

卒業論文 2002年度(平成14年度)

利用環境に応じた通信環境設定の統括的制御機構の提案

慶應義塾大学 総合政策学部

小柴 晋

s99339sk@sfc.keio.ac.jp

指導教員

慶應義塾大学 環境情報学部

村井 純

徳田 英幸

楠本 博之

中村 修

南 政樹

平成15年2月21日

利用環境に応じた通信環境設定の統括的制御機構の提案

本研究は、通信環境、用途、利用者の意思等の情報を統括的に管理し、動的にシステムの設定を制御する機構を提案する。

インターネット接続環境が充実に伴い、様々な通信デバイスを利用し、時と場所を選ばずインターネットに接続できるようになった。一方、自動車や携帯電話といった、インターネット接続性を維持しながら自由に移動する計算機が登場してきた。これらの計算機の通信環境を支援するために、Mobile IPv6 や Location Independent Network for IPv6 といった様々な移動体通信支援プロトコルが提案されている。

現状では、すべての環境で利用可能で、すべての用途に適した通信デバイスは存在しない。そのため、利用者は状況に応じて通信デバイスを選択利用すべきである。また、移動体通信支援プロトコルは、それぞれが想定している状況ごとに設計されているため、状況に応じて選択利用すべきである。

このように、通信環境、用途、利用者の意思などによって、最適な通信環境の設定が異なってくる。したがって、通信環境、用途、利用者の意思等の情報を統括的に管理し、動的にシステムの設定を制御する機構の実現が必要不可欠である。

本研究では、この機構を実現するためには、計算機が持つ情報の監視、通信状態に応じた最適な通信設定の判断、設定変更処理の 3 つの機能を実現する動的移動処理選択支援モデルを提案した、また、提案したモデルに従いシステムの設計、実装を行なった。いくつかのケーススタディを基に評価実験を行なった結果、本システムの有効性を確認することができた。

キーワード

1. インターネット, 2. 移動体通信, 3. 通信プロトコル構成, 4. 動的切り替え, 5. ポリシー

慶應義塾大学 総合政策学部

小柴 晋

A Proposal of Policy-Based Centralized Host Configuration Control Mechanism
--

This research propose the system which manage connection environment, use, needs of users and configure of the system dynamically.

We can connect the Internet because of bringing up the connectivity to the Internet. However, computers turn up which can move freely through connecting the Internet. Mobile IPv6 and Location Independent network for IPv6, which are mobile node communication support protocol , are proposed to support such a connection environment.

Communication Devices which can use any environment and proper any use doesn't consist, as it stands. Users must select communication devise by circumstances. Additionally, users should select and use by circumstances mobile node communication support protocol by circumstances, because of the protocol are desinged according to the supposition.

As just described, configures of communication environment are different in communication environment, usage, intent of users. The system is needed, which managing communication encironment, usage, requirement of users and dynamically control system configuration.

In this paper, we proposed a dynamicaly supporting movement model which are monitoring the information held by the computers, adjusting communication configuring by condition of the communication and process of modifying the configuration. Additionally, based on the proposed model, we have designed , implemented, evaluated and proved a systems effectiveness.

Keywords :

1. Internet, 2. Mobile Communication, 3. Protocol Pairing,
4. Dynamic Host Configuration, 5. User Policy

Keio University , Faculty of Policy Management

Susumu Koshiba

目次

第1章	序論	1
1.1	はじめに	1
1.2	本研究の目的	2
1.3	本論文の位置付け	2
1.4	本論文の構成	3
第2章	移動体通信環境への要求	4
2.1	対象とする移動体通信環境	4
2.1.1	インターネット ITS の目指す環境	4
2.1.2	移動体通信環境におけるサービス例	5
2.2	移動体通信環境への要求	6
2.3	想定される移動体通信環境	7
第3章	既存の移動体通信環境における問題点と解決アプローチ	10
3.1	移動体通信における問題点	10
3.1.1	移動体通信環境の多様化	10
3.1.2	利用者による差異	12
3.2	既存の移動体通信における問題点	13
3.3	問題点解決へのアプローチ	14
第4章	関連研究	16
4.1	MIBsocket	16
4.2	The Ensemble System	17
第5章	動的移動処理選択支援モデル	19
5.1	問題点の解決に必要な機能要件	19
5.1.1	移動体計算機が持つ情報の監視	20
5.1.2	通信状態に基づいた最適な通信環境設定の判断	20
5.1.3	選択された通信環境設定の実行	22
5.2	動的移動処理選択支援モデルの提案	24
5.2.1	並列処理モデル	25
5.2.2	並列協調処理モデル	26
5.2.3	動的移動処理選択支援モデル	28

5.2.4	まとめ	30
第6章	システム設計	34
6.1	動的移動処理選択機構における用語の定義	34
6.2	前提とする移動体計算機的环境	34
6.3	設計概要	35
6.4	状態情報監視機構	35
6.5	適応処理判断機構	37
6.5.1	利用環境の状態情報の抽象化	37
6.5.2	利用者に対する高度な抽象化	38
6.6	適応処理実行機構	39
第7章	動的移動処理選択支援機構の実装	40
7.1	実装環境	40
7.2	実装概要	40
7.2.1	状態情報監視機構	40
7.2.2	適応処理判断機構	40
7.2.3	適応処理実行機構	41
7.3	状態情報監視機構：Mobility Socket の利用	41
7.4	適応処理判断機構	42
7.4.1	設定のパーズング	42
7.4.2	適応処理の検索	45
7.5	適応処理実行機構	46
第8章	評価	47
8.1	定性的評価	47
8.1.1	実験環境	47
8.1.2	実験内容	47
8.1.3	実験結果	49
8.2	定量的評価	50
8.2.1	実験環境	51
8.2.2	実験内容	52
8.2.3	実験結果	53
8.2.4	定量的評価に関する考察	53
8.2.5	定量評価まとめ	57
8.3	評価まとめ	57
第9章	結論	58
9.1	まとめ	58
9.2	今後の課題	58

付録 A Mobile IPv6(MIPv6)	62
A.1 MIPv6 で解決される問題	62
A.2 MIPv6 の用語	62
A.3 MIPv6 の動作概要	63

目 次

2.1	想定する移動体通信環境：移動体計算機は利用する通信インターフェースを状況に応じて切り替えながら通信を継続させる	8
2.2	想定する移動体通信環境：移動体計算機は利用する通信技術を状況に応じて切り替えながら通信を継続させる	9
3.1	例-移動体計算機が利用する通信プロトコル構成の変更	14
4.1	MIBsocket 概要：ネットワーク情報は MIBsocket を経由してカーネルとアプリケーション間で共有している	17
4.2	Ensemble システム概要	18
5.1	適応処理の呼び出し	23
5.2	例-並列処理モデルにおける処理の流れ	25
5.3	例-並列協調処理モデルにおける処理の流れ	27
5.4	例-動的移動処理選択支援モデルにおける処理の流れ	29
6.1	MMD 動作概要	36
7.1	msock から取得する情報-net/mobilitysocket.h	41
7.2	情報の意味とプリミティブな情報の対応付け	43
7.3	処理の種類と処理を行う関数の対応付け	43
7.4	通信の状態と処理の種類に対応付け	44
7.5	通信の状態と処理の種類に対応付け	45
7.6	情報の抽象化と処理の流れ	45
8.1	通信の状態と処理の種類に対応付け	49
8.2	評価の流れ	50
8.3	評価を行ったネットワークの構成	51
8.4	インターフェース切り替えに要する時間の比較	54
8.5	本システムと SFCMIP の処理オーバーヘッドの比較	56
A.1	MIPv6 における移動時の処理の流れ	64

表 目 次

5.1	各モデルにおける処理オーバーヘッド	31
5.2	各モデルにおける処理の正確さの比較	32
5.3	各モデルにおける処理の正確さの比較	33
6.1	各モデルにおける処理の正確さの比較	36
6.2	プリミティブな情報の抽象化	37
6.3	MIPv6 における処理と状態情報の対応	38
6.4	MIPv6 における処理と状態情報の対応	38
8.1	MIPv6 における処理と状態情報の対応	47
8.2	MN 及び HA の構成	51
8.3	通信インターフェース切り替えにかかった時間の平均	53

第1章 序論

本章では，本研究における移動体通信環境の背景について述べ，本研究の目的を示す．また，本研究にて達成する内容を記す．最後に，本論文の構成について述べる．

1.1 はじめに

インターネットを用いた通信では，携帯型計算機の小型化・高性能化や，無線通信デバイスの普及に伴って，移動時の通信の継続性を提供することに注目されて久しい．その結果，Mobile IPv6(MIPv6)[1]やLocation Independent Network for IPv6(LIN6)[2]といった，移動体計算機に対し着信可能性及び移動透過性を提供する移動体通信プロトコルが提案された．現在は，これらのプロトコルを用いた通信で，ネットワーク間移動時における通信再開までの時間の短縮やパケットロス率の低減，セキュリティの向上といった，実用性の向上に関する研究が進められている．

移動体通信の実用化の向上を図る研究として，Fast MIPv6(FMIPv6)[3]やHierarchical MIPv6[4]のようなMIPv6の拡張を行った技術が提案されている．これらの技術は，ネットワーク間の移動を対象にした技術であるが，移動体通信環境では無線が途切れる等の原因で，通信が遮断されてしまう場合も多くある．よって，固定計算機間の通信を対象としてきたTCP等の技術も，移動体通信に適したものをを用いる必要がある．移動体通信を想定したTCPの技術としては，Mobile TCP[5]やTCP Westwood[6]等がある．

移動体通信の効率化に関しては，これらの技術以外にも様々な研究が既になされている．しかし，移動体通信環境で想定される全ての状況で，最適な通信を実現できるプロトコルは未だ存在しない．移動体通信環境では，様々な通信形態が想定されるため，状況に応じて利用する技術を組み合わせて利用する必要がある．

例えば，FMIPv6のみ利用できるネットワークでは，FMIPv6を用いてネットワーク間の移動を行い，HMIPv6のみ利用できるネットワークでは，HMIPv6を用いてネットワーク間の移動を行うことが可能となる．また，その際に利用するTCPの実装を切り替えることも可能となる．

このような通信の最適化は，自動車のような広範囲を高速で移動する移動体計算機に対して，インターネットを利用したサービスの提供を行おうとした場合などに必要となってくる．自動車がインターネットを用いて通信を行う場合，高速に移動を行うことから，頻繁にネットワーク間の移動が発生することが予想される．また，同時に広範囲を移動するため，複数の通信デバイスを用いて通信を継続させる必要がある．よって，自動車に対してサービスを提供するためには，頻繁に発生するネットワーク間の

