The University of Osaka
Institutional Knowledge Archive

Title	路車協調システムのための通信プラットフォームに関する研究
Author(s)	伊川, 雅彦
Citation	大阪大学, 2010, 博士論文
Version Type	VoR
URL	https://hdl.handle.net/11094/803
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

路車協調システムのための
通信プラットフォームに関する研究

2010年 1月

伊川 雅彦

大阪大学大学院 工学研究科 電気電子情報工学専攻

謝辞

本論文は，著者が三菱電機株式会社産業システム研究所，先端技術総合研究所在職中ならびに大阪大学大学院工学研究科電気電子情報工学専攻博士後期課程在学中に行った研究を，大阪大学大学院工学研究科電気電子情報工学専攻教授 小牧省三博士の御指導のもとにまとめたものである．本研究を遂行するにあたり，終始懇切丁寧なる御指導，御鞭撻を賜りました小牧省三教授に謹んで深謝の意を表します．

本論文をまとめるに際し，懇切丁寧なる御教示，御助言を賜りました大阪大学大学院工学研究科電気電子情報工学専攻教授 三瓶政一博士，同准教授 塚本勝俊博士に深く感謝の意を表します．

さらに，大阪大学大学院在学中，講義等を通じて通信工学全般ならびに本研究に関して多大なるご指導，ご教示を賜った大阪大学大学院工学研究科教授 滝根哲哉博士，同教授 北山研一博士，同教授 馬場口登博士，同教授 井上恭博士，同教授 河崎善一郎博士をはじめとする大阪大学工学研究科電気電子情報工学専攻情報通信部門の諸先生方，ならびに大阪大学産業科学研究所教授 鷲尾隆博士，同教授 溝口理一郎博士をはじめとする大阪大学産業科学研究所の諸先生方に厚く感謝申し上げます．

三菱電機株式会社自動車機器開発センター主管技師長 熊澤宏之博士，同社先端技術総合研究所システム構築技術部地理情報通信技術グループ グループマネージャー後藤幸夫博士には，著者が入社より今日に至るまで，常に御指導と御鞭撻を賜りました．深く感謝いたします．

筆者が，大阪大学大学院工学研究科電気電子情報工学専攻博士後期課程に在学する上で，終始暖かい御指導と御鞭撻を賜った三菱電機株式会社先端技術総合研究所システム構築技術部部長 武田保孝氏に心から御礼申し上げます．

また筆者を高度道路交通システムの研究に導いて頂くとともに，本研究の過程において多大な御指導，御助言を頂いた元三菱電機株式会社 ITS 推進事業本部副本部長 大石将之氏，三菱電機株式会社自動車機器事業本部技師長 尾崎稔氏，同社先端技術総合研究所専任 古澤春樹氏，同社鎌倉製作所 IT システム部 ITS 技術課主席技師長 津田喜秋氏，三菱電機エンジニアリング株式会社 瀧北守氏に深く感謝いたします．

また，常日頃より熱心な御討論と貴重な御助言，御好意溢れる御支援を賜った三菱電機株式会社の諸先輩，同僚に心より感謝いたします.

最後に，本研究遂行に際し，あらゆる面で筆者を支え励ましてくれた，妻美保，娘葵唯，息子凛太朗に心より感謝いたします.

内容梗概

本論文は，狭域通信 (DSRC:Dedicated Short Range Communication) による多様な路車協調システムの構築を効率化することを目的とした通信プラットフォームに関する研究成果をまとめたものであり，以下の 6 章から構成される．

第 1 章は序論であり，本研究の背景および関連分野の研究動向について概説を行うとともに，本研究の占める位置と研究目的を明らかにする．

第 2 章では，走行中の車両や使用するハードウェアリソースの少ない低リソースな車載システムに対するサービスをターゲットとした DSRC ローカル通信プラットフォームを提案する．本章では，まず対象とするサービスに要求される機能・性能要件をまとめ，それらを実現するためのアーキテクチャと非 IP 系高速通信プロトコルについて述べる．提案するプロトコルは路側機/車載器ローカルでアプリケーションが実行されることを前提とすることで，従来の IP(Internet Protocol) を利用する方式に比べ，高速な初期接続や高効率な通信を実現する．次にこれらのプロトコルを実際の DSRC 路側機と DSRC 車載器上に実装し，通信実験を行うことで，要求されている性能要件を満たすことができるかどうかを評価・検証し，その有効性を示す．

第 3 章では，第 2 章で提案した非 IP 系の通信プロトコル上で，走行車両への情報配信サービスを実現するためのプッシュ型プロトコルを提案する．HTTP (Hyper Text Transfer Protocol) などを利用する既存の IP 系のプッシュ型プロトコルは，常時接続や IP の機能を前提としており，通信可能時間が限られる狭域通信に適用することは難しい．これに対し，提案するプロトコルは，通信可能時間への考慮や様々な機器構成への対応など，狭域通信や車載機器特有の条件を考慮した設計とすることで，DSRC による走行車両への情報配信サービスを実現する．また，実機を用いた検証を行うことで，その動作や通信性能を確認し，実際に車両への多種多様な情報配信サービスに適用可能であることを示す．

第 4 章では，第 2 章および第 3 章で提案した単一の DSRC 路側機を用いたスポット型サービス向けの DSRC ローカル通信プラットフォームを，複数の DSRC 路側機を利用した空間的な広がりのあるサービスに適用可能とし，これらのサービス構築の効率化を実現するため，路側システム側のネットワーク化による拡張方式を提

案する．路側システムをネットワーク化する場合，機器間の通信遅延などにより，初期接続性能や通信効率が低下するという問題が生じる．提案方式では，路側システム側のプロトコル配置や機器間の連携をこれらの通信遅延が最小となるように行うことで，高速な初期接続性能や高効率な通信という DSRC ローカル通信プラットフォームがもつ特徴を継承する．異なる路側機下に存在する車両間でデータを交換する車路車間通信サービスを実装し，その遅延時間などについて評価を行うことで，本方式の有効性を示す．

第 5 章では，通信プラットフォームを用いて構築される様々な路車協調システムの設計・評価に不可欠なシミュレーション環境として，路車協調システムの構成要素である人・道路・車全ての間の協調動作を再現可能な道路交通シミュレータとドライビングシミュレータの統合シミュレータを提案する．このような複数の異なるシミュレータを統合する場合，それぞれで異なるデータ形式やシミュレーション周期をどのように扱うかが課題となる．提案する統合化手法では，高速なデータ変換機構と同期機構を持つ調停プロセスの導入により，異種分散環境として統合化を実現することで，各シミュレータの個別のバージョンアップや別のモジュールの追加が容易な高い拡張性を実現する．またこの統合シミュレータに，路車協調を必要とする ITS サービスをプロトタイプ実装し，動作を検証することにより，本シミュレータの ITS 開発プラットフォームとしての使用可能性について考察し，本方式の有効性を示す．

第 6 章は結論であり，本研究で得られた成果について総括を行う．

目 次

第 1 章 序論 1

1.1 研究の背景と従来研究 1

1.2 本研究の位置付けと目的 4

1.3 論文の構成 7

第 2 章 DSRC の多目的利用を実現する狭域通信環境向け高速通信プロトコル 9

2.1 序言 9

2.2 DSRC ローカル通信プラットフォームのアーキテクチャ 10

2.2.1 設計方針 10

2.2.2 要求条件 12

2.2.3 プロトコルスタック 13

2.3 DSRC ローカルアプリケーションのための通信プロトコル 15

2.3.1 ローカルポート制御プロトコル (LPCP) 15

2.3.2 ローカルポートプロトコル (LPP) 17

2.4 実装と評価 24

2.4.1 LPCP/LPP の実装とタグ型サービスの実現 24

2.4.2 走行車両への通信量の見積り 26

2.5 結言 28

第 3 章 車両へのプッシュ型情報配信サービスに適したアプリケーションプロトコル 30

3.1 序言 30

3.2 プッシュ型サービスのアーキテクチャ 31

3.2.1 要求条件 31

3.2.2 プロトコルスタック 32

3.3 プッシュ型情報配信のためのプロトコル 34

3.3.1 メッセージ定義 35

3.3.2 配信コンテンツの再生制御機能 36

3.3.3 車載リソース情報通知機能 38

3.4 実装と評価 41
3.4.1 再生制御機能の確認 41
3.4.2 走行車両への通信量の見積り 43
3.5 結言 45

第 4 章 複数路側機連携サービスのための DSRC 通信プラットフォーム 47
4.1 序言 47
4.2 複数路側機連携 DSRC 通信プラットフォームのアーキテクチャ . . . 48
4.2.1 サーバー集約方式による複数路側機連携方式 48
4.2.2 プロトコルスタック 49
4.3 複数路側機連携を実現する追加機能 51
4.3.1 路側機の接続管理機能 53
4.3.2 複数の路側機を用いた配信機能 54
4.4 実装と評価 55
4.4.1 通信遅延時間に関する評価 55
4.4.2 車路車間通信システムによる評価 56
4.5 結言 58

第 5 章 異種分散シミュレーションによる ITS 開発環境 59
5.1 序言 59
5.2 ITS シミュレータの構成 61
5.3 異種シミュレーションの統合化手法 63
5.3.1 データ変換機構 63
5.3.2 同期機構 67
5.4 ITS モデル実装例 74
5.4.1 全体構成 74
5.4.2 情報提供による障害物衝突防止システム 75
5.5 結言 77

第 6 章 結論 79

参考文献 82

本論文に関する原著論文 **91**

A　学術論文 91

B　国際会議（査読付き） 91

C　国内研究会 92

D　全国大会等 92

図 目 次

1.1 路車協調システムのための通信プラットフォームの全体像 5
2.1 DSRC ローカル通信プラットフォームのプロトコルスタック 13
2.2 LPCP におけるアプリケーションデータ交換 15
2.3 LPP の構成図 18
2.4 LPP におけるアプリケーションデータ交換 20
2.5 メッセージ分割・組立処理 22
2.6 同報トランザクションの再実行処理のシーケンス例 23
2.7 車載器 ID タグアプリケーションを用いた LPP の初期接続時間の計測結果 25
2.8 試験シーケンスと計測区間 26
2.9 単方向データ送信トランザクションにおけるデータ送信時間の計測結果 27
3.1 プッシュ型情報配信サービスのためのプロトコルスタック 32
3.2 配信コンテンツの再生制御機能 36
3.3 リソース情報通知機能のシーケンス例 39
3.4 大容量コンテンツの分割配信機能のシーケンス例 40
3.5 実験システムの構成 42
3.6 プッシュ配信機能におけるデータ送信時間の測定結果 44
3.7 1 台あたりの通信可能データサイズと車速の関係 (通信エリア 20[m] の場合) 45
4.1 スポット型サービスと複数路側機連携サービス 47
4.2 サーバー集約方式による複数路側機連携の概要 48
4.3 複数路側機連携 DSRC 通信プラットフォームの構成 50
4.4 複数路側機連携 DSRC 通信プラットフォームでの通信シーケンス . 52
4.5 複数路側機の接続管理機能 53
4.6 複数の路側機を用いた情報配信機能 54
4.7 複数路側機連携 DSRC 通信プラットフォームを用いた車路車間通信システム 57

4.8 車路車間通信の通信遅延時間 57
5.1 各シミュレーションにおける人・道路・車両間の相互作用 60
5.2 提案するITSシミュレータの構成 62
5.3 各シミュレーションにおける座標系 65
5.4 3D Street Graph 66
5.5 道路交通シミュレーションと車両動態シミュレーション間のデータ交換処理 68
5.6 運転車両と周辺交通間の相互作用 69
5.7 道路環境シミュレーションと実行制部間のデータ交換処理 70
5.8 データ交換に起因する通信レイテンシ 73
5.9 ITSシミュレータ上へのITSモデルの実装例 75
5.10 障害物衝突防止シミュレーションの処理フロー 76
5.11 情報提供による障害物衝突防止システムのシミュレーション例 . . . 77

表 目 次

2.1 走行中の車両や低リソースな車載システムに対するアプリケーションの例 11
2.2 LPCP のプリミティブ 16
2.3 LPP のプリミティブ 19
2.4 PPP による IP over DSRC の初期接続時間 25
2.5 試験条件 26
2.6 通信エリア長及び車両速度と通信可能な最大データ長の関係 28
3.1 プッシュプロトコルのプリミティブ一覧 33
3.2 プッシュプロトコルで使用するメッセージの一覧 35
3.3 代表的なアプリケーションタイプ 37
3.4 代表的なコンテンツタイプ 37
3.5 ClientInformation で通知する主な車載器のリソース情報 38
3.6 プッシュ型情報配信の所要時間の測定結果 43
5.1 各シミュレーションの諸元 63

第1章　序論

1.1　研究の背景と従来研究

ITS (Intelligent Transport Systems) は，先進の情報通信技術を用いて，従来の交通システムを人・道路・車が一体となったシステムとすることにより，安全性（事故削減），効率（渋滞緩和），環境（CO_2 削減），快適性の改善，向上を目指したシステムであり，世界的に様々な取り組みがなされている [1]．なかでも路車間通信技術を活用した路車協調システムは，ETC (Electronic Toll Collection System)[2] や VICS (Vehicle Information and Communication System)[3] などで実用化が進んでいるほか，安全運転支援サービス [4–6] やテレマティクス [7] など ITS における中心的なシステムとして，様々な研究開発が行われている．本論文はこの路車協調システムのうち，特に位置や時間に依存したサービスに優れた特性を有する狭域通信による路車協調システムについて，システム開発の効率化を目的とした通信プラットフォームに関する研究成果をまとめたものである．

ITS における路車協調システムでは，その用途に応じて

- 走行中に場所に特化した情報をリアルタイムに送受信
- 走行中に常時接続
- 停車中あるいは停車に近い状況で様々な情報を送受信

など様々な通信要件が必要となる．しかしながら，現時点で使用可能な通信メディアでは，これらの要求を一つの通信メディアで満たすことは不可能であり，サービス形態（広域/狭域，走行中/停車中，通信コストなど）に応じた使い分けが必要となる．ETC や VICS といった現状の路車協調システムでは，サービス専用の通信基盤を整備してきたが，サービス毎に通信基盤や車載システムを構築することはコストや開発効率の面で問題がある．このため，近年，単一の通信メディアを複数のサービスが共有したり，あるサービスが複数の通信メディアを組み合わせて使用することを可能とするような通信プラットフォームに関する研究開発や国際標準方式の開発が活発に行われており，ITS における次世代路車間・車車間通信技術の統合アーキテクチャとして ISO/TC204 WG16 おいて，CALM (Communications Access for Land Mobiles)[8] とよばれる枠組みで国際標準化が行われている．

このような通信プラットフォームに関する研究のうち，通信メディアとアプリケーションを分離し，複数の通信メディアを組み合わせて使用可能とする方式として，植原らは携帯電話や無線LAN (Local Area Network) などの複数の通信メディアを，IP (Internet Protocol)[9] に統合する方式を提案した [10]．その後，Thiery や湧川らによって，Network Mobility サポートの高度化やアドホックネットワークへの対応を実現するために，IPv6[11], NEMO (NEtwork MObility) [12], MANET (Mobile Ad-hoc NETwork)[13] などの IETF(Internet Engineering Task Force) にて標準化が進展する様々なインターネット関連技術の適用による仕様面での拡張 [14–16] が行われ，前述した CALM の一部として取り込まれている．また実用面では日本の Internet ITS[17] や欧州の CVIS (Cooperative Vehicle-Infrastructure Systems)[18] といったプロジェクトに適用され，実証実験 [19, 20] が行われるなど，活発な研究が行われている．

一方，単一の通信メディアを複数サービスで使用するための研究に関しては，携帯電話や無線LAN などの汎用の通信メディアをそのまま利用すると，無線リンクの確立やインフラでのネットワーク遅延などにより，安全運転支援システムなどの高いリアルタイム性が必要なサービスへの対応が困難なことから，軽量，高速，高信頼という特徴をもつ DSRC (Dedicated Short Range Communication) と呼ばれる ITS 専用の狭域無線通信システムの多目的利用に関する研究が活発に行われている．その実現に向けては，大きく分けて，

- 走行車両に対する料金決済向けに開発された既存の DSRC を汎用利用するための技術開発
- 汎用の狭域無線システムである無線LAN に対してリアルタイム性や高信頼性を強化する形で拡張することにより，DSRC として使用するための技術開発

という 2 つのアプローチでの研究開発が行われている．

DSRC の通信規格である ARIB STD-T55[21](現在は ARIB STD-T75[22] 改定) に，車載器からの自発的な通信が可能な Active 方式を採用した日本では，前者のアプローチによる研究が活発化し，既存の通信方式である ARIB STD-55 を拡張する方式が多数提案されている [23–25]．撫中らは，携帯電話と DSRC を組み合わせ，

DSRC を用いて位置に依存したリアルタイムデータを同報配信する方式を提案している [23]．この研究では DSRC 上での IP 利用を前提としているが，DSRC 上で IP 接続を実現する方法や前述した初期接続の問題については言及されていない．平岩らは，DSRC とアプリケーションの間にアプリケーションサブレイヤとよぶDSRC の機能を拡張する層を配置し，DSRC による IP 接続 (IP over DSRC) を実現する方式を提案している [24]．しかしながら DSRC のような狭域通信メディアによる IP 接続には，通信リンクの確立ごとに行われる PPP (Point-to-Point Protocol) や DHCP (Dynamic Host Configuration Protocol) を用いた IP アドレスの割付に一定の時間がかかってしまうため，安全運転支援システムなどの高いリアルタイム性が必要となるサービスには適用できないという問題がある．この研究では，この問題に対し IP と非 IP の共存を可能とし，リアルタイムアプリケーションでは非IP 系のプロトコルで対応するとしているが，枠組みのみを規定しており，非 IP 系のプロトコルそのものについては検討されていない．

山口らは，IP over DSRC をリアルタイムアプリケーションに適用するために，DHCP による IP アドレスの割付手順を最適化する手法を提案している [25]．しかし，車載 LAN に IP ネットワークを用いたり，路車ともに専用の DHCP モジュールの作成が必要となるなど，DSRC 路側機や車載器以外への影響も大きく，実現性を考えた場合，必ずしも有効な方式とはいえない．またブラウザなどの既存の IP アプリケーションを前提とした場合，カーナビなどの高度な情報表示・入力機能を持った機器の存在が必須となり，低リソースな車載システムを実現できないという問題もある．なおここでリソースとは，メモリ容量や CPU(Central Processing Unit) 処理性能，画面などのユーザインタフェースなど，機器に搭載されるハードウェア (以下，H/W) 規模のことを指し，低リソースな車載システムとは，例えば既存の ETC 車載器のように使用するリソース量が少なく，低コストに実現可能な車載システムのことを総称して呼んでいる．以下、本論文ではこのハードウェアリソースのことを単にリソースと記すこととする．

一方，DSRC の通信規格である CEN12258,CEN12795,CEN12834[26–28] に Passive 方式を採用した欧州や，ASTM E2213[29] に無線 LAN の通信規格である IEEE 802.11a[30] をベースとした方式を採用していた米国では，後者のアプローチがとられ，無線 LAN を車載環境で利用可能とした WAVE (Wireless Access in Vehic-

ular Environments)[32] と呼ばれる DSRC 通信規格の開発が活発に行われてきた．WAVE では MAC (Media Access Control) 層以下の下位層は IEEE 802.11p[31], LLC (Logical Link Control) 層以上の上位層は IEEE 1609[32–34] として 2010 年頃の標準化完了を目標とした開発が行われている．下位層である IEEE 802.11p では一般の無線 LAN 規格である 802.11a に対して，初期接続手順の簡略化などにより接続遅延を短縮するなど，高速走行下での利用を想定した拡張を行っている．一方，上位層の IEEE 1609 では，前述した IP 系への統合に関する規定に加え，非 IP 系に関しても安全運転支援などの高信頼なリアルタイムアプリケーション向けに，2004 年頃から Geo-Routing[35] などの ITS 専用の非 IP 系のネットワーク層や WSMP (WAVE Short Message Protocol) とよばれる IEEE 802.11p 専用のトランスポート層に関する規定も行われている．現在は，実機を用いた評価 [36,37] のフェーズであり，実用化に向けた開発が活発化している．

1.2 本研究の位置付けと目的

前節で示したように，路車協調システムにおける通信プラットフォームに関する従来の研究では，インターネット接続により通信とアプリケーションを分離することで，アプリケーション開発の効率化を図ると共に，既存の IP アプリケーションの利用を可能とする方式についての研究が多く見られる．このため，狭域通信メディアのプラットフォーム化に関する研究についても，IP 利用を可能とする方式や L2 層以下の下位層に関するものがほとんどである．しかしながら IP 接続を前提とする方式では，初期接続性能や必要リソース量に問題があり，高速走行車両に対するリアルタイムサービスや低リソースな車載システムに対するサービスに適用することは困難であり，実用化には至っていない．

この問題を解決する手法として，IP プロトコルを利用しない非 IP 系のより軽量な通信プロトコルを利用することが考えられるが，従来のプラットフォーム化に関する研究では，インターネット・プロトコル・スイート [38] に代わる非 IP 系の第 3 層 (L3: Layer 3) 以上の通信プロトコルに関しては，十分な研究開発が行われておらず，非 IP 系サービスに適用可能な実用的な通信プラットフォームはまだ存在しない．インターネットでのサービスが，TCP(Transmission Control Protocol)/UDP(User Datagram Protocol) や HTTP(HyperText Transfer Protocol)/FTP(File Transfer

Protocol)/SNMP(Simple Network Management Protocol) 等の基本プロトコルを使って多様なサービスを効率的に実現しているように，非 IP 系サービスについても同様の通信プラットフォームを構築することが重要と考える．

以上のような状況を踏まえ，本研究では，このような狭域通信による高速走行車両や低リソース車載器に対する種々の情報通信サービスの実現とその開発効率化を目指し，図 1.1 に示すような路車協調システムを構成する様々なシステム間の連携を実現する通信プラットフォームの構築を行うことを目的としている．

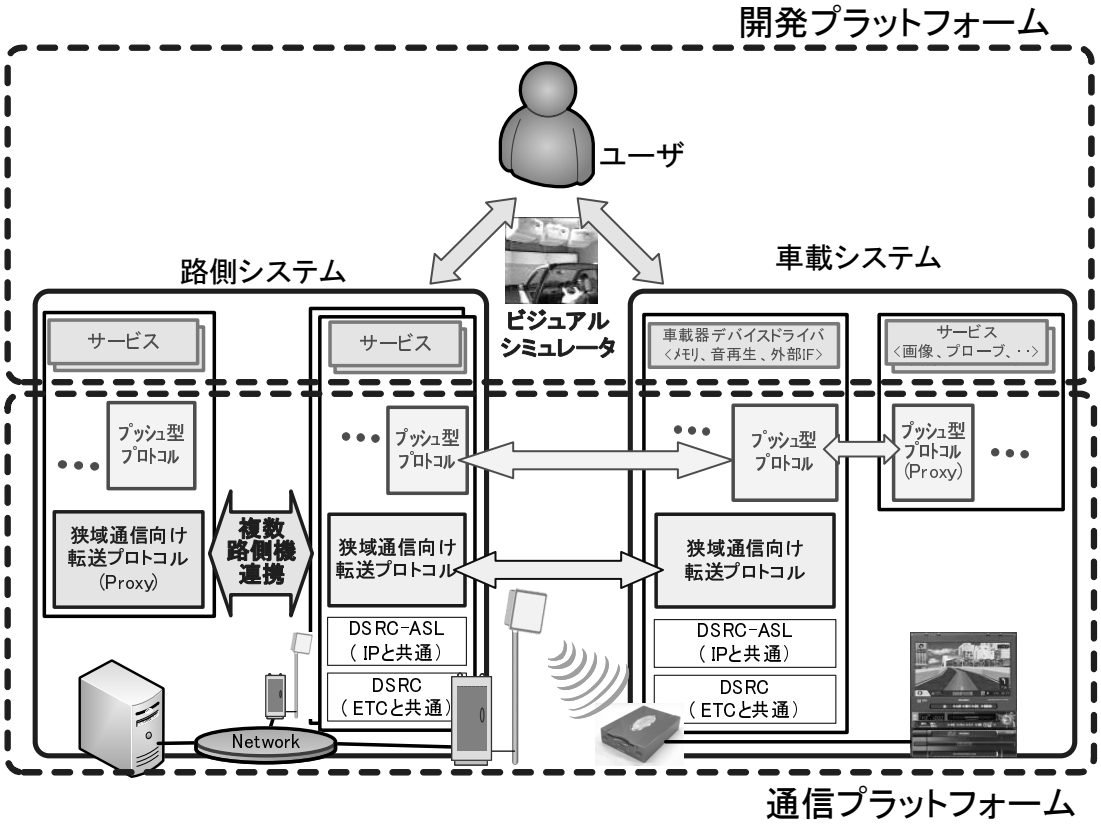


図 1.1: 路車協調システムのための通信プラットフォームの全体像

まず，単一路側機によるスポット型サービス向け非 IP 系通信プラットフォームを提案し，このプラットフォームを構成する通信プロトコルとして，

- IP 系における TCP/UDP に相当するマルチアプリケーション対応プロトコルとして，高速走行車両や低リソース車載器に対するサービスに共通に必要となる通信機能を提供する転送プロトコル [39–44]

- IP 系における HTTP や FTP に相当する基本的なアプリケーションプロトコルの一つとして，安全運転支援情報や駐車場での店舗情報など走行中の車両に対する様々な情報提供サービスを実現するプッシュ型プロトコル [44–48]

を提案する．転送プロトコルは，単一路側機によるスポットサービスを対象とし，路側機/車載器ローカルでアプリケーションが実行されることを前提とすることで，高速な初期接続や高効率な通信の実現を目指す．一方，プッシュ型プロトコルは，既存のインターネット系のプッシュ型プロトコルに対し，通信可能時間や様々な車載システム構成への対応など狭域通信や車載システム特有の要件を考慮した設計とすることで，走行中の車両に対する様々な情報配信サービスの実現を目指す．

ところで，IP のルーティング機能により様々な機器構成が可能な IP 系とは異なり，上記非 IP 系の通信プラットフォームは，路側機/車載器ローカルでアプリケーションが実行されることを前提としている．このため，交差点などにおいて異なる路側機下に存在する車両間での情報交換などは，複数の路側機間の連携が必要な面的なサービスとなるが，これまで非 IP 系通信では，サービス毎に専用の連係機能を開発しなければならず，開発効率が大きく低下するという問題があった．そこで，路側機間の連携機能をプラットフォーム側が持つことで，非 IP 系においてもこのような面的なサービスの効率的な開発を可能とする通信プラットフォームの拡張方式 [49, 50] を提案する．提案する手法では，路側機上のプロトコルスタックを分割し，アプリケーションプロトコルを路側機の外部サーバーへ配置することが可能となる．この仕組みにより，サーバーでの基本的なアプリケーションプロトコルの組み合わせによる複数路側機連携サービスの構築を可能とする．

さらに提案した通信プラットフォームを用いて実際に路車協調システムを構築する場合，特に安全運転支援や交通情報提供など高速走行中のサービスでは，コストや安全性，再現性などの問題から，実フィールドにおける試験を行うことが困難なことから，多様な条件の下で容易に評価・検証を繰り返すことができる計算機シミュレーション環境が必須となる．システムの評価においては，通信システムとしての性能評価に加えてサービスとしての評価も必要となることから，本研究では通信プラットフォームを用いて実現される様々な路車協調システムの開発プラットフォームとして，開発されるサービスに対応して，様々な交通状況や新たなシステムの動作などを容易に生成・追加可能な高い拡張性をもつシミュレータ [51–54] を

開発する．提案するシミュレータは，様々な道路交通現象や車両を評価するために開発されてきた既存の道路交通シミュレータとドライビングシミュレータを異種分散環境として統合することで各シミュレーションの個別のバージョンアップや，別のシミュレーションの追加などを容易にし，様々な路車協調システムへの適用を実現する．

1.3 論文の構成

本論文は，路車協調システムのための通信プラットフォームの構築を目的として，

(1) DSRC の多目的利用を実現する路車間通信環境に適した高速通信プロトコル
(2) 車両へのプッシュ型情報配信サービスに適したアプリケーションプロトコル
(3) 複数路側機連携サービスのための DSRC 通信プラットフォーム
(4) 異種分散シミュレーションによる ITS 開発環境

を新しく提案し，その効果を明らかにしたものである．

第 2 章では，走行中の車両や低リソースな車載システムに対するサービスをターゲットとした DSRC ローカル通信プラットフォームを提案する．本章では，まず対象とするサービスに要求される機能・性能要件をまとめ，それらを実現するためのアーキテクチャと非 IP 系高速通信プロトコルについて述べる．次にこれらのプロトコルを実際の DSRC 路側機と DSRC 車載器上に実装し，通信実験を行うことで，要求されている性能要件を満たすことができるかどうかを評価・検証し，その有効性を示す [39–44]．

第 3 章では，第 2 章で提案した高速通信プロトコル上で，走行車両への情報配信サービスを実現するためのプッシュ型プロトコルを提案する．本章では，まず走行中の車両に対する情報提供サービスに要求される要求条件をまとめ，それらを実現するためのアーキテクチャとアプリケーションプロトコルについて述べる． また，実機を用いた検証を行うことで，提案したプロトコルの動作や通信性能を確認し，実際に車両への多種多様な情報配信サービスに適用可能であることを示す [44–48]．

第 4 章では，第 2 章および第 3 章で提案した単一の DSRC 路側機を用いたスポット型サービス向けの DSRC ローカル通信プラットフォームを，複数の DSRC 路側機を利用した空間的な広がりのあるサービスに適用可能とし，これらのサービス構

築の効率化を実現するため，路側システム側のネットワーク化による拡張方式を提案する．さらに異なる路側機下に存在する車両間でデータを交換する車路車間通信サービスを実装し，その遅延時間などについて評価を行うことで，本方式の有効性を示す [49, 50].

