオフを調整できることが分かる。

以上より、超広範囲の協調運転システムの実現に必要な、「低遅延」かつ「大きな収容台数」である遠隔制御システムを提案プラットフォームによって構築できることを示した。また、UpES と

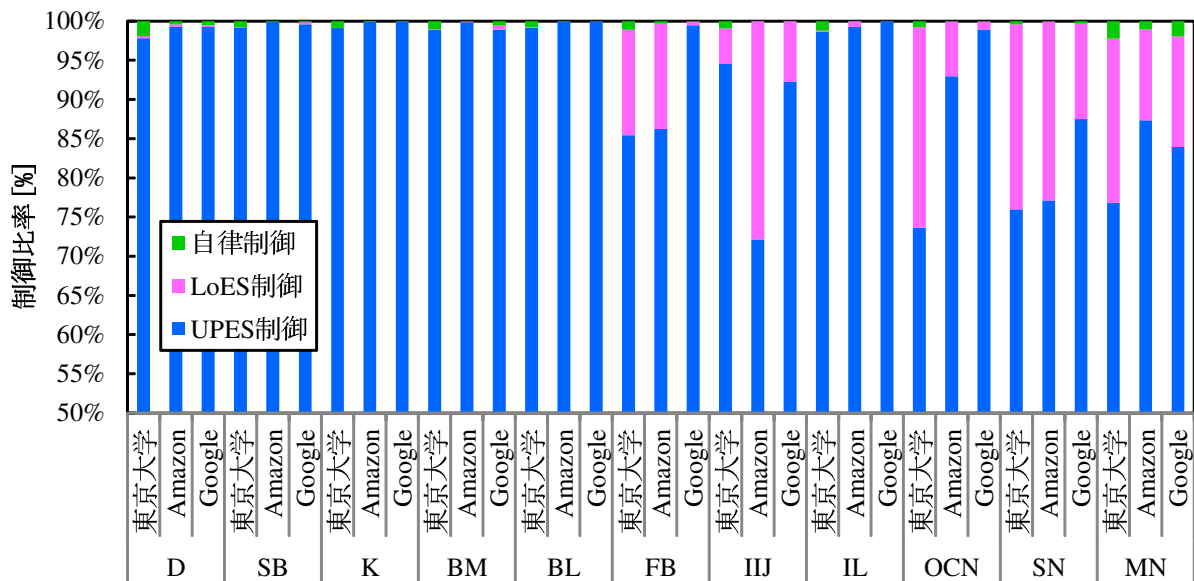


図 6.6: 制御比率 (UpES=クラウド, LoES=基地局)

LoES の配置から、制御比率とエッジサーバ数、収容台数のトレードオフを調整できることを示した。以降では、キャリア別制御比率を検証する。

6.5 シミュレーション結果 (キャリア別) : 制御比率

図 6.6 に UpES をクラウドに LoES を基地局に配置した際の制御比率を示す。横軸は通信経路のキャリアとクラウドサービスの略称 (表 5.2 参照) を示し、縦軸が制御比率を示す。青色のバーが UpES 制御の、赤色のバーが LoES 制御の、緑色のバーが自律制御の比率を示す。縦軸の制御比率が 50%-100% を表示しているのは、全ての結果で 50% 以上の制御が UpES による遠隔制御であったためである。

図 6.6 を見ると、通信経路が異なると、大きく制御比率が異なることが確認できる。いくつかの通信回線が不安定なキャリアでは、安定した制御を行うために、自律制御や LoES 制御が多くなる傾向にある。また、33 の通信経路のうち 28 経路が 80% 以上、23 経路が 90% 以上 UpES 制御が実現できることが確認できる。このことから、UpES をクラウドに配置したとしても、多く時間帯を UpES より遠隔車両制御できることが分かる。

以降は図 6.6 と比較して、UpES, LoES の配置が制御比率に与える影響を比較する。

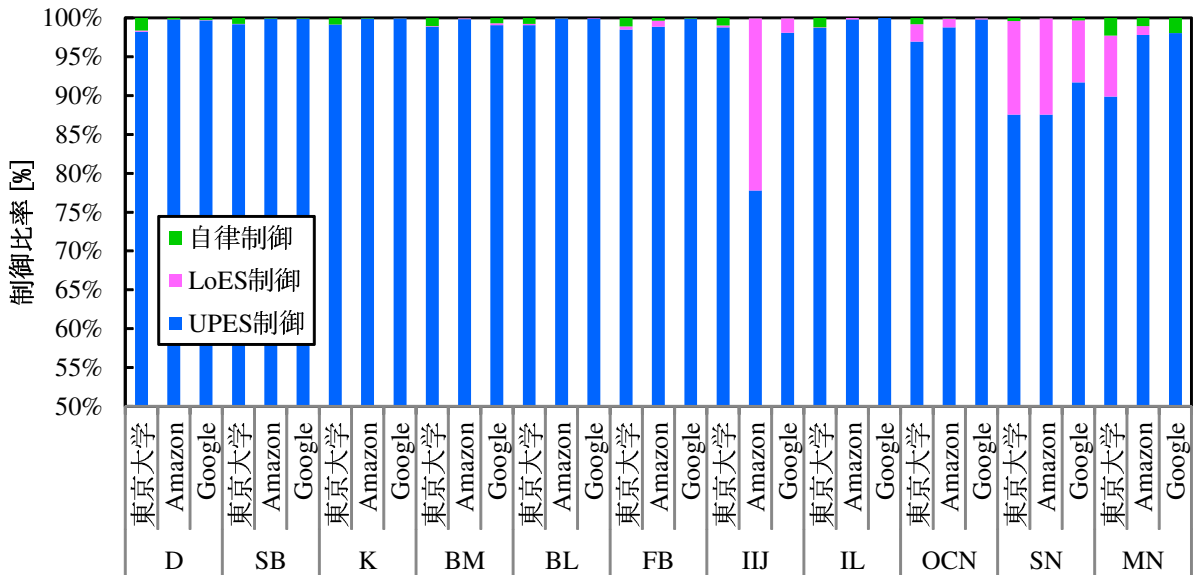


図 6.7: 制御比率 (UpES=ゲートウェイ, LoES=基地局)

6.5.1 UpES の配置効果 (LoES=基地局)

図 6.7, 図 6.8 は UpES をゲートウェイ, 局舎に配置した時の制御比率を示す. 図 6.7 より UpES をゲートウェイに配置すると, 33 経路のうち 32 経路が 80%以上, 30 経路が 90%以上クラウドから制御でき, 図 6.8 より UpES を局舎に配置すると, ほぼ全てを UpES から制御できることが確認できる. その一方で, 図 6.6, 6.7, 6.8 の間で自律制御の比率は大きく変化しない. これらのことから, UpES の配置が UpES と LoES による遠隔制御の比率に大きく影響を与え, UpES の配置を下位のネットワーク装置に変更することで, 遅延が抑制され UpES 制御の比率が増加することが確認できた.