移動にて、少しでも早く通信を再開させる必要がある。さらに、複数の通信デバイスを用いる場合は、そのデバイスの特性に合った TCP 等の通信プロトコルを選択し、パケットの制御を行う必要がある。

本研究により、移動体計算機上で利用可能な通信技術を、通信状況に応じて動的に切り替えながら通信を行うことが可能となる。移動体計算機は、通信内容に応じて適切な通信環境を実現できるようになり、様々なサービスを利用する際に通信品質を維持することが可能となる。

1.2 本研究の目的

移動体通信における現状は、特定の問題を解決するために様々な技術が存在している状態である。FMIPv6 や HMIPv6、そして Mobile TCP 等の技術は限られた環境でのみ、有効な技術である。

このような技術は、移動体通信における問題を解決するために必要だが、単一の技術で全ての問題を解決するものは存在しない。また、全ての問題を解決し得る移動体通信技術の開発は、移動体通信環境では多様な状況が想定されるため非常に困難である。

これらの問題を解決するために、本研究では既存の通信技術を組み合わせるための枠組みの提案を行う。本研究により、移動体計算機の用途、及び通信状況に応じた通信技術の選択、利用を行うことが可能となる。

1.3 本論文の位置付け

本研究の目的が、移動体計算機の通信状態・環境に適した通信技術へ動的に選択・切り替えを行い、移動体計算機の用途に合った通信環境を提供することであることは先に述べた。通信技術の動的選択・及び切り替えは利用者に対して透過的に行われることが望ましい。

つまり、利用者は通信に用いられている技術に関する知識を一切持たずとも、常に状況に適した技術が利用されている状態である。しかし、移動体通信環境では多様な環境が予想されるため、移動体計算機の用途及び通信状況に適した通信技術を特定することは非常に困難である。例えば、移動体計算機が所有している通信デバイスが IEEE802.3 Ethernet 通信デバイスと IEEE802.11b 無線通信デバイス [7] のみの場合と、IEEE802.11b 無線通信デバイスと携帯電話のみの場合では、利用しなければならない通信技術は異なってくる。

これは、複数通信インターフェースの同時利用 [8] などを想定した場合に利用できる通信インターフェースの種類によって利用すべき通信技術は違ってくるためだ。また、利用者によって通信コストなどの関係から、利用したい技術の種類も異なってくる。

本論文では、利用者に対して透過的に通信技術の動的選択・切り替えを実現するための技術の基盤として、利用者によって具体的に指定された内容に沿って通信技術の動的切り替えを実現するための枠組みを提案する。これにより、移動体計算機の通信状態に

基づいて，利用者は意図した通りに通信に用いる技術を動的に切り替えることが可能となる．

以後，目的を達成するためには，移動体計算機の用途や通信環境に応じて必要となる技術の整理及び利用者の意志を反映できるようにするためのユーザーインターフェースの構築が必要となる．

1.4 本論文の構成

本論文は以下のように構成される．第2章で，本研究で想定する移動体通信環境について，インターネット ITS システムを例に説明する．第3章では，第2章で述べた環境を実現しようとした際に解決しなければならない問題について述べる．第4章では，本研究にて用いる機構及び関連する研究について解説する．第5章で，通信状態に応じて適切な通信設定を選択し，設定を実行させるモデルである動的移動処理選択支援モデルについて述べる．第6章で，モデルを用いた動的移動処理選択支援機構の設計について述べる．第7章で，本システムの具体的な実装方法について述べる．第8章で，実装を行った動的移動処理選択支援機構の定性評価を行う．最後に第9章で，まとめと今後の課題についての考察を行う．

第2章 移動体通信環境への要求

本章では，インターネット ITS[9] を例に，移動体通信環境を利用して提供されようとしているサービスをいくつか挙げ，そこからインターネットを利用することの利点について考察を行う．また，この利点と提供されようとしているサービスから，移動体通信環境が満たさなければならない条件を帰納法の要領で導き出し，理想的な移動体通信環境に関する考察を行う．最後に，現状の移動体通信環境で既の実現されている事柄と満たさなければならない条件を比較し，解決しなければならない問題点を整理する．

2.1 対象とする移動体通信環境

本節では，移動体通信環境を利用したサービスの具体例としてインターネット ITS システムを紹介する．また，そのサービスを基に移動体通信環境が満たさなければならない条件に関する考察を行う．

2.1.1 インターネット ITS の目指す環境

これまでの通信環境では，固定計算機環境で提供されているサービスを出先でも利用することしか想定されていなかった．一方で，インターネットは様々な機器やサービス上にて，共通の通信基盤として利用でき，新たなサービスの開発や提供を低コストで実現することができる可能性を秘めている．

このような背景から，移動体通信環境の可能性を模索するために，インターネット ITS プロジェクトやインターネット・トレイン プロジェクト [10] が実施されている．これらのプロジェクトは，移動体通信環境の特色を用いた様々なサービスを提供する試みである．インターネット ITS は，自動車をインターネットに接続し，渋滞等の交通に関する諸問題を緩和するためのシステム [11] を提供することを目的に開発が進められている．インターネット・トレインは，電車をインターネットに接続させ，乗員に対してインターネット接続性を提供することや各電車の正確な位置を公開することによる到着時刻通知システム等の開発が行われている．

2.1.2 移動体通信環境におけるサービス例

本節では、インターネット ITS が提供するサービスに対して大まかに述べ、インターネットを用いることの利点に関して解説する。

既存の交通システムにおける問題点は、移動中の自動車に対して渋滞情報などの情報を提供するシステムが限定されており、導入コストが高いなどの問題から利用車が限られていることである。

例えば、ITS のサービスとして、道路交通情報システム (VICS) [12] や自動料金収集システム (ETC) が有名である。日本中の幹線道路には、VICS サービスを提供するためのインフラが整備されている。当然、VICS を利用するためには VICS 専用の車載機を搭載する必要がある。一方、ETC サービスを提供するためにも、日本中の高速道路に専用インフラが整備されつつある。こちらも、サービスを利用するためには専用の車載機を搭載する必要がある。

このように、現在の ITS ではサービスを立ち上げるために、サービス毎に専用のインフラと車載機を必要としている。そのため、導入コストが非常に高くなってしまい、利用者やサービス事業者に負担がかかっている。

インターネット ITS は、インターネットを共通の通信インフラとして用いることでこの問題を解決する試みである。特定のサービスに特化することなく、あらゆるサービスはインターネット経由で提供することが可能となる。

以下にインターネット ITS を用いたサービスの例を示す。

- 自動車からの情報発信：
自動車の速度情報や位置情報など、自動車から取得できる情報は様々である。インターネット ITS では、自動車の情報をインターネットを経由して取得することによって、各自動車をセンサーとして扱う。これにより即時性の高い渋滞情報や各車両の故障検知による迅速な対応の提供が可能となる。各車両から取得したデータを加工して渋滞情報を生成することによって、インターネットに接続した自動車が走っている道路であればどの道でも交通情報が取得できるようになる。
- 自動車への情報提供：
自動車より取得した情報を基に、目的地への経路上の渋滞情報や降水量情報などを提供する。カーナビと連携することによって渋滞を考慮した最適な経路の選択が可能となる。他には、駐車場情報やレストラン情報など自動車での移動を支援する情報の配信が可能となる。
- 車車間通信サービスの提供：
自動車で移動中に近隣の自動車と通信を行うサービス。近隣の自動車の状態情報を車車間で交換することで交通の円滑化や、安全運転支援等を行うことを目的とする。例えば、周辺の自動車の車両情報を取得することにより、交差点における直進車と右折車の衝突事故を回避することができる。

自動車をインターネットに接続することで、上記のようなサービスを低コストで導入することが可能になる。このことから、今後自動車へのサービス提供にインターネットの利用が一般的になると考える。

2.2 移動体通信環境への要求

インターネットを通信基盤として用いることで特定のシステムに依存しなければならない問題を解決することが可能になる。言い換えると、既存のシステムが独自のシステムで解決していた内容を、インターネットを用いて解決しなければならないことを意味する。VICSの代わりに交通情報を提供することを想定した場合、インターネットを用いて、高速に移動している自動車に対し、情報を確実に配信できる必要がある。つまり、自動車をインターネットに接続させることを考えた場合、自動車の特性である広域を、高速で移動することを十分考慮した上で上記のようなシステムを構築しなければならない。

第2.1節で述べた3つのサービスを自動車に対して提供することを考えた場合、移動体通信環境は自動車に対して以下のような条件を満たさなければ実用的なサービスは困難である。

- 一意のIPアドレスを用いて到達可能であること：
既存のインターネットアーキテクチャでは、ネットワーク間を移動することに伴い、インターネット上における自動車の識別子であるIPアドレスが変化する。このIPアドレスの変化により、通信相手（サービスを提供するサーバー等）は自動車の識別子を失うことになり、通信を継続できなくなる。よって、自動車を常に固定のグローバルIPアドレスで識別可能にすることで、ネットワーク間の移動に関係なく通信相手からの情報を着信できる必要がある。
- 移動してもセッションが維持可能であること：
既存のインターネットアーキテクチャでは、ネットワーク間の移動に伴いIPアドレスが変化してしまう問題があることは先に述べた。IPアドレスの変化は、通信セッションの遮断を伴うため、移動後の通信を再開することができなくなる。通信相手から自動車へサービスを提供している際に自動車が所属ネットワークを変化させた場合、通信セッションが遮断され、途中まで送信完了していた内容を再び始めから送信しなおさなければならない。このような問題を解決するために、自動車と通信相手間の通信セッションを維持可能にすることで、通信の不効率を防ぐ必要がある。
- 常時インターネットに接続できること：
自動車は広域を、高速に移動する交通手段である。自動車の特性上、ネットワーク間の移動が頻繁に発生することに関しては既に述べた。自動車のように広域を移動する場合、単一の通信インターフェースで全ての通信をまかなうことができない。携帯電話をもちいてインターネットに接続していた場合でも、電波の入ら

ない場所への移動は必ずある。インターネットへの接続性がない状態では、情報交換が行えず、サービスを楽しむことができない。よって、より広範囲に渡ってインターネットに接続できる必要がある。

- ネットワーク間の移動が効率的に行えること：
インターネットを用いたネットワーク間の移動には、移動先ネットワークの検索や、移動先ネットワークでの新たな IP アドレスの取得など、通信再開までに時間を要する処理が多々ある。インターネット ITS を利用したサービスには、交通情報の他に駐車場方法の提供や動画音声付オンラインミーティングなどのリアルタイム性を必要とするアプリケーションも考えられている。このようなアプリケーションを想定した場合、移動処理にかかる時間の短縮や、移動中に生じる情報の欠落を軽減しなければならない。
- 多様な設定に対応できること：
インターネット ITS に限らず、インターネットを共通の通信基盤として利用することによって、通信デバイスや計算機、そしてアプリケーション等、様々なものを組み合わせて通信を行うことが可能となる。利用者の都合やシステムの用途によって、移動体を形成する要素技術も異なってくる。移動体が異なることによって、移動体に対して必要とされる通信品質も異なってくるため、それぞれの需要に対応できなければならない。

