卒業論文 1999 年度 (平成 11 年度)

インターネットを利用した GNSS 補正情報配送機構設計と実装

慶應義塾大学 環境情報学部
川喜田 佑介
kwkt@sfc.wide.ad.jp

指導教員
村井 純
徳田 英幸
楠本 博之
中村 修

平成 12 年 2 月 28 日

概要

インターネットを利用した GNSS 補正情報配送機構の設計と実装

本研究ではインターネットを用いたGNSS(Global Navigation Satellite System) の補正情報配送機構を提案し，このモデルに基づいて設計と実装を行った．

はじめに，既存の補正情報配送機構に関しての考察を行い，サービスエリアの狭さや導入時のサービス提供者の負担の大きさなどの問題点を示した．これらの問題解決のため，インターネットを用いた補正情報配送機構を構築するアプローチを選択した．既存の補正情報配送機構では導入時に補正情報の有効範囲や補正の種類等について決定ている．インターネットを用いる際にそれらに幅広く対応する必要がある．これを解決するために，補正情報についての表現を行い，補正情報の有効範囲や補正の種類等に基づいた要求応答を可能にした．

また，本機構では配送局という概念を導入した．これは補正情報の配送を専門に行い，やはり補正情報の有効範囲や補正の種類等に基づき複数の要求に対し対応できる．また，利用者の要求に応じた補正情報を配送するために補正情報配送プロトコル，補正情報配送制御プロトコルを提案，設計した．同プロトコルを使用してインターネットを用いる補正情報配送機構を設計，実装，評価した．その結果既存の補正情報配送機構の問題点である送受信設備や規模性等の点を解消できた．

本機構の実現により，利用者はインターネットに接続することで，GNSS補正を容易に行える．大掛かりな無線設備等が必要でないため，既存の機構に比べ補正サービスの提供者の負担を軽減できる．将来構築されるであろう補強機構の実現に必要となる補正情報の配送管理を容易にする．以上よりインターネットを用いて補正情報配送を行うことは十分に利点があると結論づける．

慶應義塾大学 環境情報学部
川喜田 佑介

Abstract

The Design and Implementation of the GNSS Correction Data Distributing System on the Internet

In this research, a GNSS correction information distribution mechanism using the Internet is proposed, and the distribution mechanism is designed and implemented based on this model is discussed.

Evaluations of present correction information distribution mechanisms indicate problems such as limited service area and the burden on the service provider upon installation.

The approach chosen here to solve these problems is to build a correction information distribution mechanism using the Internet. With present correction information distribution mechanisms, the area of correction information validity and correction types are determined on installation. If the Internet is to be used, the validity and types need to be more flexible. Therefore, there was need for request and respond actions based on the area of correction information validity and correction types, and the correction information was made more abstract.

Also this mechanism introduces the notion of a Propagation station. This station is a server that specializes in the distribution of correction information, which distributes the correction information after altering the format according to the request. For the distribution of correction information according to the users request,correction information distributing protocol and correction information distribution control protocal are presented and implemented. In turn, a correction information distribution mechanism using the Internet in combination with these protocals are designed, implemented and evaluated.

In addition, it is shown an augmentation system that manages the correction information and enables stable positioning with high accuracy is possible with the use of this mechanism as a base structure. Uses will be able to perform GNSS correction easily by connecting to the Internet. Because there is no need for large scaled wireless equiptments, the burden on the correction service provider can be reduced compared to present mechanisms. Also, the distribution management of correction information, necessary for when this augmentation system is realized in the future, will be made easier.

Finally, it is concluded that correction information distribution using the Internet has great advantages.

Keio University Faculty of Environmental Information
Yuusuke KAWAKITA

目 次

図目次

表 目 次

「第 1 章」 序論

本章では，背景となる移動体計算機環境の概要について述べ，本研究の目的を説明する．また，本論文の構成を示す．

第1.1節」　本研究の背景

カーナビゲーションシステムなど，我々は日常的に計算機上で地理的位置情報を利用している．それらの多くはGPS(Global Positioning System)[1]を始めとする，衛星測位機構 (以下 GNSS: Global Navigation Satellite System) を利用している．しかし，衛星測位機構による測位は誤差が大きく，GNSS のうちもっとも一般的に利用されている米国の GPS の民事利用では単独測位で約 100m 程度の誤差が生じる．その結果，カーナビゲーションシステムではマップマッチングと呼ばれる測位後の位置補正技術が使われている．これは計算機内の道路地図を用いもっとも確らしい位置を表示するというものである．同様に，人間が計算機を持って移動する移動体環境では，さらに位置の粒度が細かくなり利用しにくい．そもそも，計算機とその利用者が地図道路上にいるとは限らず，マップマッチングの使用はできない．

そこで，衛星測位での精度を高める研究，サービスが多く存在する．このうち，差動方式と呼ばれるものがある．差動測位は2地点で同時に測位し，その際の共通の誤差要因について取り除こうと言うものである．ここで，2 地点の一方を基準局 (BS:Base Station)，もう一方を移動局 (RS:Rover Station) とする．基準局ではあらかじめ高精度で緯度経度高度の位置情報が分かっている．この方式はその情報を利用して測位衛星と地上の間の大気の影響などの共通の誤差要因を取り除き移動局での測位の精度の向上を図るものである．本論文ではこの測位精度の向上を補正と呼ぶ．こういった方式を取るものにディファレンシャル (differential) 方式，キネマティック (kinematic) 方式がある．

実時間的な測位を行う場合何らかの手段で補正のための情報を基準局から移動局に送信する必要がある．本研究ではこの実時間的な補正を主な精度向上の手段とする． 実時間的なキネマティック方式をリアルタイムキネマティック (RTK: Real Time Kinematic) 方式と呼ぶ．

2000 年現在，米国のGPS以外にもロシアのGLONASS，EUのGALILEO等が部分運用中，または，計画中である．それぞれのディファレンシャルやリアルタイムキネマティック方式の補正をDGPSやRTK-GPS，DGLONASS等と呼ぶ．また両方式を併せて D/RTK-GPS 等と呼び，差動方式一般に D/RTK-GNSS と呼ぶことがある．

第1.2節」　本研究の目的

一般に，D/RTK-GNSSを用いた測位にはGNSS受信機と補正情報を送受信するための機器，測位した情報を得るための計算機が必要である．従来，D/RTK-GNSSの補正情報はFMラジオの副搬送波，MCA無線，中波ビーコンを用いて配送されていた．これらの方式には導入時のサービス提供者側の負担が大きい，専用の受信機が必要になるなどの数々の問題点があった．そこで，本研究では補正情報配送の通信路としてインターネットを用い，それらの問題点を解消する．

本研究では，主に以下のことを行う．

- 補正情報をインターネットを用いて配送する機構の設計，実装を行う．
- 補正情報をインターネットで配送する際のプロトコルの仕様を設計する．

本研究では，主に移動計算機の利用者が高精度のD/RTK-GNSSを利用する際，補正情報を用意に利用できる環境を提供する．

具体的には，基準局からインターネット上のユーザへ補正情報を配送する機構および，配送プロトコルについて設計，実装する．

従来補正情報の多くはRTCM SC-104形式であった．この形式は本来電波を通信路とする際に使用されるものであり，インターネット上をRTCM SC-104形式を用いて補正情報を配送するのは効率が悪い．そこでインターネットを用いて補正情報を配送するのに適したプロトコルを設計する必要がある．

また，インターネット上ではサービスエリアを限定する必要はない．よって，移動局で使用するのに最も適した補正情報が配送されることが望まれる．そのためには補正情報について抽象化を行い要求に基づく配送が行われる必要がある．

第 1.3 節　本論文の構成

第 2 章では，既存補正情報配送技術について紹介し，その際に生じる既知の問題点について述べる．また，インターネットを用いて補正情報配送をすることの意味と期待される効果について述べる．

第 3 章では，本機構でインターネットを用いる補正情報配送のモデルについて提案し，本機構の目的と設計について述べる．また，その際に使用される配送プロトコルを提案し，その仕様を示す．

第 4 章では，3 章で述べた機構の実装について述べる．

第 5 章では，本論文で実装したインターネットを用いた補正情報配送機構についての評価を行う．既存配送機構との相対的な定性，定量的評価を行う．また，本機構におけるさまざまなパラメータについてそれが性能におよぼす影響を評価する．

第 6 章では，以上の議論をまとめ，今後の課題を述べる．

最後に第 7 章で結論を述べる．

「第2章」既存の補正情報配送技術

本章では，本論文で話題にする衛星測位と補正技術に関する概要について述べ，次に既存の配送技術について述べる．また，それらの分類と問題点の指摘を行う．

第2.1節　衛星測位と補正技術の概要

ここでは衛星測位と補正情報の配送技術に関して調査し，概要を述べる．

第2.1.1節　衛星測位概要

地球上ある地点の緯度経度について知るというのは困難である．はじめて実用的な測位方法として確立した方法は天文航法である．天文航法は方位を知ることができたが，現在位置，特に経度を知ることができなかった．その後多くの航法が研究され現在もっとも先進的な人工衛星を用いた航法が確立された．

衛星測位とは人工衛星による三次元測位機構であり，一般にGNSS(Gloval Navigation Satellite System)と呼ばれる．現在，GNSSには構想中のものを含め3つが存在する．米国のGPS(Gloval Positioning System)が運用中であり，ロシア(旧ソ連)のGLONASSは部分運用，欧州連合(EU)のGALILEOが構想中である．

日本国内では主にGPSが使用されている．現在，カーナビゲーションシステムや登山用の小型GPS受信機などの浸透によってGPSという用語は一般に広く使用されるまでになった．

衛星測位は軌道上20,000kmの円軌道(GPSの場合)を飛行する24個からなる測位システムであって，主目的は米国とその同盟国の軍事用であるが，一部の電波信号を限定し民事用にも開放されており，軍事用に対し精度が劣るものの使用されている．

GPSはNNSS(Navy Navigation Satellite System)を元にしており，古くは衛星の代わりに陸上に無線局を設置した電波航法システムを起源としている．1900年代初頭から無線通信の発達にともない，航法にも電波を応用する試みがみられた．電波の発信点を方向探知機で探るものである．発信点を3個所以上おくことにより相対的な測位をする事も可能となった．本格的な電波航法のは1940年前後の英国のDecca[2]であり，これは第二次世界大戦中の連合軍のノルマンディ上陸作戦に使われ，その成功の一翼を担ったという．第二次世界大戦後は米国によって同様なLORAN A[2]，LORAN C[2]，Omega[2]などの研究，運用が行われていた．これらの電波航法システムの弱点はそのサービスエリアの狭さであり全世界的な運用は難しかった．航空機時代の到来によってサービスエリアのグローバル化が求められ，衛星測位システムの構築へとつながっていった．

一方，ロシアでは，旧ソ連時代に米国のGPSと類似のシステムの開発が行われ，運用が行われている．経済的な混乱の中で完全運用とはいかないまでも運用は続けられており，GPSと同様に利用できる．厳密な意味では電波の周波数や信号形式が異なるが，GPS，GLONASS両方を扱うことのできる受信機も民事用に広く出回る日も近い．

さらに，欧州連合ではGPSに頼らない，独自の衛星測位機構を計画している．これはGALILEOと呼ばれ，欧州連合と欧州宇宙機関(ESA)が計画中である．2000年現在では技術，運用およびビジネスにおける研究を行う定義期間だが，2003年までに軌道上検証，2008年までに運用を開始する予定である．

以上のように，衛星測位についてGPSだけでなく，GLONASS，GALILEOといった複数の機構を考慮する必要があり，本研究でもとくにGPSに限らずGNSS一般に考慮する．

「第2.1.2節」　GNSS利用の種類

本節では現在のGNSSの利用形態についてまとめる．GNSSの主な利用形態には単独測位，相対測位，時刻同期がある．

「第2.1.2.1節」　単独測位

未知点でGNSS受信機を運用し測位を行う．実時間運用が可能である．精度は次に説明する相対測位に劣る．GNSSの最も基本的な使用法である．

- **C/A コード利用**
  C/Aコードと呼ばれる周波を用いて測位を行う．これは民事利用向けで，精度100mである．

- **P コード利用**
  Pコードと呼ばれる周波を用いて測位を行う．これは軍事利用向けで精度16mである．

「第 2.1.2.2 節」　相対測位

相対測位には，多種多様な応用例が存在する．本論文では実時間運用可能なものについて言及する．差動 GNSS 測位 (D-GNSS)，干渉測位として，実時間キネマティック測位を主な対象とする．

- 差動方式
  ディファレンシャル方式とも呼ばれる．精度は C/A コードの利用で 2-3m 未満，P コードの利用で 1m 未満である．第 2.1.3節で詳説する．
- 干渉方式
  キネマティック方式とも呼ばれる．精度は 3cm 未満である．第 2.1.4.1 節で詳説する．

「第 2.1.2.3 節」　時刻同期

高精度の時刻同期をおこなう．同期精度は 100ns 程度である．実時間運用が可能で，単独測位との両立も可能である．本論文では利用しない．

「第 2.1.3 節」　単独測位

補正技術を利用する相対測位に言及する前に，単独測位について触れたい．単独測位は GNSS のもっとも基本的な運用方法である．地表面から人工衛星軌道内で利用できる．電波の到達しない地中，水中では利用できない．図 2.1に示すように同時に 4 つ以上の衛星を観測し，3 次元座標と受信機の時計の誤差の合計 4 つの未知数を解くことで測位を行う．(緯度，経度のみの測位を行う際には未知数が一つ減少するので 3 つの衛星で良い．)

また，本研究では触れないが，ドップラー測定を行う事により速度ベクトルの測定が可能である．これは航空，宇宙分野では利用されているが一般的でない．(カーナビゲーション，過般型の小型受信機での利用も論理的には可能である．)

単独測位において精度の低下を招く要因には以下のようなものがある．特に情報が公表されている GPS について述べる．

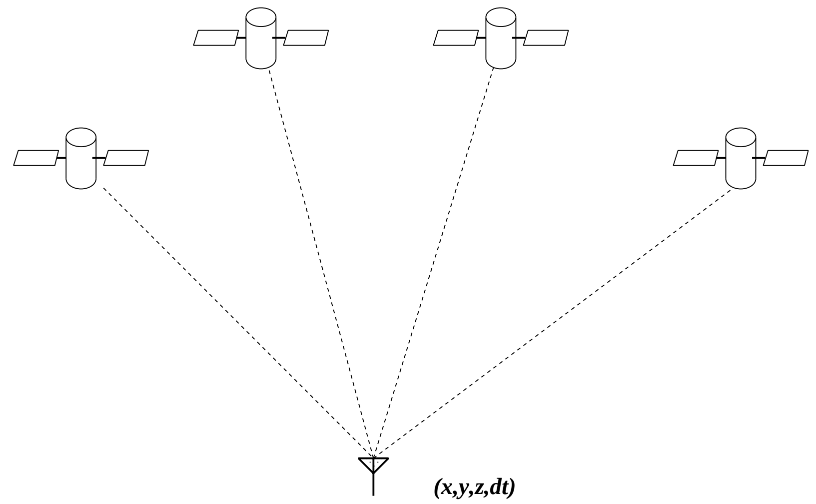


図 2.1: 単独測位

- 選択利用性 (**SA:Selective Availability**)
  米軍によって民事利用には精度劣化操作が行われる．L1，C/A コードによる測位の精度を作為的に劣化させている．

- **P** コード秘匿 (**AS:Anti Spoofing**)
  P コードはもともと軍事向であったが，手違いで 1980 年代末に誤って一般にその内容が流布されてしまった．これを再び秘匿化するために行われる操作をいう．P コード秘匿操作によってやはり作為的に精度の劣化が行われている．

- 電離層，対流圏の影響
  GPS 衛星と地上の GPS アンテナの間には電離層や対流圏がありそのあいだの状態によって電波到達までの遅延が変化する．これも誤差源となる．

- その他要因
  その他にも宇宙雑音電波，熱雑音，マルチパス，混信妨害といった誤差源が存在する．宇宙雑音電波，熱雑音はそれぞれ地球外からの電波や対流圏内の電波による電波雑音が原因である．マルチパスとは建物や地形の影響により複数の電波が衛星から到達してしまう局地的な誤差源である．混信妨害とは電波を発生する機器との電波混信が原因となる局地的な誤差源である．

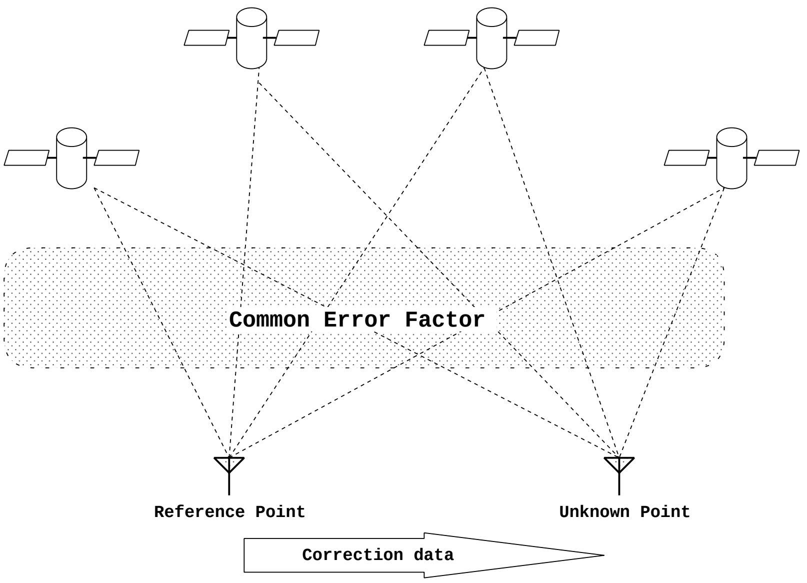


図 2.2: 差動測位

人工的な誤差発生源であるSA，ASを除き，これらの誤差発生源について，次に述べる差動測位や干渉測位で精度の改善を行うことができる．ただし，局地的な誤差源については参照地点と未知点両方で存在しないことが多くその誤差を取り除くことは難しい．

「第2.1.4節」　補正技術概要

ここでは相対測位に用いられる補正技術について，実時間測位を中心に述べる．

「第2.1.4.1節」　差動測位(ディファレンシャル方式)

