

第XII部

自動車を含むインターネット環境の 構築

第12部 自動車を含むインターネット環境の構築

第1章 序章

インターネットを使った自動車の情報化は、ここに来て急速に進められている。現在、高度道路交通システム (Intelligent Transport Systems: ITS) の名の下に進められている情報化は、システムのあまりの巨大さのため、全体を掌握した設計をすることは非常に難しい。このため、各部が閉鎖的なシステムとなっており、新たなアプリケーションの参入は非常に難しい。新たなアプリケーションを投入する為には車載システム、通信システム、路側システムなどをトータルシステムとして設計する必要があり、膨大なコストが必要となる。この現状を打破する一つの有力なツールとしてインターネットが期待されている。

インターネットカーワーキンググループでは、より広い視点で情報化アーキテクチャの考察を目指した。今年度の研究では主に以下のような分野で活動した。

- 車載システム
 - － 車載計算機用 OS
- インターネットアクセス
 - － インターネット移動体通信
 - － ハイブリッド型インターネットアクセス
- ミドルウェア
 - － 地理位置情報サービス
 - － 車両情報交換プラットフォーム
- アプリケーション
 - － GNSS 補正情報配信
 - － 音楽情報配信
 - － アプリケーション開発環境

車載システムでは、主に車載計算機の基盤機能の開発を行った。自動車の置かれている環境は、通信状況や自然状況が刻一刻と変化する。多くのアプリ

ケーションは、このような環境の変化に対応した動作をする必要がある。このため、外界の変化をアプリケーションに伝える為の機構に関して研究を行った。

インターネットアクセスでは、これまで通り、複数の通信メディアを利用したインターネットアクセス機構について研究を進めた。特に今年度は自動車の規模性を考慮して IPv6 を基盤に研究を進めた。

ミドルウェアでは、セキュリティ機能を強化した地理位置情報システムの構築や車両情報交換プラットフォームの開発を行った。これにより、自動車を利用したアプリケーションの構築を支援する環境の整備を計った。

アプリケーションでは、前年度までに研究を進めて来た GNSS 補正情報配信システムの高度化を行っている。また、本報告には含まれていないが自動車情報化のためのアプリケーション開発支援環境の整備、音楽情報配信アプリケーションの構築、位置に基づくコミュニケーション支援のための表現形式に関する検討を行っている。

以上のような研究活動により、単なる「インターネットに接続された自動車」ではなく、それを取り巻くシステムの形成を進めることができた。

第2章 車載計算機用オペレーティングシステム

車載ハードウェア (SIC2000) で動作させるオペレーティングシステムとしては、NetBSD を利用している。本章では、オペレーティングシステムを中心として SIC2000 におけるソフトウェア環境について述べる。はじめに運用状況について説明し、続いて車載コンピュータ環境の特殊性に着目し、研究対象としての車載オペレーティングシステムについて述べる。

2.1 運用状況

2.1.1 開発環境

オペレーティングシステムの開発は PC アーキ

テクチャ上でクロスコンパイル環境を用いて行う。NetBSD-1.4.1 では FreeBSD 上でクロスコンパイル環境を構築し、これを利用してきた。NetBSD-1.5 では、NetBSD-1.5/i386 上で NetBSD のクロスコンパイル用パッケージ (/usr/pkg/cross/mipsel-netbsd) を用いることが可能である。

開発環境のボトルネックの一つとして無視できないものが、デバッグ、テスト段階におけるカーネルのコピー作業である。オンボードのフラッシュメモリの更新は多くの時間を要するため開発効率が著しく低下する。この問題を解決するためには、いくつかの方法が考えられる。一つは、ネットワークブートを利用する方法である。

2.1.2 ファイルシステム

SIC2000 では、オンボードのフラッシュメモリ (8MB) のみがストレージとして利用可能である。これまで、NetBSD の利用を通じてルートファイルシステムには RAM ディスクを利用してきた。これは、NetBSD カーネルのデータ領域にファイルシステムのイメージを格納し、これを仮想的にディスクとして利用するものである。

RAM ディスクの利用は、電源断が頻繁に生じる環境下での利便性は高いが、不揮発性の記憶装置ではないこと、また記憶容量が制限される等の不便な面もある。インターネット自動車では走行データの保存などに対する要求が高いため、不揮発性な上、ある程度の容量を持った記憶装置が必要となる。

そこで、ATA フラッシュメモリカードの利用が期待される。ATA フラッシュメモリカードは、最大で 128MB 程度の記憶容量を持ち、かつ振動に強い構造となっているため、利用価値が高い。現在、SIC2000 では ATA カードの利用 (wd デバイス) が利用可能な状況でない。ATA フラッシュメモリカードの利用を可能にすべく問題解決中である。

2.1.3 デバイスドライバ開発状況

今年度は、デバイスドライバの開発が進み、自動車からより多くの情報を取得することが可能となった。利用可能となったデバイスは、ADC、温度センサ、DIN、パルスカウンタの四つである。これらを用いて、車速、ワイパー、ライト、そして車載ハードウェアの温度が測定することができる。SIC2000 に搭載されたハードウェアおよび、実験用自動車 (本

田オデッセイ) から取得可能なセンサ情報のほぼ全てを利用することができる状況である。

ハードウェアデバイス	説明
ADC	アナログデジタルコンバータ
温度センサ	オンボードの温度センサ
DIN	デジタル入力
パルスカウンタ	車速パルスをもとに自動車の速度を測定する

2.1.4 NetBSD-1.5 へのバージョンアップ

インターネット自動車プロジェクトでは、昨年より NetBSD を車載オペレーティングシステムとして利用してきた。SIC2000 とアーキテクチャの似たシステム (hpcmips) をベースとして移植を行い、運用、開発を続けてきた。

NetBSD は、多くの他のオープンソースのオペレーティングシステムと同様に、常に開発が進められバージョンアップが繰り返されている。これによって、オペレーティングシステムとしての成熟度が高まり、かつ新たに開発されたハードウェアが利用可能となる。従って、インターネット自動車プロジェクトでは常に最新の NetBSD を SIC2000 に適応させていく努力が必要である。現在、NetBSD-1.5 の SIC2000 への移植作業を進めている。NetBSD-1.5 は、NetBSD プロジェクトより昨年 (2000 年) 末に公開された最新のリリースである。移植作業は、以下の流れに従って行った。

- クロスコンパイル環境の構築
- ライブラリ (/usr/src/lib) の build
- マシン依存コード (/usr/src/sys/arch/sic2000) の開発
- SIC2000 用 NetBSD カーネルの build
- テスト、デバッグ

NetBSD に含まれる mips アーキテクチャ共通のコードは前バージョンと同様に、わずかな変更によって、利用することが可能である。また、SIC2000 マシン依存部分のコード (src/sys/arch/sic2000) は、基本的には NEC Mobile Gear などの Handheld PC が、アーキテクチャとして非常に似たシステムとなっているため、これをベースとしている。SIC2000 マ

シン依存部分のコードは基本的に、前バージョンのものをそのまま利用した。

現在、カーネルの起動完了を確認している。しかしながら、PCMCIA コントローラの動作に不具合が生じており、現在この問題解決に当たっている。今回のバージョンアップによって、SIC2000 の利用環境が向上する点として、IPv6 の標準サポートや、PCMCIA カードの動的な抜き差しが可能となることなどが挙げられる。

2.2 研究対象としての車載オペレーティングシステム

これまで汎用オペレーティングシステムを車載ハードウェアに一年以上に渡って運用、開発を行ってきた。これらの経験から、車載コンピュータ環境に特有な問題を既存の汎用オペレーティングシステムで解決することは難しいことが分かってきた。特に、自動車アプリケーションの開発に着目した場合、自動車コンピュータ環境内で発生する様々なセンサ情報、およびそれから得られる意味情報の管理は、従来のオペレーティングシステムに求められるはたらかきでは十分に解決することができない。

2.2.1 車載コンピュータ環境の特殊性

自動車をコンピュータ環境として捉えた時に、その特殊性をいくつか指摘することができる。特に、デバイスとそれに対応するセマンティクス（意味情報）の関係が複雑であるため、自動車アプリケーションの開発は困難となっている。

センサやその他の車載コンピュータに接続されたデバイスから得られるあらゆる様々な情報から、何らかの意味情報を抽出する必要がある。意味情報の生成は、意味情報と、それに関連付けられるセンサデバイスなどのハードウェアをマッピングすることによって実現する。車載コンピュータの環境情報から意味情報を生成するに当たり、次の二つを考慮すべきである。

ハードウェア依存性

自動車に取り付けられたセンサデバイスは車種やメーカーによって異なる。例えば、乗用車とトラック、あるいはAT、MTといったトランスミッションの違いによって利用されるセンサの種類や特性は異なる。また、ABS やエアバッグなどの機能の有無によっても同様の差異が生じる。

マッピングの動的な変化

車載システムによっては、ハードウェアと意味情報のマッピングが動的に変化することが予想される。例えば、自動車の速度を意味情報の例として取り上げる。速度は一般的に車速に応じて発生するパルスをカウントすることによって得られるが、最近では高精度化したGPS装置を用いて自動車の速度を割出すことも可能である。GPSの利用可能範囲は限られているため、両者を状況に応じて使い分けることが考えられる。このように、意味情報は複数のハードウェアデバイスに関連付けられる場合、両者のマッピングは動的に変化することが考えられる。

2.2.2 車載コンピュータ環境のモデル化

オープンな自動車アプリケーション開発環境を実現するためには、オペレーティングシステムあるいはミドルウェアが、車載コンピュータ環境の特殊性によって生じる問題点を解決しなくてはならない。ここでは、まず車載コンピュータ環境のモデル化を行う。

車載コンピュータ環境からハードウェア依存性を取り除くためには、個々の自動車に依存した情報を隠蔽し、ソフトウェアから全ての自動車を共通のものとして抽象化する枠組（仮想自動車）が必要である。仮想自動車から得られる情報の変化を、アプリケーションはイベントとして検知する要求を持っている。また、仮想自動車の情報は、必ずしもハードウェアと一対一で対応するものではない。従って、ハードウェアデバイスと各仮想自動車情報のマッピングが必要となる。

また、仮想自動車層とアプリケーション層の間には、アプリケーションとそれに対応するイベントのマッピングが必要となる。これを受けて、車載コンピュータ環境における、アプリケーション、イベント、仮想自動車情報、ハードウェアデバイス各々のはたらかきを明確にし、階層化した。図2.1は、車載コンピュータにおいて、イベントに着目した抽象化モデルを示している。続いて、イベント定義層、およびハードウェア抽象化層の説明をする。

イベント定義層

アプリケーションは、仮想自動車の各情報の変化を察知する要求を持っている。イベントは、複数の

情報を関連づけて定義することが望ましい。そして、イベントはアプリケーションという単位でそれぞれ独自に定義可能であることが望ましい。仮想自動車層とアプリケーション層の間に、このイベントを定義し、管理する新たなレイヤが必要である。これをイベント定義層と呼ぶ。この層では、イベントをアプリケーションという単位で管理し、それぞれに独自に名前空間を与える。イベント定義層は、仮想自動車層の上位に位置するため、車載コンピュータやオペレーティングシステム、さらに自動車には全く依存しない。

ハードウェア抽象化層

車載センサや、車載コンピュータに接続された様々なデバイスは、必ずしも仮想自動車の一つ一つのオブジェクトと対応しない。これらの自動車情報は、自動車や車載コンピュータに依存することも前に述べた通りである。従って、リアルなデバイスを仮想自動車としてマッピングする仕組みが必要である。抽象モデルとして、この機構をハードウェア抽象化層と呼ぶ。



図 2.1. 抽象モデル

2.2.3 抽象モデルの具体化

車載コンピュータに適応させたイベント機構のための抽象モデルをさらに具体化し、それぞれの層での実際の役割や管理する実体について説明する。図 2.2 は抽象モデルを具体化したものである。

最下層のハードウェア層では、四角、三角、丸などで示されたデバイスが存在する。これらは前節で述べた環境情報を構成する実体であり、具体的には車載センサや車載システムのハードウェアデバイスである。ハードウェア層では、各々のデバイスの意味は、仮想自動車の構成要素と一対一に対応しない。ここで、ハードウェア層によってそのマッピングが実

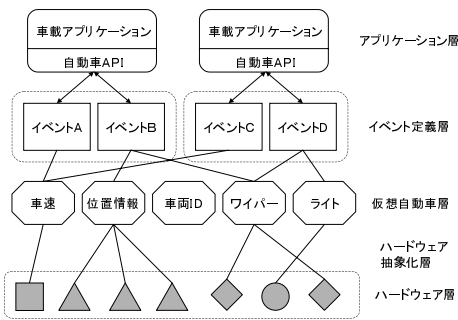


図 2.2. 具体化された抽象モデル

現される。このマッピングは、車載システムによって異なることが想定される。また、マッピングの際にデータの正規化が行われて、仮想自動車層で定義された情報に整理される。

仮想自動車層の上には、イベントが定義される。イベントは、仮想自動車層の各々の要素に関連付けられる。そして、イベントはアプリケーションを単位としてグループ化され、管理される。イベント定義層は、自動車アプリケーションの API を通じてアプリケーションからアクセス可能である。これによって、自動車アプリケーションは自分の要求に合ったイベントを定義することが可能となる。

このモデルに従って、車載コンピュータ環境に必要な非同期イベント通知機構の設計および実装を行った。ハードウェア抽象化層より上は、全てユーザ空間においてミドルウェアとして実装した。これによって、デバイスドライバよりも高い自由度の元に、デバイスと意味情報の柔軟な対応を実現し、意味情報の変化を各アプリケーションの要求に応じて通知することが可能となった。本システムを、車載ハードウェア上で動作させ、その有用性を確かめることができた。

第 3 章 自動車用通信システムの構築

3.1 はじめに

自動車を取り巻くシステムは高度に情報化されつつある。高度道路交通システム (Intelligent Transport Systems: ITS) は、道路、自動車、歩行者など道路上の要素を統合的にシステム化し、より先進的な交通

システムの実現を図っている。ITS 関連のプロジェクトは、主に政府主導でシステムの実現が進んでいるが、企業等が独自のコンテンツと提供方法を用い、ユーザに情報提供している例も多い。

