

第 13 部

自動車を含むインターネット環境の構築

第 1 章

はじめに

1.1 移動体通信環境の現状

近年、高性能なノート PC や PDA といった携帯型計算機や、携帯電話や PHS のような通信機器が手軽に入手できるようになってきており、それに伴っていわゆる“モバイルコンピューティング”— 移動体通信環境の利用者が著しく増加してきている。携帯電話によるパケット通信サービスや、PIAFS のようなデータ通信環境の進化には目を見張るものがあり、また社会的にそういったユーザを受け入れる土壌も育ちつつある。

しかしながら、依然としてデータ通信に対しては従量制課金となされるのが一般的であり、通信速度も高速モデムの域を大きく越えることはなく、いつでも好きなだけ高速なネットワークアクセスが利用できるという状況には至っていない。さらにノート PC のバッテリー寿命も通常 1 ~ 2 時間程度で、長くても数時間から半日しか連続した作業を行なえないという制約は、常に最新の情報に対するアクセスが求められる場面では厳しいものがある。

利用者はまた、自宅からオフィスへ、ある外出先から別の外出先へと移動をする際に、連続的な作業環境を必要とする場合が少なくない。安価で安定した接続性を期待できる ISP も多いが、地理的な移動に伴っていくつものアクセスポイントを渡り歩きながら、連続したネットワーク接続を続けられるような環境は、未だ一般的なものにはなっていない。携帯電話や PHS の利用可能エリアも順調に広がってはいるが、地下街や高速移動中の利用などにはやはり制限がある。

これらの様々な制約は、大まかに以下のような分類をすることができる。

1. 経済的な制約

- データ通信に対する従量性課金

2. 技術的な制約 (条件を与えることで現在の技術で解決可能な制約)

- 携帯型計算機のバッテリー寿命
- 移動に伴う利用可能な通信機器の動的な変更
- 移動に伴い連続的に利用可能なネットワーク環境

3. 技術的な制約 (現在の技術では本質的に解決が困難である制約)

- データ通信速度の上限
- エリア外に出た場合の通信の途絶

1. については社会的な背景もあり、一朝一夕に解決策を提示することは難しいが、必要な情報のみを適切なタイミングで通信するといったような、運用上の回避策を検討する価値はある。2. については、例えば大容量のバッテリーを用意したり、Mobile-IP のようないくつかの技術を組み合わせることで、十分実用的なレベルの環境を実現することは可能だと言える。3. は本質的な解決は困難であるが、技術上これらの制約を回避した運用を行なうための仕組みを検討することは可能である。

1.2 インターネットと自動車

一方自動車はバッテリーを積載し、人間の意志にしたがって人間と共に移動し、グローバルに分散するオブジェクトである。自動車には外気温センサや GPS のように周囲の情報を直接収集する機構や、例えばワイパーの ON/OFF で降雨の有無を判断したり、ライトの ON/OFF で付近の明るさを判断したりといったように、ある程度の統計処理をすることで、間接的に周囲の情報を収集しているとみなせる機構が多数搭載されている。

自動車がインターネットに対して接続性を持つことは、人間にとっての普遍的なデジタルコミュニケーションの環境を実現し、同時に自動車の持つ情報の有効な利用を実現することにつながるだろう。

本研究は、情報化社会を担うため、自動車をインターネットに接続し、情報を交換するための要素技術を吟味して、トータルなシステムの構築を行なうことを目的としている。また、ここで培われた技術を広く普及するよう努める。

1.3 WG の活動

InternetCAR-WG は本研究を行なうための分科会—ワーキンググループ (WG) として、1997 年 7 月より活動の準備を進め、1997 年 9 月に正式に WIDE プロジェクトの WG としての活動を開始した。本 WG の目標は、インターネットに自動車を接続することと、そのために必要な要素技術や将来性を検討することである。WG 活動は当初その活動を全 3 期に渡るフェーズに分割した。フェーズ構成と各フェーズの主な活動内容 (予定を含む) について、表 1.1 に記す。このうち 1997 年度は、Phase1 および Phase2 の 2 期に跨る活動を行なった。

表 1.1: InternetCAR-WG 活動のフェーズ構成

Phase1	Phase2	Phase3
1997 年 7 月 ~ 1997 年 9 月	1997 年 10 月 ~ 1998 年 3 月	1998 年 4 月 ~
<ol style="list-style-type: none"> 1. 1 台の実験車による基礎データの収集 2. 既存の PC+OS(UNIX 等) を使用した場合の移動体通信の問題点の再確認 3. 求められる機能と各種要素技術の洗い出し 	<ol style="list-style-type: none"> 1. 複数台の実験車による評価実験 2. 既存の PC+OS(UNIX 等) のための移動体通信支援パッケージの開発 3. センシング情報データベースの広域運用 	<ol style="list-style-type: none"> 1. 専用ハードウェア + OS の研究開発 2. エミュレータによるアプリケーション開発環境の構築 3. D-GPS 基盤環境の整備 4. セキュリティ関連

1.3.1 Phase1

Phase1 においては、まずインターネット接続が可能な実験車を 1 台用意し、搭載した PC に BSD/OS をベースとした実験環境を構築した。データ通信機器としてパケット通信可能な携帯電話端末を接続し、更に GPS モジュールにより地理的位置情報の取得を可能とした。

Phase1 の目標は次の 2 点とした。

- 実際に通信を行ないながら車を移動させることで、通信性能の特性やネットワーク接続の安定性、既存の技術で実現可能な技術と、新規に開発を必要とする技術の洗い出しなどを行なうこと。
- 自動車に搭載されている各種センサから取得可能なセンシング情報の調査と、その情報を有効に利用するための環境の検討を行なうこと。

1.3.2 Phase2

Phase2 においては、Phase1 に引続き同様な実験環境を搭載した実験車を複数台用意し、それらが同時に移動と通信を行なえる環境を用意した。

Phase2 の目標は次の 3 点とした。

- 利用可能な要素技術の選定と新規開発を元に、基本的な移動体通信環境を支援パッケージとして統合すること。

- 複数台の実験車で評価実験を行なうことで、スケーラビリティに関する見積りを行なうための基礎データを取得すること。
- 自動車の各種センサから取得したセンシング情報をデータベース上に構築し、アプリケーションからそれらの情報を利用するための環境を検討すること。

1.3.3 Phase3

Phase3 の目標は主に以下の点とする。

- 実用的な移動体通信環境を構築するための、専用ハードウェアや専用 OS を研究開発し、自動車のインターネットへの接続性を広く普及させるための準備を進めること。
- センシング情報データベースを利用するための、インタフェースとプロトコルおよびエミュレータを公開し、アプリケーション開発環境を広く一般に提供すること。
- インターネットに接続することで、より詳細な地理的位置情報を手軽に利用できるようにするために、WIDE バックボーン上に D-GPS 基盤環境の整備を進めること。
- 情報の可用性と匿名性を両立させるための、セキュリティ技術の研究開発を進めること。

詳細については第 7 章を参照のこと。

第 2 章

InternetCAR のアーキテクチャ

2.1 InternetCAR の定義

本報告中では、本研究の対象となる自動車を“*InternetCAR*”と呼ぶことにする¹。本章では、Phase1 および Phase2 において構築し、実験に使用した *InternetCAR* のアーキテクチャを説明する。本研究活動が将来的な開発目標とする、*InternetCAR* のアーキテクチャについては未だ流動的であるが、基本的な構成は現在のそれと大差ないものと予想される。

InternetCAR は、次のような機能を有するものとする。

インターネットへの接続を行なう機能

無線データ通信機器を経由して、インターネットに対する IP 的接続性を持つ。ただし無線データ通信機器の制約により、常に 100%の効率で接続性が確保できているとは限らない。また無線網を介する際の通信効率を考慮し、アプリケーションからは IP 的接続性が提供されているように見えるが、実際は IP 以外の通信プロトコルで通信しても良いものとする。

地理的位置情報を取得する機能

グローバルに分散する移動オブジェクトとしての自動車において、随時自身の存在する地理的位置に関する情報を、取得することができるものとする。取得する手段としては、次のいずれかもしくは複数の手段を組み合わせることができる。

- GPS(Global Positioning System) の利用
- PHS 端末の位置情報通知機能の利用
- カーナビゲーションシステムからの情報取得
- その他