相対測位の代表例として差動測位がある．これは，あらかじめ地理的な位置が既知な参照地点(reference point)で測位誤差を求め，それによって地理的な位置が未知な未知点ででの測位結果を補正して精度の向上を図るものである．図2.2のように，両地点での測定誤差の共通部分が相殺されて精度の向上が図れる．

次に述べる干渉測位も含めて，実時間測位の相対測位では，両地点で同時に測位を行い参照地点から何らかの方法で参照地点のデータを伝達する．

「第 2.1.4.2 節」　干渉測位 (キネマティック方式)

相対測位のもう一つの代表例が干渉測位である．干渉測位は差動測位の発展であり，両方式とも相対測位であるが，近年差動測位に改良が加えられその境界は曖昧である．両方式の最も大きな相違は単独測位用の受信機をそのまま使用するかどうかである．干渉測位では搬送位相の積算回路を備えた受信機を使用する．この回路で搬送位相の計算を行う．

干渉測位では差動測位と同様に未知点で最低 4 つの衛星を同時に観察する．測位の基本データは搬送波である．差動測位と同様に参照地点のデータを未知点に伝送することにより実時間測位を行う．

実時間測位でない場合は両地点の測位データを一カ所に集め解析を行う，所謂後処理補正と呼ばれる技術が存在する．相対測位一般に，実時間測位を行うには参照地点と未知点の間にデータ伝送路が必ず必要となる．

「第 2.2 節」　インターネットを用いない既存補正情報配送技術に関する調査

上記のように実時間測位の相対測位には測位データを参照地点から未知点へ，なんらかの通進路を用意して伝送する必要がある．ここでは，既存のデータ伝送技術の内伝送にインターネットを用いないものについてまとめる．

補正情報の配送には既存の技術が数多く存在する．特に伝送データに関しては補正情報として RTCM SC-104 形式が策定されている．

RTCM SC-104 形式は，誤差補正量，疑似距離補正量，座標値の各データの標準配送規格であり，正しくは Radio Technical Commission for Maritime Service SC-104 という．

RTCM SC-104 は米国の船舶電波技術委員会 (RTCM:Radio Technical Commission for Maritime Service) の中の第 104 委員会 SC-104(Special Committee No. 104) によって策定された規格である．これを単に RTCM SC-104 規格，RTCM 規格，RTCM 等と呼ぶ事がある．RTCM SC-104 の詳細に関しては付録 A を参照のこと．

以下，実時間相対測位のデータ伝送を補正情報の配送方法ごとにまとめる．

「第2.2.1節」　FMラジオの副搬送波を用いる方式

FMラジオの副搬送波を用いて補正情報を配送することで差動測位を行うことができる．この機構はすでに実用化[3]されており，カーナビゲーションシステムの多くはこのシステムを用いている．

- サービスエリア
  サービスエリアはFM電波の届く範囲．地理的に近接関係にある2地点でも電波の遮蔽などによりサービスが行えない場合がある．また，差動測位を行える範囲内で電波の到達する範囲はかなり限定されたものになる．
- 規模性
  サービスエリア内の規模性は考慮する必要がない．一対多の通信が可能．利用者側でサービスの選択を行う必要がない．むしろ，到達するものは全て使用できる補正情報であるといえる．
- 利用に際して必要なこと
  専用の受信機を必要とする．
- 精度
  帯域が狭域であるため，補正に必要な全てを伝達することができない．やむをえず変則的な補正情報を配送している．このため，十分な精度を得ることは困難である．

FMラジオの副搬送波を用いる方式ではRTCM SC-104形式のTYPE 9という補正情報の方式をとる．これは，同期信号がなく精度劣化の原因となっている．また，干渉測位には帯域が狭域過ぎるため，使用されていない．さらに，安定した補正を行うにはFMの信号強度の強い都市近傍である必要がある．

「第2.2.2節」　携帯電話を用いる方式

携帯電話をもちいて補正情報をを配送することで差動測位，干渉測位を行うことができる．

- サービスエリア
  サービスエリアは両地点が携帯電話の届く範囲であれば，隣接していなくともよい．
- 規模性
  1 つの参照点を 1 つの未知点で占有するため，規模性はない．これを共有するためには何らかの工夫が必要になる．
- 利用に際して必要なこと
  携帯電話を必要とする．通信コストを考慮する必要がある．
- 精度
  FM ラジオ隙間電波ほどでないにしろ，帯域が狭域である．差動測位には十分であるが，干渉測位には帯域が不足する場合がある．場合によっては通信路が原因で精度が劣化する恐れがある．

携帯電話を用いる方式では帯域のほぼ十分に確保できるが，最大の問題は規模性の欠如である．規模性の欠如は補正機構として最も致命的であると言える．

「第 2.2.3 節」 MCA 無線，中波ビーコン受信機を用いる方式

MCA 無線，ビーコン受信機を用いて補正情報を配送することで差動測位，干渉測位を行うことができる．この種の機構は国土地理院の国土地理院 RTK[4] や海上保安庁の海上 DGPS ビーコン [5]，運輸省の港湾建設工事 RTK[5] などが実用化されている．

- サービスエリア
  サービスエリアは FM ラジオの副搬送波に比べて狭域だが，多くの電波発信点をおくことで広い地域に対応できる．
- 規模性
  FM ラジオの副搬送波と同様，サービスエリア内の規模性は考慮する必要がない．一対多の通信が可能．利用者側でサービスの選択を行う必要がない．むしろ，到達するものは全て使用できる補正情報であるといえる．
- 利用に際して必要なこと
  専用の受信機を必要とする．また，高額な利用費を負担しなければならない場合がある．通信コストはかからない．

- 精度
  十分に干渉測位に利用できる帯域がある．よって精度は高い．

以上のように既存の補正情報配送方式は多種多用なものが存在する．既存の補正情報配送機構は専用の無線機等を利用した放送型の配送が行われているものが多い．よって，あらゆる場所でこういった利用を想定した場合，大掛かりな無線設備を用意する必要があり，その実現は容易ではない．

「第2.3節」 インターネットを用いる既存配送技術に関する調査

ここでは既存のデータ伝送技術の内，特にインターネットを用いるものについてまとめる．インターネットを用いるものも補正情報としてRTCM SC-104形式が用いられている．

「第2.3.1節」 アルプスマップ インターネット配信に関する公開実験機構

アルプスマップ(株式会社アルプス社)による機構で公開実験[6]された(実験期間は1998年12月から1999年2月)．これは差動測位の補正情報を，インターネットを通じて配信する実験である．概要を以下に示す．

- 配送データ
  RTCM SC-104形式 TYPE1．ヘッダなどの特別な操作をせずにそのまま配送した．
- 配送方法
  インターネットを通じアルプス社のサーバからHTTPでダウンロードする．(http://www.alpsmap.co.jp/cgi-bin/dgps.cgi)
- アルプス社サーバ側成
  ノヴァテル社GPS受信機からRC-232C経由でWEBサーバへ補正情報を取得する．それをCGIプログラムから配送する．
- 利用者側構成
  インターネットに接続しHTTPリクエストをアルプス社サーバに

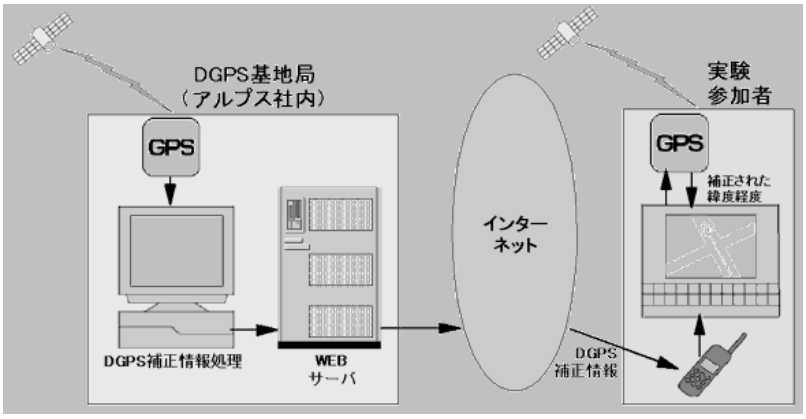


図 2.3: アルプス社のシステム

出す．補正情報を得，差動 GPS の利用できる受信機に RC-232C 経由で送る．補正後の位置情報を RC-232C 経由で得る．

なお，アルプス社サーバ側構成，利用者側の構成を図 2.3に示す．(出典はアルプス社ホームページ [6])

この機構の特徴は配送プロトコルに HTTP を用いている事である．このためヘッダ部分が実際のデータと比較して大きい．今回は参照地点，配送始点ともに 1 つずつであったが，このモデルでは利用者がどこにアクセスすると最適な補正情報を得られるか事前に知っている必要がある．これはインターネットを通信路にする補正情報配送全てに言えることだが，帯域の予約，遅延の予測は困難である．

「第 2.4 節」 既存の配送技術の評価

ここでは既存の配送機構についての定性的評価を行い問題点を指摘する．

「第 2.4.1 節」 既存の配送機構の定性的評価

既存の補正情報配送の定性的評価のための評価項目を列挙する．各通信方式に対し利用範囲，規模性の有無，利用できる補正情報の種類，機構

の利用に際し必要なもの，精度について評価を行う．それぞれの評価項目についてその意味と重要度について示す．

- 利用範囲
  利用範囲は補正情報本来の有効範囲，配送機構の有効範囲の重なる部分が利用できる範囲であるといえる．利用範囲が狭いことは衛星測位の利用範囲が地上全体であることを考慮すると致命的である．
- 規模性の有無
  規模性には，複数の利用者が使用できるかどうかという点に加え，どの程度の利用者数までを許容するかという点を評価する．規模性が全くないのは致命的である．
- 利用できる補正情報の種類
  差動測位，干渉測位が利用できるかどうかを評価する．
- 機構の利用に際し必要なもの
  通信のための機器などが必要になることがある．
- 精度
  測位精度は高ければ高いほどよい．

上記のような評価項目に基づいて各通信方式について評価をし，表 2.1 に示す．

表 2.1: 既存のシステムの特徴

通信方式	利用範囲	規模性	補正の種類	要求事項	精度
FM ラジオ副搬送波	局地的	無制限	差動測位 (不完全)	専用の受信機	低い
携帯電話	広い	なし	差動測位	携帯電話	普通
無線	局地的	無制限	差動，干渉測位	専用の受信機	高い
インターネット	広い	あり	差動測位	インターネット接続	普通

インターネットはアルプス社の機構を，無線は MCA 無線 ・中波ビーコンを指す

「第 2.4.2 節」 既存の配送機構の問題点

既存の配送機構には第 2.4.1節で全ての項目を満足させるものはなかった．

特に規模性のない携帯電話による補正情報配送機構は致命的である．また，規模性には問題がないが，十分に補正ができないFMラジオ副搬送波による補正情報配送機構は十分に機能しているとは言えない．

そういった中でMCA無線・中波ビーコンによる補正情報配送機構は規模性，精度ともに十分であるが，専用の受信機を用意する必要があり，サービスの提供者も大掛かりな設備を必要とする．

インターネットを用いるアルプス社の機構は規模性でMCA無線・中波ビーコンに劣るもののインターネットに接続するのみで利用でき，サービスの提供者も安価な設備を用意するだけで良い．しかし，インターネットを用いるサービスにはサービスエリアが限定できないことがかえって利用者に最適な補正情報をどのように与えるかという新たな問題を生んでいる．インターネットを用いる補正情報配送機構にはそのサービスの抽象化は不可欠であると言える．

まとめると，インターネットを用いて補正情報を配送する機構は有効である．その機構は既存の補正情報配送機構の問題点を考慮して設計する必要がある．特に規模性，サービスエリアの広さ，利用に必要なものについて考慮する．

第3章 インターネットを用いる補正情報配送機構の設計

本章では，インターネットを用いる補正情報配送機構のモデルを提示する．

本機構の目的

D-RTK/GNSSを用いて精度の高い測位を行う場合，移動局として以下の用件を満たすことが重要である．

- メッセージタイプの一致
  大前提として，移動局，基準局間で，測位に使用される補正情報の種類が一致しなければならない．これは最終的にGNSS受信機に対し、RTCM SC-104形式で補正情報を届ける必要があるからである．最終的にRTCM SC-104形式に変換できれば補正情報配送機構ではどのような形式で配送してもかまわない．
- 地理的な近さ
  移動局と基準局は地理的に近い方が良い．これは両局におよぼす大気中での様々な要因に大きな差がない方が良好な測位を行うに都合が良いからである．
- 遅延の少さ
  リアルタイム測位の場合には，移動局と基準局の通進路間での遅延は少なければ少ないほどよい．
- 衛星の一致
  移動局，基準局それぞれから観測される測位衛星がディファレンシャル測位では4個以上，リアルタイムキネマティック測位では8個以上同一のものが観測されなければいけない．
- 帯域の確保
  移動局，基準局間で，十分な帯域が確保されなければならない．

既存の補正情報配送機構ではこれらの用件が導入時に決定してしまうことであり，導入後の変更は容易ではない．これはこれらの用件が満たされているかどうかが，用いられる通信路によって決定していた．

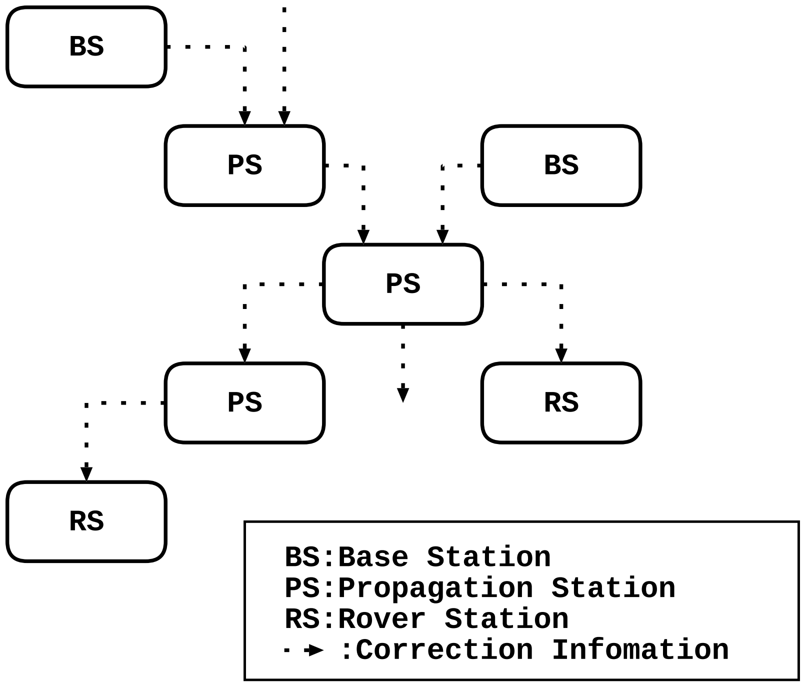


図 3.1: 補正情報配送モデル

「第3.2節」 補正情報配送モデルの提示

本機構で用いるモデルを補正情報配送モデルとし，以下に述べる要素から構成されるものとする．補正情報を生成する基準局，それを利用する移動局，さらにそれらの配送を行う配送局より構成される．モデルを図3.1に示す．これは第2.4.2節で示した既存の補正情報配送機構の問題点を考慮した結果である．

- 補正情報の表現
  補正情報を表現する．これによってどのような補正情報を取得するかを表現して要求を行える．
- 規模性を考慮する
  一極集中せず分散して補正情報を提供できるようにする．
- サービス透過性
  どこの基準局にどういった補正情報が存在するのかを考慮することなく要求を行うことができるようにする．

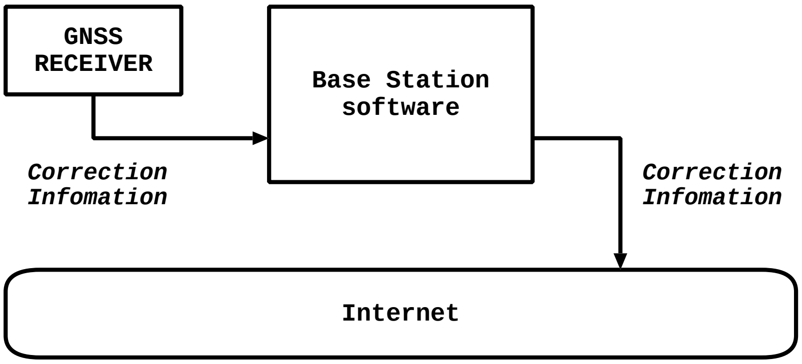


図 3.2: 基準局

上記を考慮することで図 3.1のような階層化が可能で補正情報を投下適に扱うことができるモデルを提案する．

各局の機能を以下のように定義する．さらに，補正情報の送信側を上流 (局)，受信側を下流 (局) とする．

- 基準局 (**BS: Base Station**)
  補正情報を生成し、下流局へ送信する．下流局は，移動局，配送局，またはマルチキャストグループの場合が有り得る．図 3.2を参照．
- 移動局 (**RS: Rover Station**)
  補正情報を受信し，利用する．必要な基準局の補正情報を上流局に要求する．図 3.3を参照．
- 配送局 (**PS: Propagation Station**)

  上流局からの補正情報を下流局に転送する．また，下流局の要求に応じて、どの基準局からの情報を配信するか取捨選択する．図 3.4を参照．

また，これらのうち最低限，基準局と移動局が一つずつあれば本モデルは構成できるとする．また，基準局と移動局の間にはいくつの配送局が存在しても良い．

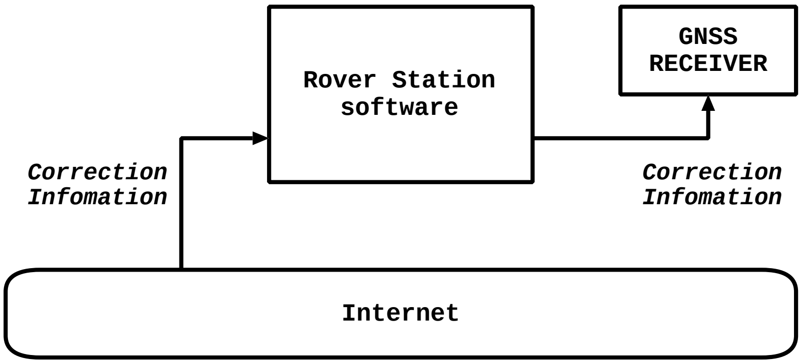


図 3.3: 移動局

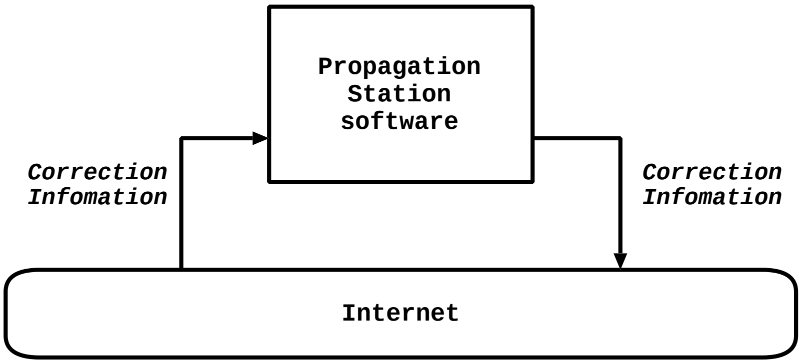


図 3.4: 配送局

「第3.3節」　本機構の概要

本機構は，D/RTK-GNSSの補正情報の配送機能を提供する．補正情報の配送を下流局の要求に基づいて行う配送局という概念を導入する．配送局の導入によって帯域の節約や補正の精度の向上などの効果が期待される．