一例として, 図 6.9 に FB-東京大学間の通信遅延を用いた, シミュレーション結果の詳細を示す. 横軸が経過時間 (24 時間) を, 縦軸が遅延の大きさを示す. 黒の点が遅延モデルに基づいて計測遅延から導出した自動車-UpES 間遅延の大きさを示す. 遅延が負の領域にある青, 赤, 緑の点はそれぞれ UpES 制御, LoES 制御, 自律制御の時間帯を示す. 図 6.9(a), (b), (c) はそれぞれ UpES がクラウド, ゲートウェイ, 基地局に配置されている.

図 6.9(a) を見ると, バーストパケットロスが原因で瞬間的に UpES 制御から, LoES 制御, 自律制御に切り換わるケースと, 通信負荷の影響によって遅延が大きくなる時間帯で長期的に LoES 制御に移行するケースが確認できる. 図 6.9(a) と比較して (b) を見ると, UpES をクラウドからゲ-

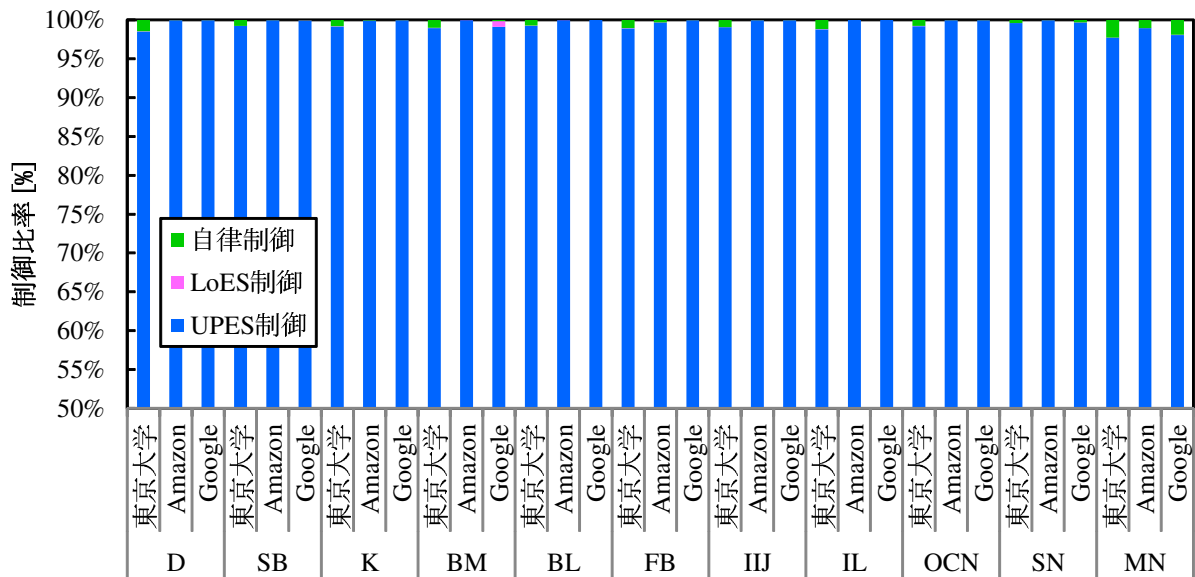


図 6.8: 制御比率 (UpES=局舎, LoES=基地局)

トウェイへと移動させることで、通信遅延が抑制され、8:30 頃と夜間の LoES 制御の多くの部分が、UpES 制御を実現できるようになったことが確認できる。さらに、図 6.9(c) より、UpES を局舎に移動させることで、突発的なバーストパケットロスを除くと LoES 制御できることが確認できる。言い換えると、このことは、突発的なバーストパケットロスの回避には自律制御が必須であることを示す。

以上より、「UpES の配置を下位のネットワーク装置に変更することで、遅延が抑制され UpES 制御の比率が増加すること」、「突発的なバーストパケットロスの回避には自律制御が必須であること」を示した。

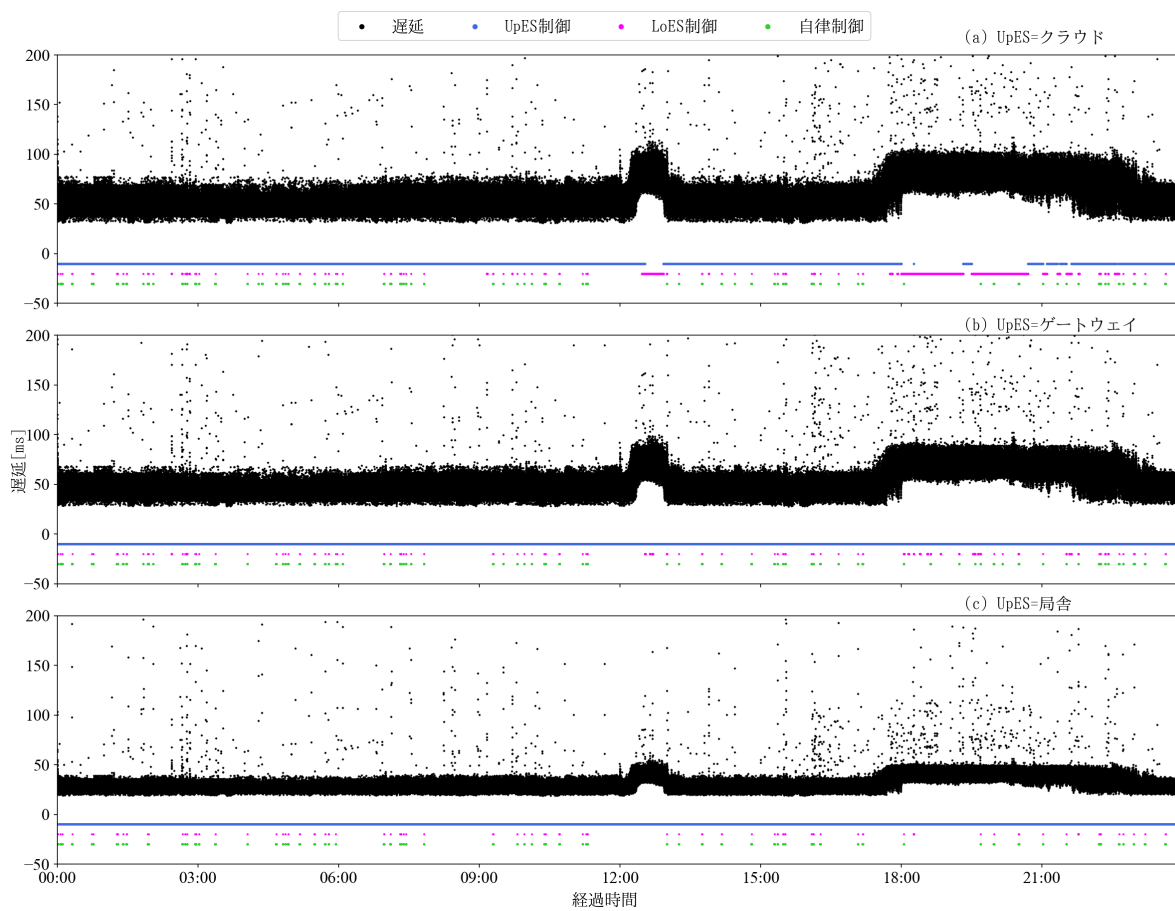


図 6.9: FB - 東京大学の制御比率 (LoES=基地局)