現在、ITS の多くの技術は独立したシステムとして構築されている。有料道路自動料金収受システム (Electronic Toll Collection system: ETC) では、自動車との双方向通信システムやユーザ認証のための IC カードを利用し、有料道路の自動料金徴収を実現している。しかし、現時点では、ここで利用されている双方向通信システムや IC カードによる認証は、他の用途に応用することは難しい。また、携帯電話などを利用した双方向通信が可能なカーナビゲーションシステムが多く実現されているが、これらのシステムには統一された規格が存在せず、メーカ等が独自にコンテンツを用意し、情報提供をするシステムとなっている。

しかし、情報化が進む中、個別のシステムが独立して運用されるモデルは非効率的であり、情報価値の低下を招く。インターネットは、個別の情報としてはさほど意味を持たないものでも、多く集めることにより十分有用な情報環境をつくり出すことが可能であることを示した [101]。また、個別のシステムとして実現する場合、システム構築のコストの増大も問題となる。

本研究では、インターネットを利用した自動車情報化のための基盤システムであるインターネット自動車 (InternetCAR) システムの概念を構築し、その設計および実装を行う。インターネット自動車システムでは、1) 通信システムをサービスから分離し、2) 自動車のデータセットを定義することにより、サービス導入時のコストの削減を支援する。これにより、ITS 分野における自動車情報化の早期実現を目指す。

本章では、上記研究のうち、主に通信システムの設計・実装およびその評価について述べる。3.2 節では、インターネット自動車の通信部分の要件をまとめ、そのアプローチについて議論する。3.3 節では、実際にインターネット自動車の通信部分の設計・実装について述べる。

3.2 インターネット自動車の通信システム

自動車は高速で移動するため、一般的なインターネット環境とは異なった通信環境を持つ。本章では、インターネット自動車が置かれている通信環境に対

応した通信システムについて考察する。

3.2.1 ホストとしての自動車

まず、インターネット上で自動車をどのように表現するかを考える。インターネット上で自動車を表現する場合、以下の二つの方法が考えられる。

- エージェントとして取り扱う方法
- ホストとして取り扱う方法

前者は、自動車をソフトウェアとして表現する方法である。この方法では、自動車が通信できない状態にある場合でも、エージェントが応えられるという利点を持つ。反面、実際に自動車と通信する際にもエージェントを介して自動車の情報を取得しなければならない。

後者は、自動車を一つの通信ノードとして表現する方法である。この方法では、自動車が通信不可能な状態にある場合には、センタシステムとの通信が成立しない。しかし、センタシステムと自動車の間に入るものが無いため、アプリケーションの機能拡張や新規サービスの導入が、他のアプリケーションに依存せずに容易に行えるという利点を持つ。

インターネット自動車システムでは、新規サービスの導入を容易にするため、後者の方法を採用する。

3.2.2 通信デバイス環境

自動車は道路上を移動する。当然、有線の通信デバイスを利用できる状況は限られており、基本的には無線通信デバイスを用いることになる。しかし、現状では、あらゆる環境において利用できる無線デバイスは存在しておらず、無線デバイスはそのサービスエリアが限られている。また、狭域高速な通信デバイス、広域低速な通信デバイスなど、デバイスによってその特長も大きく異なる。

そこで、インターネット自動車システムでは複数の通信デバイスを切り替えながら利用する必要が生じる。これにより、ある通信デバイスのサービスエリアを出ても、他の通信デバイスのサービスエリア内であれば、継続して通信路を確保することが可能となる。

3.2.3 移動透過性

自動車をホストとして扱うこと、及び、自動車が

通信デバイスを切り替えながら通信することをインターネットの視点で捉えると、移動ホストの問題に帰着する。インターネットでは、IP アドレスによってホストを識別している。しかし、IP アドレスはホストが接続されている場所に依存しているため、接続場所が変化すると IP アドレスも変化する。これは、移動前と移動後で別のホストとして認識されてしまうことを表している。

前節で述べたように、インターネット自動車システム環境では、複数の通信デバイスを切り替えて通信を行う。また、通信デバイスの切り替えが無くとも、移動によってその接続先が変化することがあり得る。このような状況では、自動車の IP アドレスが変化し、移動の前後で別の自動車として認識されてしまう。

このような問題を解決するためには、Mobile IP[127]や Virtual Internet Protocol (VIP)[154] の様な、インターネット用の移動体通信プロトコルを利用する必要がある。これらの移動体通信プロトコルは、ホストの IP アドレスが変化しても、ホストを一つの IP アドレスで識別できる環境を提供する。これにより、自動車が利用する通信デバイスや、その接続先が変化しても、継続して相手の自動車と通信を行うことが可能となる。

3.2.4 インターネット自動車通信システムモデル

インターネット自動車の通信システムは、1) 一台の自動車を一つのホストとして取り扱い、2) その自動車は複数の通信デバイスを選択しながら、3) 継続して通信できる環境を提供する。通信部の階層モデルを図 3.1 に示す。サービスを実現するためのアプリケーションは、自動車を指定して情報を送受する。このため、自動車はそれぞれに自動車の識別子 (Host ID) を持つ必要がある。また、通信デバイスは、接続可能性を監視し、接続可能であればアドレスを取得して状態変化を制御部を通じてネットワーク位置管理部に通知する。位置管理部では通信可能なデバイスのリストから、自動車 (ホスト) のネットワーク的位置を決定し、継続した通信をアプリケーションに提供する。つまり、ホストは、アドレスの取得が完了しているインターフェースのリストから適当なものを選択し、それを利用できるように自分のネットワーク的位置の管理を行う。これにより、ホスト間の継続した通信が成立する。

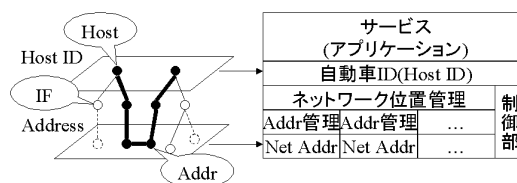


図 3.1. 通信システムの階層モデル

3.3 設計と実装

本章では、インターネットを利用した自動車情報化システム「インターネット自動車システム」の設計および実装について述べる。

3.3.1 通信システム

インターネット自動車システムでは、エンドエンドの通信を実現しなければならない。しかし、現実的に利用可能な無線通信デバイスは、携帯電話や無線 LAN 等に限られており、常に通信を保証することはできない。そこで、インターネット自動車システムの通信システムでは、複数の無線通信デバイスを用いて、可能な限り接続を維持することを考える必要がある。

しかし、現状のインターネットでは、複数の無線デバイスを切り替えながら通信を維持することはできない。そこで、インターネット自動車システムではインターフェース切り替え機構及び Mobile IP を組み合わせて用いることによってこの環境を実現する。図 3.2 に自動車側システム (車載機) の通信部のブロックダイアグラムを示す。通信部は、インターフェース毎にアドレス管理がなされている。更に Mobile IP や経路制御表を利用したインターフェース切り替え機構により、どのインターフェースを利用するかを決定するように構成されている。アドレス管理は、通信デバイスに応じて PPP のアドレス取得サブプロトコルである IP Control Protocol (IPCP) や DHCP によって行われている。

Mobile IP は、現状ではほとんど利用されていない

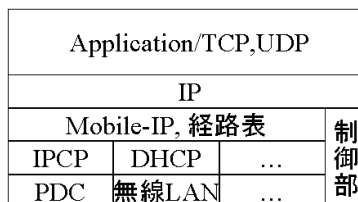


図 3.2. 車載機通信部のブロックダイアグラム

いが、標準化が行われている技術である。一方、インターフェース切り替え機構は、特に標準として定められているものは存在しない。手動で IP 層の経路表を書き換えることによってインターフェースを切り替えることも可能である。今回は、自動車環境を考慮し、高速に自動でインターフェースを切り替える機構を構築する必要がある。

図 3.2 は、図 3.1 の通信システムをインターネットに適用した例を示している。制御部が無線 LAN デバイスから電波強度等の情報を受け取り、その強度に応じて経路を変更し、また、Mobile IP の Home Agent (HA) への位置登録を指示する。ここで、携帯電話側はパケット通信を仮定した常時接続を行っており、Point-to-Point Protocol (PPP)[144] ソフトウェアによって再発呼等の処理を行う。

今回、実装はインターネット自動車システム用に開発した車載ハードウェア SIC2000[190]を用いて行った。SIC2000 は、CPU に VR4300 を搭載した計算機で、センサ信号入力用の A/D、D/D インターフェースやパルスカウンタ、携帯電話等を接続するための PC カードスロット、Ethernet インターフェース、内蔵 GPS ユニット等を備えている。自動車で利用することを前提としているため、DC12～24 V の電源で動作し、ACC スイッチ (一般的にはイグニッションキー) による電源制御が可能である。また、カーオーディオサイズ (1 DIN サイズ) のコントローラを接続できる。SIC2000 の取り付け時の写真を図 3.3 に示す。左側がコントローラ、右側が SIC2000 本体である。

実験のために SIC2000 上では NetBSD を動作させ、その上にネットワーク環境を構築した。実際の実装に沿ったブロックダイアグラムを図 3.4 に示す。実装では、PPP のモジュールとしては iij-ppp を常時接続モードで利用した。また、インターフェース



図 3.3. 取り付け時の SIC2000

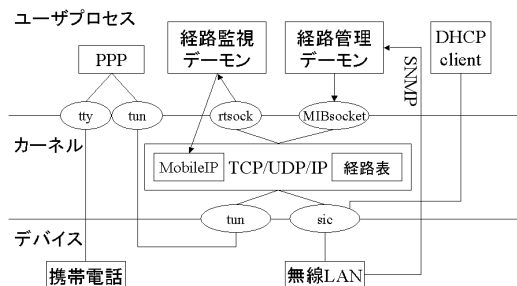


図 3.4. 通信部の実装

切り替え部分は MIBsocket[198] を利用して構築した。MIBsocket は、ネットワークエンティティの状態変化をアプリケーションに伝えたり、状態変数を変化させるのに利用できる。経路管理デモンはイーサネットに接続された無線 LAN デバイス (Breezecom) から Simple Network Management Protocol (SNMP)[137] により、電波強度を定期的に取得する。このとき、電波強度が閾値より低くなると、MIBsocket を介して経路表を書き換え、携帯電話を利用するように設定する。逆に、携帯電話を利用しているときに電波強度が強くなると、無線 LAN を利用するように設定変更する。経路監視デモンは、経路の変化を MIBsocket からのイベントにより検知し、Mobile IP の CoA (Care of Address) 登録を行う。

この機構により、自動的な通信デバイス選択と通信の継続を実現し、自動車の継続したインターネット接続を支援する。

3.3.2 動態管理アプリケーション

今回の実装では、アプリケーションとして、自動車の情報をセンター側でモニターする動態管理を実現する。

動態管理を実現するためには、車載機側で、情報を取り込むソフトウェアと外部に情報を提供するソフトウェア、二つのソフトウェア間のインターフェースが必要となる。とくに、二つのソフトウェア間のインターフェースの構築は他のアプリケーションを実現する上で重要となる。

今回は、情報を取り込むソフトウェアに User Datagram Protocol (UDP) を用いた、ネットワークを介して情報提供が可能なインターフェースを設けることにより、アプリケーションを実現した。ネットワークを介した情報提供インターフェースを構築する事により、アプリケーション依存部を車載機側、インターネット側のどちらに構築する事も可能となり、

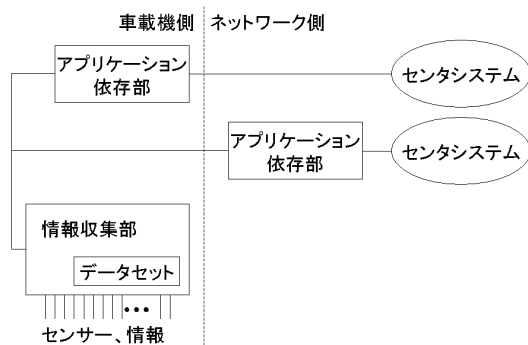


図 3.5. 情報システムのアーキテクチャ

アプリケーション設計者の自由度を高める事を狙いとしている。情報システムのアーキテクチャを図 3.5 に示す。このアーキテクチャでは、自動車をデータセットとして取り扱い、これを利用してサービスを実現する。

今回の実装では、アプリケーション依存部と情報収集部を一つのソフトウェアとして実装した。既存のアプリケーションは情報収集部を設計する段階で取り込むことが可能であり、その方が効率的な実装が可能である。実際に作成したソフトウェアでは、情報収集部組み込みの場合、変数に直接アクセスする事が可能であり、外部プログラムとして実装される場合は、UDP ネットワークインターフェースを介して情報にアクセスできるようになっている。

車載側情報システムの実装を図 3.6 に示す。この実装では、データセットとスケジューラを軸に、データセットの情報を更新する幾つかのモジュールと、それらの情報をもとにサービスを提供するアプリケーションモジュールから構成されている。ソフトウェア内部に組み込まれたソフトウェアはデータセットに直接アクセスする事が可能である。また、後に追

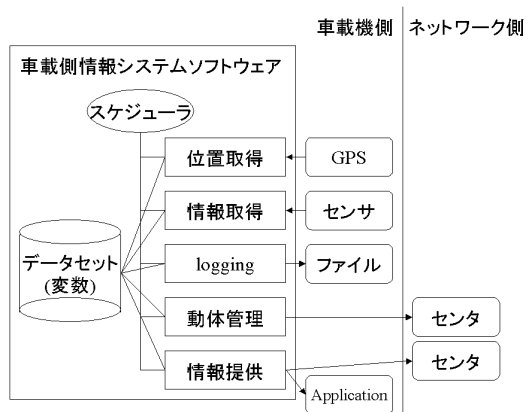


図 3.6. 車載側情報システムの実装

加するモジュールについては、情報提供モジュール (UDP/IP を用いてデータセットにアクセスする機能を提供するモジュール) を介してデータセットにアクセスする。

3.4 今後の課題と方向性

本章で示したモデル及び実装では、通信路の安全性や通信性能の保証については言及しなかった。しかし、自動車の情報化においては安全に関わる情報、課金情報、プライバシーに関わる情報等、重要度の高い情報が取り扱われる。このため、通信相手の認証と通信路の安全性の保証、頑健な通信路が必要となる。