「第3.3.1節」　各局の構成要素

本機構は補正情報配送モデルより構成される．基準局，移動局は以下から構成される．

- GNSS用アンテナ
- GNSSレシーバ
- 計算機
  以下のものを備えたものである必要がある．
  - 補正情報配送プログラム
  - ネットワークインタフェース
  - GNSS受信機インタフェース

基準局，移動局のアンテナを図3.5，図3.6に示す．

配送局は計算機のみから構成される．

「第3.3.2節」　補正情報配送要求の要素

補正情報を配送する際に様々な要素が考えられる．どのような補正情報をいつどのような送信間隔で送信するのかを表現する必要がある．本機構で抽象化するものを以下に挙げ，説明する．

- 基準局の緯度経度

  基準局の緯度経度をあらわす．緯度はdd.ddddd...，経度はddd.ddddd...である．(それぞれdoubleであらわされるため、15桁まで)

図 3.5: 基準局のアンテナ

図 3.6: 移動局のアンテナ

```
BSID on the Internet
   Area BSID + local BSID = I-BSID
```

図 3.7: インターネット基準局ID

- 基準局の識別子
  基準局を識別する必要があり、基準局ID(BSID) として第3.3.3節で後述する．
- **RTCM SC-104** にあるメッセージタイプ
  1-63 のうち複数のものである．RTCM SC-104 形式の各タイプについては付録Aを参照のこと．
- 補正情報の送信間隔
  上流局から下流局への補正情報の送信間隔をあらわす．基準局で補正情報を生成する間隔が最大値であり，単位は秒である．

「第3.3.3節」　基準局IDの割り振り方

本機構では，基準局のIDが一意に定義されている必要がある．しかし，RTCM SC-104 の規格では基準局ID(RID: Reference Station ID) が12bit しか定義できいない．これは4096局の区別しかできずインターネットのような大規模システムでは不十分である．そこで本機構では以下のように基準局IDを定義し，将来的な基準局の増加にも対応できるようにした．

まず始めに全体の基準局IDを44bit長とし，インターネット基準局ID(I-BSID: Internet Base Station ID) とする．インターネット基準局IDの前半32bit部分をエリアIDとし，後半12bit部分をローカルIDとする．図3.7を参照．

ここで，32bit長のエリアIDは後述する方法で基準局の緯度経度から一意に定まる．また，12bit長のローカルIDはそのエリア内で一意であれば良い．その定義の仕方は次に述べる．

表 3.1: RTCM SC-104 のメッセージタイプ
[RTCM SC-104 のメッセージタイプ]

TYPE NO.	STATUS	TITLE
1	Fixed	Differential GPS Corrections
2	Fixed	Delta Differential GPS Corrections
3	Fixed	GPS Reference Station Parameters
4	Tentative	Reference Station Datum
5	Fixed	GPS Constellation Health
6	Fixed	GPS Null Frame
7	Fixed	DGPS Radiobeacon Almanac
8	Tentative	Pseudolite Correction Set
9	Fixed	GPS Partial Correction Set
10	Reserved	P-Code Differential Corrections
11	Reserved	C/A-Code L1,L2 Delta Corrections
12	Reserved	Pseudolite Station Parameters
13	Tentative	Ground Transmitter Paramters
14	Tentative	GPS Time of Week
15	Tentative	Ionospheric Delay Message
16	Fixed	GPS Special Message
17	Tentative	GPS Ephemerides
18	Fixed	RTK Uncorrected Carrier Phases
19	Fixed	RTK Uncorrected Pseudoranges
20	Tentative	RTK Carrier Phase Corrections
21	Tentative	RTK/Hi-Acc. Pseudorange Corrections
22	Tentative	Extended Reference Station Parameters
23-30	–	Undefined
31	Tentative	Differential GLONASS Corrections
32	Tentative	Differential GLONASS Reference Station Parameters
33	Tentative	GLONASS Constellation Health
34	Tentative	GLONASS Partial Correction Set
35	Tentative	GLONASS Radiobeacon Almanac
36	Tentative	GLONASS Special Message
37	Tentative	GNSS System Time Offset
38-58	–	Undefined
59	Fixed	Proprietary Message
60-63	Reserved	Multipurpose Usage

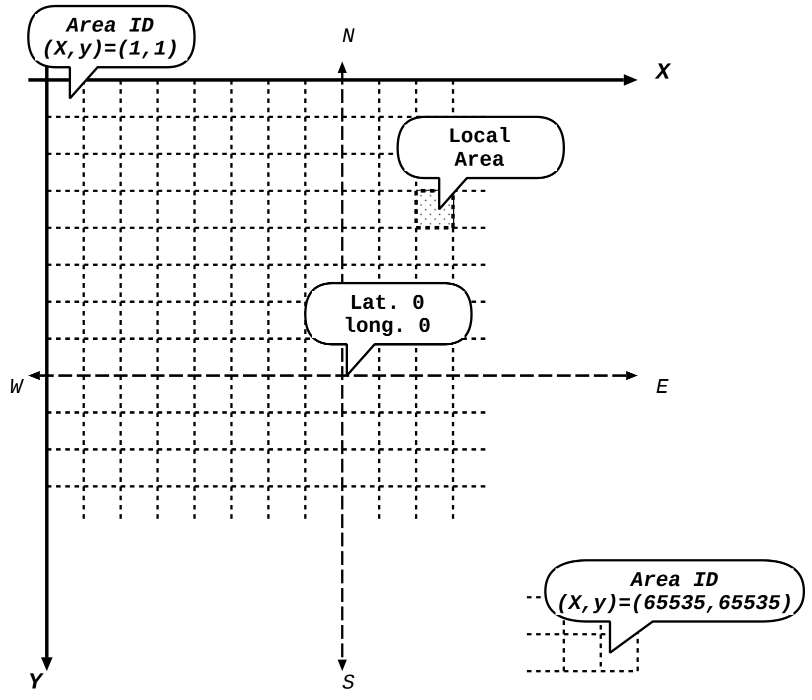


図 3.8: エリア ID

「第 3.3.3.1 節」　エリア ID の定義

図 3.8のように地球上を緯度経度に応じて分割し，それをエリアと呼ぶ．緯度経度をそれぞれに 16bit ずつで分割する．エリア ID はそれにつけられる ID である．

始点 (0,0) は北緯 90 度，西経 180 度とする．東経の正方向に向かって x 軸，南緯の正方向に向かって y軸を設定する．したがって，北緯 $long$ 度，東経 lat では次の式のようになる．ただし，南緯，西経はそれぞれ北緯，東経の負の値を取るとする．

$$(x,y) = (\left|\frac{(90-long)\times 65534}{180}\right|, \left|\frac{(lat+180)\times 65534}{360}\right|) \tag{3.1}$$

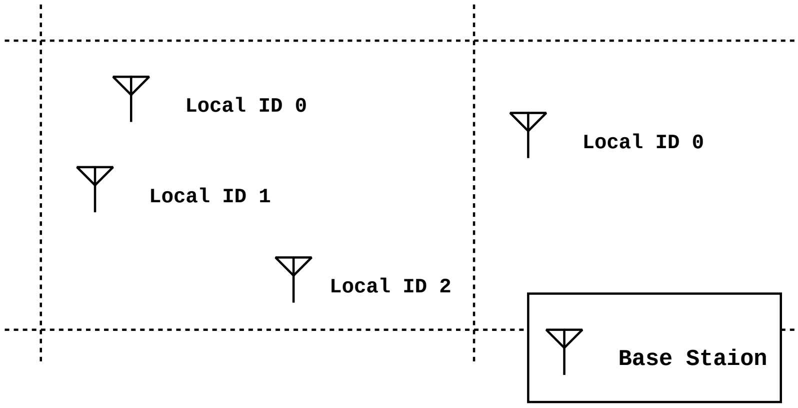


図3.9: ローカルID

「第3.3.3.2節」 ローカルIDの定義

ローカルIDは同一なエリア内でエリア内で一意である．本機構はローカルIDの割り振りの方法について議論しない．ただし、ローカルIDを前述のRTCM SC-104に定めるRIDにそのまま使用する事は可能である．図3.9のようにエリアの外で同じIDを使うことはあり得る．

「第3.4節」 プロトコル仕様

補正情報の配送のためにプロトコル仕様を定義する必要がある．基準局-移動局、基準局-配送局、配送局-移動局、配送局-配送局すべて局の間で同じプロトコルを使用できるようにする．ここでは、補正情報配送制御プロトコル、補正情報配送プロトコルを定義する．

ここで、補正情報を配送する際に使用するプロトコルを補正情報配送プロトコルと呼び、補正情報を配送する際に必要な情報を配送するプロトコルを補正情報配送制御プロトコルと呼ぶ．

また、補正情報配送プロトコルによって上流局から下流局に配送される情報を補正情報と呼び、補正情報配送制御プロトコルによって同様に配送される情報を基準局情報と呼ぶ．基準局情報に基づいて補正情報要求が下流局から上流局に行われ、補正情報が配送される．

以下、第3.4.1節 補正情報配送制御プロトコル、第3.4.2節 補正情報配送プロトコルについて記述する．

「第 3.4.1 節」　補正情報配送制御プロトコル

補正情報の配送のためには上流局が配送可能な補正情報をその下流局で認知する必要がある．上流局から配送可能な補正情報はそれを生成する基準局の補正情報による．そこで、補正情報の性質をあらわす情報を抽象化して基準局情報とする．

補正情報配送制御プロトコルは Sender-Receiver 間のコントロールプロトコルである．必要な基準局情報を上流局に対し要求し、下流局に対し応答するのが補正情報配送制御プロトコルの役割である．要求、応答として以下のメッセージを定義する．

- 基準局情報要求
  補正情報の受信側が基準局情報の要求を行う．
- 基準局情報応答
  補正情報の送信側が配送可能な補正情報を生成した基準局についての情報の集合を応答する．

「第 3.4.1.1 節」　基準局情報

ここで、基準局情報について定義する．基準局情報は基準局に関する情報と基準局が生成する補正情報に関する情報から成る．各局では補正情報配送の要求のために上流に存在する基準局について考慮した上で、補正情報要求をすることになる．

次に、各局で保持する基準局情報について触れる．基準局が保持する基準局は自局のものだけになる．また、配送局が保持する基準局情報は上流局の基準局情報の集合にならなければいけない．したがって上流に複数の基準局が存在する場合、図 3.10のように、自局の上流に位置するすべての基地局情報の集合を保持しなければいけない．

基準局情報には以下のものが必要である．

- **BSID**
  BSID のすべてが必要だがヘッダに含まれるのは緯度、経度の情報のみで良い．RID は RTCM SC-104 の中に含まれている．
- 修正 **Z** カウントの差分
  修正 Z カウントとはデータ測定時刻を毎時の最初の時間からの経過

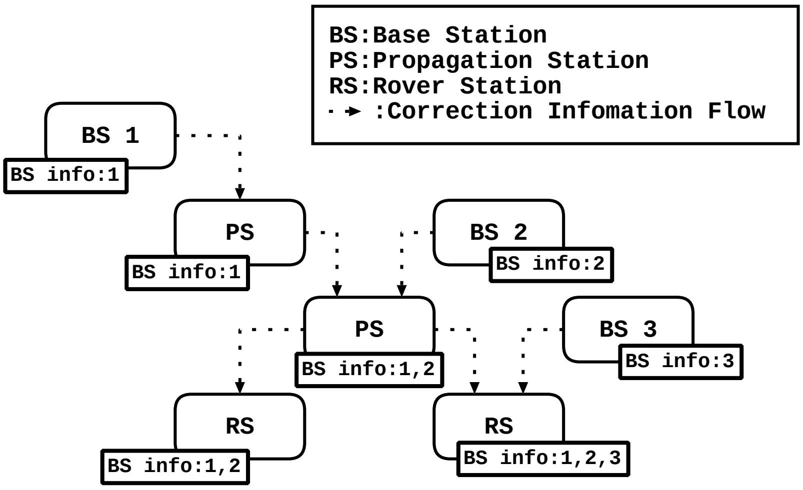


図 3.10: 基準局情報の流れ

秒数を 0.6 秒単位であらわす．RTCM SC-104 では 13bit が当てられ、0-5999 の値をとる．修正 Z カウントは RTCM SC-104 の中に含まれている．

- **RTCM SC-104 メッセージタイプ**
  該当する基準局の生成する補正情報の RTCM SC-104 形式におけるメッセージタイプ．1-64 の内、複数を指すことがある．

「第 3.4.1.2 節」　基準局情報要求メッセージ

補正情報の受信側は基準局に関する情報を必要としている．基準局情報を要求するメッセージを基準局情報要求メッセージと呼ぶ．パケットフォーマットを図 3.11に示す．

本メッセージは 3 つのフィールドから構成され、固定長 (32bit 長) である．

各フィールドの意味を示す．

- **MESSAGE TYPE(8bit 長)**
  基準局情報要求メッセージをあらわす値 0 が設定される．

- **NUMBER OF REQUESTED BASE STATION LISTS(8bit 長)**

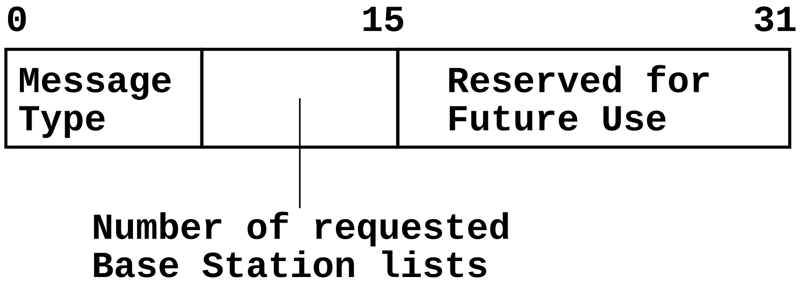


図 3.11: 基準局情報要求メッセージ

幾つの基準局情報を要求するかをあらわすフィールド．基準局情報応答メッセージに含まれる基準局情報の数をあらわす．1-255 の基準局を表現できる．0 は全てという意味である．

- **予約フィールド (16bit 長)**
  上記 16bit の後に 16bit の予約フィールドを持つ．すべての bit に 0 が設定されている．

「第 3.4.1.3 節」 基準局情報応答メッセージ

補正情報の送信側は基準局に関する情報を保持している．第 3.4.1.2節に対する応答として基準局情報を受信側に送信するメッセージを基準局情報応答メッセージと呼ぶ．パケットフォーマットを図 3.12に示す．

本メッセージはヘッダ部分 (32bit 長) と複数の基準局情報 (可変長) によって構成され、可変長となる．

各フィールドの意味を示す．

- **MESSAGE TYPE(8bit 長)**
  基準局情報応答メッセージをあらわす値 1 が設定される．

- **予約フィールド (8bit 長)**
  上記 8bit の後に 8bit の予約フィールドを持つ．これは将来 RTCM SC-104 のタイプ数が増加した場合に使用される可能性がある．現在はすべての bit に 0 が設定されている．

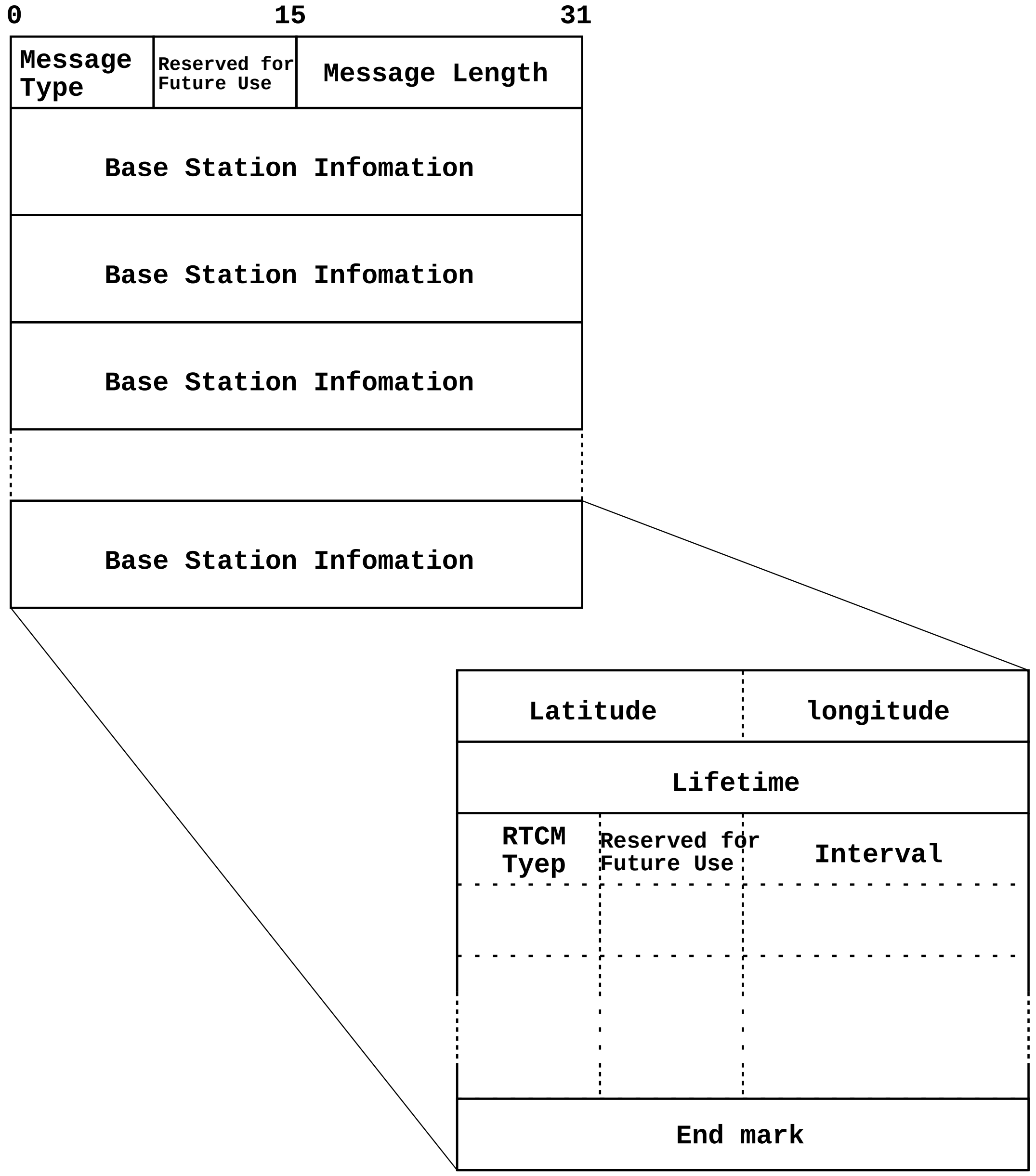


図 3.12: 基準局情報応答メッセージ

- **MESSAGE LENGTH(16bit 長)**
  メッセージ全長の byte 数が設定される．

- **基準局フィールド (可変長)**
  64bit 長の固定部と 32 $\times N$ bit 長の可変部からなる．このフィールドは複数存在する可能性がある．