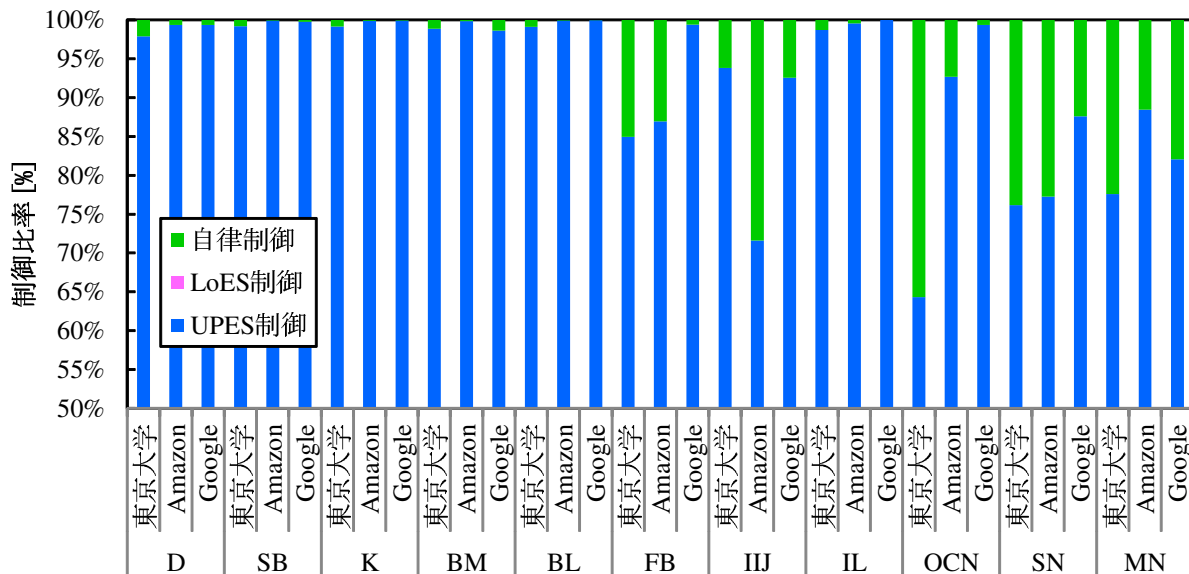


図 6.10: 制御比率 (UpES=クラウド, LoES=無し)

6.5.2 LoES の配置効果 (UpES=クラウド)

図 6.10, 6.11, 6.12 はそれぞれ LoES が存在しない場合, ゲートウェイに配置されている場合, 局舎に配置されている場合の制御比率を示す (UpES はクラウドに配置).

図 6.10 より, 33 経路の内 5 経路で 20%以上の, 10 経路で 10%以上の自律制御が必要になる. これらはいずれも MVNO の通信経路であるため, MVNO の通信経路で, LoES を配置することなく超広範囲の協調運転システムを実現することが困難であることを示す.

その一方で, 図 6.11 では 1 経路で 20%以上の, 5 経路で 10%以上の自律制御が必要になり, 図 6.12 では全通信経路にて, 図 6.6 と同等まで自律制御を抑制できている. このことは, 局舎は遠隔制御を実現する上で十分小さな遅延を持つことを示す.

一例として, 図 6.13 に FB-東京大学間の通信遅延を用いた, シミュレーション結果の詳細を示す. 図 6.13(a), (b), (c) はそれぞれ LoES が「無し」, 「ゲートウェイ」, 「局舎」に配置される場合の結果を示す. 黒の点は, 図 6.9 とは異なり, 遅延モデルに基づいて計測遅延から導出した自動車-LoES 間遅延の大きさを示す (ただし, 図 6.13 (a) は LoES が無いため, 自動車-UpES 間遅延を表示する).

図 6.13(a) では, 遅延が大きくなる時間帯は全て自律制御を行う. それに対して, 図 6.13(b) では, 遅延の大きい時間帯の半分程度を LoES 制御が代替し, 図 6.13(c) では, ほぼ LoES 制御が代替する.

以上より, 「LoES を配置することで自律制御を抑制することができ, その配置が遠隔制御と自

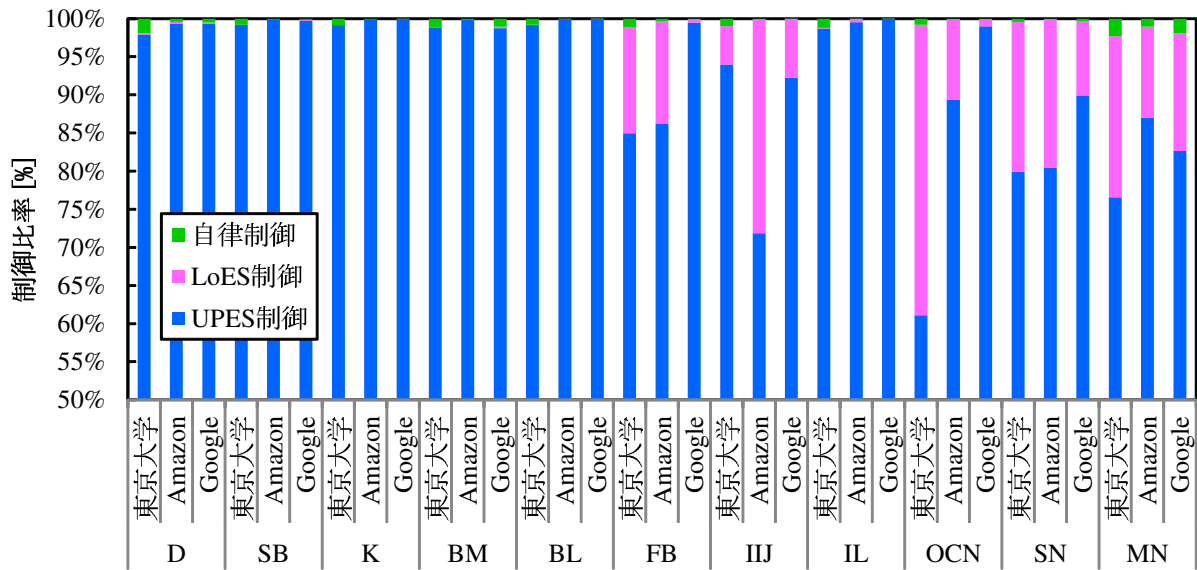


図 6.12: 制御比率 (UpES=クラウド, LoES=局舎)

