

2003年度(平成15年度)慶應義塾大学 政策・メディア研究科 修士論文

通信環境を考慮したインターネット自動車のための
アプリケーション開発環境に関する研究

慶應義塾大学大学院 政策・メディア研究科

日野哲志

zenigata@sfc.keio.ac.jp

修士論文要旨 2003年度(平成15年度)

通信環境を考慮したインターネット自動車のための アプリケーション開発環境に関する研究

-論文要旨-

本研究では、通信環境を考慮したインターネット自動車システムにおけるアプリケーション開発環境の構築を行なう。

インターネット自動車プロジェクトでは、自動車をインターネットに接続することにより、新しいアプリケーションやサービスを創造することを目的として活動している。自動車・道路・人を繋ぐネットワークを、インターネットを用いて構築することにより、自動車上であっても、インターネットの特徴である双方向性や通信メディア非依存性などを活かすことが可能である。また、既存システムとの互換性も保てるため、既存のインターネット上のコンテンツへのアクセスも確保される。

しかし、走行状態や通信状態などの自動車を取り巻く環境は、自動車の移動に応じて大きく変動する。既存のアプリケーションと違い、インターネット自動車上のアプリケーションは、変動する環境の上でその環境情報を利用した処理や、環境の変化に応じた動作を行なうことが想定される。このようなアプリケーションを自動車環境アプリケーションと位置付ける。自動車環境アプリケーションの動作は、自動車そのものに依存しているため、通常の開発手法では開発および検証が困難である。

本研究では、自動車環境アプリケーションのための開発環境の構築を行なった。まず、インターネット自動車のアーキテクチャ全体について整理を行ない、自動車通信環境およびアプリケーション基盤に関する考察を行なった。考察から、アプリケーション開発環境に対する必要要件を導き、要件を満たす開発環境のモデルとして、仮想自動車によるアプリケーション開発環境モデルを提案した。仮想自動車モデルでは、仮想的な自動車上に、アプリケーションを導入することにより、開発および検証を行なう環境を提供した。仮想自動車構築のため、自動車環境への影響要素の抽出を行ない、自動車環境の再現を実現した。アプリケーション開発者は、仮想自動車を利用することにより、実際の自動車環境と同等の動作検証が可能になる。

本研究の評価として、本システムの動作検証および、既存の開発環境との比較を行ない、本開発環境の優位性を示した。また、評価用アプリケーションの開発・導入の行程を実際に行ない、開発環境としての有効性を示した。

本研究により、通信環境を考慮したインターネット自動車上でのアプリケーション開発環境が構築された。本研究では、インターネット自動車上でのアプリケーションのモデル化を行ない、自動車環境アプリケーションの開発・検証のための開発環境を仮想自動車モデルを用いて構築した。

キーワード

- 1: インターネット自動車
- 2: InternetITS
- 3: 通信エミュレーション
- 4: アプリケーション開発
- 5: 通信環境

慶應義塾大学 政策・メディア研究科
日野哲志

Abstract of Master's Thesis Academic Year 2003

Application Development Environments for Internet CAR systems with Communication Network Emulation

- Summary -

This research provides application development environment for Internet CAR systems with Communication Network Emulation. In addition, an application architecture is summarized and designed for Internet CAR systems.

Internet CAR Project is working to connect vehicles to the Internet, and incubate developments of new applications and services. The main feature of the Internet is end-to-end communication, widely deployed network infrastructure on open platforms. Thus, the Internet can be applied to road-vehicle-human network in order to obtain the features of the Internet.

Vehicle environment, however, is more unstable environment than existing internet environment. According to vehicle's movements, environments around the vehicle such as network conditions, geographical location and so on, are fluctuating. Existing internet applications do not consider such environmental fluctuations. Applications which could make use of such environmental fluctuations will gain more importance in InternetCAR systems. In this paper, these applications are called *Vehicle Environmental Applications*.

In this paper, the application development environment called Vehicle Environmental Applications was developed. For the development of the system, requirements are determined from InternetCAR architecture. As a result, an application development model is designed with virtual vehicles to satisfy above requirements.

Development environment consists of simulating and emulating the vehicle environments in virtual vehicles. Factors for effective vehicle environments is also defined through the development of virtual vehicles. Application Developers can develop and validate their applications using virtual vehicles, since vehicle environments are emulated in virtual vehicles.

As an evaluation, behavior of development environment is validated and advantages over existing systems is reviewed. Sample application for the development environments is developed to evaluate effectiveness of this environment.

From the result of this research, application development environment for InternetCAR systems is developed. In addition, Vehicle Environmental Applications can be developed and validated as a part of application architecture.

Keywords

- 1: InternetCAR
- 2: InternetITS
- 3: Network Emulation
- 4: Application Development
- 5: Communication Environments

Keio University Graduate University of Media and Governance
Tetsuji Hino

目次

第1章	序論	1
1.1	背景	1
1.2	インターネット自動車の課題	2
1.3	研究目的	2
1.4	本論文の構成	2
第2章	インターネット自動車アプリケーションアーキテクチャ	3
2.1	想定するインターネット自動車通信基盤	3
2.1.1	物理レイヤ	3
2.1.2	ネットワークレイヤ	5
2.1.3	通信支援技術	6
2.1.4	通信環境への影響要素	7
2.2	インターネット自動車通信アーキテクチャ	9
2.3	自動車上でのアプリケーション	9
2.3.1	既存アプリケーション	10
2.3.2	インターネット自動車アプリケーション	11
2.4	自動車環境アプリケーション	12
2.5	自動車環境アプリケーションプラットフォーム	13
2.5.1	基盤技術	13
2.5.2	アプリケーションインターフェイス	15
第3章	通信環境を考慮したアプリケーション開発環境	16
3.1	自動車環境アプリケーションの開発	16
3.2	これまでのアプリケーション開発	17
3.3	本研究における開発環境	18
3.3.1	開発環境に対する要求事項	19
3.3.2	既存手法の整理	20
3.4	本研究のアプローチ	20
3.4.1	仮想自動車モデル	21
3.4.2	仮想自動車モデルによる開発環境の機能要件	23
第4章	アプリケーション開発環境の設計	24
4.1	仮想自動車によるアプリケーション開発環境	24
4.2	アプリケーション開発環境設計概要	24
4.3	仮想自動車の設計	26
4.3.1	仮想自動車設計概要	26

4.3.2	アプリケーションの導入手法	27
4.3.3	通信エミュレーション機能設計	28
4.4	通信環境シミュレータの設計	29
4.4.1	通信環境シミュレータへの入力	30
4.4.2	通信環境シミュレータからの出力	30
4.4.3	設計概要	31
4.5	シミュレーション手法	32
4.5.1	通信品質情報の算出	32
4.5.2	周辺情報の出力	33
第5章	アプリケーション開発環境の実装	35
5.1	実装環境	35
5.2	システム構築要素の決定	35
5.2.1	無線通信品質の測定	35
5.2.2	初期設定パラメータ設定	36
5.3	システム構築	38
5.3.1	仮想自動車	39
5.3.2	通信エミュレーション	40
5.3.3	通信環境シミュレータ	42
第6章	検証および評価	44
6.1	システム検証・評価	44
6.1.1	動作検証	44
6.1.2	定性的評価	49
6.2	本システムを用いたアプリケーション開発	50
6.2.1	インターフェイス切り替え機構	51
6.2.2	プローブ情報システム	51
6.3	評価総括	55
第7章	結論	57
7.1	まとめ	57
7.2	今後の課題	57
	謝辞	59
	参考文献	60

目次

2.1	PHS を利用した自動車からのインターネット接続	5
2.2	インターネット自動車通信イメージ	6
2.3	想定するインターネット自動車内部構成	8
2.4	インターネット自動車通信アーキテクチャ	10
2.5	アプリケーションの分類	11
2.6	自動車環境アプリケーション	13
2.7	自動車環境アプリケーションが必要とする情報	14
3.1	自動車環境アプリケーションモデル図	16
3.2	検証環境利用例	21
3.3	仮想自動車モデル	22
4.1	自動車環境アプリケーション開発環境システムモデル	24
4.2	システム概要	25
4.3	仮想自動車設計図	26
4.4	アプリケーションと仮想通信デバイスの対応	28
4.5	通信フィルタ動作概要	29
4.6	通信環境シミュレータ設計図	32
4.7	通信品質算出モデル	33
5.1	802.11b/PDC での帯域幅の変化	36
5.2	仮想自動車初期設定ファイル	37
5.3	802.11b 基地局初期設定ファイル	38
5.4	実装概要	38
5.5	クラス: VirtualVehicle	39
5.6	VirtualVehiclesList クラス	40
5.7	ipfw コマンド設定例	41
5.8	DUMMYNET コマンド設定例	42
5.9	WirelessMap クラス	42
5.10	抽象クラス: AccessPoint	43
6.1	評価環境	45
6.2	仮想自動車の移動・802.11b 基地局の設置	46
6.3	開発環境におけるラウンドトリップタイム変化	47
6.4	開発環境における通信帯域変化	47
6.5	電波到達範囲 500m の場合	48

6.6	電波到達範囲 250m の場合	48
6.7	仮想自動車移動経路・無線 LAN 基地局設置	50
6.8	インターフェイス切り替え時のラウンドトリップタイムの変化	51
6.9	実証実験におけるプローブ情報フォーマット	52
6.10	プローブ情報送信アプリケーション実装例	52
6.11	通信環境を考慮しない場合	54
6.12	通信環境を考慮した場合	54
6.13	通常アルゴリズム利用時	56
6.14	削減アルゴリズム利用時	56

表 目 次

2.1	携帯電話型通信デバイス	4
2.2	IEEE802 系通信デバイス	4
2.3	その他のデバイス	4
2.4	自動車通信への影響要素	9
3.1	これまでの実証実験	18
3.2	既存技術の比較	20
3.3	仮想自動車モデルと既存技術の比較	22
4.1	シミュレータへの入力	30
4.2	通信環境シミュレータからの出力	31
5.1	実装環境	35
5.2	NetworkStat クラスの属性	40
6.1	マシン仕様	44
6.2	要求事項への対応	49
6.3	通信環境の再現によるデータ損失率比較	53

第1章 序論

本章では、本研究の背景、および研究の目的について述べる。また、本論文の構成を示す。

1.1 背景

Intelligent Transport Systems(ITS)をはじめとした、自動車を情報化するための活動が進んできている。ITSは、人・自動車・道路を情報ネットワークで接続することにより、渋滞や交通事故といった道路交通問題を解決することを目的としている。今後は、情報通信に関するITSサービスに高い期待が持たれている。

同時に、情報通信の分野ではインターネットによる通信が普及してきた。インターネットでは、通信メディアに依存しない通信を行なうことが可能である。ITSに、このような特性を持ったインターネット通信を導入することにより、通信メディアを意識しない開かれたシステム開発が、自動車環境上においても可能になる。WIDEプロジェクト[1]のインターネット自動車プロジェクト[2][3]や欧州のOverDRiVE[4]などのプロジェクトが、インターネットによる自動車情報化に関する研究活動を行なってきた。

インターネット自動車プロジェクトでは、1996年から自動車環境におけるインターネット通信技術やハードウェア、およびアプリケーションの開発を行ってきた[5]。通信技術としては、移動体である自動車に対する移動支援プロトコルや、自動車における無線通信デバイス利用に関する技術開発を行ってきた。また、インターネット自動車における車載コンピュータの要件を抽出し、ハードウェアの作成も行ってきた[6]。アプリケーションについては、複数の自動車からセンサ情報を収集し、交通情報や気象情報などを生成するプローブ情報システムや、自動車内への音楽配信システムなどが開発されてきている。ミドルウェアレベルにおいては、IPアドレスと位置情報をマッピングするGLI(Geographical Location Information System)[7]の開発が行われており、インターネット自動車の基盤は徐々に整いつつある。

現在ではこれらの研究成果を基にして、インターネットITS協議会[8][9]として、様々な企業・研究所との共同研究を行っており、その中で実際の自動車を利用した実証実験が実施されている[10]。これまでの実証実験においては、1500台以上のタクシーによる実験車両や、8つの通信デバイスを搭載した実験車両を用いて、通信や様々なアプリケーションの検証を行なっている。

インターネットITS協議会の活動を通して、実際にITSに関わる企業・事業者との共同活動を行なうことにより、今後はインターネット通信による自動車情報化の普及が進んでいくことが予想される。

1.2 インターネット自動車の課題

これまでのインターネット自動車プロジェクトの活動により、自動車上でのインターネット通信に関する様々な基盤技術が研究・開発され、インターネット自動車におけるアプリケーション基盤が整備されつつある。

既存のインターネット上のアプリケーションは、通信のレイヤ構造というアーキテクチャにより下位レイヤが隠蔽されており、開発時にはネットワークなどの通信環境について考慮する必要が無かった。しかし、移動体である自動車上では、通信環境や地理位置情報をはじめとする環境の変化により動作を変化させるアプリケーションが開発されると想定される。自動車は全世界で 8 億台存在すると言われ、対象となる台数規模も大きく、信号や道路・天候など、影響を与える要素も複雑に関連しあう。

このような自動車環境上のアプリケーションの開発に必要なアプリケーションアーキテクチャに関する考察はなされていない。また現在、自動車環境でのアプリケーション・基盤技術などを検証する有効な手法は存在していない。このため、擬似的に作成したデータを利用するか、実車を利用した実証実験を行なう必要があった。擬似データでは自動車環境での環境からかけはなれており、また実証実験はコストが高い。

1.3 研究目的

本研究では、インターネット自動車における、通信環境を考慮したアプリケーション開発環境の構築を行なう。インターネット自動車におけるアーキテクチャ全体に関する考察を行ない、開発環境に必要とされる要件を抽出し、アプリケーション開発環境を構築する。

まず、アーキテクチャの考察として、インターネット自動車システム全体をいくつかのレイヤに分割し、通信基盤からアプリケーションまでの整理を行なう。その整理から、自動車環境に影響を与える要素の抽出を行なう。また、アプリケーション基盤技術として、アプリケーションの動作を支援する各技術についても考察する。

インターネット自動車におけるアーキテクチャ全体の考察から得られた、要件を満すモデルを用いて、アプリケーションの開発および検証を行なう環境の構築を行なう。

1.4 本論文の構成

本論文は以下の 7 章により構成される。第 2 章では、インターネット自動車におけるアプリケーションアーキテクチャについて考察し、自動車環境アプリケーションの重要性を述べる。第 3 章では、本研究で提唱する自動車環境アプリケーションのための、通信環境を考慮したアプリケーション開発環境の必要性と、開発のアプローチについて論じる。第 4 章で開発環境の設計、第 5 章で実装手法について述べる。第 6 章では評価として、開発環境の動作検証および既存手法との比較から本開発環境の優位性を述べ、かつ実際に開発環境を用いた自動車環境アプリケーションの開発を行ない有効性を示す。最後に、第 7 章で本研究の成果をまとめ、今後の課題について述べて結論とする。

第2章 インターネット自動車アプリケーションアーキテクチャ

本章では、本研究で想定するインターネット自動車のアプリケーションアーキテクチャについて述べる。まず、想定されるインターネット自動車の通信基盤について述べる。その後、インターネット自動車通信全体のアーキテクチャについて述べ、通信基盤の上のアプリケーションについて述べる。最後に、自動車環境アプリケーションについて述べる。

2.1 想定するインターネット自動車通信基盤

まず、インターネット自動車において想定される環境について述べる。自動車環境においては、移動に伴って、通信環境が物理レイヤおよびネットワークレイヤで変化する。また、位置情報をはじめとする自動車を取り巻く周辺環境も変化していく。

2.1.1 物理レイヤ

本節では、インターネット自動車で想定されている物理レイヤについて述べる。まず、利用が想定される無線通信デバイスを列挙し、それぞれの特徴をまとめる。その後、予備実験から判明した自動車における通信環境について述べる。

表 2.1-2.3 にインターネット自動車環境で利用が想定される無線通信デバイスを挙げる。自動車は移動体であるため、インターネットへの接続には無線通信デバイスの利用が前提となる。ここではそれらの無線通信デバイスを、携帯電話型通信デバイス・IEEE802系通信デバイス・その他デバイスに分けてまとめる。

表 2.1-2.3 で示したように、インターネット自動車環境では様々な無線通信デバイスの利用が想定されており、かつそれぞれの通信デバイスの特徴は大きく異なっている。特に通信範囲・利用可能帯域・通信方式などの差異は、アプリケーションに大きな影響を与える。

たとえば、遠隔監視など、常に自動車との通信を行なう必要のあるアプリケーションの場合、現状の無線 LAN のように限られた範囲でしか利用できないデバイスでの通信は不適である。遠隔監視などのアプリケーションで、やりとりされるデータが小さい場合には、PDC などを利用することにより、より広範囲・長時間の監視が可能になる。しかし、動画コンテンツのように大容量のデータを送受信する場合には、PDC では輻輳が不可避である。

次に、実際に自動車から無線通信デバイスを利用した際の、ネットワーク接続品質についての予備実験結果を示す。本研究におけるネットワーク品質とは、IP レベルでの通信帯域、通信遅延およびパケットロス率などを示す。予備実験では、実験車両に最大 128Kbps のデータ通信デバイスである、DDI-Pocket 社 [11] の AirHTM のコンパクトフラッシュ型カードを搭載し、IPv6 トンネル [12] を利用した IPv6 接続を行なった。トンネルの利用により、IP アド

表 2.1: 携帯電話型通信デバイス

大分類	種別	伝送速度 (理論値)	特徴
携帯電話・PHS	PDC	9.6kbps/28.8kbps	第 2 世代携帯電話通信。日本では広範囲で利用可能
	cdmaOne	14.4kbps	第 2.5 世代携帯電話通信
	cdma2000	~ 384kbps/2Mbps	第 3 世代携帯電話通信。G-BOOK で利用
	W-CDMA	384kbps ~ 2Mbps	第 3 世代 日欧で標準化 (FOMA 等)
	TD-CDMA	~ 3.3Mbps	第 3 世代 IMT-2000 の一部
	PHS	~ 128kbps	電波出力が小さく基地局から 100m 以内で通信可能 基地局は広く設置されている

表 2.2: IEEE802 系通信デバイス

大分類	種別	伝送速度 (理論値)	特徴
IEEE802 系	IEEE802.11a	~ 54Mbps	5.2GHz 帯周波数を利用。CSMA-CA
	IEEE802.11b	~ 11Mbps	2.4GHz 帯周波数を利用。CSMA-CA
	IEEE802.11g	~ 54Mbps	2.4GHz 帯周波数を利用。CSMA-CA

表 2.3: その他のデバイス

大分類	種別	伝送速度 (理論値)	特徴
その他	衛星通信	236kbps ~ 413kbps	UDLR(片方向リンク)。衛星が見えれば通信可能
	DSRC	N/A	狭域無線通信システム。ETC で利用。
	Bluetooth	N/A	電力消費が少ない。Personal Area Network などの構築向き
	IrDA	N/A	赤外線通信
	FSO	N/A	数 mW 程度の光通信を用いた通信

レスは変化しないため、通信デバイスの品質変化を測定することが可能である。本予備実験により、自動車環境における無線通信デバイスの性能がどのように変動するかを示す。

図 2.1 に各時刻ごとのネットワーク遅延の分布を示す。遅延は ping6 を利用して、ラウンドトリップタイムの分散を表示する SmokePing [13] というツールを利用して測定している。縦軸はラウンドトリップタイムを、横軸は時刻を表わし、グラフの表示の濃さによりデータの分散を表現している。

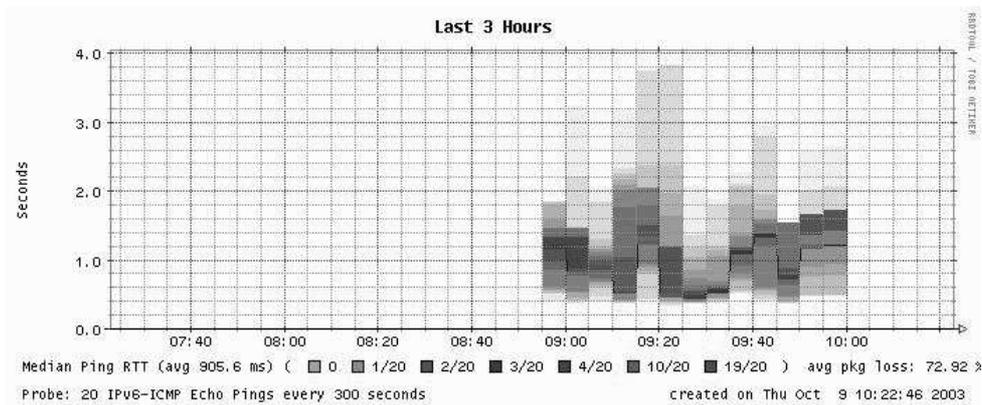


図 2.1: PHS を利用した自動車からのインターネット接続

図 2.1 から分るように、単一の通信デバイスであっても、移動に伴って通信遅延は大きく変動している。この理由としては、自動車の移動による基地局との位置関係や、遮蔽物などが影響すると考えられる。このような性能の変動は、自動車環境でなくても発生するが、自動車環境においては移動速度や距離が大きく、品質の変動が激しい。

2.1.2 ネットワークレイヤ

インターネット自動車のような移動体では、2.1.1 節で述べた無線通信品質以外にも、物理的な移動に伴うネットワーク間の移動により、ネットワークレイヤにおける IP アドレスが変化していく。IP アドレスは、インターネット通信における識別子であるため、IP アドレスの変化が起こると、継続した通信が不可能になる。特に移動体では、移動が激しく IP アドレスが頻繁に変化する環境が想定される。

有線での接続が主体となっている既存のインターネットでは、通信メディアに依存した帯域の差は存在するが、無線通信と比較すると安定した通信が可能である。既存インターネット上のアプリケーションは、有線での安定した通信を前提として構築されており、自動車をはじめとした移動体上で発生する通信環境の変化には対応できない。

図 2.2 に、インターネット自動車が街中を走行するイメージを示す。広く全体に PDC のエリアがあり、PHS エリアは一部通信できない地域を持つ。無線 LAN は基地局が設置されているごく一部でしか利用できない。自動車の走行に応じて、利用する通信デバイスは変化し、それに伴ない通信品質・IP アドレスなどが変化していく環境が想定される。

移動体間での直接通信を実現するため、MANET ルーティングプロトコル [14] などを利用する場合もある。MANET ルーティングプロトコルは、複数の移動体を經由するアドホックネットワークを構築するためのルーティングプロトコルである。代表的な MANET ルーティングプ