各種センサからの情報を収集する機能

自動車に搭載されている各種のセンサなどからの情報を収集し、デジタル情報として

¹「*InternetCAR*」は文脈により、実験車を指したり、将来的なターゲットとなる開発目標を指したりする場合があるので、留意願いたい。

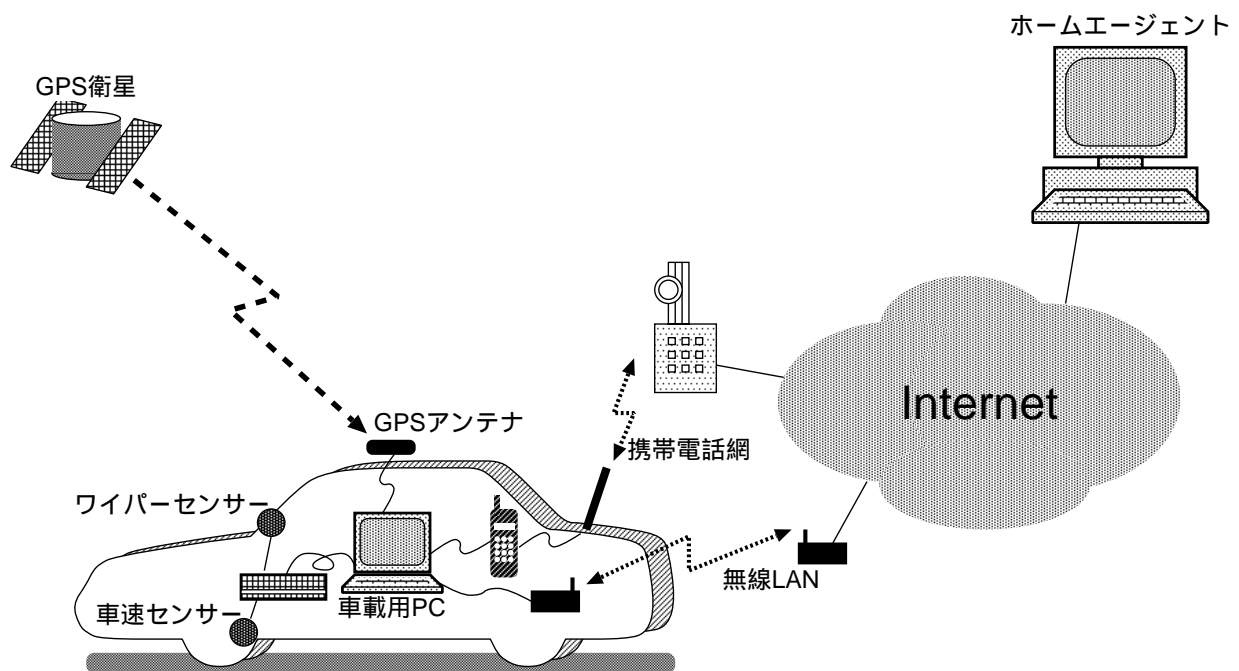


図 2.1: InternetCAR のモデル

取り扱うことができるものとする。収集することができるセンシング情報としては、以下のようなものがある。ただし車種により収集することができる情報とできない情報があるので、全てのセンシング情報を収集することができなくても良いものとする。

- 車速情報
- ワイパー (ON/OFF/間欠動作等)
- ライト (ON/OFF/スモール/上向き/後退灯/ハザード等)
- 外気温 (オートエアコン)
- その他

2.2 車載モジュールのハードウェア構成

実験に使用した *InternetCAR* の基本的なモデルを、図 2.1 に示す。このモデルにおいて、移動体計算機環境の中心は、BSD/OS をインストールしたノート PC を 1 台車載 PC として設定したものとする。

車載 PC は携帯電話網 (あるいは PHS 網) もしくは無線 LAN を通じて、インターネットへの接続を行なう。また GPS アンテナを装備し、GPS 衛星が見える範囲において、自分

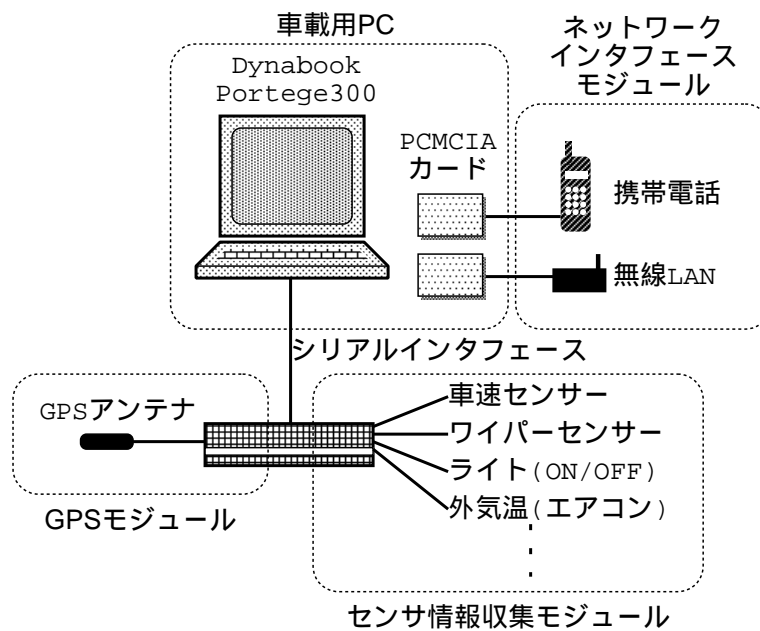


図 2.2: 車載ハードウェアの構成

の地理的位置情報を取得することができるものとする。さらに A/D 変換器などを装備した外部インタフェースモジュールにより、各種車載センサから収集したデータを、デジタル情報に変換して取得することができる。

実際の車載ハードウェアは、図 2.2 に示すような構成とした。

車載 PC として東芝製ノート PC 「PORTÉGÉ300CT」を置き、外部インタフェースモジュールとしていすゞ中央研究所製「センサ情報収集ボックス²」との間を、RS-232C シリアルケーブル (38,400bps) で接続した。センサ情報収集ボックスには、デジタル入力ポートと A/D 変換器が搭載されたアナログ入力ポートとがあり、各種センサからの入力をそれぞれのポートに接続する。

GPS アンテナは、トリンプル製のモジュールに対し、センサ情報収集ボックスとの間を RS-232C シリアルケーブル (9,600bps) で接続した。車載 PC は、センサ情報収集ボックス経由で GPS からの情報を取得することになる。

携帯電話端末は、NTT 移動通信網が提供する DoCoMo パケット通信サービス「DoPa」に対して、接続が可能な NTT 移動通信網製「P301HYPER」を使用し、車載 PC に対して専用 PCMCIA カードで接続した。また無線 LAN も、PCMCIA カード経由で接続した。

上記各機器に対して、車のバッテリー (DC 12V) から DC-AC インバータを介して生成した AC 100V を、電源として供給する。

²本モジュールは特注品であり、市販はされていない。

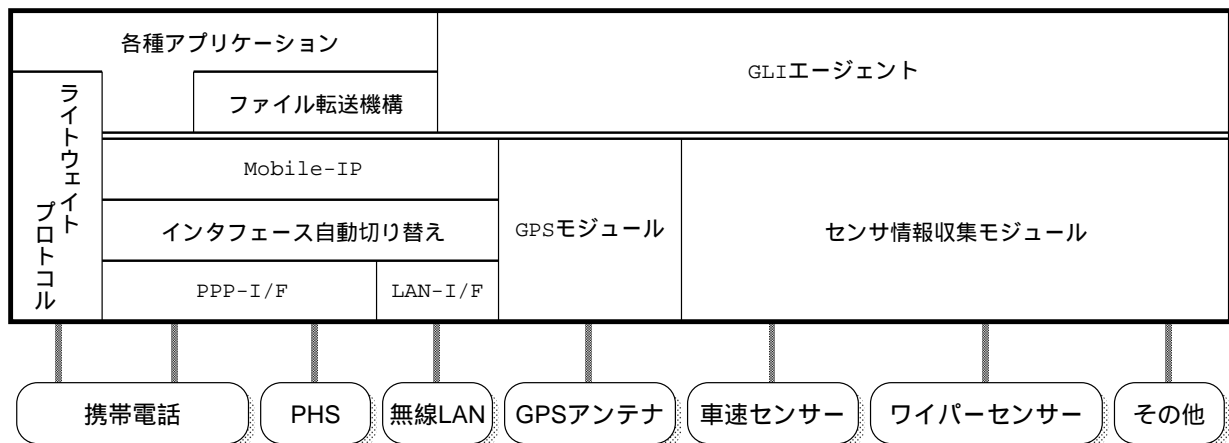


図 2.3: 車載モジュールのアーキテクチャ

2.3 車載モジュールのソフトウェア構成

車載モジュール上のネットワーク機能は、図 2.3に示すアーキテクチャ階層をとる。

上位層

中位層が提供する情報の送受信に関する API を利用し、各車載センサ類から取得したセンシング情報を処理して、インターネット経由で他のエンティティとの間で情報を交換する。

- GLI エージェント
「5.1 地理的位置情報システム」参照。
- ファイル転送機構
「5.2 スプールベースのファイル転送機構」参照。
- 各種アプリケーション
「5.3 サンプルアプリケーション」参照。

中位層

下位層が提供する機能へのインタフェースを抽象化し、上位層に対して API を提供する。ネットワークに対する通信プロトコルの処理も行なう。

- Mobile-IP
「4.1 IETF Mobile-IP」参照。
- ライトウェイトプロトコル

「4.2 ライトウェイトプロトコル」参照。

- インタフェース自動切り替え

「4.3 ネットワークインタフェースの自動切り替え」参照。

- GPS モジュール

「3.2 GPS モジュール」参照。

- センサ情報収集モジュール

「3.3 センサ情報収集モジュール」参照。

下位層

車載センサや通信機器類など、物理的なデバイス群が相当する。各デバイス固有のインタフェースにより、中位層からのアクセスや操作が行なわれる。

2.4 InternetCAR の移動

InternetCAR が地理的に移動する際の、ネットワーク的な通信イメージを図 2.4 に示す。

駐車場や車庫のように、*InternetCAR* が長時間に渡って存在する可能性が高いエリアに、無線 LAN の基地局が設置されているものとする。この場合車庫内では、無線 LAN 経由でインターネットへのアクセスが行なえる。

InternetCAR が移動するために車庫を出ると、無線 LAN の到達エリアから外れるため、ネットワークアクセスが行なえなくなる。*InternetCAR* は無線 LAN の到達エリアから外れたことを自動的に検知し、携帯電話など他のネットワークインタフェースを利用できるように、インタフェースの自動切り替えを行なう。

ネットワークインタフェースが切り替わったり、移動に伴って最寄りのアクセスポイントが変わるため、携帯電話の接続先を変更したりすることにより、*InternetCAR* は動的に自分自身の IP アドレスを順次変更していくことになる。IP アドレスの変更に関わらず通信を継続するために、*InternetCAR* では Mobile-IP 等の移動体通信プロトコルを利用する。そのため IP アドレスの変更は、随時ホームエージェントに通知される。また Mobile-IP を利用することによってインターネット上では、*InternetCAR* は常にホームアドレスを IP アドレスとして持つホストに見える。

InternetCAR 上の GLI エージェントは、車載センサ類から取得したセンシング情報を、随時 GLI サーバに送信する。GLI サーバは各 *InternetCAR* から集まってくるセンシング情報を蓄積し、センシング情報データベースとして公開する。