第 5 章では，通信プラットフォームを用いて構築される様々な路車協調システムの設計・評価に不可欠なシミュレーション環境として，路車協調システムの構成要素である人・道路・車全ての間の協調動作を再現可能な道路交通シミュレータとドライビングシミュレータの統合シミュレータを提案する．またこの統合シミュレータに，路車協調システムをプロトタイプ実装し，動作を検証することにより，本シミュレータの ITS 開発プラットフォームとしてのとしての使用可能性について考察し，本方式の有効性を示す [51–54].

第 6 章では，結論として本研究で得られた成果を総括する．

第2章　DSRCの多目的利用を実現する狭域通信環境向け高速通信プロトコル

2.1　序言

近年，カーナビゲーションなどの車載情報システムや移動体通信技術の発展に伴い，車内と車外をつなぐ情報通信へのニーズが高まっている．その一つとして，高速道路の自動料金収受システムであるETC[2]があり，急速に普及が進んでいる．

DSRC : ARIB STD-T75[22]は，このETCで使用されている通信方式であり，

(1) 常に車に接続されている
(2) 数[m]〜100[m]未満の通信領域をもつ狭域通信であり，位置に依存したサービスにむいている
(3) 初期接続に要する時間が短いため，高速走行する車両との間で双方向通信が可能
(4) プロトコル構造が簡単なため実装に必要なリソース量が少ない．また，車載器単独でのサービスが可能で，低リソースな車載システムを構築可能

といった高速移動体である車両に対するサービスに適した多くの特徴を有しており，ETC以外の様々なサービスへの適用が期待されている．

DSRCはこのようなマルチアプリケーション化の要求に対して，Application Entity ID(AID)を用いた機構を有している．しかしながら，この機構を用いるためには，詳細な仕様をアプリケーション毎に設定する必要があるとともに，国際標準規格[55]の改訂が必要となるため，その適用に時間がかかるという問題がある．またAIDの数が32個に制限されているため，個別のアプリケーション毎にIDを割り振ることは困難である．さらに，DSRCはETCでの利用を想定して策定されたプロトコルであるため，車載器からの自発的な送信や，大容量通信といった多くのアプリケーションに共通する基本的な通信サービスを具備していない．そのため，アプリケーションの開発に当たっては，これらの基本的な通信機能についても個々のアプリケーション毎に定義し，実装しなければならず，メモリ容量やCPU処理性能などのリソース量の増大や，開発効率の悪化を招いてしまうという問題もあった．

これらの問題を解決するためにITS ネットワーク [23] やアプリケーションサブレイヤ (以下，ASL)[24] と呼ばれるプラットフォームが提案されているが，第 1 章で述べたように，これらの従来研究ではマルチアプリケーションを実現する手段として，インターネット接続を実現する方式 (IP over DSRC) を用いるため，

(1) 初期接続に一定の時間がかかるため，走行中の車への適用が困難
(2) 高度な情報表示・入力機能を必要とするため，低リソースで実現できない

という問題がある．

これらの IP over DSRC の課題は，普及促進の観点や，サービス対象が車両という高速移動体であることを考慮すると，DSRC の多目的利用においては，解決しなければならない問題であるといえる．

そこで，本研究では，走行中の車両や低リソースな車載システムに対するサービスをターゲットとし，これらのサービスを実現するアプリケーションの開発効率の向上を目的とした通信プラットフォーム [39–44] を提案する．

本章では，提案する通信プラットフォームのアーキテクチャとして，設計方針や要求条件を示し，それらを実現するためのプロトコルスタック構成と開発した通信プロトコルについて示す．次に，提案した通信プラットフォームについて，低リソース端末への実装可能性，初期接続時間及び走行車両への通信量の各項目に関して検証を行い，走行中の車両や低リソースな車載システムに対するサービスに適用可能であることを示す．

2.2　DSRC ローカル通信プラットフォームのアーキテクチャ

2.2.1　設計方針

前節において，IP over DSRC には初期接続性能と必要リソース量に課題があり，高速走行車両や低リソースな車載システムに対するサービスへの適用が困難という問題があることを述べた．一方，本来 DSRC はその高速接続性とシンプルな構造が特徴で，走行中の車両に対するサービスに適した方式である．また，プロトコルの構造が簡単なため，実装に必要なリソース量が少ないうえ，ETC に代表されるように車載器上にアプリケーションを搭載することにより，車載器単独でのサービスも可能である．従って，低リソースな車載システムを構築できるメリットを有す

表 2.1: 走行中の車両や低リソースな車載システムに対するアプリケーションの例

アプリケーション	走行	リソース量 (車載機器構成)	データ量
プッシュ型情報配信	高速走行〜停止	車載器（＋外部端末）	数 Byte〜数 100KByte
プローブ情報	高速走行〜停止	車載器＋外部端末	数 Byte〜数 KByte
車載器 ID 通信	低速走行〜ワンストップ	車載器単独	数 10Byte
課金・決済	停止	車載器単独 (＋外部端末)	数 10Byte〜数 100Byte

る．よって，提案する通信プラットフォームは，これらの DSRC の特徴を最大限生かす仕様として設計する．

一方，HTTP などの豊富なアプリケーションプロトコルを有する IP over DSRC の場合と異なり，新しい通信プラットフォーム上において，マルチアプリケーションを実現するためには，新規にアプリケーションプロトコルを規定する必要がある．この作業には一定の時間を要するため，全てのサービスをこの通信プラットフォーム上で実現することは，開発効率の悪化をもたらす．従って，IP アプリケーションにより実現可能なサービスについては，IP アプリケーションで実現することが望ましい．

以上のことから，IP over DSRC を，提案する通信プラットフォームにおいて，IP アプリケーションをサービスするための補完的機能と位置づけ，本プラットフォームをベースにしたプロトコル開発は，IP アプリケーションでは実現が困難なサービス（走行中の車両や低コストな車載システムに対するサービス）のみをそのターゲットとして設計を行う．表 2.1 に本プラットフォームがターゲットとする走行中及び低コストな車載システムに対するアプリケーションの例を示す．

2.2.2　要求条件

2.2.2.1　性能要件

前節で述べたように，提案する通信プラットフォームは IP アプリケーションで実現が困難なサービスをそのターゲットとしている．従って，提案するプロトコルは IP over DSRC で問題となった

(1) 高速な初期接続
(2) 低リソースな車載システムの実現

を解決するような仕様でなければならない．

また，走行車両に対するサービスでは，通信可能な時間が限られるため，その限られた時間内で実行可能な通信量やトランザクション数を見積もれることが望ましい．

2.2.2.2　機能要件

提案する通信プラットフォームは IP over DSRC を補完的機能と位置づけていることから，IP 通信と混在可能である必要がある．また，表 2.1 であげたアプリケーションのうち，例えばプッシュ型情報配信では，比較的大容量のデータを送信する必要があるとともに，同報通信の機能も必要である．また，課金・決済などのサービスでは，データ損失に対する要求も厳しくなる．さらに，新規にアプリケーションプロトコルを定義する必要があることから，DSRC の仕様を意識させない効率的な API(Application Program Interface) や，TCP のような比較的容易に利用可能なマルチアプリケーションの開発・実行環境を提供することにより，アプリケーション構築の効率化を図ることも重要である．

以上より，提案する通信プラットフォームに要求される機能要件としては，

(1) マルチアプリケーション・プロトコル機能
(2) 大容量データ通信機能
(3) 同報通信機能
(4) 高信頼通信機能
(5) 効率的な API の提供

があげられる．

2.2.3 プロトコルスタック

提案する通信プラットフォームのプロトコルスタックを図 2.1 に示す.

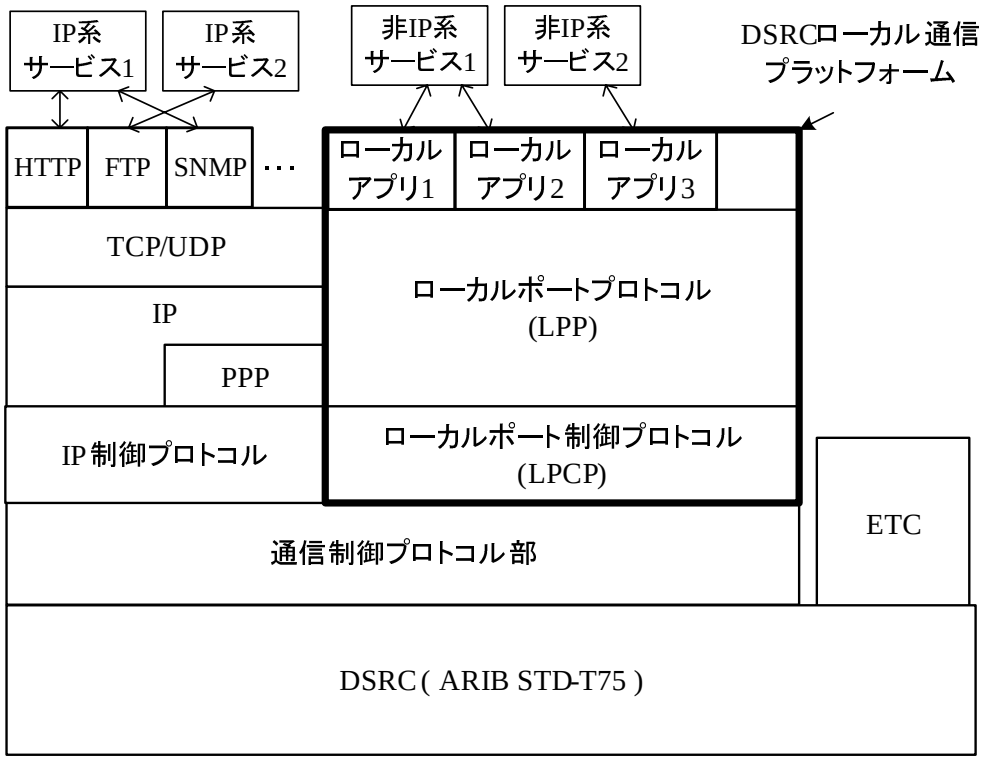


図 2.1: DSRC ローカル通信プラットフォームのプロトコルスタック

図 2.1 に示すように，提案するプラットフォームは，文献 [24] で提案されている ASL の通信制御プロトコル部上に構築することとした．この ASL は，マルチプロトコルに対応しており，一つの AID により，IP over DSRC と，その他の通信プロトコルの混在が可能である．また，前節で述べた機能要件を実現するために必要となるクライアント/サーバ型通信制御，バルク転送制御，同報モード制御などの通信サービスを有している．本プラットフォームが IP over DSRC を補完的機能と位置づけていることを考慮すると，これらの機能を IP over DSRC と共通的に利用可能とすることで，開発効率の向上やリソース量の軽減を図ることが可能となる．

一方，プラットフォームの構成としては，インターネット標準との親和性を考慮し，

- 様々なアプリケーションに共通の通信機能を提供する転送プロトコル

- サービスに共通のアプリケーション機能を提供するアプリケーションプロトコル群

からなるプロトコルスタックとして構成し，本プラットフォームを活用するサービスは，基本的なアプリケーションプロトコルの組合せで多様なサービスを構成する．これはインターネットでのサービスが，TCP/UDP，HTTP/FTP/SNMP 等の基本プロトコルを使って多様なサービスを実現していることと同様である．

転送プロトコルは，マルチアプリケーションを実現するための最小限の機能を持つプロトコルであるローカルポート制御プロトコル (LPCP:Local Port Control Protocol) と，LPCP の通信サービスを拡張し，より高度な通信サービスを提供するプロトコルであるローカルポートプロトコル (LPP:Local Port Protocol) の 2 階層で構成することとした．これは，プロトコルを 2 階層に分割し，下位レイヤである LPCP を軽量なプロトコルとすることで，低リソース車載器への実装を容易にするためである．また，保有する機能の多い LPP についても，各機能をモジュール化し，対応するアプリケーションに応じて，必要な機能のみの実装を可能とすることで，低リソース車載器への実装を考慮した設計とした．

また，高速な初期接続を実現するために，LPCP/LPP 共に明示的なコネクションの設定・解放フェーズを持たないコネクションレス型のプロトコルとして設計した．さらに，DSRC 路側機，車載器ローカルでアプリケーションが実行されることを前提とし，プロトコルスタック上，ネットワークレイヤを持たない構成とした．これにより，初期接続に必要な手順を減らし，高速な初期接続を実現すると共に，PDU(Protocol Data Unit) やプロトコル処理に要するオーバヘッドを最小化し，通信のスループットの向上を図っている．

以下，本論文では，本通信プラットフォームが対象とする DSRC 路側機/車載器ローカルで実行されるアプリケーションを DSRC ローカルアプリケーションと呼ぶこととし，さらに提案する通信プラットフォームを DSRC ローカル通信プラットフォームと呼ぶこととする．

2.3 DSRC ローカルアプリケーションのための通信プロトコル

本節では，2.2 で述べたアーキテクチャを実現するために設計した 2 つのプロトコル (LPCP, LPP) について詳述する．

2.3.1 ローカルポート制御プロトコル (LPCP)

LPCP は，DSRC ローカルアプリケーションの多重化を実現するための制御プロトコルである．前節で述べたアーキテクチャに従い，LPCP では，マルチアプリケーションに必要となる最低限の機能のみを有するものとし，DSRC ローカルアプリケーションに対して以下の通信サービスを提供するプロトコルとして設計した．

(1) ローカルポート番号によるアプリケーションの識別
(2) データグラム転送サービスの提供 (個別，同報*)
(3) 下位層におけるイベントの通知
(4) 初期接続時のローカルポート番号交換

図 2.2 に LPCP におけるアプリケーションデータ交換の仕組みを示す．

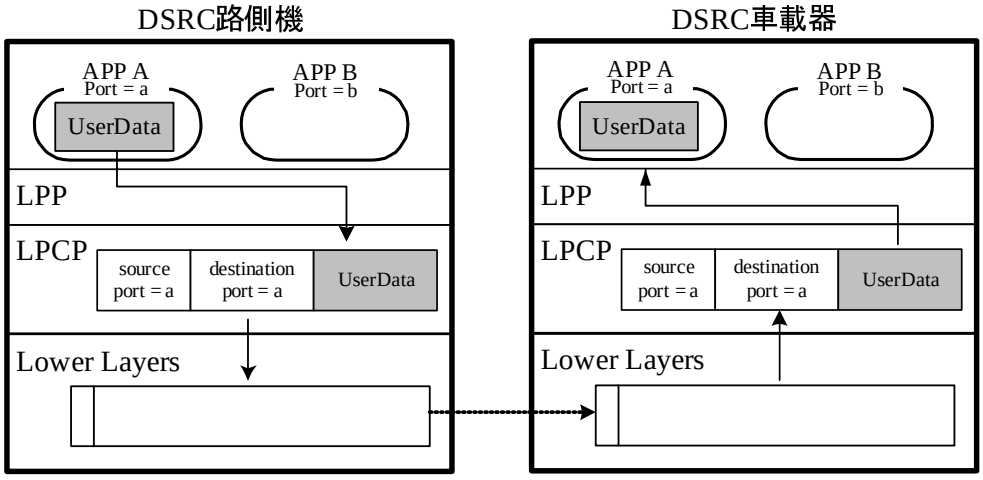


図 2.2: LPCP におけるアプリケーションデータ交換

これらの機能のうち (1),(2) は，複数のローカルアプリケーション間でデータ伝送を行うために最低限必要となる機能である．(1) については，アプリケーション

*下位レイヤが提供する同報機能を利用

表 2.2: LPCP のプリミティブ

対応する機能	プリミティブ名	拡張子	処理内容
データグラム転送サービス	TransferData	req	データ転送要求
		ind	データ受信通知
下位層におけるイベントの通知	EventReport	ind	イベント（接続・切断・相手局の有するローカルポート番号など）の通知

の識別に ARIB STD-T75 における AID/EID や ASL におけるアクセス点といった既存の枠組みではなく，ローカルポート番号 (0〜0xFFFF) と呼ぶ識別子を使用している．これは，

- AID/EID は国際標準規格として規定されており [55]，自由に利用することはできず，その適用に時間を要する
- ASL におけるアクセス点は IP などのプロトコル識別を行うための機構であり，その数が 16 個と少なく個別のローカルアプリケーションの識別子として利用することは難しい

といった理由による．一方，LPCP に識別子を持たせることで，DSRC などの他の規格を変更することなく，独立してアプリケーションの追加が可能といったメリットもある．

一方，(2) については，ETC 車載器と同程度のリソース量で実現可能とするため，LPCP におけるデータグラムの最大サイズ (MTU:Maximum Transfer Unit) は，ETC における最大転送サイズに近い 522 オクテットとした．また (3) は，初期接続や切断などをアプリケーションに対して通知するための機能であり，アプリケーションが通信の開始や終了などを判断するために必要な機能である．さらに (4) は，初期接続時に自局の有するローカルポート番号を交換することで，初期接続時の輻輳を防ぐための機能である．限られた通信時間内で，通信可能なデータ量やトランザクション数を見積もる必要のある狭域通信においては，重要な機能といえる．なお，このローカルポートの交換は基地局と車載器で非同期に行われる．また，これら (1)〜(4) の機能は表 2.2 に示すプリミティブとしてアプリケーションに対して提供される．

このように，LPCP は相互接続性に係わる基本的かつ重要な通信サービスのみを対象とした，実現される機能が単純かつ少ない軽量なプロトコルである．このため実装に必要となるリソース量も少なく，低リソース車載器への実装が可能である．また，DSRC ローカルアプリケーションに特化し，ネットワーク層の機能を有さず，初期接続に係わる手順も非同期のポート交換のみであることから，高速な初期接続を実現できる．

また，実現する機能としては，マルチアプリケーション機能と同報機能のみをサポートすることとし，2.2.2 で述べたアプリケーションによっては必要となる他の機能要件については，後述する LPP を利用することで実現するものとする．

2.3.2 ローカルポートプロトコル (LPP)

LPP は，LPCP と DSRC ローカルアプリケーションの間に介在し，LPCP の通信サービスを拡張するトランザクション処理指向の通信プロトコル [56] であり，DSRC 路側機/車載器上の DSRC ローカルアプリケーションに対して，2.2.2 節で述べた機能要件を満たす通信サービスを提供する．

図 2.3 が示すように LPP は，DSRC ローカルアプリケーションに対して，種々の通信サービスを提供するトランザクションサービス処理部と初期接続や切断などの通信状況を管理する接続管理サービス処理部から構成することとし，各サービス処理部には以下の機能を持たせた．

<u>トランザクションサービス処理部</u>

(1) トランザクションサービス（基本機能）

- 単方向データ送信トランザクション
- リクエスト・レスポンス型トランザクション

(2) トランザクションサービス（付加機能）

- メッセージの分割・組立機能
- データ再送機能
- 同報トランザクションの再実行機能

(3) トランザクション破棄サービス

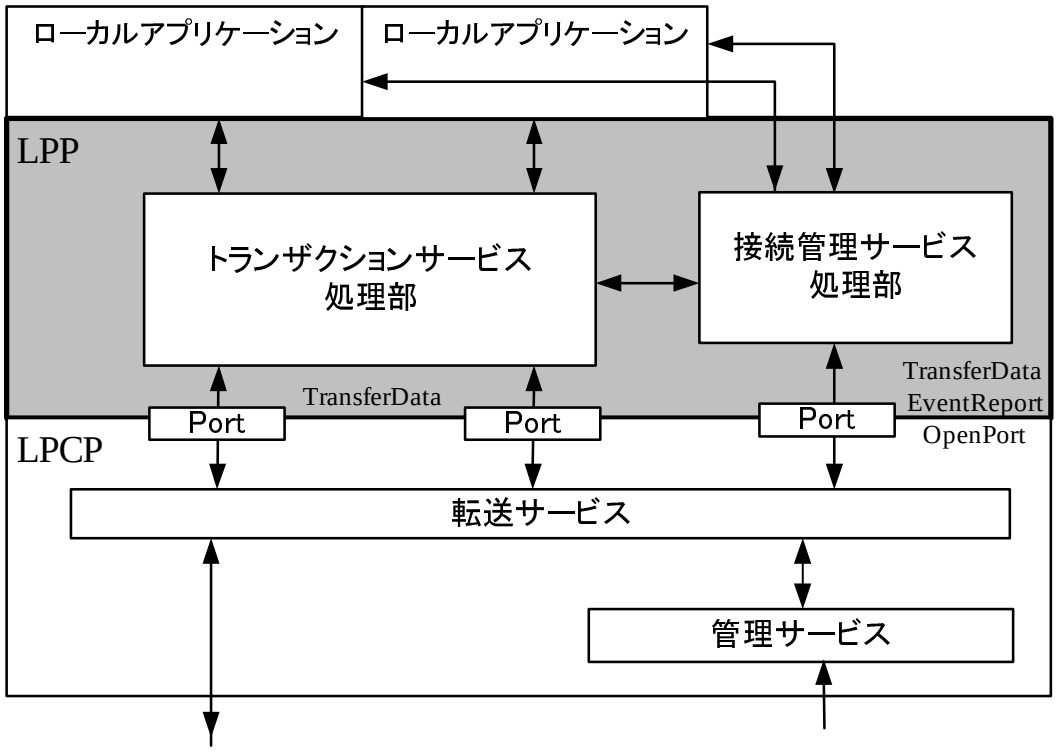


図 2.3: LPP の構成図

<u>接続管理サービス処理部</u>

(1) 接続問い合わせサービス
(2) 受信可能ポート問い合わせサービス
(3) 切断通知サービス

2.2.2 節で述べた機能要件のうち，アプリケーション構築の効率化を図るため，単純な単方向データ送信トランザクションに加え，リクエスト・レスポンス型のトランザクションや種々の接続管理サービスを用意し，アプリケーションに対しては表 2.3 に示すプリミティブとして提供することとした．さらに，大容量データ通信を実現するためのメカニズムとして，メッセージの分割・組立機能を用意した．また，高信頼通信を実現するために，データ再送機能と同報トランザクションの再実行機能を定義した．

一方，性能要件を考慮し，以下のような設計を行った．

表 2.3: LPP のプリミティブ

対応するサービス	プリミティブ名	拡張子	処理内容
トランザクションサービス	Invoke	req	トランザクション開始要求
		ind	トランザクション受信通知
		res	トランザクション応答要求
		cnf	トランザクション応答通知
トランザクション破棄サービス	Abort	req	トランザクション破棄要求
		ind	トランザクション破棄通知
接続問い合わせサービス/受信可能ポート問い合わせサービス	Connect	req	トランザクション開始可能問い合わせ要求
		cnf	トランザクション開始可能問い合わせ結果通知
切断通知サービス	Disconnect	ind	DSRC 切断通知
—	RegisterPort	req	受信可能ローカルポート登録
—	DeregisterPort	req	受信ローカルポート登録削除

(1) コネクションレス型の通信プロトコルとして設計することにより，初期接続時における通信路の開設といった動作を不要とし，高速な初期接続を実現

(2) リソース量の少ない車載器への実装を考慮し，LPP の各機能は必要な機能のみの選択的な実装が可能

以下，本節では，LPP の基本サービスであるトランザクションサービスと大容量通信を実現するためのメカニズムである分割・組立機能について詳述する．

2.3.2.1 トランザクションサービス

図 2.4 に示すように，LPP では，トランザクション単位でアプリケーションデータの交換を行う．各アプリケーションデータにはトランザクションが発行されるタイミングで，ユニークなトランザクション ID が付与され，他のデータと区別される．従って，LPP では，初期接続時にコネクションを設定する必要がなく，高速な初期接続を実現できる．

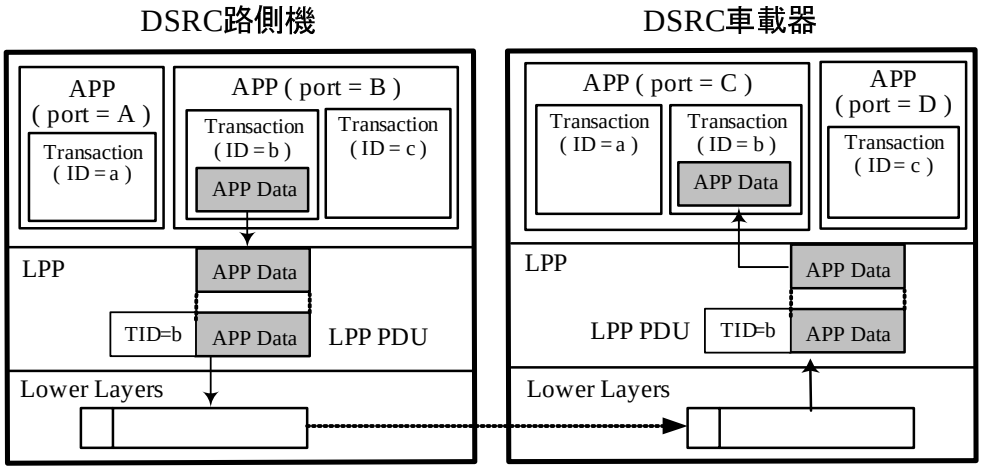


図 2.4: LPP におけるアプリケーションデータ交換

2.3.2.2 大容量通信のためのメカニズム

LPP では，大容量データ通信をアプリケーションが容易に実現できるように，LPCP の MTU を超えるメッセージの分割・組立機能を有している．一般に分割・組立機構を LPCP のようなコネクションレス型のプロトコル上で実現する場合，メッセージ順序の入れ替わりや受信バッファのオーバーフローなどが発生する可能性があるため，順序制御やフロー制御などのデータ転送制御機構が必要となる．例えば，既存のトランザクション型のプロトコルである RPC (Remote Procedure Call)[57] では，その分割・組立機構においてウィンドウや Selective Acknowledgement を用いた順序制御やフロー制御を行っている．

一方，LPP では，同報通信での分割・組立機構の実現や，実装に必要なリソース量の軽減といった要求があるため，これらの方式を直接利用することは難しい．そこで LPP では，大きな送信遅延が発生しない，事前の通信量の見積もりが可能

といった狭域通信やローカルアプリケーションの特徴を利用し，分割・組立機構として，受信バッファをアプリケーションが用意し，分割・組立のメカニズムのみをLPP が提供する方式を用いることで，これらの要求を解決した．このような方式とすることで，受信バッファが常にメッセージサイズよりも大きくなり，フロー制御が不要になるため，同報通信に対しても分割・組立機構を適用することが可能となる．さらにLPP 自身には組立のためのバッファ領域が不要となるため，システム全体としてのリソース量の軽減を図ることが可能となる．本機能の基本的な動作は以下の通りである．

LPP はアプリケーションからLPCP の MTU を超えるメッセージを受け取った場合には，送信メッセージをLPCP の MTU サイズのパケットに分割し，順次LPCP に渡すように操作する．このとき分割されたパケットには順序番号を付与する．受信側の LPP は，LPCP から渡された分割されたパケットを順次取り込み，受信側のアプリケーションが用意した受信バッファへ積み上げる．この際，受信側は，各パケットに付番された順序番号で組立順を判別してメッセージに組み上げる．

一方，パケットの欠落に関しては，個別通信では不達通知による再送処理，同報通信ではトランザクションの再実行処理による繰り返し伝送により解決する．以下本節ではこれらの仕組みについて詳述する．