基準局フィールドの内容について示す．基準局フィールド 1 つで 1 つの基準局に関する情報を保持している．

- **緯度フィールド (16bit 長)**
  緯度が第 3.3.3.1節に示した符合なし整数で設定される．

- **経度フィールド (16bit 長)**
  経度が第 3.3.3.1節に示した符合なし整数で設定される．

- **ライフタイムフィールド (32bit 長)**
  基準局の補正情報が上流局から配送可能な時間が設定される．単位は秒である．

- **各 RTCM SC-104 メッセージ領域 (32bit 長)**
  配送可能な RTCM SC-104 メッセージのタイプと各 RTCM SC-104 メッセージごとの配送間隔をあらわす．

  - **RTCM TYPE(8bit 長)**
    配送可能な RTCM SC-104 メッセージのタイプが設定される．
  - **予約フィールド (8bit 長)**
    上記 16bit の後に 8bit の予約フィールドを持つ．これは将来 RTCM SC-104 のタイプ数が増加した場合に使用される可能性がある．現在はすべての bit に 0 が設定されている．
  - **INTERVAL(16bit 長)**
    RTCM SC-104 メッセージの配送間隔が設定される．単位は秒である．
  - **END MARK(32bit 長)**
    なお、各基準局フィールドの終端の 32bit にはすべて 0 を挿入し、終端とする．

「第 3.4.2 節」　補正情報配送プロトコル

補正情報を受信する局は補正情報配送制御プロトコルによって上流局がどのような補正情報を配送可能か、あらかじめ認知することができる．これによって移動局が補正に最適な補正情報を得ることが可能になる．

補整情報配送プロトコルは、補正情報の配送のためのプロトコルである．必要な補正情報上流局に対し要求し、下流局に応答するのが補正情報配送プロトコルの役割である要求、応答として以下のメッセージを定義する．

- 補正情報要求
  下流局が上流局に基準局情報に基づき、補正情報の要求を行う．
- 補正情報応答
  上流局が下流局に補正情報を応答する．

「第 3.4.2.1 節」　補正情報要求メッセージ

補正情報の送信側は配送する補正情報に関する情報を必要としている．補正情報を要求するメッセージを補正情報要求メッセージと呼ぶ．パケットフォーマットを図 3.13に示す．

本メッセージは固定長部 (96bit 長) と複数の RTCM SC-104 に関するフィールド (可変長) によって構成され、可変長となる．

各フィールドの意味を示す．

- **MESSAGE TYPE フィールド (8bit 長)**
  補正情報要求メッセージをあらわす値 2 が設定される．
- **NUMBER OF FIELDS OF RTCM TYPE REQUESTED フィールド (8bit 長)**
  補正情報要求メッセージをに含まれる RTCM SC-104 に関するフィールドの数をあらわす．
- **予約フィールド (16bit 長)**
  上記 16bit の後に 16bit の予約フィールドを持つ．現在はすべての bit に 0 が設定されている．
- **緯度フィールド (16bit 長)**
  緯度が第 3.3.3.1節に示した符合なし整数で設定される．

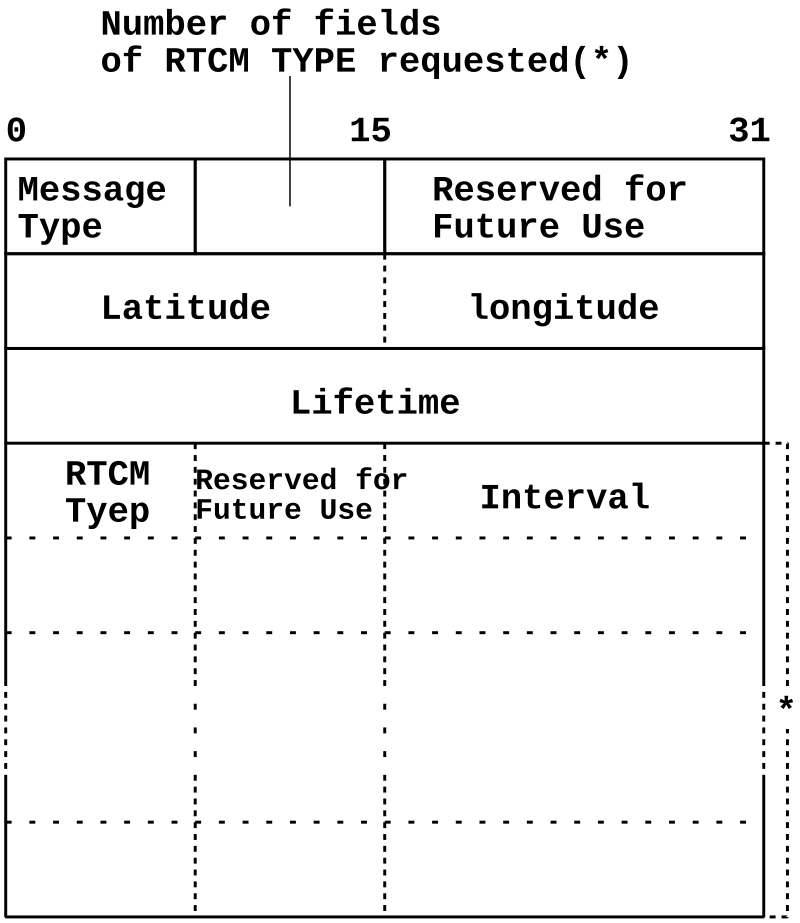


図 3.13: 補正情報要求メッセージ

- **経度フィールド (16bit 長)**
  経度が第3.3.3.1節に示した符合なし整数で設定される．

- **ライフタイムフィールド (32bit 長)**
  基準局の補正情報が上流局から配送可能な残り時間が設定される．単位は秒である．

- **RTCM SC-104 メッセージフィールド (32bit 長)**
  配送可能なRTCM SC-104メッセージのタイプと各RTCM SC-104メッセージごとの配送間隔をあらわす．このフィールドは複数存在する可能性がある．

  各RTCM SC-104メッセージフィールドの内容について解説する．

  - **RTCM TYPE(8bit 長)**
    配送可能なRTCM SC-104メッセージのタイプが設定される．
  - **予約フィールド (8bit 長)**
    上記16bitの後に8bitの予約フィールドを持つ．これは将来RTCM SC-104のタイプ数が増加した場合に使用される可能性がある．現在はすべてのbitに0が設定されている．
  - **INTERVAL(16bit 長)**
    RTCM SC-104メッセージの配送間隔が設定される．単位は秒である．

「第3.4.2.2節」　補正情報応答メッセージ

補正情報の送信側は補正情報を保持している．第3.4.2.1節の補正情報要求メッセージに対する応答として、補正情報情報を受信側に送信するメッセージを補正情報情報応答メッセージと呼ぶ．パケットフォーマットを図3.14に示す．

本メッセージは固定長部 (64bit 長) と可変長部 ($32\times N$bit 長) によって構成され、可変長となる．

各フィールドの意味を示す．

- **MESSAGE LENGTH(16bit 長)**
  メッセージ全長のbyte数が設定される．

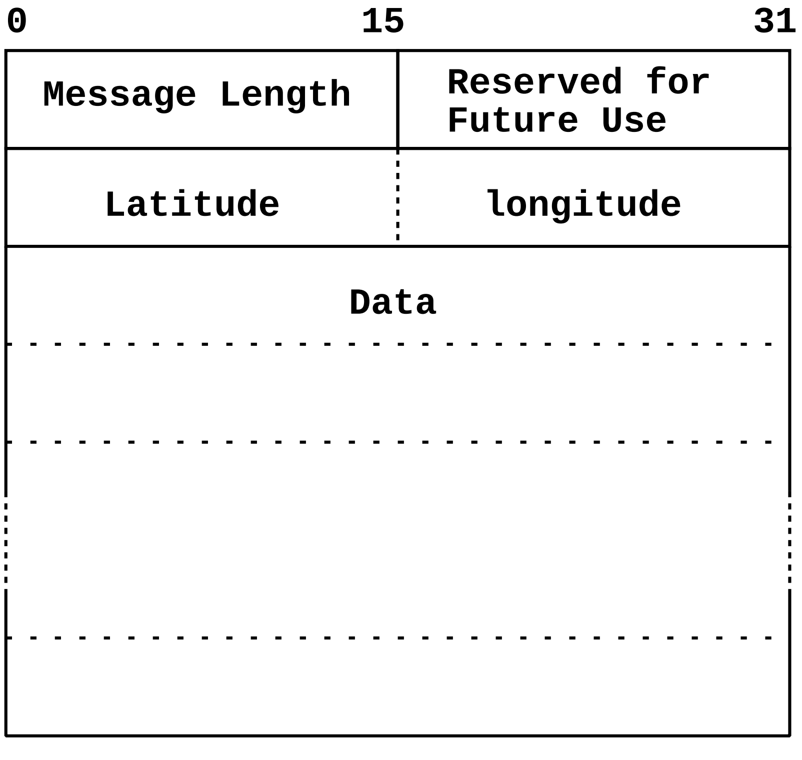


図 3.14: 補正情報応答メッセージ

- **予約フィールド (16bit 長)**
  上記 16bit の後に 16bit の予約フィールドを持つ．現在はすべての bit に 0 が設定されている．

- **緯度フィールド (16bit 長)**
  緯度が第 3.3.3.1節に示した符合なし整数で設定される．

- **経度フィールド (16bit 長)**
  経度が第 3.3.3.1節に示した符合なし整数で設定される．

- **データフィールド (可変長)**
  RTCM SC-104 を加工したデータが挿入される．この場合，デコードとパリティの削除を行う．デコードはスタートビットの削除，並びビット反転，値ビット反転を行う．このデコードについては付録 A.4.1を参照

第4章　インターネットを用いる補正情報配送機構の実装

本章では，前章にもとづき，インターネットを用いる補正情報配送機構の実装を行う．本実装を「Scrache」と呼ぶ．

「第 4.1 節」 Scrache の実装アプローチ

ここでは Scrache についての概要を述べる．実装環境、目的、内部の処理の設計について述べる．

「第 4.1.1 節」 Scrache の実装環境

Scrache は FreeBSD3.3 RELEASE 上で C 言語を用いて実装した．その動作には FreeBSD2.2.8 RELEASE 以上の POSIX Thread(以下 pthread) が必要となる．よって、C 言語のコンパイル環境と pthread が実装されている OS への移植は容易である．なお、GCC のバージョンは gcc-2.95.1 である．

「第 4.1.2 節」 Scrache の目的

Scrache はスレッドを用いた複数のモジュールによって構成される．基準局、配送局、移動局として動作する際にこれらのモジュールの必要なものを作動させることで一つのソフトウェアですべてに対応できる．

「第 4.1.3 節」 Posix Thread

モジュール的な実装のために複数のプロセスを使う手法とスレッドを使う手法とを比較検討した．その結果、スレッドには幾つかの利点があることが分かりスレッドを使用した．

Scrache ではモジュールの数が多く、さらに動的にモジュールを起動させることが多いため資源を有効に活用できるスレッドは有利である．また、スレッドが動的に終了することもあり、この場合各スレッドをデタッチすることにより資源を自動的に解放できる．

pthread は POSIX 準拠のスレッドであり、pthread ライブラリを用いることで移植性の高いソフトウェアとして実装できる．

```
pthread_mutex_lock(&gi.rs_mutex); /* lock here */
for (rs = gi.rs; rs != NULL; rs = rs->next) {
  pthread_create(&rs->myself, NULL,
                 (void *)ds_gnss, (void *)rs);
  pthread_detach(rs->myself); /* datach here */
  gi.whoami |= IAM_RS;
}
pthread_mutex_unlock(&gi.rs_mutex); /* unlock here */
```

図 4.1: mutexのロック，アンロック

「第4.1.4節」 Scracheの処理の流れ

Scracheの処理の流れを記す(図4.2関数名などは簡略化したもの)を参照．Scracheは起動する際に設定ファイルを読み、基準局、配送局、移動局のいずれかとして動作するべきかを判断し、必要な処理を行う．

はじめに*init_conf()*で設定ファイルを読み込み、プログラムのパラメタを決定する．このパラメタについては4.2.2を参照．次に*pthread_mutex_init()*で各スレッドの排他制御のために*mutex*を初期化する．さらに*inputq_init()*でinputqのスレッドの排他制御のために*mutex*を初期化する．

その後必要なモジュールを起動する．プログラムのパラメタの下流局の数だけ*(void *)ds_protocol*の下流プロトコルモジュールスレッドを起動する．

プログラムのパラメタのRSが真になっている場合は*(void *)ds_gnss*の下流GNSSモジュールスレッドを起動する．

プログラムのパラメタのPSまたはRSが真になっている場合は*(void *)us_protocol*の上流プロトコルモジュールスレッドを起動する．

プログラムのパラメタのBSが真になっている場合は*bs_init();*でBSLの初期化を行う．さらに*(void *)us_gnss*の上流GNSSモジュールスレッドを起動する．

その後全ての局で使用される3つのモジュールを起動する．

*(void *)main_control*のメインコントロールモジュールスレッドを起動し、*(void *)main_message*のメインメッセージモジュールスレッドを起動し、*(void *)periodic_watchdog*のピリオディックモジュールスレッドを起動する．

以上のようにScracheの初期化は完了する．以後は各スレッドが協調して動作する．各スレッドの動作については第4.2.3節から第4.2.11節を

参照．

初期化後、必要に応じて新たにモジュールが起動される場合がある．第4.2.11節にある下流プロトコルモジュールは新たな補正情報の要求先に対し、1 つのモジュールが起動される．要求先からの要求が断続的に到達している間は下流プロトコルモジュールは動き続けるが、要求が跡絶え一定時間 (暫定的に 30 秒としている) が経過し、モジュールが不要になった際はそのモジュールは終了する．

「第 4.1.5 節」 Scrache の排他制御

Scrache では下記のように排他制御の必要な部分が存在する．これらの中には複数のモジュール間で共有されるものが存在する．

- 全てのモジュール
  各スレッドに関して *mutex* による排他制御を行う．
- 局情報リスト
  第4.2.1節で後述するリストに関して *mutex* による排他制御を行う．
- キュー
  第4.2.1節で後述するキューに関して *mutex* による排他制御を行う．また、複数の outputq と呼ばれるキューに関してはさらにリスト管理されている．そのリスト全体に関しても *mutex* による排他制御を行う．

共有される inputq を例に *mutex* によるキューの排他制御を図 4.3に示す．inputq を操作するのは、初期化の *q_init()* 関数と inputq にエンキューする上流 GNSS モジュールの *_rtcm_decode()* 関数、上流プロトコルモジュールの *usprotocol_main()* とデキューするメインメッセージモジュールの *main_message()* 関数内である．

```
main() {
  init_conf(); /* 設定ファイルの初期化 */
  pthread_mutex_init(); /* スレッド用のmutexの初期化 */
  inputq_init(); /* inputqの初期化 */

  while (DSL == TURE) { /* 下流プロトコルモジュール */
    pthread_create (&usds->myself, NULL,
                    (void *)ds_protocol, (void *)usds);
    pthread_detach (usds->myself);
  }

  if(RS) { /* RSだったら */
    pthread_create(&rs->myself, NULL,
                   (void *)ds_gnss, (void *)rs);
    pthread_detach(rs->myself);
  }

  if (PS || RS) { /* PSだったら */
    pthread_create(&gi.usprotocol_p, NULL,
                   (void *)us_protocol, (void *)NULL);
    pthread_detach(gi.usprotocol_p);
  }

  if(BS) { /* BSだったら */
    bs_init(); /* 基準局情報のリストを初期化 */
    pthread_create(&bs->myself, NULL,
                   (void *)us_gnss, (void *)bs);
    pthread_detach(bs->myself);
  }

  /* メインコントロールモジュールを初期化 */
  pthread_create(&gi.control_p, NULL,
                 (void *)main_control, NULL);
  pthread_detach(gi.control_p);
  /* メインメッセージモジュールを初期化 */
  pthread_create(&gi.mainmsg_p, NULL,
                 (void *)main_message, NULL);
  pthread_join(gi.mainmsg_p, NULL);
  /* ピリオディックモジュールを初期化 */
  pthread_create(&gi.periodic_p, NULL,
                 (void *)periodic_watchdog, NULL);
  pthread_join (gi.periodic_p, NULL);
}
```