現在、インターネットでは、相手の認証を行うための基盤が整備されつつある。認証局 (Certificate Authority: CA) に予め公開鍵を登録しておき、CA を介してお互いを認証する方式である。また、IPsec[97] によって、通信路の安全性も保証できる段階にきている。これらの枠組を早い時期に取り入れることによって、安全な通信が可能となる。

一方で、通信性能の保証は現在のインターネットでは難しい。しかし、Diffserv[16, 33] 等の技術により、ある範囲の通信路の通信性能を保証する事は可能になりつつある。重要な情報を交換する部分に適度な通信インフラを整備し、これらの技術を適用することにより、インターネットの利点を失うことなく、通信性能を保証することができる。

更に、現在インターネットではあまり利用されていないような通信メディアの利用を考慮する必要がある。自動車には、これまで利用されてきたデジタル MCA の様な通信メディアが存在している。また、自動車に対する通信メディアとして衛星通信やデジタル FM 放送等が期待されている。自動車での通信をインターネットで抽象化するためには、これらの新しい通信メディアをどのように利用するかを検討する必要がある。

また、規模性に関しても考慮する必要がある。全ての自動車がインターネットに接続されると、日本国内で約 7,300 万台 (1997 年調べ)[175] のホストがインターネットに接続されることになる。インターネットでは、IP アドレスの枯渇問題が発生しており、より大きなアドレス空間を提供するため、IP version 6 (IPv6)[35] への移行が進んでいる。自動車をインターネットに接続する際には、IPv6 の利用も考慮する必要がある。

更に、標準化も重要な鍵となる。自動車の情報化を効率的におこなうためには、全ての自動車が同じ基盤上にある事が望ましい。これにより、新たなサービスを導入する際の導入コストを削減することができ、また、より多くの自動車をそのサービスの対象とすることが可能となる。

3.5 結論

本章では、インターネットを用いた自動車の情報化手法を提案した。現在の自動車の情報システムは、主にサービスを中心とした一つの閉じたシステムとして設計されている。しかし、これらのシステムは共通する部分も多く、その部分を抜きだして情報化のための基盤技術とすることにより、新たなサービスを導入する際のコスト削減ができる。本研究では、現状で共通基盤をつくることのできる通信部分の分離をおこない、その実現方法を示した。

我々の手法では、現在最大のデジタル情報基盤であるインターネットを利用しており、既存の通信デバイスの有効利用や発展性の面でも優れている。既に敷設されている様々な通信デバイスを利用してシステムを構築することが可能であり、徐々にシステムを大きくしていくことができる。また、インターネットを基盤とした情報家電等の製品化も進んでおり、これらとの協調システムを構築するにあたって我々の手法は有用である。

第4章 IPv6 による通信システムの構築

4.1 始めに

IPv4 アドレスの枯渇に伴う IPv6 の導入により、車や携帯電話といった計算機以外の多様なデバイスに IP アドレスが割り当てられ、インターネットに接続されることが想定される。例えば、車をインターネットに接続した場合、車は広域を高速で移動するため移動中の継続的な通信の実現が必須になる。車に対しての定期的な渋滞情報、GPS を用いた位置補正情報、そして音楽の配信等が考えられているからである。

車は広域を高速で移動するため、車をインターネットに接続する場合、以下の3つの要素が実現される

必要がある。1つ目は、移動体計算機が常にネットワークに接続する常時接続性である。2つ目は、移動透過性であり、アプリケーションから移動を隠蔽する。3つ目は、着信可能性であり、移動体計算機は一意なアドレスを用いてパケットの着信が行える。着信可能性により、通信相手は移動体計算機の一意なアドレスを用いて、常に通信の開始を行うことが出来る。現在のインターネットでは、ネットワークを移動することにより IP アドレスが変化する。IP アドレスの変化に伴い、通信が遮断され、次に通信相手は移動体計算機のアドレスを特定できないため通信の再開が行えない。車に限らず移動する計算機は、上記の問題が起こり得る。

これらを解決する研究としては、Mobile IPv6 がある。Mobile IPv6 は、移動透過性と着信可能性を提供する。Mobile IPv6 により移動体計算機はインターネットにおいて、既存の固定体計算機に近いネットワーク利用が可能となる。Mobile IPv6 を用いることにより、移動体通信環境における多くの問題点を解決することが出来る。

しかし、常時接続性の問題は Mobile IPv6 だけでは解決できない。今後、インターネットに接続される計算機や機器は広域を移動すると予測される。それらに対して、インターネットへの常時接続を実現するためのシステムは未だ存在しない。現状ではすべての状況において利用可能なインターフェースは存在しないため、環境に応じて利用するインターフェースを切替える必要が生じる。また、計算機がアプリケーションの通信ごとや通信相手ごとに用いるインターフェースを切替えたり、同時に複数インターフェースを利用することも想定される。インターフェースを同時に利用することにより、単一のインターフェースでは不足する帯域を補うことも出来る。

そこで本研究では、Mobile IPv6 を拡張し、複数インターフェースを利用可能にすることにより、常時接続性の問題、インターフェースの同時利用、インターフェースの動的切替えを実現する。移動体計算機は、通信可能範囲が狭い通信インターフェースを複数利用することにより、広域な通信可能範囲を得ることができる。また、複数インターフェースを有効利用することにより、移動体計算機での帯域不足の問題を解決できる。

4.2 将来の移動体通信環境

想定する移動体通信環境について述べる。

人が使う計算機のみならず、自動車や携帯電話等の様々な移動を行う機器がインターネットに接続される。高速に移動する機器や超小型機器等は、人が操作することが困難である。そのため、アドレス取得や通信環境に応じたインターフェースの切替え等、全ての通信設定は自動で行われる。これらインターネットに接続された計算機や機器は、移動中に通信の継続は保証される。また、インターネットにおける識別子としての IP アドレスが不変のため、常に移動体計算機はデータの着信と通信相手は移動体計算機に対しての発呼が行える。

また、通信インフラストラクチャの整備により、街中でも無線を用いた広帯域な通信が行えるようになる¹。無線通信を始めとした様々な通信網に対応するため、移動体計算機は複数のインターフェースを持つ。それらの通信インターフェースを効率良く使い分け、最適な通信を行う。

図 4.1 では、移動体計算機は携帯電話、有線 (Ethernet)、無線 (IEEE802.11b) 等の通信インターフェースを持つ。計算機は、通信環境 (インターフェースのリンク等) の状態に応じて動的にインターフェースの切替えを行う。利用者があらかじめ計算機に対して必要な通信品質を設定しておけば、計算機が最適な通信インターフェースを決定する。そのため、利用者が希望する通信品質に劣る通信状態であれば、積極的に複数の通信インターフェースを同時に利用し、通信品質を補完する。インターフェースを切替

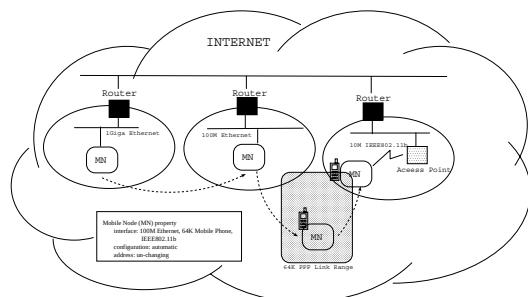


図 4.1. 将来的な移動体通信環境

える度に IP アドレスが変化してしまうが、しかし移動透過性を保証する移動体通信プロトコルを用いることにより、見掛け上の IP アドレスの変化は起こらない。通信に用いる IP アドレスが変化しないため移動体計算機は、移動に影響されず通信が行える。

4.2.1 要求

移動体通信環境に対する要求を以下に示す。

- 移動中の通信の継続
移動による IP アドレスの変化を隠蔽することにより、通信の継続を可能とする
- 一意のアドレスでのデータ着信 (通信相手からの移動体計算機の一意の識別)
移動による IP アドレスの変化を隠蔽することにより、移動体計算機は常に一意のアドレスでパケットを受信することが出来る。また、通信相手は常に移動体計算機に対して通信を開始することが出来る。
- 常時接続性
移動体計算機は、移動に伴う位置の変化に影響無く、常にネットワークへ接続され通信が行える。
- 移動に影響されない通信品質
移動体計算機は、移動場所に関わらず常に一定の品質で通信を行える。
- 自動設定
移動体計算機は、移動を検知し必要に応じて動的にネットワーク接続のための設定を更新する。利用者は、手動で設定する必要は無い。

4.2.2 これまでに解決された問題

現在の移動体通信環境は、様々な問題がありインターネット利用における制約も多い。これまでの研究において、解決中である移動体通信環境の問題点について説明する。

移動透過性

現在の通信環境では、移動体計算機がネットワークを移動する度に、割り当てられる IP アドレスが

¹ アメリカでは既に Ricochet Wireless Network と呼ばれる無線通信システムがある。このシステムは、無線型高速データ通信網である。具体的には、道端にある電灯あるいは電柱に、無線のトランシーバを設置する。クライアントは、無線のモデムを計算機に設置することによりインターネット接続が可能になる。電灯等にトランシーバを設置することで、広い地域でのサービスを実現している。

変わる。そのため、通信を行う際に IP アドレスを参照している TCP や UDP 等のトランスポート層を用いた通信は遮断されるという問題がある。移動体通信では、移動に影響されないシームレスな通信の実現が不可欠である。本問題は、Mobile IP を用いることにより解決できる。IP アドレスが不変なため、移動後も同じ IP アドレスを用いることが出来る。そのため、移動による影響を受けずに通信の継続が可能となる。

着信可能性

第 4.2.2 節で説明したように、移動体計算機が移動する度に通信の識別子としての IP アドレスが変わる。IP アドレスが変わることにより、移動体計算機の通信相手は移動体をインターネット上で一意に識別することができない。そのため、移動体計算機は常に一意なアドレスで着信を行うことが出来ず、また通信相手は移動体計算機のアドレスを特定できないため、通信を開始できない。着信可能性の実現に関しては、Dynamic DNS を用い移動体計算機が、動的に現在利用している IP アドレスを DNS に登録し更新することにより、本問題を解決することができる。しかし、Dynamic DNS は未だ研究、実験段階である。また、多数の移動体計算機がインターネット上に登場した時にそれら进行处理することが可能かどうかというスケーラビリティの問題が解決されていない。本問題も、Mobile IP を用いることにより解決できる。通信相手は、常に Mobile IP の一意な IP アドレスに対して通信を開始することが出来るからである。Mobile IP では、移動を支援するためのサーバは存在するが、Binding を通信間で保持することにより、第 3 者を介さない通信間で移動透過性を実現する。そのため、サーバへの負荷が軽減されるなど、Dynamic DNS に比べてスケーラビリティが考慮されたプロトコルである。

環境適応性の欠如

移動体通信環境は、ネットワークごとに計算機が利用可能な帯域、遅延、課金システム、セキュリティレベルが異なる。そのため、移動体計算機は移動後、移動先ネットワークの通信環境に最適な通信を実現する必要がある。そうでない場合、狭帯域なネットワーク上に大量のデータを送信し高遅延をまねいたり、ファイアウォール等のセキュリティによって、外

部ネットワークに対して接続を行えなかったりする場合がある。

移動体計算機は移動先での通信環境を迅速に把握し、最適な通信を実現する必要がある。環境適応性については、MIBsocket システムがある。計算機内の通信に関するネットワークエンティティの情報変化の通知、取得、設定が行えるシステムである。アプリケーションを、MIBsocket からのネットワークエンティティの情報変化を用いて処理することにより、移動に応じたアプリケーションの構築が可能である。MIBsocket 以外にも環境適応性については、Stanford 大学の Mosquito プロジェクトや Carnegie Mellon University の Odyssey プロジェクト等、非常に多くのグループで研究が行われている。

しかし、未だ標準的なシステムが存在するには至っていない。

End-to-End 通信の欠如

インターネットの特徴且つ最大の利点である End-to-End 通信が、既存の移動体通信では実現されてない。例えば、携帯電話によるインターネットアクセスは、既存の公衆回線網を使いインターネットにアクセスするため、公衆回線網とインターネットの間に proxy サーバが設置されている場合が多い。End-to-End 通信の欠如により、新たなサービスが生まれにくい。

しかし、移動透過性と着信可能性の問題を解決することにより、移動体計算機はインターネットの通信相手からは固定計算機と同じであり、End-to-End 通信を行うことが可能となった。

4.2.3 未解決の問題

本研究により改善される、現状の移動体通信環境の問題点と解決方法について説明する。

常時接続性の欠如

現在、地球全域を網羅するような広域、広帯域、低遅延な通信接続システムは存在しない。また、現在利用できる通信インターフェースには、広域、狭帯域、高遅延であったり、狭域、広帯域であったりとインターフェースごとに通信品質が異なる。現在、移動型計算機を常にインターネットに接続しようとする場合、主に利用されているのが電話システムである。携帯電話網は現在、非常に広域を支援している

半面、狭帯域、高遅延であり、課金の問題がある。

そこで、Ethernet に代表される有線インターフェースが可能な時は有線接続を優先する。また、移動しながらの通信を行う場合は、無線インターフェースを優先する。IEEE802.11b に代表される無線インターフェースを用いれば、広帯域でかつ広域な通信環境²を実現できる。しかし、現在、インターフェースの状態管理や選択、設定は利用者が行っている。利用者は、移動することによる無線インターフェースの到達性を失うと、携帯電話を用いて接続し直す必要がある。しかし、高速で移動する通信機器等を想定する場合、手動での設定には限度がある。

帯域の不足

移動体計算機で、ビデオや音楽等のストリーム通信を行うことを考えた時、帯域不足の問題がある。また、ストリーム等のデータ量の多い通信により、他の通信が帯域上の制約や遅延の影響を受け、通信品質が劣化する。移動体計算機で利用できる通信インターフェースで、広帯域で低遅延なものは限られている。また、広帯域なインターフェースで通信可能範囲が広域なものも限られている。そこで、複数のインターフェースを同時に利用し、通信帯域を補うことが必要である。

例えば、通信帯域が不足している状況下で、無線インターフェース網と携帯電話網の双方が利用可能であれば、計算機が両方のインターフェースを用いてネットワークに接続する。この時、ストリーム通信のようなデータ量が多い通信は無線インターフェースを用い、それ以外の通信は携帯電話を用いることにより、通信品質を一定に保つことが可能となる。