2.3.2.3 不達通知による再送処理

個別通信では，最終パケット受信時に未受信のパケットを，不達通知により送信側に通知し，送信側が未達パケットを再送する選択的再送機能を利用することで，通信の信頼性を確保する．この選択的再送機能では，再送の制御を最終パケット受信時のみに行うことで，信頼性確保のための制御メッセージの数を最小化し，大容量データの効率よい伝送を実現した．図 2.5 に，個別通信において本機能を使用した場合のトランザクションのシーケンス例を示す．

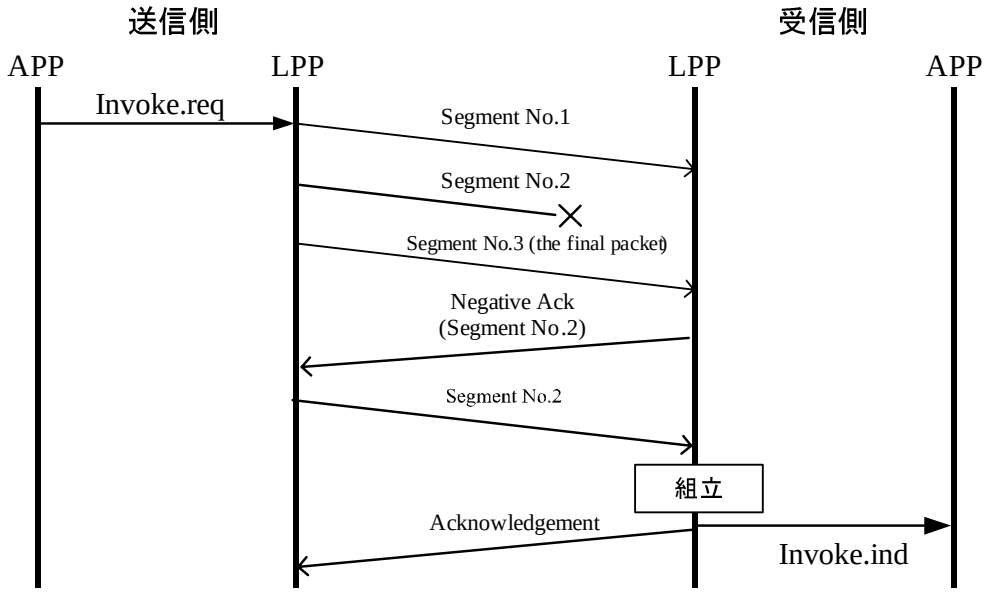


図 2.5: メッセージ分割・組立処理

2.3.2.4 同報トランザクションの再実行処理

同報通信の信頼性確保の仕組みとして，同報トランザクションの再実行機能を持たせることで対応した．本機能は送信側アプリケーションから発行される複数のトランザクションを同一のトランザクションと見なし，LPP の PDU 単位での繰り返し伝送を実現する機能である．本機能を用いることで，あるトランザクションにおいて分割された各パケットは，再実行機能により発行される複数の同一トランザクションのいずれかで受信すればよいこととなり，再実行の回数を増やすことで，同報通信においても通信の信頼性を確保することが可能となる．具体的には，複数のトランザクション開始要求に対して，同一のトランザクション ID を割り当て，受

信側において，これらの複数のトランザクションを同一のトランザクションとみなすことで本機能を実現した．

本方式では，繰り返し伝送されるデータが，LPP の PDU となることから，アプリケーションから渡される送信メッセージが大きい場合には，繰り返し送信されるデータの送信時間間隔が大きくなるため，下位層で連送を行う方式に比べバーストエラーに対してより効果的である．また，本機能では，繰り返し送信する時間間隔をアプリケーションが自由に設定できることから，トランザクションデータが小さい場合でも十分な効果を期待することができる．図 2.6 に同報トランザクションの再実行処理のシーケンス例を示す．

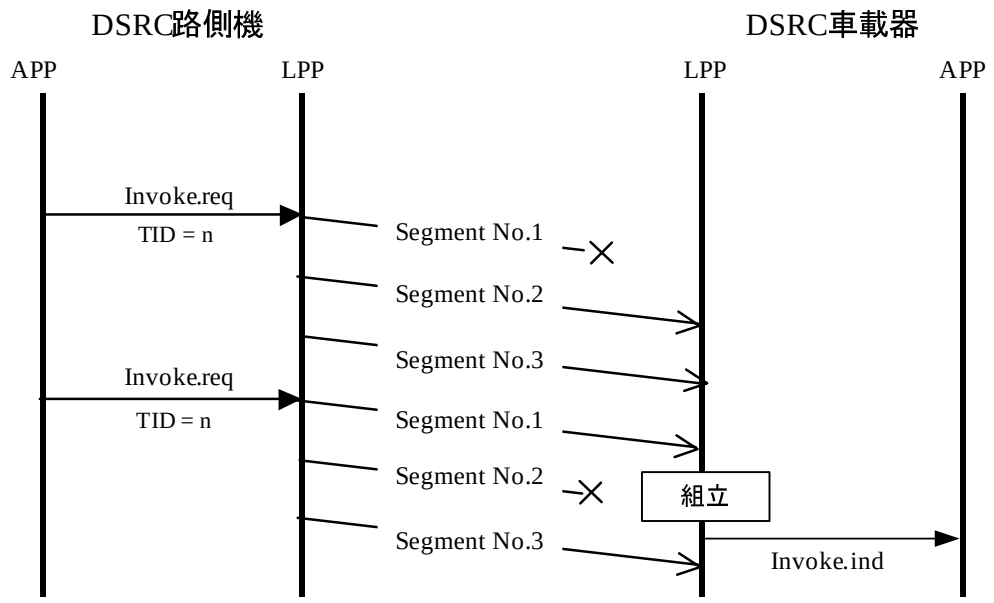


図 2.6: 同報トランザクションの再実行処理のシーケンス例

2.4　実装と評価

2.4.1　LPCP/LPP の実装とタグ型サービスの実現

本章で提案した通信プロトコルを利用したサービスとして，駐車場の入退場管理・料金決済システムがある [58, 59]．本システムは，LPP 上に構築した車載器 ID タグアプリケーションにより，駐車場に設置した DSRC 路側機と DSRC 車載器間で情報 (車載器 ID) の授受を行い，駐車場ゲートを自動的に開閉・料金精算する仕組みである．

本節では，提案したプロトコルの有効性を確認するために，このサービス用に開発した DSRC 路側機/車載器を用いて，2.2.2 で述べた性能要件と照らし合わせ，低リソース車載器への実装可能性及び初期接続時間に関する評価を行う．

2.4.1.1　低リソースデバイスへの実装可能性

本システムのために開発した DSRC 車載器では，リソースとして既存の ETC 車載器と同程度のリソースのものを使用している．一方，この車載器には提案したプロトコルのうち，LPCP の全機能と，LPP の機能のうち，本サービスで使用する機能のみを選択的に実装した．またアプリケーションとして，ETC と車載器 ID タグアプリケーションを実装している．なお，この LPCP/LPP/車載器 ID タグアプリケーションの実装で使用した ROM/RAM 容量はそれぞれ 8[KByte] と 2[KByte] であった．

この事例は，LPCP/LPP が低リソース車載器への実装が可能であることを示すとともに，LPCP/LPP を用いることで，車載器単独のサービスが実現でき，低リソースな車載システムの構築が可能なことを示している．

2.4.1.2　初期接続時間

車載器 ID タグアプリケーションを用いて，初期接続時間を計測した結果を図 2.7 に，また PPP による IP over DSRC の初期接続時間を計測した結果を表 2.4 に示す．なお，後者については，別途開発した IP over PPP over DSRC を実装した DSRC 車載器のプロトタイプを用いて計測を行った．

図 2.7 に示すように，LPP の初期接続時間は約 50[msec] 以下の値を示し，平均値は約 40[msec] である．最も即時性が要求される安全運転支援サービスなどで要求

される通信遅延時間が 100[msec] 程度である [4] ことを考慮すると，十分高速な初期接続性能を担保できているといえる．一方，表 2.4 に示すように，IP 接続の場合は，PPP によりネットワークを設定する時間が必要となるため，初期接続（認証処理を除く）に，約 500 ミリ秒の時間がかかっている．以上のことから LPCP/LPP による初期接続時間は IP over DSRC 方式に比べて短いことがわかる．

表 2.4: PPP による IP over DSRC の初期接続時間

	認証フェーズ [sec]	認証フェーズ以外 [sec]	全体 [sec]
1 回目	4.000	0.589	4.589
2 回目	3.890	0.480	4.370
3 回目	3.891	0.479	4.370
4 回目	3.890	0.495	4.385
5 回目	3.813	0.479	4.292
平均	3.897	0.504	4.401

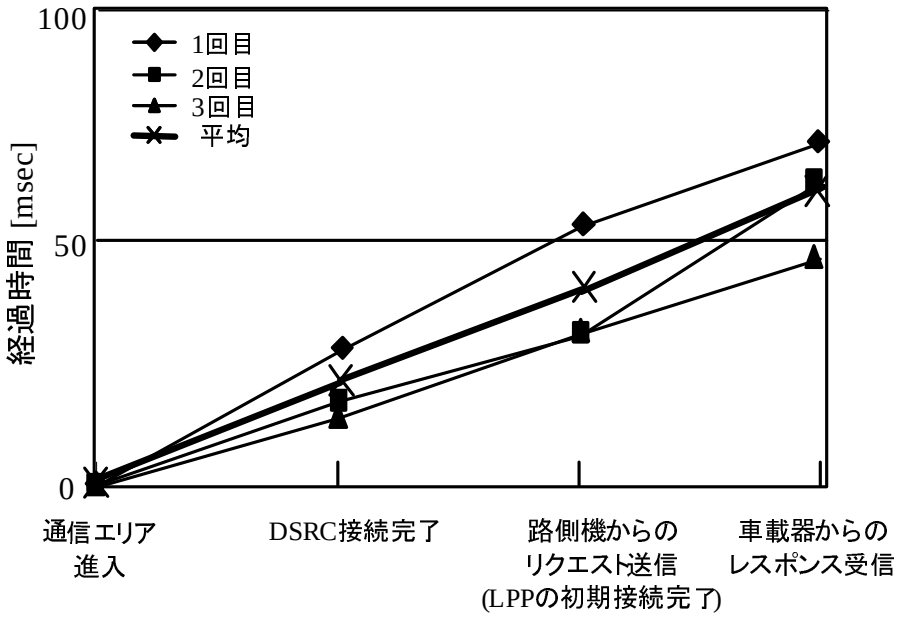


図 2.7: 車載器 ID タグアプリケーションを用いた LPP の初期接続時間の計測結果

2.4.2　走行車両への通信量の見積り

本節では，LPCP/LPP をフル実装したプロトタイプシステムを用いた通信実験の結果に基づき LPCP/LPP の走行車両に対する通信量を見積もった結果について述べる．なお，本実験における試験条件を表 2.5 に示し，図 2.8 に試験シーケンスと計測区間を示す.

表 2.5: 試験条件

通信方式	個別通信
接続方法	RF ケーブル接続
変調方式	QPSK (Quadrature Phase Shift Keying)
スロット数	4
LPP 機能	フル実装

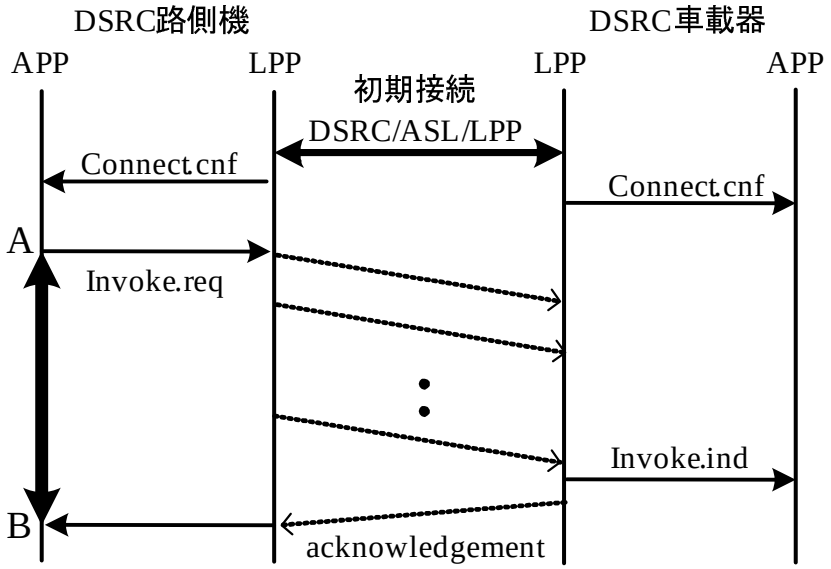


図 2.8: 試験シーケンスと計測区間

LPCP/LPP において走行車両への通信量を見積もるため，単方向データ通信トランザクションにおいて，5[KByte]，50[KByte]，100[KByte] のデータを用いて，ト

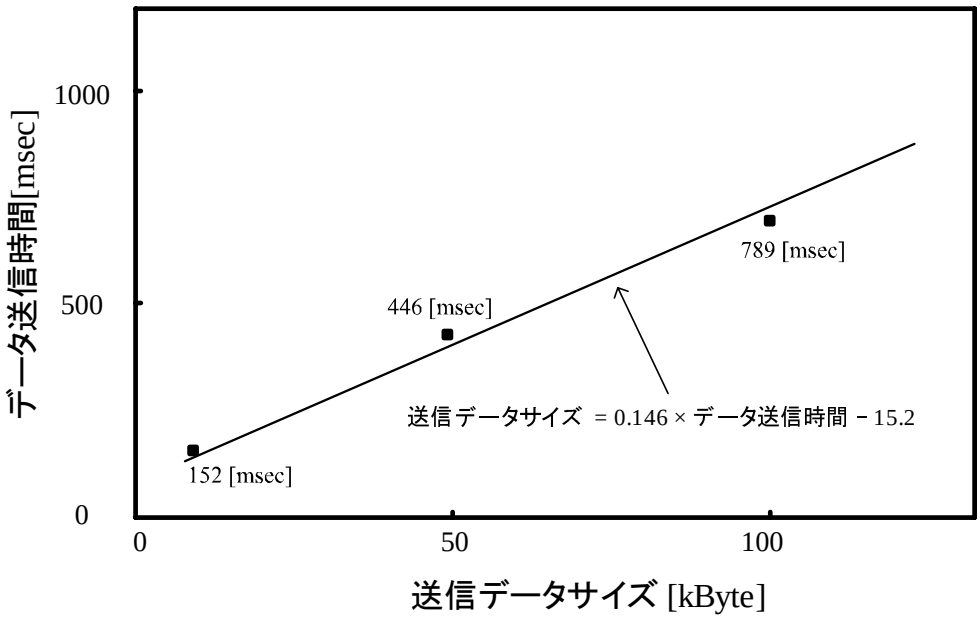


図 2.9: 単方向データ送信トランザクションにおけるデータ送信時間の計測結果

ランザクション開始から，送信データの確認応答が到着するまでの時間 (図 2.8 の A と B の間の時間) を計測した結果を図 2.9 に示す.

図 2.9 のデータ送信時間の計測結果から，実験によって，単方向データ送信トランザクションのスループットは約 1150[kbps] という結果が得られる．表 2.5 の試験条件において，全てのスロットをダウンリンクで使用した場合の，ASL の通信制御プロトコル部における理論上の最大スループットが 1390[kbps] であることから，この結果は，理論値に対して，約 83%の速度であり，通信制御プロトコル部におけるクライアント/サーバ通信制御などのオーバーヘッドの影響を考慮すると，十分な速度が出ているといえる.

また，この結果から，LPCP/LPP の初期接続時間を 50[msec] と仮定すると，車両が通信可能エリアに滞在する時間を T[msec] とした場合の通信可能データサイズ D[KByte] は,

$$D = 0.146 \times (T - 50) - 15.2 \tag{2.1}$$

で表すことができる．表 2.6 に式 (2.1) を用いて算出した通信エリア長及び車両の速度と通信可能な最大データ長の関係を示す.

表 2.6 から，例えば，通信エリア長が 30[m]，車両の速度が 120[km/h] の場合に

表 2.6: 通信エリア長及び車両速度と通信可能な最大データ長の関係

通信エリア長 [m]	車両速度 [km/h]	通信可能時間 [msec]	最大データ量 [KByte]
10	40	900	116
20	60	1200	160
20	120	600	72
30	60	1800	248
30	120	900	116

は，エリア滞在時間 T が 900[msec] となり，最大で約 116[KByte] のデータが送信可能であることが分かる．この 116[KByte] というデータサイズは，64[kbps] の音声データ†であれば約 15 秒の音声に相当する．以上の結果から，LPCP/LPP を走行中の車両に対する情報配信サービスの転送プロトコルとして適用可能であるとの見通しが得られた．

2.5 結言

本章では，走行中の車両や低リソースな車載システムに対するサービスをターゲットとした DSRC ローカル通信プラットフォームを提案した．DSRC ローカル通信プラットフォームは，IP 系と同様にマルチアプリケーションを実現する転送プロトコルと，サービスに共通の基本的なアプリケーション機能を提供するアプリケーションプロトコル群からなるプロトコルスタックとして構成されるので，IP 系と同様に基本的なアプリケーションプロトコルの組合せで多様なサービスの実現を可能とする．

また，DSRC ローカル通信プラットフォームにおける転送プロトコルを，LPCP とその拡張プロトコルである LPP の 2 階層で構成し，下位レイヤである LPCP を軽量なプロトコルとするともに，LPP については必要な機能のみの選択的な実装を可能とすることで，低リソース車載器への実装を実現した．さらに，これらのプロトコルはネットワーク層を持たず，コネクションレス型の通信プロトコルとして

†例：リニア PCM, モノラル, サンプリングレート 8KHz 8bit

設計されるので，初期接続時間が最小化され，高速走行する車両に対しても双方向通信が可能という DSRC が有するメリットを本提案プラットフォームに継承させることに成功した．

これらのプロトコルを実際の DSRC 路側機と DSRC 車載器上に実装し，その性能を測定した結果，1) 低リソース車載器への実装可能性と 2) 高速な初期接続性の 2 つの性能要件を満たしていること，を確認すると共に，スループットを計測することで，走行中の車両に対する通信量を見積もった．これらの結果から，本プラットフォームを走行中の情報配信サービスに適用可能であるとの見通しを得た．

このような成果に基づき，本論文で提案したプロトコルは，2003 年 9 月より実運用を開始した駐車場の入退場管理・料金決済システムで採用されたほか，DSRC アプリケーションサブレイヤ標準規格 (ARIB STD-T88)[60] の非ネットワーク系アプリケーションのための通信プロトコルとして標準化されている．

第3章　車両へのプッシュ型情報配信サービスに適したアプリケーションプロトコル

3.1　序言

DSRCを用いる様々なサービスのなかでも，走行中の車両に対して，安全運転支援情報や駐車場での店舗情報など，位置に特化した情報をプッシュ配信することにより，運転の安全性や快適性を向上させるサービスは，DSRCの特長を生かしたサービスとして，料金決済と並び，最も期待されるサービスの一つである．

第2章では，このような走行車両に対するサービスを実現するための非IP系の通信プロトコルについて述べたが，この通信プロトコルはトランスポート層のプロトコルとして様々なアプリケーション間でのデータ伝送に必要な機能のみを定義したものであり，上述したような様々なプッシュ型サービスを実現するには，新たにアプリケーションプロトコルを構築する必要がある．

これまでにDSRCを用いて位置に特化した情報をプッシュ配信するサービスに関する研究として，文献[61]などが存在する．しかしながら，これらの論文の主眼は，IPマルチキャストの高信頼化など路側ネットワークの高度化などに関するものであり，アプリケーションプロトコルに関する研究は存在していない．

一方，インターネットや携帯電話網の世界では，プッシュ型情報配信サービスが開始されており，プッシュ機能を実現するためのプロトコル[62–64]が規定されている．例えば，WAP (Wireless Application Protocol) プッシュ[64]では，基地局から移動機へのデータのプッシュ機能の実現や，移動機アプリケーションのアドレッシング，および各種制御情報の交換などの機能が定義されており，サーバ側が任意のタイミングで移動機に対してコンテンツをプッシュすることが可能な仕様となっている．しかしながら，携帯電話を対象に検討されたものであることから，

- 広域通信での利用が前提であり，制御情報の交換でコネクション型のプロトコルを使用するなど，狭域通信で重要な通信可能時間に対する考慮はなされていない
- クライアントが携帯電話という単一のデバイスで構成されることが前提のため，DSRCが対象とする'DSRC車載器＋カーナビゲーション'のようなク

ライアントが複数の機器で構成される場合に必要となる機能 (例えば通信機器と外部機器で受信可能なデータサイズが異なる場合など) が定義されていない

といった点で，DSRC による車両へのプッシュ配信には適しているとは言えない．

そこで，本研究では，DSRC を用いて走行中の車両に対する情報配信サービスを実現することを目指し，通信可能時間への考慮や様々な機器構成への対応など，狭域通信や車載機器特有の条件を考慮したプッシュ型情報配信アプリケーション [44–48] を提案する．

本章では，まず DSRC によるプッシュ型情報配信サービスのアーキテクチャについて述べる．次に提案するアーキテクチャを実現するために開発したアプリケーションプロトコルについて述べ，実機を用いた検証によって，車両への多種多様な情報配信サービスに適用可能であることを示す．

3.2 プッシュ型サービスのアーキテクチャ

3.2.1 要求条件

本節では，提案するプッシュ型情報配信アプリケーションが満たすべき要求条件について述べる．

まず提案するプッシュ型情報配信アプリケーションは，プッシュ型サービスのプラットフォームとして，

【要求条件 1】多種多様なコンテンツを，通信接続中の任意のタイミングで配信でき，操作レスでの情報再生を行える

必要がある．

また，提案するプッシュ型情報配信アプリケーションは，DSRC による走行車両へのプッシュ配信をそのターゲットとしていることから，前節であげた狭域通信や車載機器特有の条件を考慮した仕様でなければならない．したがって，限られた通信可能時間を有効に利用可能とするために，

【要求条件 2】通信接続からコンテンツ配信開始までの時間が短いこと

や，低リソースなDSRC車載器単体システムから比較的高リソースなカーナビゲーション連携システムまで，多種多様な車載システム[65,66]に対応するため，

【要求条件3】車載システムの構成・能力に応じたコンテンツ配信が可能なこと
【要求条件4】サービスの追加や拡張に機器構成の変更やカーナビゲーションなど比較的リソースが大きい機器でのソフトウェア(以下，S/W)更新により対応可能なこと

などが要求条件となる．

3.2.2　プロトコルスタック

図3.1に提案するプッシュ型情報配信サービスのためのプロトコルスタックを示す．

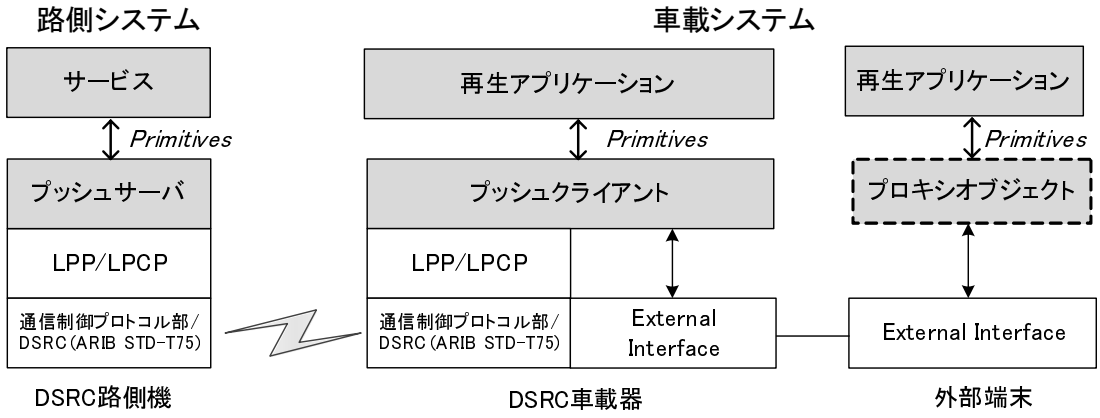


図3.1: プッシュ型情報配信サービスのためのプロトコルスタック

図3.1に示すように，提案するアーキテクチャでは，プッシュ型情報配信アプリケーションを，第2章で提案したLPP/LPCPを下位層として利用するDSRCローカルアプリケーションとして構築する．

また，車載機器がDSRC車載器とカーナビゲーションシステムなどの外部機器から構成されることを想定し，車載システムにおけるアプリケーションの構成およびその規定範囲を路車間通信部分を担当するプッシュプロトコルとコンテンツの再生処理を行う再生アプリケーションの2階層構成とするとともに，外部機器上にプッ

表 3.1: プッシュプロトコルのプリミティブ一覧

プッシュサーバ側

プリミティブ	パラメータ	概要
Push.req	アプリケーションタイプ コンテンツタイプ コンテンツサイズ コンテンツ	プッシュ配信要求
ClientInfo.ind	アプリケーションタイプリスト コンテンツタイプリスト 最大コンテンツサイズ	再生アプリケーション情報通知

プッシュクライアント側

プリミティブ	パラメータ	概要
Push.ind	コンテンツタイプ コンテンツサイズ コンテンツ	プッシュ受信通知
RegisterApplication	コンテンツタイプリスト 最大コンテンツサイズ	再生アプリケーション情報登録

シュプロトコルのインタフェースを提供するプロキシオブジェクトを用意することで，再生アプリケーションを外部機器上に搭載することを実現する．

提案するプッシュプロトコルは，DSRC を用いて路側からの自発的な情報配信を実現するために新たに設計したプロトコルであり，前述した要求条件を満足するために，上位層である路側のサービスと車載側の再生アプリケーションに対して，以下のサービスを提供する．

(1) 路側のサービスから車載の再生アプリケーションへ，任意のコンテンツをプッシュ配信するサービス
(2) 初期接続時に車載システムの再生アプリケーションの情報を，路側のサービスに対して通知するサービス
(3) 再生アプリケーションの情報を登録するサービス

表 3.1 にプッシュプロトコルが，上位層である路側のサービスや車載の再生アプリケーションに提供するインタフェースを示す．

(2) のサービスにより路側のサービスが接続と同時に再生アプリケーションの情報を取得し，配信するコンテンツを選択，(1) のサービスを用いて，コンテンツをプッシュ配信することにより，車両毎に適したコンテンツの配信が実現できる (要求条件 1,3).

また，(3) のサービスを外部機器上の再生アプリケーションが利用することで，DSRC 車載器を一切変更することなく，外部機器側の S/W 更新や追加によってサービスの発展・拡張を行うことを可能とする (要求条件 4).

一方，走行車両に対するサービスを実現するため，(1) のプッシュ配信サービスでは，コネクションレス型サービスのみを提供することとし，(2) の初期接続に係わる手順を最小化することで，LPP/LPCP の有する高速な初期接続性能を活かすことが可能な仕様とする (要求条件 2).

3.3　プッシュ型情報配信のためのプロトコル

前節で述べた各サービスを実現するため，プッシュプロトコルを以下の機能を有するように設計した．以下本章では，路側システム側をプッシュサーバ，車載システム側をプッシュクライアントと呼ぶ.

- 配信コンテンツの再生制御機能
  (1) のサービスを実現するため，配信コンテンツに再生アプリケーションと配信コンテンツの種類を付加し，これらの情報を元にプッシュクライアント側で適切な実行機器およびアプリケーションを特定し，受信コンテンツを転送する機能を定義した．なお，実行機器やアプリケーションの特定に必要となる情報は (3) のサービスで登録された情報を利用する.
- 車載リソース情報通知機能
  (2) のサービスを実現するため，初期接続時に，車載システムのリソース情報として再生アプリケーションの情報や DSRC 車載器のバッファサイズなどを路側機に通知する機能を定義した．通知する再生アプリケーションの情報は (3) のサービスにより登録する.