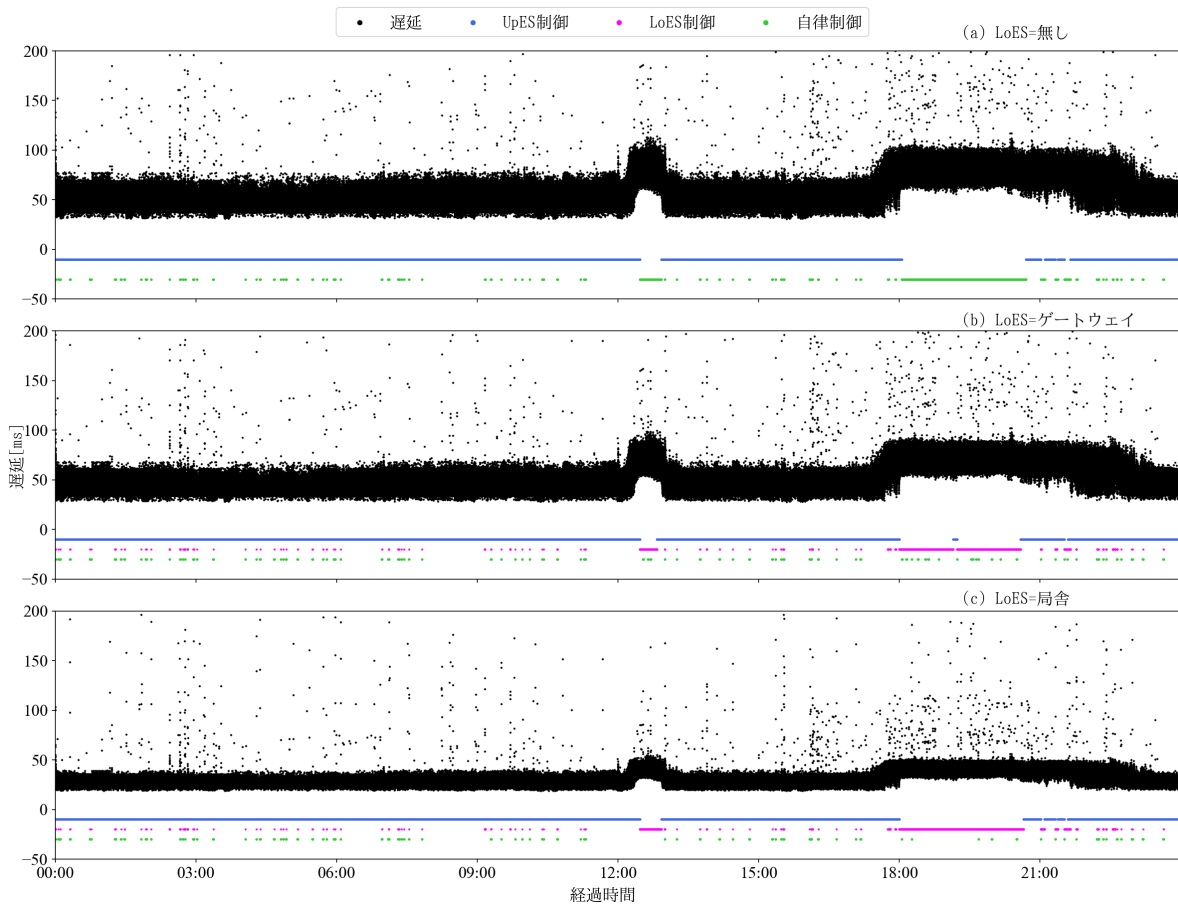


図 6.13: FB - 東京大学の制御比率 (UpES=クラウド)

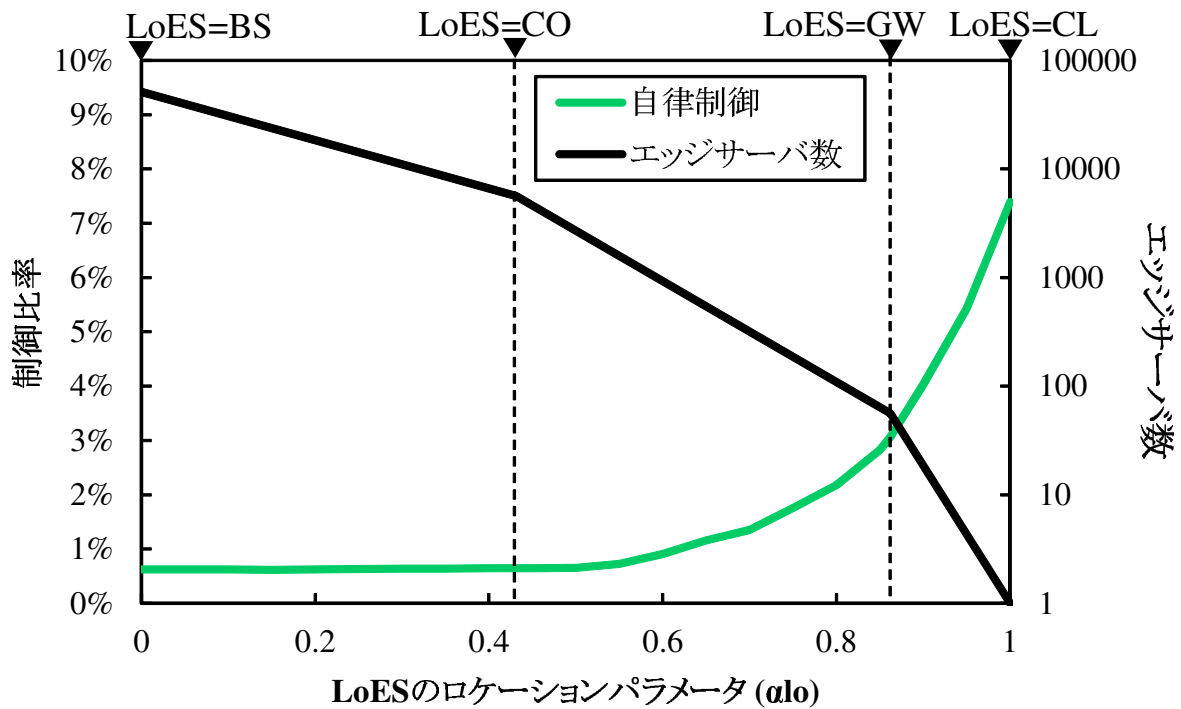


図 6.14: LoES の配置変更時の自律制御比率とエッジサーバ数 (UpES=クラウド)

6.6 考察:最適なエッジサーバ配置について

本節では, 提案プラットフォームが大きな収容台数と低遅延処理を両立できる最適なエッジサーバの配置について考察する.