InternetCAR は様々な制約のもと、常に 100% のインターネット接続性を確保できるわけではない。そのためインターネットに接続していない時間帯が少なからず存在する。インターネットに接続していない時間帯に発生したデータ通信要求に対しては、スプールベースのファイル転送機構を提供し、通信要求を貯めておいて、インターネットへの接続性が回復した時点でまとめて処理できるようにする。

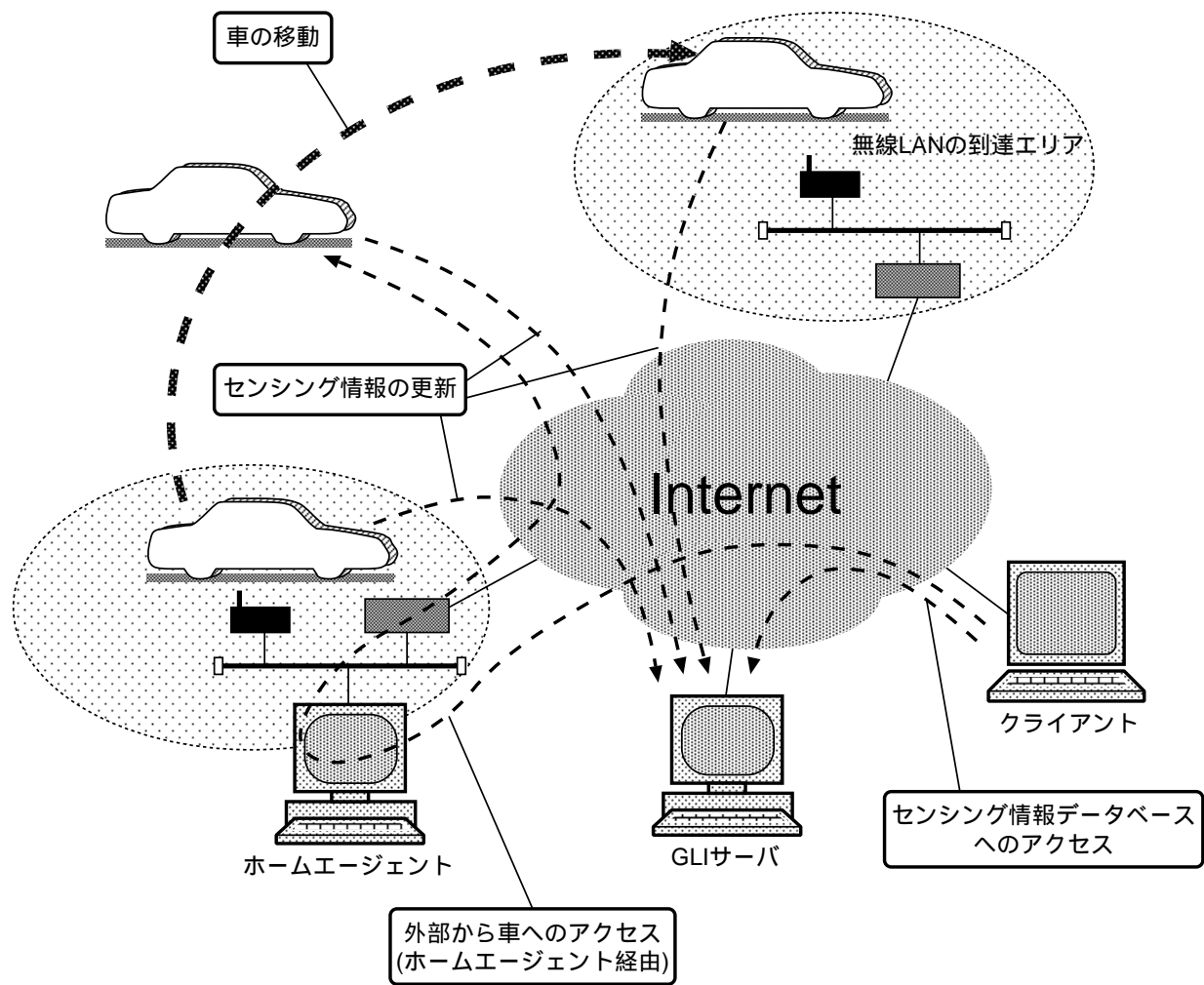


図 2.4: 移動する InternetCAR

第 3 章

車載ハードウェア

3.1 車載用 PC(プロトタイプ)

本 WG では現在のところ車載コンピュータとして、汎用のパーソナルコンピュータを使用している。現在稼働中の *InternetCAR* には、本田技研工業のオデッセイにはラックにマウントした IBM-PC/AT 互換機を搭載し、その他は、東芝製ノート PC「Dynabook PORTÉGÉ300CT」を搭載している。

InternetCAR で必要となる機能や制約は、近い将来への専用システム化およびハードウェア化を鑑みて絞り込む必要があると思われる。現状においては、その洗い出しと既存インターネット技術の適用と運用が可能で、通常の開発環境として利用している BSD/OS や FreeBSD 等の BSD 系 UNIX を導入可能な、IBM-PC/AT 互換機を選択している。

InternetCAR では、車載 PC に導入されているオペレーティングシステム(現在は BSD/OS)により、入出力インタフェースである RS-232C や PC カードのインタフェースなどを介して、接続された各種デバイスとのデータのやりとりを容易に行える必要がある。現在 *InternetCAR* では、この車載 PC は携帯電話網(あるいは PHS 網)もしくは無線 LAN を通じて、インターネットへの接続を行なう。また GPS アンテナを装備し、GPS 衛星が見える範囲において、自分の地理的位置情報を取得することができるものとする。さらに A/D 変換器などを装備した外部インタフェースモジュールにより、各種車載センサから収集したデータを、デジタル情報に変換して取得する。

3.2 GPS モジュール

GPS モジュールはトリンプル製のモジュールを利用した。GPS モジュール—センサ情報収集モジュール間は、RS-232C シリアルケーブル(9,600bps)で接続する。車載 PC は、センサ情報収集モジュール経由で GPS からの情報を取得することになる。

GPS はいくつかのメーカーから製造されている。現在、トリンプル社製の GPS を使用しているのは、GPS と接続されるセンサ情報収集モジュールが、トリンプル社製 GPS の出力を仮定して設計されているためである。次に新しく設計するセンサ情報収集モジュールは、他社の GPS の出力にも対応する予定である。

GPS モジュールは 1 秒毎にセンサ情報収集モジュールに対してデータを提供する。センサ情報収集モジュールは、これを次節で述べるセンサ情報パケットに入れて PC に出力する。

PC 側では GPS からの情報を直接受け取ることになるため、時間などの情報に関してはある程度の蓄積を行い、標準時刻に変更する必要がある。

3.3 センサ情報収集モジュール

本モジュールでは、*InternetCAR* に搭載されている様々なセンサーデバイスから出力される情報をリアルタイムに入力し、サンプリングを行った結果を RS-232C を介して PC に出力する。センサ情報収集モジュールは次に示す数の入力チャンネルを持つ。

- デジタル入出力: 40
内、以下のような構成となる。
 - パルス入力可能ポート: 16
 - エンコード入力ポート: 14
 - 出力ポート: 5
- アナログ入力: 8
- シリアル (RS-232C): 3 ポート
PC 接続用、GPS 接続用、レーダ接続用

現在はデジタル入出力に、車速パルス、ワイパー、ヘッドライト、ポジションライトを入力している。また RS-232C ポートには、PC と GPS が接続されている。

センサ情報収集モジュールは入力されたデータを 1 秒毎にサンプリングし、パケットの形で RS-232C に出力する。パケットフォーマットを表 3.1 に示す。

現在のデータパケットはヘッダやパケット境界を持たず、シリアルからデータを送るだけの機能しか持たない。このため、一度データ転送エラーが起きると自動復旧ができないという問題点を抱えている。

また、問題点としては電源の問題もある。ハードウェア設計の甘さのため、セルモータの動作による電圧変化に起因する誤動作が発生している。このため、一度エンジンをかけた後、センサ情報収集モジュールをリセットする必要がある。

3.4 ネットワークインタフェースモジュール

InternetCAR では現在、無線 LAN と NTT 移動通信網がサービスを行っている広域無線データ通信網 DoPa を利用している。

表 3.1: センサ情報収集モジュールから入力されるデータパケット構造

0—67	トリンブル社 TSIP プロトコル形式の位置情報, 時間, 速度
68—83	アナログ入力ポート 0—7 のデータ (10bits)
84—85	車間距離データ (11bits, 0.2m/bit)
86	相対速度データ (8bits, 0.72km/h/bit)
87	レーダによる自車速度データ (8bit, 0.72km/h/bit)
88	レーダ状態 (1)(8bits)
89	レーダ状態 (2)(8bits)
90	レーダ状態 (3)(8bits)
91—92	予約済
93	車速パルスデータ (8bits, 単位は車両により相違)
94—95	エンジン回転パルスデータ (16bits, 単位は車両により相違)
96—99	デジタル入出力 (32bits) DO4-DO0(5bit), DI2-DI0(3bit) Light1-Light0(2bit), WipP1-WipP0(2bit), SftP3-SftP0(4bit) DI9-DI5, Fbrk(6bit), DI4-DI3(2bit) DI7-DI10 (8bit)
100—127	予約済

無線 LAN は PC カードのものを利用している。インターフェイスはイーサネットのそれと同様なため、DHCP を介してアドレスを取得することも可能である。また、車庫のかわりとして研究室にベースステーション (無線 LAN—イーサネットブリッジ) を用意している。

一方、DoPa の方もモデムは PC カードのものを利用している。インターフェイスはモデムとほぼ同じであるが、PPP のパケットを網が理解して処理するため PPP のパケットしか送ることができない。また、アクセスポイントも DoPa の受け口に用意する必要があり、現在は NTT 移動通信網内のアクセスポイントサービスを利用している。