表 3.2: プッシュプロトコルで使用するメッセージの一覧

メッセージ (コマンド)	概要	送信方向	主なパラメータ
PushOperation	コンテンツ 配信用コマンド	サーバ ↓ クライアント	分割フラグ アプリケーションタイプ コンテンツタイプ プッシュID コンテンツサイズ 情報
NextSegment	セグメント 配信用コマンド	サーバ ↓ クライアント	プッシュID セグメント番号 情報
NextSegmentRequest	次セグメント データ要求 メッセージ	サーバ ↑ クライアント	プッシュID
PushAbort	破棄通知コマンド	サーバ ↓↑ クライアント	プッシュID 破棄コード
ClientInformation	車載リソース情報 通知用コマンド	サーバ ↑ クライアント	アプリケーションタイプリスト コンテンツタイプリスト 受信バッファサイズ 最大コンテンツサイズ

3.3.1 メッセージ定義

上述した各機能を実現するために，プッシュプロトコルで定義しているメッセージと主なパラメータを表 3.2 に示す．

PushOperation は，プッシュサーバからプッシュクライアントに対してコンテンツもしくは分割コンテンツの最初のデータを配信するためのメッセージである．アプリケーションタイプやコンテンツタイプおよび複数のプッシュを識別するためのプッシュID などをパラメータとして持ち，前述した再生制御機能で使用される．

NextSegmentRequest および NextSegment は大容量データの分割配信のためのメッセージであり，プッシュID によりコンテンツを識別する．

PushAbort は，プッシュトランザクションの破棄を通知するためのメッセージであり，プッシュクライアント側で対応不能なアプリケーションタイプやコンテンツタイプが指定された場合などに，トランザクションの失敗をプッシュクライアント

からプッシュサーバに通知する目的などに使用される．

ClientInformation は初期接続時に実行可能なアプリケーションタイプ/コンテンツタイプや，受信バッファのサイズなどのプッシュクライアントの車載リソース情報をプッシュサーバに対して通知するためのメッセージである．なお，これらのメッセージのうち PushOperation 以外のメッセージは個別通信時のみ使用される．

以下，本節ではこれらの各メッセージを用いて，各機能をプッシュプロトコルがどのように実現するかについて詳述する．

3.3.2　配信コンテンツの再生制御機能

路側機から配信されたコンテンツを，複数の機器から構成される車載システムにおいても，適切な機器で自動的に再生することができるよう，配信コンテンツの再生制御機能を定義する．図 3.2 に，本機能の概要を示す．

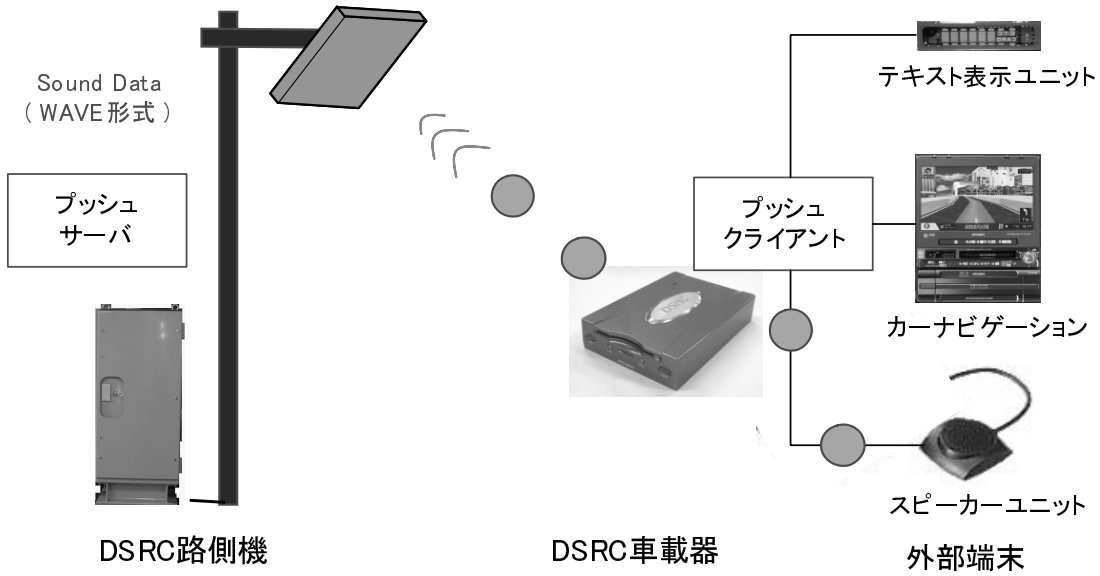


図 3.2: 配信コンテンツの再生制御機能

本機能では，プッシュサーバから配信されるメッセージ (PushOperation コマンド) に，再生アプリケーションとコンテンツの種類を表す識別子（アプリケーションタイプ/コンテンツタイプ）を付与して配信し，プッシュクライアント側でこれらの識別子の組み合わせにより配信されたコンテンツに適した機器および再生アプリケーションを特定し，様々なコンテンツの自動再生を実現する．

表 3.3: 代表的なアプリケーションタイプ

アプリケーションタイプ	アプリケーション
text-display	テキスト表示
image-display	画像表示
sound-player	音声再生
browser	Web ブラウザ

表 3.4: 代表的なコンテンツタイプ

コンテンツタイプ	内容
text/plain	プレーンテキスト
text/tts	TTS (Text To Speech) 中間言語
image/png	PNG (Portable Network Graphics) 形式の画像
audio/wav	WAVE 形式の音声
dsrc/smart-pull	擬似プッシュ

ここでコンテンツタイプはコンテンツの種類を表す識別子であり，配信されるコンテンツのフォーマットを表す．アプリケーションタイプは画像表示やテキスト表示といった再生アプリケーションでの再生方法を表す識別子であり，再生アプリケーションは，これらの機能要件を満たすことができれば，どのような再生アプリケーションを利用してもかまわない．表 3.3，表 3.4 に，プッシュ型情報配信アプリケーションで規定済みの代表的なアプリケーションタイプ及びコンテンツタイプを示す．プッシュプロトコルを利用するサービスが，表 3.3, 表 3.4 などの規定済みの標準的なアプリケーションタイプやコンテンツタイプを利用することでサービスレベルでの相互接続性を担保することができる．一方，新規にアプリケーションタイプやコンテンツタイプを追加し，車載システム側に再生アプリケーションを追加することにより，将来のサービス追加や拡張にも容易に対応が可能である．

なお，WAP プッシュ[64] にも同様の機能として，アプリケーション ID によるアプリケーションアドレッシング機能が定義されているが，提案プロトコルでは，ア

プリケーションタイプに加えて，コンテンツタイプもアプリケーションの特定に利用している点が異なる．これは，あるアプリケーションが複数の機器でサポートされ，かつそのアプリケーションが対応可能なコンテンツの種別がそれぞれの機器で異なる場合にも，適切な機器への転送を実現するためである．

3.3.3　車載リソース情報通知機能

車載システムの構成・能力に応じたコンテンツ配信を実現するため，本プロトコルではプッシュクライアントからプッシュサーバに対して，実行可能なアプリケーションタイプ/コンテンツタイプや，受信バッファのサイズなどの車載リソース情報 (ClientInformation) を，初期接続時に通知する機能を定義した．路側システムのサービスやプッシュサーバがこの情報を参照することにより，車両毎のシステム構成・リソースに応じた最適なコンテンツの配信が実現できる．表 3.5 に ClientInformation に格納される情報の一覧を，図 3.3 に，リソース情報通知機能を利用したコンテンツ配信シーケンスを示す．なお，本機能は車側から路側へのアップリンクが可能な個別通信によるサービスでのみ利用可能な機能であり，同報通信によるサービスではコンテンツタイプ毎に，最大コンテンツサイズなどの必要なクライアントリソースを定義しておくことで対応する．

表 3.5: ClientInformation で通知する主な車載器のリソース情報

データ種別	内容
アプリケーションタイプリスト	車載システムが扱うことが可能な音声再生や画像表示の指示，または再生機器の種別など
コンテンツタイプリスト	車載システムが扱うことが可能な WAVE や PNG などのメディアデータの種別
受信バッファサイズ	車載器が配信されたデータを一次格納できるデータサイズの最大値
最大コンテンツサイズ	車載システムが扱うことのできる最大のコンテンツサイズ

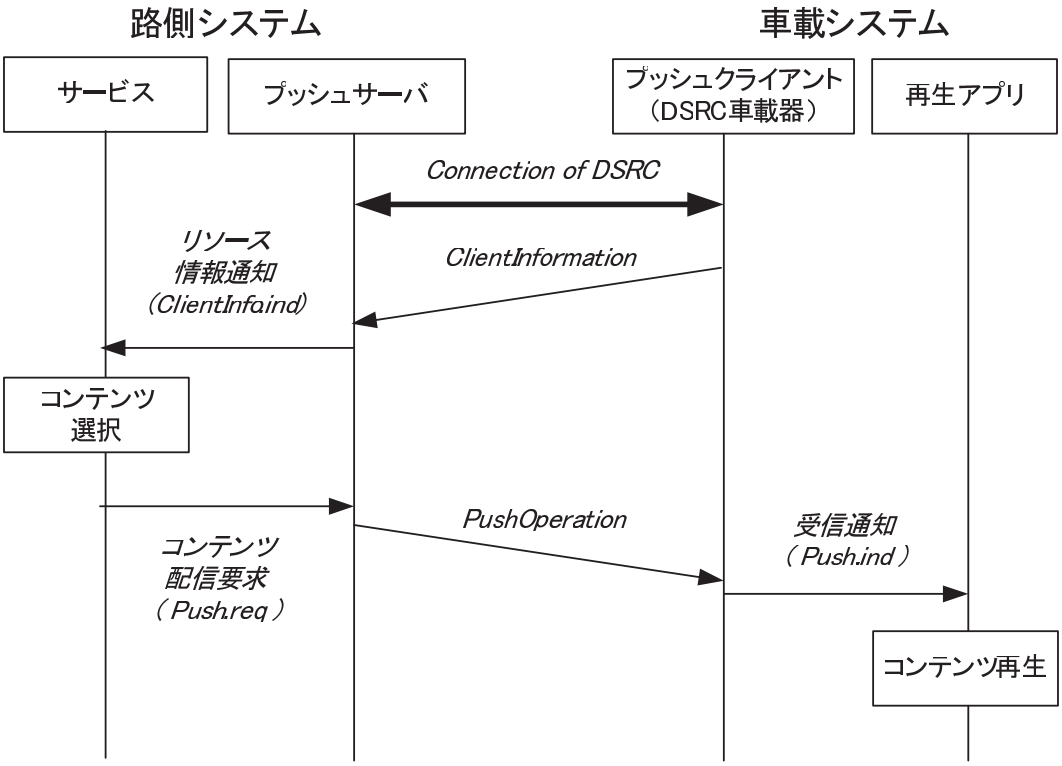


図 3.3: リソース情報通知機能のシーケンス例

以下，この車載器リソース情報を利用した拡張機能として大容量コンテンツの分割配信機能について述べる．この機能は，車載システムがDSRC車載器と外部機器で構成される場合に，DSRC車載器のバッファサイズを超えるコンテンツの配信を実現する機能であり，停止中のサービスなどにおいて，動画や音声など大容量のコンテンツを配信する場合に使用する．本機能の基本的な動作は以下の通りである．図3.4に大容量コンテンツの分割配信機能のシーケンス例を示す．

(1) プッシュサーバは，路側のサービスからPush.reqプリミティブにてプッシュ配信要求を受信すると，配信コンテンツをリソース情報取得機能で取得した受信バッファのサイズに分割して，配信する．このとき先頭セグメントの送信には配信コンテンツ全体のサイズなどの情報を通知するため，分割フラグを1としたPushOperationコマンドを使用する．

(2) プッシュクライアントは，3.3.2節で述べた再生制御機能により，受信したコンテンツを外部機器のプロキシオブジェクトに転送する．その際，プッシュクライアントは転送先とプッシュIDを記憶する．

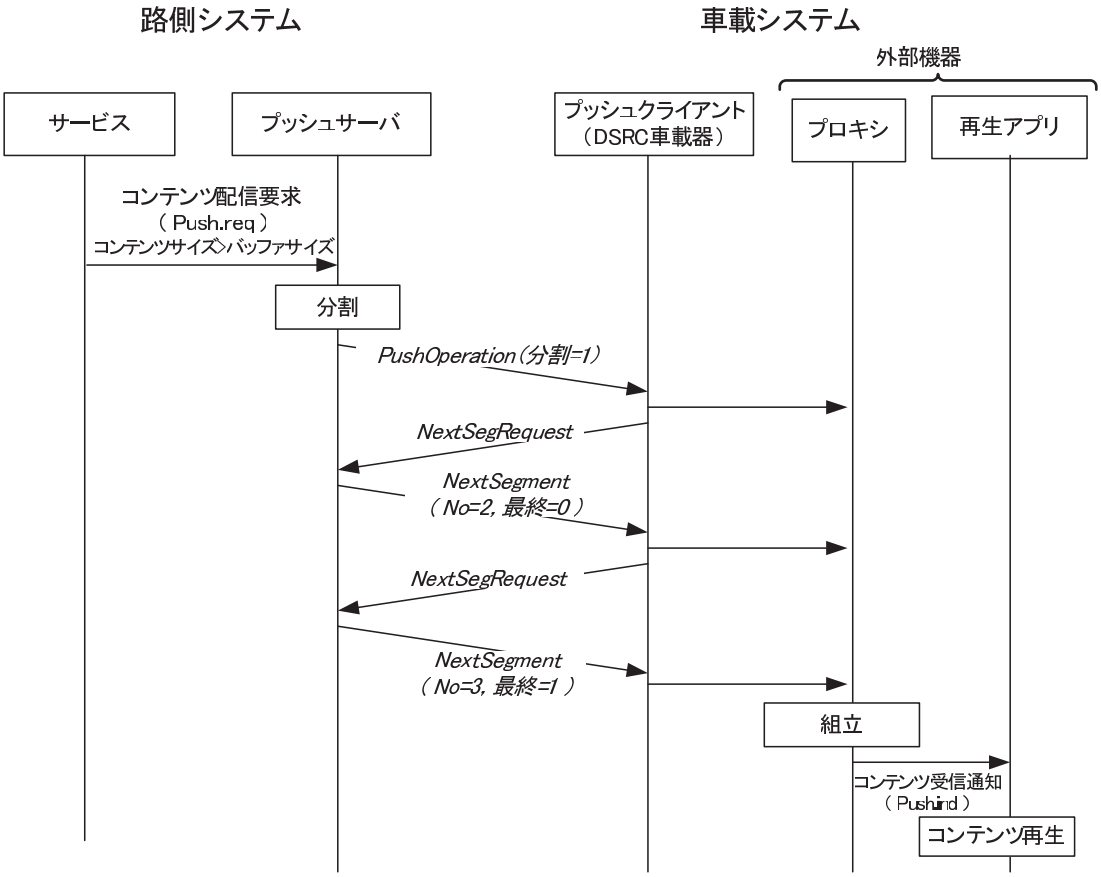


図 3.4: 大容量コンテンツの分割配信機能のシーケンス例

(3) (2) で転送を完了するなど，プッシュクライアントが次のデータを受信可能となったタイミングでプッシュサーバに NextSegmentRequest コマンドを送信する．

(4) (3) で送信された NextSegmentRequest コマンドをプッシュサーバが受信すると，次セグメントの送信を，NextSegment コマンドで行う．このような手順とすることで，分割データの送信フローの制御を実現し，車載器のリソースを超えたデータが一度に配信されることで，配信データが欠落してしまうことを防止する．

(5) (4) で送信された NextSegment コマンドを受信したプッシュクライアントは，プッシュID をキーに (2) で記憶した転送先を取得し，受信したコンテンツを外部機器のプロキシオブジェクトに対して転送する．

(6) 外部機器上のプロキシオブジェクトが，(2) および (5) で DSRC 車載器から

転送された分割データを組立て，最終セグメント（受信データサイズにより判断）の受信後に，Push.ind プリミティブにて再生アプリケーションに通知，再生処理を行う．

本機能により，比較的低リソースな DSRC 車載器であっても，DSRC 車載器側は一切変更することなく，接続する外部機器を変えるだけで，大容量コンテンツに対応した車載システムを実現することができる．

一方，本機能では，分割サイズに，固定の値ではなく DSRC 車載器のバッファサイズを利用することで．下位層の LPP が持つ高速な分割転送機能を最大限に活用することが可能な仕様としている．

3.4 実装と評価

本節では，提案したプロトコルの有効性を確認するため，実際の DSRC 路側機，DSRC 車載器上に実装したプロトタイプシステムを用いて，通信実験を実施した結果に基づき，転送機能の有効性の確認結果と，走行車両に対する通信量を見積もった結果について述べる．

なお，本実験における試験条件は 2.4.2 節の表 2.5 の試験条件と同一とした．

3.4.1 再生制御機能の確認

本実験では，構成・リソースの異なる 3 種類の車載システムを用いることで，車載器リソース情報通知機能と再生制御機能がどのように動作するかを確認する．図 3.5 に，本実験で使用する実験システムの構成を示す．

本実験システムは，DSRC 路側機とその路側機に接続された音声や画像などの情報を配信する情報配信サーバから構成される路側システムと，DSRC 車載器及び本車載器に接続されたカーナビゲーションシステムやスピーカーユニットなどの外部機器から構成される車載システムからなる．

路側の情報配信サーバには，配信コンテンツとして，

- WAVE 形式の音声ファイル [24KByte]
- TTS 中間言語のファイル [1KByte]
- PNG 形式の画像ファイル [12KByte]

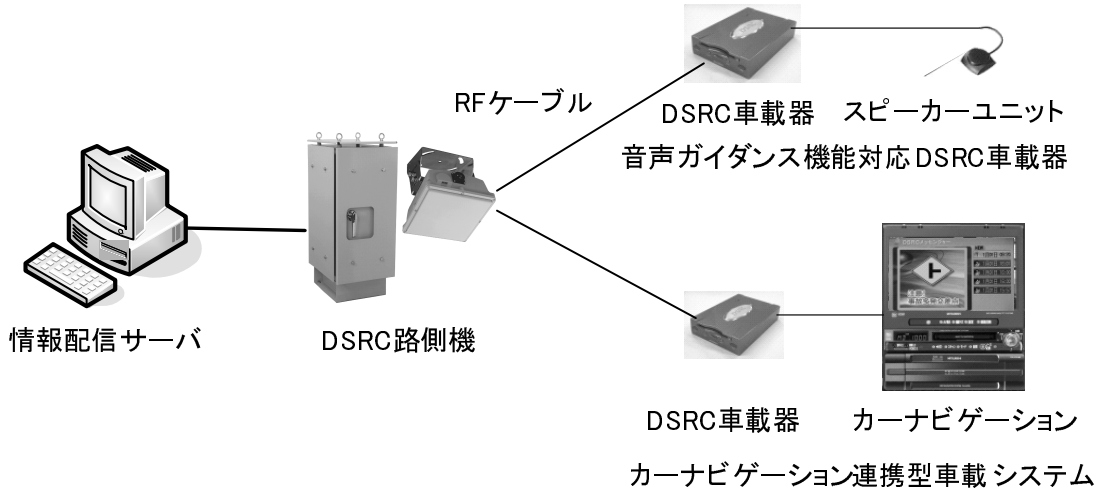


図 3.5: 実験システムの構成

を用意し，初期接続時に受信する ClientInformation のコンテンツタイプリストに含まれるコンテンツを選択して配信する．

一方，車載システムとして，

(1) 外部機器としてスピーカーユニットを接続．WAVE 形式の音声ファイルの再生に対応．受信バッファサイズは 32[KByte].
(2) 外部機器としてカーナビゲーションシステムを接続．TTS 中間言語と PNG 形式の画像ファイルの再生に対応．受信バッファサイズは 32[KByte].
(3) 外部機器としてカーナビゲーションシステムを接続．TTS 中間言語と PNG 形式の画像ファイルの再生に対応．受信バッファサイズは 8[KByte].

の 3 種類を用意し，実験では，これら 3 つの車載システムをそれぞれ DSRC 路側機に接続させ，車載システム側がどのように動作するかを確認した．

実験の結果，スピーカーユニットが接続された車載システムでは，WAVE 形式の音声ファイルが再生され，カーナビゲーションが接続された車載システムでは (2),(3) ともに TTS による音声と PNG 形式の画像がそれぞれ再生された．

以上の実験結果から，

- 配信コンテンツの再生制御機能の働きにより，配信コンテンツに適した再生アプリケーションの自動選択・再生が可能なこと

- リソース情報通知機能の働きにより，車載システムが有するリソース・機能に応じて異なるコンテンツの配信が可能なこと
- 分割転送機能により，低リソースな車載器であっても接続する外部機器により，自身の受信バッファサイズを超えるコンテンツを受信・再生可能なこと

が確認できる．

3.4.2 走行車両への通信量の見積り

DSRC のような狭域通信による走行車両に対するプッシュ型情報配信サービスでは，通信可能時間が限られるため，サービス設計段階で，通信量の見積を行っておく必要がある．そこで，プッシュ配信機能における走行車両への通信量を見積もるため，5[KByte]，50[KByte]，100[KByte] のコンテンツを用いて，車載器が通信エリアに進入してから，配信コンテンツが DSRC 車載器に到着するまでの時間を計測した．なお，使用した車載器の受信バッファサイズは 100[KByte] 以上とした．実験結果を表 3.6 および図 3.6 に示す．

表 3.6: プッシュ型情報配信の所要時間の測定結果

	メッセージサイズ	結果
リソース情報通知機能	-	52 msec
プッシュ配信機能	5KByte	152 msec
	50KByte	446 msec
	100KByte	789 msec

なお，下位層である LPP の初期接続時間を，第 2 章の実験結果に基づき 50[msec] と仮定すると，$L[m]$ の通信エリアを車速 $v[m/s]$ で走行した場合の，本プロトコルにおける通信可能データサイズ $M_{size}[KByte]$ の計算式は，図 3.6 より，

$$M_{size} = 146 \times L/v - 29.8 \tag{3.1}$$

と導くことができる．この計算式から得られるスループットは，2.4 節における LPP のスループット 1150[kbps] とほぼ同一の値であり，プッシュプロトコルのオーバーヘッドが極めて小さいものであることが分かる．

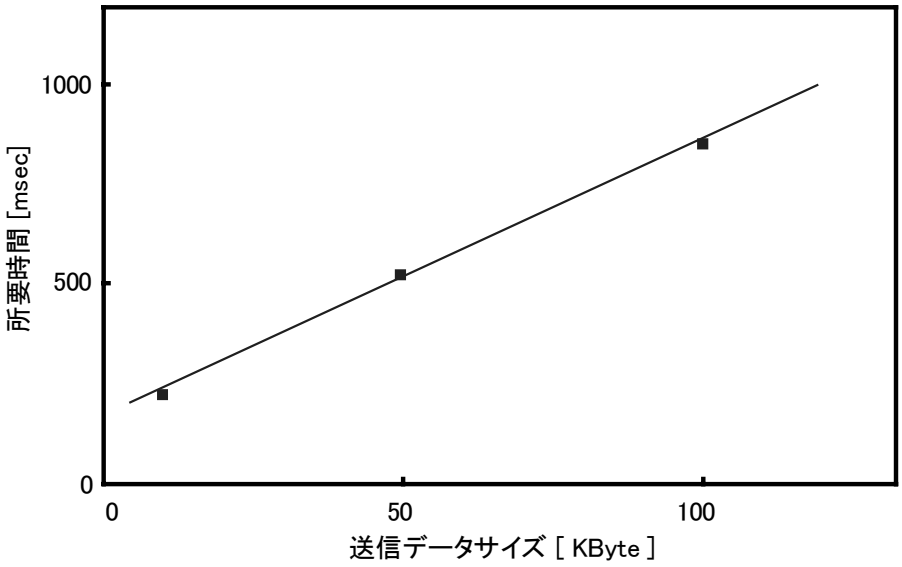


図 3.6: プッシュ配信機能におけるデータ送信時間の測定結果

次に，通信エリア内に N 台の車両が存在する場合について考察する．DSRC では，同時通信台数が 1 フレームあたりのスロット数以下の場合には，その初期接続時間は台数によらず一定であることから，その通信可能時間も台数によらず一定となる．一方で，データ配信時には利用可能なスロットを複数台で共有するため，そのスループットが $1/N$ となることから，車両 1 台あたりの通信可能データサイズ $m_{size}[KByte]$ は

$$m_{size} = M_{size}/N \qquad (N \leq 1\,\text{フレームあたりのスロット数}) \tag{3.2}$$

となる．

通信可能時間が問題となる高速走行車両へのサービスでは，車間距離を考慮すると，通信エリア内に同時存在する車両台数は，多くの場合 4 台以下である．したがって，スロット数が 4 以上の通信プロファイルを使用すれば，式 (3.1)(3.2) により，同時通信可能台数，走行速度，通信エリアサイズから通信可能なデータサイズが容易に計算でき，サービスはこの値に基づいて配信するデータを用意することができる．図 3.7 に通信エリアが 20[m] の場合の 1 台あたりの通信可能データサイズと車速の関係を示す．例えば車両 3 台で 20m の通信エリアを 100[km/h] で走行した場合の通信可能なデータサイズを求めると，約 25[KByte] となる．

なお，走行車両への情報配信サービスでは，安全性の観点から，音声を用いた

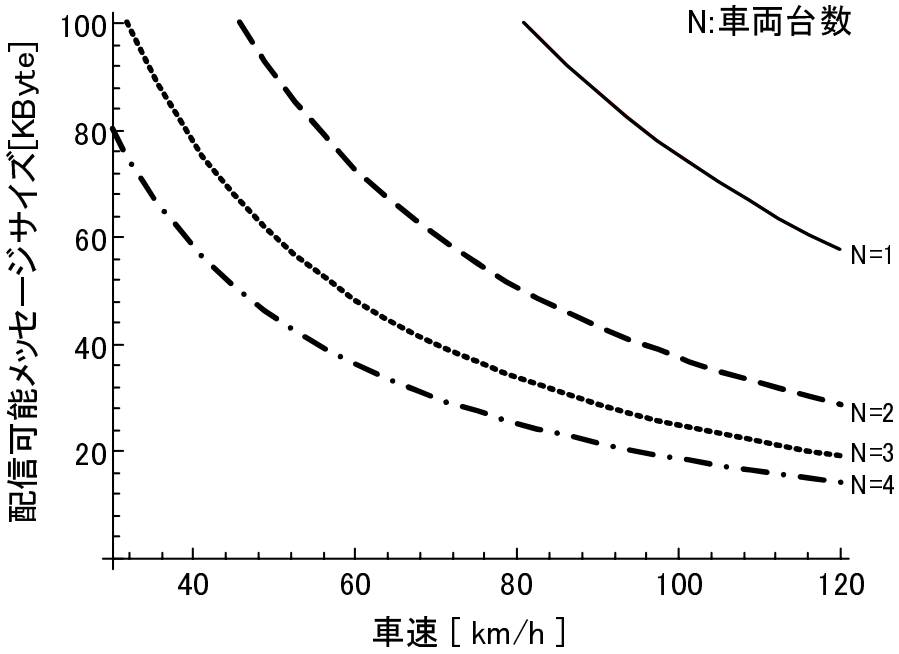


図 3.7: 1 台あたりの通信可能データサイズと車速の関係(通信エリア 20[m] の場合)

サービスが有効である．上記の値は，配信データとして TTS (Text to Speech) を利用すれば 1 分以上の，CELP（Code-Excited Linear Prediction）など 8[kbps] 程度の高圧縮音声であれば約 25 秒の長さの音声再生が可能な値であり，通信可能データ量としては十分な値であると考える．

3.5 結言

本章では，走行車両に対する情報配信サービスをターゲットとしたプッシュ型情報配信アプリケーションのアーキテクチャとプロトコルについて提案した．

走行車両へのサービスを実現するために，LPP/LPCP 上に，ローカルアプリケーションとしてプッシュ型情報配信アプリケーションを設計した．さらに，プッシュ型情報配信アプリケーションを通信部であるプッシュプロトコルと再生アプリケーションに分離するとともに，プッシュプロトコルに車載システム側での操作レスでの自動再生が可能となる機能や，車載システムに応じた様々なコンテンツの配信を行う機能などを定義したことにより，様々な形態の車載システムに対する情報配信サービスを可能とした．

また，提案したプロトコルを実際のDSRC路側システムとDSRC車載システムに実装し，その初期接続性能とスループットを計測することで，走行車両に対する情報配信サービスに適用可能であるとの見通しを得ると共に，複数の車載システムとコンテンツを用意し，車載システムに応じた最適なコンテンツを配信する実験を実施することで，様々な車載システムに対するサービスへの適用が可能なことを確認した．

なお，本章にて提案したプロトコルは，ITS情報通信システム推進会議が規定したDSRC基本アプリケーションインタフェース仕様ガイドライン[67]として標準化されている．また，本アプリケーションは，ITS車載器[68–70]と呼ばれる標準的な車載器に備わる一機能として採用されており，次世代道路サービス提供システム[71]において，道路交通情報提供や安全運転支援情報提供等のサービスに適用されている．

第4章　複数路側機連携サービスのためのDSRC通信プラットフォーム

4.1　序言

DSRCによって実現が期待されるサービスの中には，料金決済や安全運転支援などの単一の路側機によるスポット型のサービスに加えて，異なる路側機下に存在する車両間でデータを交換する車路車間通信サービス[72]や大規模駐車場などにおいて入口と出口など複数の路側機に跨って連続したサービスを提供するなど，複数のDSRC通信エリアを跨った空間的な広がりのあるサービスも存在する．