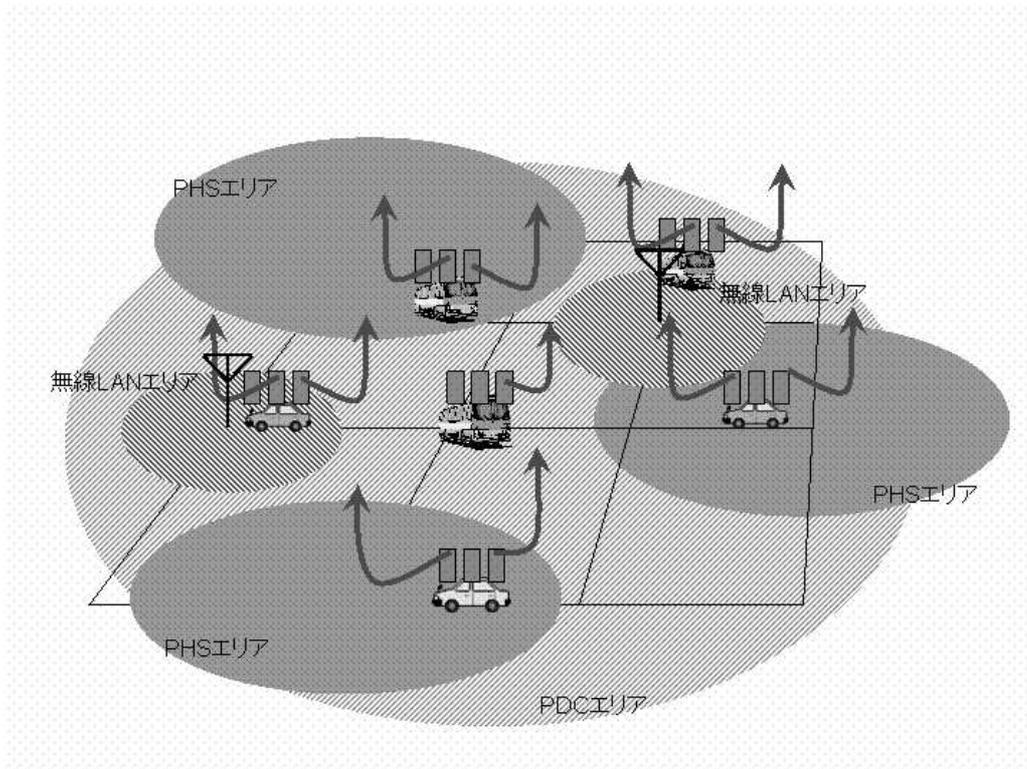


図 2.2: インターネット自動車通信イメージ

ロトコルとしては、AODV(Ad Hoc On-Demand Distance Vector)[15]・DSR(Dynamic Source Routing)[16]・TBRPF(Topology Dissemination Based on Reverse-Path Forwarding)[17]・OLSR(Optimized Link State Protocol)[18]などが提案されている。AODV・DSRは通信が発生した後に経路を作成する re-active 型、TBRPF・OLSRは前もって経路を作成する pro-active 型ルーティングプロトコルと分類される。全ての MANET ルーティングプロトコルでは、電波到達範囲内に自動車が存在しない場合には、ネットワークレイヤでのルーティングを提供することができなくなる。

2.1.3 通信支援技術

2.1.1- 2.1.2 節で挙げたように、自動車環境における通信環境は大きく変動する。そのような環境上でのアプリケーション動作のため、現在研究されている通信支援技術について述べる。

複数通信デバイス利用支援技術

2.1.1 節で述べたように、インターネット自動車システムで利用が想定される無線通信デバイスは、それぞれ様々な特徴を持っている。各々のデバイスの利点・欠点を補完するため、インターネット自動車では、複数の無線通信デバイスを搭載し、それらを通信品質・用途などに応じて使い分けることが想定されている。これらの技術を総括して、複数通信デバイス利用支援と呼ぶ。利用可能範囲に応じて、使用する通信デバイスを切り替える機構としては、

インターフェイス切替機構 [19] がある。この機構では、通信デバイスの電波強度の情報をもとに利用するインターフェイスを切り替える。

また、通信先や通信プロトコル、ポートなどの通信の内容に応じて、通信デバイスの使い分けを実現している研究も存在する [20]。この機構では、通信デバイスの切り替えだけではなく、複数の通信デバイスを並行して利用することが可能になる。たとえば、広帯域な通信帯域を必要とするアプリケーションは無線 LAN を利用し、それ以外の通信は携帯電話網を利用することにより、2 つの通信機器の並行利用が可能となり、より効率的な通信デバイスの利用が実現される。

これらの技術により、アプリケーションが利用する通信デバイスが切り替えられた際には、IP アドレスの変化が発生する。このため、次で述べる移動支援プロトコルを利用しなければ、アプリケーション通信は継続できない。

移動支援プロトコル

2.1.2 節で述べたネットワークレイヤでの変動を隠蔽するため、移動支援プロトコルが考案されている。主な移動支援プロトコルとしては、Mobile IP[21] や Mobile IPv6[22]、Network Mobility 技術 [23][24] が研究、開発されている。インターネット上の移動ホストは、移動に伴って通信に利用する IP アドレスが変化し、継続的な通信が行えない可能性がある。これらの移動支援プロトコルでは、移動ホストの現在の IP アドレス (Care Of Address: CoA) と、Home Address と呼ばれるアドレスの対応を保持することによって、Home Address による継続的な通信を提供する。

Mobile IP では、単一の移動体の移動支援を実現するプロトコルである。また、Network Mobility 技術は複数のノードを持つネットワーク全体の移動を支援する技術である。自動車内では、ユーザが持ちこむ PDA や、カーナビゲーションシステム・カーオーディオ、また各種センサなどが、それぞれ独立したインターネットノードとしてインターネットに接続する。このように、自動車内には、複数のインターネットノードが存在する環境が想定される。そのため、インターネット自動車システムでは、Network Mobility 技術を利用し、自動車自体を移動するネットワークと位置付ける。

図 2.3 は、想定されるインターネット自動車内の構成を示している。車載ルータは、IEEE802.11b・携帯電話・PHS の 3 つの通信デバイスを持っている。また、車内には LAN が設置されており、センサなど車内のデバイスが、それぞれ LAN に接続している。

このようにインターネット自動車では、様々な特徴を持つ複数の通信デバイスを切替および並行して利用することが想定されている。このため、単一の通信デバイスの性能だけではなく、それらを組み合わせて利用した場合の性能がアプリケーションにとって重要な要素となる。

2.1.4 通信環境への影響要素

本節では自動車通信環境に影響を与える要素について述べる。表 2.4 に要素を列挙する。ここでは多数の要素を網羅的に整理するため、4 つの大分類を設ける。

通信状態情報としては、対象となる自動車がどのような通信を行なっているかが影響する。

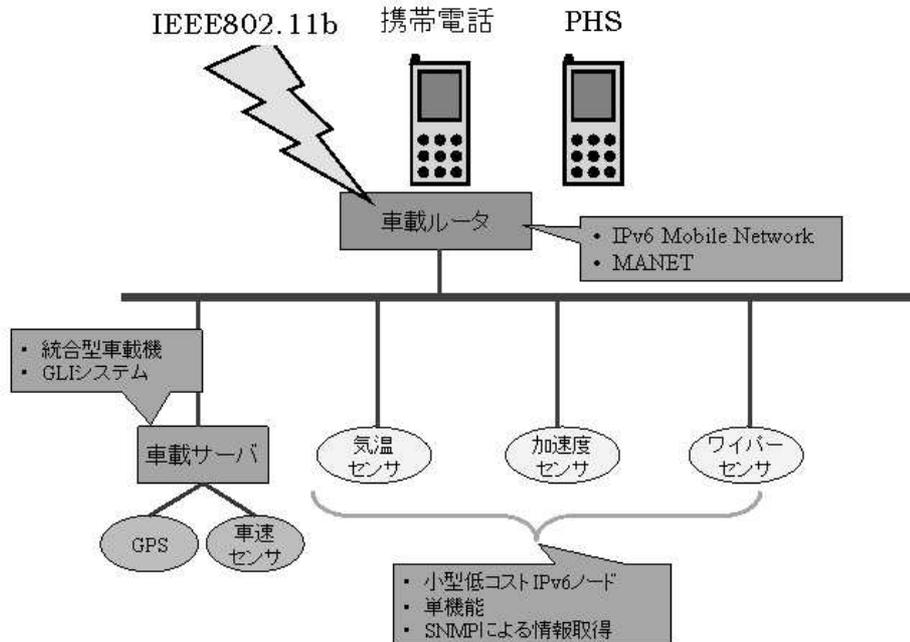


図 2.3: 想定するインターネット自動車内部構成

発生するトラフィック量やその通信プロトコルなどの情報がある。

交通情報としては、対象となる自動車の位置・速度などの走行情報や、周辺車両の密度などがある。これらは特に他車両を中継する通信など、車両間の直接通信を行なう上で非常に大きな影響を与える。

基地局情報としては、各無線通信機器の基地局側の情報がある。設置の密度や、現在の利用率などがある。これらは、自動車側の利用通信デバイスの実効性能に影響する。

最後に道路周辺情報としては、自動車位置している道路に関わる様々な情報がある。気象情報などは、無線の電波伝搬に影響を及ぼす場合もあり、また工事・事故などのイベントによりアプリケーションの通信に変更が発生することも考えられる。

これらの情報は、それぞれ互いに影響をしながら自動車通信環境を決定付ける。自動車通信情報は、道路周辺に発生するイベントや、走行情報によって変化する。また、基地局情報は、各自動車の位置や、それぞれが発生するトラフィック量などにより変化する。このように自動車の通信環境には、通信そのものだけでなく、自動車を取り巻く様々な情報が影響している。

表 2.4: 自動車通信への影響要素

分類	要素
自動車通信情報	発生トラフィック量
	通信プロトコル
	利用通信デバイス
走行情報	位置情報
	速度情報
	周辺車群密度
基地局情報	基地局設置密度
	利用度合
道路周辺情報	気象情報
	工事・事故などのイベント情報

2.2 インターネット自動車通信アーキテクチャ

図 2.4 に本研究で想定するインターネット自動車通信アーキテクチャを示す。2.1 節で述べてきた、インターネット自動車上の通信基盤技術および、アプリケーションをいくつかのレイヤに分割して整理する。レイヤの分割は、OSI 参照 7 階層モデルを参考にしているが、ここでは物理レイヤ、ネットワークレイヤ、インターネット自動車トランスポートレイヤ、アプリケーション基盤レイヤ、アプリケーションレイヤの 5 つに分割する。この分割により、インターネット自動車に関わる要素技術の機能整理が可能になる。

最下位には物理レイヤ が位置する。ここには、OSI 7 階層モデルで言うデータリンク層までが含まれる。インターネット自動車上では、表 2.1-2.3 で示した無線通信デバイスが主に利用されると想定される。

物理レイヤの上には、ネットワークレイヤがある。IP(Internet Protocol)[25] [26] や IPsec[27]、そして、移動体支援プロトコルである MobileIP、Mobile IPv6 や Network Mobility などのネットワークプロトコルが相当する。

また、インターネット自動車におけるトランスポート層として、インターネット自動車トランスポート層も考案されている。個別技術としてはまだ開発には至っていないが、パケット到着のジッタなどの通信の揺らぎを吸収する技術がこの層に導入される。

4 つ目のレイヤをアプリケーション基盤レイヤ と呼ぶ。ここには一般的にミドルウェアと呼ばれる、アプリケーションに対する基盤ソフトウェアが相当する。

5 つ目のレイヤには、アプリケーションレイヤがくる。ここにはインターネット自動車上で構築される様々なアプリケーションが入る。

ここで述べたレイヤモデルに基づいて、インターネット自動車環境上でのアプリケーションについて述べる。

2.3 自動車上でのアプリケーション

ここでは、自動車環境におけるアプリケーションについて述べる。既存の自動車環境上で実現されているシステムについてまとめ、その後、インターネット自動車により実現可能な

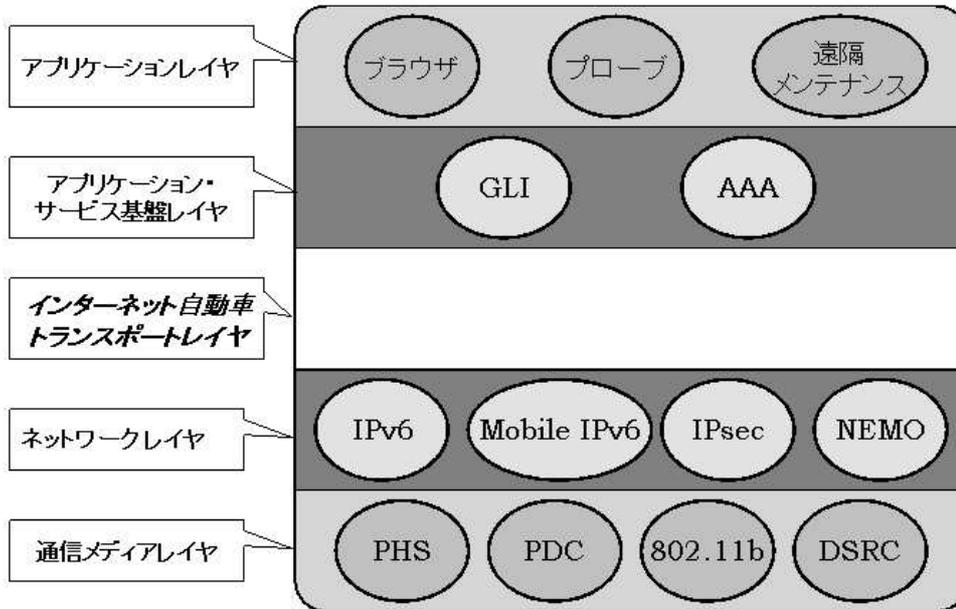


図 2.4: インターネット自動車通信アーキテクチャ

アプリケーションについて述べる。

インターネット上のアプリケーションの分類を行ない、それぞれのエリアで考えられているアプリケーションについてまとめる。

図 2.5 にインターネット自動車上でのアプリケーションをまとめる。図の縦軸では、アプリケーションの対象による分類を行なう。上からユーザ・車・道路を対象するアプリケーションを挙げる。横軸には、実現に必要なとされる技術を示す。左側には既存技術で実現可能なアプリケーション、右側にはインターネット通信を利用することにより実現可能なアプリケーションを示す。

2.3.1 既存アプリケーション

図 2.5 の中軸より左側に示す、既存技術によるアプリケーションについて述べる。これらのアプリケーションの中では、ITS の枠組において開発がされてきたシステムが比較的大規模なものである。

日本における ITS では、ETC(Electronic Toll Collection: 自動料金収受システム)[28]・VICS(Vehicle Information and Communication System: 道路交通情報通信システム)[29]などが現在サービスされている代表的なものである。

ETC では、高速道路などの料金所に DSRC の路側アンテナを設置し、車両内の車載器との通信により自動での料金収集を実現している。

VICS では、道路管理者や都道府県警察から収集した道路状態の情報を VICS センタで収集・道路交通情報を生成し、電波ビーコン・光ビーコン・FM 多重放送などを利用して自動

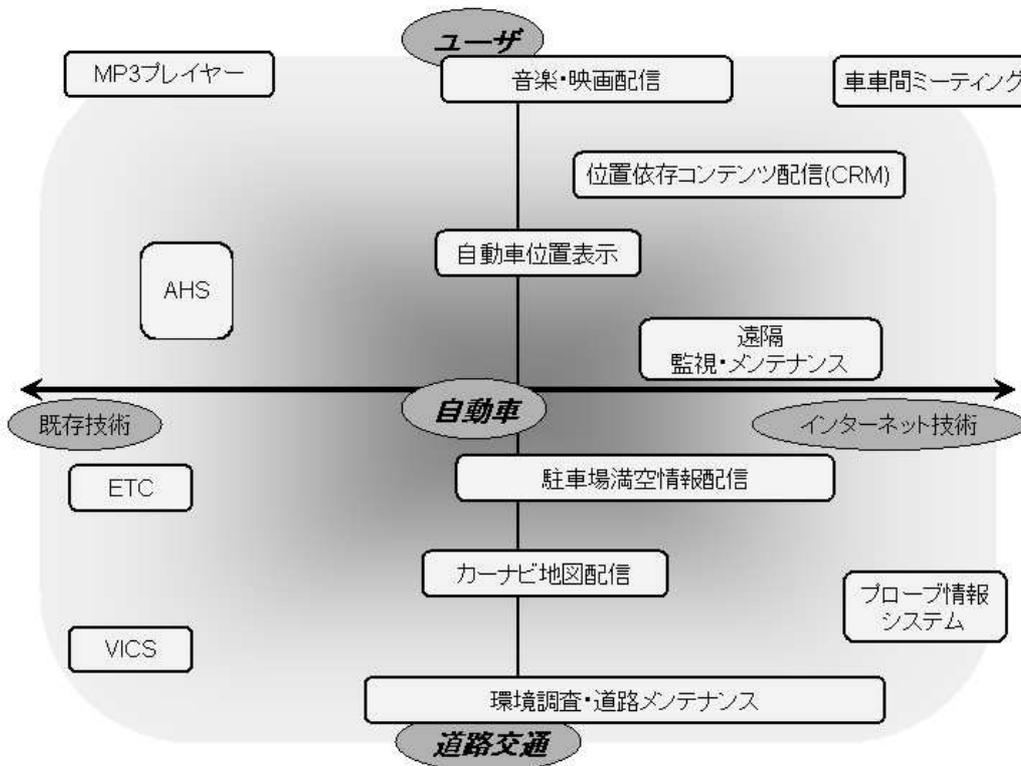


図 2.5: アプリケーションの分類

車に提供している。

また、AHS(Advanced Cruise-assist Highway Systems: 走行支援道路システム) [30] なども実用化に向けて、開発が行なわれているにある。AHS では、道路・自動車の協調により、ドライバーに対して情報提供・警報・操作支援を実現することにより、ドライバーの走行を支援することを実現する。

この他にも、ユーザを対象としたアプリケーションとして、車載の MP3 プレイヤーやラジオ、DVD やハードディスクなどを利用したカーナビゲーションシステムなども既存アプリケーションである。

2.3.2 インターネット自動車アプリケーション

図 2.5 の中軸より右側には、インターネット技術を利用したアプリケーションを列挙した。その中でも、特徴的なアプリケーションについて詳述する。

車車間ミーティング [31]

ユーザを対象としたアプリケーションである。インターネットへの接続を利用せずに、車両間の無線アドホックネットワークを利用して直接コミュニケーションを行なう。

位置依存コンテンツ配信 [32]

主に車両の位置情報を取得し、その地理位置に依存した広告配信や、音楽・映像配信を行なう。インターネットを利用して位置情報取得・情報配信を行なうため、既存のコンテンツ会社が自由に参入可能である。位置情報交換のための仕様として、MOSTEC(Mobile

Information Standard Technical Committee:モバイル標準化検討委員会)で策定された POIX(Point Of Interest eXchange language)[33] がある。

遠隔車両診断

自動車そのものを対象としたアプリケーションとなる。車両の様々な情報を遠隔から監視し、異常発生時にすみやかな対応を行なう。盗難・事故などを主な対象とするが、カー用品店などにより、CRM に利用されることも考えられる。

プローブ情報システム [34]

主に道路交通全体を対象としたアプリケーションである。自動車が持つワイパーや速度などの情報を、収集し地図との対応付けを行なうことにより、リアルタイムでの天候情報や、渋滞情報などの生成を目的としたシステムである。

これらの自動車環境における様々な情報をもとにした動作を行なうアプリケーションを自動車環境アプリケーションと呼ぶ。これまで述べてきたように、自動車環境では移動に伴ってアプリケーションの動作環境が大きく変動する。また、自動車上では、ユーザインターフェイスが乏しい可能性が高く、それらの環境の変動にアプリケーションが動的に対応する必要性が生じる。このため、本研究では、自動車環境への動的な対応を行なうアプリケーションとして、自動車環境アプリケーションという分類を行なう。

2.4 自動車環境アプリケーション

本節では自動車環境アプリケーションの詳細について述べる。図 2.6 に自動車環境アプリケーションの概要を示す。自動車環境アプリケーションは、自動車に関わる様々な環境情報の変化に対応して動作する。走行状態・通信状態・周辺環境の状態を自動車が備えるセンサから取得し、それに基づいた動作の変更や、環境情報自体の送信などを行なう。また、車内で動作するアプリケーションだけではなく、車外のアプリケーションであっても自動車内の環境情報をもとにしたものであれば、自動車環境アプリケーションに含まれる。

次に、自動車環境アプリケーションが必要とする情報を大きく位置情報や速度情報などの車両情報と、通信状態に分類する。ここでの通信状態はインターネットへの通信状態を示す。図 2.7 では、自動車環境アプリケーションを必要とする情報の種類によりさらに分類した。横軸には、車両情報への依存度を示し、縦軸には通信状態への依存度を示す。

通常の WEB ブラウジングなどを行なう場合には、車両情報にも通信状態にも依存しない。また、ETC・AHS・VICS などの既存 ITS システムは、専用通信インフラを持つため、自動車情報にのみ依存する。

プローブ情報システムは、車両の情報を収集するシステムであるため、車両情報に高く依存する。また、通信量を削減するため通信状態に応じた情報発信なども検討されているため、通信状態にも依存している。

ストリーミングアプリケーションでは、通信品質に応じて、動画品質やサンプリングレートを変化させ、利用帯域を調節することにより、輻輳が発生しない通信が可能である。

遠隔車両診断でも、車両の情報をもとに通信を行なうため、車両情報および通信状態に依存する。

位置依存コンテンツ配信は、車両の状態に応じて、車両の通信状態に沿った形態でコンテンツを配信する配信システムである。このようなアプリケーションは、車両情報・通信状態