4.3 複数インターフェース支援システムモデル

既存の計算機システムでは、複数インターフェースの同時利用や切替えを行うことが想定されていない。本章では、始めに「複数インターフェース支援」について定義し、複数インターフェース支援を可能とする計算機システムモデルについて述べる。本章で用いる、インターフェースとは Ethernet や IEEE802.11b 等の無線インターフェース等の通信インターフェースを指す。

4.3.1 複数インターフェース支援

複数インターフェース支援とは、**複数インターフェースの同時利用とインターフェースの状態に応じたインターフェースの切替え**を用いて実現することである。複数インターフェースの同時利用では、通信可能範囲の広域性確保や帯域不足の解消のために不可欠である。また、インターフェース切替えは、通信可能範囲の広域性確保や、常に有効なインターフェースを利用する通信環境適応性を保証する。移動などに伴いインターフェースが無効になった場合、有効なインターフェースに切替えることで、通信は継続することが出来る。

4.3.2 システムモデル

移動体計算機に限らず、複数インターフェースを利用する場合、計算機は各々のインターフェースの状態を把握し、それらの情報を基に最適なインターフェースを選択することが必要である。また、インターフェースの情報以外にも、利用者が手動で選択ポリシーを設定できる機構が必要である。インターフェースが持つ情報には、通信を行う上で重要な情報が含まれており、それらの情報が変化することによって通信が遮断されてしまう。したがって、インターフェースの状態変化を隠蔽し、通信を継続させることが必要である。

図 4.2 は、移動体計算機における、OSI の 7 階層モデルを用いて示したインターフェース支援システム

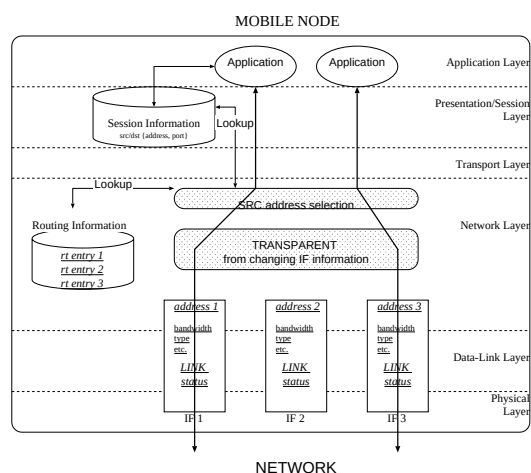


図 4.2. 複数インターフェース支援システムモデル

² ローミングが可能な場合

モデルである。下線が引いてある情報は、インターフェースの情報である。また、イタリック文字で示される状態は、動的に変化することがある。

複数インターフェースを支援するために、既存のインターフェース選択機構を強化し、新たに移動透過性機構を追加する。

現在の計算機システムでは、インターフェースは図4.2の通り3層にまたがって情報を保持している。ネットワーク層では、経路情報やアドレス等の情報を持ち、インターフェース層ではリンクの状態やインターフェースの帯域情報等を持っている。また物理層では、実際にデータをネットワークに送信する時に用いるハードウェアに関する情報等を持っている。3層以外では、セッション層やアプリケーション層等では、これら3層で保持されているインターフェースの情報を参照し利用する。

計算機は、通信の開始時に、送信元と送信先のアドレスとポートを決定し、セッション層で保持する。これらの情報は通信において必須であり、各層は、この情報を元にデータの送受信処理を行う。データは、セッション層で開始時に決定された送信元アドレスに従い、ネットワークに送信される。つまり、**送信元アドレスの決定は、その通信において利用するインターフェースを選択することである**。このインターフェースを決定する部分が、インターフェース選択機構 (SRC address selection) である。既存のシステムでは、送信先アドレスを用いて経路情報を検索し、最適な送信元アドレス (インターフェース) を決定するようになっている。しかし、インターフェース選択の指針としては、経路情報のみでは不十分である。インターフェースの有効利用のためには、インターフェースの情報や利用者が手動で設定する選択ポリシーを用いることが重要である。

また、インターフェースが持つ情報は、静的な情報だけではない。特に、移動体計算機では移動に伴いネットワーク間を移動するため、ネットワーク層で保持されるインターフェース情報は動的に変化する。移動に伴い、リンクの情報も同じく動的に変化する。インターフェースの状態が変化すると、計算機は利用しているインターフェースが通信可能なかを逐次評価する必要がある。もし、インターフェースが通信不能であれば、それ以外のインターフェースに切替える必要がある。通信に用いるインターフェースを変更した場合、セッション層での送信元アドレ

スの情報が新しいインターフェースのアドレスに変更されなければならない。しかし、既存の計算機システムではこのような状況を想定していないため、通信中に送信元アドレスを切替える機構は存在しない。インターフェースを変更されると、通信は切断されてしまう。そこで、通信を継続するためにあるのが、移動透過性機構 (TRANSPARENT) である。移動透過性機構は、移動に伴うインターフェースの切替えを、アプリケーション層等のネットワーク層より上位の層から隠蔽し、インターフェース切替えによる通信の切断を防ぐ。

既存の計算機モデルは、主に固定体計算機を想定して構築されている。そのため、移動体計算機における複数のインターフェースを利用するには、既存の計算機モデルでは不十分であることがわかる。

インターフェース選択機構

インターフェース決定に用いる情報としては、送信先アドレスを用いた経路情報、インターフェースのアドレスの有無、リンクの状態から判定する通信可能か否か、計算機の利用者や管理者が設定する選択ポリシー等がある。

経路情報を用いて、データを送信先に届ける上でネットワーク的に最適な経路を決定し、その経路を持つインターフェースを利用する。

例えば、計算機がインターフェース A と B を持ち、インターフェース A はネットワーク A に接続されており、インターフェース B はネットワーク B に接続されている場合を考える。この計算機が、ネットワーク A に接続されている計算機と通信を行う場合、経路情報からインターフェース A が選択される。インターフェース A を用いることにより1ホップで送信先計算機に届けることが出来る。インターフェース B を用いた場合には、インターフェース B はネットワーク A への経路を持っていないため、ネットワーク B のルータに渡され、インターネットを介してネットワーク A の送信先計算機の届けられる。後者は、非効率的であり通常は選択されない。

インターフェースの状態とは、そのインターフェースが実際にデータを送信が出来るかを示す情報である。インターフェースを決定する上で、通信可能なインターフェースを選択するのは当然である。インターフェースは、複数の層で情報を保持するため、それら各層の情報を評価した上で、インターフェース

の状態を決定する必要がある。インターフェースにアドレスが設定されていても、リンクが無効な状態であれば、そのインターフェースは通信不能である。つまり、通信に関する全ての状態が整って初めて通信可能と判断できる。しかし、既存の実装モデルでは、これらの情報を統合的に管理する機構は不十分である。各層でのインターフェースの情報は各層で管理されているため、層を跨って情報を管理する機構が整備されていない。

複数通信インターフェースの同時利用においては、これまで説明した通信に関する情報からインターフェースを決定するのではなく、計算機の利用者や管理者が、任意に選択ポリシーを設定できることが重要である。インターフェースの選択において、計算機の特徴、利用目的、利用環境によって最適なインターフェースは異なる。そのため、計算機から得られる情報だけでは、インターフェースを選択する上での指針としては不十分である。利用者によるポリシー設定を可能にすることにより、様々な計算機を想定した複数インターフェース支援機構を構築することが可能となる。設定可能なインターフェース選択ポリシーとしては、以下の役目が考えられる。

- 通信相手ごと
通信相手のアドレスに応じて通信インターフェースを切替える方法。
- 通信プロトコルごと
通信相手との通信のプロトコルに応じて通信インターフェースを切替える方法。
- コネクションのフローごと
通信相手との通信フローに応じて通信インターフェースを切替える方法。

インターフェース選択機構では、通信開始時に計算機から得られる情報と利用者が設定した情報からその通信に最適なインターフェースを決定することが必要である。また、現在も新たな種類の通信インターフェースが登場しており、今後登場するインターフェースにも対応できる、汎用的かつ拡張可能な機構でなくてはならない。

移動透過性機構

移動透過性機構は、移動に伴う通信インターフェースの状態変化をアプリケーションから隠蔽する機構

である。移動に伴い、計算機内で利用可能なインターフェースの状態や経路情報は変化する。現在の計算機システムでは、アドレスはインターフェースごとに設定されている。アドレスはインターフェースが接続されているネットワークごとに割り当てられるため、ネットワーク間を移動することにより、新たなアドレスを取得しなくてはならない。また、計算機が常にインターフェースの利用可能範囲を移動するとは限らない。そのため、移動によりインターフェースのリンクが無効になり、利用不能になることもある。例えば、無線インターフェースの場合はアクセスポイントあるいは通信相手との間での無線のシグナル強度が頻繁に変わるため、リンクが通信可能な状態と不可能な状態との間を頻繁に変化する。

このように、移動に伴いインターフェースの状態は動的に変化する。インターフェースの状態変化が起きると、計算機はインターフェース選択をやり直し、最適なインターフェースを再度決定する必要がある。移動前に選択したインターフェースが、移動後の通信環境で最適なインターフェースとは限らないからである。インターフェースを再選択し、移動前とは異なるインターフェースが選択された場合、その通信で利用する送信元アドレスは変わる。現在の計算機システムは、通信中に動的に送信元アドレスを変更することは出来ない。そのため、ネットワーク層より上位層に対してこのアドレスの変化を隠蔽しなくてはならない。

現在まで、移動透過性の研究は様々行われて来ており、代表的な物に Mobile IP、VIP、LIN6 がある。これら、技術を用い移動透過性を保障することが必要である。

4.4 Mobile IPv6 を用いた複数インターフェース通信機構の設計

Mobile IPv6 を用いた複数インターフェース支援システムモデルの設計について述べる。Mobile IPv6 を用いた複数インターフェース支援機構の通信方法について述べる。また、利用可能な Mobile IPv6 プロトコルスタックの実装は、その設計上、複数インターフェースを扱うことが出来ないため本研究では新たに Mobile IPv6 も設計する。

4.4.1 Mobile IPv6 の利用

本システムは、自動車等の台数の多い機器に用い

ることが想定されるため、IPv6 を用いる。また、移動透過性機構は Mobile IPv6 を用いる。

Mobile IPv6 は、現在 IETF で標準化の議論が進んでおり、仕様も確定段階に入っている。Mobile IPv6 が世界標準となった時、本システムの機器への普及を容易に行う事が出来る。自動車や携帯電話等の既に普及されている機器に対して、本システムを新たに導入するには、標準や普及に関して考慮する必要がある。

また、Mobile IPv6 の仕様書には複数の CoA を扱うことにも触れられており、Mobile IPv6 の複数インターフェース支援のための拡張も容易である。仕様書で定義されている複数 CoA に関連するものとしては、複数の CoA を通信相手に通知するためのパケットフォーマットのみが定義されており、実際の利用方法などについては触れられていない。そのため、必要に応じ、定義されたパケットフォーマットを用いる。

それ以外にも、Mobile IPv6 は、ホームアドレスと実際に通信するアドレスの関係である Binding を、通信ノード間で保持する。そのため、複数インターフェースを用いる際に、ポリシーを反映することが容易に行える。ポリシー毎に Binding を保持すれことにより実現できる。

この様に、Mobile IPv6 は、複数インターフェースを支援する上で適したプロトコルであり、標準化が行われる事による普及面からも有利である。

4.5 複数インターフェース支援機構のアプローチ

複数インターフェース支援機構モデルを実現するためのアプローチとしては、移動透過性プロトコルとインターフェース選択機構を追加する必要がある。第3章で述べたように、移動透過性プロトコルには幾つかあるが、本研究では Mobile IPv6 を用いる。Mobile IPv6 を用いた理由は、Mobile IPv6 の Binding を用いることにより、複数インターフェース支援と移動透過性を同時に解決することが出来るからである。また、Mobile IPv6 の Binding は、移動体計算機と通信相手間で共有される情報である。Binding には、通信において移動体計算機が利用しているインターフェースの情報が保持されている。そのため、通信相手は、移動体計算機の受信すべきインターフェースを明示的に指定することが出来る。明示的に指定しない場合、通信相手が送信したデータが、移動体

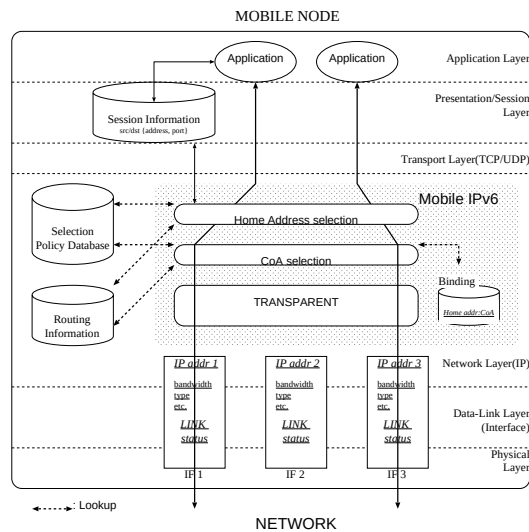


図 4.3. Mobile IPv6 を用いた複数インターフェース支援機構

計算機の違うインターフェースに届く可能性がある。この場合、送信と受信のインターフェースが異なるため、利用者のポリシーが半分しか反映することが出来ない。

図 4.3 は、前章で述べたモデルを、Mobile IPv6 に適用したモデルである。Mobile IPv6 では、通信の識別子のアドレス (ホームアドレス) と実際に通信に使うアドレス (CoA) を持つ。そのため、インターフェース選択機構はこの 2 種類のアドレスについて扱う必要がある。

アプリケーション層等の IP 層のより上位の層に対するインターフェース選択は、ホームアドレス選択機構が行う。ホームアドレスを用いることにより、移動に伴うアドレス変化を隠蔽することが出来る。ホームアドレスの選択は、移動体計算機が複数のホームアドレスを用いて通信を行う時に有効となる。例えば、移動体計算機はホームリンクとして会社内のファイアウォールネットワークと、個人で加入するプロバイダネットワークがあるとすると、この時、会社内のサーバにアクセスする通信は会社のネットワークで割り当てられているホームアドレスを用いる必要がある。このホームアドレスを用いることにより、ファイアウォールを越えて通信を行うことが出来る。それ以外の通信にはプロバイダから割り当てられているホームアドレスを使うことも出来る。ホームアドレス選択機構では、利用するホームアドレスを、主に利用者が設定するポリシーに従い決定する。ポリシーが存在しない場合は、経路情報等から最適なホーム