第 4 章

移動体通信プロトコル

4.1 IETF Mobile-IP

2.4節でも述べたように、*InternetCAR*に割り当てられる IP アドレスは時々刻々と変化すると考えられる。アドレスが変化する主な原因は以下のようなものである。まず、利用するインターフェイスが切り替わる場合がある。*InternetCAR*では複数のインターフェイスを持ち、ある時点において通信可能なインターフェイスを選択し通信を行なう。この利用するインターフェイスが切り替わった際に、同じ IP アドレスを割り当てられることを期待することは出来ない。また、車は高速かつ広範囲に移動するため、携帯電話を用いた PPP で接続していても、トンネル等の一時的に電波の届かない領域に入り、接続不能な状態がしばらく続いた後、再び電波が届く範囲に入り、PPP で再接続するといった状況がしばしば起きると考えられる。だが、この再接続を行なったときに、必ず前回と同じ IP アドレスが割り当てられるとは限らず、たとえばトンネルに入る前と入ったあとでは、割り当てられた IP アドレスが異なっていることも十分にありうる。

IP アドレスが変更されると、その以前に行なっていた通信の継続が出来なくなり、効率的な通信が期待できない。また、*InternetCAR*を 1つの IP ノードとして見た場合、*InternetCAR*の IP アドレスが次々と変化してしまうため、*InternetCAR*へ発呼して *InternetCAR*との通信を行なう事ができなくなる。

この問題を解決するため、*InternetCAR*では IETF Mobile-IP を使用した。

4.1.1 IETF Mobile-IP の概要

Mobile-IP では、移動ノードはホームアドレスと呼ばれる 1つのアドレスを、IP ネットワークのいかなる位置に移動しても使用する。このため、Mobile-IP を使用している移動ノードと通信している相手は、移動ノードの IP アドレスはホームアドレスであるように見えるので、移動ノードの移動によって通信が切断されることはなくなる。同時に、移動ノードの物理的位置に関わらず、移動ノードと通信したい場合はホームアドレスに対して発呼すればよい。

ただし移動ノードは、移動した先のネットワークで利用できる IP アドレスを取得する

必要がある。このアドレスを Care-of-Address(CoA) と呼ぶ。また、移動ノードの移動を支援するホームエージェントを、ホームネットワークに設置する必要がある。ホームネットワークとはホームアドレスが属するサブネットである。移動ノードは定期的にホームエージェントに対して、自分の現在位置である CoA を通知する。ホームエージェントは移動ノードのホームアドレス宛ての packets を、CoA に向けて転送する。

IETF Mobile-IP の詳しい動作原理は [112] を参照されたい。

4.1.2 InternetCAR における Mobile-IP の問題点

InternetCAR では複数のネットワークインターフェイスを利用するが、どのネットワークインターフェイスが現在接続可能なのかを知る必要がある。

Mobile-IP では、フォーリンエージェントと呼ばれる、移動先のネットワークで移動ノードを支援するためのエージェントが規定されている。フォーリンエージェントは、自分が支援するサブネットに対してその存在を広告するため、移動ノードはその広告を聞くことによって、そのインターフェイスが利用可能であることを知ることができる。

しかし、*InternetCAR* が接続する可能性のあるすべてのネットワークにおいて、フォーリンエージェントが配置されていることを期待することは出来ない。また、*InternetCAR* で利用するインターフェイスには、携帯電話のような常時接続が難しいインターフェイスもある。加えて、同時に複数のインターフェイスが利用できる状況も起こりうる。この時にどのインターフェイスを選択するのが最良であるかを判断することもできない。たとえば同時に 2 つの使用可能なインターフェイスがあるが、1 つは無料、1 つは有料のリンクであるような場合には、無料のリンクを選択したいということが考えられる。

InternetCAR ではこの問題を、ネットワークインターフェイスの自動切り替え技術を導入することで解決している。この技術については 4.3 節で述べる。

4.2 ライトウェイトプロトコル

InternetCAR で利用する無線インターフェイスには、無線 LAN と携帯電話や PHS 等の広域無線網の 2 つがある。前者は車庫やガソリンスタンドなどの停車時、後者は走行時に利用する。走行時に携帯電話等のインターフェイスを利用した場合、無線の特性のためにインターネット上で広く利用されている TCP が、本来のデータリンク層の性能を十分に引き出すことができない。性能を十分に引き出せない理由としては、使用しているインターフェイスのバンド幅が小さいことも影響している。そこでこれらの問題を解決するために、ライトウェイトプロトコル (LWP: Light Weight Protocol) を導入する。

4.2.1 概要

ライトウェイトプロトコルのアーキテクチャでは、TCP のコネクションを有線部分と無線部分とに分割する。これにより、無線の特性による性能低下を有線部分ではおさえることが可能となる。また、無線部分に TCP/IP/PPP というプロトコル群の代用として、LWP と呼ぶ専用のプロトコルを用いることにより、ヘッダによるオーバーヘッドやプロトコルのオーバーヘッドを削減している。LWP のアーキテクチャを図 4.1 に示す。

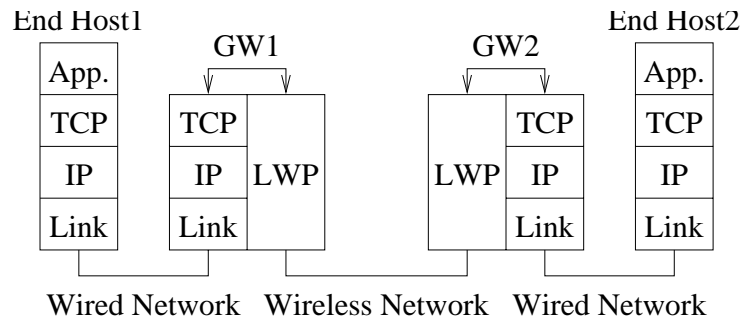


図 4.1: ライトウェイトプロトコルのアーキテクチャ

LWP は以下のようなデータリンクを仮定して設計されている。

- 低スループット
- 高遅延
- 遅延の揺れが大きい
- エラー訂正機能を持つ
- パケットの順序入れ換えが起きない

これらの特性を考慮し、ACK の省略、NACK によるエラー通知、ヘッダ情報の省略等をおこない、オーバーヘッドを削減している。

4.2.2 クルマでの LWP の利用

走行中に携帯電話を利用して接続をおこなう場合、データリンクとしての特性は劣悪なものとなる。移動しているために電波状態が変化し、遅延の揺れが大きくなったり一時的な切断などが起きる。この結果、TCP を使ったデータ転送の効率は著しく悪くなる。

一方、LWP を用いた場合、効率の問題は解決することができる。LWP は遅延の変化や中断に無関係に動作するように設計されている。このため、データリンクの状態が変化し

ても、可能な限りの速度でデータ転送をおこなうことができる。ライトウェイトプロトコルのアーキテクチャでは TCP の接続を分割しており、LWP を利用する部分は 1 ホップしかないために可能となっている。

しかし、LWP を使った場合にも幾つかの問題点が存在する。以下にその代表的な例を示す。

ネットワーク上の移動ができない。

LWP は PPP、IP、TCP の置き換えとなるプロトコルである。そのため、4.1 節で述べたような、Mobile-IP の技術をそのまま用いることができなくなる。結果として、無線 LAN と広域無線網を切替えて運用したり、サーバを変更することができなくなる。

LWP 用サーバを用意する必要がある。

PPP の場合、多数存在するサーバを利用することができるが、LWP を使用するためには専用のサーバを用意する必要がある。このため、NTT 移動通信網が提供する DoPa 等の、PPP を前提としたサービスも利用できない。

4.2.3 今後の課題と予定

InternetCAR の環境では安定した通信路を提供するために、LWP のような技術は必要不可欠である。しかし、一方で前節で述べたような問題点を解決しなければ実用には耐えない。問題を解決するためのアプローチとしては以下のようなものが考えられる。

1. LWP を IP 上に実装する。

LWP を IP 層の上に TCP の代替として実装する。これにより、Mobile-IP や次節で述べるインターフェイスの自動切り替えの技術をそのまま利用することができる。ただし、この手法は、ヘッダの圧縮率が悪くなる、インターフェイスの状態を把握するのが難しくなる等の問題点がある。

2. LWP に移動ホスト機能を追加する。

LWP 使用時にも Mobile-IP の登録を行い、Mobile-IP の技術と LWP の技術の融合をはかる。この方法では TCP のコネクションを途中から分割するような技術が必要となり、実現が難しい。

3. LWP とは異なるアーキテクチャの導入。

LWP は 1) プロトコル、ヘッダによるオーバヘッドの削減、2) 非直接接続技術、3) 層を無くすことによる肌理細やかな制御によって性能の向上を目指した。このようなアプローチではなく、1) TCP の改良 2) データリンク層による支援等のアプローチによる性能向上を目指す。

また、近年の技術進歩により広域無線通信網の性質も変化してきた。バンド幅は緩やかに大きくなりつつあり、安定度も向上してきている。このため、将来の広域無線データ通信網を考慮したプロトコルを設計する必要がある。

4.3 ネットワークインタフェースの自動切り替え

前節でも述べたように *InternetCAR* の環境では、無線 LAN と広域無線通信網を利用する。これは、場所にあった通信路を利用することにより、大量のデータ転送と接続の継続性を両立させるためである。本節では必要に応じたインタフェースの自動切り替えに関して説明する。

4.3.1 概要

インタフェースの自動切り替えを行う場合、切り替えだけではなく、多くのことを考慮する必要がある。インタフェースの自動切り替えを行うためには以下のような機能が必要となる。