しかしながら第2章や第3章で提案したDSRCローカル通信プラットフォームは，路側機/車載器ローカルでアプリケーションが実行されることを前提としているため，このような複数の路側機間の連携が必要な面的なサービスを実現するには，サービス毎に専用の連係機能を開発しなければならず，開発効率が低下するという問題がある．このような複数の路側機連携が必要となるサービスでは，設置場所の形状や路側機の数などに応じて，路側機間の連携方法が多様化していくことが想定されることから，連携サービスの構築に必要な機能をプラットフォームが共通機能として提供することで，サービスの開発効率の向上が期待できる．

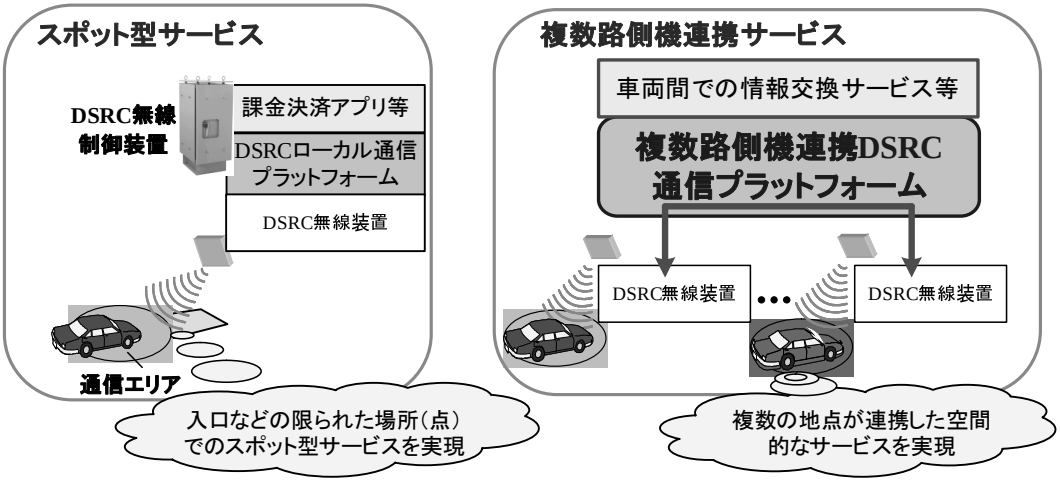


図 4.1: スポット型サービスと複数路側機連携サービス

そこで本研究では，複数の路側機を利用して，位置に依存したサービスを行うア

プリケーションの開発効率の向上を目的とした DSRC ローカル通信プラットフォームの拡張方式 [49, 50] を提案する（図 4.1）．

本章では，まず提案するサーバー集約方式による複数路側機連携方式のメリットとデメリットについて述べ，デメリットである初期接続性能や通信効率の劣化を最小限に抑えることが可能なプロトコルの分割方式と連携機能について述べる．また異なる路側機下に存在する車両間でリアルタイムに情報を交換する車路車間通信サービスを実装し，その遅延時間などについて評価を行うことで，本方式の有効性を示す．

4.2　複数路側機連携 DSRC 通信プラットフォームのアーキテクチャ

4.2.1　サーバー集約方式による複数路側機連携方式

複数の路側機間の連携が必要なサービスの構築方法として，連携サービスを一つの機器に集約し，その機器が各路側機を制御してサービスを行う方式が考えられる．図 4.2 にサーバー集約型での複数路側機連携方式の概要を示す．

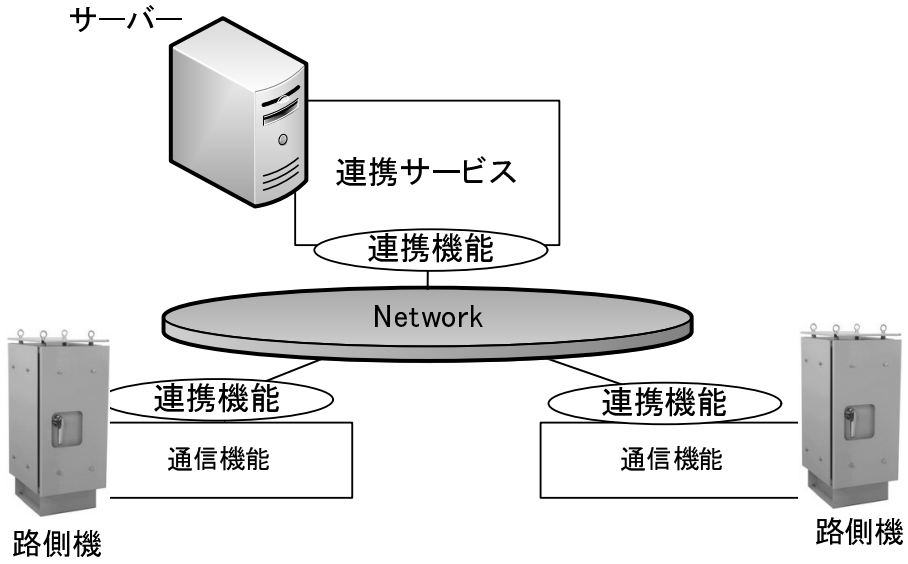


図 4.2: サーバー集約方式による複数路側機連携の概要

本方式は，路側機を単なる通信装置と見なし，サーバー上のサービスアプリケーションが複数の路側機を制御する方式である．この方式では，サービス毎の開発はサーバー上のサービスアプリケーション開発のみとなることや，全てのサービスで

路側機の共通化が可能となることから，効率的なサービス開発が可能になるという利点がある．またリソースを必要とする部分をサーバー側に集約させることで，低リソースな路側機を構築できるというメリットもある．一方，サービスアプリケーションの実行場所が路側機の設置場所と異なることや，全ての通信で，路側機－サーバー間の連携のオーバヘッドが発生することから，狭域性や初期接続性能・通信効率が一部犠牲となるという問題がある．

初期接続性能や狭域性は，DSRCローカル通信プラットフォームにおける最も重要な特徴であり，サーバー集約方式による複数路側機連携の実現に当たってはこれらの課題を解決することが必要となる．

4.2.2 プロトコルスタック

DSRCローカル通信プラットフォームでは，路側機上でアプリケーションが実行されることを前提としていることから，サーバー集約方式を実現するためには，プロトコルスタックを分割し，サービスアプリケーションを路側機外部で実行可能とする必要がある．サーバー集約方式のメリットを活かし，デメリットを最小限にとどめる方式として，LPCPに機器間連携のための機能を追加することにより，路側システム側のプロトコルスタックを，LPCP層で分離し，LPPより上位層をサーバーに，LPCPより下位層を各路側機にマッピングする方式を提案する．図4.3に提案する複数路側機連携DSRC通信プラットフォームの構成を示す．

本方式では，プッシュ型情報配信アプリケーションなどの基本的なアプリケーションプロトコルがサーバー側に配置されることから，サービスアプリケーションは，直接これらのアプリケーションプロトコルのインタフェース(以下，I/F)を利用する方法を採用する．この手法を使用することにより，サービスアプリケーションが路側機上で実行される従来のDSRC応用システムと同様の方法で，サービスを構築することができる．さらに，リソースを必要とするLPPやサービスアプリケーションがサーバー上に集約可能となるため，低リソースな路側機が構築可能というメリットもある．

一方，LPCPより下位層を路側機上に配置することで初期接続性能や通信効率に関する問題の改善を図っている．具体的には，DSRCローカル通信プラットフォームにおける通信の高速化には，ASLの通信制御プロトコル部におけるバルク転送

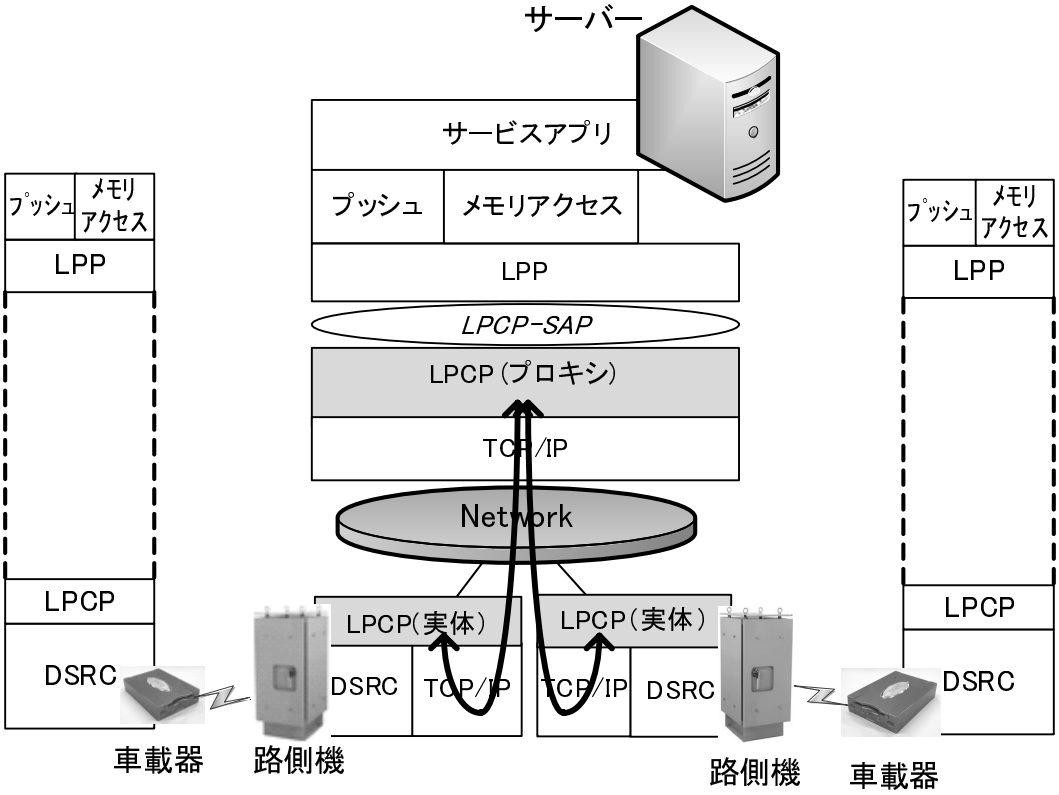


図 4.3: 複数路側機連携 DSRC 通信プラットフォームの構成

機能やサーバー・クライアント制御の最適化が必要となるが，通信制御プロトコル部が路側機上に配置されていることにより，既存の路側システムと同等の通信効率を得ることができる．また，LPCP の機能である受信可能ポートの管理機能を路側機上に配置することにより，DSRC ローカル通信プラットフォームにおける唯一の初期接続手順であるポート交換の機能を路側機－車載器間で行い，初期接続に掛かるオーバヘッドを最低限に抑えることが可能となる．また，路側機においてポート番号によるルーティングを行うことで，路側機上にサービスアプリケーションを配置する従来のシステムとの共存も可能なため，これらの遅延時間が許容できない，よりリアルタイム性が必要なサービスについては路側機上で実現し，それ以外のサービスはサーバー側で実現するといった対応も可能である．

さらに，大容量データ送信時の通信効率の劣化への影響に関しては，LPP の分割・組立機能を利用することでその影響を小さくすることができる．これは LPP の分割・組立機能では，路車間でのフロー制御が不要なため，図 4.4 に示すように，

2 番目以降の分割データの，サーバー—路側機間の通信遅延の影響が，パイプライン効果により隠蔽されるためである．例えば，第 3 章で提案したプッシュ型情報配信アプリケーションや車載器メモリアクセスアプリケーション [67] と呼ばれるアップリンク用のローカルアプリケーションにおける従来方式に対する通信可能時間の減少は，それぞれ 1 および 2 往復分にとどまる．

以上のように，LPCP 層でプロトコルスタックを分離するこのアーキテクチャは，サーバー集約方式によるメリットを生かしつつ，デメリットである初期接続性能や通信効率の劣化を最小限に抑えることが可能な方式であるといえる．

4.3 複数路側機連携を実現する追加機能

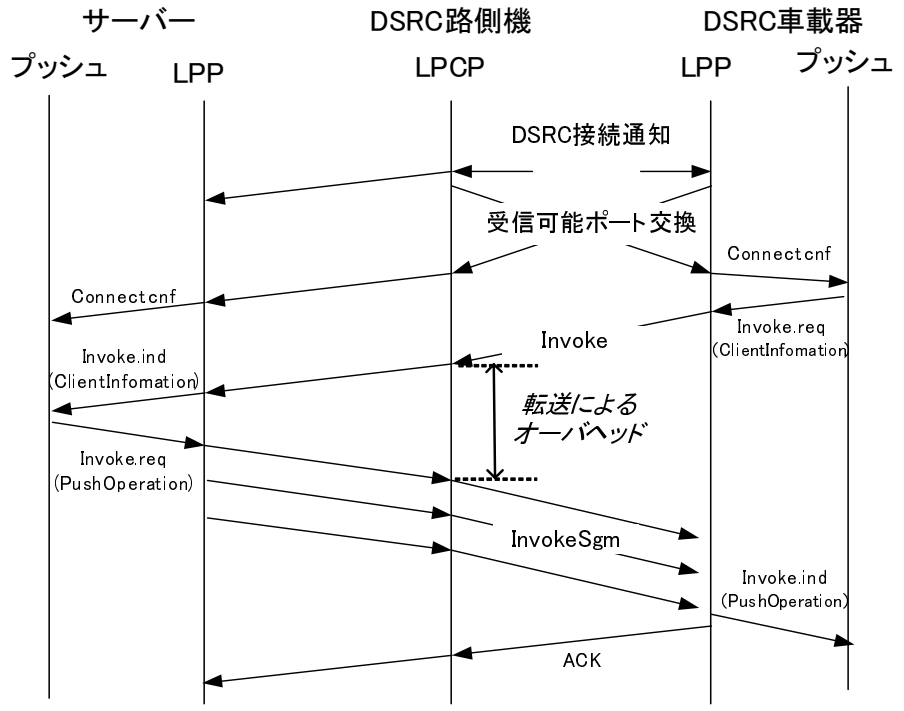
本方式では，4.2 で述べたアーキテクチャを実現するため，LPCP のパケットを IP でトンネリングすることにより，路側システムにおいて分離型構成を実現する．このトンネルは，各路側機を単位として複数開設され，サーバー上で各トンネルの開設状態や，車載器がどの路側機下に存在するかを管理することによってサーバーからの複数路側機を用いた個別・同報通信を実現する．

前節で述べたアーキテクチャに従い，初期接続時のローカルポート番号の交換など LPCP の通信機能は路側機側に配置し，サーバー側には，そのプロキシーとして，LPCP が提供する各プリミティブを LPP に対して提供するインタフェース機能のみを配置する．

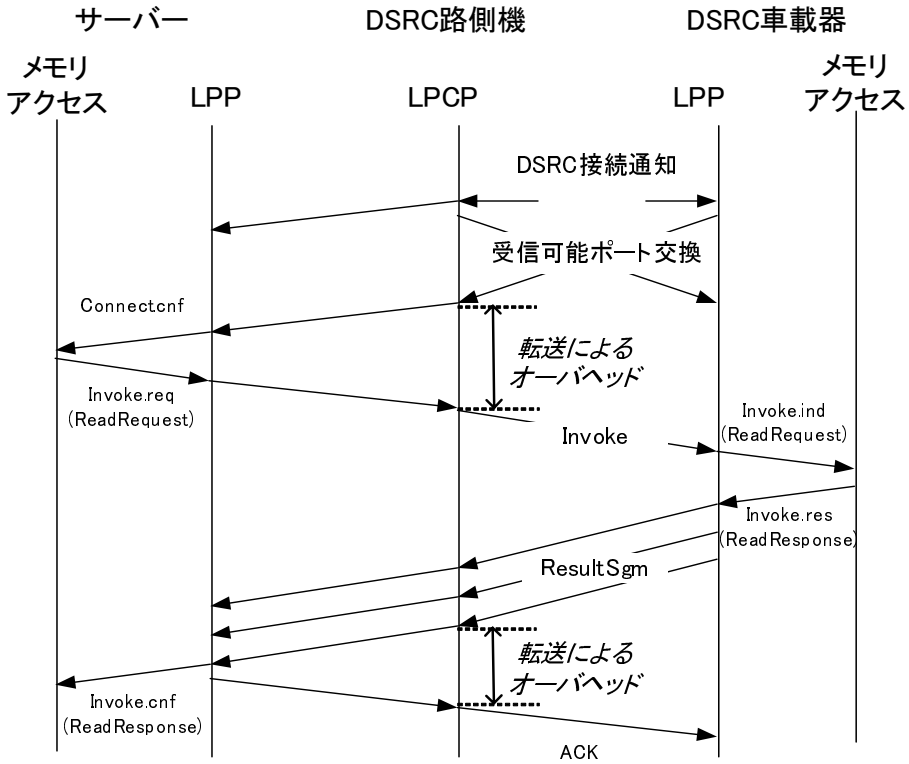
一方，サーバー側のサービスアプリケーションが複数路側機を用いて通信を行うため，サーバー側には

(1) 複数路側機の接続管理機能
(2) 複数の路側機を使用した通信機能
    - 路側機を指定した同報配信
    - 車両を指定した個別配信

の各機能が新たに必要となる．以下，本節では，これら複数路側機連携を実現するための各機能について詳述する．



(a) プッシュ型情報配信アプリケーションにおける通信シーケンス



(b) メモリアクセスアプリケーションにおける通信シーケンス

図 4.4: 複数路側機連携 DSRC 通信プラットフォームでの通信シーケンス

4.3.1 路側機の接続管理機能

サーバー－路側機間のトンネリングに必要な通信制御情報には，路側機の IP アドレスや搭載するローカルアプリケーションのローカルポート番号（受信可能ポート番号）があり，これらの情報は初期化時にサービスアプリケーションから指定される．図 4.5 に路側機との接続時の動作の概要を示す．

サーバー側ではサービスアプリケーションから指定された IP アドレスを元に，路側機との間でトンネルの開設を行い，成功した場合，サービスアプリケーションに対して通信制御情報と関連付けた路側機 ID を通知する（接続が失敗した場合，失敗した事を通知）．以後，サービスアプリケーションは通知された路側機 ID を用いて，路側機を判別する．また，トンネルの開設に成功した場合，サービスアプリケーションは，生成したトンネルを利用して，受信可能ポート番号をサーバー側から路側機側に登録する．

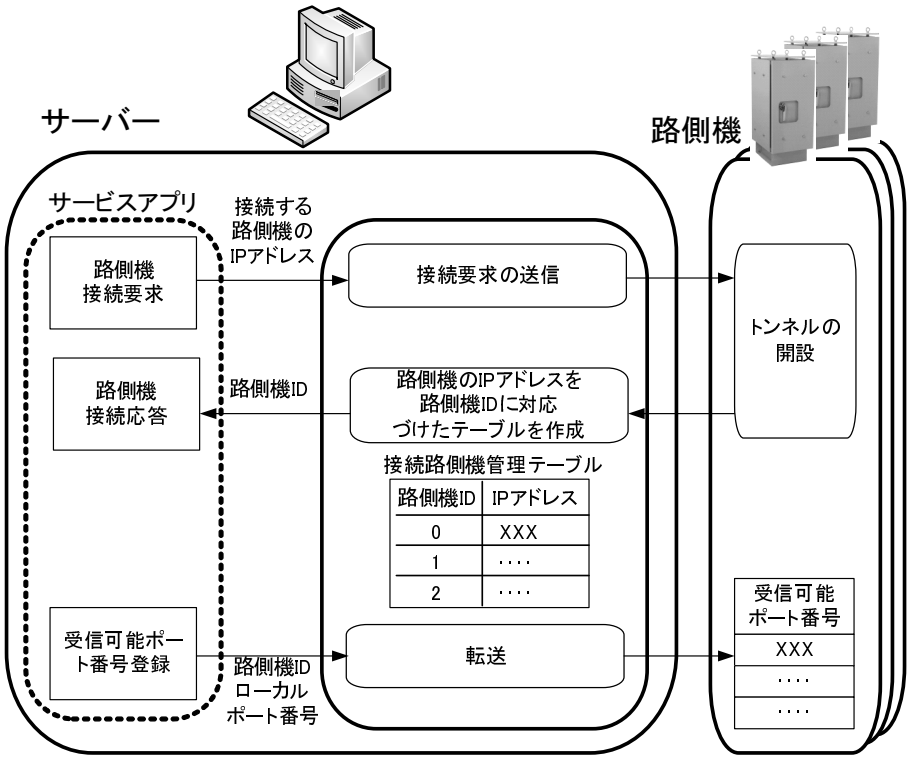


図 4.5: 複数路側機の接続管理機能

4.3.2　複数の路側機を用いた配信機能

DSRC ではその狭域性から，路側機の設置位置に応じたサービスの提供が求められる．提案する拡張方式においては，サービスアプリケーションの実行位置と路側機の設置位置が異なるため，サービスアプリケーションに対して車両がどの路側機下に存在するかを提供する仕組みや，サービスアプリケーションが路側機を指定して，配信を行う機能が必要である．一方，複数の路側機に跨って連続的な通信を行う必要のあるサービスでは，複数の断続的な通信エリアにおいて車両の同一性を確認するための機能が必要となる．これらの機能を実現するために，同報・個別通信として以下の機能を提供する．図 4.6 に本方式における情報配信時の動作の概要を示す．

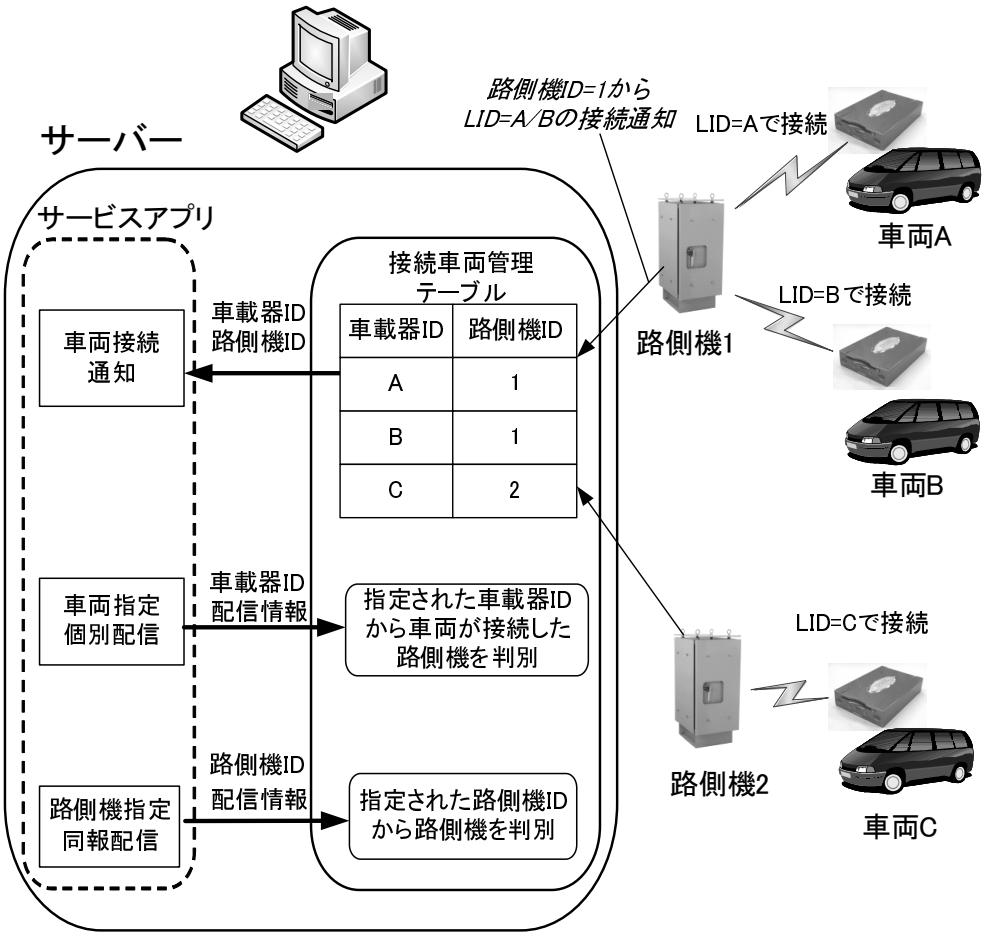


図 4.6: 複数の路側機を用いた情報配信機能

まず同報通信に関しては，路側機の接続処理時にアプリケーションに通知された

路側機 ID を宛先として配信先を指定することで，指定した路側機からの同報配信を実現する．これにより，路側機単位での同報通信が可能となる．

一方，個別通信に関しては，車両接続時に車両から通知される車載器 ID(ARIB STD-T75 における LID や ARIB STD-T88 における ASL-ID など) と路側機 ID を対応付けたテーブルを生成し，これらをアプリケーションに対して通知する．サービスアプリケーションは接続時に通知されたこれらの情報を使用することで，通信対象となる車両が存在する路側機を知ることが可能となるため，路側機の設置位置に依存したサービスを容易に実現可能となる．また個別通信の配信先の指定には車両の識別子である車載器 ID のみを用いる．サービスアプリケーションからの情報配信要求受付時に，上記車両の初期接続時に生成したテーブルを参照することで，車両が存在する路側機を自動的に判別し，その路側機に対してデータ配信要求を転送する．これにより，車載器 ID を指定するだけで，車両がどの路側機下にいても情報配信が可能となることから，複数の路側機に跨った連続的なサービスも容易に実現することができる．

4.4　実装と評価

4.4.1　通信遅延時間に関する評価

提案した複数路側機連携 DSRC 通信プラットフォームでは，LPP 層より上位層で行われる全ての通信に，サーバーと路側機間の通信遅延が付加される．本実験では，プッシュ型情報配信アプリケーションを用いて，それらの遅延時間を実際に測定することで提案したプラットフォームの有効性を確認する．

実際の機器に提案する機能を実装し，サーバーと路側機を 100[Mbps] の LAN で接続した構成で，図 4.4(a) における転送によるオーバヘッドの部分を計測した結果，サーバー－路側機間の通信遅延時間は平均で 5.5[msec] となった．第 2 章で示したとおり，LPP の初期接続時間は 50msec であることから，本プラットフォームでの初期接続時間は 55.5msec 程度となる．最も即時性が要求される安全運転支援サービスなどで要求される通信遅延時間が 100[msec] 程度であることを考慮すると，十分高速な初期接続性能を担保できているといえる．

また 4.2.2 で示したとおり，通信可能時間の減少が問題となる大容量データ通信

を行う場合のサーバー－路側機間の通信遅延による通信可能時間の減少はプッシュ型情報配信アプリケーションで 1 往復分 (5.5[msec])，車載器メモリアクセスアプリケーションで 2 往復分 (11[msec]) にとどまる．例えば高速走行中の車両に対するサービス条件として，20m の通信エリアを 100km/h で走行する場合を考えると，従来の路側機上でサービスアプリケーションが実行される形態での初期接続時間を除いた通信可能時間が 670[msec] であることから，従来方式と比較して，98%以上のサイズのデータが通信可能であり，高速走行中の車両に対するサービスにも適用可能であると考える．

4.4.2　車路車間通信システムによる評価

本プラットフォームを活用したサービス例として，車載器メモリアクセスアプリケーションとプッシュ型情報配信アプリケーションを利用した車路車間通信システムを示す．図 4.7 は実装したシステム構成と通信シーケンスである．

本システムでは，DSRC 車載器 A に接続された車両情報書込端末が 100[msec] 周期で車両情報を書き込んでおり，DSRC 車載器 A が路側機 A に接続すると，サーバー上のサービスアプリケーションがメモリ読出し要求コマンド (ReadRequest) を用いてこの車両情報を取得し，取得した車両情報を，即座に路側機 B に対してコンテンツ配信用コマンド (PushOperation) を用いて，同報配信する．DSRC 車載器 A が路側機 A と通信している間，このアップリンクとダウンリンクの処理を繰り返し行うことにより，車両 A から車両 B へのリアルタイムな情報交換を実現する．このように，本プラットフォームを利用することにより，サーバー側で車載器メモリアクセスアプリケーションとプッシュ型情報配信アプリケーションを単純に組み合わせるだけで，車路車間通信を実現できることが分かる．

本実験ではシステムの遅延時間を評価するため，図 4.7 における X 時刻と Z 時刻の差（Z-X 時間）を測定した．測定結果を図 4.8 に示す．なお本測定における試験条件は第 2 章の表 2.5 と同一とし，扱う車両情報のデータサイズは 80[byte] とした．図に示すとおり，平均で 45[msec]，最大でも 70[msec] の遅延時間となっている．最も即時性が要求される安全運転支援サービスなどで要求される通信遅延時間が 100[msec] 程度であることを考慮すると，十分なリアルタイム性能が得られている．