インターネット ITS システムは、既存の移動体通信環境の要素技術すべてを組み合わせるシステムである。インターネット ITS が必要とする上記のような条件を満たすことによって、理想的な移動体通信環境を提供することが可能となる。

2.3 想定される移動体通信環境

インターネット自動車システムは、現在考え得るすべての通信サービスを自動車及び搭乗者に対して提供することを想定して設計されており、第 2.2 節で述べた条件を全て満たすことによって、あらゆる移動体計算機上でも効率的な移動体通信を実現することが可能としている。

上記の条件を満たす移動体通信環境の具体例を図 2.1 と図 2.2 を用いて説明する。移動体通信環境において効率的な通信を行うことを考えた場合、図 2.1 のように状況に応じて最適な通信デバイスを利用しながら通信を継続したい。

これにより、移動体計算機は必要な帯域に応じて利用する通信デバイスを選択したり、利用可能な通信デバイスの中で優先順位を決めて動的に切り替えながら通信を継続させたりすることが可能となる。

例えば、図中 A の状態では、移動体計算機は広帯域かつ低遅延な無線通信デバイス IEEE 802.11b の利用を優先するように設定されているため、それを用いた通信を行う。また、無線基地局 (Access Point) の切り替えも電波状況等を見ながら電波が届かなくなる前に次の基地局での設定に変更し、通信が途切れるのを防ぐ。B の状態では、IEEE

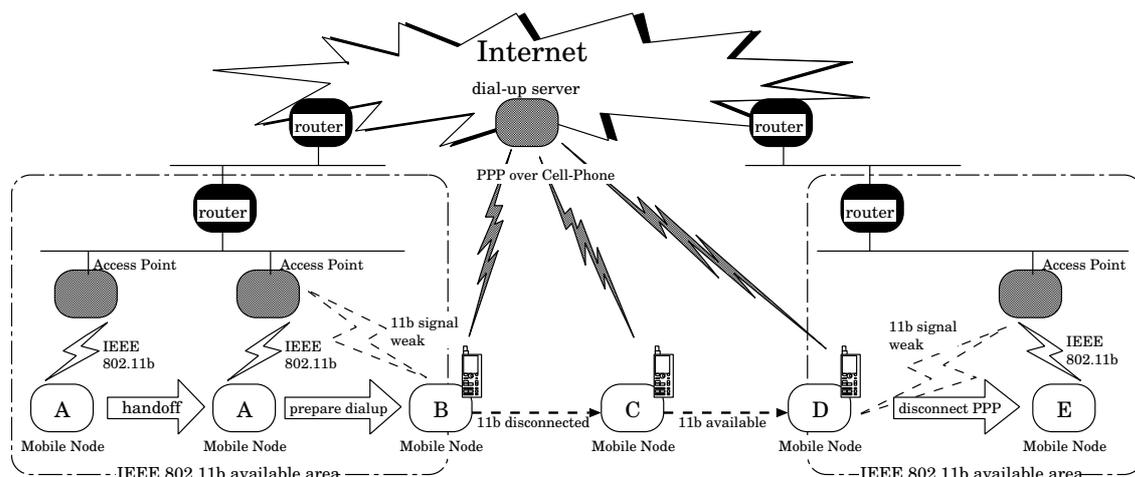


図 2.1: 想定する移動体通信環境：移動体計算機は利用する通信インターフェースを状況に応じて切り替えながら通信を継続させる

802.11b の電波が弱くなってた状態だが、次の基地局を発見することができなかった場合である。移動体計算機は IEEE 802.11b での通信が途切れる前に携帯電話を用いた dial-up 接続に切り替え IEEE 802.11b が利用できない状態 C に備える。再び IEEE 802.11b の電波を受信した状態である D では、移動体計算機は IEEE 802.11b を用いた通信の準備を行う。そして E の状態では通信の準備は完了しており、IEEE 802.11b での通信に切り替えた後に携帯電話での dial-up 接続を切断する。

このようにして、移動体計算機は通信状況に応じた通信デバイスの選択及び利用を実現し、効率的な通信を行うことができる。

また、通信デバイスの切り替えだけでなく、利用する通信技術の切り替えも同時に行いたい。これにより、図 2.2 のような、ネットワーク毎に利用可能な移動体通信支援技術を判別し、移動体計算機の環境に応じた技術に切り替えながら通信を継続させることが可能となる。

移動体計算機は、FMIPv6 が利用可能な A のネットワークに属しているため、FMIPv6 を用いて移動処理を行う。また、A から B のネットワークへ移動する際に HMIPv6 のみ利用可能であることを検地し、HMIPv6 を用いた移動処理を行うように移動体通信支援技術の切り替えを行う。しかし、A から B へ移動する際には移動体通信支援技術の切り替えを伴うため、移動処理にかかる時間は通常の MIPv6 と同等またはそれ以上となる。移動体計算機は B のネットワーク内で移動を行っている際には HMIPv6 を用いて移動処理を行うことができるため通信再開までの時間を短縮することが可能となる。B から C のネットワークへ移動する場合、移動体計算機は IEEE 802.11b 無線通信デバイスを利用することができず携帯電話を用いた dial-up 接続に切り替える。この場合、MIPv6 以外のプロトコルを利用することができないためあらかじめ携帯電話での通信を確立してから MIPv6 による移動処理を行う。

このように、ネットワーク間の移動の際に利用可能な移動体通信支援技術の検索及

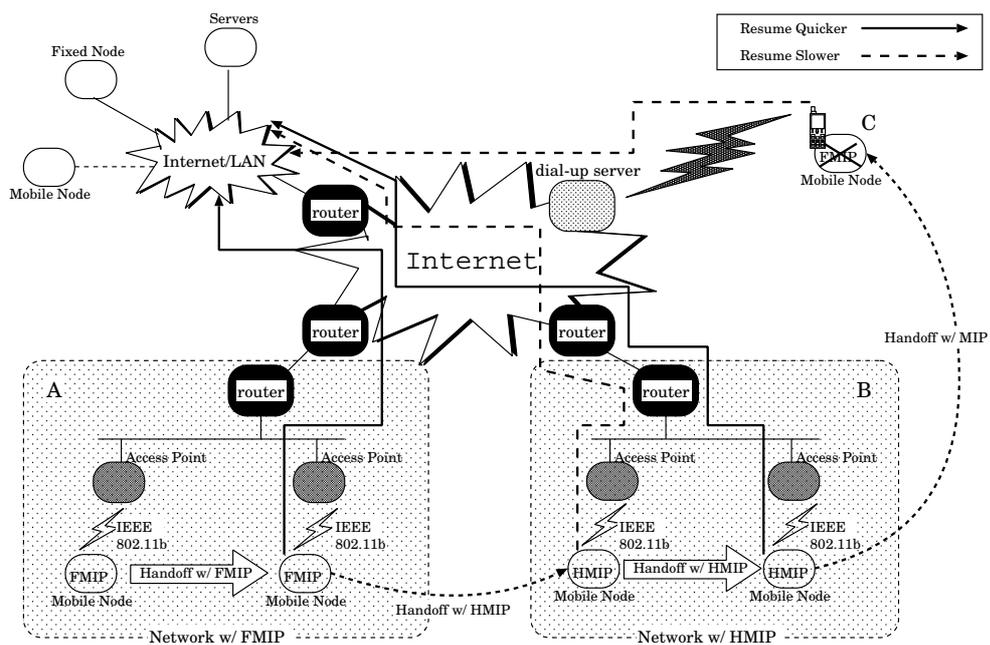


図 2.2: 想定する移動体通信環境：移動体計算機は利用する通信技術を状況に応じて切り替えながら通信を継続させる

び選択を自動化することによって効率的な移動体通信が可能となる。

以上で本研究が想定する移動体通信環境について図 2.1 と図 2.2 を用いて説明した。理想的な移動体通信として、通信デバイス及び通信技術両方を移動体計算機の通信状況に応じて動的に切り替えながら通信を行いたい。

第3章 既存の移動体通信環境における 問題点と解決アプローチ

本章では、移動体通信環境にて通信の用途及び状態に適した技術を動的に選択・切り替えながら利用する際に問題となる点について議論する。次に、これらの問題点を解決する為に本研究で用いる解決アプローチに関して述べる。最後に、解決する際に考慮しなければならない事柄について考察を行う。

3.1 移動体通信における問題点

急速に成長している計算機環境では、常に目標とされていることとして、人々の現実生活において計算機の存在を意識することなくその恩恵を享受することが可能な世界の実現がある。これは、ユビキタス・コンピューティングという概念で知られており、その実現には移動体通信環境の充実は必須である。このような背景のもとに、現在移動体通信環境における様々な需要に応えるべく多様な技術の開発が行われている。

その結果移動体通信環境では、固定計算機環境と違い、理想とされる通信形態が計算機の用途や利用可能な技術ごとに大幅に異なることになった。また、利用者によっても理想的な通信の定義が異なることから、状況に応じた通信技術の選択の基準を定めるのは非常に困難である。

本節では、移動体通信環境において利用する通信技術の選択に影響する項目について整理する。

3.1.1 移動体通信環境の多様化

まず始めに、移動体通信環境の多様化について述べる。移動体通信環境の多様化は、通信デバイスの多様化や計算機の用途の多様化、さらに移動体通信技術の多様化などが考えられる。

通信デバイス

現在販売されているノート型計算機には、IEEE 802.11b 無線通信デバイス及び IEEE 802.3 100Mbps Ethernet 通信デバイス等、複数の通信デバイスが装備されていることが多い。そして、利用者は自宅など有線通信デバイスが利用可能な環境ではそちらを

利用し，外出時は無線通信デバイスを利用するといった様に状況に応じて切り替えながら通信を行うことが多くなってきた．

今後，移動体通信環境が普及するにつれてさらに多くの通信デバイスが利用可能になる．また，それと共に切り替えの頻度が上がってくることが予想される．

本論文では移動体計算機に対して利用可能な通信デバイスの増加を指して通信デバイスの多様化と呼ぶ．以下に移動体通信環境にて利用される通信デバイスについて例を示す．

- IEEE802.11a 無線通信デバイス
- IEEE802.11b 無線通信デバイス
- IEEE802.3 100Mbps Ethernet 有線通信デバイス
- 携帯電話
- Bluetooth
- 衛星通信