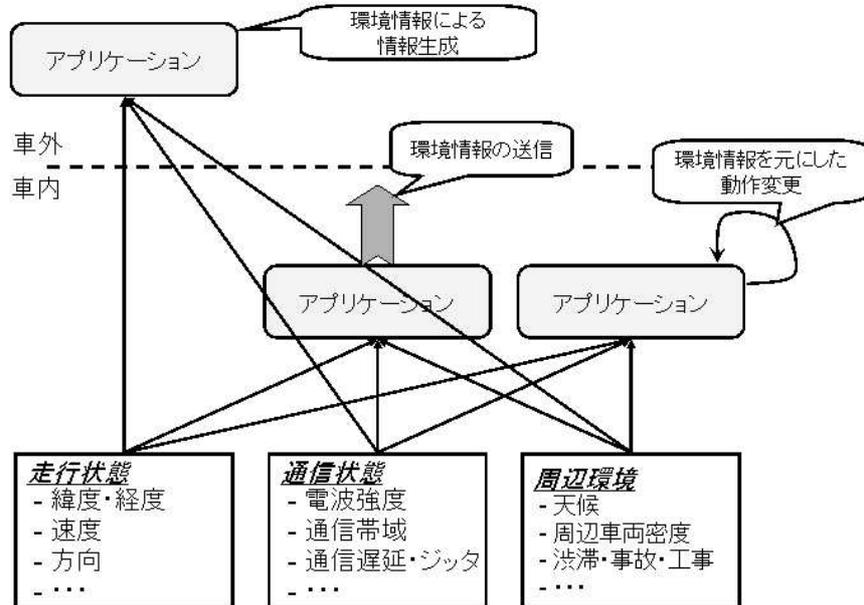


図 2.6: 自動車環境アプリケーション

にそれぞれ強く依存する。

車両間ミーティングにおいては、車両間での MANET を利用した直接通信が必要となる場合がある。そのため、車両同士の位置情報および、その車両間での通信の情報に著しく依存する。

これまでのインターネット上のアプリケーションは、移動支援プロトコルなどにより、通信状態に関する情報を隠蔽されて動作してきた。これは、アプリケーションが移動に対応した形で構築されていなかったからである。しかし、インターネット自動車の実用化に伴ない、移動体からのインターネットアクセスが一般化していくと、ネットワーク的および地理的な移動に、適応して動作する自動車環境アプリケーションが重要となってくる。

2.5 自動車環境アプリケーションプラットフォーム

ここでは、自動車環境アプリケーションのプラットフォームについて述べる。まず、アプリケーション開発の基盤となる技術について述べる。そして、自動車環境アプリケーションが、各情報にアクセスするためのアプリケーションインターフェイスについて述べる。また、車載器側のアプリケーションだけではなく、外部から車内の情報にアクセスするインターフェイスも必要となる。

2.5.1 基盤技術

アプリケーションは単体で存在するのではなく、レイヤ構造に分割された基盤技術の上に存在する。ここでは、インターネット自動車上でのアプリケーション基盤技術について述べ

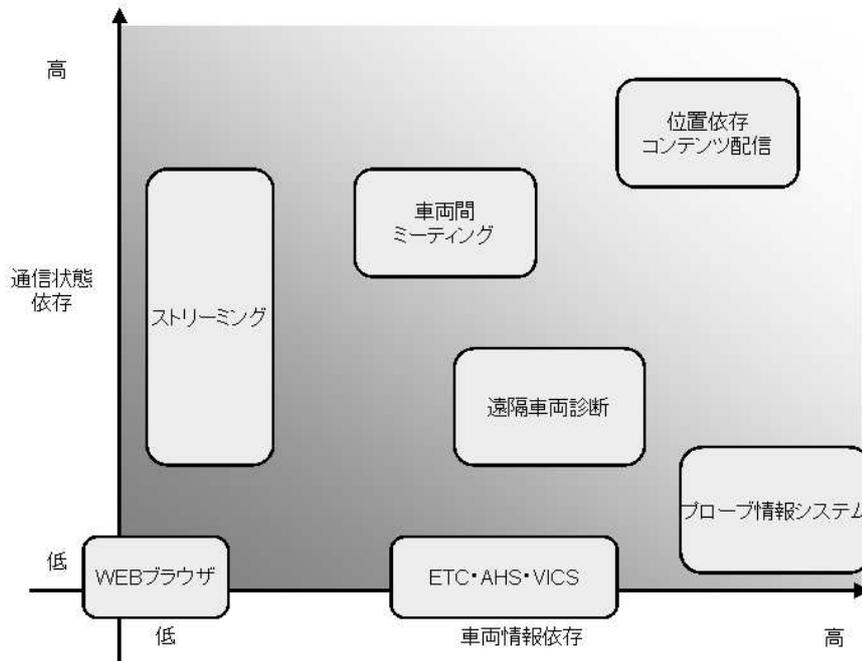


図 2.7: 自動車環境アプリケーションが必要とする情報

る。ネットワーク層以下については既述のため、トランスポート層以上の基盤技術について述べる。

インターネット自動車トランスポート層

インターネット自動車におけるトランスポートレイヤでは、ネットワーク層での通信の不安定性を吸収する機構が必要である。インターネット自動車では無線通信を利用していることから、ジッタやパケットロスなどが発生しやすい環境が発生する。トランスポートレイヤにおいて、バッファを保持することにより、ジッタを回避することが可能である。

トランスポート層において、通信の不安定性を吸収する取り組みは、既存インターネットでも行なわれてきた。しかし、インターネット自動車上での通信は、既存インターネットと比較して大きく不安定である。そのため、自動車環境に対応したトランスポート層が必要となる。

アプリケーション基盤レイヤ

アプリケーション基盤レイヤには、アプリケーション動作のための基盤となる技術が相当する。地理位置情報の管理や、ネットワーク利用の認証などのミドルウェアが存在している。

地理位置情報管理を行なうシステムとしては、GLI(Geographical Location Identification System)の開発が行なわれている。GLIでは、自動車をはじめとする移動体の緯度・経度・高度の情報と、インターネット上の識別子であるIPアドレスの対応を保持する。GLIでは、正引き検索と逆引き検索の2つの検索手法がある。正引き検索では、車両のIDから、そのIDを持つ移動体の緯度経度高度を取得する。逆引き検索では、緯度経度高度で表現された

エリアから、そのエリア内にいる移動体の ID リストを取得することが可能である。現状の GLI は HasheID を利用したプライバシー保護や、データベースの分散化などの特徴を持ち、高度化されている。

また、ネットワーク認証を行なう機構が検討されている。IETF AAA ワーキンググループでは、ネットワーク層プロトコルである Mobile IP の利用に関する AAA の仕様 [35] を定めている。しかし、自動車は高速で移動するため、複数の無線 LAN 基地局間をシームレス、つまり通信が途切れることなく、移動するための機構が必要である。このため、自動車に特化した認証機構が必要とされている。

2.5.2 アプリケーションインターフェイス

自動車環境アプリケーションでは、車両情報と通信状態への情報アクセスが必要となる。ここでは、アプリケーションからそれらの情報へのインターフェイスについて述べる。

車両データ辞書モデル [36] では、車両が保持する位置などの車両状態を示す情報の書式および保持方法を定めている。この車両データ辞書に対して、SNMP[37] などの標準的な手法でのアクセスすることによりアプリケーションが車両情報を取得することが可能になる。現在、ISO/TC204/WG16/SWG16.3 において、車両データ辞書の標準化が進められている [38]。

また、通信状態の情報取得インターフェイスも必要となる。たとえば、利用しているインターフェイスの利用可能帯域に応じて、コンテンツのダウンロード時に圧縮を行なうなどが考えられる。このような通信環境の情報は、上で述べたデータ辞書には、現状では含まれていない。

このため、通信環境情報取得 API を定める必要がある。MIBSocket[39] などの研究では、アプリケーションは MIB(Management Information Base) [40][41][42] を利用してインターフェイスの情報にアクセスすることが可能である。SNMP による取得では、アプリケーションが車内であっても車外であっても統一したインターフェイスで自動車を持つ情報にアクセスすることが可能である。

この他にも、Odyssey[43]・CM1 [44]・Environment Server[45] などの研究で、ネットワーク層の情報を利用したアプリケーション動作変更のためのフレームワークが考案されている。

第3章 通信環境を考慮したアプリケーション開発環境

本章では、通信環境を考慮したアプリケーション開発環境について述べる。

3.1 自動車環境アプリケーションの開発

本研究では、インターネット自動車上でのアプリケーション開発を行なうためのアプリケーション開発環境の構築を目的とする。特に、自動車の移動により変化する情報をもとにして動作する自動車環境アプリケーションをターゲットとする。

一般的なアプリケーション開発の工程をモデル化すると、1)設計 2)実装 3)デバッグ 4)実験・動作検証という4工程に分割できる。通常アプリケーションでは、これらの工程を単一の計算機もしくは、数台の計算機により実現することができる。

自動車環境アプリケーションでは、この工程の中で1)から2)までは、通常の開発と同様に行なうことが可能である。しかし、自動車環境アプリケーションの開発において、3)のデバッグ 4)の実験・動作検証を行なうためには、実際に自動車や車載器を用意する必要がある。また、緊急事態などを想定したアプリケーションの場合、緊急事態を実際の道路上で再現することは困難である。このため、自動車環境アプリケーションの開発を行なうための環境が必要である。

図3.1に自動車環境アプリケーションのモデル図を示す。車内に多数存在するセンサが持つ情報は車両データ辞書で保持されている。各自動車環境アプリケーションは、車両データ辞書のインターフェイスを用いて情報にアクセスし、車両データに基づいた動作を行なう。

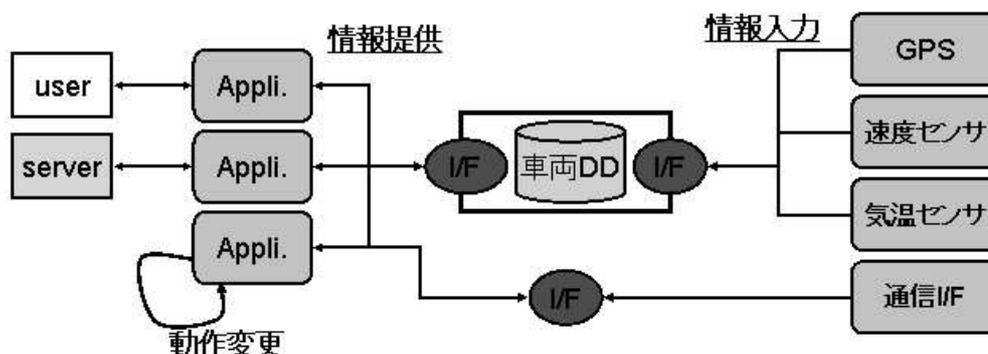


図 3.1: 自動車環境アプリケーションモデル図

ユーザに対する出力や、他のサーバなどのアプリケーションとの通信を行なう。また、通信インターフェイスに関する情報を通信基盤からインターフェイスを介して取得し、その値による振舞いの変更も行なう。

このように自動車環境アプリケーションは、通信状態および自動車周辺をもとにした動作を行なう。このようなアプリケーションのデバッグ・開発や、振舞いを変更するために必要なパラメータの設定を行なうには、実際にデータ辞書や通信基盤からの情報取得が可能である必要がある。

3.2 これまでのアプリケーション開発

3.1 節で述べたように、インターネット自動車上におけるアプリケーションの開発が行なわれていく。また、実際にサービスとして提供するためには、その有効性の検証も必要となる。本節では、既存のアプリケーション開発・検証方法とその問題点について述べる。本研究では、自動車環境におけるインターネットアプリケーションを対象としているため、これまでのインターネットアプリケーションの開発で用いられていた手法を主にまとめる。

シミュレータによる開発

大規模かつ複雑なネットワーク上で動作するアプリケーションは、シミュレータにより開発される。実際の環境をシミュレートした環境で、アプリケーションを動作させ、検証を行なう。

これまでのインターネットでは、ネットワークシミュレータを利用した、ルーティングプロトコルの検証が行なわれてきた。多くの MANET(Mobile Ad Hoc Network) のルーティングプロトコルの検証では、Random Way Point や Freeway といったいくつかの簡単な移動モデルによる移動を再現し、ルーティングプロトコルの動作検証を行なっている。また、通常の動的ルーティングプロトコルでも、多数のルータ間での検証が必要な場合には、シミュレーションを用いた検証を行なう。既存のネットワークシミュレータには ns-2[46] や、移動体のシミュレーションに特化した GloMoSim[47] などがある。

ランダムデータによる開発

ランダムデータを用いた検証は主にスケーラビリティの検証に利用される。アプリケーションによって、実現方法は様々存在するが、たとえば、サーバの耐故障性を検証する際には、ランダムで生成したデータを送信させ、最悪のケースを再現することにより耐故障性を検証している。また、あるイベントに基づいて動作するシステムであれば、ランダムな間隔でイベントを発生させ、動作の検証を行なう。既存のツールとしては、ICMP Echo Request を発生させる ping や、利用可能帯域を測定するための netperf などのツールがある。

ランダムデータによる検証では、非常に柔軟性の高い検証が可能である。しかし、ある 1 つのパラメータを測定するために、1 つのアプリケーションが必要となる。このため、いくつかのパラメータを同時に測定することが困難である。また、ランダムなデータであるため実際の利用環境とは異なった環境での検証となる。

表 3.1: これまでの実証実験

年度	Internet CAR(WIDE)			IPCar(JSK)			Internet ITS		
	1996	1997	1998	1999	2000	2001	2001	2002	2003
特徴	初めての 実験	Mobile IPの導 入	専用機 の投 入	プローブ の可 能性 検証	社会実 験	精度向 上	基盤技 術の 確立、 IPv6	ビジネ スモ デル 模 索	ビジネ スモ デル 確立、 モバ イル ネッ トワ ーク
台数	1	7	10	10	270	270	1640	1490	1490+30
地域	藤沢	藤沢	藤沢、 奈良、 石川	港北	横浜	横浜	川崎、 名古 屋	名古 屋	名古 屋、 横 浜
車種	・実験車	・一般車	・一般車	・実験車	・タクシー ・バス ・トラック ・塵芥車	・タクシー ・バス	・タクシー ・一般車	・タクシー	・バス ・タクシー
車載機	・PC	・Note PC	・sic2000	・専用機	・専用機	・専用機	・専用機	・専用機	・専用機 ・車載ルータ ・センサー ノード
取得情 報	・位置 ・車速 ・ワイ パー ・ライト	・位置 ・車速 ・ワイ パー ・ライト	・位置 ・車速 ・ワイパー ・ライト	・位置 ・車速 ・ワイパー ・ライト	・位置 ・車速 ・ワイパー ・ライト	・位置 ・車速 ・ウイン カー ・サイドブ レーキ ・ライト	・位置 ・車速 ・ワイパー ・実車空車 ・ライト	・位置 ・車速 ・ライト	・位置 ・車速 ・車外カメ ラ ・温度 ・加速度 ・ウインカ ー ・ライト
通信機 器	・DoPa	・DoPa ・PHS ・無線 LAN	・DoPa ・無線LAN	・DoPa	・DoPa	・DoPa	・DoPa, CPA, PHS, AirH", 無線 LAN, DSRC	・DoPa ・SWIFT COM	・DoPa ・無線LAN ・AirH"

実証実験

これまでのインターネット自動車プロジェクトおよびインターネット ITS におけるシステム開発においては、実証実験による検証を行ってきた。

表 3.1 にこれまで実施してきた実証実験をまとめる。これまでの実証実験で最大のものとしては、2001 年度の、名古屋・川崎地域での 1640 台での実証実験である。この実験では、プローブ情報システム・車両間コミュニケーションや、位置依存コンテンツ配信システムなどの検証を行なった。

これらの実証実験では、タクシーなどの実験車両を用意し、車載器・通信デバイスの搭載を行ない、その上でアプリケーションの検証を行なっている。

3.3 本研究における開発環境

ここでは、本研究において実現する開発環境について述べる。まず、本研究で想定する自動車環境アプリケーションから見た、開発環境への要求事項をまとめる。その後、既存手法がその要求事項を満しているか整理する。

3.3.1 開発環境に対する要求事項

本研究では、自動車環境アプリケーションの開発を想定し、自動車環境アプリケーション開発環境に必要とされる要求事項をまとめる。

開発環境は、アプリケーション開発を支援するシステムであるため、評価コストが低い必要がある。ここでのコストとは、経済的な費用だけでなく、導入への作業量・利用における必要機材なども含む。

自動車環境アプリケーションでは、自動車環境で発生する様々な情報をもとにして動作する。このため、開発環境では、アプリケーションに対して、これらの情報を提供していく必要がある。これらの情報は、図 2.4 などで述べたように様々な種類の情報を含むため、単一の情報生成機構で生成することは困難である。このため、情報生成機構と開発環境を切離して開発し、開発環境と、外部情報生成機構との連携を可能にする必要がある。またこの連携により、より高性能な情報生成機構が開発された場合に、開発環境全体を更新する必要がなくなる。

開発されたアプリケーションは、様々な環境を走行する自動車上に搭載される。このためアプリケーションの検証も、検証の汎用性を保った形で行なわれるべきである。ある特定の環境での動作検証のみではなく、あらゆる環境での検証が可能である必要がある。

また、アプリケーションは、一つだけで動作しているのではなく、自動車上に複数のアプリケーションが同時に動作している状況が想定される。このため、柔軟な設定ができる必要がある。ある特定のアプリケーションの特定の機能のみを検証するのではなく、複数のアプリケーションの同時実行など、検証対象が柔軟に設定できる必要がある。

さらに、自動車上のアプリケーションは、自動車上で閉じて実行されるのではなく、自動車外部との通信を行なう。通信アプリケーションを開発するには、通信の両端での検証を行なう必要がある。このため、開発環境上だけではなく、実際のアプリケーションサーバなどの、外部機構との通信が可能である必要がある。

下に開発環境への要求事項をまとめる。本研究では、以下の要求事項を満たす開発環境の構築を行なう。

- 評価コストが低い
評価に必要な経済的・人的コストが低い。
- 外部情報生成機構との連携
外部の情報生成機構と協調して動作することが可能。
- 検証の汎用性
開発環境が汎用性を持つ。ある特殊な環境だけではなく、汎用性を持ちあらゆる場所での検証が行なえる。
- 柔軟な設定
アプリケーションのある特定の性質のみを検証するのではなく、より柔軟に複数の検証が行なえる。
- 外部機構との通信
通信は開発環境上のみ閉じられるのではなく、外部のサーバや、他の車両との実際の通信を行なえる。

3.3.2 既存手法の整理

既存の手法が 3.3.1 節で述べた要求事項をどの程度満たすかをまとめる。

表 3.2: 既存技術の比較

	低評価コスト	外部機構との連携	検証の汎用性	柔軟な設定	外部通信
シミュレータ					×
ランダムデータ				×	
実証実験	×	—	×	×	

表 3.2 に既存技術と、要求事項の対応を示す。○は要求を満たしていることを示す。△は改良を加えることで対応可能であることを示す。×は満たすことが不可能な機能を示している。

ns-2 などのシミュレータでは、ノードの移動モデルを高度化・一般化することにより、汎用的な環境を再現することが可能である。また、アプリケーションのみで実現可能であるためコストも低い。外部機構との連携は、想定されていないが、交通流シミュレータとの連携を行なっている研究 [48] などもあるため、拡張により実現可能である。柔軟な設定に関しては、複数のアプリケーションを実装することが可能であるが、各車両ごとに詳細な設定を行なうことなどが想定されていないため △とする。外部通信に関しては全く考慮されていない。

ランダムデータによる開発では、低コストおよび外部通信が実現される。データがランダムであるだけで、通常の通信を行なうため外部との実際の通信も行なえる。汎用性については、データの送信がランダムであるため汎用的ではあるが、現実的とは言えない。データ送信を現実に沿った形で行なえれば汎用的と言える。外部通信との連携も想定はされていない。データの送信には、既存ツールを利用する 경우가多く柔軟ではない。

実証実験は、実際に車両にアプリケーションを搭載して動作させるため、外部および車両間での通信は可能となる。車両から実際にデータを取得できるため、外部機構との連携は不要である。最も実環境に近い非常に重要な手法であるが、コスト・汎用性・柔軟性いずれも実現できない。評価コストは、経済的・時間的にも規模が拡大するにつれ高くなってゆく。汎用性に関しても、汎用的な環境を実際に構築しない限り実現できない。一度、搭載したアプリケーションを途中で変更することなどは困難であるため柔軟性も低い。このように、実証実験は、実環境に最も近い環境での検証が可能であるが、開発環境としての有用性は低い。開発環境での開発および検証をあらかじめ行ない、必要となるアプリケーションおよび各種パラメータを検討付けした後に、実証実験を行なうことにより、実証実験を有効に行なうことが可能になる。

このように既存のアプリケーション開発・検証手法は、自動車環境アプリケーションの開発には不適である。

3.4 本研究のアプローチ

ここでは、本研究でのアプローチについて述べる。先述した要求事項を満たす開発環境として、仮想自動車を用いたアプリケーション・開発環境を提案する。

3.4.1 仮想自動車モデル

本研究では仮想自動車を利用したアプリケーション開発環境を用いて、自動車環境アプリケーションの開発環境とする。仮想自動車による開発環境については、インターネット自動車プロジェクトの活動として研究が行なわれている [49][50]。図 3.2 に現在開発された開発環境の利用例を示す。この例では、タクシー事業者が各タクシーの位置などの情報を管理するアプリケーションが実装されている。

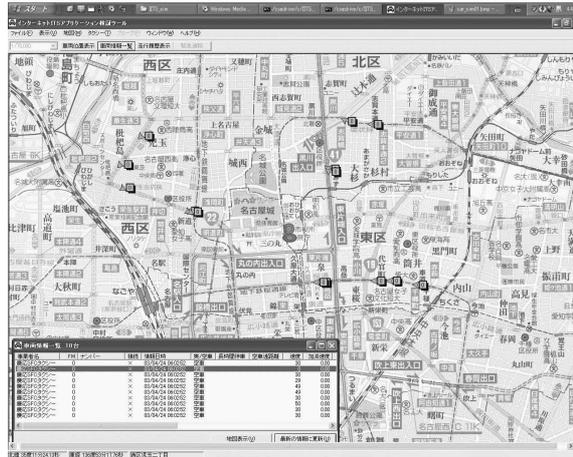


図 3.2: 検証環境利用例

仮想自動車は、インターネット自動車内に設置される車載器を仮想空間上でエミュレートしたものである。図 3.3 に仮想自動車モデルを示す。仮想自動車に対して、交通流シミュレータをはじめ、気象シミュレータなど様々な情報生成機構から、実環境に応じた情報が仮想車両に入力されている。交通流シミュレータからの入力により位置・速度・方向などが設定され、複数の仮想自動車が仮想空間上の地図に沿って走行している環境が再現されている。