図 4.2: プログラムの流れ

```
main() {
      ...
      /* init */
      q_init (gi.inputq, &gi.inputq_mutex);
      ...
}
_rtcm_decode() {
      ...
      if (q_enqueue(gi.inputq, inputmsg,
          &gi.inputq_mutex) < 0) {
        free (inputmsg_buf);
      }
      ...
}
usprotocol_main() {
      ...
      q_enqueue(gi.inputq, inputmsg, &gi.inputq_mutex);
      ...
}

main_message() {
      ...
      /* Dequeue from inputq */
      if ((inputmsg =
           q_dequeue((struct msg_queue *)gi.inputq,
           inputmsg, &gi.inputq_mutex)) == NULL) {
         pthread_mutex_unlock(&gi.inputq_mutex);
      }
      ...
}
```

図 4.3: 排他制御の例:inputq の共有

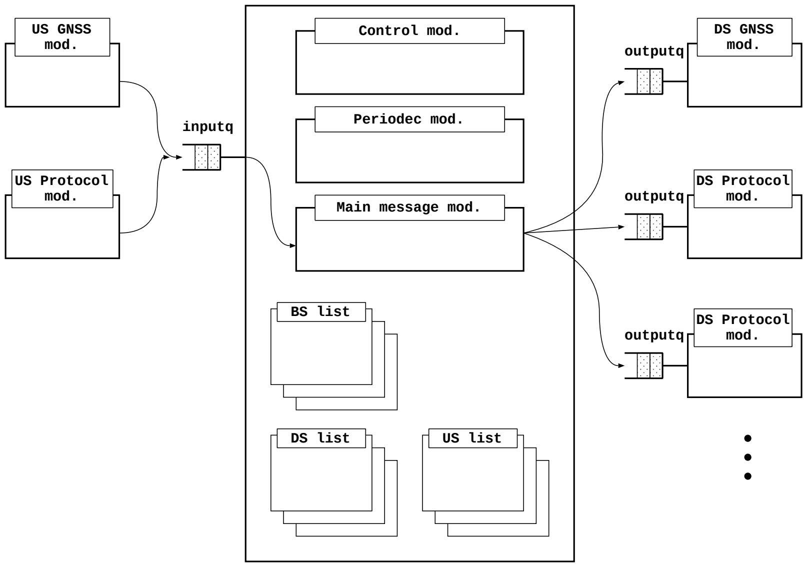


図 4.4: モジュール構成

「第 4.2 節」 Scrache の実装

はじめに Scrache のモジュール構成について述べ、次に Scrache に使用されるデータ構造について述べる．次に各モジュールの実装について述べる．

「第 4.2.1 節」 モジュール構成

Scrache は図 4.4のようなモジュールから構成される．中心の 3 つのモジュールは基準局、配送局、移動局に共通のモジュールである．また、各局で共通に管理されるリストとして基準局情報のリスト (BSL)、隣接する上流局に関するリスト (USL)、下流局に関するリスト (DSL) が存在する．これらをまとめてメイン部と呼び、図 4.4では中心の四角で示されている．

また、図 4.4のメイン部の左側に示されているのは上流側の処理をするモジュール群である．これには 2 種類が存在し、それぞれ上流 GNSS モジュール (UGM:Upstream GNSS Module)、上流プロトコルモジュール (UPM:Upstream Protocol Module) と呼ぶ．

同様に、図 4.4のメイン部の右側に示されているのは下流側の処理す

るモジュール群であり、同様に 2 種類が存在し、それぞれ下流 GNSS モジュール (DGM:Downstream GNSS Module)、下流プロトコルモジュール (DPM:Downstream Protocol Module) と呼ぶ．

上流側モジュール群とメイン部、下流側モジュールとメイン部の間にはキューが存在しそれぞれ inputq、outputq と呼び、情報のやり取りに使用される．また、inputq は 1 つ、outputq は下流側モジュールの数だけ存在する．

Scrache が基準局、配送局、移動局として動作するに動作するモジュールの関係を図 4.5、図 4.6、図 4.6に示す．

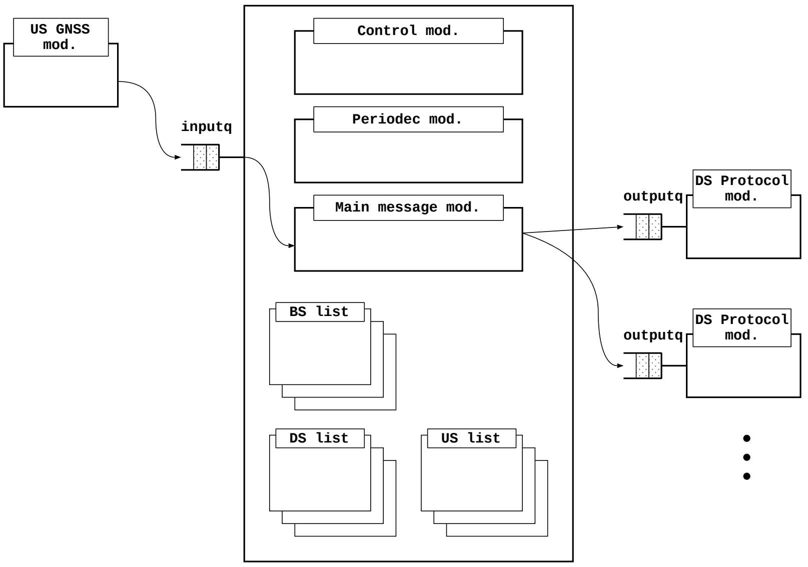


図 4.5: 基準局のモジュール

基準局では、図 4.5のように、上流 GNSS モジュールと下流プロトコルモジュールが動作している．下流プロトコルモジュールは隣接する下流ホストの数だけ動作する必要がある．

また、下流局からのリクエストに応じて配送相手局が増える場合、動的に下流プロトコルモジュールが追加、削除されることがある．

配送局では、図 4.6のように、上流プロトコルモジュールと下流プロトコルモジュールが動作している．上流プロトコルモジュールは上流は 1 つ、下流は要求のあった隣接ホストの数だけ動作する必要がある．上流プロトコルモジュールは複数の上流局に対する処理を 1 つで行う．

また、下流局からのリクエストに応じて配送相手局が増える場合、動

的に下流プロトコルモジュールが追加、削除されることがある．しかし、上流局から補正情報が配送されてくることにより新たなモジュールが追加されることはない． 上流プロトコルモジュールは複数の上流局に対する処理を1つで行う．

移動局では、図4.7のように、上流プロトコルモジュールと下流GNSSモジュールが動作している．上流プロトコルモジュール、下流GNSSモジュールはともに1つずつ動作している．

また、上流局から補正情報が配送されてくることにより新たなモジュールが追加されることはない．

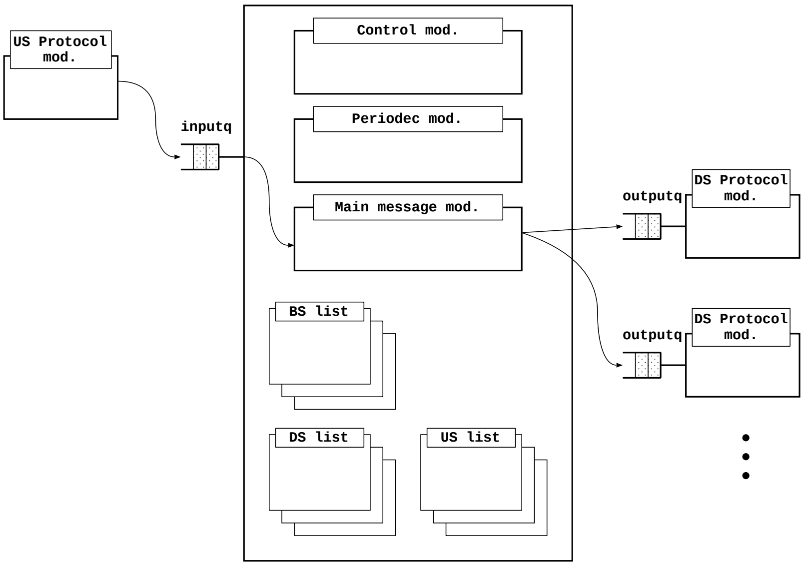


図 4.6: 配送局のモジュール

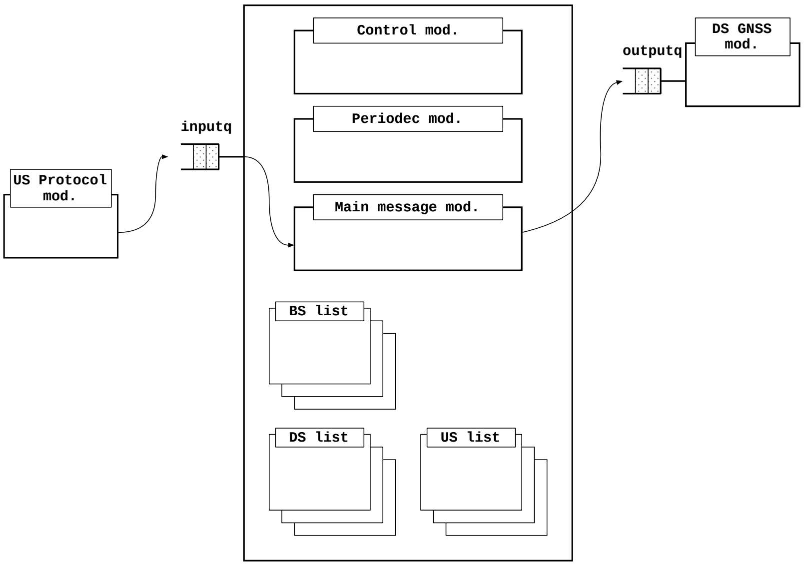


図 4.7: 移動局のモジュール

「第4.2.2節」 データ構造

ここではScracheの内部で使用されるデータ構造について示す．Scrache内部では主に4種類のデータ構造を持つ．プログラム全体のパラメタに関するデータ構造、キューに関するデータ構造、基準局・隣接局情報のリストに関するデータ構造、メッセージに関するデータ構造である．

プログラム全体のパラメタに関するデータ構造は図4.8のように構成され、gnss_info構造体と呼ばれる．この構造体に格納されるものは大きく分けて基準局・隣接局情報に関する情報、設定ファイルから読み込まれ任意の局として動作するさいのその振る舞いに関する情報、キューに関する情報、*mutex*に関する情報、スレッドに関する情報である．

```
struct gnss_info {
  struct usds_list *us,     /* 上流隣接局情報 */
                   *ds;     /* 下流隣接局情報 */
  struct bs_list   *bsl;    /* 基準局情報 */
  struct bs_info   *bs;     /* この局が基準局の場合の情報 */
  struct rs_info   *rs;     /* この局が移動局の場合の情報 */
  struct sockaddr_in sai;   /* sockaddr_in構造体 */
  int us_sockfd; /* 上流局のソケットファイルディスクリプタ */
  int  bslist_num; /* 基準局情報数 */
  /* キューに関して */
  struct msg_queue *inputq; /* inputq */
  pthread_mutex_t inputq_mutex; /* inputqのmutex */
  int  outputq_num;         /* outputqの数 */
  /* Mutexに関して */
  pthread_mutex_t us_mutex; /* us mutex lock */
  pthread_mutex_t bsl_mutex; /* bsl mutex lock */
  pthread_mutex_t bs_mutex; /* bs mutex lock */
  pthread_mutex_t rs_mutex; /* rs mutex lock */
  /* スレッドに関して */
  pthread_t control_p;      /* コントロールスレッド */
  pthread_t periodic_p;     /* ピリオディックスレッド */
  pthread_t mainmsg_p;      /* メインメッセージスレッド */
  pthread_t usprotocol_p;   /* 上流プロトコルモジュール（1つ）*/
} gi;
```

図4.8: プログラムの共通データ構造

次に基準局 隣接局に関するデータ構造に触れる．このbs_info構造体、rs_info構造体自体はgnss_info構造体に含まれる．これらの構造体には基準局、移動局として動作する場合のシリアルデバイス名やそのbaudrate、トランスポートプロトコルやポート番号、基準局ID、RTCM SC-104の

タイプ、シリアルデバイスをオープンした際のファイルディスクリプタ、GNSSモジュールのスレッドが格納される．

```
struct bs_info {
  char devname[32];
  int  baudrate;
  int  protocol;
  int  prot;
  struct bsid bsid;
  u_char type[MSGTYPES];
  FILE *bsfd;
  pthread_t myself;
  struct bs_info *next;
};
```

図 4.9: 基準局用のデータ構造

```
struct rs_info {
  char devname[32]; /* serial device name */
  int  baudrate;
  int  protocol;
  int  prot;
  u_char type[MSGTYPES];
  FILE *rsfd; /* serial file descriptor */
  pthread_t myself; /* thread value */
  struct msg_queue *outputq; /* my output que */
  pthread_mutex_t outputq_mutex;
  struct rs_info *next;
};
```

図 4.10: 移動局用のデータ構造

キューに関するデータ構造は図4.11のように構成され、msg構造体と呼ばれる．この構造体はキューへエンキューするまたデキューする単位であり、補正情報を表現するのに用いられる．その補正情報を生成した基準局ID、RTCM SC-104メッセージタイプ、メッセージ長、次のmsg構造体へのポインタである．

また、基準局IDをあらわすbsid構造体は図4.11のように構成されるbsid構造体には緯度、経度、RIDが格納されている．

基地局情報・隣接局情報のリストに関するデータ構造はそれぞれ、図4.12，図4.13のように構成され、基準局情報にがbs_info構造体、隣接局

```
struct bsid {
  u_short lat, lng;
  u_short rid;
};

struct msg {
  struct msg    *next;
  struct bsid   bsid; /* 基準局 ID */
  u_char type;  /* RTCM SC-104 メッセージタイプ */
  int  msglen;
  u_char *msg;
};
```

図 4.11: bsid 構造体

情報のリストがusds_list構造体である．bs_info構造体には基準局情報を格納する．具体的には緯度経度、補正可能な残り時間、RTCM SC-104のメッセージタイプ・配送間隔が格納される．

```
struct base_info {
  u_short lat;
  u_short lng;
  u_long  lifetime;
  struct rmsg rmsg[MSGTYPES]; /* RTCM SC-104 のメッセージタイプ・
配送間隔 */
};

struct rmsg { /* RTCM SC-104 のメッセージタイプ・配送間隔 */
  char rtype;     /* 8bit RTCM SC-104 TYPE */
  char pad;       /* 8bit reserved */
  short intvl;    /* 16bit interval */
};
```

図 4.12: 基準局情報の構造体

また、usds_list構造体には隣接局の情報が格納される．上流、下流を両方とも表現するように実装したので、全ての情報を同時に使っていない．上流なのか下流なのかをあらわすフラグ、ホスト名や、sockaddr_in構造体、補正配送の有効時間、リクエスト再送に使用されるタイマ、基準局ID、RTCM SC-104のメッセージタイプ、スレッド、次のリストへの

ポインタが格納される．

```
struct usds_list {
  int usds;
  char hostname[32];
  struct sockaddr_in sa;
  u_int period;
  u_int timer;
  struct bsid bsid;
  u_char type[MSGTYPES];
  pthread_t myself;
  struct usds_list *next;

  char local;

  struct msg_queue *outputq; /* 使用するoutputq（下流局のみ）*/
  pthread_mutex_t outputq_mutex; /* outputqのmutex lock */
  pthread_cond_t  ds_breakmyself;
  int ds_breaktrigger;
  pthread_mutex_t ds_breaktrigger_mutex;

  int ds_sockfd; /* socket file descriptor for down stream */
};
```

図 4.13: 局リストの構造体

また、下流隣接局のために outputq のポインタや outputq の mutex、スレッドの状態変数スレッドの mutex、ファイルディスクリプタが格納される．

メッセージに関するデータ構造 4 つ存在する．これは補正情報配送プロトコル、補正情報配送制御プロトコルの合計 4 つのメッセージに対応している．

補正情報配送プロトコルのメッセージのデータ構造について述べる．はじめに、補正情報配送プロトコルの補正情報要求メッセージの構造体、request_msg 構造体を図 4.14に示す．格納されるものは、前述した補正情報要求メッセージを参照のこと．

次に、補正情報配送プロトコルの補正情報応答メッセージの構造体、data_msg 構造体を図 4.15に示す．格納されるものは、前述した補正情報応答メッセージを参照のこと．

補正情報配送制御プロトコルのメッセージのデータ構造について述べる．はじめに、補正情報配送制御プロトコルの基準局情報要求メッセージ

```
struct request_msg {
  char msg_type;  /* 8bit message type */
  char rnum;      /* 8bit number of request */
  char pad[2];    /* 16bit reserved */
  u_short lat;    /* 16bit latitude */
  u_short lon;    /* 16bit longitude */
  u_int ltime;    /* 32bit lifetime */
  char *data;     /* data area */
};
```

図 4.14: 補正情報要求メッセージの構造体

```
struct data_msg {
  u_short msg_len;
  char pad[2];
  u_short lat;
  u_short lng;
  char *data;
};
```

図 4.15: 補正情報応答メッセージの構造体

```
struct bsrequest_msg {
  char msg_type;
  char rnum;
  char pad[2];
};
```

図 4.16: 基準局情報要求メッセージの構造体

```
struct bsreply_msg {
  char msg_type;
  char pad;
  u_short msglen;
  char *data;
};
```

図 4.17: 基準局情報応答メッセージの構造体

の構造体、bsrequest_msg 構造体を図 4.16に示す．格納されるものは、前述した基準局情報要求メッセージを参照のこと．

次に、補正情報配送制御プロトコルの基準局情報応答メッセージの構造体、bsreply_msg 構造体を図 4.17に示す．格納されるものは、前述した基準局情報応答メッセージを参照のこと．

「第 4.2.3 節」　メインコントロールモジュール

コントロールモジュールは全ての局で動作する．主な役割は以下のようになる．

1. 基準局リストの管理を行う．
2. 定期的に上流局へ基準局情報を要求する．
3. 下流局からの基準局情報要求に応答する．

「第 4.2.4 節」　ピリオディックモジュール

ピリオディックモジュールは全ての局で動作する．主な役割は以下のようになる．

1. 基準局リストを調べライフタイムが満了していたら破棄する．
2. 下流局リストを調べライフタイムが満了していたら破棄する．
3. 上流局リストを調べライフタイムが満了していたら、再び補正情報の要求を行う．
4. 全てのリストのエイジングを行う．