接続される計算機

移動体通信環境では，インターネット ITS システムにて利用される Mobile Router[13]であったり，利用者が携帯しながら利用する PDA のような携帯型計算機であったりと，移動体計算機の種類も多様化してくる．

接続される計算機の種類，及びそれらの用途は，利用する通信デバイスや通信技術に大きく影響する．また，今後ユビキタス・コンピューティングが普及するにつれて計算機の種類や用途は無限に多様化する．

本論文では，このような接続される計算機の種類増加や用途の多様化を，移動体の多様化と呼ぶ．以下に移動体通信環境にて接続されることが想定される移動体について例を示す．

- 自動車 (Mobile Router)
- 電車 (Mobile Router)
- PDA
- ノート型計算機

通信技術

インターネットに接続した計算機が、通信を継続した状態でネットワーク間を移動した場合、既存の固定計算機を対象とした通信技術では様々な問題が生じる。ネットワーク間の移動の際も、通信相手から移動を隠蔽することで、あたかも固定計算機と通信しているように見せる技術として、MIPv6 や LING があることは先に述べた。

移動体通信技術を用いた通信をより効率的にする技術として FMIPv6 や HMIPv6 がある。本論文ではこれらを移動体通信支援技術と定義する。また、移動体通信技術や移動体通信支援技術に対して必要な情報を提供するための技術も重要となる。

移動体通信技術及び移動体通信支援技術のような通信技術に関しても、現在様々な研究が行われており、移動体通信環境における様々な状態に対して有用な通信手法が多々提案されている。

このような移動体通信環境における、通信技術の多様化を指して通信技術の多様化と呼ぶ。以下に移動体通信環境にて用いられる通信技術について例を示す。

移動体通信技術

- MIPv6
- LING

移動体通信支援技術

- FMIPv6
- HMIPv6
- Mobile TCP
- TCP-Westwood

その他の技術

- Geographical Location Information(GPS)[14]
- Candidate Access Router Discovery(CARD)[?]

3.1.2 利用者による差異

移動体通信環境では、利用者ごとに異なる通信への要求によっても利用すべき通信技術であったり通信デバイス等が違ってくる。

利用者の通信への要求により、先に述べた通信技術の組み合わせは、さらに複雑化する。利用者の通信への要求は、統計処理を行うことである程度一般化することが可能であるが、それは本研究では取り扱わない。

利用者の要求は、移動体計算機の用途や環境によって複雑に変化してくる。しかし、利用者が常に意識する移動体通信環境における通信要素は大別することが可能である。以下に利用者によって異なってくる通信要素の例を示す。

- 通信コスト
- 通信速度
- 消費電力
- 処理コスト

3.2 既存の移動体通信における問題点

既存の移動体通信における問題点は、多様な移動体通信環境における通信への要求を全て満たす通信技術が存在しないため、個々の問題に対する解決手法が提案されているだけになっていることである。また、これらの技術は動的に切り替えながら通信することを前提に設計及び実装されていない為、その性能を活かせる場所が限定されてしまう。

既存の移動体通信技術は、インターネットへの接続性を保持したままネットワーク間の移動を行う移動体計算機に対して、以下の条件に対する解決を提供している。

1. 一意の IP アドレスを用いて到達可能である
2. 移動してもセッションが維持可能である
3. 常時インターネットに接続できる
4. ネットワーク間の移動が効率的に行える

まず、1と2の2つの条件は、MIPv6を用いることで解決することが可能である。次に3の条件は、MIPv6を用いた複数インターフェース支援機構 [8] を用い、利用可能な通信インターフェースを切り替えながら通信の継続を行うことで解決することができる。4の条件に関しては、現在 FMIPv6 や HMIPv6 を用いることで解決することができる。

しかし、既存の移動体通信技術は第 2.2 で述べた項目の一つである、多様な設定に対応できることに関しては解決手法が提案されていない。

これは、前章までで述べてきたように、将来の移動体通信環境では様々な要素が多様化して来ることによって、想定しなければならない移動体通信環境が複雑化してくるからである。

既存の移動体通信技術及び移動体通信支援技術は、ある特定の環境において最適な通信を実現することができるが、想定される環境全てにおいて効率的な通信を実現する技術は現在存在しない。

また，一意に最適な通信の定義を定めることができないため，単一の移動体通信技術及び移動体通信支援技術でそれを実現することは不可能である．

これらの理由により，既存の移動体通信環境では，あらゆる状況にも適した通信環境を提供することができない．また，利用者の意志を通信全般に反映させることも考えられていないため，利用者は自分の理想とする通信環境を構築することができない．

3.3 問題点解決へのアプローチ

多様な通信要素を持っている移動体通信環境において，単一の通信プロトコル構成では効率的な通信を実現するのは困難であることは先に述べた．移動体計算機が持っている多様な通信要素を用い，通信状態に適した通信を実現するためには，適宜通信プロトコル構成を調節できる必要がある．

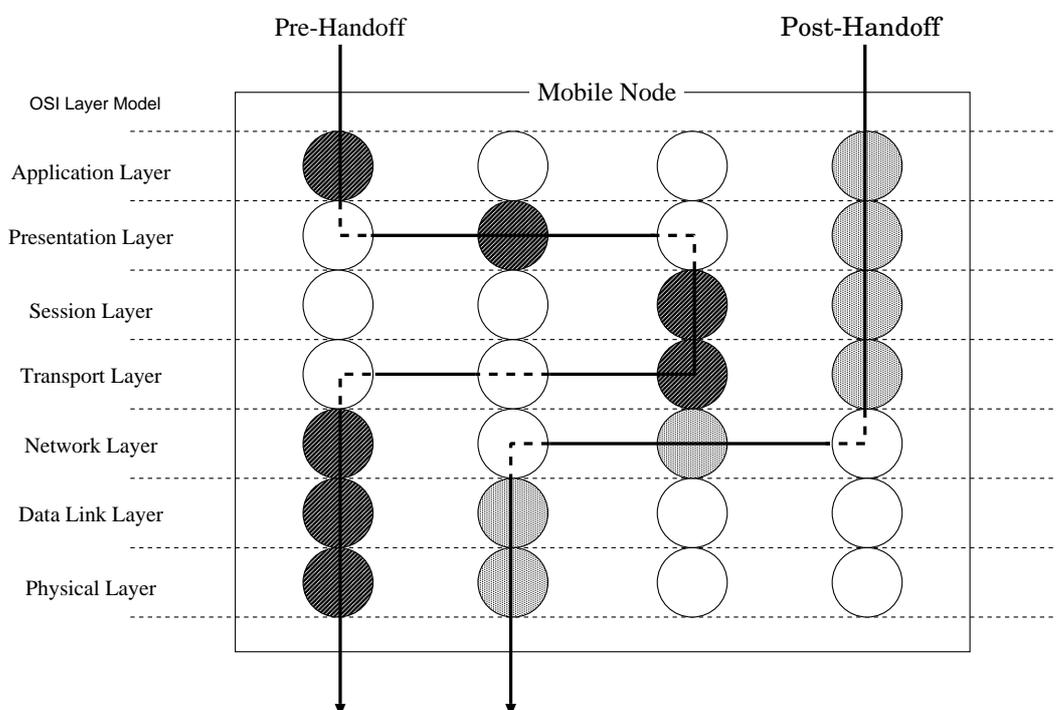


図 3.1: 例-移動体計算機が利用する通信プロトコル構成の変更

本論文における通信プロトコル構成とは，移動体計算機が所持する通信プロトコルの組み合わせを指す．図 3.1に通信状態によって，通信プロトコル構成を調節している様子を示す．図中の球体は，移動体計算機が OSI 階層モデルにおける各層にて所有している通信プロトコルを示している．矢印は，移動体通信計算機が通信に利用する通信プロトコル構成を示す．

通信プロトコル構成の変更は，例えば，移動体計算機が IEEE802.11b 無線通信デバイスを用いて通信を行っていた状態で，FMIPv6 が利用できるネットワークから HMIPv6

が利用できるネットワークへ移動した時などで必要となる．移動前の状態では，IP 層のプロトコルは FMIPv6 を用いて通信を行った方が効率良く移動することができる．しかし，移動先のネットワークで FMIPv6 を利用して通信を継続させようとしても，HMIPv6 しか利用できないため，通常の MIPv6 での通信となる．HMIPv6 が利用可能である場合，IP 層のプロトコルを HMIPv6 での通信に切り替えた方が効率良く移動を行うことができる．

このように，移動体計算機の状態によって利用する通信プロトコルを切り替えることができなければ，効率的な通信を行うことはできない．よって，本研究では移動体計算機上で利用可能な通信プロトコルの構成を状況に応じて調節し，効率的な通信を実現するアプローチを取る．

第4章 関連研究

本章では，本研究に関連する研究について述べる．MIB Socket[15] は，ネットワークエンティティの動的な状態変化に対応する管理機構を提供する．本研究では，ノードの状態情報を収集するために利用する．Ensemble[16] は，アプリケーションの要求に応じて複数のプロトコルを選択利用するための枠組である．各プロトコルをモジュール化し，それらの組み合わせを定義し，必要に応じて切替えながら利用する．この点で本研究と同じアプローチだが，具体的なシステム設計に関して不明確であり，そのまま利用することはできない．

4.1 MIBsocket

MIBsocket は，ネットワークエンティティの動的な状態変化に対応する管理機構を提供する．アプリケーションは，ソケットを用いてカーネルとメッセージを交換することでネットワークエンティティにアクセスする．カーネルは移動等によって起きるネットワークエンティティの動的な状態変化を検知し，ソケットを経由してアプリケーションに通知する．本研究では，ノードの状態情報を収集するために利用する．

MIBsocket では，ネットワークエンティティの状態管理を実現するために，以下の5つの機能を提供している．

1. ネットワークエンティティの状態取得機能
2. ネットワークエンティティの設定機能
3. ネットワークエンティティの状態変化通知機能
4. ネットワークエンティティへのアクセス制御機能
5. 必要なネットワークエンティティの情報フィルタリング機能．

MIBsocket のシステム概要を図 4.1 に示す．

ネットワークソフトウェアはネットワークエンティティの操作を，MIBsocket への要求メッセージを用いて行う．MIBsocket は要求に対する回答として応答メッセージをネットワークソフトウェアに返す．もし要求に対する処理が失敗した場合はエラー番号が返される．また，ネットワークエンティティの状態変化の通知はトラップメッセージとして MIBsocket からネットワークソフトウェアに通知される．これらの処理を1つの MIBsocket で行うために，設定 (図中, Set)，取得 (Get)，トラップ (Trap) の