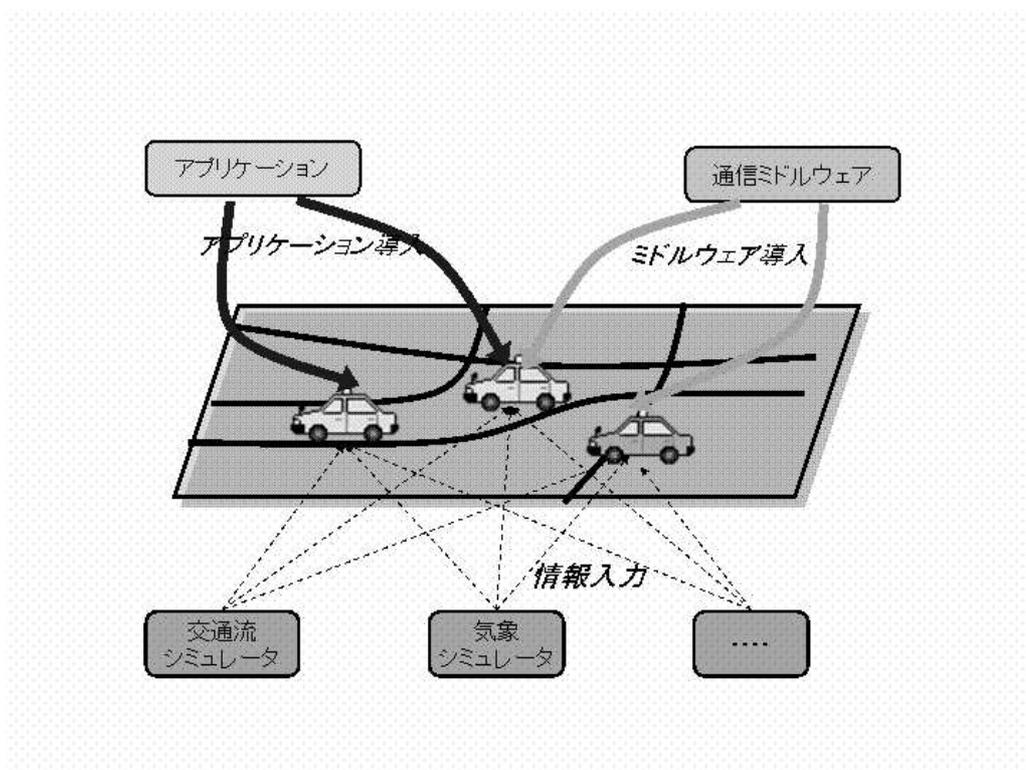


図 3.3: 仮想自動車モデル

表 3.3: 仮想自動車モデルと既存技術の比較

	低評価コスト	外部機構との連携	検証の汎用性	柔軟な設定	外部通信
シミュレータ					×
ランダムデータ				×	
実証実験	×	—	×	×	
仮想自動車モデル					

自動車上のアプリケーションや通信システムの開発者は、仮想自動車上に自ら構築したシステムを導入する。仮想自動車では、前述のように、実際の自動車を持つ情報が再現されているため、それらを利用したアプリケーションを動作させることが可能である。

このように、実際に仮想的な自動車を構築し、その上で実際にアプリケーションを動作させ、開発・検証を行なっていく開発モデルを仮想自動車モデルと呼ぶ。

表 3.3 に、仮想自動車モデルによる開発環境と既存手法との比較を示す。仮想自動車モデルでは、すべての要求事項を満すことが可能である。現在、既存研究 [50] により実装されているものでは、外部通信については、全く考慮されていないが、モデルとしては全て満すことが可能である。

3.4.2 仮想自動車モデルによる開発環境の機能要件

既存の研究により、交通流シミュレータの出力から、走行情報を入力する機構は存在している。しかし、自動車環境アプリケーションの開発には車両情報だけではなく、通信状態の再現も不可欠である。現在は、仮想自動車における通信環境に関しては全く考慮されていない。そのため、外部通信に関しては、要求を満たしているとは言えない。本研究でターゲットとする自動車環境アプリケーションは、通信の状態に応じた動作変更も行なう。そのため、通信環境を再現し仮想自動車に入力する機構が必要となる。

また、外部との通信を行なうためには、実際のインターネットに対する接続を確保する必要がある。各仮想自動車がそれぞれインターネットへの接続を持ち、かつその接続性が、当該仮想自動車の通信環境に応じた接続である必要がある。

また、通信環境の変化に応じたアプリケーションの動作変更を検証するためには、アプリケーションから通信環境に関する情報を取得するインターフェイスを用意する必要がある。

仮想自動車モデルによるアプリケーション開発環境の機能要件をまとめる。以下の 3 つの機能要件を満たす環境を構築していく。

- 仮想自動車における通信の再現
- 仮想自動車への通信情報の入力
- 通信環境情報を提供するインターフェイス

第4章 アプリケーション開発環境の設計

本章では、アプリケーション開発環境の設計について述べる。

4.1 仮想自動車によるアプリケーション開発環境

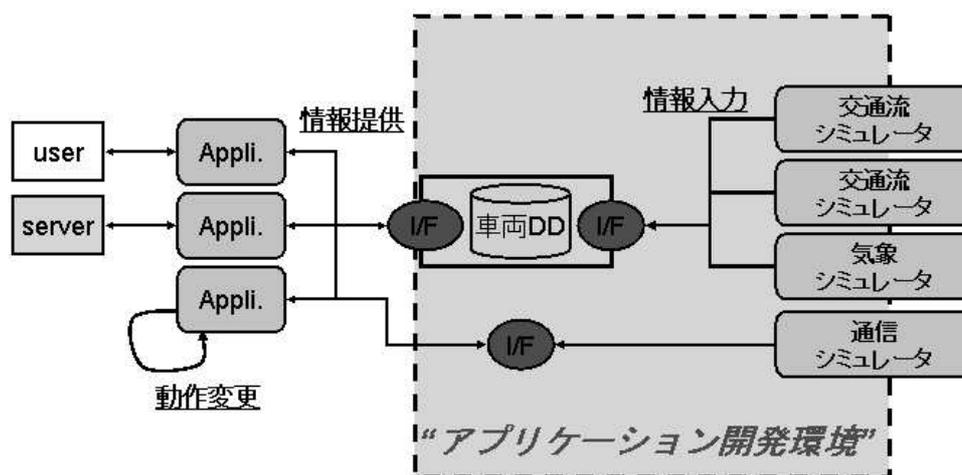


図 4.1: 自動車環境アプリケーション開発環境システムモデル

図 4.1 に、自動車環境アプリケーションの開発環境上での動作を示す。現実の自動車環境アプリケーションでは、GPS などをはじめとするセンサから情報を取得していた。しかし、図 4.1 で示すように、開発環境においては、実際の自動車からの情報は取得できないため、交通流シミュレータをはじめとするシミュレータからの情報を車両データ辞書に入力する。

自動車走行情報に関する再現に関しては、既にインターネット ITS 協会内の活動において開発が行なわれている。本研究では、通信環境を考慮したアプリケーション開発環境を提供するため、自動車の通信環境を開発環境上で再現するシステムを構築する。アプリケーション開発者は、開発したシステムを本研究で構築した開発環境上で動作させることにより、インターネット自動車環境上でのシステム動作の検証・評価を実現することが可能となる。

4.2 アプリケーション開発環境設計概要

本研究では仮想自動車モデルを利用して、通信環境を考慮した自動車環境アプリケーション開発環境を構築する。2.4 節で述べたように、自動車通信環境には様々な要素が影響を与

第 4 章 アプリケーション開発環境の設計

える。本システムでは、それらの要素を入力し、仮想自動車の通信環境を再現し、仮想自動車上での自動車環境アプリケーションの開発を実現する。

以下に本開発環境が満たすべき要求事項を再掲する。

1. 仮想自動車における通信の再現
2. 仮想自動車への通信情報の入力
3. 通信環境情報を提供するインターフェイス

図 4.2 にシステムの概要図を示す。本システムは大別して、仮想自動車部分と通信環境シミュレーション部分に 2 つの部分から構成される。要求事項 2-3 に関しては、仮想自動車内において実現する。要求事項 1 は通信環境シミュレーション部分および、仮想自動車内の通信エミュレーション機能で実現する。

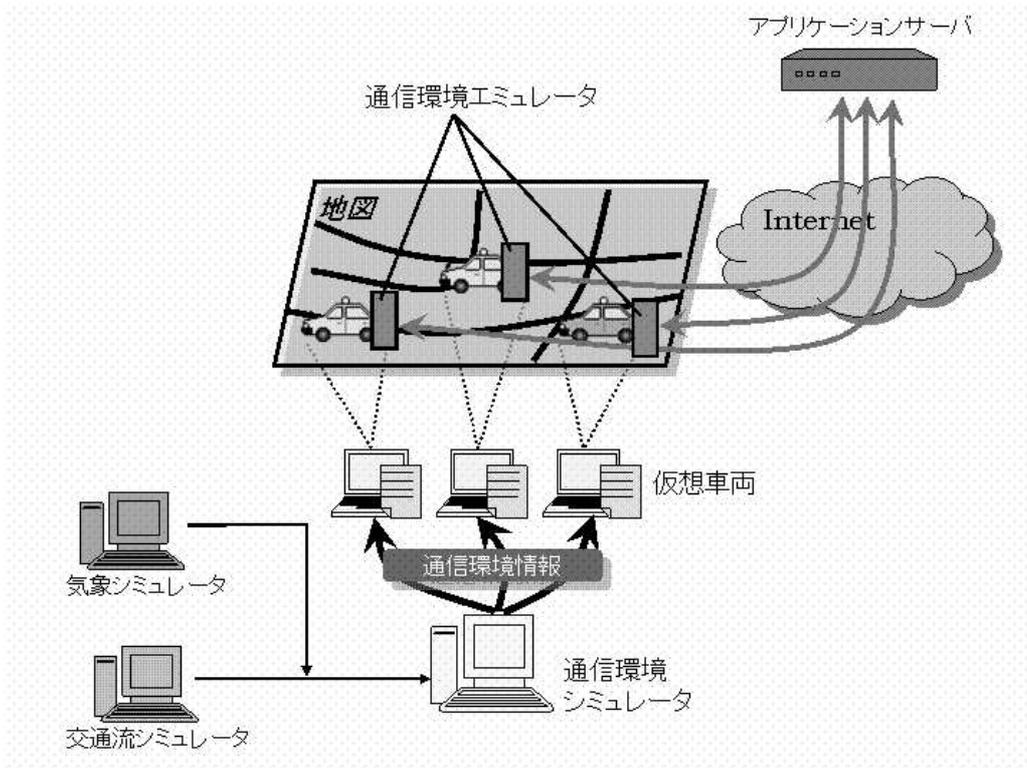


図 4.2: システム概要

通信環境シミュレーション部分では、交通流シミュレータをはじめとした各外部機構から情報を取得し、仮想車両がおかれている環境をシミュレートする。同時に仮想車両部分では、検証対象である自動車環境アプリケーションが動作する。仮想空間全体は、インターネットに接続されており、各仮想車両上のアプリケーションは、仮想環境外部のアプリケーションサーバや、他の仮想車両とインターネットを用いて、通信を行なうことが可能となる。この際の通信品質は、通信環境シミュレータにより出力された通信状態に応じた品質がエミュレートされている。

この時、外部サーバとの通信によりサーバ側の耐故障性の検証も行なう場合には、仮想空間上の時間進行が実環境の時間進行と同じ単位で進む必要がある。そのためには、通信環境

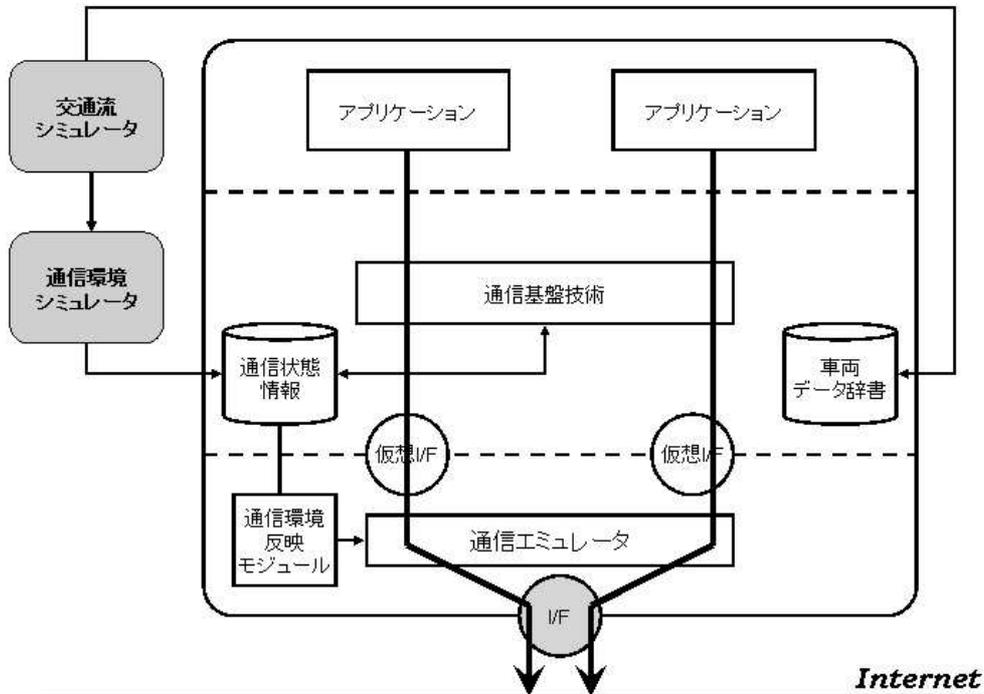


図 4.3: 仮想自動車設計図

のシミュレーションを、実環境での時間進行以内に行なう必要がある。つまり、1秒ごとに通信状態の再現を行なうためには、シミュレーションも1秒以内で終了する必要がある。

4.3 仮想自動車の設計

仮想自動車は、仮想空間上に構築するインターネット自動車の機能を持った機構である。

4.3.1 仮想自動車設計概要

仮想自動車で実現するインターネット自動車の機能としては、アプリケーション動作・状態の情報取得・通信品質に沿った外部通信の3つがある。

仮想自動車上のアプリケーションは、それぞれの手法でトラフィックを発生させる。また、車両データ辞書にアクセスするためのインターフェイスも用意される。アプリケーションの下に通信プラットフォームが位置する。通信プラットフォームは、最下位に位置する各通信デバイスの情報を取得し、各アプリケーションごとに利用デバイスの決定や、Network Mobility機能など移動支援の実行を担う。通信デバイスは1つ以上存在し、それぞれ通信環境シミュレータから出力された通信品質のエミュレーションを行なっている。また、それぞれの通信品質を上位レイヤに対して提供するインターフェイスも持つ。

図 4.3 に仮想自動車の設計図を示す。仮想自動車は、通信環境シミュレータから出力された仮想車両の通信状態を取得する。また、仮想通信デバイスを保持しており、通信環境シミュ

レータから取得した通信状態が仮想通信デバイスでエミュレートされているため、アプリケーションからの通信は状態に沿った通信を再現する。

アプリケーションおよび、通信デバイスはそれぞれ複数存在する可能性がある。このため、仮想自動車上ではアプリケーション・通信デバイスのリストを保持する。また、それらは将来的に新規開発および追加されることが想定されるため、拡張可能な形で実装される必要がある。

現実の自動車では、車両内の車載器に搭載されたアプリケーションや通信プラットフォームは、車載器のアーキテクチャに合った形式で通信デバイスの情報を取得する。UNIX 系の OS であればルーティングソケットなどを利用して取得可能である。通信状態の統一的な情報取得手法として、SNMP(Simple Network Management Protocol)[37] や、MIBSocket などが研究されている。また、自動車全体のデータセットの標準化として自動車データ辞書モデルなどが研究されており、国際標準化が進められている。現在の標準化プロセスにおいては、自動車の通信環境に関するエントリは存在していないが、自動車データ辞書内にエントリを作成することも可能である。自動車データ辞書モデルによる利点としては、情報の取得主体が車内であっても車外であっても統一的に利用できる点が挙げられる。

各仮想車両はそれぞれ搭載する通信デバイスごとに仮想通信インターフェイスを保持しており、仮想車両上のアプリケーションや、通信プラットフォームはその仮想通信インターフェイスを、通信インターフェイスとして利用する。仮想通信インターフェイスを利用した通信は、実際の通信インターフェイスを利用してインターネット上の外部機構との通信を行なうことが可能である。

4.3.2 アプリケーションの導入手法

アプリケーション開発者が開発したアプリケーションを開発環境上に導入する手法が必要となる。

インターネットでのアプリケーションは、車内アプリケーションと車外アプリケーションに分割できる。この内、車外アプリケーションに関しては、仮想自動車との外部通信が可能になっているため、通常アプリケーション開発と全く同様に行なうことが可能である。通信の対象を仮想自動車に設定するだけである。

車内アプリケーションの仮想自動車への導入に関しても、通常の開発と可能な限り同様に行なえるべきである。しかし、開発環境上で動作させるため、ある程度の制限が加えられる。通常のインターネットアプリケーションは、通信を開始する際に通信相手のみを指定する。しかし開発環境では、1つの開発環境上に複数の仮想自動車が存在するため、アプリケーションが仮想自動車上の仮想通信デバイスを指定する必要がある。

図 4.4 にアプリケーションと仮想通信デバイスの対応のモデルを示す。仮想通信デバイスは、実際のインターフェイスにバインドされて作成される。各アプリケーションは、開発環境上にある複数の仮想インターフェイスから、自らが搭載されている仮想自動車を持つ仮想通信デバイスを選択して利用する必要がある。この機構は、各アプリケーション開発者がアプリケーション開発時に配慮する必要がある。本研究における実装で利用した JAVA では、ソケットを作成する際に、送信元の IP アドレスとポート番号を指定することが可能である。

また、実際の利用環境においては、アプリケーションの動作はユーザによって行なわれることが多い。しかし、開発環境では1台ごとにユーザからの入力を持つことはできないため、

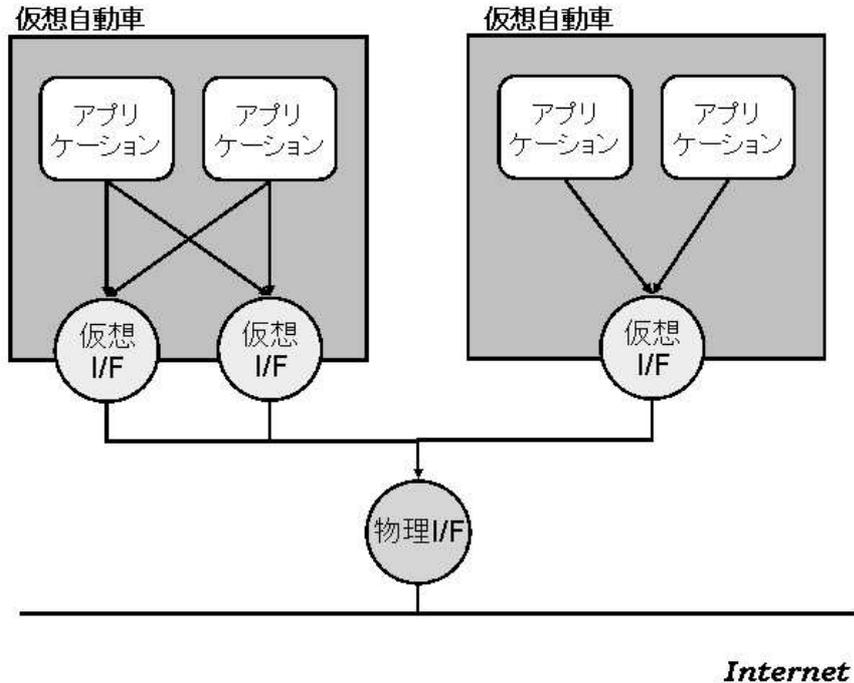


図 4.4: アプリケーションと仮想通信デバイスの対応

各アプリケーションが動作のスケジューリングを行なっておく必要がある。

本開発環境では、アプリケーション開発者に対して、仮想インターフェイスとのバインド、およびスケジューリングを行なうためのインターフェイスを提供する。仮想インターフェイスとのバインドについては、TCP でのソケット作成用の関数を定義し、通信先の IP アドレス/ポート番号に加えて、送信元の仮想自動車の IP アドレス/ポート番号の入力をもとめるインターフェイスを提供する。UDP に関しては送信元の IP アドレス/ポート番号の入力を求めるインターフェイスを提供する。また、アプリケーション動作のスケジューリングについては、仮想自動車側にアプリケーション動作を一括してスケジューリングする機能を持たせ、そこへの登録インターフェイスを提供する。

4.3.3 通信エミュレーション機能設計

本開発環境はある計算機上に存在している。開発環境と実際のインターネットは、その計算機のインターネット接続を利用して接続される。各仮想自動車上のアプリケーションは、仮想通信デバイスを通して、インターネットへの接続を行なうことになる。計算機のインターネット接続は、仮想空間上の接続とは異なる性質を持つため、仮想空間における通信品質をエミュレートする必要がある。

通信エミュレーションは、通信環境シミュレータからの出力をもとに、仮想インターフェイスに対して、品質を反映させるフィルタを追加することにより実現する。フィルタは、仮想的な通信品質をエミュレートする機能を持つ。現在、そのような機能を持つフィルタとしては、FreeBSD 上に実装されている DUMMYNET[51] や、Linux 上で実装されている tc な

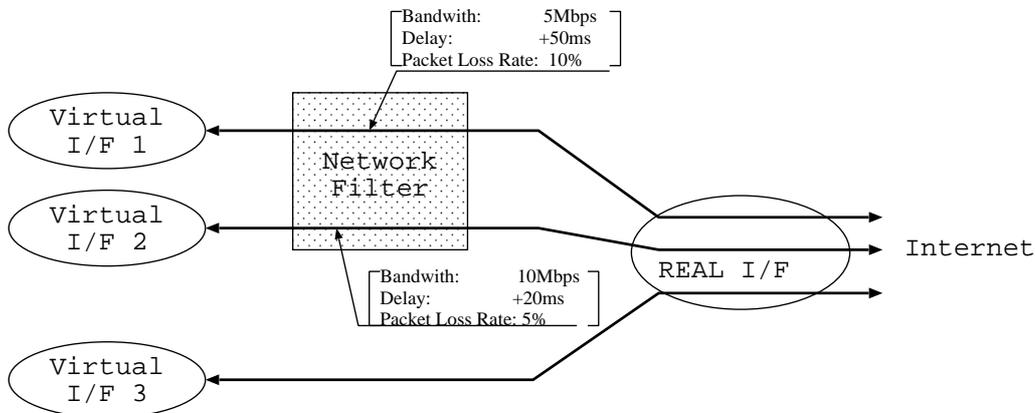


図 4.5: 通信フィルタ動作概要

どが存在している。たとえば、DUMMYNET では帯域幅・パケットロス率・通信遅延の 3 項目をフィルタとして設定可能である。

図 4.5 に通信フィルタの動作概要を示す。仮想インターフェイス (Virtual I/F)1 および 2 から送信されるデータは、それぞれフィルタを通過する際に、設定された通信品質にエミュレートされているため、実際の物理インターフェイス (REAL I/F) を通過する時点で、エミュレートされた通信を行なう。仮想インターフェイス 1・2 にはそれぞれ違った品質が適用されている。図では、DUMMYNET を想定しているため、帯域幅・パケットロス率・通信遅延の 3 項目がインターネット上までで変化している。仮想インターフェイス 3 からのデータはフィルタを通過していないため、実際の物理インターフェイスと同じ品質の通信を行なう。インターネット上から仮想自動車に対するインバウンド通信も同様に变化する。