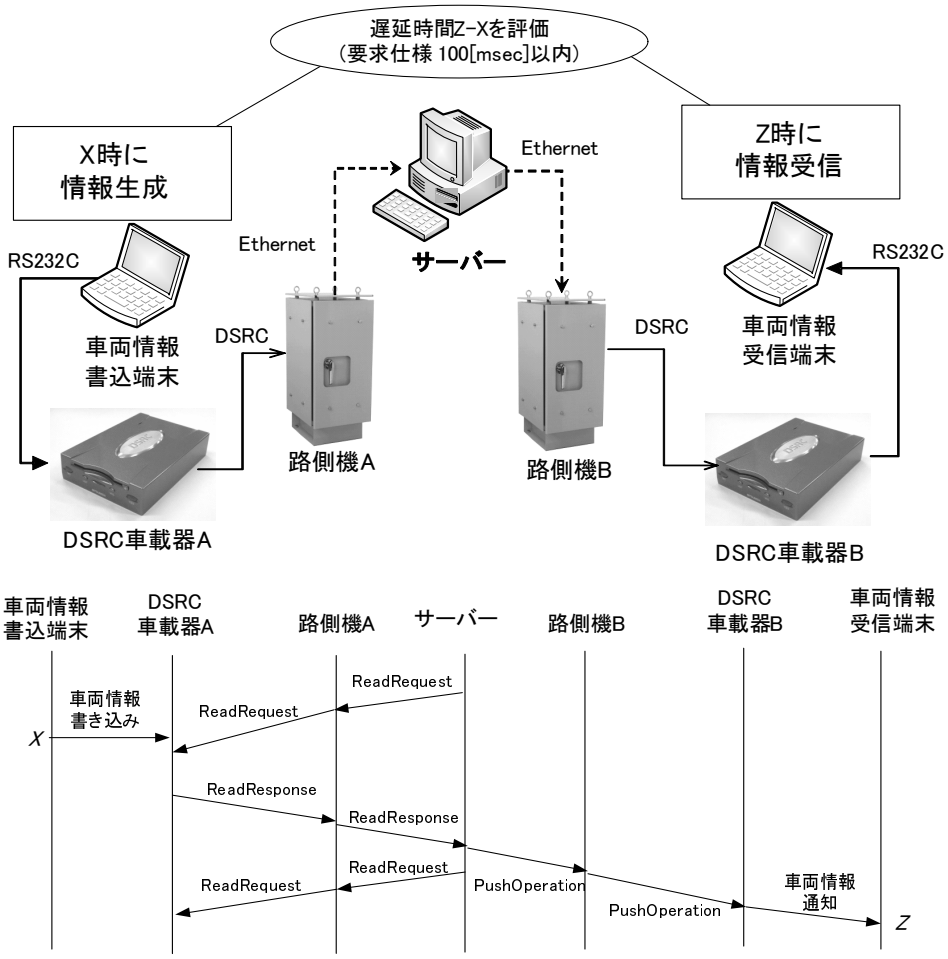


図 4.7: 複数路側機連携DSRC通信プラットフォームを用いた車路車間通信システム

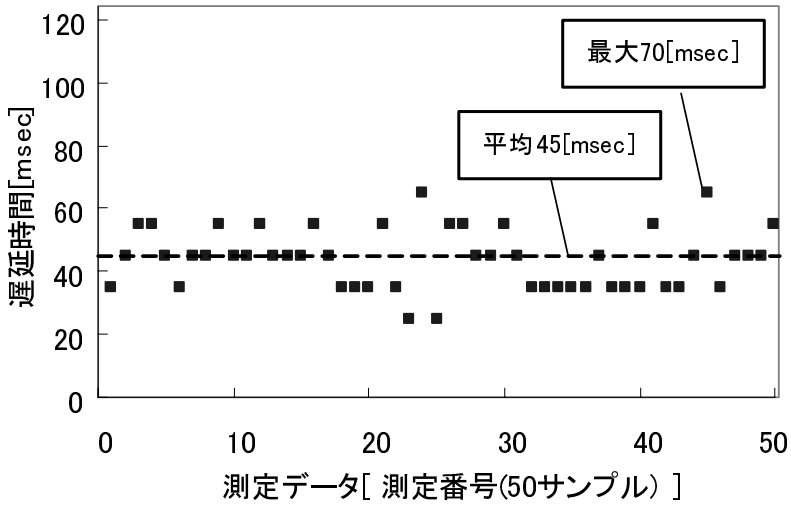


図 4.8: 車路車間通信の通信遅延時間

4.5　結言

本章では，単一の DSRC 路側機を用いたスポット型サービス向けの通信プラットフォームである DSRC ローカル通信プラットフォームを，複数の DSRC 路側機を利用した空間的な広がりのあるサービスに適用するための拡張方式について提案した．

提案する複数路側機連携 DSRC 通信プラットフォームでは，LPCP 層で DSRC ローカル通信プラットフォームのプロトコルスタックを分離し，LPP 層より上位層を路側機の外部サーバーへの配置を可能とすることにより，車載器側には一切の変更を加えることなく，サーバー集約型での複数路側機連携サービスを実現する．本プラットフォームを用いることで，サーバー上で単純に基本アプリケーションを組み合わせるだけで，複数路側機の連携サービスの構築が可能なため，サービス構築の効率化が期待できる．

また，提案した複数路側機連携 DSRC 通信プラットフォームを実際の機器に実装し，サーバー—路側機間の転送遅延時間を測定することで，高速な初期接続性能や高効率な通信という DSRC ローカル通信プラットフォームの特徴が継承されていることを確認すると共に，複数路側機連携 DSRC 通信プラットフォームを利用したサービスアプリケーションの例として，車路車間通信による車両間の情報交換サービスへの適用例を示し，その有用性について示した．

第5章　異種分散シミュレーションによるITS開発環境

5.1　序言

通信プラットフォームを利用して構築される路車協調システムのなかでも，安全運転支援システムや交通情報提供システムは，広域の社会公共インフラシステムであるため，システムの導入に当たっては，入念な設計/評価によりシステムの有効性を実証し，社会的なコンセンサスを前もって得ることが重要である．

しかしながら，これらのシステムはシステムの規模が大きくかつマンインループシステムであることなどの理由により，実フィールドでの試験によるシステム設計/評価を行うことは，コスト面ならびに安全性の問題があり，極めて困難である．これらの問題に対しては，多様な条件下で繰り返し試験が可能なコンピュータシミュレーションが極めて有効な手段となりうることから，従来より道路・交通・車両のシステムを設計/評価することを目的として

- 道路交通シミュレーション（RTS : Road Traffic Simulation）: 交通流を再現
- 車両動態シミュレーション（VDS : Vehicle Dynamics Simulation）: 車両の実挙動を再現
- 道路環境シミュレーション（RSS : Road Scene Simulation）: 走行シーンを再現

などの様々なシミュレーションシステム [73–76] が開発されてきている．

一方，ITS は人・道路・車の一体的なシステムであることから，その設計/評価に使用するシミュレーションには，人・道路・車の全てが協調して動作する環境を生成する機能が求められるが，これら既存のシミュレーションでは，このような環境を生成できないという問題がある．例えば，道路交通シミュレーションでは，交通流の再現が目的であるため，車載機器などの Human Factor に基づく評価に利用することはできない．また，車両動態シミュレーションやその 3D-CG 応用であるドライビングシミュレータでは，周辺の交通や情報システムとしての道路施設の動作を模擬することはできないため，ドライバーの交通流に対する影響等を評価することができない．

このような人・道路・車の全てが協調して動作する環境を実現する方法として，

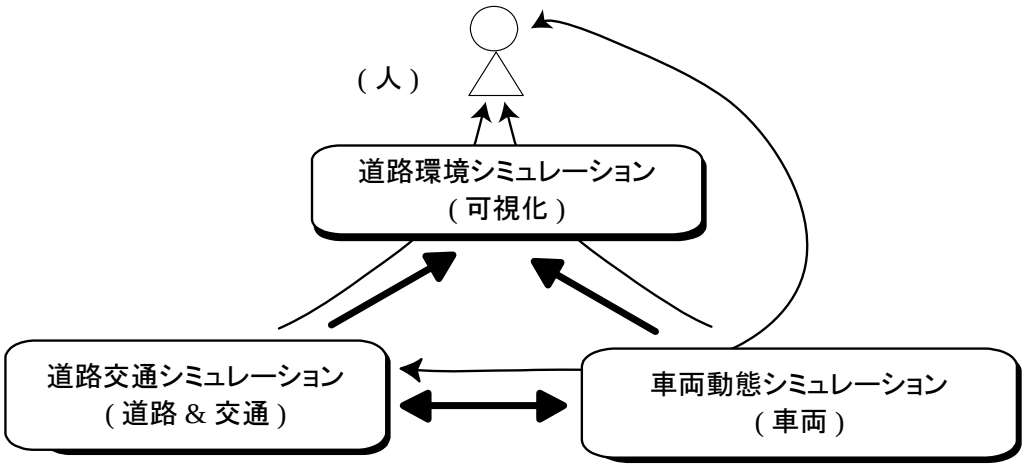


図 5.1: 各シミュレーションにおける人・道路・車両間の相互作用

前述した3つのシミュレーションシステムを1つのインタラクティブなリアルタイムシミュレーションシステムとして統合する方法がある．図5.1が示すように，統合されたシミュレーションでは，道路交通シミュレーション（道路&交通）と車両動態シミュレーション（車）のシミュレーション結果を道路環境シミュレーションが3D-CGを用いて可視化（人との間のI/F）することで人・道路・車の全てが協調して動作する環境を生成することが可能となる．ここでインタラクティブ性・リアルタイム性はシミュレーション内に人間が介在することから必要となる重要な特質である．

一方，このような複数の異なるシミュレーションシステムをリアルタイムシミュレーションシステムとして統合する場合には，個々のシミュレーション間のデータ形式やシミュレーション周期の違いをどのように扱うかが課題となる．これは，これらのシミュレーションでは，その目的に応じた最適化がなされており，個々のシミュレーション間で，データ形式やシミュレーション周期が異なっていることに起因している．

本章では，この課題に対する解決策として，

- 3D Street Graphと呼ぶ道路ネットワークデータベースを用いたデータ変換機構
- 描画周期に基づくシミュレーション間でのデータ交換処理とDead Reckoningによる車両位置の補間処理を用いたデータ同期機構

の 2 つの機構を有する実行制御部を導入し，異種分散シミュレーションとして統合化する手法を提案し，開発した統合シミュレータの路車協調システムの開発プラットフォームとしての活用例について述べる [51–54]．以下本論文では，この統合シミュレータを ITS シミュレータと呼ぶ．

5.2 ITS シミュレータの構成

前節で述べたように，複数の異なるシミュレーションシステムの統合を行う場合，個々のシミュレーションシステムが用いるデータ形式やシミュレーション周期の違いをどのように扱うかが課題である．例えば，文献 [77] では，それぞれのシミュレーションを独立したモジュールとして扱い，共通の通信プロトコルを用いて，それぞれの機能モジュールを組み合わせることにより統合を実現している．データ形式とシミュレーション周期の違いに関する問題については，道路交通シミュレーションが利用する道路ネットワークの幾何構造に 3 次元モデルを用いたり，シミュレーション周期をドライビングシミュレーションの描画周期と統一することにより解決を図っている．この手法は，個々のモジュールが基本的には独立していることにより，各モジュールの変更や追加などが容易で，システムの変更容易性・拡張性に優れている．一方で，道路モデルに 3 次元のモデルを使用することが前提であり，それ以外のモデルを使用する道路交通シミュレーションとの統合には適用できない．

また，文献 [78, 79] では，道路ネットワークをリンクとノードで抽象化する道路モデルを用いる既存の道路交通シミュレーションとドライビングシミュレータを組み合わせる手法が示されている．この統合シミュレーションでは，データ形式が異なる問題を，車両データの受け渡し時に座標変換処理を行うことで，解決している．一方，シミュレーション周期については，道路交通シミュレーションをドライビングシミュレーションに組み込むことで，2 つのシミュレーションの周期を統一し，同期処理を実現している．この方式は，比較的容易に統合が実現できる反面，同一の計算機上で全てのシミュレーションが実行されるため，文献 [77] の方式に比べ，拡張性が低い．

これらに対して，本研究で提案する手法では，統合対象の道路交通シミュレーションとして，道路ネットワークをリンクとノードで抽象化する道路モデルを使

用する既存のシミュレータを使用しつつ，統合においては，各シミュレーションのデータ形式やシミュレーション周期などは変更することなくそのまま使用することとし，それぞれのシミュレーション間の調停を行う実行制御部を導入することで異種分散環境として統合を実現する．つまり，図 5.2 に示すように，統合に当たって拡張が必要となるデータ交換などの処理は，実行制御部として各シミュレーションのコアとは独立したファンクションやプロセスにより実現する．

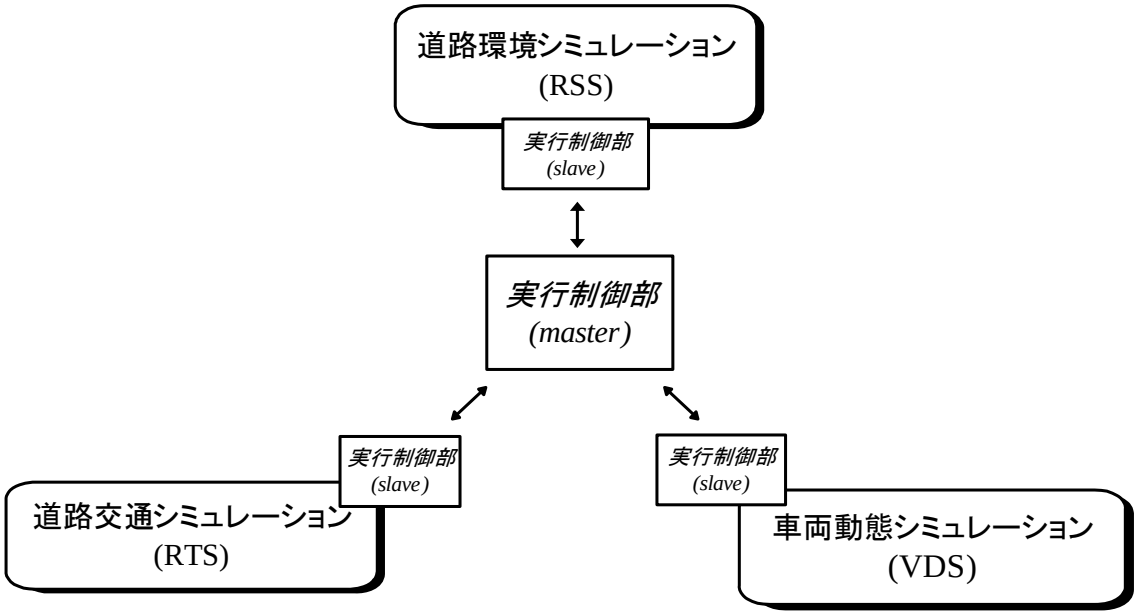


図 5.2: 提案する ITS シミュレータの構成

このような方法をとることには，

- 既存システムをほぼそのまま流用することが可能であり，開発期間・コストが短縮できる
- 道路交通シミュレーションとして道路ネットワークをリンクとノードで抽象化する道路モデルを使用し，かつシミュレーション周期として長めの値を使用することで，比較的広域のシミュレーションが可能である
- それぞれのシミュレーションを独立したモジュールとして扱うため，各シミュレーションの個別のバージョンアップや別のシミュレーションの追加などが容易で，システム全体としてのスケーラビリティに優れている

といったメリットがある．また本方式では，各シミュレーションはそれぞれ独立に

表 5.1: 各シミュレーションの諸元

	RTS	RSS	VDS
H/W	PentiumIII 450MHz	OnyxII	PentiumIII 450MHz
OS	Linux	IRIX6.0	Windows NT
シミュレーション周期	100msec（固定）	33-50msec（可変）	1-3msec（可変）
データ形式 (座標系)	Node & Link	3 次元座標系	3 次元座標系

動作し，同期点において車両の情報（位置・向きなど）を交換しあうことで，統合を実現する．このとき各シミュレーション毎にデータ形式やシミュレーション周期などが異なることから，実行制御部には

- 各シミュレーション間でやりとりするデータの同期をとるデータ同期機構
- 各シミュレーション間で異なるデータ形式を変換するデータ変換機構

の二つの機能が必要になる．次節以降では，統合における技術的側面であるこれらの手法について詳述する．なお，これらの統合処理（車両データの交換）は，統合シミュレーションにおける車両動態シミュレーションと道路環境シミュレーションの役割が，ドライバーに対して運転車両の挙動と視界を提供することであることを考慮すると，運転車両の周辺についてのみ行えばよい．

ここで本統合で利用した各シミュレーションの H/W・S/W 構成，シミュレーション周期およびデータ形式を表 5.1 に示しておく．ネットワークには 100Base-T のイーサネットと TCP/IP を利用している．

5.3 異種シミュレーションの統合化手法

5.3.1 データ変換機構

ITS シミュレータでは，各シミュレーションが独自のデータベースを保持している．これは個々のシミュレーションがそれぞれに最適化されたデータベースを保持することにより，高品位かつ高速なシミュレーションを可能とするためである．

また，ITS シミュレータでは，個々のシミュレーション間で車両データを交換す

る必要があるため，交換に当たってはそれぞれが認識できる形式に車両データを変換する必要がある．

本節ではこのデータ変換機構について，各シミュレーションにおける車両の表現形式を述べた後，本研究の提案手法である3D Street Graphを用いた座標変換手法について述べる．

5.3.1.1 各シミュレーションにおける車両の表現形式

ITSシミュレータを構成する3つのシミュレーションにおいては，それぞれデータ形式は異なるものの，同じ道路に関するデータベースが必要である．各シミュレーションで必要となるデータベースの形態は以下の通りである．

- 道路交通シミュレーション：ノードとリンクによって構成される2次元の道路ネットワーク
- 車両動態シミュレーション：勾配などの道路構造や路面摩擦等の路面状況
- 道路環境シミュレーション：3D-CG表示用道路・周辺地形データ

一方，各シミュレーションで共有すべきデータとしては，車両の位置と姿勢のデータがあるが，道路のデータベースがそれぞれ異なるため，これらの表現方法はそれぞれのシミュレーション毎に異なる．

具体的には，車両動態シミュレーションと道路環境シミュレーションにおいて使用される座標系はともに3次元座標系であり，図5.3.（a）が示すように，車両の位置と姿勢はそれぞれ$\boldsymbol{L}=[x,y,z]$，$\boldsymbol{E}=[\phi,\theta,\psi]$によって表現される．

一方，道路交通シミュレーションでは，図5.3.（b）が示すように，車両の位置はリンク番号（n）と次ノードまでの距離（h），リンク上での横方向の位置（l）によって表現され，車両の姿勢はhl平面上でのh軸に対する回転角（s）によって表現される．

以下，本論文では前者の座標系をRSS/VDS座標系，後者の座標系をRTS座標系と呼ぶこととする．

5.3.1.2 3D Street Graphを用いた座標変換処理

ITSシミュレータでは，シミュレーションのリアルタイム性を確保するため，このような2種類の全く異なる座標系を持つ多数の車両データの高速な座標変換処理

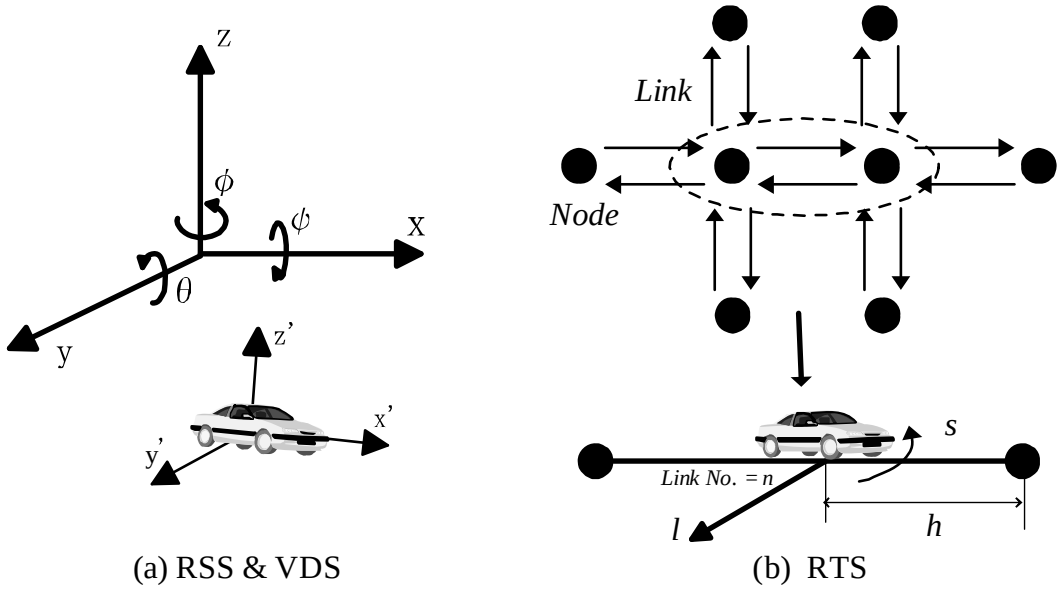


図 5.3: 各シミュレーションにおける座標系

が必要となる．このような要求を満たすため，ITS シミュレータでは，3D Street Graph と呼ばれる，3 次元情報を持った道路ネットワークデータベースを生成・利用することにより，高速な座標変換処理を実現する．

なお，以下では，車両位置についての変換処理のみを記述する．車両の姿勢についての処理も同様の手法により高速な変換が可能である．

3D Street Graph

3D Street Graph とは図 5.4 に示すように，複数の頂点オブジェクトの集合によって表現されるデータベースであり，頂点 P_k は RTS 座標系のデータとして次ノードまでの距離を表す h_{P_k}，RSS/VDS 座標系のデータとして 3 次元位置 $\boldsymbol{L_{P_k}} = [x_{P_k}, y_{P_k}, z_{P_k}]$ を保持している．

また，以下の 2 種類のベクトルデータをシミュレーションの実行前に作成，保持することで，実行時の計算量を減らし，座標変換処理の高速化を図っている．

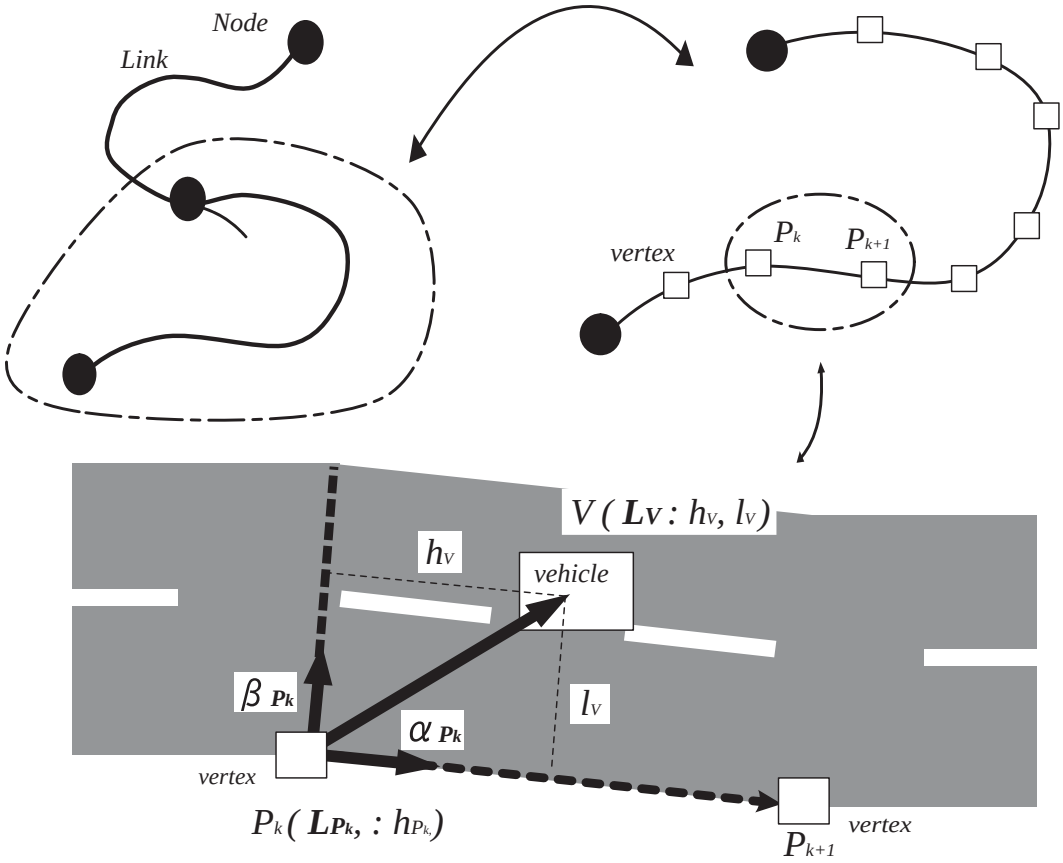


図 5.4: 3D Street Graph

$\boldsymbol{\alpha_{P_k}}$: 直線 P_kP_{k+1} 上を車両が $1m$ 移動したときの車両の 3 次元位置 $\boldsymbol{L_v}$ の変化量

$\boldsymbol{\beta_{P_k}}$: 路面を表す平面上で，直線 P_kP_{k+1} と直交する単位ベクトル

なお，図 5.4 中 $\boldsymbol{L_V} = [x_V, y_V, z_V]$ は RSS/VDS 座標系における車両の位置，h_V, l_V は RTS 座標系における車両の位置を表している．

図 5.4 が示すように，3D Street Graph では，道路環境シミュレーションにおいて道路平面を構成する各頂点のうち，車両の進行方向に対して，右端に位置する頂点データをつなげていくことでできあがる折れ線を道路交通シミュレーションにおけるリンクとみなすことで，RTS 座標系と RSS/VDS 座標系の対応をこれらの頂点において実現している．

車両データの座標変換処理

車両位置の座標変換処理は 3D Street Graph を基に以下のような手順で行われる．図 5.4 において，V を変換対象の車両，P_k, P_{k+1} を 3D Street Graph 内のある頂点とすると，RTS 座標系から RSS/VDS 座標系への変換処理は式 (5.1) により，RSS/VDS 座標系から RTS 座標系への変換処理は式 (5.2)(5.3) により求められる．

$$\boldsymbol{L_v} = \boldsymbol{L_{P_k}} + (h_{P_k} - h_V) \cdot \boldsymbol{\alpha_{P_k}} + l_V \cdot \boldsymbol{\beta_{P_k}}. \tag{5.1}$$

$$h_V = h_{P_k} - (\boldsymbol{L_V} - \boldsymbol{L_{P_k}}) \cdot \boldsymbol{\alpha_{P_k}}. \tag{5.2}$$

$$l_V = (\boldsymbol{L_V} - \boldsymbol{L_{P_k}}) \cdot \boldsymbol{\beta_{P_k}}. \tag{5.3}$$

ここで式 (5.2)(5.3) で用いられている記号 $\cdot$ はベクトルの内積を表す．

5.3.2 同期機構

ITS シミュレータはそれぞれが独立に動作する複数のシミュレーションで構成されるため，個々のシミュレーションの実行周期は表 5.1 が示すようにそれぞれ異なっている．従って，どのようにこれらの同期をとるかが重要である．ここでは，シミュレーションの同期をとる際のキーポイントである

(1) 各シミュレーション間でのデータ交換処理
(2) 同期ポイントである表示タイミングにあわせた Dead Reckoning による車両データの補間処理

について述べる．

5.3.2.1 シミュレーション間でのデータ交換処理

道路環境シミュレーションは，描画周期（T_{RSS}）毎に描画対象となる交通情報や運転車両位置など，他のシミュレーションの結果が必要である．また道路交通シミュレーションは，生成する交通流が運転車両の挙動を認識するために，シミュレーション周期（T_{RTS}）毎の運転車両データが必要である．以下，本節ではこれらのデータ交換について詳述する．