アドレスを決定する。

IP 層より下位の層では、CoA 選択機構が物理的に送信するインターフェースを決定する。CoA は、実際にインターフェースで有効なアドレスであるため、CoA の決定がインターフェースの決定となる。本 CoA 選択機構が、インターフェース支援システムモデルにおけるインターフェース選択機構にあたる。CoA 選択機構では、経路情報、インターフェースの状態、そして利用者が設定するポリシから最適な CoA を決定する。

また、CoA を決定した後は、Mobile IPv6 の処理が行われ、通信相手側に選択した CoA を含む Binding 情報を、Binding Update を用いて通知する。Binding を通知した後は、通信相手は通信時に Binding 情報で指定された CoA を用いて通信を行う。移動体計算機側には、Binding Update の情報を管理するため、通信相手に対して決定した CoA の情報を保持する。つまり、一度 CoA を決定した後、Binding 情報が有効である限り、データを送信する度に CoA 選択を行う必要は無い。Binding が無効になる移動時や、利用者によるポリシの変更が起きた時に、CoA 選択が再度行われる。

インターフェース選択機構は、Mobile IPv6 の処理に影響を与えないため、通常の Mobile IPv6 の処理のみで十分である。このように、Mobile IPv6 にホームアドレス選択機構と CoA 選択機構を追加することにより、複数インターフェース支援機構を実現できる。

4.6 複数インターフェース支援システムの設計概要

本節では、Mobile IPv6 の Binding 機構を用いた、複数インターフェース支援システムの設計について説明する。始めに、Mobile IPv6 を用いた複数インターフェースを支援するための仕組みについて述べ、複数インターフェースを利用するためのインターフェース選択ポリシについて説明する。

4.6.1 複数インターフェース支援の仕組み

複数インターフェースを有効利用するために、Mobile IPv6 の Binding を用いる。その仕組みを図 4.4、図 4.5 に示す。図 4.4 は通信開始時の処理を示しており、図 4.5 は処理後の通信形態について示している。

図 4.4 では移動体計算機が複数インターフェースを持ち、それぞれのインターフェースで CoA を取得

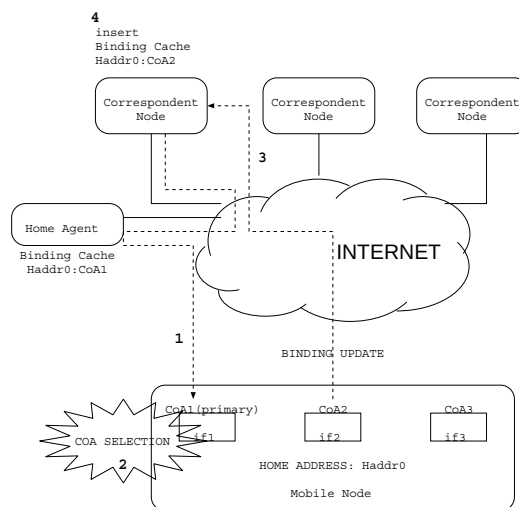


図 4.4. 複数の CoA を用いた Binding 登録

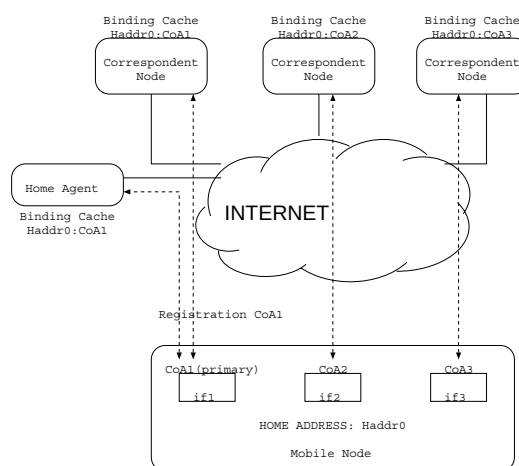


図 4.5. 単一のホームアドレスを用いた複数の CoA の利用

している。そのうちの 1 つの CoA を Mobile IPv6 の仕様通りに primary CoA として Home Agent に登録を行う。そのため、通信相手が移動体計算機に通信を開始すると、(1) Home Agent 経由でデータが届く。移動体計算機は Home Agent 経由のデータを受け取ると、そのデータの通信相手に Binding を登録する必要がある。この時点で、(2) 移動体計算機は複数の CoA から一定のインターフェース選択機構により 1 つの CoA (図では CoA2) を選択する。(3) 選択された CoA2 を、(4) 通信相手に Binding Update を用いて Binding として登録する。他の通信相手に対しては、インターフェースの選択機構に従い、異なる CoA を登録されることができる。

図 4.5 では、通信相手が移動体計算機に関する Bind-

ing の情報を保持している状態である。この時、通信相手は、Binding 内に保持されている移動体計算機の CoA を用いて通信を行うため、その CoA 以外のインターフェースヘータを送信することは無い。通信相手 B は、送信開始時に Binding の情報を検索し、移動体計算機に関する Binding を取得する。そのため、通信相手 B は、Binding 内で指定されている CoA2 に向けて移動体計算機にデータを送信する。通信相手 C は、同じく Binding を検索した後、CoA3 に向けてデータを送信する。移動体計算機からの通信では、移動体計算機が管理する Binding Update 情報を検索し、通信相手に対して登録した CoA を用いてデータを送信する。このように、通信相手に応じて利用するインターフェースを変えることが可能である。インターフェースを切り替える単位としては、通信相手ごと以外にも考えられる。これについては、次節で述べる。

移動時や利用者によるポリシー変更の場合、インターフェース選択機構によりインターフェースを再選択する。その後、移動体計算機が Binding Update を用いて Binding を更新することにより、新たな CoA に切り替えることが出来る。

また、移動透過性は、Mobile IPv6 を用いて解決される。Mobile IPv6 では、IP 層より上位層からは Mobile IPv6 のホームアドレスを用いて通信するように見せかける。そのため、移動などに伴うインターフェースの切り替えが起きた場合でも、移動体計算機が行っている通信はホームアドレスを用いているため、影響を受けることは無い。そのため、移動中の通信の継続が実現できる。

4.6.2 通信インターフェース選択のポリシー

複数通信インターフェースを利用するためのインターフェース選択機構で用いるポリシーとしては以下が考えられる。

- 通信相手ごと
通信相手の IPv6 アドレスに応じて通信インターフェースを切替える方法である。本方法では、通信相手に対して通知する CoA をポリシーに応じて切替えで実現できる。そのため、通信相手は移動体計算機に対して通信を行う場合、Binding をホームアドレスを用いて通常通り検索を行う。
- 通信プロトコルごと (HTTP、TELNET、SSH、

FTP 等)

通信相手との通信のプロトコルに応じて通信インターフェースを切替える方法である。通信相手ごとに切替えるためには、通信相手は利用するプロトコルごとに移動体計算機の Binding を管理する必要がある。そのため、通信時には移動体計算機のホームアドレスと通信プロトコルを用いて Binding を検索しなくてはならない。また、1 台の移動体計算機に対して複数の Binding が存在することになるため、管理が複雑になる。また、通信プロトコルごとにどのインターフェースを用いるかを、利用者あるいは管理者が設定する必要がある。

- 通信のフローごと

通信相手との通信フローに応じて通信インターフェースを切替える方法である。本方法は、通信プロトコルごとの切替えより更に細かなインターフェースの切替えを行うことが出来る。通信フローの開始時にアプリケーション或は利用者がどの通信インターフェースを利用するかを明示的に指定する必要がある。

通信相手が移動体計算機に対する Binding を検索する場合は、ホームアドレスとフローの識別子 (フロー番号) を用いる。本方法の場合、通信フローごとに Binding を登録することとなり、Binding の管理が複雑になりかつ、Binding テーブルのエントリが増長する。そのため、Binding の検索コストが高くなる。また、現在の IPv6 では、通信フローに対してフロー識別子が、必ず割り当てられているとは限らない。これは、IPv6 の仕様に曖昧な部分があるのと、実装がなされていない部分があるためである。

これら通信インターフェース選択ポリシーを比較したのが表 4.1 である。インターフェースの選択ポリシーを細かく設定することにより、Binding に関連する操作のコストも増大する。そのため、一概にどのポリシーが優位であるかを決定することは出来ない。そのため、移動体計算機の利用形態によりポリシーを選ぶことが重要である。本研究では、以下に挙げるポリシーについて、通信相手ごとの切替えを選んで実装し評価した。通信インターフェースの選択ポリシーは独立して設計されているため、ポリシーの追加や入れ換え (切替え) は比較的容易に行える。

表 4.1. 通信インターフェース選択ポリシーの比較

ポリシー	Binding の 管理コスト	Binding の 検索コスト	Binding の 登録コスト	利用頻度	IF の 有効利用度
相手ごと	小	小	小	○	○
プロトコルごと	中	中	中	○	◎
フローごと	大	大	大	○	◎

†: Binding に関する

4.7 Mobile IPv6 を用いた複数インターフェース 通信機構システムの実装

Mobile IPv6 における複数インターフェース支援機構及び Mobile IPv6 の実装について述べる。

4.7.1 実装環境

複数インターフェース支援機構及び Mobile IPv6 を FreeBSD 4.1.1-RELEASE 上に実装した。FreeBSD 4.1.1-RELEASE 上には、KAME プロジェクトの IPv6 コードが含まれており、そのコードを拡張して実装した。

4.7.2 複数インターフェース支援機構

本節では、複数インターフェース支援機構の実装について述べる。ポリシーデータベース部、ポリシーデータベースにアクセスするためのポリシーインターフェース部について述べ、最後にポリシーを用いてインターフェースを選択するインターフェース選択部について説明する。現在、利用者によるポリシーの設定はコマンドを用いて行う。

ポリシーインターフェース部

ポリシーインターフェース部は、ポリシーデータベース部に実際にアクセスするためのものである。設計で述べた通り、取得、登録、削除のポリシーインターフェースを用意してある。選択ポリシーインターフェース以外のインターフェースは、利用者から設定されるため、ioctl を用いて実装されている。また、選択ポリシーインターフェースは、選択時に複数ポリシーが一致した場合、最適なポリシーを一致したポリシーから選ぶ。

インターフェース選択部

インターフェース選択部は、通信開始時にインターフェースを決定する時に用いる。インターフェースの決定には、ポリシー情報、経路情報、インターフェー

スの状態情報を用いる。インターフェース選択部では、これらの情報を取得しそれらの中から最適なインターフェースを決定する。

4.7.3 Mobile IPv6 の実装

Mobile IPv6 に実装について説明する。第 4.4 章で述べたように、Mobile IPv6 は複数インターフェースを扱えるように実装されている。全体の処理について解説した後、仮想インターフェース、データ構造、Binding 処理、送受信処理について詳しく解説する。また、Mobile IPv6 を制御するためのコントロール部、移動検知、移動通知の処理についても述べる。

実装概要

図 4.6 は、Mobile IPv6 プロトコル処理の簡単な全体図である。本節の後半で各々を細かく詳述する。シャがかかっている部分は MN のみが持つ部分である。それ以外は基本的に共通である。

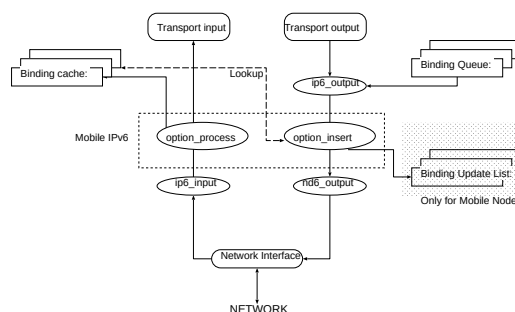


図 4.6. Mobile IPv6 処理概要

Mobile IPv6 は、IPv6 のオプションヘッダを用いて処理の大半を行うため、IPv6 の送受信処理の中でオプションを挿入することで実現する。また、Binding Cache、Binding Queue、Binding Update List (MN のみ) といった Binding 関連のデータを管理する。Binding を検索してオプションに挿入したり、Binding Update List を更新したり、受け取っ

たパケットから Mobile IPv6 のオプションヘッダを処理し Binding Cache を更新等を行う。

第5章 地理的位置情報システム

5.1 はじめに

我々は地理位置情報システム (GLI: Geographical Location Information System) を提案した。本システムは現実世界を移動する移動体を対象とし、その識別子と位置情報および付帯情報の登録・検索機能の実現している。同時に地球規模で動作するスケーラビリティを実現している。

GLI システムでは、移動体はサーバに位置や状態の情報を登録し、クライアントは、識別子や位置を鍵とした検索要求をサーバに送信することにより、移動体を検索することができる。

しかし、現在の GLI システムでは、誰でも自由に登録・検索ができるため、以下のようなプライバシーに関する問題が発生する。

- 移動体の特定・追跡
- なりすましによる偽の登録
- 登録情報のインターネット上での盗聴・改竄

本章では、これまでの地理位置情報システムの利点である検索手法を損なうことなく、プライバシー保護を実現した地理位置情報システムを提案する。本章で目標とするプライバシー保護とは、移動体と信頼関係にある検索者には移動体の識別子や位置情報および付帯情報を公開するが、第三者には個々の移動体の識別を不可能にした統計情報のみを公開することである。その実現のため、HID (Hashed ID) を導入する。同時になりすましや盗聴、改竄も防止する。