6.6.1 UpES の配置

測定した全通信経路の平均制御比率 (図 6.2) より, UpES は配置に拘らず 90%以上の制御比率を持つため, その中で最大の収容台数を持つクラウドを UpES の最適な配置とする.

6.6.2 LoES の配置

LoES の配置を決定するために, UpES の配置をクラウドに固定したうえで, LoES のロケーションパラメータ (α_{lo}) を 0.0 から 1.0 まで 0.05 刻み¹で変化させて制御比率を評価する. 図 6.14 に, LoES の配置変更時の自律制御の制御比率とエッジサーバ数を示す. 横軸が LoES のロケーション

¹0, 0.05, ..., 0.4, 0.431, 0.45, ..., 0.85, 0.862, 0.9, 0.95, 1.0

パラメータ (α_{lo}) であり, 左側の縦軸が制御比率を, 右側の縦軸がエッジサーバ数を示す. また, CL がクラウド, GW がゲートウェイ, CO が局舎, BS が基地局に LoES が配置された時の結果を示す.

図 6.14 を見ると, α_{lo} が 0.5 以下なら, LoES を基地局に配置した場合と同等の自律制御比率となることが確認できる. その一方で, 必要となるエッジサーバ数は LoES が基地局に配置される場合と比較して, 局舎に配置される場合 11%, ゲートウェイに配置される場合 0.1% の台数となる. ただし, ゲートウェイに配置される場合, α_{lo} が 0.5 を超えており, 自律制御の比率が増加している. 超広範囲の協調運転を実現するためには可能な限り自律制御の比率を下げる必要があるため, 本論文では, 最適な LoES の配置を局舎とする.

以上より, UpES をクラウドに, LoES を局舎に配置することで, エッジサーバ配置のコストを抑制しつつ, 最大の収容台数と低遅延を両立する提案プラットフォームを実現できることを示した.

6.7 まとめ

広範囲の協調運転システムを構築するために, 様々な通信キャリアの実通信遅延を用いて提案プラットフォームのシミュレーション評価を行った.

シミュレーションの結果, 「平均的な通信経路では UpES を最大の収容台数を持つクラウドに配置しても 90% 以上の時間帯で制御が可能であること」, 「LoES を局舎・基地局に配置することで自律制御の制御比率を 0.6% にまで抑制できること」から超広範囲の協調運転システムの実現可能性を示した.

次に, キャリア毎のシミュレーション結果より, 「同じエッジサーバの配置だと, その通信経路の特性によって制御比率は大きく異なること」, 「UpES の配置を調整することで, UpES と LoES 間の制御比率と収容台数のトレードオフを調整できること」, 「LoES の配置を調整することで, 遠隔制御と自律制御の制御比率とエッジサーバ数のトレードオフを調整できること」, 「UpES, LoES の配置を調整しても, 突発的なバーストパケットロスには抑制できないため, 自律制御が必須であること」を確認した.

最後に, 最適な UpES/LoES の配置について検討し, UpES をクラウドに, LoES を局舎に配置することで, エッジサーバ配置のコストを抑制しつつ, 最大の収容台数と低遅延を両立する提案プラットフォームを実現できることを示した.

次章では, 本博士論文の結論をまとめる.

第7章

結論

本論文では, 超広範囲の協調運転システムを構築するための3つの課題

- プラットフォームに関する課題
- プラットフォームの評価に関する課題
- 実通信環境への適応に関する課題

を提示し, それぞれを

- プラットフォーム提案
- テストベッド構築・評価
- 実通信環境の測定・評価

により解決を行うことで, 超広範囲の協調運転システムの実現可能性を示した. 以下では, まず本論文の総論をまとめ, 続けて今後の課題について述べる.

7.1 まとめ

7.1.1 プラットフォーム提案

提案する「インフラ型車両制御プラットフォーム」は, 多層にエッジサーバを配置する「ネットワークアーキテクチャ」と, その多層エッジサーバによる遠隔制御と自動車による自律制御を

連携させた「車両制御アーキテクチャ」によって構成される。本論文では、多層エッジサーバは UpES, LoES の 2 層で構成され、それらのエッジサーバによる遠隔制御と自動車の自律制御が連携して、車両制御を実現する。

また、2 層のエッジサーバによる遠隔制御と自動車の自律制御を、通信の状況に応じて切り換える手法も提案した。提案する切手手法では、超広範囲の協調運転システムを実現するためには、可能な限り上位のエッジサーバから制御を行い、車両の自律制御を抑制する。

最後に、提案プラットフォームが動作するネットワークトポロジについても検討し、移動体通信網内の各ネットワーク装置の「通信遅延」、「エッジサーバ数」、「収容台数」に関するトレーオフを統計情報と通信モデルより導いた。

7.1.2 テストベッド構築・評価

提案プラットフォームが安定して車両を制御できるか評価するために、複数のエッジサーバからマイクロカーの遠隔制御を実現できるテストベッドを構築し、評価を行った。

テストベッドは実車の 1/10 スケールのマイクロカーを用いる MPT と、UpES および、LoES を実現する FLARE, UpES と LoES の間で実通信遅延を再現する WAN エミュレータで構成されている。

テストベッドを用いた評価は「基本特性評価」と「実通信特性評価」の 2 種類を行った。基本特性評価では、WAN エミュレータ上で「定常遅延」、「ランダムパケットロス」、「バーストパケットロス」を加え、遠隔制御の基本的な通信負荷に対する特性を評価した。基本特性評価により、下記の結果を得た。

- 車両-エッジサーバ間の定常遅延が 100ms 以下なら大きな走行軌跡のずれが発生することなくエッジサーバから車両の遠隔制御を実現できる
- バーストパケットロスの期間が 400ms 以下なら、エッジサーバから安定した遠隔制御が実現できる
- ランダムパケットロスが 10%以下なら、エッジサーバから安定した遠隔制御を実現できる

また、以上の結果より、提案プラットフォームが安定して遠隔制御・自律制御を切替可能な遅延閾値 ($L_{hth} = 100, L_{lth} = 70$) とバーストパケットロス閾値 ($N_{burst} = 3$) を決定した。