道路交通シミュレーションと車両動態シミュレーション間のデータ交換

図5.5を用いて，道路交通シミュレーションと車両動態シミュレーション間のデータ交換処理を実現する制御フローについて説明する．

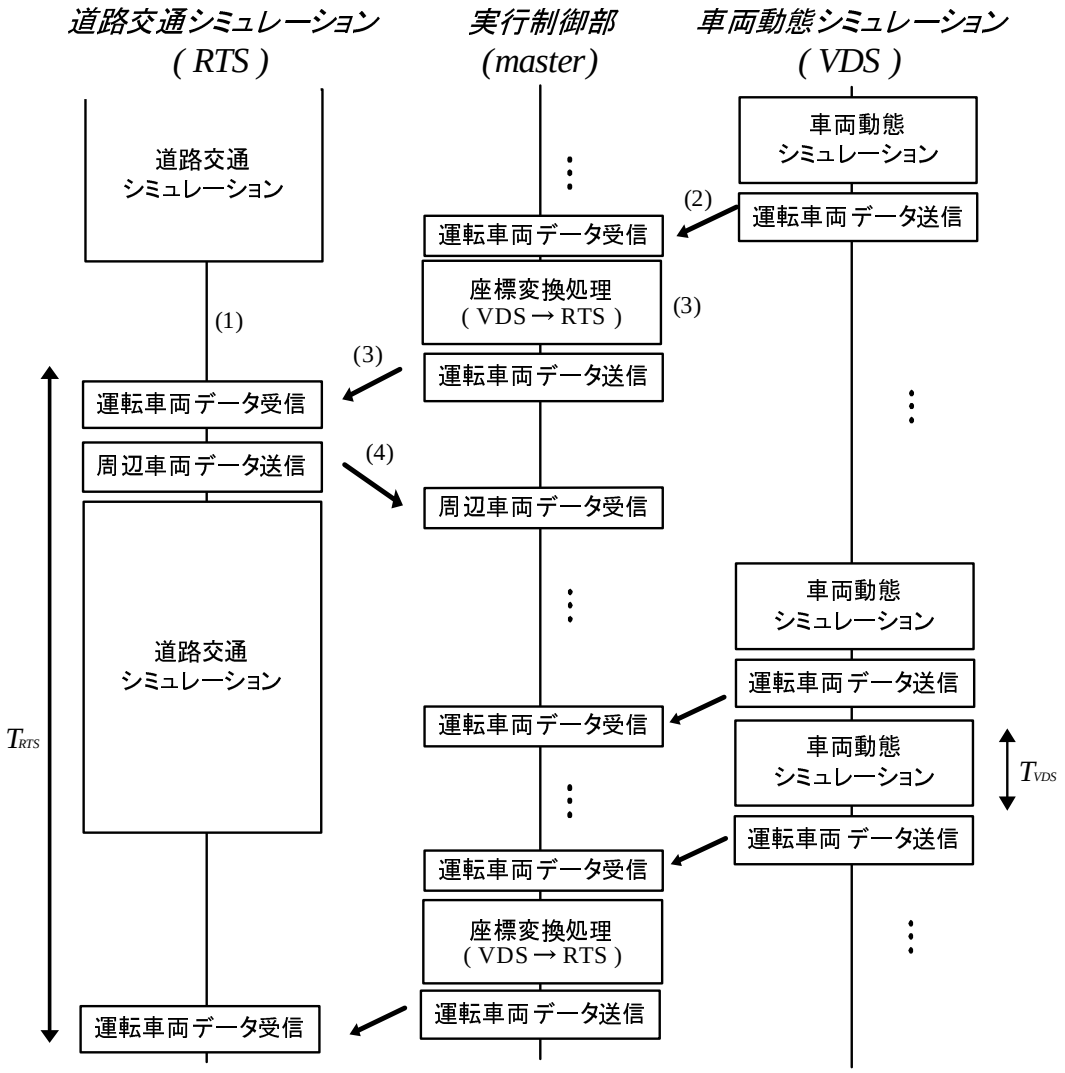


図 5.5: 道路交通シミュレーションと車両動態シミュレーション間のデータ交換処理

(1) 道路交通シミュレーション（RTS）は，実行制御部（$master$）から運転車両データが送られてくるまで，実行を中断する．

(2) 車両動態シミュレーション（VDS）は，逐次運転車両データを生成し，実行制御部に対して送信する．

(3) 実行制御部は前回，RTSに対して運転車両データを送信してから$T_{RTS}[sec]$

後のタイミングで，運転車両データを *RTS* に送信する．その際，車両データの座標系を RTS 座標系に変換する．

(4) *RTS* は運転車両位置を受信すると，シミュレーション結果を実行制御部に対し送信し，道路交通シミュレーションを再開する．

これらの処理により，道路交通シミュレーションは運転車両位置を認識できるようになり，図 5.6 に示すように，運転車両の挙動が道路交通シミュレーションの生成する交通流に影響を与えることが可能となる．

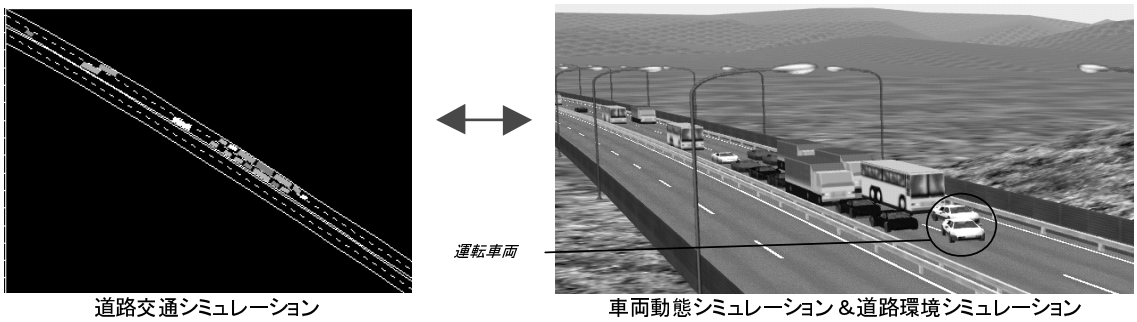


図 5.6: 運転車両と周辺交通間の相互作用

道路環境シミュレーションと車両動態/道路交通シミュレーション間のデータ交換処理

図 5.7 を用いて，道路環境シミュレーションに対して，車両動態/道路交通シミュレーションのシミュレーション結果を送信する制御フローについて説明する．なお，本節では，実行制御部と道路環境シミュレーション間の関係のみ記す．実行制御部と車両動態シミュレーションおよび道路交通シミュレーションの関係については，図 5.5 を参照されたい．

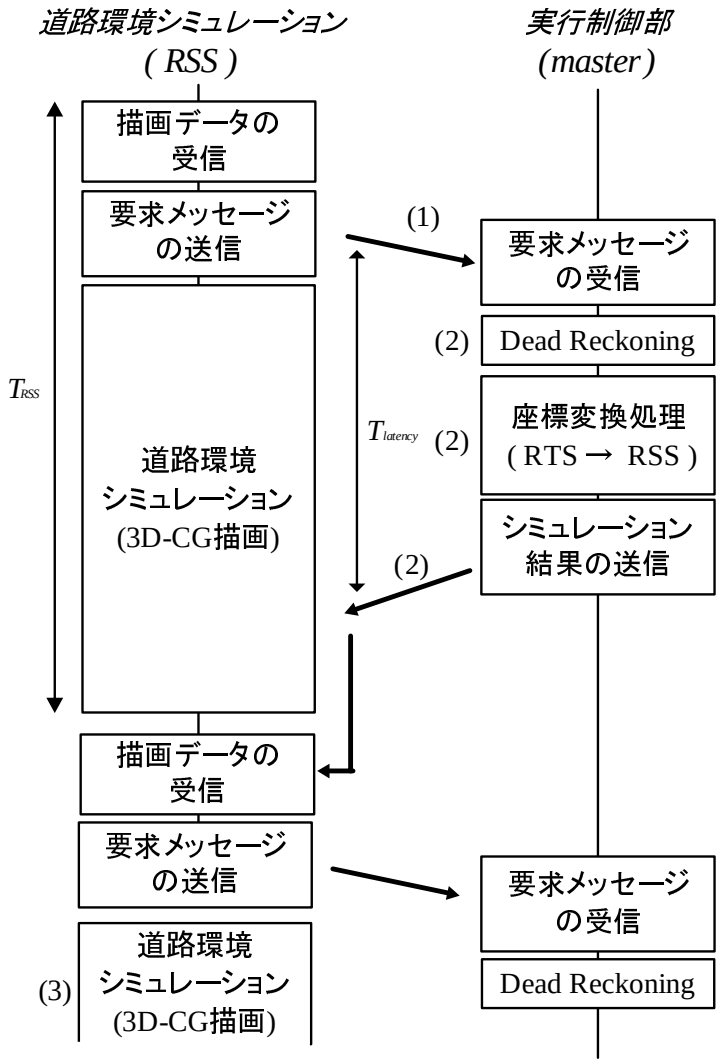


図 5.7: 道路環境シミュレーションと実行制部間のデータ交換処理

(1) 道路環境シミュレーション (RSS) が，実行制御部に対し，シミュレーション結果の要求メッセージを送信する．

(2) 要求メッセージを受信した実行制御部は次フレーム表示タイミングで必要となるシミュレーション結果を RSS に伝送する．シミュレーション周期の違いから，描画周期毎のシミュレーション結果は存在しないため，表示タイミングに合致したデータは5.3.2.2節にて後述するDead Reckoningによって生成する．

(3) RSS は受け取ったデータを次の表示タイミングで3D-CGとして表示する．

このようにシミュレーションデータの要求・受信処理の間に描画処理を行うことによって，要求・受信間の通信レイテンシを隠蔽し，道路環境シミュレーションでの描画速度の低下を防いでいる．

5.3.2.2 Dead Reckoning による車両位置の補正

道路交通シミュレーションのシミュレーション周期 T_{RTS} は，道路環境シミュレーションの描画周期 T_{RSS} よりも大きいため，描画回数よりもシミュレーション結果のほうが少なくなってしまう．そのため，周辺車両については，描画タイミングに合致したデータを道路交通シミュレーションの結果を基に作成する必要がある．そこで，道路交通シミュレーションは，道路環境シミュレーションからの要求メッセージの受信時に，1 つ前の要求メッセージ受信からの経過時間を次フレームの描画周期と仮定し，Dead Reckoning を行うことで，描画周期に合致した道路交通シミュレーション結果を作成する．

道路環境シミュレーションから j 回目の要求メッセージを受信した時の道路交通シミュレーションの車両位置情報を $(X(j), Y(j))$ の 2 次元座標で表すこととすると，Dead Reckoning は以下の式で定義される．

$$X(j) = X(j-1) + \frac{x(i) - x(i-1)}{T_{RTS}} \cdot \Delta t. \tag{5.4}$$

$$Y(j) = Y(j-1) + \frac{y(i) - y(i-1)}{T_{RTS}} \cdot \Delta t. \tag{5.5}$$

$$\Delta t = t_{req}(j) - t_{req}(j-1).$$

ここではシミュレーションの開始時刻を 0 とし，$t_{req}(j)$ は道路環境シミュレーションからの j 回目の要求メッセージを受信した時の時刻，$x(i), y(i)$ は i 回目の道路交通シミュレーションの結果を表している．なお，この i 回目のシミュレーション結果 $x(i), y(i)$ は，j 回目の要求メッセージを受信したタイミングでの最新の道路交通シミュレーションのシミュレーション結果である．

ここでは，2 つのシミュレーション結果間の車両の速度を一定と仮定しているが，Dead Reckoning を行う刻み幅が約 33-50[msec] と短いので，精度良く車両位置を補間できていると考える．

5.3.2.3 同期機構のリアルタイム性能

本節では，統合化処理がシミュレーションのリアルタイム性能に及ぼす影響を評価するために，前節で述べた道路環境シミュレーションと車両動態/道路交通シミュレーション間のデータ交換処理に要する時間について考察を行う．

このデータ交換処理においては，要求・受信間の通信レイテンシ $T_{latency}$ と道路環境シミュレーションのシミュレーション周期 T_{RSS} の関係が

$$T_{RSS} > T_{latency} \tag{5.6}$$

であれば，道路環境シミュレーションは描画速度を低下させずに実行できることになる．つまり，この $T_{latency}$ が小さいほど，より高速な描画速度を持つ道路環境シミュレーションとの統合が可能であるといえる．逆に $T_{latency}$ が T_{RSS} よりも大きい場合，道路環境シミュレーションのシミュレーション周期はディスプレイのリフレッシュレート $T_{refresh}$ の整数倍になるため，$T_{latency} \leq n \times T_{refresh}$（$n$ は自然数）となる最小の $n \times T_{refresh}$ が道路環境シミュレーションのシミュレーション周期となる．

このデータ交換に必要な処理時間は

（a）道路環境シミュレーション側での送受信処理時間
（b）データ伝送時間
（c）実行制御部での送受信処理時間
（d）Dead Reckoning 処理および道路交通シミュレーションデータの座標変換処理時間

である．これらのうち（a）の処理は，道路環境シミュレーション内の処理であることから，(b）～（d）の合計が $T_{latency}$ となる．

本シミュレータでは，運転車両位置を利用することにより，実行制御部を介して，道路交通シミュレーションと道路環境シミュレーション間でやりとりされる車両を限定し，送信データ量を削減することでデータ伝送時間の短縮を図っている．

また，Dead Reckoning 処理や座標変換処理および送信前処理を要求メッセージ受信前にあらかじめ済ませておくことで，(c），(d）の処理時間の短縮が可能である．これによりさらに通信レイテンシを短縮することが可能である．

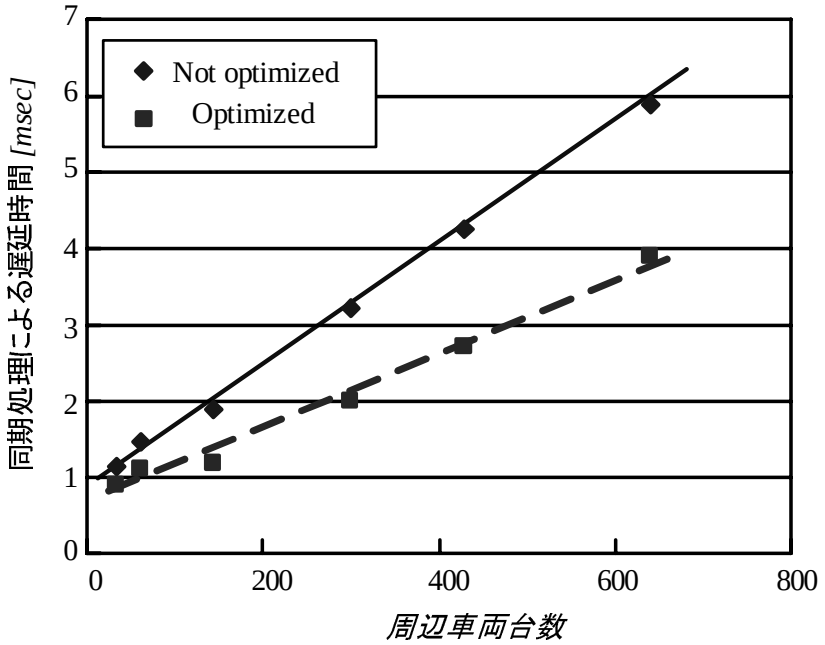


図 5.8: データ交換に起因する通信レイテンシ

図 5.8 は，表 5.1 に示した H/W 構成において，送信する車両数を変化させた場合の，$T_{latency}$ の変化を示す．実線は（c）,（d）を要する処理をあらかじめ前処理しない場合であり，破線は前処理を行った場合を示している．同図が示すように，道路交通シミュレーションと道路環境シミュレーション間でやりとりされる車両台数が 600 台を越えるような場合でも，$T_{latency}$ は $6msec$ 以下であり，送信する車両数と $T_{latency}$ の関係が比例関係にあることを考慮すると，道路環境シミュレーションの性能が許せば，やりとりされる車両数としては約3000 台程度まで，30 フレーム/秒以上の描画速度が達成可能である．

一方，各シミュレーション間でやりとりされる車両データの数は，運転車両の周辺に限定されることから，実際には多くても 1000 台程度しか車両がやりとりされることはなく，この 3000 台という値は，同期処理のリアルタイム性能として十分な性能を備えているといえる．なお，本研究で統合した道路交通シミュレーションでは，表 5.1 の H/W 及び

- シミュレーション周期：100msec
- 道路構造：高速道路・50km・上下線（車線数 2）

というシミュレーション条件で，約8000台の同時動作が可能で，比較的大規模な交通現象が再現可能である．さらに図5.8の破線の結果が示すように，通信レイテンシの短縮化前処理を行うことにより，3割程度の通信レイテンシの短縮が達成可能であり，さらに高速な描画速度をもつ道路環境シミュレーションへの対応も可能となる．

5.4 ITSモデル実装例

これまで述べてきたように，ITSシミュレータでは，道路交通シミュレーションが“道路施設”および“交通”の動作を生成し，車両動態シミュレーションが“運転車両”の動作を生成する．そして，それらのシミュレーション結果を道路環境シミュレーションが3D-CGを用いてビジュアライズすることで，人・車・道路間のインタラクションの存在するシミュレーション環境を実現している．従って，道路交通シミュレーションと車両動態シミュレーションに今後開発が進められる“スマートウェイ”や“スマートカー”の機能を追加することで，本シミュレータをITS開発プラットフォームとして活用することが可能となる．

本節ではこれらの拡張例として，本研究で実装を行ったシステム構成を示した後，次に，これらの拡張によって実現可能となったシミュレーション例としてAHS-iシステム（Advanced cruise-assist Highway Systems - *information*）[80]の一つである情報提供による障害物衝突防止システムへの適用例について述べる．

以下，本節では道路交通シミュレーションを中心としてスマートウェイとしての道路交通システムの動作を模擬する部分をスマートウェイエミュレーションモジュール，車両動態シミュレーションを中心として，スマートカーとしての運転車両をドライバーに提供する部分をスマートカーエミュレーションモジュールと呼ぶことにする．

5.4.1 全体構成

図5.9にITS開発プラットフォームの構成図を示す．核となる3つのシミュレーションの他に，スマートカーエミュレーションモジュールには，車載のHMIとしてのITS多機能端末と，複数のスマートカーモデルを，スマートウェイエミュレー

ションモジュールには複数のスマートウェイモデルを追加した．また拡張性を考慮し，追加した各モデルとシミュレーションとの間のデータのやりとりについては常に実行制御部を介して行うこととした．

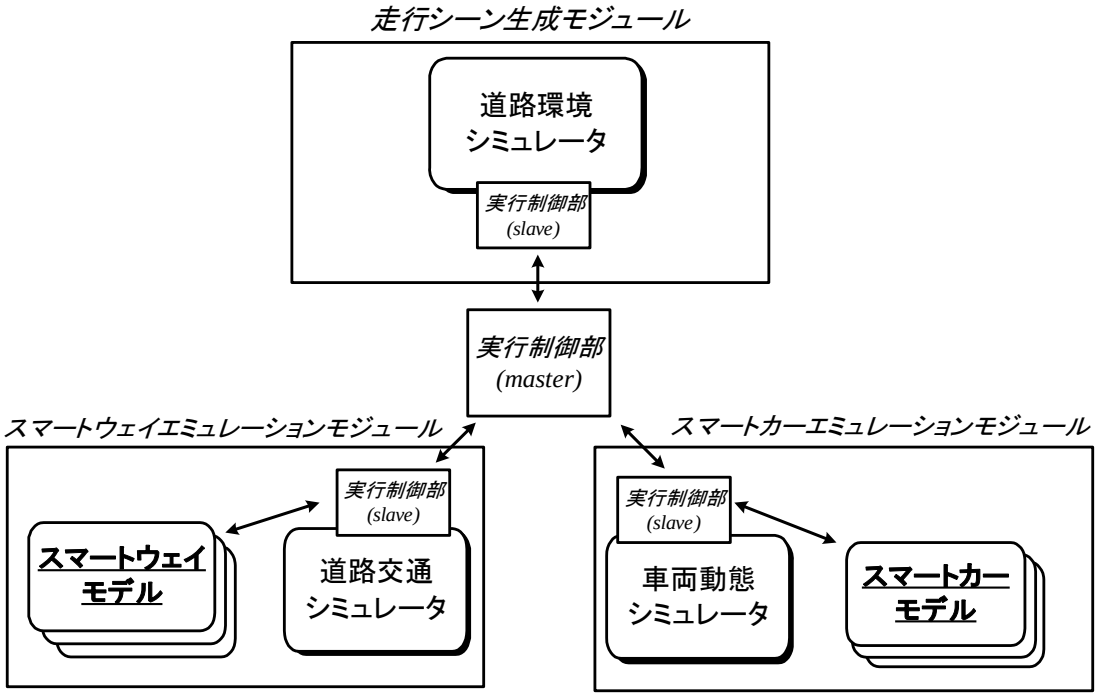


図 5.9: ITS シミュレータ上への ITS モデルの実装例

5.4.2 情報提供による障害物衝突防止システム

ここでは路車協調システムである AHS-i シミュレーションの 1 例として，情報提供による障害物衝突防止シミュレーションの実現例について述べる．

本シミュレーションを実現するために，スマートウェイエミュレーションモジュール側に障害物検知モデル，情報提供モデル，スマートカーエミュレーションモジュール側に情報提供モデルを実装した．

実装した障害物衝突防止シミュレーションの動作フローについて，図 5.10 を用いて説明する．図中の数字は以下に示す動作フローの項目番号である．

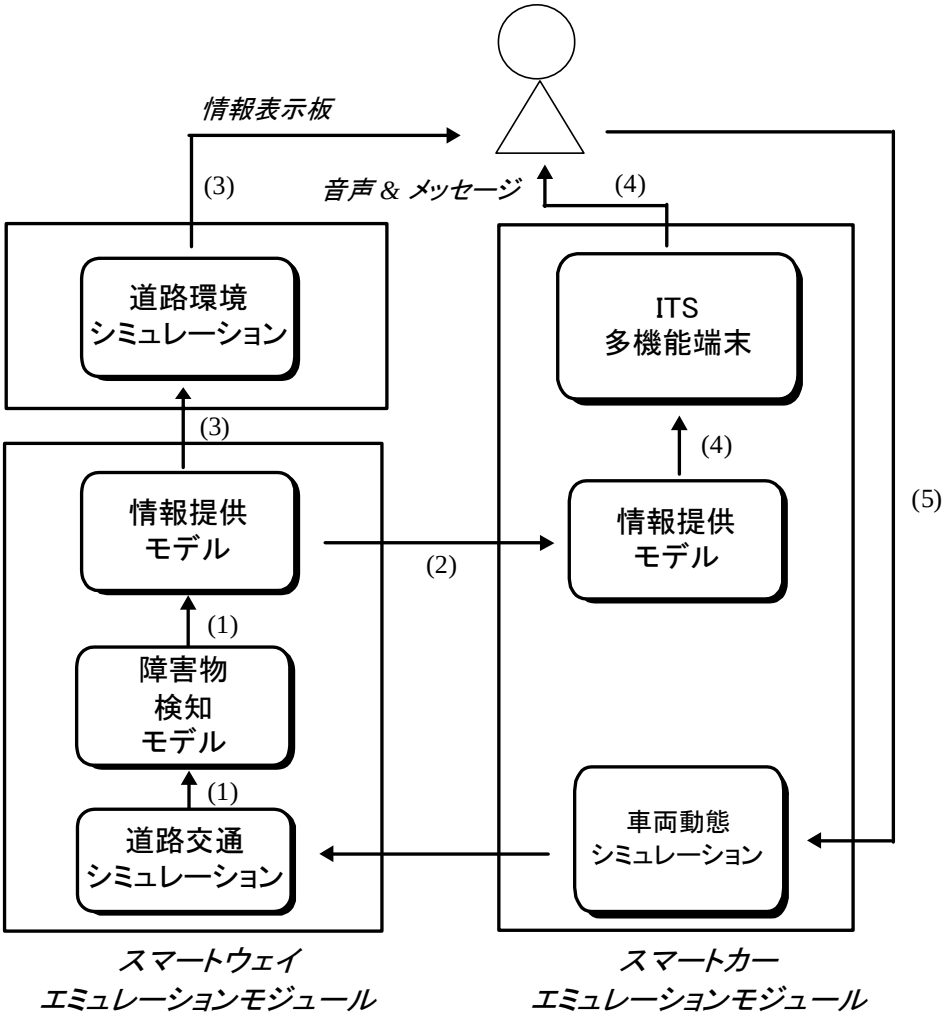


図 5.10: 障害物衝突防止シミュレーションの処理フロー

(1) 障害物検知モデルは，道路交通シミュレーションが生成する車両群から障害車両を検出し，情報提供モデルに通知する．

(2) 情報提供モデルは指定しておいた情報提供タイミングで障害車両位置をスマートカーエミュレーションモジュールに送信する．

(3) 同時に，道路環境シミュレーションを制御することにより，障害車両より手前にある可変表示板に障害車両が存在することを表示する．

(4) スマートカーエミュレーションモジュール内の情報提供モデルは，受信した障害車両位置を基に警告メッセージを作成し，ITS多機能端末を介して，ドライバに対し音声とディスプレイを用いて警告を発する．

(5) ドライバーは (3) と (4) の情報に基づいて，ハンドル・アクセル・ブレーキを操作し運転することで安全性を向上できる．

図 5.11 に情報提供による障害物衝突防止システムのシミュレーション例を示す．図 5.11 が示すように ITS シミュレータを用いることで，道路交通シミュレーションが生成する様々な交通状況下において，路側の情報板を用いた情報提供や，路車間通信を用いた情報提供システムの設計・評価を行うことが可能である．

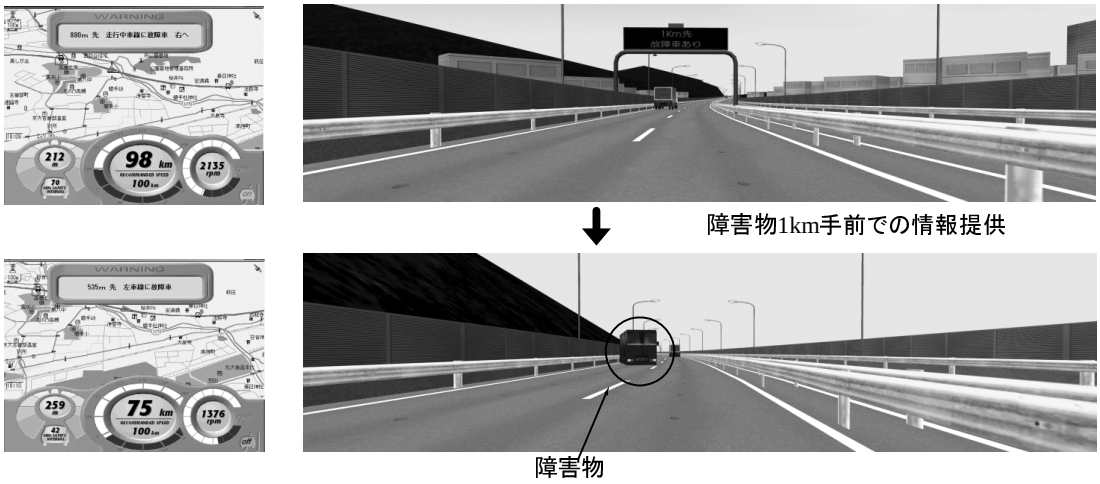


図 5.11: 情報提供による障害物衝突防止システムのシミュレーション例

これらのスマートウェイモデルやスマートカーモデルのプロトタイプ実装はあくまで一つの事例であり，本シミュレータは，別途開発が進展する様々な ITS サービスのためのインフラシステム及び車載システムの性能評価と機能改良のためのツールとして活用できる．

5.5　結言

本章では，各種路車協調システムの設計・評価・コンセンサス形成を可能とするシミュレーションシステム構築を目的として，道路・交通・車両のシミュレーションシステムの統合化手法を提案した．統合化においては，統合対象の道路交通シミュレーションとして，道路モデルをリンクとノードで抽象化する一般的なタイプを使用することとし，各シミュレーション間でデータ形式やシミュレーション周期が異なる問題を，描画タイミングに基づいたデータ交換をベースとした同期機構，Dead

Reckoning，3D Street Graphを用いた座標変換処理を導入することにより解決した．その結果リアルタイム性を保持したまま，全てのシミュレーションが協調動作することで，人・車・道路間のインタラクションが実現できていることを確認した．また，同期機構の性能評価を通じて，シミュレータのスケーラビリティについても示した．なお，本研究で行った実装において統合化により必要となったコードの変更量は，道路交通/車両動態シミュレーションについては，各シミュレーションにおける全コード量に対して，約5.0%程度であった．また，実行制御部を含めた統合化全体の変更量については，シミュレータ全体のコード量に対して，約23.8%程度であった．

さらに，このITSシミュレータの使用例として,「情報提供による障害物衝突防止システム」をスマートウェイモデルやスマートカーモデルとしてプロトタイプ実装することにより，本シミュレータを路車間の協調を必要とするような各種ITSサービスの開発プラットフォームとして使用可能なことを示した．

第6章　結論

本論文は，路車協調システムの開発効率化を実現する通信プラットフォームに関して筆者がこれまでに行ってきた研究成果をまとめたものである．以下に本論文の各章で得られた結論をまとめる．

(1) DSRC の多目的利用を実現する狭域通信環境向け高速通信プロトコルの提案

- 走行中車両/低リソース車載器への DSRC サービスを対象としたローカルアプリケーション向け通信アーキテクチャと非 IP 系高速通信プロトコルを提案した．
- 提案プロトコルは LPCP/LPP の 2 階層で構成され，LPP に必要な機能のみの選択的な実装を可能とすることで低リソース車載器への実装を実現した．
- 路側機/車載器ローカルでアプリケーションが実行されることを前提とし，これらのプロトコルをネットワーク層を持たないコネクションレス型の通信プロトコルとして設計することで，高速な初期接続性能や高効率な通信性能を必要最小限のリソース量で実現した．

(2) 車両へのプッシュ型情報配信サービスに適したアプリケーションプロトコルの提案

- 第 2 章で提案した通信プロトコル上に，走行車両に対する情報配信サービスを対象としたプッシュ型情報配信サービス向けのアプリケーションプロトコルを提案した．
- プッシュ型情報配信サービスを路車間通信部分であるプッシュプロトコルと再生アプリケーションに分離するとともに，プッシュプロトコルに車載システム側で操作レスでの自動再生が可能となる機能や，車載システムに応じた様々なコンテンツの配信を行う機能などを定義することで，様々な形態の車載システムに対する情報配信サービスを可能とし，実験によってその動作を確認した．
- 実機を用いた通信実験によって，その初期接続性能とスループットを計測することで，走行車両に対する情報配信サービスへの適用可能性

を検証するとともに，通信エリア滞在時間と通信可能データサイズの関係を明らかにし，事前の通信見積もりを可能とした．

(3) 複数路側機連携サービスのための DSRC 通信プラットフォーム拡張方式の提案

- 単一の DSRC 路側機を用いたスポット型サービス向けの DSRC ローカル通信プラットフォームを複数の DSRC 路側機を利用した空間的な広がりのあるサービスに適用するための拡張方式を提案した．
- サーバー—路側機間の転送遅延時間の測定の結果，約 55.5[msec] の初期接続性能，98%以上の通信可能時間を確保できることを確認し，高速走行中の車両へのサービスに適用可能という見通しを得た．
- サービスアプリケーション例として車路車間通信による車両間の情報交換サービスを実装し，その遅延時間を計測した結果，最大 70[msec] となることが分かり，実サービスへの適用可能性を示すことができた．

(4) 異種シミュレーションを分散統合した ITS 開発環境の構築

- 通信プラットフォームを利用して構築される路車協調システムの設計/評価に必須となるシミュレーション環境の構築方法として，モジュール毎の拡張や新たなモジュールの追加が容易な拡張性に優れた道路交通シミュレーションとドライビングシミュレーションの統合方式を提案した．
- 3D Street Graph を用いた高速な座標変換処理と，描画タイミングに基づいた Demand Driven 型データ共有機構の導入により，30[frame/sec] 以上の描画周期でのリアルタイム動作を実現した．
- 「情報提供による障害物衝突防止システム」をプロトタイプ実装し，様々な交通状況下でのシステムの設計・評価が可能なことを確認し，本シミュレータを路車協調システムの開発プラットフォームとして使用可能なことを示した．

本研究成果により，以下に述べる社会的貢献を行っている．まず，単一通信メディアの多目的利用の観点では，本研究で提案した非 IP 系の通信プロトコルは国内標

準化され，駐車場における料金決済サービスや道路交通情報提供や安全運転支援情報提供サービスへの適用を実現した．適用先はまだ限定的であり，今後，DSRC という新しい通信メディアの普及を促進するためにも，本プラットフォームを利用したサービスアプリケーションの研究開発が推進されるものと予想される．現在，筆者も，この観点から，本プラットフォームを交差点における安全運転支援サービスに適用するための研究開発を行っている [81].