便宜上、これまでの GLI システムを旧 GLI システムと呼び、本章で提案するものを GLI システムと呼ぶ。

5.2 地理位置情報システムの概要

図 5.1 に旧 GLI システムの構成を示す。旧 GLI

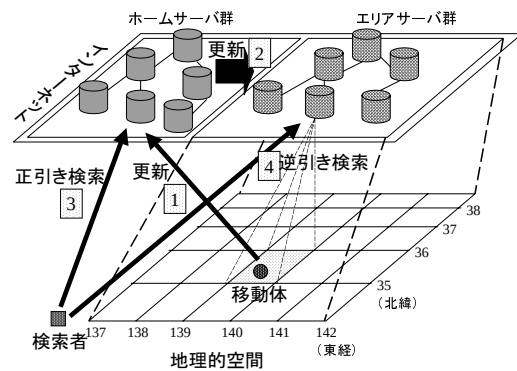


図 5.1. 旧 GLI システムの構成

システムは位置情報および付帯情報を登録する移動体、これらの情報を管理するホームサーバ群とエリアサーバ群、および検索者からなる。移動体の識別子としては FQDN (Fully Qualified Domain Name)¹ を用いる。地理位置情報としては緯度・経度・高度を用いる。付帯情報とは、移動体の移動方向や移動速度などである。旧 GLI システムは 2 種類の検索機能を提供する。1 つは移動体の識別子を鍵とし、その移動体の位置情報および付帯情報を検索するものである (正引き検索)。もう 1 つは地理的な領域を鍵とし、その領域に存在する移動体の識別子、位置情報および付帯情報の集合を検索するものである (逆引き検索)。図 5.1 に示すように旧 GLI システムでは、ホームサーバ群は正引き検索を担当し、エリアサーバ群は逆引き検索を担当する。それぞれのサーバ群は階層構造をとっており、分散処理によって大規模性を実現している。

図 5.1 では mobile.sfc.wide.ad.jp という識別子を持つ移動体が北緯 35 度 18 分 18 秒、東経 139 度 30 分 40 秒に存在している。移動体は識別子から決定されるホームサーバに識別子、地理位置情報および付帯情報を登録する (図 5.1-(1))。登録を受けたホームサーバは、移動体の地理位置情報から決定されるエリアサーバに、移動体の識別子、地理位置情報および付帯情報を登録する (図 5.1-(2))。正引き検索を行う検索者は、移動体の識別子から決定されるホームサーバに mobile.sfc.wide.ad.jp という識別子を鍵として検索要求を送信する (図 5.1-(3))。検索要求を受信したホームサーバは、検索結果として北緯 35 度 18 分 18 秒、東経 139 度 30 分 40 秒という地理位置

¹ ドットで区切られたホスト名。例えば、mobile.sfc.wide.ad.jp

情報および付帯情報を返す。逆引き検索を行う場合は、例えば、北緯 35 度～36 度、東経 139 度～140 度という領域を鍵とし、この領域から決定されるエリアサーバに検索要求を送信する(図 5.1-(4))。検索要求を受信したエリアサーバは、mobile.sfc.wide.ad.jp という識別子、北緯 35 度 18 分 18 秒、東経 139 度 30 分 40 秒という地理位置情報および付帯情報を返す。指定された領域に他の移動体も登録されている場合は、その情報も返す。

5.3 プライバシ保護の目標

図 5.2 に GLI システムがおかれる環境とその脅威を示す。GLI サーバ群と移動体および検索者はインターネットを介して通信する。GLI システムでのプライバシー保護を考える上で、GLI サーバ群は不正を行わないという前提条件を置く。また、移動体にとって、信頼関係にある検索者と信頼関係のない検索者(第三者)という区別があるものとする。このような前提条件のもと、GLI システムが目標とするプライバシー保護の目標を述べる。

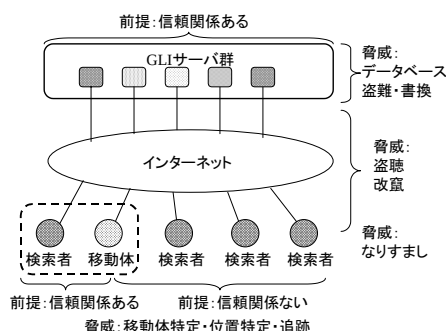


図 5.2. GLI システムにおける環境と脅威

5.3.1 移動体の特定・追跡の可否

旧 GLI システムにおいては、誰でも自由に登録・検索を行うことができる。従って、ある移動体の識別子を鍵とする正引き検索によってその位置情報および付帯情報を取得できる。また、ある領域を鍵とする逆引き検索によって得られた結果から、その領域に存在する移動体の特定・追跡が可能である。移動体のプライバシーを保護するためには、信頼関係の有無によるアクセスの制御が必要である。正引き検索では、信頼関係にある検索者だけがある移動体を特定し、その識別子を鍵とした検索を行えるようにする。信頼関係のない検索者は移動体の正引き検索

を行えないようにする。逆引き検索では、鍵として指定した領域に存在する移動体と信頼関係にある検索者は、検索結果からその移動体の識別子と位置情報および付帯情報を得られるようにする。その移動体と信頼関係のない検索者は移動体を特定できず、位置情報・付帯情報といった統計情報のみを得られるようにする。また信頼関係のない検索者によって追跡が行えないようにする。

5.3.2 通信データの盗聴・改竄防止

移動体はネットワークを介して GLI サーバに自らの位置情報を登録する。ネットワークを介した通信を行うことで、通信データを盗聴されたり改竄される可能性がある。やりとりされるパケットには移動体の送信元のアドレスが含まれるため、データ部分に含まれる登録情報との対応関係が明らかになる。データ部分を暗号化するなどの方法により、盗聴されても移動体と位置情報との対応関係を明らかにさせないこと、また、通信データが改竄された場合は GLI サーバがその事実を検出できるようにすることを目標とする。

5.3.3 なりすまし防止

悪意のある移動体が別の移動体になりすまして GLI サーバに偽の位置情報および付帯情報を登録することも考えられる。GLI システムでは GLI サーバが位置情報を登録する移動体を認証し、正しい情報のみを扱うようにする。

5.3.4 データベース盗難への対処

GLI システムのサーバ群は、それぞれ移動体の情報を登録するデータベースを持つ。そのうち 1 つのデータベースが盗難されても、移動体の特定ができないように工夫する。

5.4 解決手法

本節では、前節で掲げたプライバシー保護を実現するための手法について述べる。

5.4.1 Hashed ID の導入

第 5.3.1 節における移動体の特定・追跡の可否を実現するために、本章では移動体と信頼関係にある検索者間でのみ理解できる情報(以下、符丁と呼ぶ)を移動体の識別子として用いる方式を提案する。移

動体は符丁と位置情報・付帯情報を GLI サーバに登録する。正引き検索では、信頼関係にある検索者だけが符丁を鍵として GLI サーバに検索要求を行うことができる。本章では移動体と移動体と信頼関係にある検索者で秘密鍵を共有し、移動体の真の識別子を鍵付きハッシュ関数に作用させることによって、符丁を生成する方法を提案する。以降符丁を *HID* (Hashed ID) と呼ぶ。

逆引き検索では、領域を鍵として GLI サーバに検索要求を行い、検索結果としてその領域に存在する移動体の集合が得られる。移動体の集合における検索者と信頼関係にある移動体の特定は、GLI サーバ側で行う方法と検索者側で行う方法がある。前者の方法では GLI サーバが検索者から *HID* の集合を受け取り検索の結果得られる移動体の *HID* と比較する。そして GLI サーバが検索結果を返す際、*HID* が一致したエントリについては *HID* を付けた位置情報・付帯情報を返し、そうでないものは、位置情報・付帯情報のみを返す。後者の方法では GLI サーバは *HID* をつけた位置情報・付帯情報を返し、検索者が *HID* を比較する。前者の方法では信頼関係にない検索者に移動体の *HID* が公開されない利点があるが、比較する *HID* の数の増加が GLI サーバの負荷を増加させる欠点を持つ。GLI システムは大規模環境で動作することを目標としているため、サーバのスケーラビリティを重視し、本章では後者の方法を採用する。

逆引き検索では、*HID* が第三者である検索者に伝わるため、第三者である検索者は *HID* を監視し続けることにより、*HID* から移動体を特定することはできないが、追跡することはできてしまう。追跡を防止するには *HID* を頻繁に変更する必要がある。*HID* を変更するごとに移動体と信頼関係にある検索者間で通信を行うとコストが大きくなってしまう。そこで本章では移動体と信頼関係にある検索者間で時刻同期を前提とし、鍵付きハッシュ関数の鍵として時刻情報を使用する方法を提案する。この結果、真の識別子を *ID* とすると *HID* は次の式で表される。

$$HID = hash(ID \oplus key(t)) \quad (5.1)$$

$hash()$ はハッシュ関数を表し、 $key(t)$ は時刻の関数、 t は時刻を表す。 $\oplus$ は結合を意味する。ある移動体と信頼関係にある検索者は、その移動体と時刻が同期しているので、同時刻に同じ *HID* を生成す

ることができ、これを検索の鍵として使用する。

移動体の識別子として *HID* を導入することにより、旧 GLI システムにおいて正引き検索の役割を持っていた階層的ホームサーバの代わりに、*HID* を鍵とした正引き検索を行う階層的 *HID* サーバを導入する。旧 GLI システムにおいてはホームサーバは FQDN によって階層化を行っていたが、*HID* の分散管理は *HID* の値によるサーバの階層化によって行う。また、エリアサーバは、登録される移動体の情報のうち識別子が *HID* となる (図 5.4 参照)。

5.4.2 IPsec の利用

インターネット上での通信における盗聴・改竄防止は、IP Security (IPsec) を利用することで実現できる。IPsec は AH (Authentication Header) と ESP (Encapsulating Security Payload) の二つの機能からなる。

AH はパケットにハッシュ署名を使用することによって、パケットの改竄防止と送信者認証を実現する。ESP は、暗号を使用して改竄防止と発信者認証と機密性を確保する。GLI システムでは移動体を特定する情報がやりとりされる場合は、ESP を利用して改竄防止と発信者認証と機密性を実現する。それ以外の情報の場合は、AH を利用して改竄防止と発信者認証を実現する。

なお、IPsec を利用するためにはノード間において秘密鍵の共有などを事前に行う必要がある。このような関係を Security Association(SA) という。GLI システムにおいてはサーバ間には予め SA を確立しておくものとする。

5.4.3 登録サーバの導入

移動体が登録を行う際にはなりすまし防止のため GLI サーバはその移動体を認証する必要がある。前述のように IPsec はノード認証の機能を持つので、GLI システムは IPsec を用いて移動体の認証を実現する。しかし、移動体と *HID* サーバ群の関係や、移動体とエリアサーバ群との関係は、移動体の *HID* と位置情報によって固定されず刻々と変化するので、移動体が登録を行うたびに *HID* サーバおよびエリアサーバと SA を確立しなければならない可能性があり、オーバーヘッドとなる。そのため、登録サーバ群を導入し、登録サーバが移動体を認証するようにする。移動体を 1 つの登録サーバに関連づけることに

より、移動体と登録サーバ間に予め固定的な SA を確立することが可能となる。

5.4.4 データベースの分割

本章で提案する GLI システムは、HID サーバ群・エリアサーバ群・登録サーバ群の 3 つのサーバと移動体により構成される。各サーバで管理される情報を表 5.1 に示す。詳細は第 5.5.3 節で述べる。

このように、データベースを分割することにより、各サーバに特定情報は存在しない。従って、各サーバのうちどれか一つが盗難にあった場合に、盗まれた登録情報から移動体を特定することは不可能であり、実用上問題は少ない。

5.5 GLI システムの設計

本節では、前節で述べた解決手法を導入した GLI システムの設計について述べる。

5.5.1 HID の生成

5.4.1 節で示した HID 生成式 (式 5.1) における鍵 $key(t)$ は時刻情報とともに変化する。登録と検索において同一の HID を移動体と信頼関係にある検索者間で共有するため、 $key(t)$ をある時間間隔で変化させるようにする。本章では $key(t)$ を以下のように定義する。

$$key(t) = ts + intvl * n \quad (5.2)$$

ts は基準時刻、 $intvl$ は HID 変更の時間間隔、 n は 0 以上の整数である。ここで $ts + intvl * n \leq t < ts + intvl * (n + 1)$ とする。真の識別子を ID とすると HID は次の式で表される。

$$HID = hash(ID \oplus (ts + intvl * n)) \quad (5.3)$$

n の値は現在時刻から計算でき、 $ts + intvl * n < \text{現在時刻} < ts + intvl * (n + 1)$ のような関係となる。この式から求められる HID の値が時刻によって変化する過程を図 5.3 に示す。時刻 ts 時に、移動

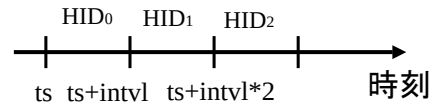


図 5.3. HID 生成の時間変化

体は $HID_0 = hash(ID \oplus ts)$ と位置情報等をサーバに送信する。 $ts + intvl \leq \text{時刻} < ts + 2 * intvl$ では、 $HID_1 = hash(ID \oplus ts + intvl)$ となる。

しかし、 ID と ts および $intvl$ が何らかの理由によって第三者に洩れた場合、第三者は時間とともに変化する HID の生成が可能となってしまう。したがってさらに $intvl$ より大きい一定間隔 (INTVL) で信頼関係にある移動体と検索者間で ID と ts および $intvl$ を交換し鍵を変更することにより、より安全性が保たれる。

5.5.2 HID 生成のハッシュ関数

HID の生成に用いる一方向性ハッシュ関数としては、鍵付きハッシュ関数 (HMAC: Keyed Hashing for Message Authentication Code) が適当である。また HMAC と組み合わせて使用するハッシュ関数としては、処理速度やセキュリティの強度から SHA-1 (Secure Hash Algorithm) が適当である。SHA-1 は異なる入力値から同一の出力値を生成する可能性がほとんどなく、出力値に偏りもないハッシュ関数で、MD4 や MD5 に比べて一意性の点で優れている。したがって、 HID の生成には、HMAC-SHA1 を使用する。

HMAC-SHA1 は、入力値として任意の文字列と 160bit の鍵により、160bit の値を出力する。本章では、入力値に ID 、鍵には ts (32bit) と $intvl$ (32bit) に加えて、予測不可能な値として残りの 96bit は乱数を使用する。 ID は真の識別子であり、形式が自由な任意の文字列とする。

表 5.1. 各サーバで管理される情報

サーバ	情報の種類
登録	移動体の IP アドレス・HID サーバの IP アドレスおよび tag ・エリアサーバの IP アドレスおよび tag
HID	HID ・位置情報・付帯情報
エリア	HID ・位置情報・付帯情報