このように、本研究では 1 つの物理インターフェイスに対して複数の仮想インターフェイスが割り当てられる。このため、全て仮想インターフェイスの合計利用可能帯域が、物理インターフェイスの性能を上回った場合には、通信環境の再現が行えない。たとえば、100Mbps の物理インターフェイスを搭載するマシン上で、10Mbps の通信帯域を持った仮想インターフェイスを 11 個以上設定すると、10Mbps の性能を発揮することができない。このため、このようなケースでは、本開発環境を動作させるマシンが、複数の物理インターフェイスやより高速な物理インターフェイスを備える必要がある。

各仮想自動車の搭載する通信デバイスの性質は、それぞれ異なっているため、通信デバイスごとに異なる品質をエミュレートする必要がある。各仮想デバイスが持つ IP アドレスごとに、フィルタを既述することにより、仮想デバイスごとにそれぞれ異なる通信品質を再現することが可能となる。また、フィルタに対する設定を行なうだけでなく、アプリケーションに対して、情報取得を提供するインターフェイスへの入力も行なう。

4.4 通信環境シミュレータの設計

ここでは、通信環境シミュレータ部分の設計を行なう。まず、シミュレータへの入力および出力項目をまとめる。その後、シミュレータの設計について述べる。

4.4.1 通信環境シミュレータへの入力

インターネット自動車の通信環境を再現するためのシミュレータへの入力およびシミュレータからの出力をまとめる。シミュレータへの入力としては、表 2.4 で挙げた通信環境に影響を与える要素群の入力を行なう。これらの各要素を入力として、現在の自動車通信環境の計算を行なう。シミュレータからの出力としては、各自動車の通信環境をそれぞれ出力する。

表 4.1: シミュレータへの入力

分類	要素	単位
自動車通信情報	移動支援プロトコル	[プロトコル名]
	発生トラフィック量	bps
	通信プロトコル	TCP/UDP
	利用通信デバイス	[デバイス名]
走行情報	位置情報	lng,lat(degree),alt(m)
	速度情報	km/h
	方向	0-15
基地局情報	基地局位置	lng,lat(degree),alt(m)
道路周辺情報	気象情報	未定
	工事・事故等のイベント	未定

表 4.1 にシミュレータへの入力をまとめる。この時、自動車通信情報およびに関しては、仮想自動車からのフィードバックを受ける。また、表 2.4 と比較して、基地局利用度合などが記載されていないが、基地局利用度合に関しては、仮想自動車からのフィードバックをもとにシミュレータ内部で計算を行なうため入力項目としては記載していない。また、道路周辺情報に関しては、通信プラットフォーム・アプリケーションでの利用がまだ想定されていないため未定として、本論文では実装を行なわないが、将来的には機能の追加が想定されるため、拡張性を考慮した構築を行なう。

走行情報については、交通流シミュレータから入力を受ける。交通流シミュレーションの手法は、大きくマクロシミュレーションとミクロシミュレーションに分類できる [52]。マクロシミュレーションでは、交通の流れを連続的な流体として捉える。このため、ある地域全体での交通環境の検証が可能である。一方、ミクロシミュレーションでは、1 台の車両もしくは数台の車両単位での交通の流れを扱うため、各車両/車群の移動を検証することが可能である。通信環境シミュレータでは、各仮想自動車の移動を取得する必要があるため、ミクロシミュレータを利用して、移動情報の取得を行なう。

実際の計算手法は、各要素ごとに異なってくる。そのため、実装段階においては各入力要素ごとに計算機能をモジュール化し、追加・編集を容易なものとする。これにより、新たな影響要素の追加や、影響の計算手法の改善がモジュールを組替えることで可能となる。

4.4.2 通信環境シミュレータからの出力

出力に関しては、自動車通信環境に関わる情報の出力を行なう。各自動車の持つ無線通信デバイスの通信品質の情報としては、自動車-アクセスポイント間の通信遅延・利用可能帯域・

パケット損失確率などがある。また、アプリケーションによっては、通信品質以外の情報も必要とする。たとえば、車車間でのアドホックネットワークを構築し、車両間での通信を行なうアプリケーションの場合、車両間での通信が可能な範囲内に何台の車両が存在しているか、などの情報を必要とする。要求される出力に応じて、入力要素も変化する。入力・出力ともにモジュール化を行ない、利用者の要求に応じて追加・改変などを可能にしていく必要がある。

表 4.2: 通信環境シミュレータからの出力

分類	要素	単位
通信品質情報	利用可能帯域	bps
	通信遅延	msec
	パケットロス率	%
周辺情報情報	周辺車両台数	台

また、通信品質以外の情報としては、周辺に存在する車両の台数なども含まれる。周辺車両数は車両間アドホック通信の検証などに用いられる。これらの情報をどのように仮想車両に対して提供するかは、様々な手法が考えられる。仮想車両は現実の車両と同様のシステムを備えるため、実際の情報取得と同様の手法を提供する必要がある。これらの情報取得に関しては本研究ではこれ以上触れない。関連研究としては、自動車に関する情報を扱うデータ辞書モデルが提案されている。

4.4.3 設計概要

図 4.6 に通信環境シミュレータの設計図を示す。

通信環境シミュレータでは、4.4.1–4.4.2 節で述べた入出力を行なう。自動車通信情報に関しては、仮想自動車からのフィードバックとして入力を得る。

位置情報および速度情報といった走行情報は、4.4.1 節で述べたようにミクロ交通流シミュレータから取得する。

基地局情報に関しては、実際の基地局設置情報を取得し、それをもとに入力を行なう。PDC の設置情報であれば、各キャリアがサービスエリアなどを公開している [53][54][55] ため、これらの情報が利用可能である。また、今後道路向けの無線 LAN エリアが設置されていくことが予測されるが、その場合も同様に各キャリアなどの、無線 LAN ISP が公開するサービスエリア情報をもとに入力を行なうことが可能である。道路向け無線 LAN エリアの設置に関しては、インターネット ITS 協議会ロードサイド店舗分科会でも議論が行なわれている。ロードサイド店舗分科会では、道路に面した店舗を持つコンビニエンスストア・ガソリンスタンド・駐車場会社などに、無線 LAN エリア設置の協力を求めている。

これらの情報の精度に関しては、本研究では深く取り扱わない。より高精度な情報を取得し、本開発環境に入力することで、より精度の高い通信環境の再現が可能になる。本研究においては、そのような情報の入力を想定した設計を行なう。

シミュレータでは、入力された情報をもとに通信品質の状態および周辺情報を生成する。生成した情報は仮想自動車に入力され、仮想自動車で通信環境がエミュレートされる。仮想自動車上のアプリケーションの通信状態は、シミュレータにフィードバックされる。

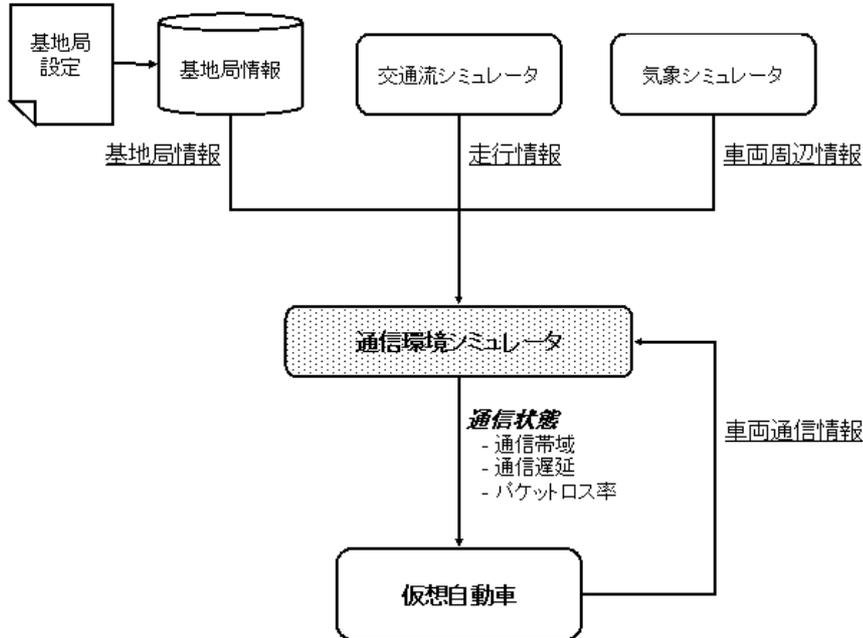


図 4.6: 通信環境シミュレータ設計図

4.5 シミュレーション手法

4.4 節では、通信環境シミュレータへの入出力および全体構成について述べた。ここでは、出力する情報をどのように算出するかについて述べる。

4.5.1 通信品質情報の算出

通信品質情報は、自動車の位置・基地局の位置・周辺車両の通信状態・移動支援プロトコルの有無から計算される。図 4.7 に通信品質の算出モデルを示す。

本システムでは、各通信インフラごとに通信品質を地理位置情報にマップさせたものを保持している。実装するデバイスごとに 1 枚のマップを持ち、それらが重ねられている。仮想自動車が地図上を走行するに伴い、搭載する無線デバイスに対応する各マップから品質情報を取得する。各通信インフラの通信品質マップは、あらかじめ作成して保持する。

無線通信デバイスごとの通信品質は、ns-2 や GloMoSim のような既存シミュレータでも算出可能である。仮想自動車と基地局の位置関係もとに、既存シミュレータを利用して通信遅延などの計算が行える。これらのシミュレータでは、無線の MAC レイヤからトランスポートレイヤまでのシミュレーションを行なっているため、正確な性能が計算可能である。しかし、逆にシミュレーション時間は膨大なものになる。特に大量のノードが同時に移動する環境のシミュレートを行なう場合、既存のシミュレータでは、計算に実際の時間以上の時間を必要とする。このため、既存シミュレータをそのまま本システムで利用することは出来ない。

本システムでは、通信デバイスごとの通信品質マップを、あらかじめ既存シミュレータを

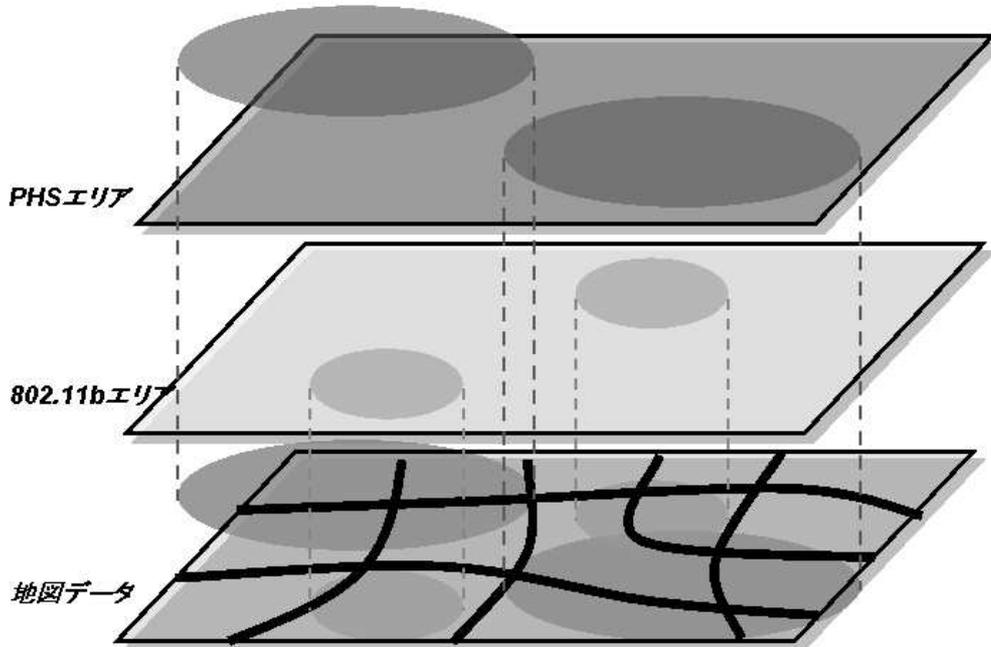


図 4.7: 通信品質算出モデル

利用して作成する。発生しうるシチュエーションを抜粋し、そのシチュエーションでの通信品質の計算をあらかじめ既存シミュレータで行ない、品質データをあらかじめ保存する。本システム上で各シチュエーションが発生した際には、実際にシミュレーションを行なうのではなく、保存されたデータを読み込み通信品質を出力する。

4.5.2 周辺情報の出力

通信環境シミュレータでは主に、通信デバイスの品質に関わる出力を行なう。しかし、周辺車両の情報も通信に影響を与える。特に車車間での通信を行なう場合には、周辺車両の情報が必要となる。

車車間通信は主に 2 つの用途で利用されることが想定される。1 つは、インターネットのバックボーンを利用しない車両間コミュニケーションのため、2 つ目は、無線インフラの電波到達外にいる際に、他の車両を経由してインターネットへの通信を行なう場合に利用される。

このように車車間通信では、通信先まで複数の自動車を経由して到達するマルチホップ通信が必要となる。このため、無線でのマルチホップ通信を実現するプロトコルとして、主に MANET を利用したルーティングが行なわれる。MANET では、様々な種類のルーティングプロトコルが存在するが、全てのプロトコルは近隣車両の探索を、無線伝播範囲内へのブロードキャストにより行なうことが基礎となっている。しかし、本システムでは、4.5.1 節で述べたように、無線伝播に関するシミュレーションは行ないない。無線到達範囲内に存在する車両の台数、および IP アドレスを、このため周辺情報として提供する必要がある。この機能は、車両間通信用無線通信デバイスの到達範囲と、車両の位置情報を用いて算出する。ま

た、通信車両間で互いに同じルーティングプロトコルを利用する必要がある。近隣の車両がどのルーティングプロトコルを利用しているかの判断は各プロトコルに依存する。

第5章 アプリケーション開発環境の実装

本章では、インターネット自動車における通信環境再現システムの構築手法について述べる。実装環境について述べ、その後、構築の前段階として、構築に必要な要素の決定を行なう。最後に実際の実装について述べる。

5.1 実装環境

本研究では、インターネット自動車通信環境再現システムの実装を行なう。表 5.1 に実装環境を示す。

表 5.1: 実装環境

OS	FreeBSD 5.1R
CPU	PentiumIII 1.06GHz
Memory	192MB
JDK	Java(TM) 2 Runtime Environment SE.1.4.2_02-b03

開発環境上では、仮想空間上で複数の仮想自動車を独立して動作させる。このため本システムでは、1台の仮想自動車を Java の VM(Virtual Machine) 上で動作させる。また、通信品質のエミュレーションには、通信帯域・通信遅延・パケットロス率を再現できる DUMMYNET を利用する。DUMMYNET は FreeBSD 上で実装されているため、FreeBSD を利用する。

5.2 システム構築要素の決定

ここでは、あらかじめ決定すべき要素について述べる。無線通信の品質測定および、仮想自動車・無線基地局の初期パラメータの初期設定を行なう。

5.2.1 無線通信品質の測定

本研究においては、無線通信の品質をあらかじめ実験を行ない測定する。図 5.1 に 802.11b から PDC への切替えを行なった際の帯域幅の変化を示す。この測定で得た結果をもとに無線通信デバイスの品質を決定する。

ここでは、802.11b の基地局における性能を実際に測定し、入力を行なった。実測値の入力以外にも、無線通信デバイスの通信品質シミュレータを利用し、シミュレーション結果を入力することも可能である。

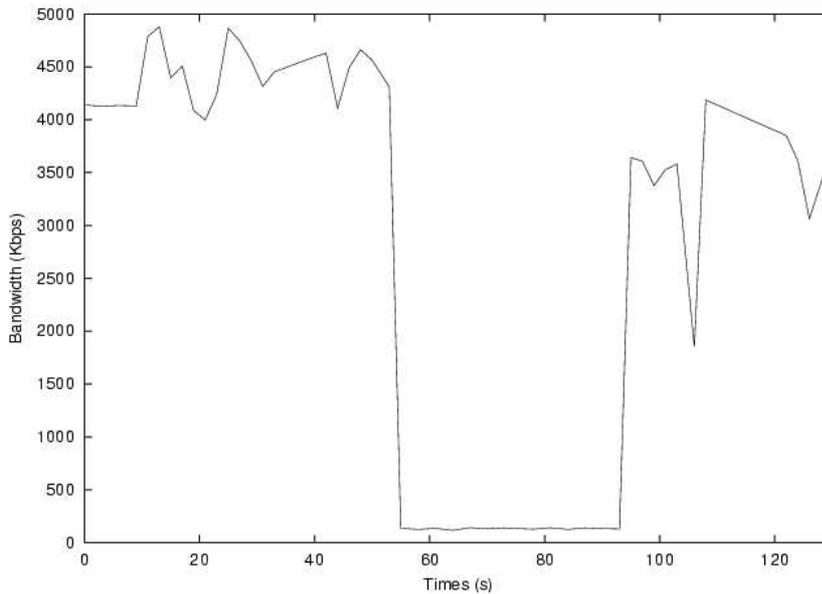


図 5.1: 802.11b/PDC での帯域幅の変化

5.2.2 初期設定パラメータ設定

シミュレータへの入力として、あらかじめ固定すべき項目の設定を行なう。設定項目としては仮想自動車パラメータと無線基地局設置がある。

仮想自動車パラメータ

図 5.2 に仮想自動車に関する初期設定ファイルを示す。

仮想自動車の走行に関する設定は、SOUND の設定ファイルにおいて行なわれる。SOUND では、OD 情報 (出発地・目的地)・加減速の程度・車両 ID を設定する。開発環境での設定では、車両 ID (VID)・HomeAddress・通信デバイス・搭載アプリケーションに関する設定を行なう。開発環境で設定した車両 ID と、SOUND の車両 ID を照合して、走行情報の取得を行なうため、両者での車両 ID が合致する必要がある。

また、仮想自動車に搭載される通信デバイスについては、デバイスの種類・そして IP アドレスの設定を行なう。初期設定で設定する IP アドレスとは、移動支援プロトコルで利用する Home Address であり、常にこのアドレスで通信が行なえるアドレスである。通信デバイスは 1 個以上で自由に設定できる。

搭載アプリケーションの設定は、APPLICATION タグで囲みアプリケーション名を記載する。アプリケーションごとの詳細な設定は、各アプリケーションごとに設定ファイルを用意して行なうこととし、開発環境自体の設定と独立させる。

現状の設定ファイルでは、各自動車ごとに設定を行なう必要がある。これにより、個々の自動車に対して柔軟な設定が可能である反面、大量の仮想自動車の設定を行なう場合には煩雑でもある。自動車における通信デバイスの搭載確率などが判明すれば、確率の入力により自動的に自動車の設定を行なうインターフェイスなどが実現できる。

```
<?xml version="1.0" ?>
<VVehicle>
  <VID id="1">
    <HOMEADDRESS>2003:500:1:ffff::1</HOMEADDRESS>
    <COM_IF id="1">
      <TYPE>802.11b</TYPE>
      <IPADDR>2002:200:0:8899::1</IPADDR>
    </COM_IF>
    <COM_IF id="2">
      <TYPE>PDC</TYPE>
      <IPADDR>2002:200:0:8888::5</IPADDR>
    </COM_IF>
    <COM_IF id="3">
      <TYPE>PHS</TYPE>
      <IPADDR>2002:200:0:9999::10</IPADDR>
    </COM_IF>
    <APPLICATION id="1">
      <NAME>ProbeInfoSender</NAME>
    </APPLICATION>
    <APPLICATION id="2">
      <NAME>MusicDownloader</NAME>
    </APPLICATION>
  </VID>
</VVehicle>
```

図 5.2: 仮想自動車初期設定ファイル

無線基地局設置

設計で述べたように、本システムでは各無線通信デバイスごとに地図上に依存して、通信品質を保持する。図 5.3 に無線 LAN 基地局に関する初期設定ファイルを示す。

本実装では、802.11b および AirHTM の実装を行なう。設定項目としては、基地局の位置を緯度経度高度で指定する。緯度経度高度は、度の単位で小数点以下 7 桁までを入力する。0.0000001 °の差を距離にすると、緯度方向・経度方向ともに約 1cm である。基地局の設置としては、十分な精度を持つ。

また無線 LAN の特性項目として、電波到達範囲を指定する。無線 LAN 以外の携帯電話および PHS についての入力省略する。また、この基地局に接続することにより、取得可能な IP アドレスのサブネットを記載する。このサブネットの変化を検知して、移動支援プロトコルの動作を行なう。この設定ファイルをもとに、通信デバイスごとの地理位置情報と通信品質の対応を生成できる。

基地局の設置に関しては、現実の設置を反映させて方針を決定していく必要がある。現在では、交差点の緯度経度を取得し、そこに配置している。しかし現実には、路面に店舗をもつガソリンスタンドや駐車場などに設置されることが想定されているため、それらの位置を取得し設置していくことになると予測される。

```

<?xml version="1.0" ?>
<AIRSTATION>
  <APID id="1">
    <SPEC>
      <TYPE>802.11b</TYPE>
    </SPEC>
    <POSITION>
      <LNG>136.8927000</LNG>
      <LAT>35.1693000</LAT>
      <ALT>0</ALT>
    </POSITION>
    <WAVE_INFO>
      <LENGTH>200m</LENGTH>
      <FORM>CIRCLE</FORM>
    </WAVE_INFO>
    <IPSUBNET>
      <NETADDR>2003:300:0:8800::/64</NETADDR>
    </IPSUBNET>
  </APID>
</AIRSTATION>