- インタフェースが利用できる状態かどうかを検査する
- インタフェースを利用できる状態にする
- 経路を変更する
- インタフェースを選択する

等の機構により実現する。

まず、インタフェースが利用可能かどうかを検査する機構について述べる。インタフェースが利用可能かどうかは、幾つかのネットワークデバイスではハードウェア、ドライバレベルでは検出できる。しかし、全てのデバイスで検出可能なわけではなく、それらのデバイスではより上位層での検出が必要となる。上位層で検出する場合は、ネットワーク層でのパケットの受信や到達性を利用することができる。パケットの受信ではインタフェースを監視しておき、パケットが届いていることによってインタフェースの接続性を確認する。イーサネット等のデバイスでは一般的に ARP や RIP、OSPF 等のブロードキャストやマルチキャストパケットが流れている。また、プロミスキュアス・モードにすることにより、他人宛のパケットを受信することもできる。このようなパケットが一定時間検出できなくなった時点で、インタフェースが利用不可能になったと判断する。また、パケットの到達性による検出では、指定したホストに対して ICMP echo 等のパケットを送出し、その応答を得ることによってネットワークが利用可能かどうかを判断する。この方法では、予めネットワークに関する設定が全て終わっている必要がある。このため、判断は 1) ネットワークの設定ができるか 2) 相手までの到達性があるかの 2 段階によっておこなわれる。今回の実装では、全てのインタフェースで利用できることを最優先とし、最後の手法、パケットの到達性によって検出をおこなった。

インタフェースを利用できる状態にするためには幾つかの設定が必要である。アドレスやネットマスクの設定である。インタフェースを設定する方法としては自動なもの

手動があるが、自動切り替えを前提としているため、当然、自動設定を考える。インターフェイスを自動設定するための機構としては、現在、DHCP が最も有力である。その他に BOOTP、RARP、IPCP 等の技術も存在する。本切り替え機構では、設定するインターフェイスによって DHCP と IPCP を利用する。設定するインターフェイスがイーサネット等のブロードキャストメディアの場合、DHCP を利用する。一方、PPP の場合には IPCP を利用する。インターフェイスが利用可能かどうかを検査する機構からのメッセージにより、自動的にどちらかのソフトウェアが起動されることになる。

次に、経路の設定について述べる。インターフェイスを設定した後、経路を正しく設定しなければパケットは送出されない。経路を設定するための機構としては、OSPF や RIP 等のプロトコルを利用してルーティングデーモンに委ねる方法がある。しかし、インターフェイスの接続先により利用しているプロトコルが異なる、あるいは経路情報交換プロトコルを利用していない場合がある。また、一般的に経路情報交換プロトコルを利用した経路切り替えは時間がかかる。このような問題を解決するため、本切り替え機構ではデフォルト経路を、IPCP や DHCP で指定されたものに変更することによりおこなうものとした。

最後にインターフェイスを選択する手法について述べる。インターフェイスを切り替えるにあたって、どのインターフェイスを利用するかを予め決めなければならない。これを決定するために本切り替え機構ではインターフェイスの優先度表を導入した。優先度表には全てのインターフェイスの優先順位を重み付けをして記述する。利用するプロトコルによって優先順位を変えることが可能なように、プロトコル毎に優先順位を表でもつ。優先度表の例を表 4.1 に示す。プロトコル毎にインターフェイスの優先順位を設定できるため、HTTP は無線 LAN のみで利用、TELNET は携帯電話を優先する、といった細かな設定が可能となる。ただし、プロトコル毎に利用するインターフェイスを変えるためには、経路制御機構の変更が必要となる。

表 4.1: 優先度表の例

プロトコル	ポート	イーサネット	無線 LAN	PHS
UDP	53	1	2	3
TCP	80	1	2	-
default	default	2	1	3

4.3.2 実装

本インターフェイス切り替え機構は、大きく次の 3 つの機能より構成される。

- 利用可能インターフェイス選択機能

- 利用可能インターフェイス設定機能
- ネットワーク接続/切断ハンドリング機能

利用可能インターフェイス選択機能は、通信開始時に利用するインタフェースを選択し、選択したインタフェースで通信を行う機能を提供する。TCP/IP モジュールからプロトコルとポートを受け取り、優先度表を基にインタフェースを選択し、選択されたインタフェースを基に決定した経路を TCP/IP モジュールに返す。インタフェースの選択では、通信の内容に応じて利用者が設定する優先度表を検索する。この選択の結果によっては新たな接続の確立が必要となるため、利用可能インタフェース設定機能に選択結果を通知する。また、選択したインタフェースで通信を行うために、利用可能インタフェース設定機能によって設定されるインタフェースと、ネクストホップゲートウェイの対応表を基に経路を決定する。

表 4.1に、インタフェースの選択に用いられる優先度表の例を示す。優先度表は、通信の内容を表すプロトコル、ポートの組み合わせに対して各インタフェースの優先度を設定し、値が小さい程優先度が高いことを表す。この例では、プロトコル UDP、ポート 53 である DNS では、イーサネット、無線 LAN、PHS の優先度で選択される。また、プロトコル TCP、ポート 80 である HTTP は、イーサネット、無線 LAN の優先度で選択され、PHS が利用可能な時でも選択されない。利用者が選択の優先順位を予め優先度表に設定することで、通信の内容に基づいた通信を可能にする。

利用可能インタフェース設定機能は次のような機能を持つ。

- ネットワーク接続の確立
- ネットワーク接続の確認
- ネットワーク接続の切断
- 選択の基準となる優先度表の設定

ネットワーク接続の確立は、インタフェースに応じて常時及び必要時に応じて行われる。常時とは定期的に接続の確立を試みることを表し、必要時とは利用可能インタフェース選択機能およびネットワーク切断/接続ハンドリング機能より送られる選択結果を基に必要性を判断し、接続を試みることを表す。DHCP を利用して接続するインタフェースは常時接続を試み、PPP を利用して接続するインタフェースは必要時に接続を試みることとする。この分類は、初期設定の段階で行う。ネットワークが利用可能になると、Mobile-IP の登録を開始し、登録の完了をもって接続の確立とする。

DHCP や PPP を用いて IP アドレスの設定及び経路表の設定を行いネットワークが利用可能になり、Mobile-IP の登録が完了することで接続が確立すると、利用可能インタフェース選択機能およびネットワーク切断/接続ハンドリング機能で経路の決定に利用されるインタフェースと、ネクストホップゲートウェイの対応を設定する。その後ネットワーク切断/

接続ハンドリング機能に、通信中のコネクションに対するインタフェースの再選択を要求する。

接続性の確立後は、定期的に接続性の確認を行う。接続性は、Mobile-IP のホームエージェントへのパケットの送受信を用いて確認する。登録要求に対して複数回連続で登録応答を得られないときに切断と判断する。

切断を検知した時には、ネットワーク切断/接続ハンドリング機能へ、通信中のコネクションに対する利用インタフェースの再選択を要求し、再選択が完了すると切断したインタフェースと、ネクストホップゲートウェイの対応を削除する。

また、利用可能インタフェース選択機能、ネットワーク切断/接続ハンドリング機能で、インタフェース選択の基準に用いられる優先度表の設定を行う。

最後に、ネットワーク接続/切断ハンドリング機能について説明する。本機能は利用可能ネットワーク設定機能よりネットワーク接続の確立、切断に伴うインタフェース再選択の要求を受け、現在通信中の利用インタフェースの再選択を行う。接続の確立時には、各コネクションの利用インタフェースの優先度と、新たに接続の確立したインタフェースの優先度を比較する。切断時には、切断されたインタフェースを利用しているコネクションで再選択を行う。

比較及び再選択の結果、利用インタフェースの切り替えが必要な時は、コネクションの経路を保持する情報を書き換えることで、利用インタフェースの切り替えを行う。選択結果によっては新たな接続の確立を必要とするので、利用可能インタフェース設定機能に選択結果を通知する。

再選択が完了すると利用可能インタフェース設定機能に通知し、切断時には対象となるインタフェースと、ネクストホップゲートウェイの対応が削除される。

4.3.3 考察

今回の実装では、パケットの到達性により、インターフェイスの状態を検出した。しかし、この手法では、確実に状態を把握することは難しく、また、検出時間も長い。実際には、オペレーティングシステムを介したインターフェイスの状態変化を伝える機能を用意し、これを利用するのが望ましい。イーサネットや無線 LAN 等のうち、ハードウェアによって状態変化を知ることができるものに関してはこれを利用し、そうでないものに関しては今回行ったような手法を用いる。このようにして検知された状態変化は、カーネルを通じて管理デーモンに、一つの統合されたインターフェイスで通知されるような機構が必要となる。

第 5 章

アプリケーション

5.1 地理的位置情報システム

地理的位置情報システムでは、ホストの地理的な位置情報の登録要求、地理的位置情報からのホストの検索要求、ホストの識別子からの地理的位置情報の検索要求、ならびに検索要求を受け付けて地理的位置情報データベースからの検索を行うための、機能部品を提供する。

本システムは、これらの機能を実現するために次のような構成となっている。1) それぞれのホストの地理的位置情報の登録要求を受け付け、地理的位置情報データベースで管理するエリアサーバ、2) エリアサーバにホストの地理的位置情報を登録要求するエージェント、3) ホストまたは位置を問い合わせるクライアント。地理的位置情報データベースでは、エンティティの識別子と地理的位置情報からなるエントリを管理する。クライアントは指定した位置を鍵として識別子を検索し、逆に指定した識別子から位置を検索する。

InternetCAR における地理的位置情報システムでは、*InternetCAR* に搭載されている様々なセンサー情報など、多様な情報収集および検索システムを構築するため、従来開発された地理的位置情報システム (GLI-00) を拡張し、GLI-01 を設計実装した。

GLI-01 地理的位置情報システムでは、クルマの地理的位置情報に加えて、クルマの多種多様なセンサー情報等の状態情報の登録要求、地理的位置情報からのクルマの検索要求、クルマの識別子からの地理的位置情報の検索要求、ならびに検索要求を受け付けて、地理的位置情報データベースの地理的位置情報テーブルと、移動情報テーブルからの検索を行うための機能部品を提供する。