実通信特性評価では、携帯端末とクラウドサービスの間で終日通信データを計測し、計測データの中で遅延の変化が激しい時間帯の通信遅延を WAN エミュレータで再現し、遠隔制御の評価を行った。実通信特性評価により、次のような結果を得た。

- 基本特性評価で導いた切換閾値に用いて、UpES/LoES を切換えながら遠隔制御を行うことで、LoES だけの遠隔制御と同等の安定制御を実現できる

7.1.3 実通信環境の測定/評価

実通信環境を考慮した評価を行うために、様々なクラウドサービス、キャリアを使用して実通信遅延データを測定し、提案プラットフォームの車両制御シミュレーションを行った。

実通信環境の測定では、3つのクラウドサービスに対して、11のキャリアの通信端末で実遅延データを測定し、パケットロスと通信遅延について評価した。評価の結果、以下の結果を得た。

- 大規模なクラウドサービスはキャリアとインターネットを経由することなく直接接続するパスを持つ
- MNO の3社はパケットロスを発生させにくい安定したネットワークを持つ
- 通信負荷が集中する時間帯で、大きく遅延が増加する通信経路が存在する
- 通信遅延の分布に時間帯による偏りが存在する通信経路が存在する

車両制御シミュレーションでは、UpES と LoES の配置を移動体通信網内で変化させ、計測した全通信経路にて「UpES 制御」、「LoES 制御」、「自律制御」の制御比率を評価した。また、シミュレーション結果とエッジサーバ数と収容台数の見積もり結果から、エッジサーバ配置のトレードオフを検証した。

評価の結果、以下のような結果を得た。

- UpES を最大の収容台数を持つクラウドに配置しても、平均的な通信経路では、90%以上の時間帯で制御が可能である
- LoES を局舎・基地局に配置することで、平均的な通信経路では、自律制御の制御比率を0.6%にまで抑制できる

- UpES の配置を調整することで UpES と LoES 間の制御比率と収容台数のトレードオフを調整できる
- LoES の配置を調整することで、遠隔制御と自律制御の制御比率とエッジサーバ数のトレードオフを調整できる
- UpES, LoES の配置を調整しても、突発的なバーストパケットロスは抑制できないため、自律制御が必須である

最後に、最適な UpES/LoES の配置について検討し、UpES を「クラウド」に、LoES を「局舎」に配置することで、エッジサーバ配置のコストを抑制しつつ、最大の収容台数と低遅延を両立する提案プラットフォームを実現できることを示した。

以上をまとめると、3つの課題に対して、

- 超広範囲の協調運転システムを実現するためのプラットフォームを提案
- テストベッドによる評価から、提案プラットフォームが安定して動作する条件を確立
- シミュレーションによる評価から、超広範囲のセンサ情報収集と低遅延通信を両立可能な提案プラットフォームの実環境配置の導出

を実現したことにより、超広範囲の協調運転システムの実現可能性を示した。

7.2 今後の課題

今後の課題を以下に示す。

- 多層エッジサーバモデルの拡張
- 5G 環境での評価
- CORD[47] 上での実装
- ローカル 5G[76] との連携

7.2.1 多層エッジサーバモデルの拡張

本論文では、多層エッジサーバの中で一番単純なモデルである2層エッジサーバを検討し、複数レイヤにエッジサーバがある場合の効果を検証してきた。その一方で、設備投資コスト等が許容できるならば、多段にエッジサーバを配置し、遠隔制御の切り換えを行う方が柔軟な対応が可能である。また、本ネットワークモデルでは有線ネットワークのアーキテクチャを単純化し4層（基地局-局舎-ゲートウェイ-クラウド）のネットワークアーキテクチャとしたが、実際には4層より多くのネットワーク装置が車両-クラウド間ネットワーク中に存在する。そこで、本プラットフォームを社会に適應するためには、2層のモデルを多層に拡張、定式化し、状況に応じてエッジサーバ数を変更できる汎用的な車両制御アーキテクチャを構築する。

7.2.2 5G 環境での評価

本論文のテストベッド評価では、無線通信に Wi-Fi を使用した。しかし、実環境に適應する場合、無線通信は 5G となる。5G の URLLC を考慮すると、通信遅延は Wi-Fi と同程度かそれ以下となることが想定されるが、実際に 5G 環境を導入し遠隔制御の成立性について評価を行う必要がある。

7.2.3 CORD 上での実装

本論文では、UpES をクラウドに LoES を局舎に配置することが、広範囲の協調運転システムを実現するうえで最適であると結論づけた。その一方で、現在、局舎の重要性が再認識されており、CORD（Central Office Rearchitected as a Datacenter）[47] と呼ばれる開発プロジェクトに注目が集まっている。CORD は、元来ハードウェアで構成される局舎設備を、データセンタとソフトウェアに置き換えることを目標としている。本プラットフォームを CORD で動作する形に実装すれば、クラウド-局舎という提案する配置と同様の環境でプラットフォームを動作させることが可能になる。

7.2.4 ローカル 5G との連携

本論文の提案プラットフォームを用いて、超広範囲の協調運転システムを実社会で試行するためには、5G 通信を利用したサービス提供は必要不可欠である。その一方で、携帯事業者による全国系のサービス展開では、IoT や自動運転のような個別のニーズに対応できない。

そこで現在, ローカル 5G[76] に注目が集まっている. ローカル 5G は地域ニーズや個別ニーズに応じて, 様々な主体が 5G を活用したシステムを導入できる制度であり, 携帯事業者以外の様々な企業や地方自治体が 5G を用いたシステムを構築することが可能になる.

ローカル 5G を利用して本システムを試行していくことで, 超広範囲の協調運転システムの社会実装につなげたい.

以上の課題に加えて, 本研究で対象とした超広範囲の協調運転システムのように, 今後, さらに複雑化するシステムを実現するためには, 横断的に物事をとらえることが重要である. このことを意識しながら, 今後の研究開発活動, 技術情報の発信に努めたい.