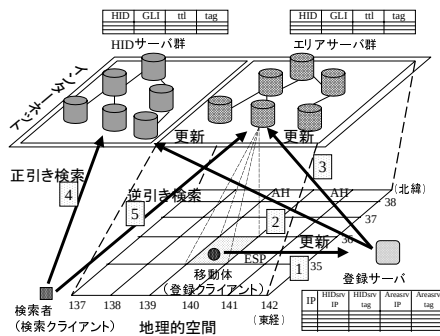


図 5.4. プライバシ保護を考慮した GLI システム 構成

5.5.3 構成

本提案システムの構成を図 5.4 に示す。本システムは、登録サーバ群、HID サーバ群、エリアサーバ群という 3 種のサーバ群と、登録クライアント、検索クライアントという 2 種のクライアントから構成される。

登録クライアントは、GPS などの位置取得装置から位置情報を取得し、その時刻における *HID* を生成する。*HID* と位置情報・付帯情報および有効期限 *ttl* を予め関連づけられた登録サーバへと送信する（図 5.4-(1)）。

登録サーバは、登録を受け付ける登録クライアントを認証し、予め関連づけられた登録クライアント以外からの登録は受け付けない。登録クライアントから登録サーバへの通信は移動体を特定する情報を含むため、ESP を使用し盗聴・改竄を防止する。同時に、登録サーバは ESP によって登録クライアントを認証する。登録サーバは、受け付けた位置情報および付帯情報を蓄積せず、*HID* の値によって決定される HID サーバ、および位置情報の値に従って決定されるエリアサーバに、*HID*、位置情報・付帯情報、有効期限を送信する（図 5.4-(2)、(3)）。HID サーバおよびエリアサーバは登録されたエントリをそれぞれのデータベース内で一意に示す情報である *tag* を登録サーバに返す。登録サーバと HID サーバおよびエリアサーバの通信は、移動体を特定する情報を含まないため、AH を使用して改竄のみを防止する。

各サーバで管理される情報を表 5.1 に示す。登録サーバは、登録クライアントから登録を受け付る際にその IP アドレス、*HID* の値によって決定される

HID サーバの IP アドレス、位置情報の値によって決定されるエリアサーバの IP アドレスを保持する。また、HID サーバ、エリアサーバへの登録時に返されるデータベースエントリの *tag* も保持する。HID サーバは、登録サーバから *HID*、位置情報・付帯情報、有効期限を受信して蓄積する。また、HID サーバは、検索クライアントから *HID* を鍵とした移動体の検索である正引き検索を受け持つ（図 5.4-(4)）。エリアサーバは、そのエリアサーバが管理している位置領域に存在する移動体の *HID*、位置情報・付帯情報、有効期限を登録サーバから受信して蓄積する。また、検索クライアントからの位置領域を鍵とした移動体の検索である逆引き検索を受け持つ（図 5.4-(5)）。

5.6 結論

本章では、移動体のプライバシー保護機能を付加した GLI システムを提案した。本提案方式では、移動体の識別子と位置情報・付帯情報のうち、識別子をハッシュ関数に入力させることで得られた値を *HID* (Hashed ID) として利用する。これにより移動体の信頼関係にない検索者からの特定を防止した。また、ハッシュ関数の入力値として時間変化する鍵を信頼関係のある検索者と共有することにより、信頼関係のある検索者には特定を許し、時間とともに変化するため追跡の可能性を減少させることができた。また、データベースを分割することにより盗難による移動体の特定を防止した。また、この設計を基にした実装を行い、実験運用を行った。さらに、グループでの *HID* 生成情報共有、鍵の変更と配送などに関する検討を行う予定である。

第 6 章 自動車情報の抽象化

自動車の持つ個々の情報を収集し、蓄積/加工するシステムでは、扱う情報に関する標準規格の必要性は高い。本章では、普及を踏まえたインターネット自動車システムを構築する為に必要とされる、自動車の持つ情報の規格化・標準化について述べる。

6.1 自動車の抽象化

インターネット自動車システムは、自動車の情報化において、通信とサービスを分離している。自動車への情報の提供や、自動車が持つ情報を他システムが利用できる基盤システムを独立させることで、自由な情報の交換を支援している。

また、自動車をインターネット上の一つのノードとして捉え、インターネット上から取り扱える環境を構築している。このため、インターネット自動車システムでは、インターネット上で動作する様々なソフトウェアを利用できる。これはインターネット自動車システムの大きなメリットである。

一方、自動車の情報をインターネット上で利用する為のシステムも構築している。自動車の情報は、車載計算機上で自動車が搭載されている各種センサの状態を保持している。インターネット側と車載計算機は、通信インターフェースを介して情報の利用が可能である。

6.2 規格化・標準化

規格化・標準化によって構築される共通基盤に最も重要な要素は、必要な情報について容易に、統一的に収集することができる点である。

車両毎に保持する情報の種類、精度、単位などにばらつきがあると、情報を集約、共有する事が困難である。また、システムの普及には、どこにどのような情報があるかをシステム内外に明示する必要がある。同時に、既存/新規の情報システムと協調動作を実現すれば、車両データと、既存の情報システムの基礎データから、より高精度で利用価値の高い情報の生成が可能となる。それぞれが補完しあう事により、データ取得範囲の広域化や、データ自身の細密化、さらにそれを扱う利便性の向上が見込まれる。

こうして収集された情報を元にした情報の生成、および利用者への提供は、各サービス主体が独自に行えば良い。これらはニーズに応じて柔軟に細分化されるべきであり、統一規格による基盤整備の要望は薄い。

したがって、規格化・標準化は自動車における情報の保持部分と、他システムとの通信が発生する通信部について行われる必要がある。

このうち通信プロトコルに関しては、扱うデータや利用法によって最適なものが異なる。通信プロト

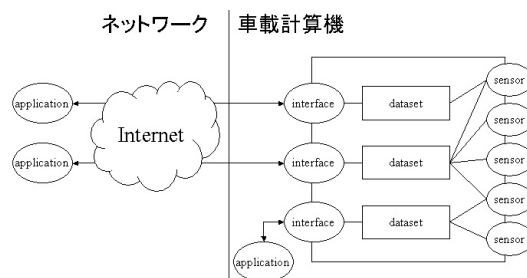


図 6.1. システム独自のインターフェースによる情報収集

コルの規格化・標準化を検討するためには、その通信によって、どのような情報が、どのような目的で取得されるかを明らかにする必要がある。そのため、データの保持部に関してデータ構成を定め、どのような情報がどの部分に存在するかをシステム内外に明らかにする必要がある。

6.2.1 自動車情報インターフェース

現在、自動車の情報をインターネットから利用する場合には、システム毎に独自の通信インターフェースを車載計算機とインターネット上のノードに用意している。これは、システムが主として車載計算機での情報取得や通信機器によるインターネットへの接続実証実験のために構築されてきた事に起因する。現在の車載計算機上の情報システムを図 6.1 に示す。

この状態では、新規アプリケーションの実現には自動車上のシステム、インターネット上のノードのシステム双方を構築しなければならない。また、車載計算機が持つ情報がサービスに依存しているため、複数の自動車の情報を画一的に取得することは難しい。

これまでの研究で、自動車をノードとしてインターネット上から認識し、接続性の確保および通信の継続が可能となった。今後、インターネット自動車システム上で、誰もが自由にアプリケーションを開発できる環境を提供するためには、自動車の持つ情報に容易にアクセス可能とする、統一されたインターフェースが必要である。

統一した情報の利用インターフェースが確立されれば、それらを用いることでサービスシステムの開発コストを大幅に下げることが可能となる。また、情報の加工・生成や交換、共有や再利用が容易になり、情報環境全体の価値を高めることができる。

図 6.2 に示すような、何らかの情報管理システムによって自動車上の各種センサから取得した情報を、

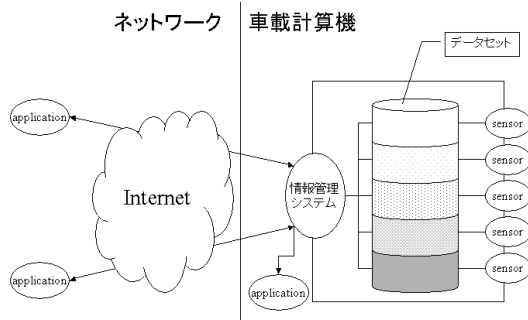


図 6.2. 情報管理システムの概要

データセットとして保持する形が、情報を利用する上では利便性が高い。

そこで、自動車の持つ情報についてのデータセットを規格化を検討した。現在の車両は個々の車両種により取得できる情報に差異があるため、画一的な情報収集を行うことが非常に困難である。自動車の持つ情報について統一されたデータフォーマットが用意され、車両が持つセンサ取得データがそれに基づいて正規化されて保持されていれば、インターネット上から自動車の情報を画一的に利用することが可能となる。

6.2.2 データセット

自由な情報のやりとりをインターネット上で行う為には、自動車で用いるデータセットは拡張が容易でなければならない。新たなデバイスからの情報も、データセットの拡張のみで対応が可能であれば、自動車の扱う情報が変化しても、データセットのみを更新すれば良い。

自動車の持つ情報は車両毎にセンサの種類や精度にばらつきがあり、その全てを網羅したデータ構成を用意することは難しい。車両が持つ情報には車両の性質による特殊な情報や、事業主体に依存した情報も多く、これらを全て標準化するのは非常に困難であり、得策でない。また、全てを網羅したデータ構成を、全ての車載システムに搭載することは増長であり、システム自体の肥大化を招く。

そのため、今回の標準化作業では、データ構成を基本データセット（必須）と拡張データセット（任意）の二種類に分け、規格化を行った。基本データセットについては IPCar システム車載機器には必ず対応していなければならないものとした。こうする事で最低限必要な情報に関しては、システム上で必ず利用できる環境が整う。

拡張データセットについての対応は任意であり、利用者や事業主体が必要に応じてその内容を吟味し、必要なものについては対応するという形を取る。データセットは木構造を取っているため、容易に任意の拡張データセットに対して追加/編集が可能である。また、利用者や事業主体による拡張も自由に行うことが可能であり、必要に応じてそれを公開することで、システム・センタ間での拡張データの情報共有を図る事が可能である。データセットの構造概念を図 6.3 に示す。

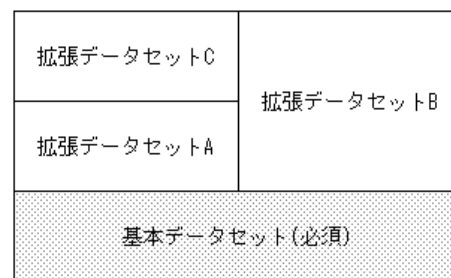


図 6.3. データセットの構造

基本データセットとして扱われるべきと思われる情報には、以下のような情報が挙げられる。

carid

車両の識別子を示す。

status

車両の状態（車載機の電源状態等）を示す。

timestamp

情報の更新時間を示す。

latitude

車両の緯度を示す。

longitude

車両の経度を示す。

altitude

車両の高度を示す。

speed

車両の速度を示す。

また、拡張データセットの例としては、以下のようない情報が挙げられる。

light

車両のヘッドライト点灯状況を示す。

wiper

車両のワイパ動作状況を示す。

width

車両の車幅を示す。

height

車両の車高を示す。

weight

車両の重量を示す。

6.3 ASN.1 を用いたデータセットの定義例

実際のデータセットについては、ASN.1 によって記述し、自動車の情報を MIB として扱うことや、XML によって記述し、http によって情報交換を行うという設計も考えられる。特に、インターネット上からの情報収集を考慮すると、システムに対応した車両が扱う情報に対して、SNMP(Simple Network Management Protocol) 等、既存の情報管理システムの利用も検討すべき要素である。

ここでは例として、ASN.1 によるタイムスタンプと緯度・経度データ構造の定義を挙げる。ASN.1 によってデータセットを定義する事で、MIB による自動車情報の管理は容易に実現でき、これを規格化する事で、自動車間の情報粒度の差異や、センサの精度に捕られる事なく、自由なシステムの構築が可能となる。

● ASN.1 によるタイムスタンプデータ構造の定義

```
-- timestamp BASIC
timestampTable OBJECT-TYPE
    SYNTAX SEQUENCE OF TimestampEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A list of timestamp"
    ::= { datasetbasic 1}

timestampEntry OBJECT-TYPE
    SYNTAX TimestampEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "table of ts"
    ::= { timestamptable 1 }
```

```
timestampEntry ::=
    SEQUENCE {
        basicyear INTEGER
        basicmonth INTEGER
        basicday INTEGER
        basichour INTEGER
        basicminute INTEGER
        basicsecond INTEGER
    }

basicyear OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    ::= {timestampEntry 1}

basicmonth OBJECT-TYPE
    SYNTAX INTEGER (1..12)
    ACCESS read-write
    STATUS mandatory
    ::= {timestampEntry 2}

basicday OBJECT-TYPE
    SYNTAX INTEGER (1..31)
    ACCESS read-write
    STATUS mandatory
    ::= {timestampEntry 3}

basichour OBJECT-TYPE
    SYNTAX INTEGER (0..23)
    ACCESS read-write
    STATUS mandatory
    ::= {timestampEntry 4}

basicminute OBJECT-TYPE
    SYNTAX INTEGER (0..59)
    ACCESS read-write
    STATUS mandatory
    ::= {timestampEntry 5}

basicsecond OBJECT-TYPE
    SYNTAX INTEGER (0..59)
    ACCESS read-write
```

```

STATUS mandatory
:: = {timestampEntry 6}

• ASN.1 による緯度・経度データ構造の定義

-- position
positionTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PositionEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A list of latitude & longitude."
    ::= { datasetbasic 2}

positionEntry OBJECT-TYPE
    SYNTAX PositionEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "table of positionEntry"
    INDEX { latitudeDeg }
    ::= { positionentryTable 1 }

PositionEntry ::=
    SEQUENCE {
        latitudeDeg INTEGER
        latitudeMin INTEGER
        latitudeSec INTEGER
        longitudeDeg INTEGER
        longitudeMin INTEGER
        longitudeSec INTEGER
    }

latitudeDeg OBJECT-TYPE
    SYNTAX INTEGER (-90..90)
    ACCESS read-write
    STATUS mandatory
    ::= {positionEntry 1}

latitudeMin OBJECT-TYPE
    SYNTAX INTEGER (0..59)
    ACCESS read-write
    STATUS mandatory
    ::= {positionEntry 2}