システム全体の基本構造を図 5.1 に示す。

5.1.1 地理的位置情報の登録送信とデータベースへの登録 (エージェントとエリアサーバ)

1. 位置情報取得 / GLI エージェント

GPS 機器、各種センサー等からのデータを GLI エージェントに集積する。

ここで取得するデータ種別は、緯度・経度・高度、速度 (北・東・上各方向)、車の走

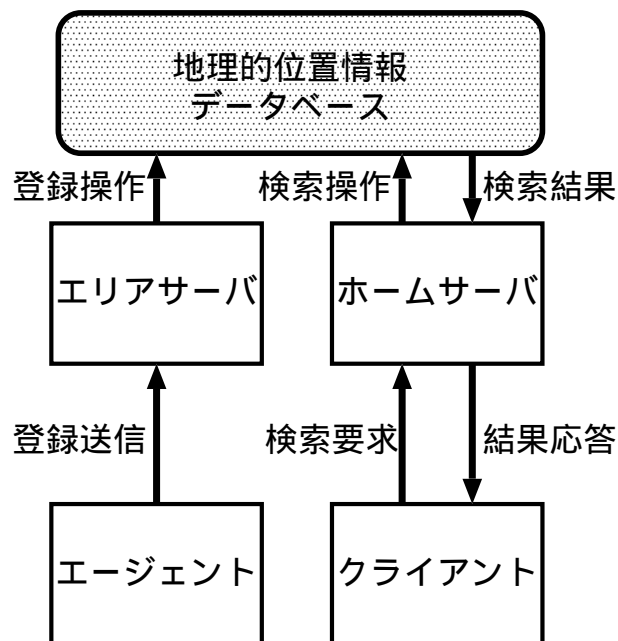


図 5.1: GLI-01 基本構造

行状態を示すデータとして車速パルス、ワイパー、ヘッドライトの動作、シフト位置などである。

2. GLI パケット生成 / GLI エージェント
集積された各種データを以下の図に示したパケットフォーマットに変換する。
バイトオーダーの変換 (ホスト → ネットワーク) はここで行なう。
3. GLI パケット送信 / GLI エージェント
生成された GLI パケットを、GLI エリアサーバに対して送信する。
エージェント・エリアサーバ間通信には UDP を用いる。
4. GLI パケット受信 / GLI エリアサーバ
GLI エージェントから送信されたパケットを、GLI エリアサーバが受信する。
5. GLI パケット解析 / GLI エリアサーバ
受信された GLI パケットから、各種データ値を取り出す。
バイトオーダーの再変換 (ネットワーク → ホスト) はここで行なう。
6. DB レコード作成 / GLI エリアサーバ・DB
取り出されたデータ値を、以下に示す DB レコード形式に変換する。

0-3	送信元 IP アドレス
4-7	データ取得時刻 [UNIX 時]
8-11	送信データ種別個数
12	データ種別 ID(2:位置)
13	データ種別サブ ID(3:緯度経度)
14	測地系 (2)
15-16	緯度 [度]
17	[分]
18	[秒]
19-20	[ミリ秒]
21-22	経度 [度]
23	[分]
24	[秒]
25-26	[ミリ秒]
27	データ種別 ID(3:高度)
28-29	高度
30	データ種別 ID(4:移動速度)
31	データ種別サブ ID(2:East,North,Up)
32-33	速度 [東方向:m/s]
34-35	[北方向:m/s]
36-37	[上方向:m/s]
38	データ種別 ID(5:ワイパー)
39	ワイパー状態 [0:OFF,1:INT,2:LOW,3:HI]
40	データ種別 ID(6:ヘッドライト)
41	[0:OFF,1:SMALL,2:HEAD]
42	データ種別 ID(9:外界照度)
43-44	外界照度
45	データ種別 ID(10:外界気温)
46	外界気温

表 5.1: 標準的な GLI パケットフォーマット

glirecord							
IP アドレス	取得時刻	速度 (東)	速度 (北)	速度 (上)	ワイパー	ヘッドライト	シフト
		エンジン回転	外界照度	外界温度	車間距離		
gliposition							
IP アドレス	取得時刻	データ種別 ID	測地系	経度	緯度	高度	
gliroom							
IP アドレス	取得時刻	教室 ID					

表 5.2: レコードセット

具体的には、SQL 文を生成して DB へ挿入するための用意をする。

ここで DB に対しての GLI システムの独立性を保持し、DB の機種が交換されても GLI システムへの影響が最小限にとどめられるようにする。

7. DB レコード挿入 / DB

DB レコードを DB に挿入する。前の段階で生成された SQL 文を DB に加える。

5.1.2 地理的位置情報の検索 (クライアントとホームサーバ)

地理的位置情報の検索は、ユーザからの検索要求をホームサーバに対して送信するクライアントと、クライアントからの検索要求を受信して地理的位置情報データベースを検索し、その結果をクライアントに送信するホームサーバからなる。

1. 検索要求

検索クライアントでは検索要求送信にあたって、検索形式の指定と検索キーとなる引数の設定、および検索オプションの設定を行う。

ユーザから入力させるデータは、検索の形式および引数の指定、および検索オプションの指定である。それぞれには、コマンド ID とサブ ID が定義されている。この、コマンド ID、サブ ID と引数を含めて、1 単位の検索コマンドとする。検索コマンドの概要を以下に示す。

(a) 検索形式と引数の指定

cmd	sub	検索形式の種類	検索キーとなる引数	引数の型
2	2	1 点指定最短距離検索	1 点 (緯度・経度・(高度))	(未実装)
2	3	2 点指定矩形範囲内検索	2 点 (緯度・経度・(高度))	GLI_Position 構造体
2	4	1 点+半径指定円内検索	1 点 (緯度・経度・(高度))、半径	(未実装)
2	5	識別子からの検索	移動ホストの識別子	unsigned long

(b) 検索オプションの指定

cmd	sub	検索オプションの種類	引数	引数の型
3	2	検索範囲を時間で指定	時刻 t1, t2	unsigned long
3	3	検索レコードの個数を指定	レコード数	unsigned char

これらの検索コマンドの設定から、検索要求の送信に必要なパケットを生成する。パケットの基本的な構造を表 5.3 に示す。

表 5.3: ホームサーバへ送信されるパケットの構造

0	バージョン番号 (1)
1	レコードタイプ (1)
2	検索コマンドの個数
3	予約済 (未使用)
4-7	クライアントホストの識別子
8-	データ

パケットのデータ部は、1 個以上の検索コマンドの連続したバイト列が挿入されている。

全ての引数、GLI_Position 構造体内部の変数は、それぞれ必要に応じて個別にネットワークバイトオーダに変換する必要がある。

2. 検索操作

ホームサーバはクライアントから送信される検索要求を受信し、パケットを解釈して、検索コマンドと引数から SQL 文を記述し、地理的位置情報データベースを検索する。

以下にデータベースに出力される検索要求の文を示す。

```
SELECT * FROM gliposition WHERE (Lng < 127530000 and Lat > 501810000
    and Lng > 126620000 and Lat < 502011000 ) ORDER BY UTime DESC
```

この例では、テーブル gliposition から 2 点指定矩形範囲内で検索し、時間の大きい順に並べることを意味している。

```
SELECT * FROM gliposition WHERE (IPaddr = 177189579) ORDER BY UTime
    DESC
```

この例では、テーブル gliposition から移動ホストの識別子 (IP アドレス: 203.178.143.10) で検索し、時間の大きい順 (新しい順) に並べることを意味している。

3. 結果応答

ホームサーバは、検索の結果発見されたレコードの数とレコードの連結からなるパケットを、クライアントに対して応答する。パケットの構造は表 5.4 に示す。

表 5.4: クライアントへ送信されるパケットの構造

0	バージョン番号 (1)
1	レコードタイプ (1)
2	発見されたレコード数
3	予約済 (未使用)
4-45	レコード 1
46-87	レコード 2

レコードは、地理的位置情報データベースに作成された、地理的位置情報のテーブルにしたがって格納されたデータのエン트리である。自動車の地理的位置情報のデータのレコード 1 単位は、gliposition テーブルと glirecord テーブルをつないで、重複する部分を削除したもので、42 バイトである (GLI_Record 構造体)。レコード部分の構造と内容は、表 5.5 に示す。

4. データベースのテーブル

エリアサーバで地理的位置情報の登録に使用される地理的位置情報データベースは、ホームサーバで参照される。

外部プログラムとして用意する地理的位置情報データベースとしては、フリーソフトウェアの簡易 SQL データベースサーバである mSQL2.0.3 を使用する。mSQL では、glidb という名前のデータベースを設定し、そのデータベースにテーブルを作成する。テーブルは登録するデータの種類と型からなるレコードセットを設定する。本システムでは、基本的な地理的位置情報のテーブルとして gliposition、センサー情報などの状態情報のテーブルとして glirecord を設定する。それぞれのテーブルのレコードセットを、表 5.6 に記述する。

5. 結果応答受信および表示

クライアントはホームサーバでの検索結果応答を受信する。結果応答は、検索の結果発見されたレコードの数と、レコードの連結からなるパケットに生成される。パケットの構造は表 5.4 に示す。

レコードは地理的位置情報データベースに作成された、地理的位置情報のテーブルにしたがって格納されたデータのエン트리である。自動車の地理的位置情報のデータのレコード 1 単位は、gliposition テーブルと glirecord テーブルをつないで、重複する部分を削除したもので、42 バイトである (GLI_Record 構造体)。レコード部分の構造と内容は、表 5.5 に示す。