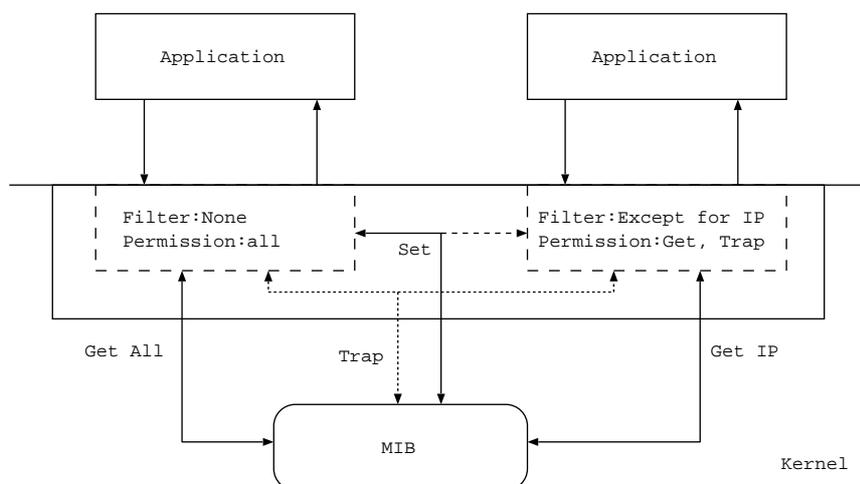


図 4.1: MIBsocket 概要：ネットワーク情報は MIBsocket を経由してカーネルとアプリケーション間で共有している

3つのモードが定義されている。また、フィルタを設定することにより、必要な情報のみを受け取ることができる。例えば、IP アドレスの変化情報だけを取得したい場合は、そのように設定することができる。許可も同様で、ネットワークソフトウェアのネットワークエンティティへのアクセス制御を行うことができる。複数の MIBsocket が同時にネットワークエンティティの設定を行った時に起こる競合を、許可の機能を用いて回避することができる。例えば、あるアプリケーションに対しては Trap の動作だけを許可するという設定ができる。

4.2 The Ensemble System

Ensemble System は、アプリケーションの要求に応じて複数のプロトコルを取捨選択し、それらを組み合わせて利用するための枠組のひとつである。Ensemble では、レゴ (TM) のブロックを組み合わせ、ひとつのオブジェクトを組み立てるように、様々なプロトコルをそれぞれモジュール化し、モジュールを組み合わせることである機能を実現する。

この種のシステムでは、性能と柔軟性の両立が問題となる。Ensemble では、動的に最適化されたプロトコルスタックを生成し、そのスタックを用いてイベントを処理することでこの問題を解決している。

Ensemble システムの概要を図 4.2 に示す。

Original stack 中のプロトコルモジュールは、モジュールの実行履歴にしたがって Optimized stack に組み込まれる。再び、この実行履歴を満たすイベントが実行された場合、そのイベントは Trace Conditions で傍受され、極度に最適化された Trace Handler を経由して実行される。図 4.2 において、Original stack 中の色のついた部分が利用されるモジュールであり、図中の点線矢印の順番に処理が実行されていることが見て取

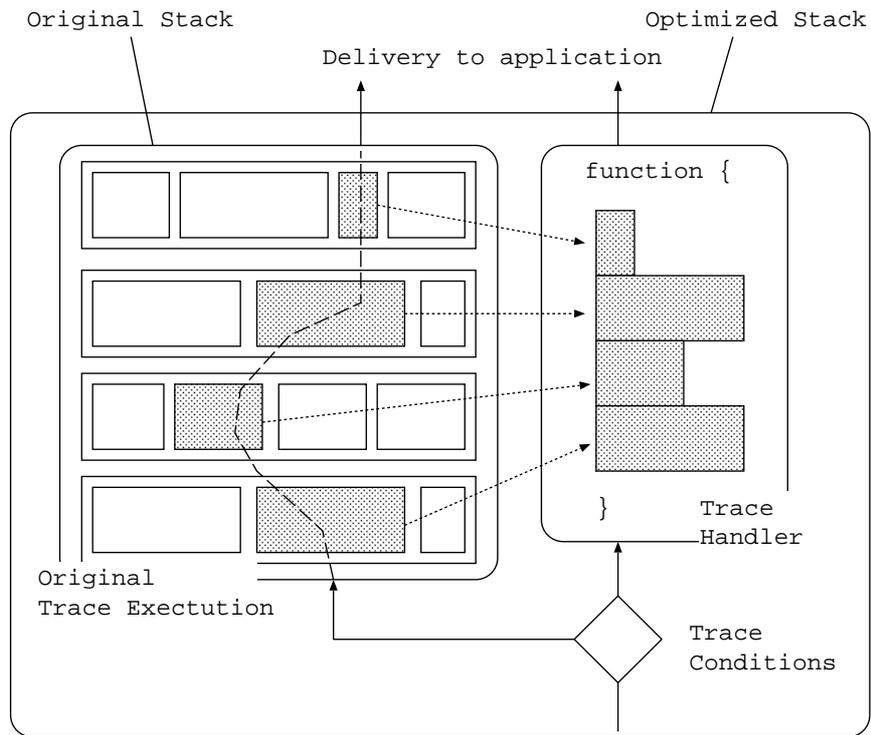


図 4.2: Ensemble システム概要

れる．これによって Optimized stack が生成され，それ以降同じイベントを処理するときにはこの Optimized stack を直接用いることで処理速度が上がるなどのメリットがある．しかし，Ensemble の参照システムは，ML 言語を用いて設計，実装されているため，既存の protocol stack をこのシステムに合わせる形で作り直す必要がある．また，C 言語で実装されていないのも問題として挙げられる．

第5章 動的移動処理選択支援モデル

これまでの章を通じて、今後移動体通信環境で理想的な通信を実現しようとした場合、様々な問題があることが分かった。これは、多様化する通信デバイスや計算機の処理に伴い、移動体計算機が管理しなければならない項目も多様化してくるからである。また、移動体計算機の利用者や用途によっても理想的な移動体通信の定義は変わり、一意に最適な計算機の処理を決めることはできない。よって、単一の通信プロトコルの構成を常時用いる既存のモデルでは理想的な通信の実現は非常に困難である。

本章ではまず、移動体通信環境において、最適な通信を実現するにあたって満たさなければならない要件を整理し、それぞれについて解説する。そして、既存の移動体通信技術及び移動体通信支援技術など、通信の管理を行う処理を組み合わせることで上記の要件を満たす手法を検討する。その後、本論文における解決手法として動的移動処理選択支援モデルの提案を行う。

5.1 問題点の解決に必要な機能要件

第3.3節で述べたアプローチを用い、既存の問題点を解決するには、以下のような機能が必要となる。

- 移動体計算機が持つ情報の監視：
移動体計算機が通信に利用する通信技術の構成を決定するためには、まず、移動体計算機の通信状態を正確に把握する必要がある。移動体計算機が持つ情報を可能な限り取得することでより正確に移動体計算機の状態を把握できる。
- 通信状態に基づいた最適な設定の判断：
移動体計算機は、取得した情報から移動体計算機の通信状態を把握し、効率的な通信を行うために必要な通信技術の設定を判断する。通信技術の設定の調節は、同時に複数の通信技術を変更する場合と、特定の通信技術のみを変更する場合と様々である。
- 選択された通信技術の設定への切り替え及び処理の実行：
状態情報から判断された通信技術の設定を利用するように切り替えを行う必要がある。

以下にそれぞれに関する詳細を述べる。

5.1.1 移動体計算機が持つ情報の監視

移動体計算機が持つ情報は，利用している通信デバイスや通信プロトコル，アプリケーションによって異なる．移動体計算機が持つ情報の例として，以下のようなものが挙げられる．

- インターフェースの情報：
リンク情報や無線電波強度など，移動体計算機が所有しているインターフェースが通信を行える状態にあるかを示す情報．
- IP アドレス：
移動体計算機をインターネット上で一意に識別するための識別子．移動体計算機は，ネットワーク間の移動ごとに移動先ネットワークから新たな IP アドレスを取得する．
- ユーザーランドからの情報：
Geographical Positioning System(GPS)[14] の情報やストリーミングアプリケーションの状態情報など，カーネルから取得することができない情報で，移動体計算機の状態を示す情報．

他にも，移動体計算機の状態情報だけでなく，通信相手の情報などをネットワーク越しに取得し，通信技術の選択に利用することも考えられる．

5.1.2 通信状態に基づいた最適な通信環境設定の判断

通信状態に基づいて，最適な判断を行う際に最も重要となるのは，利用者の意志を実際の通信に反映させることである．

これを実現するには以下の二つの機能が実現されていなければならない．

- 利用者による判断基準の設定：
理想的な移動体通信の定義は，移動体計算機的环境や用途によって異なる．そのため一意に決めることができない．利用者が判断基準を設定できるようにすることで，多様な適応処理に柔軟に対応できるようになる．
- 適応処理を判断する機能：
状態情報と利用者が設定した判断基準を基に次に利用する通信環境設定を返す機能．利用者の意志を実際の通信に反映させる部分である．

利用者による判断基準の設定

利用者によって設定可能でなければならない項目は，特定の状態情報と関連付けられる通信技術の構成や新たなユーザーランドの処理まで多様である．本論文では，こ

のような状態情報と関連付けられる通信技術の構成の変更や，ユーザーランドの処理を実行させることを適応処理と呼ぶ．

利用者が判断基準を設定する際に必要となる項目を以下に述べる．

- 情報の特徴に基づいた抽象化：
状況に応じて実行する処理を切り替えながら通信を行うモデルでは，移動体計算機から取得する情報が環境に関わらず同じ意味を持つことを前提としなければならない．デバイスやプロトコルから取得できる情報はプリミティブであり，そのままでは意味を成さない．しかし，既存の移動体通信技術や移動体通信支援技術などの適応処理は，全て情報が持つ意味を基に設計されている為，プリミティブな情報では対応付けを行えない．よって，利用者の意思を忠実に通信に反映するには，利用者の意図に正確に，情報とそれが持つ意味の関連付けが行えるように抽象化できていなければならない．
情報の抽象化に関しては第 6.5.1 節にて詳しく述べる．
- 利用者による新たな状態情報の定義：
適応処理は通信技術の設定の切り替えからユーザーランドアプリケーションの処理の実行まで多岐に渡る．よって，利用者が理想とする通信を実現するには，利用者が必要に応じて自由に用いる情報を定義できる必要がある．利用者が定義する情報に関しては本節にて例を示す．
- 設定書式における一貫性：
効率的な通信を行う際に利用しなければならない適応処理の数は，移動体通信環境が多様化すると共に，増加することが予想される．そのため，利用者は数多い適応処理の関連付けを容易に行えなければ，移動体通信において最適な処理の設定が行えない．この問題は，設定書式に一貫性を持たせることで緩和することが可能である．これは，設定書式に一貫性があることで，異なった適応処理の設定も容易に行えるようになるためである．