参考文献

- [1] 外務省. SDGs とは？ <https://www.mofa.go.jp/mofaj/gaiko/oda/sdgs/about/index.html>, 2015.
- [2] TRI. <https://www.tri.global/>, (accessed: 2018-12-03).
- [3] Waymo. <https://waymo.com/>, (accessed: 2018-12-03).
- [4] Apple. <https://www.christianpost.com/news/apple-self-driving-car-news-research-paper-reveals-tech-giants-autonomous-vehicle-software-208090/>, 2017.
- [5] Panasonic. 福井県、永平寺町とパナソニックが自動運転車両走行の実証実験を「永平寺参（まい）ろード」で実施. <https://news.panasonic.com/jp/press/data/2017/11/jn171113-3/jn171113-3.html>, (accessed: 2019-02-20).
- [6] 三菱電機. <http://www.mitsubishielectric.co.jp/me/synergy/01-autonomous-driving/>, (accessed: 2019-02-20).
- [7] K. Abboud, H. A. Omar, and W. Zhuang. Interworking of dsrc and cellular network technologies for v2x communications: A survey. *IEEE Transactions on Vehicular Technology*, 65(12):9457–9470, Dec 2016.
- [8] Steven E. Shladover. Recent international activity in cooperative vehicle–highway automation systems. Technical report, 2012.
- [9] L. Chen and C. Englund. Cooperative intersection management: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):570–586, Feb 2016.
- [10] Network Architecture Committee 5GMF. Network technologies for 5g. *The Fifth Generation Mobile Communications Promotion Forum*, 2015.

- [11] M. Agiwal, A. Roy, and N. Saxena. Next generation 5g wireless networks: A comprehensive survey. *IEEE Communications Surveys Tutorials*, 18(3):1617–1655, thirdquarter 2016.
- [12] International Telecommunication Union. ITU: Committed to connecting the world. <http://www.itu.int/en/Pages/default.aspx>, (accessed: 2018-12-03).
- [13] M Series. Imt vision–framework and overall objectives of the future development of imt for 2020 and beyond. *Recommendation ITU*, pages 2083–0, 2015.
- [14] Yun Hu, Chao, Milan Patel, Dario Patel, M, Nurit Sprecher, and Valerie Young. Mobile edge computing a key technology towards 5g. *White Paper, Mobile-edge Computing (MEC) industry initiative*, September 2015.
- [15] Alex Reznik, Rohit Arora, Mark Cannon, Luca Cominardi, Walter Featherstone, Rui Frazao, Fabio Giust, Sami Kekki, Alice Li, Dario Sabella, et al. Developing software for multi-access edge computing. *ETSI, White Paper*, (20), 2017.
- [16] Automotive Edge Computing Consortium. <http://aecc.org/>, (accessed: 2018-12-03).
- [17] 鈴木 有也, 佐々木 健吾, 佐藤 健哉, 高田 広章. クラウド型自動運転を指向したストリーム処理型 Idm の低遅延処理手法. In *組込みシステムシンポジウム 2015 論文集*, volume 2015, pages 84–92, oct 2015.
- [18] Lei Kang, Wei Zhao, Bozhao Qi, and Suman Banerjee. Augmenting self-driving with remote control: Challenges and directions. In *Proceedings of the 19th International Workshop on Mobile Computing Systems & Applications*, HotMobile '18, pages 19–24, New York, NY, USA, 2018. ACM.
- [19] Masatoshi Ito and Masakazu Ikeda. Vehicle control using edge computing and mobile network. *23rd ITS World Congress*, 2016.
- [20] Nokia. NOKIA Introduced 5G Autonomous Car. <http://us.aving.net/news/view.php?articleId=1365627>, (accessed: 2018-12-03).

- [21] K. Sasaki, S. Makido, and A. Nakao. Vehicle control system for cooperative driving coordinated multi-layered edge servers. In *2018 IEEE 7th International Conference on Cloud Networking (CloudNet)*, pages 1–7, Oct 2018.
- [22] K. Sasaki, N. Suzuki, S. Makido, and A. Nakao. Layered vehicle control system coordinated between multiple edge servers. In *2017 IEEE Conference on Network Softwarization (NetSoft)*, pages 1–5, July 2017.
- [23] 佐々木 健吾, 牧戸 知史, 中尾 彰宏. 協調運転実現のための多層エッジサーバによる車両制御システム ～遠隔車両制御と自律制御の連携～. 電子情報通信学会技術研究報告, IEICE-118(124):161–166, 2018.
- [24] 佐々木 健吾, 牧戸 知史, 中尾 彰宏. 協調運転実現のための多層エッジサーバによる車両制御システム ～インターネット遅延計測と計測結果を用いた車両制御評価～. 電子情報通信学会技術研究報告, IEICE-117(459):375–380, 2018.
- [25] K. Sasaki, N. Suzuki, S. Makido, and A. Nakao. Vehicle control system coordinated between cloud and mobile edge computing. In *2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pages 1122–1127, Sep. 2016.
- [26] 佐々木 健吾, 牧戸 知史, 鈴木 有也, 中尾 彰宏. Mobile edge computing を利用したインフラ型車両制御システム ～通信異常時の走行性能評価～. 電子情報通信学会技術研究報告, IEICE-115(483):323–328, 2016.
- [27] Md Whaiduzzaman, Mehdi Sookhak, Abdullah Gani, and Rajkumar Buyya. A Survey on Vehicular Cloud Computing. *J. Netw. Comput. Appl.*, 40:325–344, April 2014.
- [28] Mohamed Eltoweissy, Stephan Olariu, and Mohamed Younis. Towards autonomous vehicular clouds. In Jun Zheng, David Simplot-Ryl, and Victor C. M. Leung, editors, *Ad Hoc Networks*, pages 1–16, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
- [29] Dejan Dimitrijevic, Nemanja Nedic, and Vladimir Dimitrieski. Real-time carpooling and ride-sharing: Position paper on design concepts, distribution and cloud computing strategies. In *Com-*

- puter Science and Information Systems (FedCSIS), 2013 Federated Conference on*, pages 781–786. IEEE, 2013.
- [30] Andrew M Amey. *Real-time ridesharing: exploring the opportunities and challenges of designing a technology-based rideshare trial for the MIT community*. PhD thesis, Massachusetts Institute of Technology, 2010.
- [31] S. Ancona, R. Stanica, and M. Fiore. Performance boundaries of massive floating car data of floating. In *2014 11th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, pages 89–96, April 2014.
- [32] R. Stanica, M. Fiore, and F. Malandrino. Offloading floating car data. In *2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pages 1–9, June 2013.
- [33] Zubaida Alazawi, Mohmmad B. Abdljabar, Saleh Altowaijri, Anna Maria Vegni, and Rashid Mehmood. Icdms: An intelligent cloud based disaster management system for vehicular networks. In Alexey Vinel, Rashid Mehmood, Marion Berbineau, Cristina Rico Garcia, Chung-Ming Huang, and Naveen Chilamkurti, editors, *Communication Technologies for Vehicles*, pages 40–56, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [34] Lionel Nkenyereye and Jong-Wook Jang. A study of big data solution using hadoop to process connected vehicle’s diagnostics data. In Kuinam J. Kim, editor, *Information Science and Applications*, pages 697–704, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.
- [35] E. Lee, E. Lee, M. Gerla, and S. Y. Oh. Vehicular cloud networking: architecture and design principles. *IEEE Communications Magazine*, 52(2):148–155, February 2014.
- [36] K. Lee, J. Kim, Y. Park, H. Wang, and D. Hong. Latency of cellular-based v2x: Perspectives on tti-proportional latency and tti-independent latency. *IEEE Access*, 5:15800–15809, 2017.
- [37] Hideki Shimada, Akihiro Yamaguchi, Hiroaki Takada, and Kenya Sato. Implementation and evaluation of local dynamic map in safety driving systems. *Journal of Transportation Technologies*, 5(02):102, 2015.