受信した検索結果はレコードごとに表示を行う。

表 5.5: パケットのレコード部分 1 単位

0-3	発見された IP アドレス
4	データ種別サブ ID(3:緯度経度)
5	測地系 (2)
6-7	緯度 [度]
8	[分]
9	[秒]
10-11	[ミリ秒]
12	データ種別サブ ID(3:緯度経度)
13	測地系 (2)
14-15	経度 [度]
16	[分]
17	[秒]
18-19	[ミリ秒]
20-23	高度
24-25	速度 [東方向:m/s]
26-27	[北方向:m/s]
28-29	[上方向:m/s]
30	ワイパー状態 [0:OFF,1:INT,2:LOW,3:HI]
31	ヘッドライト [0:OFF,1:SMALL,2:HEAD]
32	シフトポジション (未使用)
33-34	エンジン回転数 (未使用)
35	外界照度
36	外界気温
37	車間距離
38-41	データ取得時刻 [UNIX 時]

表 5.6: 地理的位置情報データベースのテーブルとレコードセット

glirecord							
IP アドレス	取得時刻	速度 (東)	速度 (北)	速度 (上)	ワイパー	ヘッドライト	シフト
		エンジン回転	外界照度	外界温度	車間距離		
gliposition							
IP アドレス	取得時刻	データ種別 ID	測地系	経度	緯度	高度	

5.2 スプールベースのファイル転送機構

クルマからの無線網を使ったネットワーク接続では、切断がおきるような状況になることが容易に予想できる。例えば、クルマがトンネル内を通過するような場合、携帯電話が利用できない状態になるため接続が切れる。*InternetCAR* ではこのような時でも処理を継続して行い、次にネットワークに接続された時に自動的にデータ転送をおこなう仕組みを、スプールベースのファイル転送機構として実装した。本機構の構成を図 5.2 に示す。

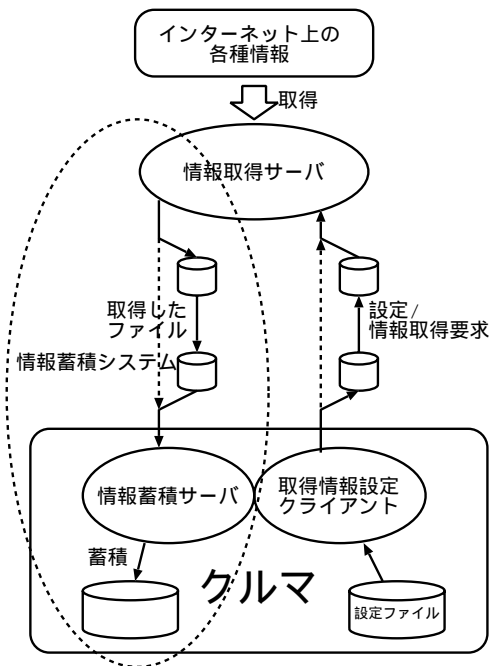


図 5.2: スプールベースのファイル転送機構の基本構造

本機能は以下のような機能より構成されている。

取得情報設定機能

ユーザの設定した取得する情報に関する要求および設定を読み込み、これをクルマから情報取得サーバに送信する取得情報設定クライアント

情報取得機能

取得情報設定クライアントからの要求および設定を受け付け、これに従って各種情報を取得する情報取得クライアント

情報蓄積機能

取得した情報をクルマに送信し、蓄積する情報蓄積サーバ

取得情報設定クライアントの扱うユーザ設定は、ファイル取得/コマンドの実行などの情報の種別、取得する情報の存在する場所、ファイルの場合は取得するファイル名、実行/情報取得した後、次の実行/取得までの待ち時間からなる。

本機構は、クルマでの低速/高遅延/不安定である通信環境をカバーするため、情報の一時蓄積機能を実現するものとして TCP/IP 上で動作する UUCP システムを使用したアプリケーションプログラムとして実装されている。

スプールベースのファイル転送機構が参照/作成/管理する設定ファイル等の種類と形式については以下の通りである。

取得情報要求の為の設定ファイル

本設定ファイルは、取得情報設定クライアントが使用するファイルである。設定内容は、次のように、取得する情報に対する要求を記述する。

```
[TYPE] [PROTOCOL or COMMAND] [LOCATION] [FILENAME] [INTERVAL]
```

各エントリの内容と取り得る値は以下の通りである。

TYPE: COMMAND, FILE

情報の種類を記述する。COMMAND が選択された場合には、次のエントリで指定されたコマンドの実行と解釈する。FILE が選択された場合には、ファイルの取得と解釈する。

PROTOCOL or COMMAND: HTTP(http), FTP(ftp), (Command)

情報取得の為に仕様するプロトコル、実行結果するコマンドを記述する。プロトコルとして記述できるものは、FTP, HTTP の 2 種類である。コマンドとして記述する際は、情報取得サーバ上で動作可能である `exec1p()` 関数で実行可能な UNIX コマンドに限定される。

LOCATION: 取得する情報の存在する場所を記述する。

HTML ドキュメントの場合は、`www.sfc.wide.ad.jp/~peropero`

FTP ファイルの場合は、`bash.cc.keio.ac.jp:/pub/.x1/GNU`

のように、ファイル名を含まない、ファイルパスを含むアドレスを記述する。要求がコマンドの実行である場合には、コマンドの第一引数を記述する。

FILENAME: 取得するファイル名

`index.html` や `hello-1.3.tar.gz` のように、取得するファイル名を記述する。要求がコマンドの実行である場合には、コマンドの第二引数を記述する。

INTERVAL: 0 以上の整数

ある情報を取得した後、次回に同じ情報を取得する迄の時間を、秒単位で記述する。

UUCP のための設定ファイル

本設定ファイルは、UUCP を使用して情報取得要求を送信する為に取得情報設定クライアントが使用するファイルである。

設定ファイルは、以下の順番で記述される。

1. UUCP で設定されている、取得情報設定クライアントのシステム名
2. UUCP で設定されている、情報取得サーバのシステム名
3. UUCP で設定されている、スプールディレクトリ
4. 取得情報設定クライアントが生成し、情報取得サーバに送信され、情報取得要求が記述されたファイル名
5. UUCP の実行ファイル (`uucp`) が存在するディレクトリ

また、本機構は、以下のような手順で処理を行う。

1. 初期設定

初期設定では、UUCP を使用する為の設定ファイルの読み込み、ユーザが作成した情報取得要求ファイルの読み込みを行う。

2. UUCP の実行

UUCP の為の設定ファイルの内容を基に、情報取得サーバに情報取得要求ファイルを送信する為、`exec1p()` を実行し、起動した UUCP を使用してファイルを送信する。

3. 終了処理

本機能内で使用した一時ファイルを削除し、終了する。

また、本機構はリモートでのコマンド実行も可能である。この機構はファイル転送機能の延長として実装されている。

5.3 サンプルアプリケーション

ここでは、慶應義塾大学湘南藤沢キャンパスにおいて 97 年度秋学期に開講された、情報処理 IIb(N) の講義¹において実施された課題とその作品の紹介を行う。

本課題は、*InternetCAR* に代表される「インターネットに接続された動くオブジェクト」が実用化された状況を想定して、そのオブジェクトによって提供される情報等を利用する、アプリケーションのサンプルを作成することを目的としている。

課題の内容は以下の通りである。

¹この講義を担当したのは、慶應義塾大学環境情報学部 村井 純 教授である。

以下の定義 (仕様) を満たす「動くオブジェクト」を実現する

- 位置情報 (緯度/経度/高度等) を有する。
- その他の何らかの情報を有する。
 - 車の場合 … 車速・ワイパーの ON/OFF 等
 - 人の場合 … 心拍数・血圧・体重・身長等
 - その他
- インターネット上のサーバに、上記の情報を一定間隔で送る機能を有する。
- 10 以上のオブジェクトが同時に動くことができる。
- 必ずしも常にインターネットに接続されているわけではない。
 - ネットワークに繋がっている場合 … サーバに情報を送り続ける。
 - ネットワークから切れている場合 … 次に繋がるまで情報を保持する。

以下、本課題に対する優秀作品である、「どこかな?かなちゃん」、「地図を描こうよ」、「信号を制御しよう」の 3 つを紹介する。これらはいずれも、動くオブジェクトとして車を想定した作品である。

5.3.1 どこかな?かなちゃん

1. コンセプト

交通システムとしてのバスを動くオブジェクトとし、インターネット上で各バスの現在位置をリアルタイムに知ることができる、バス交通視覚化システムである。本システムを導入することにより、次のようなメリットを期待することができる。

- 遠隔地のバス利用者がスムーズにバスに乗車することを可能にする。
- 災害時、緊急時のバスの運行ルートを調べることを可能にする。
- バス停以外からバスを利用したいユーザーへの援助を可能にする。
- どうしてもスヌーピー号に乗りたい人への援助を可能にする。
- 自分が待っているバスの混雑具合がわかることを可能にする。

2. サンプルイメージ

図 5.3 参照。

3. 製作

近藤 智則・根井 大樹・岩井 将行

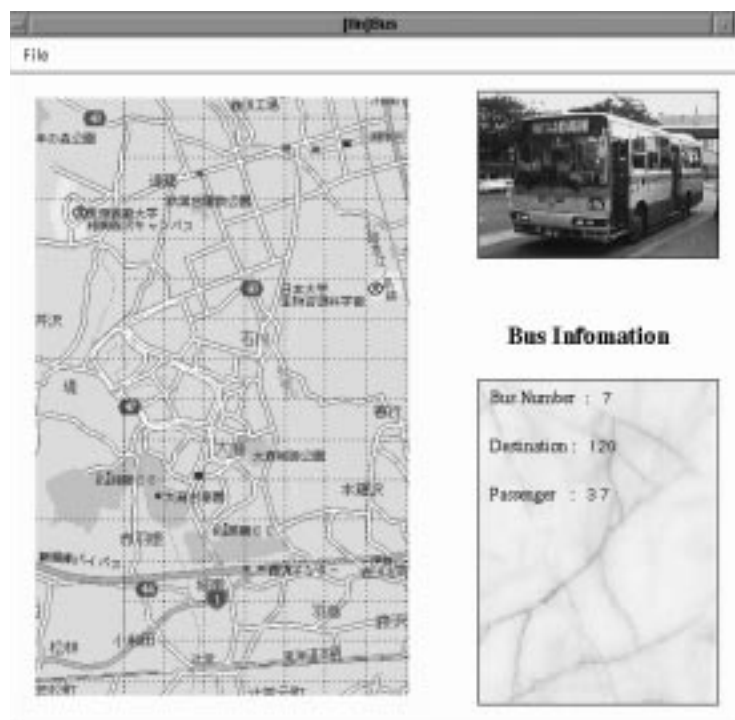


図 5.3: どこかな?かなちゃん

5.3.2 地図を描こうよ

1. コンセプト

動くオブジェクトとして 64 台の車を想定し、個別に移動する各車から提供される位置情報を元にして地図を作成する。地図の作成手順は以下の通り。

- 各車から一定間隔で位置情報 (緯度/経度/高度) をサーバに送る。
- クライアントはサーバから位置情報を取得し、地図上に記入する。
- 面が描けたら、面の傾き具合によって色をつける。