以下に利用者によって定義することができる状態情報の例を示す．

- 通信性能：
通信デバイスで利用されている帯域，通信相手との RTT，さらに接続を保持できる時間及び確立を基に適応処理を実行するタイミングを決断する方法．それぞれの通信デバイスに対して状態の基準値を設定し，それに基づいて通信デバイスに重み付けを行う．この重みを監視し，移動体計算機が現在利用している通信デバイスが他の利用可能通信デバイスと比較して通信品質が劣るとみなされた場合，最も通信品質が高いと思われる通信デバイスへ切り替える．
- 時間：
移動体計算機の通信状態などとは一切関係なく，あらかじめ設定された特定の時間に，特定の処理を実行させる方法．定期的に移動体計算機の状態を確認し，得

られた値をもとに適応処理を判断させることなどが可能となる。直接適応処理とは関係ないが、定期的に状態を確認しなければならないような状況では非常に有用である。

- 通信コスト：
移動体計算機の用途及び利用者によって一定の金額以上支払いたくない時に利用することができる。通信デバイスの利用額があらかじめ設定してあった金額に達した時に他の移動体通信技術及び移動体通信支援技術に切り替えを行う等の設定が可能となる。

このように、適応処理の判断を行う部分は利用者によって設定された関連付けに基づいて通信環境の変更を可能にするための要となる部分である。よって、利用者の移動体通信への要求を容易に反映できるような理解し易い形で設定を行えなければならない。

適応処理を判断する機能

利用者による判断基準を設定するのに必要な機能に関しては前に述べた。本項では、利用者によって設定された判断基準と移動体計算機の状態情報を基に利用される適応処理を判断する機能について述べる。

適応処理を判断する機能は、以下の2つの条件を満たしている必要がある。

- 取得した状態情報を基に利用者が設定した判断基準の検索を行い、必要な適応処理を知ることができること
- 検索によって得た適応処理が設定された内容を正確に反映できていること

この2つが満たされていれば、移動体計算機は利用者の意図に基づいた通信の管理が行えるようになる。

また、適応処理の判断に要する時間は短い程移動体通信において効率の良い移動を実現することができる。なぜなら、これら一連の処理に要する時間が長引いた場合、高速で移動する移動体計算機において最適な通信環境の設定などが間に合わない可能性があるからだ。

5.1.3 選択された通信環境設定の実行

最後に、選択された適応処理を実際の通信に反映させる部分である。

選択された適応処理へ切り替えを行う場合や、現在利用されている通信環境設定の処理を実行する場合、以下の条件が満たされている必要がある。

- 適応処理は各機能ごとにモジュール化されていること：
 移動体通信環境において，通信技術の構成の変更を行うだけでなく，通信技術の特定の処理のみを迅速に実行したい場合などである．
 例えば，FMIPv6 を用いた通信を行っていた場合に，実際に移動する前に移動に関する情報を取得した時などである．この場合，移動体計算機はあらかじめ移動先ネットワークに対して Fast Binding Update(FBU) を送信することで移動にかかるオーバーヘッドを軽減することができる．
- 適応処理呼び出し部と適応処理間のインターフェースが提供されていること：
 適応処理となる通信技術やユーザーランドアプリケーションは多種多様であり，実装の仕様は大きく異なる．外部から各機能呼び出すためのインターフェースが既に実装されているものから，一切外部からの呼び出しに対応していないものまである．よって，既に外部からの呼び出し用のインターフェースを持っている適応処理に関してはそのインターフェースをそのまま利用するように呼び出し側で対応し，インターフェースがない場合は適応処理と呼び出し側両方に手を加える必要がある．

図 5.1 に適応処理の呼び出しの概要を示す．図中の適応処理の切り替えを実現する機構には，適応処理の呼び出しや処理の実行命令を送信するためのインターフェースがある．また，通信技術側にも，呼び出しや処理の実行命令を理解し，処理を実行させるためのインターフェースがある．

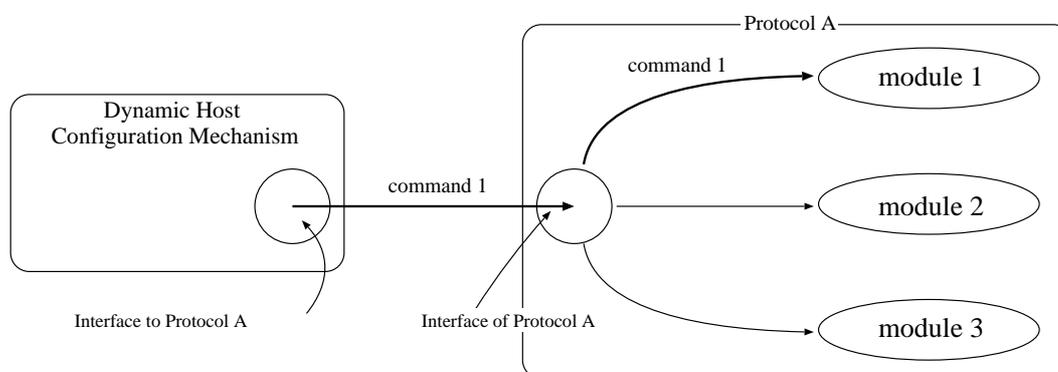


図 5.1: 適応処理の呼び出し

よって，図中 module 1 の処理を実行させようとした場合，処理の実行命令は各機構の間にあったインターフェースを用いて送信される．その後，通信技術側のインターフェースが module 1 への実行命令を理解し，module 1 を実行する．

呼び出しを行う機構と各適応処理の間のインターフェースはそれぞれ異なっても，統一したインターフェースを利用するようにしても利用できるようにする．これにより，利用者の移動体計算機の環境に合わせて自由に適応処理を組み合わせることが可能となる．

5.2 動的移動処理選択支援モデルの提案

本節では、第 5.1 節で述べた要件を満たすモデルである動的移動処理選択支援モデルを提案する。

まず、通信状態に応じて利用する適応処理を切り替えながら通信を継続させられるモデルとして以下の 3 つが考えられる

- 並列処理モデル
- 並列協調処理モデル
- 動的移動処理選択支援モデル

並列処理モデルは、移動体計算機が利用する通信技術を全て同時に走らせることによって実現するモデルである。

並列処理モデルに関する詳しい説明は第 5.2.1 節で詳しく述べる。

並列協調処理モデルは、並列処理モデル同様、利用する通信技術を全て同時に走らせるモデルであるが、適応処理の管理を一意に行えるようにしたモデルである。

並列協調処理モデルに関する詳しい説明は第 5.2.2 節で詳しく述べる。

動的移動処理選択支援モデルは、上記のモデルとは異なり、同時に走っている通信技術は各層一つずつである。変わりに、移動体計算機の状態情報の監視や利用すべき適応処理の選択などを管理する機構を外部に持つ。

動的移動処理選択支援モデルに関する詳しい説明は第 5.2.3 節で詳しく述べる。

次に、上記のそれぞれのモデルについて考察を行い、第 5.1 節で述べた要件をどの程度満たしているか解説する。

考察を行うにあたって、提案するモデルを以下の 3 つの機能について分類して特長を述べる。

- 情報監視部：
第 5.1.1 節で述べたような移動体計算機の状態情報を監視する機能を持った機構
- 適応処理判断部：
第 5.1.2 節で述べたような状態情報に基づいて適応処理を判断する機能を持った機構
- 適応処理実行部：
第 5.1.3 節で述べたような選択されて適応処理への切り替え、及び処理の実行を行う機能を持った機構

最後に、本論文で動的移動処理選択支援モデルを採用した理由について解説し、移動体計算機内における本モデルを用いた機構の利用例を示す。

5.2.1 並列処理モデル

まず始めに，計算機上で利用可能な複数の処理を通信状況に応じて利用するために，利用する全ての処理を同時に動作させるモデルを並列処理モデルとする。

移動体通信技術における並列処理モデルでは，例えば FMIPv6 や HMIPv6 のような複数の技術が同時に移動体計算機上で動作しているような状態を指す。移動体計算機上で動作している技術はそれぞれ独自に状態情報を監視し，あらかじめ利用者によって設定が行われた条件に見合った状態情報を取得した場合，対応する適応処理を実行する。並列処理モデルでは利用する移動体通信技術はそれぞれ独自に開発されたものを用いるため，利用者はそれぞれに対して独自の設定を行わなければならない。

図 5.2 に並列処理モデルにおける処理の流れの例を示す。図は全て移動体計算機の内부를表す。また，移動体計算機は以下に述べる前提条件を全て満たしているものとする。

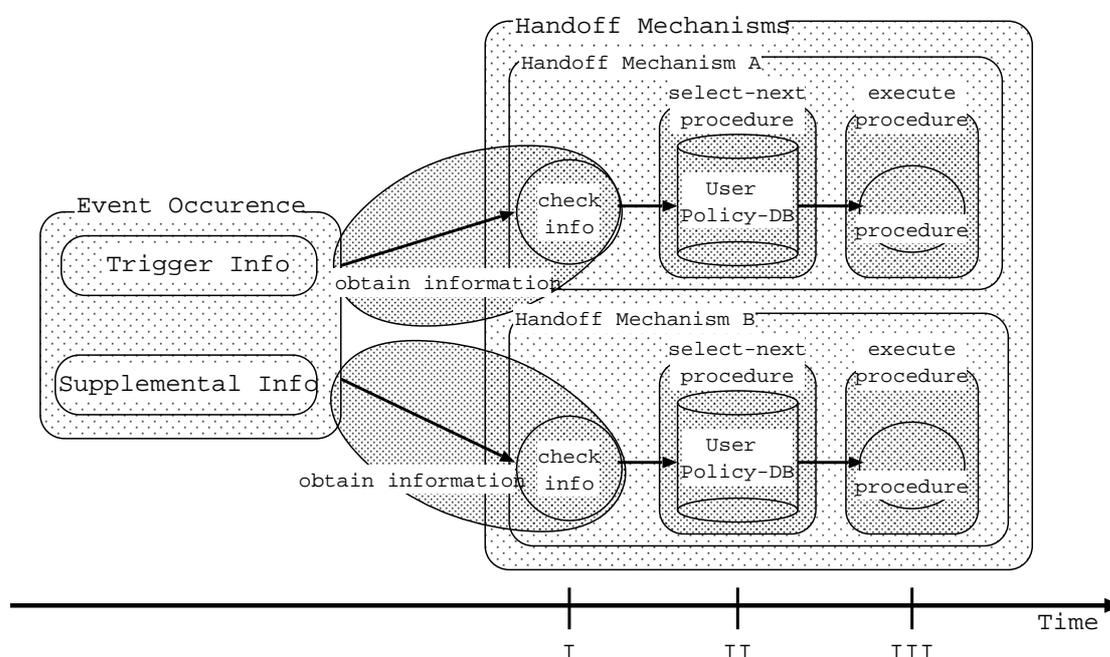


図 5.2: 例-並列処理モデルにおける処理の流れ

- 移動体計算機上では，移動体通信技術 A, B が同時に動いている
- 移動体通信技術 A, B は同時に同じ状態情報を取得している
- 利用者は移動体通信技術 A, B に対してそれぞれ適応処理を実行するための条件をあらかじめ設定している