- [38] Swarun Kumar, Lixin Shi, Nabeel Ahmed, Stephanie Gil, Dina Katabi, and Daniela Rus. Carspeak: A content-centric network for autonomous driving. In *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '12, pages 259–270, New York, NY, USA, 2012. ACM.
- [39] Ruilin Liu, Daehan Kwak, Srinivas Devarakonda, Kostas Bekris, and Liviu Iftode. Investigating remote driving over the lte network. In *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '17, pages 264–269, New York, NY, USA, 2017. ACM.
- [40] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys Tutorials*, 19(4):2322–2358, Fourthquarter 2017.
- [41] Niklas Hehenkamp, Christian Facchi, and Stefan Neumeier. How to achieve traffic safety with lte and edge computing. In Kohei Arai and Rahul Bhatia, editors, *Advances in Information and Communication*, pages 164–176, Cham, 2020. Springer International Publishing.
- [42] K. Zhang, Y. Mao, S. Leng, A. Vinel, and Y. Zhang. Delay constrained offloading for mobile edge computing in cloud-enabled vehicular networks. In *2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM)*, pages 288–294, Sept 2016.
- [43] C. Tang, X. Wei, C. Zhu, W. Chen, and J. J. P. C. Rodrigues. Towards smart parking based on fog computing. *IEEE Access*, 6:70172–70185, 2018.
- [44] R. Shinkuma, S. Kato, M. Kanbayashi, Y. Ikeda, R. Kawahara, and T. Hayashi. System design for predictive road-traffic information delivery using edge-cloud computing. In *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pages 1–6, Jan 2018.
- [45] 加藤 晋, 津川 定之. DGPS を用いた自律車両のための速度制御. 日本機械学会ロボティクス・メカトロニクス講演会, 2000.
- [46] Ministry of Internal Affairs and Communications (in Japanese). <http://www.tele.soumu.go.jp/j/musen/toukei/>.

- [47] L. Peterson, A. Al-Shabibi, T. Anshutz, S. Baker, A. Bavier, S. Das, J. Hart, G. Palukar, and W. Snow. Central office re-architected as a data center. *IEEE Communications Magazine*, 54(10):96–101, October 2016.
- [48] NTT ドコモ. <https://www.nttdocomo.co.jp/>, (accessed: 2018-12-03).
- [49] ソフトバンク. <https://www.softbank.jp/>, (accessed: 2018-12-03).
- [50] KDDI. <https://www.kddi.com/>, (accessed: 2018-12-03).
- [51] Ministry of Internal Affairs and Communications (in Japanese). Publication about trend of mvno use (http://www.tele.soumu.go.jp/main_content/000362104.pdf), 2015.
- [52] 一般財団法人自動車検査登録情報協会. <http://www.airia.or.jp/publish/statistics/number.html>, (accessed: 2018-12-03).
- [53] 日本経済新聞. 5G 基地局に信号機開放 全国 20 万基 23 年度に. <https://www.nikkei.com/article/DGXMZO45619080T00C19A6MM8000/>, (accessed: 2019-07-01).
- [54] 和田 錦一. カメラ映像が遠隔運転のパフォーマンスに与える影響. 第 20 回日本バーチャルリアリティ学会大会論文集, 2015.
- [55] Akihiro Nakao. Revisiting “clean-state” approach to re-designing the internet. In *14th GENI Engineering Conf. Global Environment for Network Innovations*, 2012.
- [56] tc-netem (8) - Linux Man Pages. <https://www.systutorials.com/docs/linux/man/8-tc-netem/>, (accessed: 2018-12-03).
- [57] Nick McKeown. Software-defined networking. *INFOCOM keynote talk*, 2009.
- [58] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee. Network function virtualization: Challenges and opportunities for innovations. *IEEE Communications Magazine*, 53(2):90–97, Feb 2015.
- [59] Robert Morris, Eddie Kohler, John Jannotti, and M. Frans Kaashoek. The click modular router. *SIGOPS Oper. Syst. Rev.*, 33(5):217–231, December 1999.

- [60] 国土交通省. http://www.mlit.go.jp/road/road_e/r1_standard.html, (accessed: 2018-12-03).
- [61] Google. Google public dns. <http://developers.google.com/speed/public-dns/>, (accessed: 2018-12-03).
- [62] 楽天モバイル. <https://mobile.rakuten.co.jp/>, (accessed: 2018-12-03).
- [63] XERA. <https://xera.jp/entry/speedtest>, (accessed: 2018-12-03).
- [64] Rakuten. 携帯キャリア事業への新規参入表明に関するお知らせ. http://corp.rakuten.co.jp/news/press/2017/1214_02.html, (accessed: 2018-12-03).
- [65] WhoIS. <http://www.rfc-editor.org/rfc/rfc3912.txt>, (accessed: 2018-12-03).
- [66] Amazon. Amazon web service. <http://aws.amazon.com/jp/>, (accessed: 2019-02-26).
- [67] 日本通信. <http://www.bmobile.ne.jp/>, (accessed: 2018-12-03).
- [68] ビッグロブ. https://join.biglobe.ne.jp/?cl=head_logo, (accessed: 2018-12-03).
- [69] フリービット. <https://freebit.com/>, (accessed: 2018-12-03).
- [70] インターネットイニシアティブ. <https://www.iijmio.jp/>, (accessed: 2018-12-03).
- [71] インターリンク. <https://www.interlink.or.jp/service/sim/>, (accessed: 2018-12-03).
- [72] NTT コミュニケーションズ. <https://www.ntt.com/personal/services/mobile/one.html>, (accessed: 2018-12-03).
- [73] ソニーモバイルコミュニケーションズ (株). <http://lte.so-net.ne.jp/sim/retail/>, (accessed: 2018-12-03).
- [74] ケイ・オプティコム. <https://mineo.jp/>, (accessed: 2018-12-03).
- [75] 国立情報学研究所 (NII). 学術情報ネットワーク sinet5. <https://www.sinet.ad.jp/>, (accessed: 2019-02-29).
- [76] 総務省. http://www.soumu.go.jp/main_sosiki/joho_tsusin/policyreports/joho_tsusin/5th_generation/local_5g/index.html, (accessed: 2019-02-27).