一方，複数の通信メディアをアプリケーションで共有するという観点では，提案プロトコルの他メディアとの統合利用の可能性も高く，今後の応用が期待される．本研究では下位層として日本の DSRC である ARIB STD-T75 の利用を想定しているが，軽量・高速・高信頼といった提案プロトコルの特徴は，WAVE など他の狭域通信システムや車車間通信システムにも大変有用であり，これらの他の通信メディアに対する技術的な寄与も期待できる．

参考文献

[1] 財団法人 道路新産業開発機構, ITS HANDBOOK 2007-2008, 2007.

[2] ETC 総合情報ポータルサイト, http://www.go-etc.jp/ . (最終アクセス: 2009 年 12 月 18 日) .

[3] 財団法人 道路交通情報通信システムセンター (VICS センター), http://www.vics.or.jp/ . (最終アクセス: 2009 年 12 月 18 日) .

[4] Hiroshi Inoue, Susumu Osawa, Atsushi Yashiki and Hiroshi Makino, "Dedicated Short-Range Communications (DSRC) for AHS Services", *IEEE Intelligent Vehicles Symposium (IVS 04)*, pp.369-374, Jun. 2004.

[5] G. Vivo, P. Dalmasso and F. Vernacchia, "The European Project "SAFESPOT" - How ADAS applications co-operate for the driving safety", *IEEE Intelligent Transportation Systems Conference (ITST 2007)*, pp.624-629, Sep. 2007.

[6] Jack Opiola and Booz Allen Hamilton, "Vehicle Infrastructure Integration (VII) in the US - Enhancing Safety, Enabling Mobility", *The Institution of Engineering and Technology Seminor on RFID and Electronic Vehicle Identification in Road Transport, 2006*, pp.79-90, Nov. 2006.

[7] 藤田憲一, テレマティクス, 日経工業新聞社, 2002.

[8] ISO TC204 Working Group 16, CALM, http://www.isotc204wg16.org/ . (最終アクセス: 2009 年 12 月 18 日) .

[9] DARPA INTERNET PROGRAM, RFC791 - Internet Protocol, Sep. 1981.

[10] 上原啓介, 湧川隆次, 佐藤雅明, 渡辺恭人, 砂原秀樹, 寺岡文男, 村井　純, "自動車情報化のためのインターネットを用いた通信システムの構築", 情報処理学会論文誌, Vol. 42, No. 2, pp. 286–295, 2001 年 2 月.

[11] S. Deering and R. Hinden, RFC2460 - Internet Protocol, Version 6 (IPv6) Specification, Dec. 1998.

[12] V. Devarapalli, R. Wakikawa, A. Petrescu and P. Thubert, RFC3963 - Network Mobility (NEMO) Basic Support Protocol, Jan. 2005.

[13] S. Corson and J. Macker, RFC 2501 - Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, Jan. 1999.

[14] Thierry Ernst and Keisuke Uehara, "Connecting Automobiles To The Internet", *3rd International Workshop on ITS Telecommunications*, Nov. 2003.

[15] Ryuji Wakikawa, Koshiro Mitsuya, Keisuke Uehara, Thierry Ernst and Jun Murai, "Basic Network Mobility Support for Internet ITS", 情報処理学会論文誌, Vol. 44, No. 12, pp.2925-2935, Dec. 2003.

[16] Thierry Ernst and Arnaud De La Fortelle, "Car-To-Car and Car-To-Infrastructure Communication System Based on NEMO and MANET in IPv6", *The 13th World Congress on Intelligent Transport Systems*, Oct. 2006.

[17] インターネット ITS 協議会, http://www.internetits.org/. (最終アクセス: 2009 年 12 月 18 日) .

[18] CVIS : Cooperative Vehicle-Infrastructure Systems. Web page, http://www.cvisproject.org/. (最終アクセス: 2009 年 12 月 18 日) .

[19] 時津直樹，高橋邦彦，"インターネット ITS プロジェクト (実験編)", 情報処理, Vol. 43, No. 4, pp.376-385, 2002 年 4 月.

[20] Eric Koenders and Jaap Vreeswijk, "Cooperative Infrastructure", *IEEE Intelligent Vehicle Symposium (IVS.2008)*, pp. 721-726, Jun. 2008.

[21] 社団法人 電波産業会, "ARIB STD-T55 有料道路自動料金収受システム 標準規格 1.0 版", 1997 年 11 月.

[22] 社団法人 電波産業会, "ARIB STD-T75 狭域通信（DSRC）システム 標準規格 1.5 版", 2008 年 12 月.

[23] Tatsuji Munaka, Yoshihiko Ito and Soichi Kubota, "A Study of Info-communication Technologies Required in ITS Networks", *Symposium on Applications and the Internet Workshops*, Jan. 2001.

[24] 平岩賢志, 坂本敏幸, 森光正, 野明俊道, 西澤隆彦, "DSRC(ARIB STD-T75 準拠) システムの実装及び評価", 電子情報通信学会 A 論文誌, Vol. J86-A, No.12, pp.1382-1393, 2003 年 12 月.

[25] Chikara Yamaguchi, Masahiro Tsuda and Osamu Matsuda, "A DATA TRANSFER METHOD AND AN ADDRESS MANAGEMENT SYSTEM FOR TCP/IP OVER DSRC IN A PILOT PARKING LOT", *The 7th ITS World Congress*, 2000.

[26] CEN Central Secretariat, ENV12253 - Road Traffic and Transport Telematics (RTTI). Dedicated Short Range Communications (DSRC). Physical Layer using Microwave at 5.8GHz, Dec. 1997.

[27] CEN Central Secretariat, ENV12795 - Road Traffic and Transport Telematics (RTTI). Dedicated Short Range Communications (DSRC). Data Link Layer. Medium Access and Logical Link Control, Nov. 1997.

[28] CEN Central Secretariat, ENV12834 - Road Traffic and Transport Telematics (RTTI). Dedicated Short Range Communication (DSRC). Application Layer, Jan. 1998.

[29] American Society for Testing and Materials (ASTM) International, ASTM E2213-03 - Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems - 5GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2003.

[30] IEEE, IEEE Std 802.11a-1999: Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, High-speed Physical Layer in the 5 GHz Band, 1999.

[31] IEEE, IEEE Std 802.11p/D.4.02: Part 11: Wireless LAN Media Access Control (MAC) and Physical (PHY) specifications - Amendment 7: Wireless Access in Vehicular Environments", 2008.

[32] U.S. Department of Transportation, IEEE 1609 - Family of Standards for Wireless Access in Vehicular Environments (WAVE) , Jan. 2006.

[33] IEEE, IEEE 1609.2 - Trial Use Standard for Wireless Access in Vehicular Environments (WAVE) - Security Services for Applications and Management Messages, 2006.

[34] IEEE, IEEE 1609.3 - Trial Use Standard for Wireless Access in Vehicular Environments (WAVE) - Networking Services, 2007.

[35] Hamid Menouar, Massimiliano Lenardi and Fethi Filali, "On Mac and Routing Protocols Cooperation for Inter-Vehicle Communications", *5th International Conference on ITS Telecommunications (ITST 2005)*, Jun. 2005.

[36] Yi Wang, A. Ahmed, B. Krishnamachari and K. Psounis, "IEEE 802.11p Performance Evaluation and Protocol Enhancement", *IEEE International Conference on Vehicular Electronics and Safety (ICVES2008)*, pp.317-322, Sep. 2008.

[37] Masato Hayashi, Shoji Fukuzawa, Hiroki Ichikawa, Toshiyuki Kawato, Jun Yamada, Tsutomu Tsuboi, Susumu Matsui and Tetsuhiro Maruyama, "Development of Vehicular Communication (WAVE) System for Safety Applications", *7th International Conference on ITS Telecommunications, 2007*, pp.1-5, Jun. 2007.

[38] スティーヴンス, W. リチャード (著), 橘康雄 (訳), 井上尚司 (監訳), "詳解 TCP/IP ＜ Vol.1 ＞ プロトコル", ピアソン・エデュケーション, 2000/12/20 出版.

[39] 伊川雅彦, 後藤幸夫, 熊澤宏之, 小泉薫, 伊藤益夫, "DSRC ローカル通信アーキテクチャ", 信学総大, No. A17-7, pp.336, 2003 年 3 月.

[40] 後藤幸夫, 伊川雅彦, 熊澤宏之, 小泉薫, 瀧北守, “DSRC ローカル通信の機能設計と実装”, 信学総大, No. A17-8, pp.337, 2003 年 3 月.

[41] Hiroyuki Kumazawa, Masahiko Ikawa, Yukio Goto, Yoshiaki Tsuda, Mamoru Takikita and Takeo Iwata, “REALIZATION OF NON-IP MULTIPLE APPLICATIIONS ON DSRC - Application Sub-Layer for Non-IP Communication -”, *The 10th ITS World Congress*, 2003.

[42] 後藤幸夫, 伊川雅彦, 熊沢宏之, 津田喜秋, 横山謙悟, ”DSRC における Non-IP マルチアプリケーション規約とその応用”. 日本機械学会　第 12 回交通・物流部門大会, pp.63-64, 2003 年 12 月.

[43] 伊川雅彦, 後藤幸夫, 熊澤宏之, 津田喜秋, 岡賢一郎, “DSRC の多目的利用を実現する路車間通信の環境に適した通信プロトコルの設計と実装”, 電子情報通信学会 A 論文誌, Vol. J88-A, No. 2, pp.218-227, 2005 年 2 月.

[44] Masahiko Ikawa, Yukio Goto, Yuji Igarashi, Hiroyuki Kumazawa, Kaoru Koizumi and Kenichiro Oka, “DSRC Local Communication Platform and Its Application to Information Push Service”, *IEEE Intelligent Vehicles Symposium (IVS 04)*, pp.105-110, Jun. 2004.

[45] 五十嵐雄治, 伊川雅彦, 後藤幸夫, 熊沢宏之, ”次世代 DSRC 応用システムにおける情報配信技術：プッシュ型情報配信アプリケーションの応用”, 電子情報通信学会技術研究報告, Vol. 104, No. 506, pp.1-6, 2004 年 12 月.

[46] Yuji Igarashi, Masahiko Ikawa, Yoshitsugu Sawa, Yukio Goto, Hiroyuki Kumazawa and Yoshiaki Tsuda, “A PUSH TYPE INFORMATION DELIVERY TECHNOLOGY ON DSRC SYSTEM”, *The 12th World Congress on ITS*, No.472, Nov. 2005.

[47] 伊川雅彦, 五十嵐雄治, 後藤幸夫, 熊沢宏之, 津田喜秋, 森田茂樹, ”DSRC によるプッシュ型情報配信サービスのためのプロトコルの設計と実装”, 情報処理学会研究報告, ITS, Vol. 116, pp.115-120, 2007 年 11 月.

[48] 伊川雅彦，五十嵐雄治，後藤幸夫，熊澤宏之，津田喜秋，森田茂樹, “車両への情報配信サービスに適したプッシュ型プロトコルの設計と実装”，情報処理

学会論文誌，Vol.50, No.1, pp.42-50, 2009 年 1 月.

[49] 五十嵐雄治, 伊川雅彦, 後藤幸夫, 山本彰, 森田茂樹 “DSRC 路側機通信制御プラットフォームの開発”, 電気学会全国大会, pp.403, 2009 年 3 月.

[50] 伊川雅彦, 五十嵐雄治, 後藤幸夫, 小泉薫, 小牧省三, “複数路側機連携サービスのための DSRC プラットフォームの拡張”，信学技報, ITS2009-10, Vol. 109, No. 128, ITS2009-10, pp. 13-17, 2009 年 7 月.

[51] Masahiko Ikawa, Hiroyuki Kumazawa, Yukio Goto, Haruki Furusawa and Yukiyasu Akemi, ”Simulation Environment For ITS -Real Time 3D Simulator -”, *The 5th ITS World Congress*, No.3025, Oct. 1998.

[52] 伊川雅彦, 熊澤宏之, 後藤幸夫, 古澤春樹, “分散ネットワーク環境によるＩＴＳ評価シミュレーションの実現”, 電気学会 道路交通研究会, RTA-99-24, pp.49-54, 1999 年 9 月.

[53] Yukio Goto, Masahiko Ikawa and Hiroyuko Kumazawa, “A Prototype of Smartways In ITS Simulator”, *6th World Congress on ITS*, Oct. 1999.

[54] 伊川雅彦, 後藤幸夫, 熊澤宏之, 古澤春樹, “異種分散シミュレーションによる ITS 開発環境の構築”, 情報処理学会論文誌, Vol.45, No.12, p.2805-2814, 2004 年 12 月.

[55] ISO TC204, ISO 15628:2007, Road transport and traffic telematics –Dedicated short range communication (DSRC) – DSRC application layer, Jan. 2007.

[56] R. Braden, RFC955 - Towards a Transport Service for Transaction Processing Applications, Sep. 1985.

[57] Sun Microsystems, Inc., RFC1050 - RPC:Remote Procedure Call Protocol specification, Apr. 1988.

[58] 津田喜秋, 小泉薫, 伊川雅彦, 牧田覚, 西脇剛史, “駐車場 DSRC システムの開発”, 信学総大, No. A17-25, pp.376, 2004 年 3 月.

[59] Yoshiaki Tsuda, Kenji Togashi, Masahiko Ikawa, Satoru Makita and Takeshi Nishiwaki, "DEVELOPMENT OF DSRC PARKING SYSTEM", The 11th ITS World Congress, No.3084, Oct. 2004.

[60] 社団法人 電波産業会, "狭域通信 (DSRC) アプリケーションサブレイヤ 標準規格 ARIB STD-T88 1.1 版", 2007 年 12 月.

[61] Tatsuji Munaka, Tatsushi Yamamoto, Masahiro Kuroda, Tadanori Mizuno and Takashi Watanabe, "A Reliable Multicast Mechanism for Location Dependent Data in DSRC-Based ITS Networks", *IEICE TRANS. INF. SYST.*, Vol. E85-D, No. 11, pp.1809-1821, Nov. 2002.

[62] 石川憲洋, 木下真吾, 高橋修, "プッシュ型情報配信のためのプロトコルとそのコンテンツ配信システムへの適用", 情報処理学会論文誌, Vol. 41, No. 2, pp.245-253, 2000 年 2 月.

[63] 高橋修, 上野英俊, 石川憲洋, 角野宏光, 鈴木偉元, 水野忠則, "移動機向けプッシュプロトコルの提案と評価", 情報処理学会論文誌, Vol. 43, No. 10, pp.3107-3117, 2002 年 10 月.

[64] WAP Forum, "Push OTA Protocol Version 25-April-2001", 2001.

[65] 五十嵐雄治, 森田茂樹, 伊川雅彦, 荒木宏, 後藤幸夫, 熊沢宏之, 津田喜秋, "音声再生機能を搭載した高機能 DSRC 車載器の開発", 信学ソ大, A-17-20, pp.253, 2005 年 9 月.

[66] 澤良次, 伊川雅彦, 河野篤, 横山謙悟, 後藤幸夫, 熊沢宏之, "外部端末を利用した DSRC 応用システムの検討", 電気学会研究会 交通・電気鉄道研究会, pp23-28, 2005 年 11 月.

[67] ITS 情報通信システム推進会議, "狭域通信 (DSRC) 基本アプリケーションインタフェース仕様ガイドライン ITS FORUM RC-004 1.1 版", 2007 年 3 月.

[68] 電子情報技術産業協会, "ITS 車載器標準仕様 JEITA TT-6001A", 2008 年 3 月.

[69] 電子情報技術産業協会, "ITS 車載器 DSRC 部標準仕様 JEITA TT-6002A", 2008 年 3 月.

[70] 電子情報技術産業協会, “ITS 車載器カーナビ部標準仕様 JEITA TT-6003A”, 2008 年 3 月.

[71] 国土交通省国土技術政策総合研究所, “次世代道路サービス提供システムに関する共同研究報告書”, 2006 年 3 月, http://www.nilim.go.jp/japanese/its/1top/kyouken. (最終アクセス: 2009 年 12 月 18 日) .

[72] 荒木宏, 竹中憲郎, 森田茂樹, 小泉薫, 浜田誠也, 平井節生, “DSRC 車路車間通信システムの一検討”，電子情報通信学会ソサイエティ大会講演論文集, no. A-17-6, pp.179, 2006 年 9 月.

[73] Qi Yang and Haris N.Koutsopoulosg, “A Microscopic Traffic Simulator for Evaluation of Dynamic Traffic Management Systems”, *Transportation Research Part C*, Vol. 4, No. 3, pp.113–129, 1996.

[74] Yukio Goto, F. Corrado, Kiyotoshi Komaya, T. Fukuda and Haruki Furusawa, “A Microscopic Traffic Flow Simulator”, *2nd World Congress on ITS*, Vol. 4, pp.1905-1910, 1995.

[75] D. Smith and J. M.Starkey, “Effects of Model Complexity on the Performance of Automated Vehicle Steering Controllers: Model Development, Validation and Comparison”, *Vehicle System Dynamics*, Vol. 24, pp.163–181, 1995.

[76] Hiroyuki Kumazawa and Haruki Furusawa, “Application of multimedia technologies to highway supervisory and control system”, *30th International Symposium on Automotive Technology and Automation*, pp.227-234 , 1997.

[77] Alexis Champion, Rene Mandiau, Christophe Kolski, Alexandre Heidet and Andras Kemeny, “Traffic generation with the SCANeR©II simulator : towards a multi-agent architecture”, *Proceedings of Driving Simulation Conference '99, Paris, France*, pp.311-324, 1999.

[78] Majid Sarvi and Masao Kuwahara, “A STUDY ON FREEWAY RAMP MERGING PHENOMENA IN CONGESTED TRAFFIC SITUATION BY TRAFFIC SIMULATION COMBINED WITH DRIVING SIMULATOR, *8th World Congress on ITS*, 2001.

[79] 西川功, Majid Sarvi, 桑原雅夫, 森田綽之, “首都高速道路合流部における車両挙動に関する研究　ー交通流シミュレーションを組み込んだドライビングシミュレータの活用ー”, 第 20 回交通工学研究発表会講演論文集, 交通工学研究会, pp.45-48, 2000 年 11 月.

[80] G. Kamata, T. Kondo and H. Mashimo, “Design of a Cruise Assist System Based on Requirements”, *The 6th World Congress on ITS*, 1999.

[81] Masahiko Ikawa, Yukio Goto, Shigeki Morita, Akira Yamamoto, Nobuyoshi Asanuma and Yuriko Ino, ”A COOPERATIVE DRIVING SUPPORT SYSTEM BASED ON ITS ON-BOARD UNIT”, *The 15th World Congress on ITS*, Nov. 2008.

本論文に関する原著論文

A　学術論文

(1) 伊川雅彦, 後藤幸夫, 熊澤宏之, 古澤春樹, “異種分散シミュレーションによる ITS 開発環境の構築”, 情報処理学会論文誌, Vol.45, No.12, p.2805-2814, 2004 年 12 月.

(2) 伊川雅彦, 後藤幸夫, 熊澤宏之, 津田喜秋, 岡賢一郎, “DSRC の多目的利用を実現する路車間通信の環境に適した通信プロトコルの設計と実装”, 電子情報通信学会論文誌 (A), Vol.J88-A, No.2, pp.218-227, 2005 年 2 月.

(3) 伊川雅彦, 五十嵐雄治, 後藤幸夫, 熊澤宏之, 津田喜秋, 森田茂樹, ”車両への情報配信サービスに適したプッシュ型プロトコルの設計と実装”, 情報処理学会論文誌, Vol.50, No.1, pp.42-50, 2009 年 1 月.

B　国際会議（査読付き）

(1) Masahiko Ikawa, Hiroyuki Kumazawa, Yukio Goto, Haruki Furusawa and Yukiyasu Akemi, ”Simulation Environment For ITS -Real Time 3D Simulator -”, The 5th ITS World Congress, No.3025. Oct, 1998.

(2) Yukio Goto, Masahiko Ikawa and Hiroyuko Kumazawa,, “A Prototype of Smartways In ITS Simulator”,　The 6th World Congress on ITS, Oct. 1999.

(3) Hiroyuki Kumazawa, Masahiko Ikawa, Yukio Goto, Yoshiaki Tsuda, Mamoru Takikita, and Takeo Iwata, ”REALIZATION OF NON-IP MULTIPLE APPLICATIIONS ON DSRC - Application Sub-Layer for Non-IP Communication -”, The 10th ITS World Congress, Nov. 2003.

(4) Masahiko Ikawa, Yukio Goto, Yuji Igarashi, Hiroyuki Kumazawa, Kaoru Koizumi and Kenichiro Oka, ”DSRC Local Communication Platform and Its Application to Information Push Service”, IEEE Intelligent Vehicles Symposium, pp.105-110, June 2004.

(5) Yoshiaki Tsuda, Kenji Togashi, Masahiko Ikawa, Satoru Makita and Takeshi

Nishiwaki, ”DEVELOPMENT OF DSRC PARKING SYSTEM”, The 11th ITS World Congress, No.3084, Oct. 2004.

(6) Yuji Igarashi, Masahiko Ikawa, Yoshitsugu Sawa, Yukio Goto, Hiroyuki Kumazawa, and Yoshiaki Tsuda, ”A PUSH TYPE INFORMATION DELIVERY TECHNOLOGY ON DSRC SYSTEM”, The 12th ITS World Congress, No.472, Nov. 2005.

C　国内研究会

(1) 伊川雅彦, 後藤幸夫, 熊沢宏之, 古澤春樹, ”分散ネットワーク環境によるＩＴＳ評価シミュレーションの実現”, 電気学会 道路交通研究会, RTA-99-24, pp.49-54, 1999 年 9 月.

(2) 五十嵐雄治, 伊川雅彦, 後藤幸夫, 熊沢宏之, ”次世代 DSRC 応用システムにおける情報配信技術：プッシュ型情報配信アプリケーションの応用”, 信学技報, Vol. 104, No. 506, pp. 1-6, 2004 年 12 月.

(3) 澤良次, 伊川雅彦, 河野篤, 横山謙悟, 後藤幸夫, 熊沢宏之, ”外部端末を利用した DSRC 応用システムの検討”, 電気学会研究会 交通・電気鉄道研究会, pp.23-28, 2005 年 11 月.

(4) 伊川雅彦, 五十嵐雄治, 後藤幸夫, 熊沢宏之, 津田喜秋, 森田茂樹, ”DSRC によるプッシュ型情報配信サービスのためのプロトコルの設計と実装”, 情報処理学会研究報告, MBL, Vol. 116, pp.115-120, 2007 年 11 月.

(5) 伊川雅彦, 五十嵐雄治, 後藤幸夫, 小泉薫, 小牧省三, ”複数路側機連携サービスのための DSRC プラットフォームの拡張”, 信学技報, Vol. 109, No. 128, ITS2009-10, pp. 13-17, 2009 年 7 月.

D　全国大会等

(1) 伊川雅彦, 熊沢宏之, 後藤幸夫, 古澤春樹, ”ITS 評価のための 3D-CG リアルタイムシミュレータ”, 電気学会産業部門応用全国大会, No. 314, pp.415-416, 1998 年 8 月.

(2) 伊川雅彦, 後藤幸夫, 熊沢宏之, 津田喜秋, 岡賢一郎, ”DSRC ローカル通信

アーキテクチャ”, 信学総大, A-17-7, pp.336, 2003 年 3 月.

(3) 後藤幸夫, 伊川雅彦, 熊沢宏之, 小泉薫, 瀧北守, ”DSRC ローカル通信の機能設計と実装”, 信学総大, A-17-8, pp.337, 2003 年 3 月.

(4) 後藤幸夫, 伊川雅彦, 熊沢宏之, 津田喜秋, 横山謙悟, ”DSRC における Non-IP マルチアプリケーション規約とその応用”. 日本機械学会　第 12 回交通・物流部門大会, pp.63-64, 2003 年 12 月.

(5) 津田喜秋, 小泉薫, 伊川雅彦, 牧田覚, 西脇剛史, ”駐車場 DSRC システムの開発”, 信学総大, A-17-25, pp.376, 2004 年 3 月.

(6) 五十嵐雄治, 森田茂樹, 伊川雅彦, 荒木宏, 後藤幸夫, 熊沢宏之, 津田喜秋, ”音声再生機能を搭載した高機能 DSRC 車載器の開発”, 信学ソ大, A-17-20, pp.253, 2005 年 9 月.

(7) 伊川雅彦, 五十嵐雄治, 津田喜秋, 浜田誠也, 平井節生, ”ITS 車載器によるプローブシステムの実現”, 信学ソ大, A-17-14, pp.187, 2006 年 9 月.

(8) 五十嵐雄治, 伊川雅彦, 後藤幸夫, 山本彰, 森田茂樹,“DSRC 路側機通信制御プラットフォームの開発”, 電気学会全国大会, pp.403, 2009 年 3 月.