以下に並列処理モデルに関して要件ごとに考察を行う。

移動体計算機上の状態情報の取得

並列処理モデルでは、利用する処理を含む技術は全て同時に動作しており、情報の取得も独自に行っている。よって、図中 I の情報の発生から II の情報取得までで示されているように、同一の情報を移動体通信技術 A, B それぞれが取得しなければならない。

このようなモデルでは、状態情報の発生頻度が高くなったり、情報の監視を行っている計算機上の処理の数が増えるにつれて監視に伴う処理が増加し、オーバーヘッドが大きくなる。また、既存の技術に一切手を加えずに利用することができる反面、新たに監視を行う情報の追加や、既にある情報に対する定義の変更を行うことはできない。

移動体計算機にて利用可能な処理の実行

並列処理モデルでは、利用する技術全てが同時に動作している。図中 IV の時点で各技術が状態情報にあった処理をそれぞれ実行しているように、移動体計算機上で利用可能な処理は全て実行することが可能である。また、既存の技術に一切変更を加えることなく実現することができる。しかし、それぞれ独自に処理を進めるため、複数の処理が同時に進行する等の問題が起こる可能性が極めて高い。

状態情報に基づいた実行すべき処理の判断

並列処理モデルでは、既存の技術に対して一切の変更を加えないで利用する。よって、図中 III の時点で示されているように各技術が管理している処理実行条件に沿った処理を行わなければならない。希望する処理のみを希望したタイミングで実行されるようにするのは非常に困難である。ここで処理実行条件は、状態情報とその情報の移動体通信環境における意味の対応、そして情報の意味と実行されるべき処理の対応を指す。この対応により、取得した情報を基に実行されるべき処理を判断することが可能になる。

並列処理モデルにて移動体計算機の通信を管理しようとした場合、利用者は利用する技術の処理実行条件設定手法に基づいた設定を行わなければならない。また、利用者は新たに情報や処理の追加を行うことができない。よって、既存の技術を組み合わせることで理想的な移動体通信の実現が行えるようするためには、利用者が各技術の設定を注意して行わなければならない。

5.2.2 並列協調処理モデル

次に、並列処理モデルに対して利用する技術間で処理の決定を可能にしたモデルを並列協調処理モデルとする。並列協調処理モデルでは、並列処理モデルと異なりそれぞれの技術は利用者によって設定される処理実行条件の集まりを共有し、それぞれの挙動を決定する。よって、利用される技術はあらかじめ定義された処理実行条件の検

索手法に対応していなければならない。利用者は、必要な処理全てに対する処理実行条件を一括して設定することが可能である。

図 5.3 に並列協調処理モデルの処理の流れを示す。また、移動体計算機に対する前提条件の中、並列処理モデルと唯一異なるのは、処理実行条件を設定する際にそれぞれの技術に特化した設定を行うのではなく、全てに対して共通の設定手法を用いて行うことである。

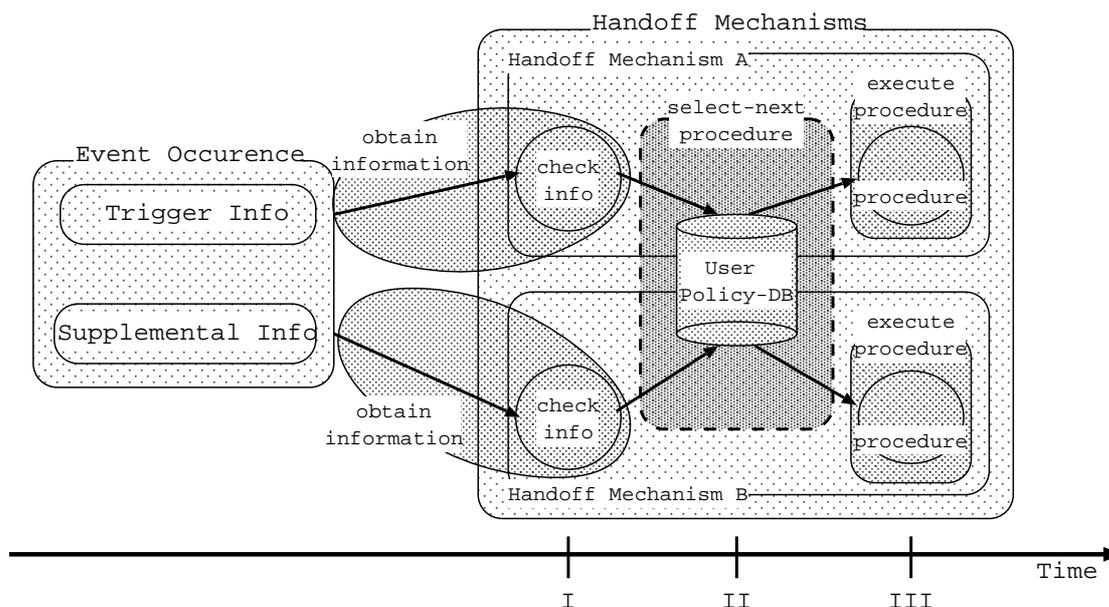


図 5.3: 例—並列協調処理モデルにおける処理の流れ

以下並列協調処理モデルについて要件ごとに考察を行う。

移動体計算機上の状態情報の取得

並列協調処理モデルは、並列処理モデル同様利用する全ての技術が同時に動作していることを前提にしているため、情報の取得に関しては並列処理モデルとほぼ同じである。よって、図中 I から II の時点で示されているように、各技術が情報の取得を独立して行っている。そのため、利用する技術の数及び状態情報が変化する頻度が高くなる程情報取得に生じるオーバーヘッドは大きくなる。一方、並列処理モデルと異なり、利用者は新たに監視を行う情報や既にある情報に対する定義の変更を行うことが可能である。なぜなら、取得した状態情報と情報が持つ意味の対応を技術ごとではなく、移動体計算機ごとに管理することができるからである。これに関しては実際に取得した情報と実行されるべき処理の判断を行う部分の解説にて詳しく述べる。

移動体計算機にて利用可能な処理の実行

先程も述べたように、並列協調処理モデルは並列処理モデル同様必要な技術全てが同時に動作している。よって、図中 IV の時点で表されているように移動体計算機上で利用可能な処理は全て実行可能である。また、それぞれの処理が同一の処理実行条件に基づいて動作するため、複数の処理が同時に進行してしまう等の問題の回避が容易に行える。

状態情報に基づいた実行すべき処理の判断

並列協調処理モデルでは、実行すべき処理の判断は全て一箇所にまとめて定義されている処理実行条件にて判断される。図中 III の時点にてその様子を示す。動作している技術はそれぞれ取得した状態情報を基に、次に実行すべき処理の検索を行う。このように、並列協調処理モデルでは利用者によって設定が行われる処理実行条件を一括して管理することができるため、並列処理モデルのように複数の処理が同時に進行してしまう問題は起こらない。また、一括して管理を行うことにより、並列協調処理モデルでは利用者が意図したタイミングで適切な処理を実行させることが可能となる。また、新たに情報の定義を行ったり、それに対応付けられる処理を定義したりすることが容易に行える。

並列協調処理モデルにおける問題点は、各技術が情報を取得した時一斉に実行されるべき処理の検索を行うことによるオーバーヘッドである。状態変化の頻度が高くなったり、利用したい技術の数が増えてきた場合検索にかかるオーバーヘッドが大きくなり、意図したタイミングで処理を実行することができなくなる可能性がある。

5.2.3 動的移動処理選択支援モデル

動的移動処理選択支援モデルは、並列処理モデル及び並列協調処理モデルとは全く違った適応処理の管理を行う。先に述べた2つのモデルは、既存の移動体通信技術及び移動体通信支援技術がそれぞれ独自に動きながら環境に応じて適切な処理を行うことで、多様な状態への対応を試みるモデルである。しかし、動的移動処理選択支援モデルでは、移動体計算機の状態情報の監視及び適応処理の選択を行う機構を実際に処理を行う技術の前に介入させ、処理の管理を行うことで多様な処理に対応する。よって、移動体計算機上の処理に関する全ての情報は一括に管理され、移動体通信技術や移動体通信支援技術は呼び出しを受けた時のみ処理を行う。また、処理実行条件の設定も一括に行うことが可能である。

図 5.4 に動的移動処理選択支援モデルにおける処理の流れを示す。また、本モデルは移動体計算機が以下に述べる全ての条件を満たしていることを前提とする。

- 移動体計算機上で必要となる状態情報は全て実行される処理の判断を行う機構が監視できる状態にある

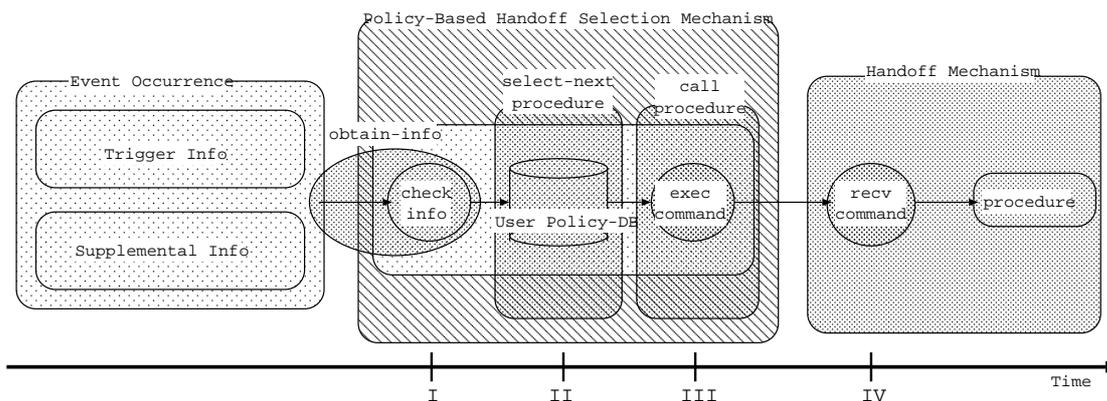


図 5.4: 例-動的移動処理選択支援モデルにおける処理の流れ

- 既存の移動体通信技術及び移動体通信支援技術は呼び出しが無い限り，一切処理の判断及び実行を行わない
- 既存の技術と処理の呼び出しを行う機構の間には第 2.2 節で定義された書式サポート及び統一のインターフェースのどちらかが無ければならない