```

図 5.3: 802.11b 基地局初期設定ファイル

5.3 システム構築

ここでは実際のシステム構築手法について述べる。仮想自動車、通信環境シミュレータおよび通信エミュレーション機構それぞれについて述べる。

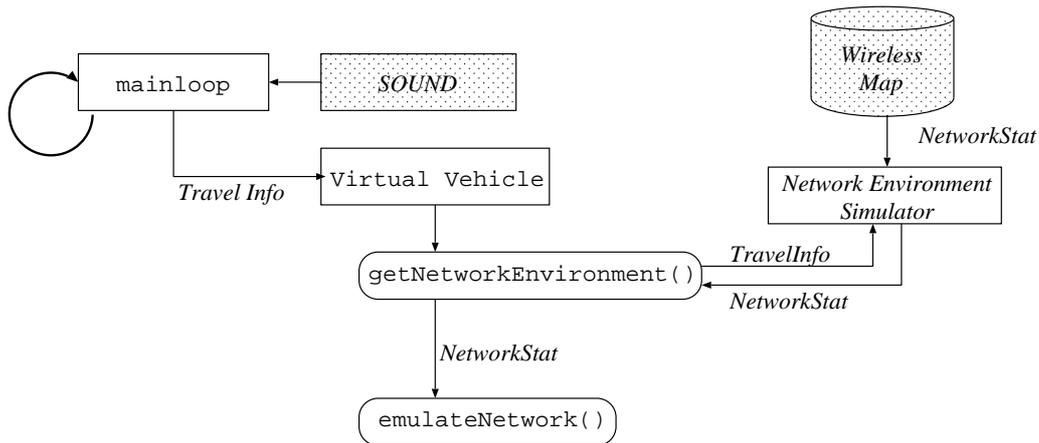


図 5.4: 実装概要

図 5.4 に実装の全体像を示す。mainloop の 1 処理ごとに SOUND から得た走行情報 (travel info) を仮想自動車 (VirtualVehicle) に対して出力する。仮想自動車では、その走行情報をもとに通信環境シミュレータ (Network Environment Simulator) から通信環境の情報 (NetworkStat) を取得し、実際に通信のエミュレーションを行なう。以下に仮想自動車と、通信環境シミュレータまた、通信エミュレーションそれぞれの実装について示す。

5.3.1 仮想自動車

仮想自動車部分の実装手法について述べる。図 5.2 に示した仮想自動車初期設定ファイルをもとに図 5.5 に示す VirtualVehicle クラス作成する。

```
public class VirtualVehicle{
    private double lat;
    private double lng;
    private double alt;
    private double vel;
    private Vector ComDeviceList; /* IPNetDevice クラスの配列 */
    private Vector InetAppList; /* InetApplication クラスの配列 */
    private IcarDD icarDD; /* 車両データ辞書クラス */
    private HashMap AppliSchedule; /* アプリケーションスケジュール*/
    private Vecotr CarNearBy /* 周辺車両リスト */

    private void getNetworkEnvironment(){
        /* 現在の通信品質を取得 */
    }
    private void emulateNetwork(){
        /* 現在の通信品質をエミュレート */
    }
}
```

図 5.5: クラス: VirtualVehicle

VirtualVehicle クラスのメンバ変数として、仮想車両に関わる全ての情報を保持する。SOUND から出力される走行情報および、後述する NetworkStat クラスにより示される通信環境の情報を持つ。また、初期設定ファイルの情報をもとに IPNetDevice クラスおよび InetApplication クラスの配列を持つ。この 2 つのクラスはそれぞれ搭載する通信デバイスおよびアプリケーションの情報を持つ。NetworkStat クラス内の通信品質に関わる情報は、対応する IPNetDevice クラス内に格納される。CarNearBy リストには、周辺に存在する車両の ID が格納される。SOUND から全車両の位置情報を取得し、それぞれとの距離の計算を行ない、車車間通信用デバイスの電波到達範囲内に存在する車両の ID が格納される。

また InetApplication クラスのオブジェクトは実際にデータ送受信を行なう。AppliSchedule マップには、各アプリケーションのスケジュールを設定する。InetAppList 内に格納された各アプリケーションは、AppliSchedule マップ内のスケジュールに沿って実行される。

VirtualVehicles クラスのメンバ関数としては、getNetworkEnvironment() と emulateNetwork() が重要である。getNetworkEnvironment() では、SOUND からの位置情報をもとに通信状態を NetworkStatus クラスとして取得し、ComDeviceList 内の IPNetDevice クラスに格納する。emulateNetwork() では、その位置情報をもとに算出された NetworkStat クラスの情報を仮想インターフェイスに設定しエミュレートする。エミュレーションに関しては、5.3.2 節で詳述する。

開発環境の初期設定メソッドで、VirtualVehicle クラスのリストである VirtualVehiclesList クラスを作成する。図 5.6 に VirtualVehiclesList クラスの概要を示す。四角で示されるエンティティはクラス、円で囲まれるエンティティはメソッド、円柱はリストを示す。

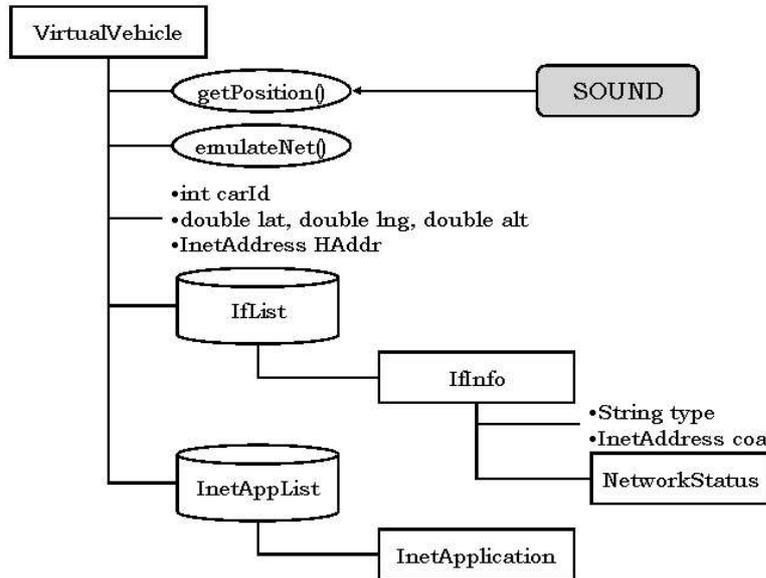


図 5.6: VirtualVehiclesList クラス

NetworkStat クラスの詳細を述べる。表 5.2 に NetworkStat クラスの属性を示す。本クラスは各通信デバイスごとに作成される。ネットワークを抽象化したパラメータとして、利用可能帯域・配送遅延・パケットロス率を持つ。また、各通信デバイスごとに取得した InetAddress クラスも保持する。本研究では NetworkMobility 技術を利用するため、常に一意に利用できる Home Address と、移動先で取得した Care of Address の 2 つの InetAddress を持つ。Home Address は 64bit のプレフィクス長で保持され、このアドレスを Home Prefix[56] として内部ネットワークに広告する。Home Address は複数のデバイスで共有する場合もある。

表 5.2: NetworkStat クラスの属性

フィールド名	型名	説明
bandwidth	int	利用可能帯域
delay	int	配送遅延
packetloss	int	パケットロス率
HomeAddress	InetAddress	Home Address
CoAddress	InetAddress	Care of Address

5.3.2 通信エミュレーション

ここでは、仮想自動車内で行なわれる通信エミュレーションの実現手法について述べる。通信エミュレーションは FreeBSD 上で実装されている DUMMYNET を利用して実現する。DUMMYNET では通信帯域・通信遅延・パケットロス率の設定が可能である。他の tc などのフィルタでは、パケットロスの再現が出来ないため、DUMMYNET を利用する。

本システムでは 1 つの計算機上で複数の仮想自動車を実現しており、またそれぞれの仮想車両では 1 つ以上の通信デバイスを持つことが可能である。このため、システムが動作する計算機上では、複数の仮想デバイスが存在することになる。DUMMYNET では各仮想通信デバイスごとに通信品質を設定する必要がある。

開発環境の初期処理時に、全ての仮想通信デバイスを作成し、物理デバイスとの対応付けを行なう。FreeBSD 上では alias アドレスを付与することにより、仮想デバイスの作成を行なう。各仮想自動車上のアプリケーションは、この alias アドレスを利用して通信を行なう。

DUMMYNET は FreeBSD 上のファイアウォールである ipfw のモジュールとして、実装されている。このため、各仮想通信デバイスが持つ IP アドレスをソースとして、ファイアウォールを設定することにより、仮想デバイスごとのエミュレーションを実現する。図 5.7 に ipfw、図 5.8 に DUMMYNET の設定例を示す。図 5.7 では、ipfw の初期設定を表示している。この設定では、192.168.100.120 への通信を 20 の DUMMYNET を利用するよう設定している。以下 10 台に対して同様に設定している。

```
% ipfw list
00020 pipe 20 ip from any to 192.168.100.120
00020 pipe 20 ip from 192.168.100.120 to any
00021 pipe 21 ip from any to 192.168.100.121
00021 pipe 21 ip from 192.168.100.121 to any
00022 pipe 22 ip from any to 192.168.100.122
00022 pipe 22 ip from 192.168.100.122 to any
00023 pipe 23 ip from any to 192.168.100.123
00023 pipe 23 ip from 192.168.100.123 to any
00024 pipe 24 ip from any to 192.168.100.124
00024 pipe 24 ip from 192.168.100.124 to any
00025 pipe 25 ip from any to 192.168.100.125
00025 pipe 25 ip from 192.168.100.125 to any
00026 pipe 26 ip from any to 192.168.100.126
00026 pipe 26 ip from 192.168.100.126 to any
00027 pipe 27 ip from any to 192.168.100.127
00027 pipe 27 ip from 192.168.100.127 to any
00028 pipe 28 ip from any to 192.168.100.128
00028 pipe 28 ip from 192.168.100.128 to any
00029 pipe 29 ip from any to 192.168.100.129
00029 pipe 29 ip from 192.168.100.129 to any
```

図 5.7: ipfw コマンド設定例

図 5.8 では、20 の DUMMYNET に対して、通信帯域 11Mbps で通信遅延 10ms、パケットロス率が 20% という品質を設定している。これにより、図 5.8 で設定されている、192.168.100.120 のアドレスからの通信がエミュレートされる。同様に各アドレス、つまり仮想通信デバイスごとに設定を行なう。

VirtualVehicle クラス内の、emulateNetwork() メソッドでは、Runtime クラスを利用して ipfw を呼び出す。また、ipfw の操作には root 権限が必要であるため、本システム全体の実行も root 権限を持ったユーザで行なう必要がある。

```
% ipfw pipe 20 conf bw 11264Kbit/s delay 10 plr 0.2
```

図 5.8: DUMMYNET コマンド設定例

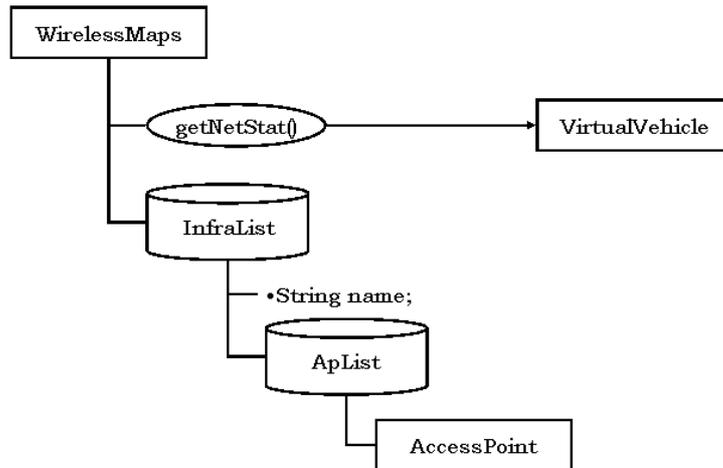


図 5.9: WirelessMap クラス

5.3.3 通信環境シミュレータ

通信環境シミュレータでは、環境要素の入力をもとに、5.2.1 節で示した無線品質の算出を行なう。具体的には、NetworkStat クラスを算出し、VirtualVehicle クラスに格納する。

実装では、交通流シミュレータ SOUND からの走行情報の出力を受け、通信環境のシミュレーションを実行する。SOUND は、本来車群を 1 つの要素としたマイクロシミュレーションを行なうが、本研究では 1 台の仮想自動車を再現するため、設定により、1 車群を 1 台で構成して SOUND の実行する。

通信環境シミュレータでは、図 5.3 に示した無線基地局初期設定ファイルを読み、無線基地局の生成を行なう。図 5.9 に WirelessMap クラスの概要を示す。WirelessMap クラスでは、全ての無線基地局の情報を、その種類ごとに保持する。WirelessMap クラスの主要メンバ関数としては、getNetStat() がある。getNetStat では、VirtualVehicle クラスを引数にとり、それに NetworkStat クラスを格納して返す。getNetStat は、WirelessMap クラス内に格納されている各無線基地局のリストから、NetworkStat クラスを算出する。実際の算出は以下の AccessPoint クラス内の getStat() メソッドで行なわれる。

図 5.10 に示す AccessPoint クラスが、各無線基地局を示す抽象クラスである。あらゆる種類の基地局は AccessPoint 抽象クラスを継承して作成される。各基地局クラスは、緯度経度高度で示す位置情報と、基地局種別を入力する。基地局種別は String 型で表現され、自動車側の搭載デバイス名と同じ命名を行なう。本実装では、利用する 3 つのデバイスを、"80211b",

“AIRH”, “PDC” と命名した。また、車両の位置情報をもとに通信環境を返す `getStat` メソッドおよび、IPv6 ネットワークアドレスを設定する `setNetwork` メソッドをオーバーライドする必要がある。基地局情報は `AccessPoint` クラスにより抽象化されているため、新しい基地局の導入が容易になる。

```
abstract class AccessPoint{
    private double lat;
    private double lng;
    private double alt;
    private String type;
    public void setPosition(double aplat, double aplng, double apalt){ ; }
    public void setType(String aptype){ ; }
    abstract public NetworkStatus getStat(double lat,double lng,double lat);
    abstract public void setNetwork(InetAddress networkAddress);
}
```

図 5.10: 抽象クラス: `AccessPoint`

`getStat` メソッドが、通信環境の計算を行なうクラスである。`getStat` メソッドは、SOUND の 1 ステップごとに全仮想車両に対して呼ばれる。`getStat` メソッドは各基地局クラスによりオーバーライドされるが、その際に各基地局の特性に合致したメソッドを実装する必要がある。本実装における 802.11b では、基地局からの距離に応じて、あらかじめ測定した値を返却する。また、PDC は位置に依存せず共通して 9600bps、PHS ではセル内外の判定のみを行ない、32kbps の帯域を返す。

第6章 検証および評価

本章では、本研究で構築した自動車環境アプリケーション開発環境の検証および評価を行なう。まず、開発環境の動作検証と既存手法との定性的な評価を行なう。そして、本開発環境を利用したアプリケーションの動作を、サンプルアプリケーションの実装を行ない、実際に検証することにより、本機構の有効性を示す。

6.1 システム検証・評価

本研究で構築したアプリケーション開発環境のシステム評価として、動作の検証と既存技術との定性的な比較を行なう。

6.1.1 動作検証

評価環境

動作検証では、評価用マシンと開発環境マシンの2台のマシンを利用する。表 6.1 に各マシンの仕様を示す。評価用マシンでは、データ収集を行なうための ping や netperf といったツールや、開発環境との通信を行なうためのアプリケーションが動作する。

図 6.1 に評価環境全体を示す。図に示すように、評価用マシンと開発環境マシンを、100Mbps のイーサネットで接続しデータの収集を行なう。開発環境マシン上には、通信デバイスは1つのみ搭載されており、その通信デバイスには複数のエイリアスアドレスが付与されている。評価用マシンと、開発環境上の仮想自動車プロセス間での通信を測定する。

表 6.1: マシン仕様

評価用マシン	CPU	PentiumIII 750MHz
	メモリ	192MBytes
	OS	NetBSD 1.6E
開発環境マシン	CPU	Pentium4 3GHz
	メモリ	512MBytes
	OS	FreeBSD 5.1-Release(IPFIREWALL/DUMMYNET enabled)

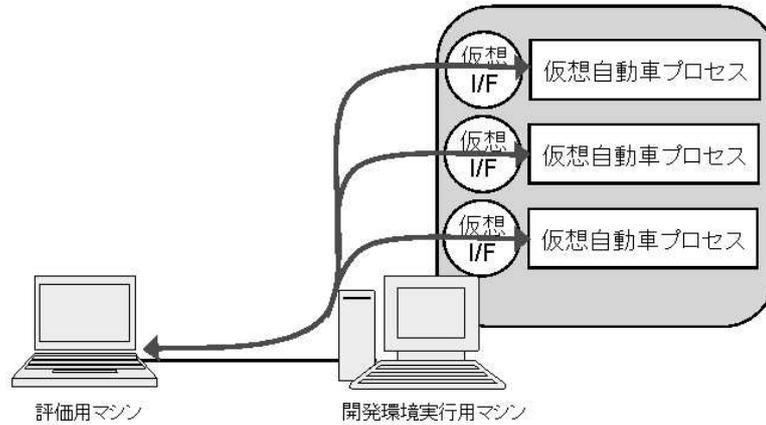


図 6.1: 評価環境

通信環境再現の評価

本研究で構築した開発環境において、通信環境の再現が動作しているかを検証する。検証用のデータとして、走行情報および 802.11b の基地局情報、PHS(AirHTM) 基地局情報を入力する。動作の検証を目的とするため、ここでは 2 つの通信デバイスのみを対象とする。図 6.2 に動作検証で用いる、10 台の仮想車両の走行軌跡と、802.11b 基地局の情報を示す。横軸に経度 (Longitude)、縦軸に緯度 (Latitude) を取る。各数値は度で表現される。各点が、交通流シミュレータ SOUND から出力される車両の移動軌跡を示す。今回 SOUND からは、30 秒ごとの車両の移動を 3 秒ごとに取得した。検証は 300 ステップ行なったため、現実での 2 時間 30 分の動作を再現した。SOUND は度単位で小数点第 7 位までの情報が出力する。また、x 点は基地局の設置位置、円は電波到達範囲を示す。電波到達範囲は 500m の円形であると設定した。

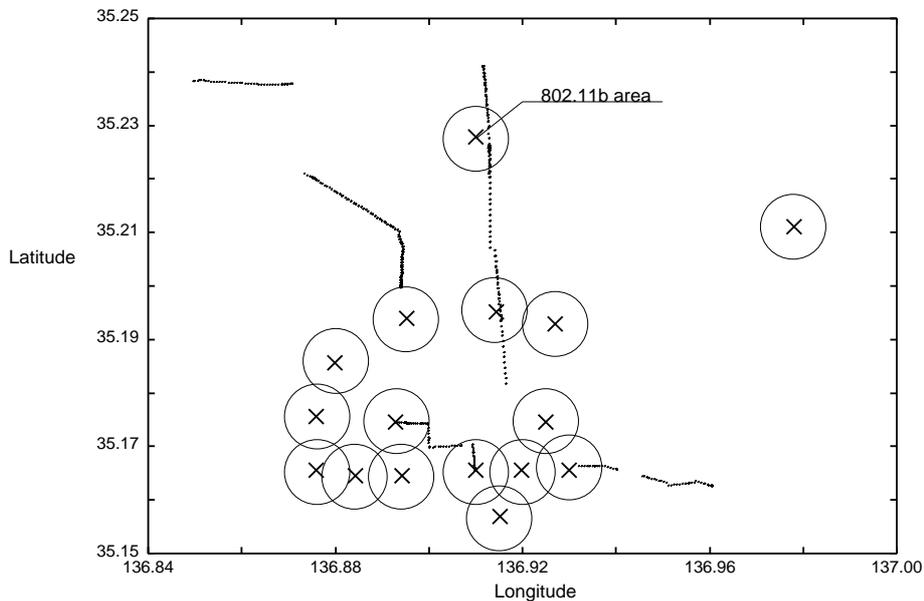


図 6.2: 仮想自動車の移動・802.11b 基地局の設置

各仮想自動車は全て 802.11b もしくは AirHTM のどちらかのデバイス 1 つを搭載する。Mobile IP を利用しているため、常に一意のアドレス (ホームアドレス) で通信可能であるように設定した。

図 6.3 に、開発環境上の仮想自動車に対する ping の出力を示す。横軸にはシーケンス番号を示す。ping は 1 秒毎に、仮想自動車のホームアドレスに対して送信している。縦軸には、ラウンドトリップタイム (RTT) をミリ秒の単位で示す。横にはシーケンス番号を示す。図からも分るように、各々の車両の位置情報の変化と共に、利用する通信デバイス・通信デバイスの電波強度が変化しているため、ラウンドトリップタイムが刻々と変化している。また、ラウンドトリップタイムが 0 を示す点では、電波範囲外を走行しているためか、パケットロスが発生しているため、通信が行なえていない。

図 6.4 に、開発環境上の仮想自動車との利用可能帯域の変化を示す。この車両は、図 6.3 とは別の自動車である。利用可能帯域は、評価用マシンから netperf を実行して測定した。縦軸には、利用可能帯域を Kbps の単位で示し、横軸には経過時間を示している。netperf は 3 秒ごとに連続して実行しているため、3 秒単位での変化が確認できる。図からも分るように、移動に伴って、実際の利用可能帯域が変化している。