「第 4.2.5 節」　メインメッセージモジュール

メインメッセージモジュールは全ての局で動作する．主な役割は以下のようになる．

1. 全ての outputq を管理、把握する．
2. inputq からデキューする．
3. 全ての outputq にエンキューする．

「第 4.2.6 節」　上流 GNSS モジュールの実装

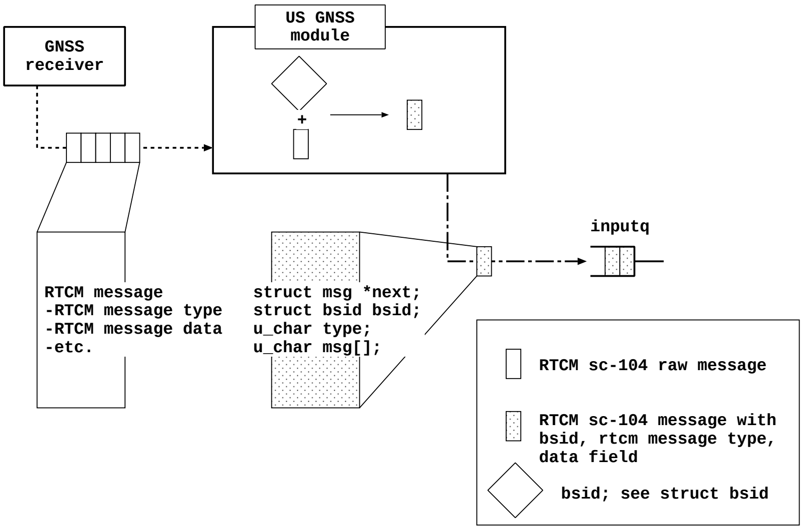


図 4.18: 上流 GNSS モジュールの詳細

上流 GNSS モジュールは基準局に用いられる．主な役割は以下のようになる．

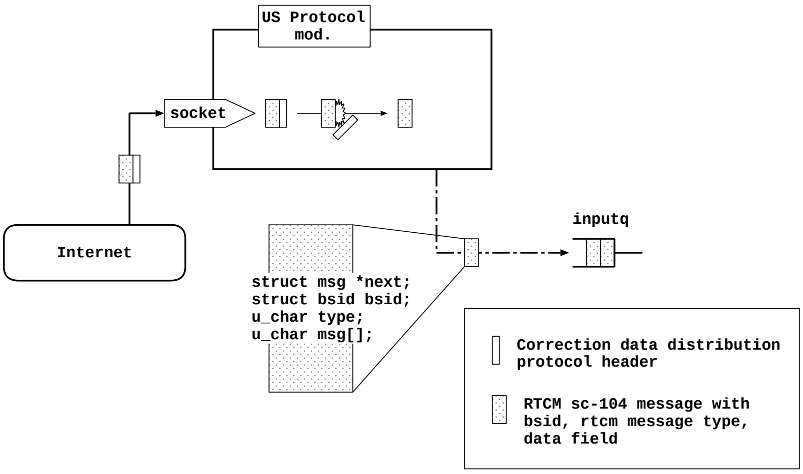
1. GNSS 受信機からシリアルデバイス経由で補正情報を読み込む．その補正情報は RTCM SC-104 形式である．
2. RTCM SC-104 形式のデコードを行う．第 4.2.7節を参照．
3. 第 4.2.2節にある msg 構造体の作成をする．
4. inputq にエンキューする．

「第 4.2.7 節」　補正情報のデコード

GNSS 受信機から読み込まれた補正情報はそのままでは RTCM SC-104 形式ではない．これに bit 操作を施すことにより RTCM SC-104 形式にする必要がある．具体的な操作は付録を参照のこと．RTCM SC-104 形式に変換された補正情報は複数のメッセージタイプを含んでいる．Scrache

では補正情報はメッセージタイプごとに分割して扱う．そこでフレーム(付録を参照) 単位に分割し、msg 構造体に格納し、エンキューする．

第 4.2.8 節」　上流プロトコルモジュールの実装



```
struct msg *next;
struct bsid bsid;
u_char type;
u_char msg[];
```

図 4.19: 上流プロトコルモジュールの詳細

上流プロトコルモジュールは配送局、移動局に用いられる．主な役割は以下のようになる．

1. ネットワークから第 3.4.2.2節にある補正情報応答メッセージの形で受信する．
2. 第 4.2.2節にある msg 構造体の作成をする．
3. inputq にエンキューする．

第 4.2.9 節」　下流 GNSS モジュールの実装

下流 GNSS モジュールは移動局に用いられる．主な役割は以下のようになる．

1. outputq からデキューする．

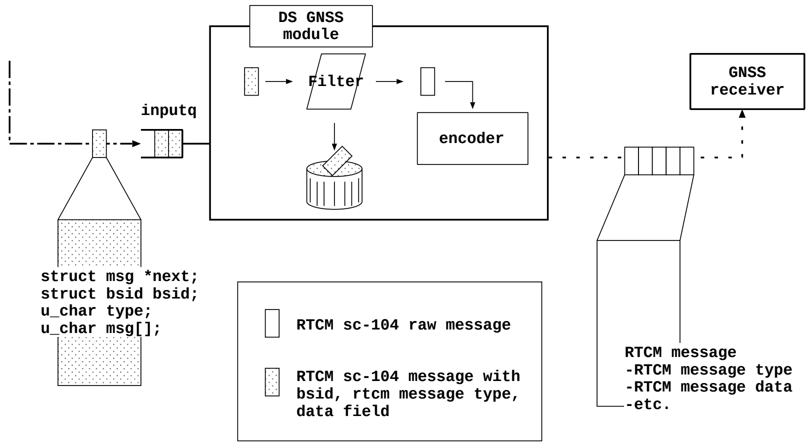


図 4.20: 下流 GNSS モジュールの詳細

2. シリアルデバイスに接続されている GNSS 受信機に不必要なメッセージをフィルタリングする．
3. 第 4.2.2節にある msg 構造体から、RTCM SC-104 形式のエンコードを行う．第 4.2.10節を参照．
4. GNSS 受信機へシリアルデバイス経由で RTCM SC-104 形式の補正情報を書き込む．

「第 4.2.10 節」　補正情報のエンコード

基本的な操作は第 4.2.7節の逆の操作になる．Scrache 内部で扱われる補正情報は msg 構造体単位になっている．これを outputq からデキューする．この段階ではやはりフレーム単位になっている．これに bit 操作を施し、複数のフレームを結合して GNSS 受信機に送り込む．具体的な bit 操作は付録を参照のこと．

「第 4.2.11 節」　下流 GNSS プロトコルモジュールの実装

下流 GNSS プロトコルモジュールは基準局、配送局に用いられる．主な役割は以下のようになる．

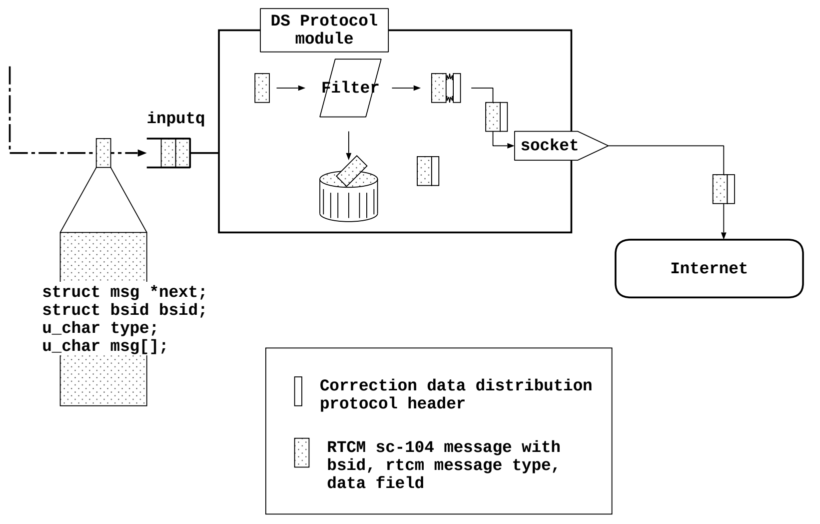


図 4.21: 下流プロトコルモジュールの詳細

1. outputq からデキューする．
2. 下流局に不必要なメッセージをフィルタリングする．
3. 第 4.2.2節にある msg 構造体から図 4.15構造体を作成する．
4. ネットワークに第 3.4.2.2節にある補正情報応答メッセージの形で送信する．

「第 4.3 節」 実装の問題点とその解決

今回の実装で幾つかの問題点が明らかになった．スレッドを使用した実装とくに FreeBSD の pthread に関して問題が起こることが多かった．問題点とその解決法を記す．

「第 4.3.1 節」 FreeBSD の Pthread に関する問題

FreeBSD の pthread は完全に実装されておらず，今回は *pthread_cancel* を使用できなかった．*pthread_cancel* はスレッド外部から目的とするスレッ

```
void
ds_protocol(struct usds_list *myds) {
  while (ds_breaktrigger != 1) { /* 状態変数が 0 の間、実行する */
    pthread_cond_wait(ds_breakmyself, breaktrigger_mutex);

    q_dequeue(outputq, dequeuemsg, outputq_mutex);

    sendto(sockfd, buf, msg_len , 0, server , sizeof(struct sockaddr));
    free(dequeuemsg);
  }
  return;
}
```

図 4.22: 状態変数で待っているスレッド

```
void
thread_delete(del) {
        pthread_mutex_unlock (&del->ds_breaktrigger_mutex);
        del->ds_breaktrigger = 1; /* 状態変数を 1 にする */
        pthread_mutex_unlock(&del->ds_breaktrigger_mutex);
        pthread_cond_signal (&del->ds_breakmyself);  /* ds のスレッド
に変更を通知 */
}
```

図 4.23: 状態変数を変更するスレッド

ドの終了に使用される関数である．このため外部からスレッドを終了させるために下記のような手段を用いた．

状態変数は，条件をテストし，偽の時はその条件に対してスリープし，真になったら目覚めさせる事が出来る機構である．他のスレッドが状態変数の値を変更する時は、*cond_signal()* を呼びスリープしているスレッドを起こし，状態変数の条件を再評価する．今回は，この機構を用いて外部から動的にスレッドを cancel する．

全ての DSL は状態変数が 0 の間 thread が実行される．しかし，外部のスレッドが DSL を終了させる時は状態変数を 1 にすればよい．

図 4.22は DS 内の状態変数の実装である．全ての DS は自分の状態変数が 0 の間実行される．*pthread_cond_wait()* で他のスレッドからの呼出を待つ．

図 4.23は DS を終了させる関数である．まず，状態変数の値を変える

前にmutexでロックを取得する．そして，状態変数を終了条件である1にする．その後mutexを開放し，*pthread_cond_signal()*を呼んでDSのスレッドを呼び起こす．

「第4.3.2節」 gethostbyname関数に関する問題

今回Scracheはpthreadを用いてマルチスレッドで実装した．その中にネットワーク操作をするスレッドが存在するが、スレッド内で名前を引くために*gethostbyname*をしている．この際に*gethostbyname*の戻り値を待つ間すべてのthreadがブロックされてしまうという現象が起きた．

応急的な措置だが、引くと想定される名前をetchostsに記載しておくことで多少の改善が見られた．さらに*gethostbyname*を使用しているスレッド内で返答を待っている間にsleepを入れることでさらに改善が見られた．しかしこれは場当たり的な対処法であると思われるので今後検討するべき問題点である．

第5章　評価

本章では，前章に基づき，インターネットを用いる補正情報配送機構の実装である Scrache の評価を行う．はじめに，プロトタイプの実装である wgpd の評価を行った．wgpd は基本的には Scrache と同様なプロトコルを使用しており，Scrache と同様の結果を得ることができる．定性的評価については既存の配送機構との比較を行い，また，複数の定量的な評価を行う．

「第 5.1 節」 プロトタイプモデルの評価

Scrache の実装を行う以前にプロトタイプの実装を行った．この実装を wgpd と呼ぶ．

「第 5.2 節」 wgpd の概要

wgpd も Scrache と同様，補正情報の配送に用いられるプログラムである．基準局では補正情報を GNSS 受信機から取得し，反対に移動局では GNSS 受信機に渡す．たま，Scrache と同様にそのあいだの通信路にはインターネットを用いる．しかし，配送局としては動作しない．

wgpd は FreeBSD3.1RELEASE 上で実装した．Scrache とは異なりスレッドを用いず一つのプロセスで動作する．

wgpd の処理の流れは以下の通りである．起動際に設定ファイルを読み込む．設定ファイルによって基準局として動作するのか，移動局として動作するのかを決定する．さらに設定ファイルからシリアルデバイス名や baudrate，ポート番号などのプログラム全体のパラメタを決定する．

基準局の場合は，シリアルデバイスを開き読み込む．また，あるポートを開き接続があるまで待機する．移動局からの接続があると，移動局を Scrache と同様に下流局のリストに追加し，シリアルデバイスから補正情報が読める場合には直ちに移動局へ送信する．

移動局の場合は，移動局のあるポートに接続をする．こうすることで基準局の下流局のリストに追加されると補正情報が送信されてくる．あらかじめシリアルデバイスを開きその補正情報を書き込む．

「第 5.3 節」 定量的評価

wgpd については，主に差動測位を用いた測位実験を行った．評価項目は以下の通り．

- 測位結果を元にした地図を作成し，これをができるか，既存の地図との比較を行う．
- 上記と同様に干渉測位ではどうかを調べる．
- 差動測位ではどの程度の補正が成功するか，補正成功率を調べる．

- また，その際の捕捉衛星数はどうなっているかを調べる．

「第 5.3.1 節」 実験環境

wgpd は Scrache に比べ単純なモデルだが確実に動作した．1999 年度の ORF[7] の公開実験の際には 3 つの移動局を同時に運用したが問題なく測位できた．ただし，インターネットとの通信路に用いた携帯電話の帯域の関係で干渉測位は不可能となることがあった．これは無駄な補正情報も配送しており，帯域を圧迫していることが原因である．なお，携帯電話は NTT DoCoMo[8] の Dopa[9] を用いた．

実験環境について記す．基準局は慶應義塾大学湘南藤沢キャンパス [10]o 棟に設置した．基準局の座標は WGS-84 座標系で，(N 35 23' 19.51845 E 139 25' 35.08721 94.64) である．アンテナを 5 階建ての建物の屋上に設置した．GNSS 受信機は基準局にアシュテック社製 Z18 受信機，移動局に同社 GG24 受信機 [11] を用いた．この受信機は GPS と GLONASS を利用できる．RTCM SC-104 のタイプは 1,18,19,31 に設定した．通信路は WIDE Internet を経由して基準局に接続した．携帯電話を除いて WIDE Internet 内の帯域は十分に広く，遅延は無視できる．

実験に用いたホストは以下のホスト A，ホスト B の 2 台である．ここではホスト A を基準局ホスト B を移動局とした．

- **ホスト A**

hostname	gps-bs-sfc.sfc.wide.ad.jp
CPU	Cyrix 6x86 120MHz
RAM Size	32Mbyte
Network Device	10BaseT
OS	FreeBSD3.1RELEASE

- **ホスト B**

hostname	SIC2000
CPU	VR4300
RAM Size	64Mbyte
Network Device	携帯電話 (9600bps)
OS	NetBSD CURRENTE

それぞれのホスト間の接続は実際のインターネットを用いた．実験ネットワークを敷設せず実際のインターネットを用いた理由は，基準局が地理

的に固定されたものであるためである．各ホストの間は1G，100M，10Mのイーサネットで接続されている．実験時，ホストAとホストBの間は1.503ms(ping100回の平均値)であった．

「第5.3.2節」　実験結果

まず，wgpdを用いた差動測位によって作成した湘南台駅-慶應義塾大学湘南藤沢キャンパス間の地図(図5.1)と湘南台周辺地図(図5.2)とを比較する．

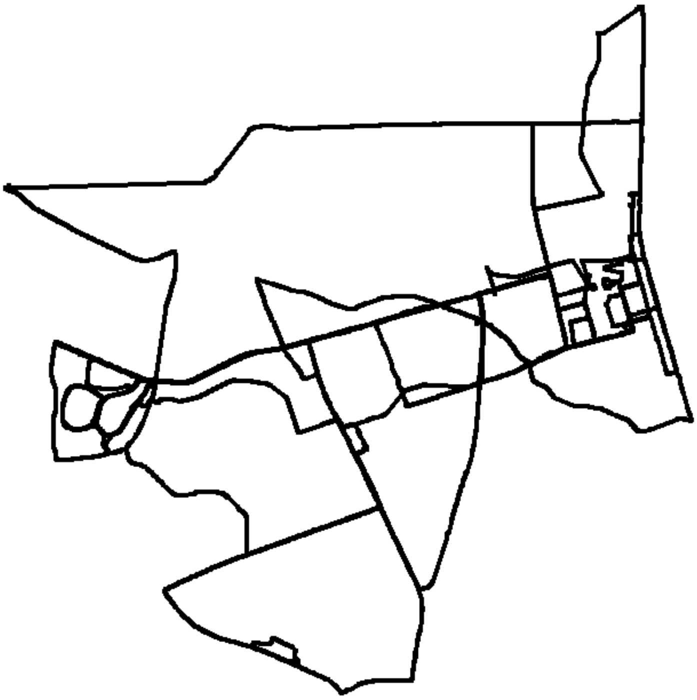


図5.1: wgpdによる差動測位

細かい精度での評価を行えないが，道路地図程度の精度をwgpdによる差動測位で出していることが読み取れる．また，拡大図(図5.3)をみると車線間の差異が読み取れる．wgpdを用いない単独測位での誤差が約