```

```

latitudeSec OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    ::= {positionEntry 3}

longitudeDeg OBJECT-TYPE
    SYNTAX INTEGER (-180..180)
    ACCESS read-write
    STATUS mandatory
    ::= {positionEntry 4}

longitudeMin OBJECT-TYPE
    SYNTAX INTEGER (0..59)
    ACCESS read-write
    STATUS mandatory
    ::= {positionEntry 5}

longitudeSec OBJECT-TYPE
    SYNTAX INTEGER (0..59)
    ACCESS read-write
    STATUS mandatory
    ::= {positionEntry 6}

```

このようにデータの表現形式とその解像度を定める事で、情報の円滑な利用を促進する。しかし、こうしたデータ構造の定義を標準化案とする為には、扱う情報のデータ形式や解像度が必要十分であるかの検討や、実際の利用による不具合の修正等を行う必要がある。

第7章 高精度な衛星測位基盤に関する研究

7.1 はじめに

インターネット自動車のみならず、物理的なオブジェクトがネットワーク化される場合において位置情報はもっとも重要な情報のうちの一つである。このような位置情報を得るシステムは数多く提案されており、移動体においてはその中でも電波を用いた手法が利用されているが衛星測位以外の手法では、

小型の移動設備により広範囲に高精度な位置情報を得ることは難しい。

衛星測位は地球上での絶対位置を測定出来ることや、事前の調査などが必要でないことから将来的にもっとも一般的な測位技術として利用されると考えられる。しかしながら、衛星測位はそのシステムの限界により単独での精度が 10 m 程度となっており、より高精度な測位が求められている。

衛星測位システムの高精度化は位置情報の提供サービスに必須の機能であり、用途にもよるが数 cm から数 m 程度の精度の高い測位が求められており、高精度での測位を広域で利用するためのシステムが必須である。

高精度かつ高信頼性をもった衛星測位システムには地上での定点観測を行う観測局とそこからの補正情報および健全性を知らせるメッセージが不可欠であり、観測局を広域に展開する必要がある。本グループではこれらのシステムをインターネットを用いて構築することを目標として研究を行っている。

本年度は、インターネットを用いた基準局を構築し、その応用として RTK-GPS を広域で行う仮想基準点方式での実験を行った。また本報告には含まれないが、主にプロトコルを設計するために必要なデータを得るための実験として、D-GPS におけるデータ遅延時間と精度の関係および、プロトコルや帯域幅と精度の関係を計測している。

7.2 基準局ネットワークを利用した補正環境の構築

7.2.1 仮想基準点方式による測位

仮想基準点方式では、複数の基準局からの情報を元に観測地点近傍において仮想的な基準局をおき、その基準局からの補正データを GPS 受信機に与える。仮想的に基線長を短縮することが可能であり、従来の RTK-GPS では不可能な範囲であっても高精度な測位が行える。

観測地点近傍にある基準局を計算するためには、現在の観測局の位置を元に基準局を計算する必要がある。通常の基準局からの情報とは異なり、2 次元的な広がりをもつ必要があるため、通常の RTK-GPS の補正情報に比べ多くのデータを必要とする。

このようなネットワーク RTK を行うためには複数の基準局からの情報をリアルタイムに収集し、処理する必要がある。そこで我々はインターネットを用いた基準局ネットワークを構築することにより低

コストかつ安定した広域基準局ネットワークを構築した。

7.2.2 インターネットを用いた基準局ネットワーク

インターネットを用いた GPS 基準局を構築するため、情報配布モデルと同モデルを用いたシステムの設計および実装について述べる。

7.2.3 インターネットを用いた双方向 GPS 補正

インターネットは世界でもっとも多くユーザが利用しているコンピュータネットワークである。インターネットは階層構造を持つプロトコルにより物理的な通信手段を仮想化することで、さまざまな媒体を利用することが可能である。そのため物理的な通信媒体の進歩にあわせてネットワークを拡張することが可能であり、将来の通信技術の発展にも簡単に対応できる。

インターネットは双方向の通信であり、サーバからの情報伝達のみならず、個々のクライアントからの要求を伝えることが可能である。また、インターネット上の通信は端末間のみで行われるため独自にプロトコルを開発し実装することが容易である。インターネットを用いて基準局から情報を配布する場合には同時に多数の相手に対してサービスを行う必要がある。一つのサーバから提供できるデータの量には限界があるため、補正情報の利用者と基準局サーバの間に中継サーバをおきデータの再配布を行う。本システムでは図 7.1 に示すように双方向通信を利用し、利用者は自分の目的に適合した補正情報を選択することが可能である。

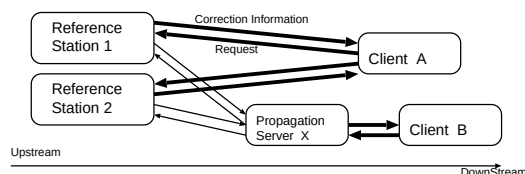


図 7.1. インターネット補正情報配信モデル

7.2.4 インターネット基準局の構築

我々の提案するシステムはユーザからのリクエストを元に配布する情報を決定する。このプロトコルでは基準局のリストや補正情報の種類といった情報を得ることが可能である。これらの情報を配布することにより、クライアントや下流の中継サーバは基

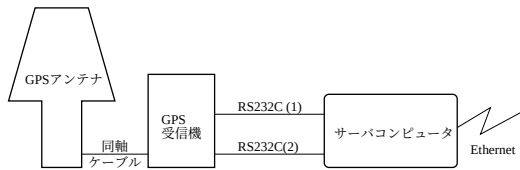


図 7.2. インターネット基準局

基準局情報の動的な変化に対応することが出来る。本プロトコルはインタラクティブに設計されており、コマンドとその返答で構成される。コマンドにより、基準局のリスト、補正情報のリストを閲覧できる。ユーザはこれらの情報を元に、自分の必要とする補正情報を決定し、要求コマンドを送付する。

補正情報の配信は信頼性のある双方向通信プロトコルである TCP を用いて実装されている。近年のインターネットではファイアウォールを構築することが多い。基準局を多くの場所に設置する場合には組織内部の場所を利用することを考慮し、サーバ側からコネクションを開始し、下流へとデータを送るサーバプッシュ型のプロトコルを採用した。さらに Socks と呼ばれるアプリケーションゲートウェイを用いたファイアウォールにも対応している。これらの方法により、組織ネットワークや企業ネットワークといったファイアウォールによって管理されている LAN から基準局情報を提供することが可能となる。

上流のサーバを発見するための方法は別途定義することとし、本プロトコル内部では扱わない。現在のシステムでは設定として記述する方法を用いているが、将来的にはディレクトリサービスなどを利用することを想定している。

7.2.5 サーバの設計と実装

我々はインターネットに接続することを考慮した基準局を設計した。図 7.2 にインターネット基準局のシステム模式図を示す。基準局はアンテナおよび受信機から構成される GPS 部と計算機によりデータの処理や通信をうけもつネットワーク部から構成される。ネットワーク部はインターネット接続用の UNIX サーバを構築し、シリアルポートを経由して受信機を接続している。シリアルポートは2つ接続されており、一つは基準局の設定やモニタリングに利用し、他方を補正データの取り出しに利用している。そのため外部へのネットワーク接続を確保できる地点であれば、容易に基準局を構築可能であり、外

部から設定や調整を行える。

GPS 部はマルチパスを除去できる高精度測量用のアンテナおよび、RTK のためのデータを毎秒出力可能な測量用 GPS 受信機を用いる必要がある。サーバ上ではインターネットへの接続を受け持つ独自プログラムが動作しており、サーバプッシュ型で動作する。補正情報は RTCM の形式で入力され、RTCM のタイプ毎にプッシュすべきクライアントを判断する。ユーザは複数の補正情報の中から必要とする情報を選択し要求することが出来る。

インターネットは汎用のデータ通信手法であり、世界でもっとも多くユーザが存在するネットワークである。インターネットのプロトコルは階層構造を持ち、物理層に依存しない接続を実現できる。そのため、新しいデバイスへの対応が簡単に行える。インターネットは双方向の通信であり、放送のように単一方向ではない。そのため、ユーザからの要求をサーバに対して伝えることが出来る。サーバはユーザからの要求に応じたデータをそれぞれのユーザに対して通信することが出来る。また、新しいデータを配送することが必要な場合の対応を簡単に行うことが出来る。インターネットを用いた通信は基本的に 1対1の通信を行う。多くのユーザをサポートするために中継サーバを利用する。中継サーバは基準局と移動体の間に置かれ基準局からのデータを再配布する。中継サーバは複数置くことが出来る。また、中継サーバ同士を接続することも可能であり、これらが複合してシステム全体はツリー構造を作ることが出来る。

クライアントが接続する中継サーバはディレクトリサービスなどの手法により発見する。これはインターネットが単一の通信のみならず別のサービスを同時に行うことが可能であるためである。このように、サービスを分類することにより一つのシステムが持つ必要のある機能を減らし、単純化することが可能になる。

7.2.6 仮想基準点方式の実験

提案するシステムを実装し評価実験を行うことにより、本システムの有効性を検証した。特に基準局からの基線長が長い場合には RTK-GPS では FIX 解を得ることが出来ないが、本システムの場合には解を求めることが可能である。また、インターネットを用いることにより、広域での基準局が展開可能

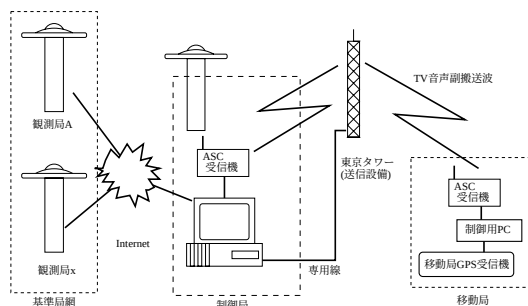


図 7.3. VRS 実験システム

であり、ASC を用いることにより、広範囲での補正データの受信も可能にしている。

7.2.7 ネットワーク RTK システムの構成

本システムはネットワーク RTK のシステムを構成するために、最低 1 つの制御局と 4 つ以上の観測局から構成される。図 7.3 にシステム全体の構成を示す。

7.2.8 実験局配置

観測局は GPS 受信装置およびインターネット接続用サーバから構成されており、それぞれ Ethernet を用いて LAN 環境に接続されている。これらのシステムが接続されている LAN はそれぞれの組織に既存のものである。制御局はコンピュータと GPS 受信機、インターネット回線および TV 送信設備への通信回線から構成される。制御局では GPS 受信機からの補正情報とインターネットを通じて受信される観測局からの補正情報を元にグリッドデータと呼ばれる補正情報を構成する。再構築されたデータは専用回線を用いて TV 送信設備に送られ、ASC データへエンコードされ、東京タワーより放送される。

本実験では東京都新宿区に設置した制御局を含め、7 つの観測局を配置した [41]。それぞれの局は表 7.1

のとおりであり、そのおおまかな位置を図 7.4 に示す。

観測局からの補正情報はインターネットを通して情報は制御局へあつまる。これらの補正情報と制御局が持つ受信機からの補正情報を利用してネットワーク RTK のための補正情報が計算され、専用線を通じて東京タワーにある放送センターへと送られる。放送センターでは送られたデータを ASC にエンコードし、TV 音声に重ねて放送を行う。

移動局はホイップアンテナあるいは八木アンテナおよび専用のデコーダを搭載した受信機を持つ。受信機から出力された情報を用いて、観測地点に設置したコンピュータで観測地近傍の仮想的な基準局を計算し、GPS 受信機へ送る。GPS 受信機は観測地点にあわせて再計算された補正情報を元に RTK 測位を行うことが可能であり、この場合には基準局からの基線長が問題となることはない。

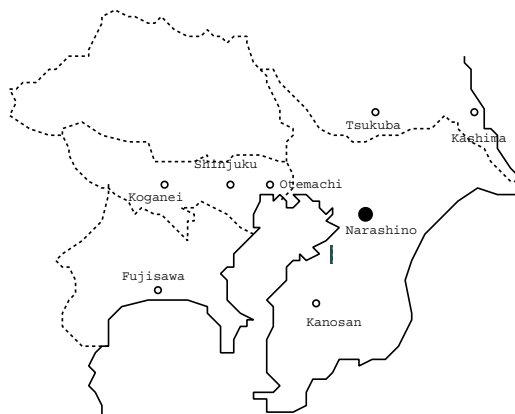


図 7.4. 実験に利用した観測局配置

7.3 結論

本年度の研究では主に伝送プロトコルに関する実験を行った。まず、D-GPS の補正情報の遅延と精度

表 7.1. 利用した観測局

局番号	組織	場所	種別
1	オープンネットワーク社	東京都新宿区	制御局
2	WIDE 大手町	東京都千代田区	観測局
3	慶応大学 SFC	神奈川県藤沢市	観測局
4	国土地理院	茨城県つくば市	観測局
5	国土地理院	千葉県鹿野山	観測局
6	CRL 小金井	東京都小金井市	観測局
7	CRL 鹿島	茨城県鹿島市	観測局
8	日立習志野	千葉県習志野市	移動局

に関する評価を行った。インターネットを用いた場合の遅延時間がどのように精度に影響するかを考察し、現在のインターネットを用いた遅延では十分な精度が得られることを確認した。

次に伝送帯域を制限した場合において TCP、UDP それぞれの性能を比較した。その結果、帯域を制限した場合においては TCP を用いた結果がよいことが判った。

その他にインターネットを用いた基準局ネットワークを構築し、ヘテロなネットワーク環境においてもインターネット基準局網を構築できることを示し、その応用として仮想基準点方式による測位実験を行った。

今後はこれらの結果を元に基準局ネットワークおよび移動局ユーザに対するプロトコルの設計および実装を進めることを予定している。

第8章 総括

自動車は国民に深く浸透した身近な存在である。それだけに、情報化され、より便利に利用できることは大きなインパクトを生む。非常に身近に存在する自動車が情報化されることにより、人々は情報化の動きを直接体験できると同時に、そのリテラシ向上にも貢献すると考えられる。また、我が国において自動車は経済を支えて来た産業の一つであり、先行して情報化を進めることにより、これからの流れにおいても重要な地位を占めることが可能となる。

現在、本ワーキンググループでの活動は社会的も注目度の大きいものとなっており、産業界をリードするにも適当な時期にきている。このことを鑑みて、研究分野と実用化分野を見極め、産学官共同でプロジェクトを進めていくことが不可欠となる。来年度は、これまで開発してきた技術の一部の実用化と研究のさらなる深化を進める。