このように、開発環境上の仮想自動車への通信が実現され、かつ仮想自動車の移動に伴って、通信環境の再現が実現されている。

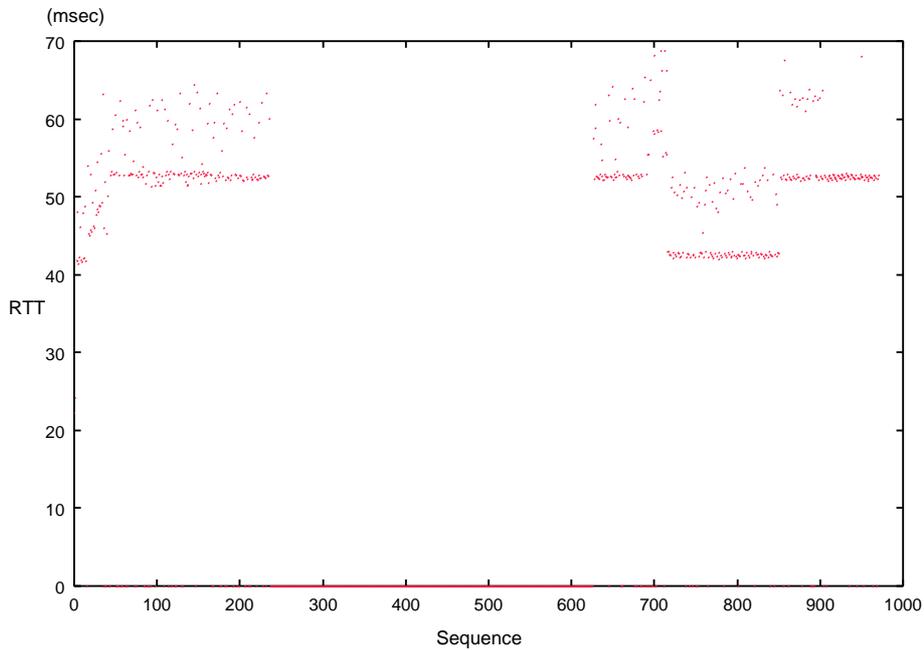


図 6.3: 開発環境におけるラウンドトリップタイム変化

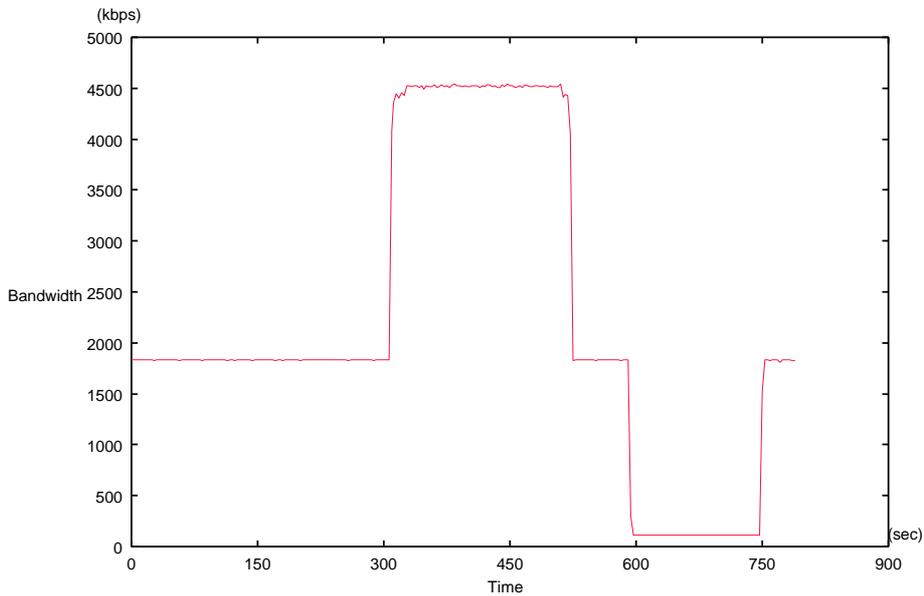


図 6.4: 開発環境における通信帯域変化

アクセスポイント性能変化

また、アクセスポイントの性能パラメータの変化が、仮想自動車の通信環境にどのように影響を与えるかを検証する。図 6.5 および 6.6 に性能の変化による影響を示す。

図では、縦軸に緯度、横軸に経度を示している。全ての自動車は 802.11b のみを利用して通信を行っており、定期的に UDP により自分の位置情報をサーバに送信している。受信

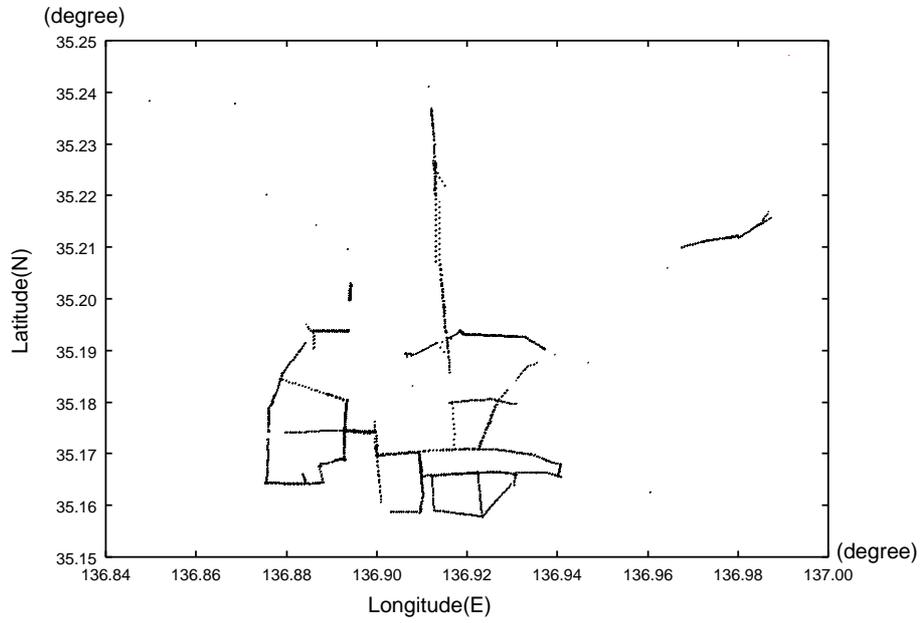


図 6.5: 電波到達範囲 500m の場合

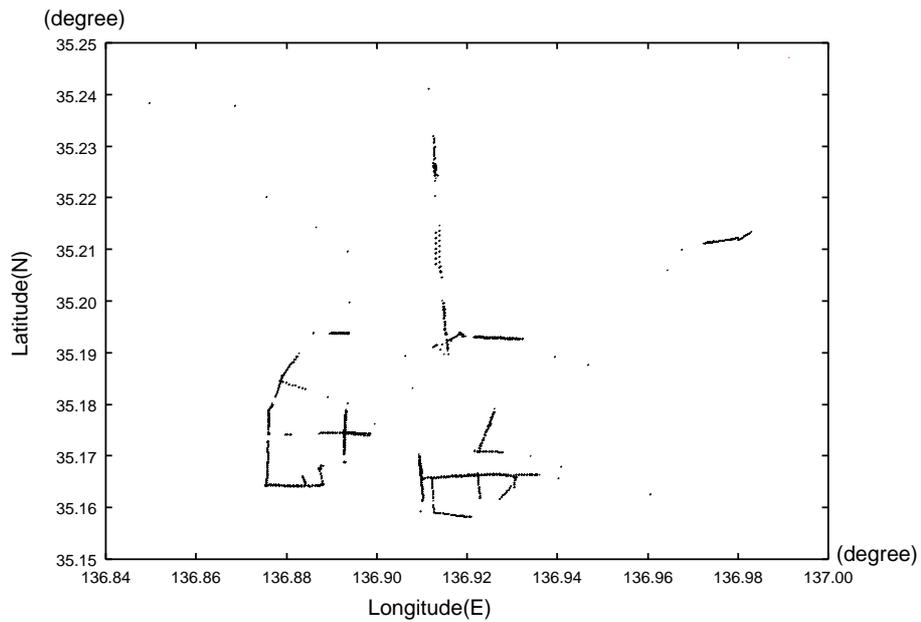


図 6.6: 電波到達範囲 250m の場合

した位置情報を全てプロットした。図 6.5 では、802.11b 基地局の電波到達範囲を半径 500m の円であると設定した。図 6.6 では、半径 250m の円であると設定した。

図からも明らかに分るように、電波到達範囲のパラメータを変更するだけで、仮想自動車の通信環境に大きく変化が表われている。

6.1.2 定性的評価

ここでは、自動車環境アプリケーション開発環境に対する要求事項を満たすシステムを構築したことを示す。表 6.2 に、本システムが、既存手法と比較して、どのように要求事項を満たしているかを示す。

本システムは単一のソフトウェア上で実現できるため、評価にかかるコストは低いと言える。処理能力向上のため、複数の計算機による分散処理なども考慮されているが、コストとのトレードオフが発生する。

仮想自動車モデルは外部シミュレータからの情報収集を前提としているため、外部情報生成機構との連携も可能である。また、設計段階においても、情報入力部分に汎用性を持つ設計を行なっているため、新しい機構の導入も容易に行なえる。しかし、本実装においては、交通流シミュレータ以外の機構からの入力インターフェイスなどは提供していないためとする。実装は今後の課題となる。

また、利用する交通流シミュレータを変更し、その環境に応じた無線インフラ配置情報を提供することにより、様々な自動車環境の再現が可能である。シミュレータの変更に関しては、今回の入力項目とした、緯度・経度・速度・方向の出力が行なえるシミュレータであれば、実装を改変することなく利用可能であるため、汎用性も確保している。

また、仮想自動車上では、InetApp 抽象クラスで抽象化されたアプリケーションを複数登録することが可能である。InetApp 抽象クラスを継承してアプリケーションを作成することにより、汎用的にアプリケーションを導入することが可能である。同様に、仮想自動車でのアプリケーション動作のスケジューリングをすることにより、柔軟な開発・検証も可能となるため、柔軟性も満たしている。

表 6.2: 要求事項への対応

	低評価コスト	外部機構との連携	汎用性	柔軟性	外部通信
シミュレータ					×
ランダムデータ				×	
実証実験	×	-	×	×	
本システム					

また、開発環境をインターネットに接続し、外部との通信を実現した。通信エミュレーション機能により、外部のサーバもしくは他車両との通信は、仮想自動車上の通信環境を再現している。

このように、本システムは自動車環境アプリケーション開発環境への要求事項を一部を除いて満たすことが出来た。外部機構との連携に関しては、実装レベルでの洗練が必要である。

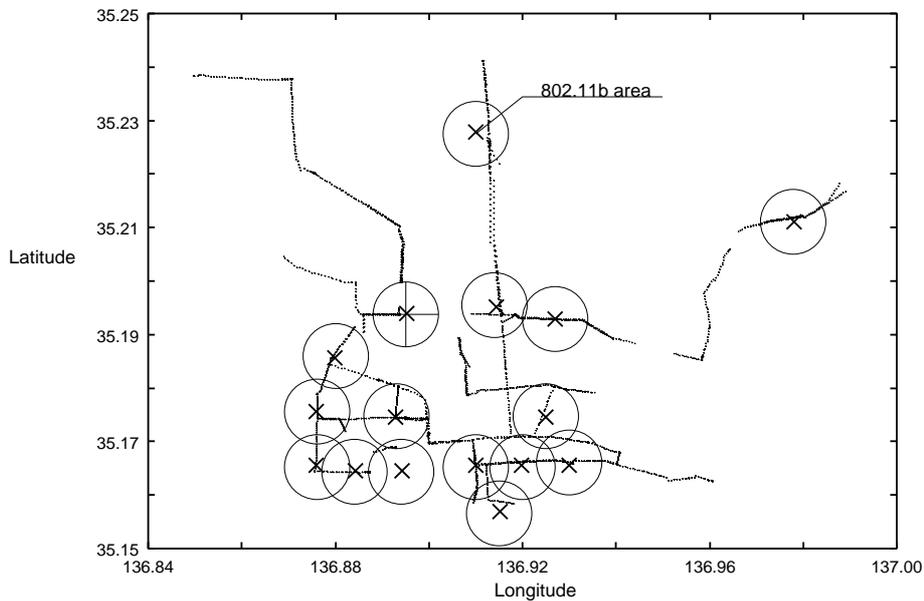


図 6.7: 仮想自動車移動経路・無線 LAN 基地局設置

本システムを利用することにより、自動車環境アプリケーションをより効果的に開発することが可能になる。

6.2 本システムを用いたアプリケーション開発

本研究で構築された自動車環境アプリケーション開発環境を利用して、実際にアプリケーションの開発・検証を行なう。開発されたアプリケーションが、本開発環境を利用して検証可能であることを示し、本システムの有用性を示す。

図 6.7 に開発で利用する仮想自動車の走行軌跡、および無線 LAN 基地局の設置状態を示す。図 6.2 と同じ地域での走行履歴であるが、車両台数が 100 台に増加している。横軸に経度 (Longitude)、縦軸に緯度 (Latitude) を取る。各数値は度で表現される。各点が、交通流シミュレータ SOUND から出力される車両の移動軌跡を示す。今回 SOUND からは、30 秒ごとの車両の移動を 5 秒ごとを取得した。検証は 300 ステップ行なったため、現実での 2 時間 30 分の動作を再現した。SOUND は度単位で小数点第 7 位までの情報が出力する。また、x 点は基地局の設置位置、円は電波到達範囲を示す。電波到達範囲は 500m の円形であると設定した。16 台の無線 LAN 基地局を設置した。

今回は、インターフェイス切り替え機構・プローブ情報送信アプリケーションの 2 つのアプリケーションを実装し、検証を行なう。インターフェイス切り替え機構に関しては、通信支援技術であるが、本研究ではアプリケーションレイヤで実現したため、アプリケーションとする。

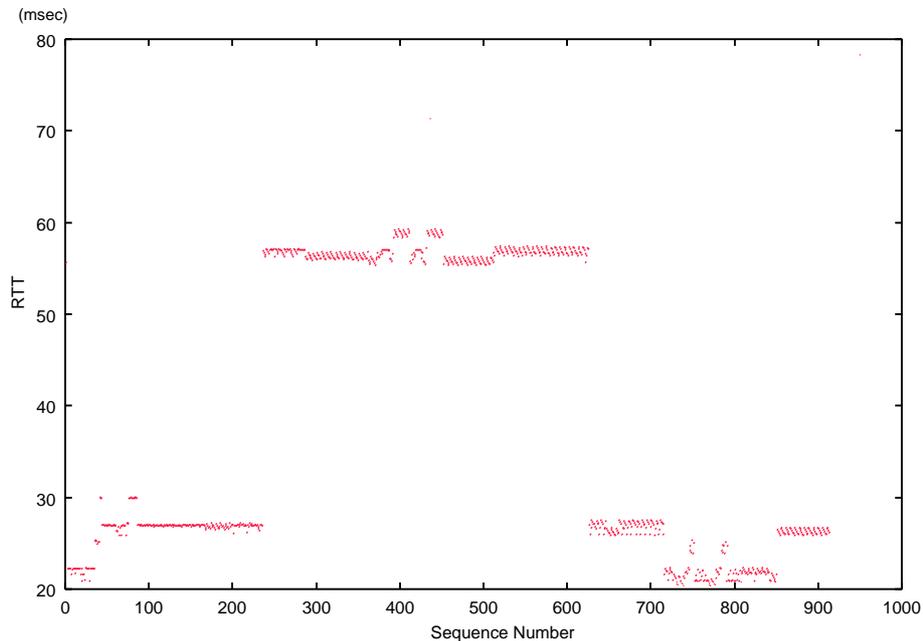


図 6.8: インターフェイス切り替え時のラウンドトリップタイムの変化

6.2.1 インターフェイス切り替え機構

インターフェイス切り替え機構は、自動車が複数の通信デバイスを搭載している場合に、それらを電波状況に応じて切り替えて利用するための機構である。この機構により、より広範囲での通信を提供することが可能である。インターネット ITS 協議会で 2001 年度に作成した高機能実験車では、8 つの通信デバイスを切り替えながら通信を行なうことが可能である。

インターフェイスを切り替える機構を仮想自動車上に実装し、インターフェイスを切り替えることにより、広範囲での通信が可能になることを示す。仮想自動車は Mobile IP 機能を持つため、インターフェイスの切り替えにより、IP アドレスが変化しても、Home Address により継続的に通信が可能である。

実装では、VirtualVehicle クラス内で通信状態のエミュレーションを行なう前に、全ての通信デバイスの状態を確認し、あらかじめ設定された優先度をもとに、電波強度の比較を行ない選択して、エミュレーションを行なうように実装した。電波強度に関しては、今回は最も近接したアクセスポイントを選択するように実装した。

図 6.8 に複数インターフェイスの利用を行なったときのラウンドトリップタイムの変化を示す。図 6.3 では、シーケンス 200 から 600 程度までパケットが到達していないが、インターフェイス切り替えにより、ラウンドトリップタイムの変化はあるが、継続して通信が行なえている。

6.2.2 プローブ情報システム

本システム上でプローブ情報システムの検証を行なう。プローブ情報システムでは、多数の車両が持つセンサ情報を一括して収集し、渋滞情報などの生成を行なうシステムである。プローブ情報システムの構築にあたっては、1) 通信環境によるプローブ情報の精度 2) デー

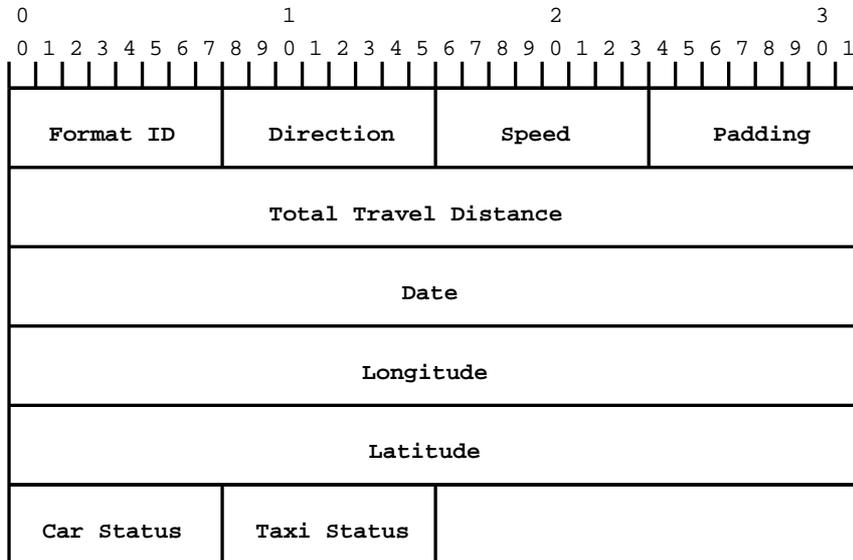


図 6.9: 実証実験におけるプローブ情報フォーマット

```

public class ProbeInfoSender extends netsim.InetApplication{
    public void setup(String sa, int sp, String la, int lp){
    }
    public void execute(){
        DatagramSocket dsocket = getUDPsocket(laddr, lport);
        /* スーパークラスのメソッドを利用して DatagramSocket を取得*/
        /* プローブ情報の作成処理 */
        dsocket.send(dp);
    }
}

```

図 6.10: プローブ情報送信アプリケーション実装例

タ送信量の削減 が大きな懸案事項である。プローブ情報システムは常時、大量の車両データを受信するため通信コストが高くなる。1つのデータ量だけでなく、全体としてのデータ送信量を削減する必要がある。また 2) に関しては、生成するプローブ情報の精度と車両台数の関係を明らかにすることにより、適切なプローブ車両の配置が可能になる。

インターネット ITS 協議会が 2001 年度に行なった実証実験でのパケットフォーマットを図 6.9 に示す。パケットには、緯度・経度・速度・方向や、自動車としての情報・タクシーとしての情報などを格納するフィールドがある。パケット全体は 22Bytes である。

現在、実証実験で利用されているプローブ情報システムでは、UDP を利用しているため、本検証においても仮想自動車に UDP のパケットを送信するアプリケーションを搭載した。図 6.10 にプローブ情報送信アプリケーションの実装例を示す。本システム上のアプリケーションは netsim.InetApplication 抽象クラスを継承する必要がある。スーパークラスで定義された getUDPsocket メソッドを利用して DatagramSocket を取得している。これにより、呼出し元仮想自動車の仮想インターフェイスを利用した通信を実現している。

100 台の仮想自動車を設定し、300 ステップの再現を行なった。1 ステップは 30 秒ごとの環境をシミュレートしているため、150 分間の再現を行なうことになる。また、各仮想自動

車は 802.11b と AirHTM のデバイスを搭載するように設定しており、6.2.1 節で実装したインターフェイス切り替え機構をもちいて、デバイスを切り替えながら通信を行なう。

通信環境によるプローブ情報の精度

通信環境の再現により、どの程度プローブ情報の精度が変化するかを検証する。

表 6.3: 通信環境の再現によるデータ損失率比較

	送信データ数	データ受信数	データ損失率
通信環境再現無	30,000	30,000	0%
通信環境再現有	30,000	21,807	27.31%

表 6.3 に通信環境再現の有無によるデータ損失率の差を示す。通信環境の再現を行っていない場合には、送信した 30,000 データ中 100 のデータが到着している。しかし、通信環境を再現した場合には、30,000 データ中 21,807 データしか到着せず、27.31%のデータ損失が発生している。

図 6.11 には、本システムを利用せずに仮想自動車からの情報をプロットした図を示す。また、図 6.12 には、本システムを利用し、各仮想自動車の通信環境を考慮したデータ送信によるプロットの結果を示す。

図 6.12 では、図 6.11 と比較すると、円で囲んだ地域内でのデータが欠損している。8193 個のデータ損失が影響を与えている。全仮想自動車は、802.11b と AirHTM を搭載し、インターフェイス切り替え機構を利用して通信するように設定されているため、欠損が発生した地域では、通信メディアが両方ともに利用不可能であったと考えられる。より広範囲でのデータ収集を実現するためには、PDC や衛星通信などの利用を検討する必要があることが分かる。