2. サンプルイメージ

図 5.4 参照。

3. 製作

芦田 宙・楠本 晶彦・板橋 清馬・千葉 竜介・穎原 桂二郎・竹野 祐樹

5.3.3 信号を制御しよう

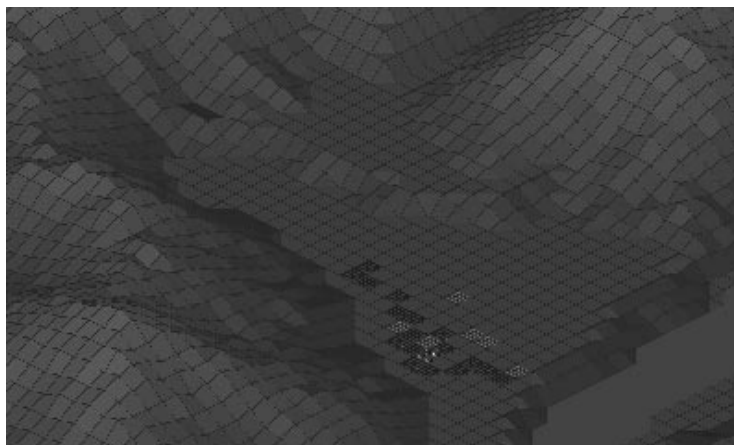


図 5.4: 地図を描こうよ

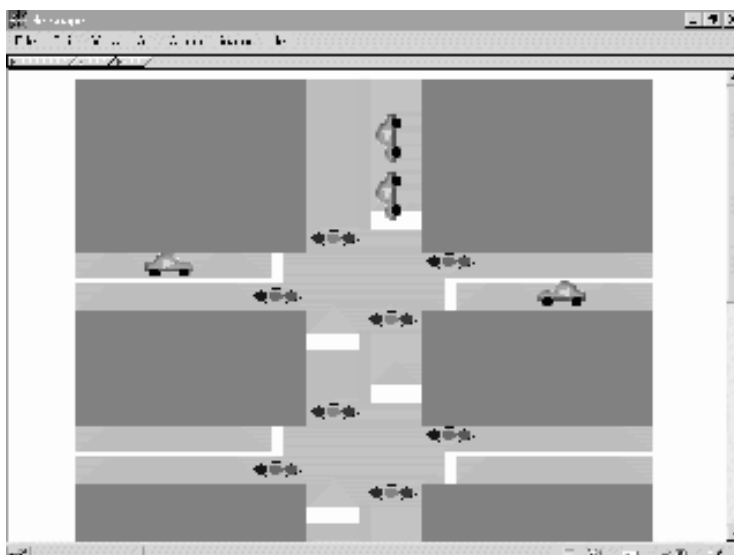


図 5.5: 信号を制御しよう

1. コンセプト

走っている車から位置情報、速度情報、ウィンカー情報等を得てサーバで集計し、道路の込み具合等に応じて、信号の動作を適切な形で制御するシステムである。本システムは次のようなケースにおいて有効となることが期待される。

- 交差点で明らかに片方向のみが渋滞している場合に、信号制御が固定的なため更に渋滞を助長させてしまうケース。
- 信号間の連携がとれていないために、信号と信号の間で渋滞が発生してしまうケース。
- 深夜、交差する道に車がないにも関わらず、信号のために止まらなくてはならないケース。

2. サンプルイメージ

図 5.5 参照。

3. 製作

牧 兼充・大石 賢司・佐藤 雅明・林 亮・広石 透

第 6 章

評価実験

Phase2 においては、最終的に以下の 7 台の実験車を用意した。(図 6.1)

- 首都圏
 - ホンダオデッセイ (改) 1997 年式
 - 日産セフィーロ 1996 年式
 - トヨタカリブ 1997 年式
 - トヨタマーク II 1989 年式
- 近畿圏
 - 三菱レグナム 1997 年式
- 北陸圏
 - ホンダインテグラ 1996 年式
 - トヨタカローラ 1991 年式

本章では、上記の内の 5 台を使用して行なった評価実験の報告を行なう。

5 台の実験車には、それぞれ異なったソフトウェアを搭載した(表 6.1 参照)。各実験車には車載計算機として東芝の PORTÉGÉ300CT (MMX Pentium 133MHz, 64MB) を設置し、図 6.2 に示す実験環境を用意した。

この図で H1 は Mobile-IP のホームエージェント、VIP のホームルータ、PFS のマスターサーバを担当する。H2 は nttcp のサーバである。また、H3 は GLI のサーバを担当する。B1 は無線 LAN のベースステーション(有線—無線ブリッジ)である。この図では大きな楕円が携帯電話が支援する範囲を、小さい楕円が無線 LAN が支援する範囲を示している。無線 LAN が支援する範囲は車庫などを想定している。

この環境下で、次のような手順で実験を行なった。

1. 車庫において PC を起動し、無線 LAN を使って接続する。



図 6.1: 実験車 (一部)

表 6.1: 実験用 InternetCAR の構成

CAR #	移動体支援	IF SW	GLI	PFS
CAR1	東芝 MIP	自動	Yes	Yes
CAR2	Stanford MIP	手動	Yes	Yes
CAR3	Sun MIP	手動	Yes	Yes
CAR4	VIP	手動	Yes	Yes
CAR5	LWP	NA	Yes	Yes

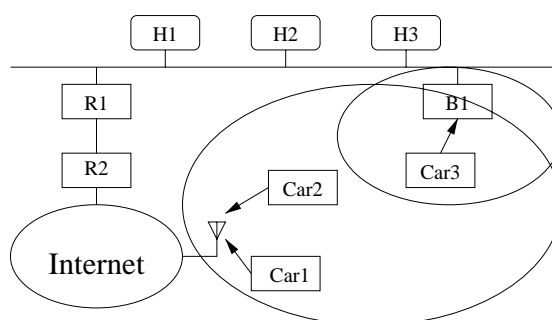


図 6.2: InternetCAR 実験環境

表 6.2: 切替え時間

通信環境の変化	切断検知	アドレス取得	全体
無線 LAN → 携帯	14.2 秒	9.60 秒	25.3 秒
携帯 → 無線 LAN	—	7.27 秒	8.14 秒

2. GLI と PFS を移動計算機上で起動する。
3. nttcp によってスループットを測定する。
4. クルマを走らせ車庫の外にでる。この際、自動的または手動で携帯電話による PPP に切り替わる。
5. もう一度 nttcp によりスループットを測定する。
6. H1 とクルマの間で ftp によるファイル転送を始める。
7. 車庫に戻る。無線 LAN 接続に切り替わる。
8. ファイル転送が終る。(切り替わった時の様子を記録)
9. もう一度 nttcp によりスループットを測定する。

また、実際に 1 台のクルマで藤沢から筑波まで往復し、その時のインターフェイスの状態を記録した。結果としてはトンネル、高架の下を走行中等は携帯電話が利用できないため PPP 接続の切断が観測されたが、その後の復旧などでは概ね良好な通進路を提供することができた。また、切替えにかかった時間を表 6.2 に示す。

第 7 章

今後の課題

これまでの実験により、幾つかの問題点が明らかになっている。前章までに述べたことも含めて、現在明らかになっている問題点は以下のようなものである。

OS の問題

現在の実験システムでは、オペレーティングシステムとして UNIX を使用している。このため、起動に時間がかかる、停車時に明示的にシャットダウンを行わなければならない、ファイルシステムが故障に弱い等の問題点を含んでいる。今後、操作をより簡単にして多くの人に利用してもらい、実験の規模を大きくするために UNIX ではない、より *InternetCAR* 環境に適したオペレーティングシステムの導入を検討している。

センサ情報収集ボックスの問題

現在のセンサ情報収集ボックスは電源部分の設計が弱く、クルマのエンジンスタータが回った時のノイズに耐えることが出来ない。このため、ハードウェアの再設計が必要である。

新しいハードウェア

現在の車載ハードウェアは、ノート PC、センサ情報収集ボックス、GPS アンテナ、携帯電話端末等から成る。これらがバラバラになっているため、取り付け作業や操作等が非常に複雑になる。この問題を解決するために、新しいハードウェアを設計する必要がある。

地理的位置情報・センシング情報利用環境の構築

地理的位置情報や、各車載センサから提供されるセンシング情報を利用するための環境を広く公開し、アプリケーション開発環境、実験環境を整える必要がある。また D-GPS 基盤環境の整備を進め、より高精度の地理的位置情報が利用可能となるようにする。

セキュリティ技術

アプリケーションが発達するに伴って、広く利用されるようになり、プライバシーの侵

害などにつながることを予想される。このようなことが起きないように、認証機構、アクセスコントロール機構を導入する必要がある。

Phase3 においては上に挙げたような問題点を解決し、より広く実験を行える環境を構築することを目標とする。