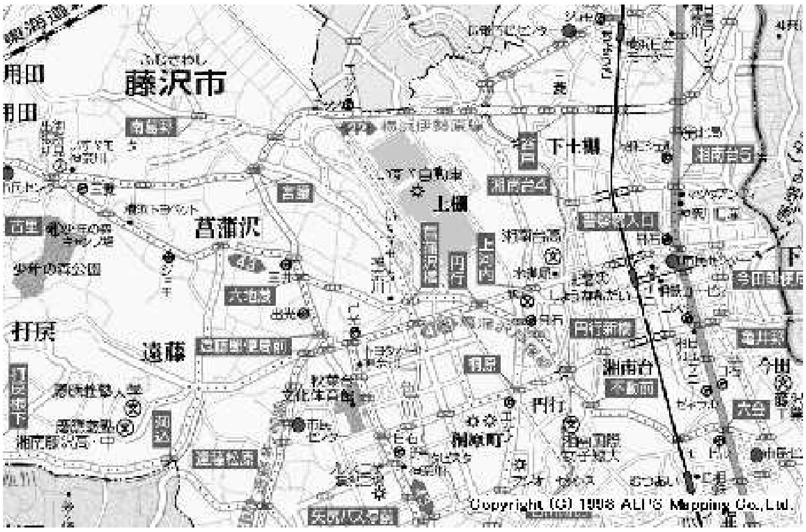


図 5.2: 湘南台-大学間周辺地図

100m である事を考えると補正されていることがわかる．

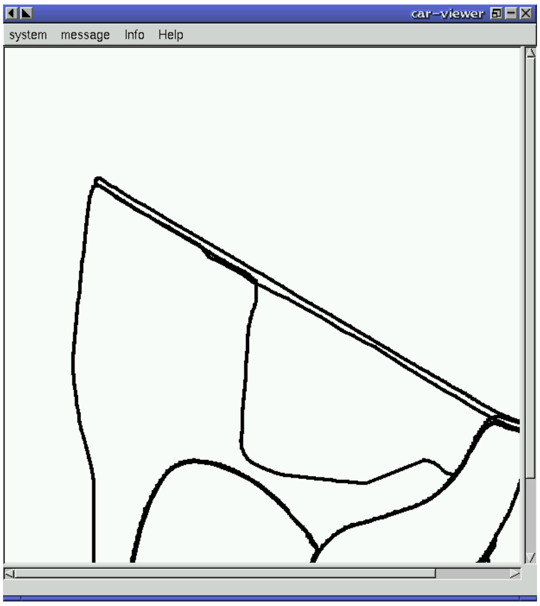


図 5.3: wgpd による差動測位 (拡大図)

次に wgpd による干渉測位について述べる．干渉測位を行うのは困難であった．これは捕捉する衛星数が多い際に補正情報が大きくなってしまう．すると携帯電話での帯域の不足が生じることがあるからである．まれに，天空の衛星数が少ない際に補正がかかることがあった．その際の干渉測位の結果を図 5.4に示す．

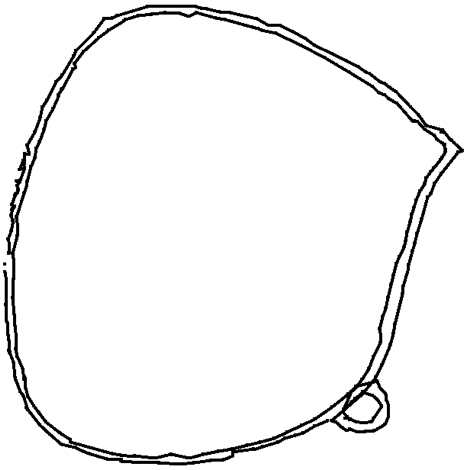


図 5.4: wgpd による干渉測位 :SFC 内周道路

その際の差動測位の成功数について表 5.1に示す．かなりの効率で補正に成功しているといえる．また，衛星の捕捉数による補正成功状況を表5.2に示す．衛星が多く見えているときに補正が成功する確率が高いことが分かる．

表 5.1: 差動測位の補正成功数

補正情報要求回数	12442 回
補正情報取得回数	8617 回
単独測位回数	1189 回
衛星捕捉失敗回数	1850 回
補正成功率	69.3%
衛星捕捉成功率	85.1%

表 5.2: 衛星の捕捉数による補正成功状況

衛星捕捉数	成功回数	失敗回数	成功率
3	743	156	82.6%
4	1923	300	86.5%
5	1480	363	80.3%
6	1528	226	87.1%
7	1718	107	94.1%
8	1226	29	97.7%
9	597	8	98.5%
10	140	0	100%
11	39	0	100%
12	6	0	100%

「第 5.4 節」 Scrache の評価

Scrache は wgpd と比較して定量的に同等以上性能を示している．特に通信路間に携帯電話などの無線リンクが存在する場合，Scrache の方が良い結果を生むと予測される．

Scrache は配送局を提案し，大規模は補正情報網を構築することが可能であり，来るべき補強機構に使用できる．また，移動局に最適な補正情報をどこから得れば良いかを意識することがなくアクセスの集中を防ぐこともできる．

Scrache については，主に差動測位を用いた測位実験を行った．評価項目は以下の通り．定性的評価については既存の配送機構との比較を行う．

- 差動測位の測位結果が単独測位の測位結果と比べてどうか調べる．
- 配送局の精度劣化に対する影響を調べる．

「第 5.4.1 節」 実験環境

Scrache の評価のための実験環境について記す．wgpd の場合と同様に，基準局は慶應義塾大学湘南藤沢キャンパス *o* 棟に設置した．アンテナを 5 階建ての建物の屋上に設置した．GNSS 受信機はアシュテック社製 Z18 受信機を用いた．この受信機は GPS と GLONASS を利用できる．RTCM SC-104 のタイプは 1,18,19,31 に設定した．通信路は WIDE Internet を経由して基準局に接続した．携帯電話を除いて WIDE Internet 内の帯域は十分に広く，遅延は無視できる．

実験に用いたホストは以下のホスト A からホスト H までの 7 台である．

- ホスト **A**

hostname	gps-bs-sfc.sfc.wide.ad.jp
CPU	Cyrix 6x86 120MHz
RAM Size	32Mbyte
Network Device	10BaseT
OS	FreeBSD3.1RELEASE

- **ホスト B**

hostname	kuju.mist.sfc.wide.ad.jp
CPU	Intel Pentium 75MHz
RAM Size	112Mbyte
Network Device	10BaseT
OS	FreeBSD3.3-RELEASE

- **ホスト C**

hostname	airmax.mag.keio.ac.jp
CPU	Intel PentiumII 650MHz
RAM Size	128Mbyte
Network Device	10BaseT
OS	FreeBSD2.2.8-RELEASE

- **ホスト D**

hostname	zero.sfc.wide.ad.jp
CPU	Intel Pentium 133MHz
RAM Size	64Mbyte
Network Device	10BaseT
OS	FreeBSD3.3-RELEASE

- **ホスト E**

hostname	tintin.mist.sfc.wide.ad.jp
CPU	Intel Pentium 133MHz
RAM Size	32Mbyte
Network Device	10BaseT
OS	FreeBSD 3.3-RELEASE

- **ホスト F**

hostname	neo.sfc.wide.ad.jp
CPU	Intel Pentium II/Xeon/Celeron 375MHz
RAM Size	128Mbyte
Network Device	10BaseT
OS	FreeBSD 3.3-RELEASE

- **ホスト G**

hostname	baystars.sfc.wide.ad.jp
CPU	Intel Celeron 300MHz
RAM Size	64Mbyte
Network Device	10BaseT
OS	FreeBSD 3.1-RELEASE

- **ホスト H**

hostname	azure.sfc.wide.ad.jp
CPU	Intel Pentium Pro 200MHz
RAM Size	128Mbyte
Network Device	10BaseT
OS	FreeBSD 3.3-RELEASE

- **ホスト H**

hostname	SIC2000
CPU	VR4300
RAM Size	64Mbyte
Network Device	携帯電話 (9600bps)
OS	NetBSD CURRENTE

それぞれのホスト間の接続は図5.5のようになっている．

それぞれのホスト間の接続は実際のインターネットを用いた．実験ネットワークを敷設せず実際のインターネットを用いた理由は，基準局が地理的に固定されたものであるためである．各ホストの間は1G，100M，10Mのイーサネットで接続されている．

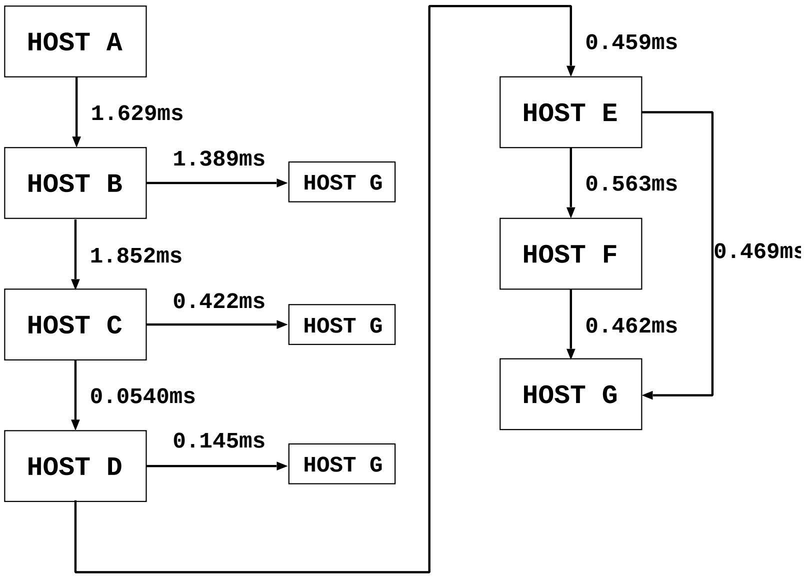


図 5.5: 各ホスト間の接続

「第 5.4.2 節」　定量的評価

Scrache についての定量的に評価を行った．まずはじめに Scrache を用いない単独測位について，1 時間の測位を行い測位結果を緯度経度でプロットした．その結果を図 5.6に示す．単独測位ではこのように大きく測位点がぶれることが分かる．因に経度方向で約 100m のぶれが認められる．

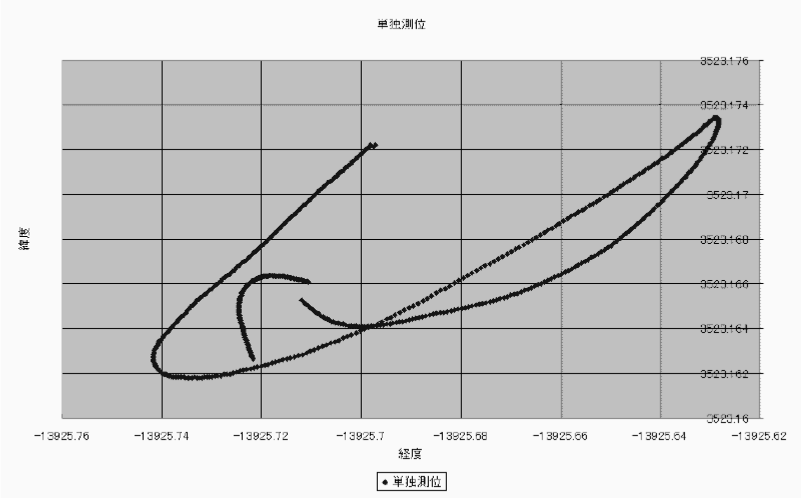


図 5.6: 単独測位結果

つぎに，Scrache を用いた差動測位を行い，単独測位と同時に測位結果を緯度経度でプロットした．その結果を図 5.7に示す．基準局にホスト A，移動局にホスト B を用いた．

この結果から差動測位で経度方向に約 20m 程度までぶれが小さくなっているのが認められる．よって Scrache をもちいた差動測位は有効である．

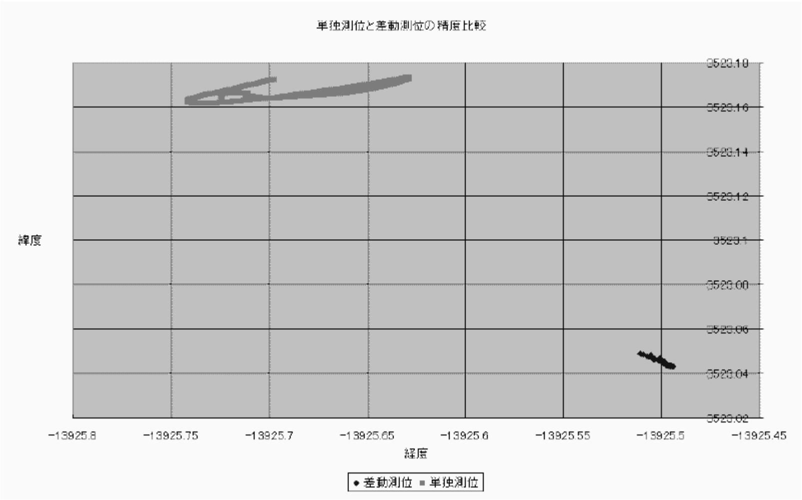


図 5.7: 単独 ･差動測位比較

差動測位の際には，基準局と移動局の間の通信路での遅延が増加するに従い，精度の劣化や差動測位が使用できなくなる現象が一般に知られている．そこで，配送局を増やした際の遅延の増加による精度への影響を調べた．

はじめに，配送局を用いないもの，配送局を 1 つ用いるもの，配送局を 2 つ用いるものについて図 5.8に示す．それぞれ基準局にはホスト A を用い，1 つ目の配送局にはホスト C，2 つ目の配送局にはホスト E を，移動局にはホスト D を用いた．

ここで，配送局を用いない差動測位は図 5.7の差動測位と同じ結果である．配送局の数を増やすにしたがって多少の精度の劣化や，差動測位をできずに単独測位になるといった現象が見られるものの配送局の数が増えることによる影響はあまり認められない．配送局を 0 個から 2 個まで増やした際には，それぞれ約 20m 程度のぶれを認めた．

次に上記と同様に配送局を 3 つから 5 つまで増加させ，それを配送局を用いないものと比較を行った．これを図 5.9にしめす．この図でも配送局を 4 つにした際に差動測位ができず単独測位になってしまっているも

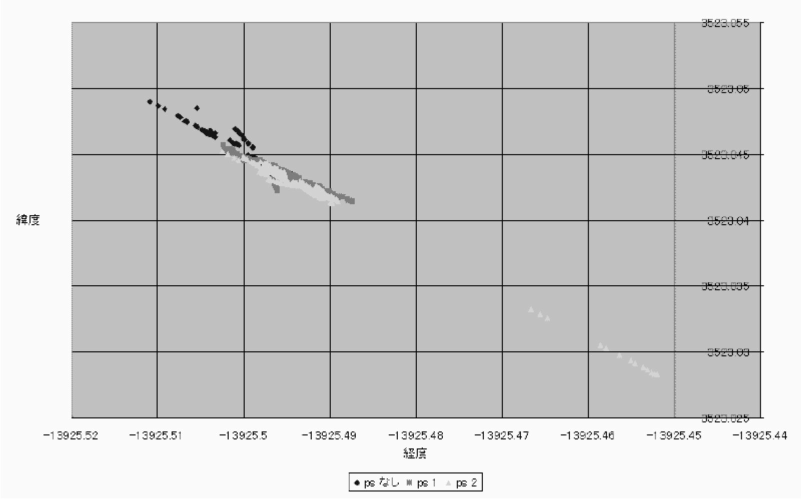


図 5.8: 配送局の増加と測位精度の劣化 (1)

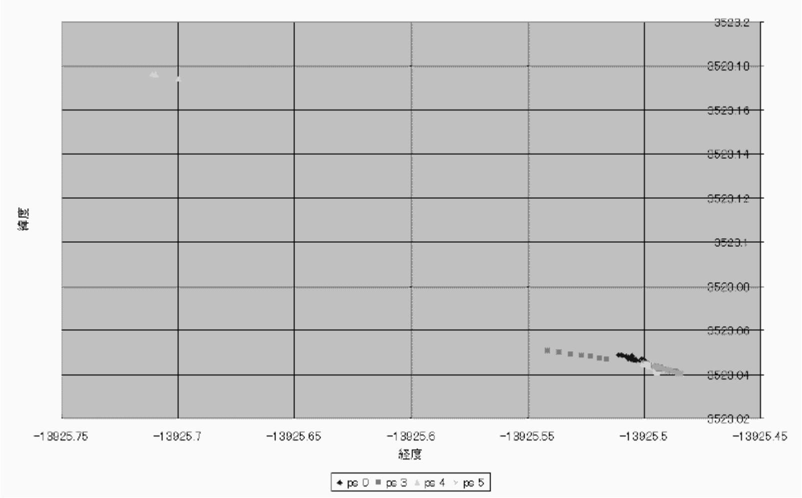


図 5.9: 配送局の増加と測位精度の劣化 (2)

のが見受けられるもの，それ以外の場合は図5.7のとほぼ同じ結果が見受けられる．つまり，配送局の数が増えることによる影響はあまり認められない．

「第5.4.3節」　定性的評価

Scracheはwgpdと比較して定量的に同等の性能を示している．特に通信路間に携帯電話などの無線リンクが存在する場合，Scracheの方が良い結果を生むと予測される．

Scracheは配送局を提案し，大規模は補正情報網を構築することが可能であり，来るべき補強機構に使用できる．また，移動局に最適な補正情報をどこから得れば良いかを意識することがなくアクセスの集中を防ぐこともできる．

既存の補正情報配送機構とwgpd，Scracheの相対評価を表にしたものを表5.3に示す．

表5.3: Evaluation of All Systems

[Evaluation of All Systems]

通信方式	利用範囲	規模性	補正の種類	要求事項	精度
FMラジオ副搬送波	局地的	無制限	差動測位(不完全)	専用の受信機	低い
携帯電話	広い	なし	差動測位	携帯電話	普通
無線	局地的	無制限	差動、干渉測位	専用の受信機	高い
アルプス	広い	あり	差動測位	インターネット接続	普通
wgpd	広い	あり	差動、干渉測位	インターネット接続	高い
Scrache	広い	あり	差動、干渉測位	インターネット接続	高い

無線はMCA無線・中波ビーコンを指す

Chapter 6

結論

本章では，結論を述べるとともに，インターネットを用いる補強機構について触れる．インターネットを用いる補正情報配送の今後の課題について言及する．

「第 6.1 節」 今後の課題

なお，Scrache は基準局選択の機構および補強機構 (Augmentation System) については提供しないが，将来的にそれらを導入することを考慮して設計されている．

「第 6.2 節」 Internet Based Augmentation System について

補強機構とは多数の基準局から補正情報を収集しそれらの固有の誤差を打ち消した上で配送する機構である．米国の WAAS や日本の MSASS が計画中でありどちらも軌道上の静止衛星から修正後の補正情報を配送するという機構である．

「第 6.3 節」 Scrache の将来的な実装に関する問題

「第 6.3.1 節」 位置情報の共有に関する問題

Scrache の移動局は現在の緯度経度を補正情報の要求の際に用いるが，これは GNSS を用いるほかのプログラムと排他的である．つまり Scrache が動作している際はほかのプログラムは測位情報を使用できないことになる．

この問題を回避するために，位置情報共有の手段を幾つか提案する．第一に Scrache が位置情報を提供するプログラムになるというものである．これはほかのプログラムが何らかの方法で Scrache にアクセスし，位置情報を得る形になる．