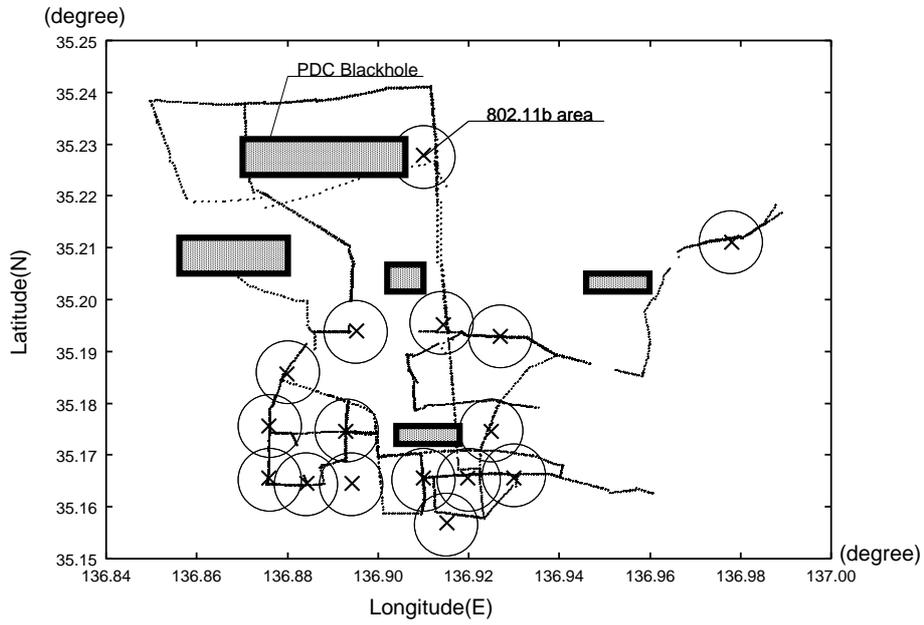


図 6.11: 通信環境を考慮しない場合

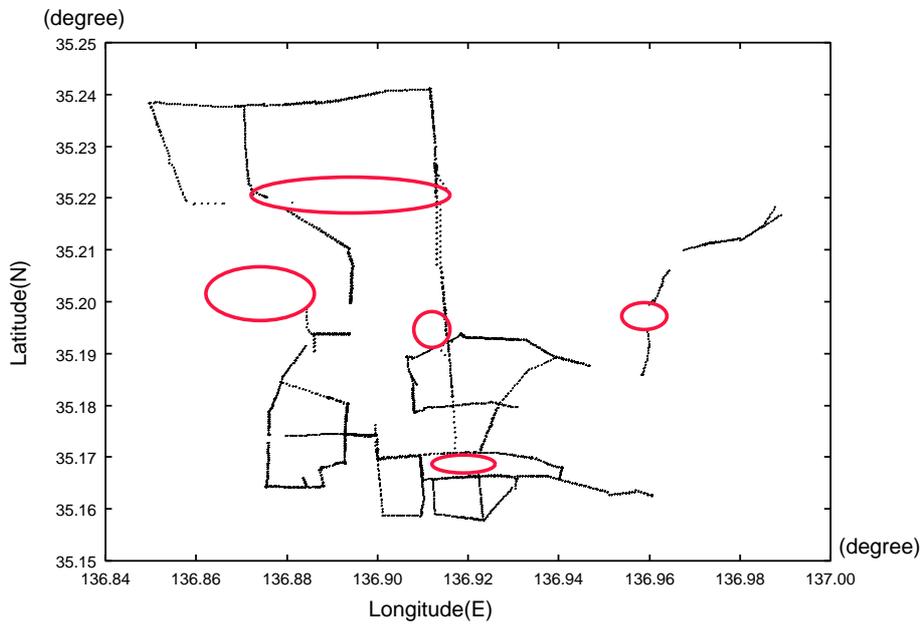


図 6.12: 通信環境を考慮した場合

データ送信量の削減

プローブ情報システムでは、全体のデータ送信量を削減するための手法が検討されている。有力な手法としては、代表自動車による送信という手法が挙げられる。プローブ情報システムでは、送信されるプローブ情報に含まれる降雨・速度・温度などの情報は、近隣自動車間であれば大きく異ならないという前提が可能である。この前提に基づき、周辺にプローブ車両が密集している環境では、全ての車両がデータを送信するのではなく、代表車両のみがデータを送信することにより、全体としての送信量を削減しようとしている。また、イギリスの ISIT で行なわれているサービスでは、センタが情報を必要とする時に、オンデマンドでデータの取得を行なう手法を用いている。

この手法の検証を行なうため、本研究では代表車両による送信を擬似的に実装した。本実装では、周辺に仮想自動車が存在する場合には、ある確率でランダムで送信する機構を実装した。現実はこの手法を実現するためには、代表車両の決定などのアルゴリズムが必要になるが、今回は実装例としてランダム数を用いた送信ポリシーの決定を行なった。

本システムでの、通常のプローブ送信アルゴリズム、つまり全自動車が定期的にデータを送信するアルゴリズムでは、計 21,807 パケットが送信されている。実装したデータ送信量削減アルゴリズムを利用して、データの収集を行なったところ、パケット数は計 7,526 パケットまで削減できた。検証中のある時点でのプローブ情報の受信状況を位置によりプロットした結果を、図 6.13-図 6.14 に示す。図 6.13 では全ての自動車が通常のアルゴリズムで情報送信を行なうケース、図 6.14 では削減アルゴリズムを利用した送信を行なっているケースの結果を示している。図 6.14 では、パケット総数が約 1/3 まで削減されているため、いくつかの地域で情報が取得できていない。しかし、丸で囲まれた地域では、通常アルゴリズムでは多数のパケットが送信されているが、削減アルゴリズム利用時には 1 つのパケットで、情報の収集が可能になっている。

6.3 評価総括

評価として、実装した開発環境の動作検証、既存手法との定性的比較、評価用アプリケーションの実装・検証の 3 つを行なった。

動作検証では、仮想自動車および無線基地局を設定し、通信環境のシミュレーションおよびエミュレーションが動作していることを示した。また、無線基地局の設定の変更が、仮想自動車の通信に影響を与えることも示した。

既存手法との定性的比較では、3 章で挙げた要求事項をより適切に満たしていることを示し、本開発環境の優位性を示した。

また、評価用アプリケーションの実装・検証として、構築した開発環境上で、実際に 2 つの自動車環境アプリケーションを実装し、検証を行なった。これにより、本開発環境によりアプリケーションの開発から検証までの工程を実行することが可能であることを示した。

これらの評価から、本研究で構築した、自動車環境アプリケーション開発環境の有効性及び優位性が証明された。

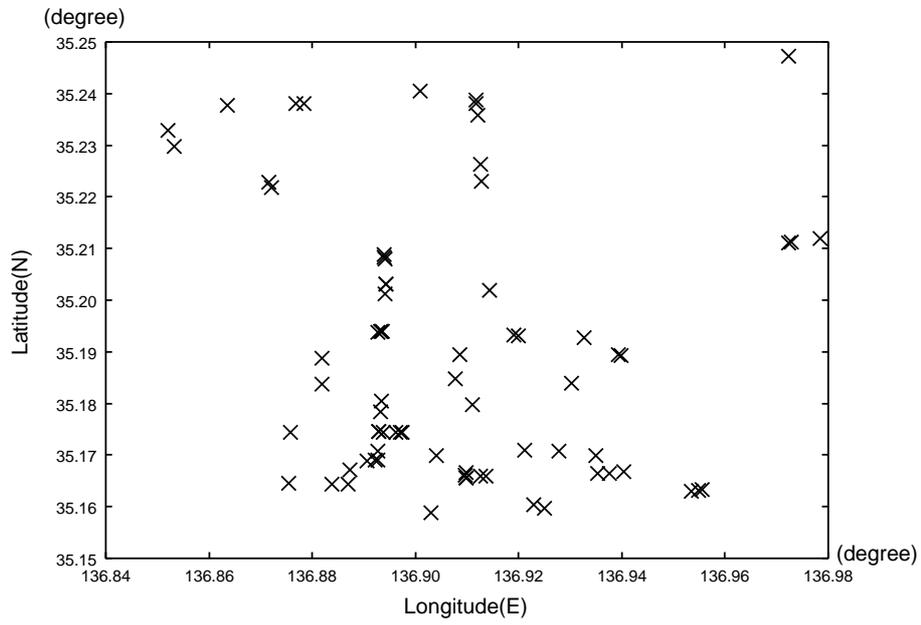


図 6.13: 通常アルゴリズム利用時

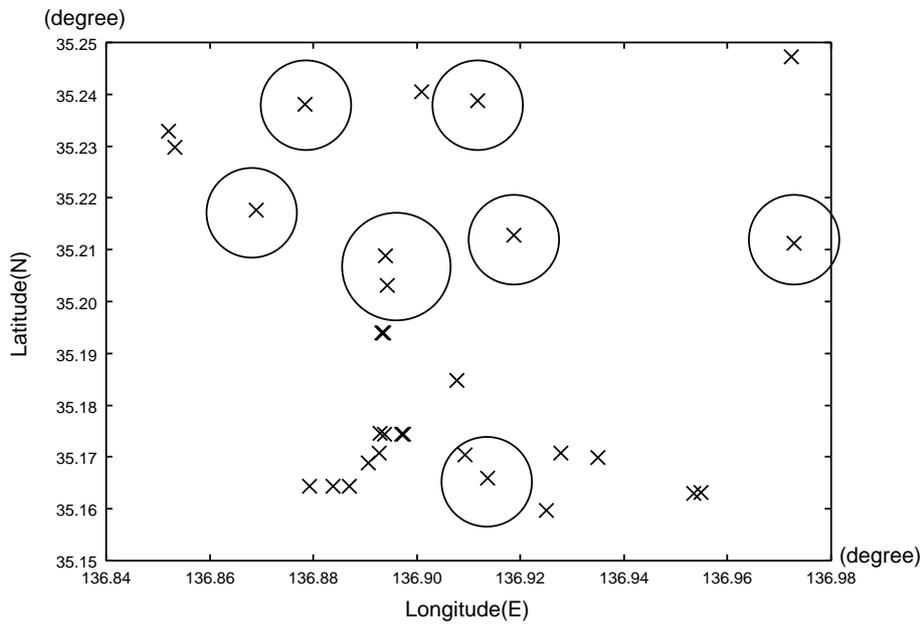


図 6.14: 削減アルゴリズム利用時

第7章 結論

7.1 まとめ

本研究では、インターネット自動車における、通信環境を考慮したアプリケーション開発環境の構築を行なった。インターネット自動車におけるアーキテクチャ全体に関する考察を行ない、その結果得られた開発環境への要求事項をもとにしたアプリケーション開発環境を構築した。

インターネット自動車上では、本研究で自動車環境アプリケーションと呼んだ、自動車の環境情報を元にした動作を行なうアプリケーションの開発が重要となる。自動車環境アプリケーションの開発環境を構築するため、インターネット自動車全体のアーキテクチャを物理層から、アプリケーション層までをレイヤ構造でモデル化し、自動車環境への影響要素を抽出した。その上で、仮想自動車モデルによる開発環境を提唱した。これは、抽出した影響要素をもとに、自動車環境を仮想自動車上で再現し、その上でのアプリケーション開発を実現するモデルである。

本研究では、交通流シミュレータ SOUND からの走行情報と、IEEE802.11b・AirHTM の無線品質を入力し、仮想自動車上での通信品質および周辺環境の再現を行なった。仮想自動車上に導入されたアプリケーションは、外部のサーバと、通信品質にあった通信を行なうことが可能になった。また、評価として動作検証および既存手法との定性評価を行ない、優位性を示した。同時に、本開発環境を用いたアプリケーションの開発および検証を実際に行ない、本開発環境の有効性を示した。

本研究により、インターネット自動車上でのアプリケーション開発環境が構築された。仮想自動車を用いて構築することにより、自動車環境アプリケーションの開発および検証を行なう優れた開発環境を提供した。本開発環境の整備により、今後アプリケーション開発も促進されると考えられる。

7.2 今後の課題

- 外部機構との連携
本システムでは、外部機構との連携を考慮した設計を行なった。しかし、実装では交通流シミュレータ以外の外部機構からの入力を受けていない。より多くの情報を仮想自動車に対して入力し、仮想自動車上が持つ環境情報を増やしていく必要がある。
- 開発者側インターフェイスの整備
本研究では、開発者が開発環境自体にアプリケーションの組込みを行なう必要があった。システムの開発者が、容易に組込めるインターフェイスを提供していく必要がある
- 規模性の確保

本研究では、規模性に関する考察は行なっていない。実際の自動車環境では、非常に多くの車両が走行し、かつ環境に影響する要素も複雑である。そのため交通流シミュレータには、様々なものがあり、出力する情報の粒度・精度もまた多様である。また、本開発環境は交通流以外のシミュレータからの入力も受けうるため、それらのシミュレータを利用した際の規模性も確保する必要がある。

謝辞

本研究を進めるにあたり、ご指導をいただきました慶應義塾大学 環境情報学部教授 村井純博士、同学部助教授 楠本博之博士、奈良先端科学技術大学院大学教授 砂原秀樹博士に感謝します。また、日々厳しく助言いただいた慶應義塾大学 環境情報学部助教授 中村修博士に深く感謝します。

インターネット自動車というかけがいのないフィールドでの研究の機会を与えていただきました、慶應義塾大学 植原啓介博士、佐藤雅明氏、渡辺恭人博士、奈良先端科学技術大学院大学 和泉順子博士に感謝します。

また、HAKONIWA というアイデアの中で、交通の分野からの貴重なご指導を頂いた、千葉工業大学建築都市環境学科教授 赤羽弘和博士、(株) i-TransportLab. 堀口良太博士にも感謝いたします。実装の基礎に協力していただいた(株) NEC ソフト 春田仁氏にも感謝いたします。

学部から修士課程に至るまでの4年半、日々の研究活動を支えてくださった、重近範行博士、湧川隆次氏、今泉英明氏、小原泰弘氏、藤枝俊輔氏、海崎良氏をはじめとする研究会の諸先輩に感謝致します。また、年末のお忙しい時期に赤入れをしていただいた(株) Ericsson Research Japan 杉本信太氏、通信シミュレーションに関してサポートいただいた University of Strusburg Mr. Nicolas Montavont にも感謝します。

また、これまで苦楽をともにしてきた同期である三屋光史朗氏、小畠元氏、宮嶋慶太氏、まーべる、片岡広太郎、岩谷晶子氏、西原サヤ子氏。ありがとう。これからもよろしく。

色々と研究活動をサポートしてくれた渡里雅史氏、小柴晋氏、岡田耕司氏、入野仁志氏、苧阪浩輔氏にも感謝します。そして、ともに学位論文を執筆した清水崇史氏、廣瀬峻氏、高橋宏明氏、成瀬大亮氏、堀岡大輔氏、橋本和樹氏 感謝します。

研究活動の視野を大きく広げてくれた WIDE プロジェクトの皆様、特に Dr. Thierry Ernst 他の皆様へ感謝いたします。また、私の大学、大学院生活の基礎となった慶應義塾大学 徳田・村井・楠本・中村・南研究室諸氏、とくに NACM の皆様、sing の皆様へ感謝します。

日々の糧を与えてくれた、きのこ、ニューオリンズ、Kaito、Pepita Lion に感謝。執筆で疲れた目や心を癒してくれた、湘南の海、富士山に感謝。

いつも見守り励ましてくれた卓山慶子氏と、その愛犬ジュリーに感謝を捧げます。

2004年1月12日

日野哲志

参考文献

- [1] WIDE プロジェクト. <http://www.wide.ad.jp/>.
- [2] 植原啓介, 砂原英樹, 村井純. インターネット自動車概要. In 情報処理学会会誌「インターネットと自動車」特集号, volume 43-4, pages 350–356, 2002.
- [3] Keisuke UEHARA, Yasuhito WATANABE, Hideki SUNAHARA, Osamu NAKAMURA, and Jun MURAI. InternetCAR -Internet Connected Automobiles-, 1998.
- [4] Hong Lach. *Laboratory and 'Field' Experiments with IPv6 Mobile Networks in Vehicular Environments*, October 2003. Work in Progress,draft-lach-nemo-experiments-overdrive-01.txt.
- [5] Keisuke UEHARA, Tatemichi TATEOKA, Yasuhito WATANABE, Hideki SUNAHARA, Osamu NAKAMURA, and Jun MURAI. A Network Architecture for Continuous Mobility. In *LNCS1368(Proc. of WWCA '98)*, Springer-Verlag, pages 224–269, Feb 1998.
- [6] 杉本信太, 植原啓介, 三屋光史朗, 村井純. 車載コンピュータにおける BSD の活用, July 2000. 第3回プログラミングおよび応用システムに関するワークショップ.
- [7] 渡辺恭人, 竹内奏吾. インターネット自動車と地理位置情報サービス. In 情報処理学会会誌「インターネットと自動車」特集号, volume 43-4, pages 357–362, 2002.
- [8] 和泉順子, 湧川隆次, 川喜田佑介, 秋山由和. インターネット ITS プロジェクトの概要. In 情報処理学会会誌「インターネットと自動車」特集号, volume 43-4, pages 369–375, 2002.
- [9] Internet ITS プロジェクト. <http://www.internetits.org/>.
- [10] 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). In 情報処理学会会誌「インターネットと自動車」特集号, volume 43-4, pages 376–385, 2002.
- [11] DDI Pocket. <http://www.ddipocket.co.jp/top/index.html/>.
- [12] A. Conta and S. Deering. *Generic Packet Tunneling in IPv6 Specification*, December 1998. RFC 2473.
- [13] SmokePing: About SMOKEPING. <http://people.ee.ethz.ch/oetiker/webtools/smokeping/>.
- [14] S. Corson and J. Macker. *Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations*, January 1999. RFC 2501.

- [15] C. Perkins, E. Belding-Royer, and S. Das. *Ad hoc On-Demand Distance Vector (AODV) Routing*, July 2003. RFC 3561.
- [16] Yih-Chun Hu David B. Johnson, David A. Maltz. *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)*, April 2003. Work in Progress, draft-ietf-manet-dsr-09.txt.
- [17] M. Lewis R. Ogier, F. Templin. *Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)*, April 2004. Work in Progress, draft-ietf-manet-tbrpf-11.txt.
- [18] T. Clausen, Ed., P. Jacquet, and Ed. *Optimized Link State Routing Protocol (OLSR)*, October 2003. RFC 3626.
- [19] 湧川隆次, 植原啓介, 田村陽介, 徳田英幸. ネットワークエンティティの状態変化に対応する管理機構の設計と実装, Oct 1998. 情報処理学会全国大会.
- [20] Ryuji Wakikawa, Susumu Koshihara, Keisuke Uehara, and Jun Murai. Multiple Network Interface and Policy-Based Routing on Mobile IPv6, July 2002. Proc. of the 2002 International Conference on Wireless Networks (ICWN).
- [21] C. Perkins and Ed. *IP Mobility Support for IPv4*, August 2002. RFC 3344.
- [22] Jari Arkko Dave Johnson, Charles Perkins. *Mobility Support in IPv6*, July 2003. Work in Progress, draft-ietf-mobileip-ipv6-24.txt.
- [23] Ryuji Wakikawa, Keisuke Uehara, Thierry Ernst. *Basic Network Mobility Support*, February 2003. Work in Progress, draft-wakikawa-nemo-basic-00.txt.
- [24] Thierry Ernst. *Network Mobility Support Goals and Requirements*, May 2003. Work in Progress, draft-ietf-nemo-requirements-01.txt.
- [25] J. Postel. *Internet Protocol*, September 1981. RFC 791.
- [26] R. Hinden and S. Deering. *Internet Protocol Version 6 (IPv6) Addressing Architecture*, April 2003. RFC 3513.
- [27] S. Kent and R. Atkinson. *Security Architecture for the Internet Protocol*, November 1998. RFC 2401.
- [28] 国土交通省道路局 ITS ホームページ ETC サービス. <http://www.its.go.jp/ITS/j-html/index/indexEtc.html>.
- [29] 道路交通情報通信サービス VICS. <http://www.vics.or.jp/>.
- [30] 国土交通省道路局 ITS ホームページ 走行支援サービス. <http://www.its.go.jp/ITS/j-html/index/indexAhs.html>.
- [31] 岡田耕司. MANET を用いた車車間マルチホップ通信環境の構築, 2002. 慶應義塾大学環境情報学部 卒業論文.

- [32] 日野哲志, 佐藤雅明, 植原啓介, 村井純. インターネット自動車システムにけるユーザ支援型アプリケーションの設計および実装, Nov 2000. 情報処理学会 第 15 回モバイルコンピューティングとワイヤレス通信研究会第 3 回高度交通システム研究会 合同研究会.
- [33] POIX:Point Of Interest eXchange Language Specification. <http://www.w3.org/TR/poix>.
- [34] 和田光示. プローブ情報システム (IPCar) プロジェクト. In 情報処理学会会誌「インターネットと自動車」特集号, volume 43-4, pages 363–368, 2002.
- [35] S. Glass, T. Hiller, S. Jacobs, and C. Perkins. *Mobile IP Authentication, Authorization, and Accounting Requirements*, October 2000. RFC 2977.
- [36] 佐藤雅明. インターネットにおける自動車情報の抽象化およびデータ辞書モデルの設計, 2001. 慶應義塾大学 政策・メディア研究科 修士論文.
- [37] J.D. Case, M. Fedor, M.L. Schoffstall, and J. Davin. *Simple Network Management Protocol (SNMP)*, May 1990. RFC 1157.
- [38] 佐藤雅明, 杉浦孝明, 高橋寿平. 国際動向. In 情報処理学会会誌「インターネットと自動車」特集号, volume 43-4, pages 386–391, 2002.
- [39] Ryuji Wakikawa and Keisuike Uehara and Fumio Teraoka and Jun Murai. MIBsocket: An Integrated Mechanism to Manipulate Network Resource Information in Mobile Communications, 2001. IEICE special issue on Internet Technology.
- [40] K. McCloghrie and Ed. *SNMPv2 Management Information Base for the Internet Protocol using SMIPv2*, November 1996. RFC 2011.
- [41] K. McCloghrie and Ed. *SNMPv2 Management Information Base for the Transmission Control Protocol using SMIPv2*, November 1996. RFC 2012.
- [42] K. McCloghrie and Ed. *SNMPv2 Management Information Base for the User Datagram Protocol using SMIPv2*, November 1996. RFC 2013.
- [43] R. Braga, C. Werner, and M. Mattoso. *Odyssey: A Reuse Environment based on Domain Models*, 1999.
- [44] Takamichi Tateoka and Hideki SUNAHARA. CM1: Communication Monitor for Applications Adaptive to Execution Environments.
- [45] Tatsuo Nakajima, Hiroyuki Aizu, Masaru Kobayashi, and Kenji Shimamoto. Environment Server: A System Support for Adaptive Distributed Applications. In *WWCA*, pages 142–157, 1998.
- [46] The Network Simulator - ns-2. <http://www.isi.edu/nsnam/ns/>.
- [47] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks. In *Workshop on Parallel and Distributed Simulation*, pages 154–161, June 2003.

- [48] Christian Lochert, Hannes Hartenstein, Jing Tian, Holger Fubler, Dagmar Hermann, and Martin Mauve. A Routing Strategy for Vehicular Ad Hoc Networks in City Environments. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2003*, pages 156–161, June 2003.
- [49] 植原啓介, 村井純. インターネットによる自動車情報化支援環境の構築, Nov 2000. 情報処理学会 第15回モバイルコンピューティングとワイヤレス通信研究会第3回高度交通システム研究会 合同研究会.
- [50] 春田仁, 渋谷裕久, 堀口良太, 赤羽弘和, 森川高行, 中村英樹, 植原啓介. インターネット ITS アプリケーション検証シミュレータ, June 2003. 土木計画学研究発表会.
- [51] Luigi Rizzo. Dummynet: a simple approach to the evaluation of network protocols. *ACM Computer Communication Review*, 27(1):31–41, 1997.
- [52] 社団法人 交通工学研究会編. やさしい交通シミュレーション, 2000.
- [53] Ntt docomo サービスエリア. http://www.nttdocomo.co.jp/p_s/area/.
- [54] Vodafone サービスエリア. http://www.vodafone.jp/japanese/service_area/index.html.
- [55] Au サービスエリア. http://www.au.kddi.com/service_area/index.html.
- [56] Thierry Ernst, Hong-Yon Lach. *Network Mobility Support Terminology*, May 2003. Work in Progress, draft-ietf-nemo-terminology-00.txt.