第二に何らかの仮想位置デバイスを作成しそこから位置情報を得る形にする．この場合位置情報を必要とするプログラムのデバイスを読むことになる．実装の仕方にもよるが，複数のプログラムには複数のデバイスを

用意する．

「第 6.3.2 節」　補強機構の実装に関する問題

Scrache はインターネットを用いた補強システム (以下，IBAS: Internet Based Augmentation System) に対応できるように設計された．

具体的には下流側に outputq をもったモジュールを新たに追加し新たな補正情報発信局として補正情報を生成，それを inputq へエンキューするという手法をとる．補正情報のループが起こらないように，このモジュール内にもフィルタをもうける．

「第 6.3.3 節」　FreeBSD の Pthread に関する問題

FreeBSD の libc_r には *pthread_canlcel()* が含まれていない．しかし，OpenBSD の libc_r には *pthread_canlcel()* が含まれている．これを FreeBSD に移植することにより，この問題を解決する予定である．

「第 6.3.4 節」　プロトコルの標準化に関する問題

Scrache を用いて複数の基準局が協調して動作できることが分かった．そこで，Scarache をもちいてあらたな基準局網を作成する際に Scrache と同様のプロトコルを用いるプログラムであれば，それらも協調して動作をすることができる．したがって，プロトコルの標準化は必要である．

「第 6.4 節」　まとめ

最後に，本研究で行ったことについてまとめる．

本論文ではインターネットを用いた GNSS(Global Navigation Satellite System) の補正情報配送機構を提案し，このモデルに基づいて設計，実装した配送機構について述べた．また，利用者の要求に応じた補正情報を配送するために補正情報配送プロトコル，補正情報配送制御プロトコルを提案，設計した．同プロトコルを使用してインターネットを用いる補正情報配送機構を設計，実装，評価した．

はじめに，既存の補正情報配送機構に関しての考察を行い，サービス

エリアの狭さや導入時のサービス提供者の負担の大きさなどの問題点があることを示した．これらの問題解決のため，インターネットを用いた補正情報配送機構を構築するアプローチを選択した．既存の補正情報配送機構では導入時に補正情報の有効範囲や補正の種類等について決定したが，インターネットを用いる際にそれらに幅広く対応する必要がある．よって，補正情報についての表現を行い，補正情報の有効範囲や補正の種類等に基づいた要求応答が可能になった．

また，本機構では配送局という概念を導入した．これは補正情報の配送を専門に行い，やはり補正情報の有効範囲や補正の種類等に基づき複数の要求に対し対応できる．

その結果既存の補正情報配送機構の問題点である送受信設備や規模性等の点を解消することができた．

本機構の実現により，利用者はインターネットに接続することで，GNSS補正を容易に行うことができる．大掛かりな無線設備等が必要でないため，既存の機構に比べ補正サービスの提供者の負担を軽減することができる．将来構築されるであろう補強機構の実現に必要となる補正情報の配送管理を容易にする．以上よりインターネットを用いて補正情報配送を行うことは十分に利点があると結論づける．

Bibliography

[1] International GPS Service IGS. *http://igscb.jpl.nasa.gov/*. NASA, January, 2000.

[2] 土屋 淳 ･辻 宏道. *GPS* 測量の基礎. 社団法人 日本測量協会 , July, 1995.

[3] FM 多重放送 D-GPS システム. *http://www.gpex.co.jp/*. (株) 衛星側位情報センター, January, 2000.

[4] 国土地理院 RTK. *http://vldb.gsi-mc.go.jp/sokuchi/gps/*. 国土地理院, January, 2000.

[5] 港湾建設工事 RTK. *http://www.motnet.go.jp/*. 運輸省 港湾建設局, January, 2000.

[6] ディファレンシャル GPS 補正情報のインターネット配信に関する公開実験の実施について . *http://www.alpsmap.co.jp/overview/letter/19981120/dgps.html*. 株式会社アルプス社, January, 2000.

[7] Open Research Forum '99. *http://www.sfc.keio.ac.jp/ORF99/*. SFC 研究所, January, 2000.

[8] NTT DoCoMo 中央. *http://www.nttdocomo.co.jp/*. NTT 中央移動通信網株式会社, January, 2000.

[9] DoPa. *http://www.nttdocomo.co.jp/products/dopa/home.html*. NTT 中央移動通信網株式会社, January, 2000.

[10] 慶應義塾大学 藤沢キャンパスの位置 . *http://www.sfc.keio.ac.jp/info/index.html*. 慶應義塾大学, January, 2000.

[11] Z18 GG24 受信機速 *http://www.ashtech.com/*. アシ ュッテ ック社, January, 2000.

謝辞

本研究を進めるにあたり，御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，同学部教授理事・教授の徳田英幸博士に感謝いたします．また，絶えず御助言と御指導をいただきました，同学部助教授楠本博之博士と同学部講師の中村修博士に深い感謝の意を表します．

また，日頃より研究活動のご指導を頂きました，植原啓介氏，渡辺恭人氏に感謝します．公私にわたる指導頂いた関谷勇司氏，杉浦一徳氏，土本康生氏をはじめとする慶應義塾大学政策メディア研究科後期博士課程の方々，

並びに，実装に関し多大なご迷惑をお掛けしたとともに面倒をみて頂いた湧川隆次氏に特に感謝するとともに，論文についてご指導頂いた杉本信太氏，大西孝義氏，西田視磨氏，土井祐介氏，宅間啓氏，増田康人氏，小原泰弘氏をはじめとする慶應義塾大学政策メディア研究科修士課程の方々，

英文の指導をして頂いた石井かおり氏，三川荘子氏，単位取得などの公私にわたるご支援を頂いた三屋光史朗氏，服部亜紀子をはじめとする慶應義塾大学環境情報学部・総合政策学部の徳田村井楠本中村研究室の諸氏には，本研究を進めていく上で様々な議論をして頂き，また耐えざる励ましとご援助を頂きました．ここに，深い感謝の念を表します．

さらに執筆の苦労を共にした藤沢キャンパス七期生のみなさまに感謝します．

学外では，WIDEプロジェクトのみなさま，特に，roverグループ，Internet-CARグループのみなさまに暖かいご指導を頂きました．特に奈良先端技術大学院大学大学附属図書館研究開発室の羽田久一氏に感謝します．

研究室入室より絶え間ないご指導を頂いたマイクロソフト株式会社の石井徹氏，論文執筆のご指導を頂いた宮川祥子氏，研究の指針を頂いた新美誠氏に深く感謝します．

不馴れなGNSSに関しご迷惑をお掛けしたとともにご教授を頂いた(株)DXアンテナの方々特に茶珎俊一氏に深く感謝します．

また，私的な面で多大なる支援をしてくれた慶應義塾大学環境情報学部の東田あすか氏に感謝します．

以上を持って謝辞といたします．

Appendix A

About RTCM SC-104

A.1 RTCM SC-104の概要

RTCM SC-104は，誤差補正量、疑似距離補正量、座標値の各データの標準配送規格であり，正しくはRadio Technical Commission for Maritime Service SC-104という．

RTCM SC-104は米国の船舶電波技術委員会(RTCM:Radio Technical Commission for Maritime Service)の中の第104委員会SC-104(Special Committee No. 104)によって策定された規格である．これを単にRTCM SC-104規格、RTCM規格、RTCM等と呼ぶ事がある．

A.2 RTCM SC-104の構成

RTCM SC-104のメッセーのひとまとめのデータをフレームと呼び，30bitからなる複数のワードから構成されている．すべてのフレームのは「2ワードヘッダ」と呼ばれる2ワードから開始される．

A.3 2ワード各メッセージ共通ヘッダ

2ワードヘッダについて解説する．文字通り2ワードから構成されるフレームのヘッダである．1ワード目のの最初の8bitは0110110のプリアンブル，次に6bitの フレームIDが続く．これは後述するメッセージのタイプをあらわす．次の12bitは基準局のIDである．最後の8bitがパリティチェック用のbit列であり，これで1ワード目が終了する．

```
First Word of 2 words header
```

```
          1         2         3
0123456789012345678901234567890
01100110M     LM        LM    L
|      ||     ||        ||    |
|      ||     ||        |[PARITY]
[PREAMBLE]    ||        |
        |     ||        |
     [MESSAGE TYPE]     |
              ||        |
         [REFERENC STATION ID]
```

2 ワード目は13bit の修正Z カウントから始まる．これはデータ測定時刻を毎時の最初の時間からの秒数をLSB を0.6 秒としている．(意味不明) このフィールドは0-5999 のいずれかとなる．次の3bit は連続番号と呼ばれ0-6 を連続的に変化してデータの連続性をチェックするのに使用される．次の5bit はフレームが幾つのワードから構成されるかを示す．次の5bit は基準局の健康状況を示す．7 は基準局が稼働していないこと，6 は稼働しているがモニタされていないことを示す．それ以外の状況についてはサービス提供者が定義して良い．

```
Second Word of 2 words header
          1         2         3
0123456789012345678901234567890
M            LM LM   LM LM    L
|            || ||   || ||    |
|            || ||   || |[PARITY]
[MODIFIED Z-COUNT]   || |
              | ||   || |
[SEQUENCE NUMBER]|   || |
                 |   || |
     [LENGTH OF FRAME]| |
                      | |
                [STATION HEALTH]
```

A.4 RTCM のエンコード, デコード

RTCM SC-104 はエンコード, デコードともに複雑な計算を要する．以下にデコード, エンコードの順に解説する．

A.4.1 RTCM SC-104 のデコード

1. フレームの切り出し

   RTCM SC-104 は 0x0d を区切りとして 1 フレームを切り出すことができる．切り出した行の先頭から f または Y を探し，それ以降の文字数が 5 の倍数でないときはその行を破棄するべきである．(もう少し正しく書くべき)

   ```
     (補正情報の例　キャラクタにキャストしたもの）
   fAF{t}aX}b_CAMmV^{Gz‘kT{[KCw_gR~}ZJI@‘y
   A_dHR]CHb]PCDPuDvmcO
   Y~yDK|xkAs_CLhKcaDxeP}WoLKGwHuCCe‘
   O@TKuHOd}L@XKdVIftGcBEtY@}{|M\GWUD
   ```

2. ワードの切り出し

   1 フレームは 2+N ワードで構成されている．先頭の 2 ワードは前述の 2 ワードヘッダである．それ以降のワード数はメッセージタイプによって異なる．1 ワードは 30bit で構成されている．ただし，これを 5byte で表現しているため，各 byte の先頭の 2bit は破棄し，下位 6bit を使用する．下位 6bit を「並び bit 反転」し，それを 5byte 連結して 1 ワードを構成する．(下位 6bit X 5byte = 30bit)

   次の 1 ワードをデコードするには直前のワードの 30 ビット目が 1 の場合上位 24bit を「値 bit 反転」することに注意する．これは 1 ワード目をデコードする際に，先頭が f でなくて Y であるときにも同様である．(到達したワードの直前のワードによって値 bit 反転している)

   （例）

   （1）最初の 5 バイトが `fAF{t` の場合。

   ```
   f  1st byte    b7 b6 b5 b4 b3 b2 b1 b0         下位６ビット
   の並びを反転したもの
                   0  1  1  0  0  1  1  0   >>>        0 1 1 0 0 1
   A  2nd byte    b7 b6 b5 b4 b3 b2 b1 b0
                   0  1  0  0  0  0  0  1   >>>        1 0 0 0 0 0
   ```

```
F  3rd byte    b7 b6 b5 b4 b3 b2 b1 b0
                0  1  0  0  0  1  1  0   >>>        0 1 1 0 0 0
{  4th byte    b7 b6 b5 b4 b3 b2 b1 b0
                0  1  1  1  1  0  1  1   >>>        1 1 0 1 1 1
t  5th byte    b7 b6 b5 b4 b3 b2 b1 b0
                0  1  1  1  0  1  0  0   >>>        0 0 1 0 1 1
```

上記５バイトから得られた１ワードのデータ

```
 |    1st     |    2nd     |    3rd     |    4th     |    5th     |
  0 1 1 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 1 1 0 1 1 1 0 0 1 0 1 1
 |   preamble      |   type     |    station id      |  parity    |
```

(2) 次の５バイトが `}aX}b` の場合。

まず (1) と同様に３０ビットを得る。

```
}  1st byte    b7 b6 b5 b4 b3 b2 b1 b0          下位６ビット
の並びを反転したもの
                0  1  1  1  1  1  0  1   >>>        1 0 1 1 1 1
a  2nd byte    b7 b6 b5 b4 b3 b2 b1 b0
                0  1  1  0  0  0  0  1   >>>        1 0 0 0 0 1
X  3rd byte    b7 b6 b5 b4 b3 b2 b1 b0
                0  1  0  1  1  0  0  0   >>>        0 0 0 1 1 0
}  4th byte    b7 b6 b5 b4 b3 b2 b1 b0
                0  1  1  1  1  1  0  1   >>>        1 0 1 1 1 1
b  5th byte    b7 b6 b5 b4 b3 b2 b1 b0
                0  1  1  0  0  0  1  0   >>>        0 1 0 0 0 1
```

上記５バイトから得られた１ワードのデータ

```
 |    1st     |    2nd     |    3rd     |    4th     |    5th     |
  1 0 1 1 1 1 1 0 0 0 0 1 0 0 0 1 1 0 1 0 1 1 1 1 0 1 0 0 0 1
```

↓

前のワード（1）の３０ビット目が1なので値を反転する。

但し最後のパリティを表す6ビットは反転しない。

↓

最終的に得られた2ワード目のデータ

```
  0 1 0 0 0 0 0 1 1 1 1 0 1 1 1 0 0 1 0 1 0 0 0 0 0 1 0 0 0 1
 |    modified z count     | seq#|  length |helth|  parity   |
```

以上の操作を繰り返し、１フレーム分のデータを得ます。

（3）行の最初が f でなく Y で始まる場合。

（2）の例と同じく最初のワードの先頭の２４ビットについて、値を反転します。２ワード目からは（1）(2）と同じ処理をします。

3. パリティのチェック

得られたビットのパリティチェックを行う際には次のようにする．得られたワードの先頭24bitの値をd1-d24，直前のワードの29,30番目のbitの値をpD29，pD30とする．これらの26bitを以下のように演算してD25-D30のパリティを求める．

```
(パリティ演算の例)
ここで ^ は XOR 演算を行うことを表す

D25 = pD29^d1^d2^d3^d5^d6^d10^d11^d12^d13^d14^d17
                                             ^d18^d20^d23

D26 = pD30^d2^d3^d4^d6^d7^d11^d12^d13^d14^d15^d18
                                             ^d19^d21^d24

D27 = pD29^d1^d3^d4^d5^d7^d8^d12^d13^d14^d15^d16
                                             ^d19^d20^d22

D28 = pD30^d2^d4^d5^d6^d8^d9^d13^d14^d15^d16^d17
                                             ^d20^d21^d23

D29 = pD30^d1^d3^d5^d6^d7^d9^d10^d14^d15^d16^d17
                                         ^d18^d21^d22^d24

D30 = pD29^d3^d5^d6^d8^d9^d10^d11^d13^d15^d19^d22^d23^d24
```

これによって得られたD25-D30がそのワードのパリティとして25-30番目のbitとなり，これと一致しないワードはパリティエラーとなる．このパリティの演算には直前のワードが必要であり，直前のワードが送られてこない際にはエラーチェックはできない．

A.4.2 RTCM SC-104のエンコード

2.2 RTCM SC-104のエンコード基本的にはデコードの逆を行えば良い．ただし，デコードの際に無視すれば良いスタートビットなどに注意する必要がある．

- ワードの生成

エンコード前のメッセージは，各6bitの5byte，計30bitの「エンコード前ワード」からなる．各「エンコード後ワード」を生成するには，各byteを「並びbit反転」しなくてはならない．さらに，先頭に「スタートビット」と呼ばれる01を挿入する．したがってこの時点で各byteは8bitである．

1フレームは2+Nワードで構成されている．先頭の2ワードは前述の2ワードヘッダである．それ以降のワード数はメッセージタイプによって異なる．

- パリティの演算

  得られたビットのパリティ演算を行う際には次のようにする．得られたワードの先頭24bitの値をd1-d24，直前のワードの29,30番目のbitの値をpD29，pD30とする．これらの26bitを以下のように演算してD25-D30のパリティを求める．

  これはデコード時と同様に行えば良い．

- フレームの生成
  RTCM SC-104は0x0dを区切りとしており，フレーム間にそれを挿入する必要がある．

3. メッセージタイプ別解説

各メッセージタイプごとに解説を行う．

3.1.1 RTCM SC-104のメッセージタイプ

表 A.1: RTCM SC-104 MESSAGE TYPES(再掲)
[RMCM SC-104 MESSAGE TYPES(再掲)]

TYPE NO.	STATUS	TITLE
1	Fixed	Differential GPS Corrections
2	Fixed	Delta Differential GPS Corrections
3	Fixed	GPS Reference Station Parameters
4	Tentative	Reference Station Datum
5	Fixed	GPS Constellation Health
6	Fixed	GPS Null Frame
7	Fixed	DGPS Radiobeacon Almanac
8	Tentative	Pseudolite Correction Set
9	Fixed	GPS Partial Correction Set
10	Reserved	P-Code Differential Corrections
11	Reserved	C/A-Code L1,L2 Delta Corrections
12	Reserved	Pseudolite Station Parameters
13	Tentative	Ground Transmitter Paramters
14	Tentative	GPS Time of Week
15	Tentative	Ionospheric Delay Message
16	Fixed	GPS Special Message
17	Tentative	GPS Ephemerides
18	Fixed	RTK Uncorrected Carrier Phases
19	Fixed	RTK Uncorrected Pseudoranges
20	Tentative	RTK Carrier Phase Corrections
21	Tentative	RTK/Hi-Acc. Pseudorange Corrections
22	Tentative	Extended Reference Station Parameters
23-30	–	Undefined
31	Tentative	Differential GLONASS Corrections
32	Tentative	Differential GLONASS Reference Station Parameters
33	Tentative	GLONASS Constellation Health
34	Tentative	GLONASS Partial Correction Set
35	Tentative	GLONASS Radiobeacon Almanac
36	Tentative	GLONASS Special Message
37	Tentative	GNSS System Time Offset
38-58	–	Undefined
59	Fixed	Proprietary Message
60-63	Reserved	Multipurpose Usage