以下に動的移動処理選択モデルについて要件ごとに考察を行う。

移動体計算機上の状態情報の取得

動的移動処理選択支援モデルでは，既存の技術とは異なった場所で情報の監視を一括して行う。よって，並列処理モデルや並列協調処理モデルのように既存の技術が同時に動作し，それぞれ情報の監視を行うことは一切しない。一括して情報の監視を行うことで，他のモデルでは利用する技術の数の増加に伴いオーバーヘッドが大きくなっていた問題を解決することができる。また，図中 I の時点で示されているように動的移動処理選択支援モデルは並列協調モデル同様，情報の監視を一括して行うことができるため，利用者による新たな監視情報の追加及び既にある情報とその意味の対応を変更することができる。

移動体計算機にて利用可能な処理の実行

本モデルでは，移動体通信技術や移動体通信支援技術などの計算機上の処理は，外部機構より呼び出しが行われるまで一切処理を進めない。つまり，移動体計算機上で常時動作しているのは状態情報の監視及び実行されるべき処理の呼び出しを行う外部機構だけである。外部機構を用いる場合，第 2.2 節で述べた処理の実行手法のどちらかに対応していなければならない。動的移動処理選択支援モデルでは，技術ごとの実行書式をサポートさせる手法をとる。新たな処理の追加は，統一インターフェースを持っている場合と比較して非常に高くなる一方，既存の技術に対して行う変更を最小にす

ることができる。技術ごとに実行書式をサポートし、既存の技術に対して行う変更を最小にすることで、本モデルはより多くの環境にて利用することが可能となる。

本モデルのように外部機構にて状態情報の監視を行い、処理の呼び出しを適宜行わせるモデルは、他のモデルと比較して状態情報の監視及び処理の呼び出しにかかるオーバーヘッドは大きい。これは、他のモデルと異なり利用される技術が常に動作しておらず、外部機構からの呼び出しに依存しなければならないからである。

状態情報に基づいた実行すべき処理の判断

動的移動処理選択支援モデルによる、状態情報に基づいた実行すべき処理の判断は、並列協調処理モデルと同様一箇所にまとめて定義された処理実行条件にて判断が行われる。図中 II の時点で実行すべき処理の判断を行う。図から分かるように、本モデルは並列協調処理モデルと異なり外部機構内部で判断が行われるため、実行すべき処理の判断において既存の技術に一切変更を加える必要はない。

本モデルは、並列協調処理モデルで述べた実行すべき処理の判断における利点を全て実現する。まず、並列処理モデルのような一つの状態情報に対して複数の処理が実行されない。そして、利用者が意図したタイミングで処理の呼び出しを行うことも可能である。また、外部機構にて判断を行うため、利用者があらかじめ設定することにより新たに情報の定義を行ったり、対応付けられる処理の定義を変更することが容易にできる。

並列協調処理モデルにおける処理実行条件の検索とは異なり、本モデルにおける処理実行条件の検索は情報の監視を行っている部分と検索を行う部分が一対一で対応している。よって、並列協調処理モデルで問題であった利用する技術の数に比例してオーバーヘッドが大きくなる問題は起こらない。

5.2.4 まとめ

前章までで述べた各モデルは、それぞれ長所・短所を持っている。本節では、以下の4項目に関して各モデルの性能を表にしてまとめ、考察を行う。

- 処理オーバーヘッド：
移動体計算機の状態情報を常に監視し、状態に応じて適切な処理を行うには様々な部分で処理オーバーヘッドが発生する。処理オーバーヘッドが最も小さいモデル、移動体通信に適している。
- 処理の正確さ：
移動体計算機上の通信技術の組み合わせの変更、処理の実行など、利用者が意図したタイミングで適切な処理を行うのは困難である。利用者の意図した処理を適宜実行できるモデルが好ましい。
- 拡張性：

新たに切り替え対象となる通信技術を追加することができなければ，多様な移動体通信環境に対応することはできない．新たな通信技術や状態情報の定義を追加する際に，容易に行えることが望ましい．

以下に各項目に関してそれぞれのモデルの性能を検討し，考察を述べる．

処理オーバーヘッド

表 5.1にて各処理に対して発生するオーバーヘッドをモデルごとに比較し，3段階に分けて表した．

表中の記号は，オーバーヘッドの小さい順から ， ， × と大きくなってゆく．

表 5.1: 各モデルにおける処理オーバーヘッド

モデル / 項目	情報監視部	適応処理判断部	適応処理実行部
並列処理モデル	×	×	
並列協調処理モデル	×	×	
動的移動処理選択支援モデル			

各処理におけるそれぞれのモデルのオーバーヘッドについて検討する．

情報監視部は，一定の時間内に状態情報監視に関する処理を実行する回数からオーバーヘッドを予想した．まず，並列処理モデル及び並列協調処理モデルであるが，この2つのモデルにおける情報監視部の処理オーバーヘッドは同じであると考えられる．なぜなら，各モデル共に利用する通信技術が独自に情報の監視を行うからである．このようなモデルでは，同一の通信層にあるプロトコルが同じ状態情報の監視を行う．よって，同じ状態情報の監視を行っている通信技術の数が増加する毎，さらに状態情報の変化が頻繁になる毎に処理オーバーヘッドは乗数的に大きくなる．

動的移動処理選択支援モデルは，状態情報の監視を行うのは監視用の機構だけであり，利用する通信技術などの増加の影響を受けない．また，状態情報の変化が激しくなった場合も他の2つのモデルと比較してオーバーヘッドの増加は少なくなる．しかし，動的移動処理選択支援モデルでは，監視用の機構が移動体計算機上の全ての情報を監視するため，一箇所にかかる負荷が大きい等の問題がある．

次に適応処理判断部である．並列処理モデルでは，各処理が処理実行条件を持っており，情報を取得する度に検索を行う．そのため，情報の監視におけるオーバーヘッドと同様，利用する通信技術の数や状態変化の数の増加によって乗数的にオーバーヘッドが増加する．

並列協調処理モデルも，処理実行条件を一箇所で行っている以外は検索回数は並列処理モデルと同じである．そのため，処理オーバーヘッドは乗数的に増加する．

動的移動処理選択支援モデルは、情報監視部の時と同様、適応処理検索にかかるオーバーヘッドは他の2つ比べて小さくて済む。

適応処理実行部は、各適応処理におけるオーバーヘッドを比較する。並列処理モデル及び並列協調処理モデルは、どちらも利用する通信技術が同時に動作している状態にあることと、処理は通信技術内部から直接呼び出すことができるため処理オーバーヘッドが低くて済む。

一方動的移動処理選択支援モデルでは、適応処理の検索を行ったあと、通信技術に合ったインターフェースを用いて処理の呼び出しを行わなければならないため、処理オーバーヘッドは他の2つのモデルと比較して大きくなる。

処理の正確さ

表 5.2に各モデルが利用者の意図をどこまで正確に通信に反映することが可能かを表した。

表中の記号は、利用者の意図を正確に通信に反映することができれば○、できない場合があれば×で表す。

表 5.2: 各モデルにおける処理の正確さの比較

モデル / 項目	処理の正確さ
並列処理モデル	×
並列協調処理モデル	
動的移動処理選択支援モデル	

利用者の意図を正確に通信に反映させることを考えた場合、並列処理モデルのみ困難である。なぜなら、並列処理モデルは処理実行条件を各プロトコル毎に保持しているため、排他的な制御を行うのは非常に困難であるからだ。また、第 5.2.1節で述べたように利用者が各通信技術に対して設定を行わなければならないことから、設定が非常に困難であり、利用者は自分の望む通信を設定することができない。

並列協調処理モデル及び動的移動処理選択支援モデルは、処理実行条件を一括して管理することができるため、排他的な処理を行うのに適している。これら2つのモデルでは、設定は一括して行えるため、利用者にも理解しやすい。よって、利用者は自分の望む通信を容易に設定することができる。結果として、利用者の意図に忠実な移動体通信を提供することが可能となる。

拡張性

表 5.3に各モデルにおける拡張性を示す。

表中の記号は，拡張性の高いものから順に ， ， × と低くなっていく．

表 5.3: 各モデルにおける処理の正確さの比較

モデル / 項目	拡張性
並列処理モデル	
並列協調処理モデル	×
動的移動処理選択支援モデル	

拡張性は，新たに利用する通信技術の追加を行うことを考えた場合，各プロトコルに対して加えなければならない変更点の量から決定している．

まず，並列処理モデルは既存の通信技術に一切の変更を加えることなく利用することを前提としているため，拡張性が高い．

一方，並列協調処理モデルは全ての通信技術に対して特定の処理実行条件を扱えるように変更を加えなければならない為，拡張性は低いと言える．

動的移動処理選択支援モデルは，全ての通信技術に対して呼び出しや，処理の実行を行うためのインターフェースを提供することを条件としている．そのため，通信技術に対して変更点を加えなければならないが，拡張性は低い．しかし，既に独自のインターフェースを持っている場合は，呼び出し側で対応することも可能なため，拡張手法が豊富である．

これまでで，各モデルの長所・短所を比較した．結果，動的移動処理選択支援モデルは他の2つのモデルと比較して，処理オーバーヘッドや処理の正確さ，そして拡張性において最もバランスが良いモデルだと判断する．

本論文では，動的移動処理選択支援モデルを用いて移動体計算機の通信制御を行い，効率的な通信の提供を試みる．

第6章 システム設計

第5章で提案した動的移動処理選択支援モデルを実現する動的移動処理選択支援機構の設計概要や全体構成，さらに基本動作について述べる．また，第3.1章で述べた多様化する情報を移動体通信環境における特性を基に抽象化を行い，本モデルではどのようにして取り扱うか解説する．

6.1 動的移動処理選択機構における用語の定義

本節では，既存の移動体通信技術及び移動体通信支援技術から切り離された場所に位置し，移動体計算機の状態情報の監視及び適応処理の選択を行う機構である動的移動処理選択機構にて新たに定義する用語について解説を行う．

- User Intension(UIn) :
利用者によって定義される理想的な移動体通信及び移動体通信の処理の流れ．
- Mobility Management Daemon(MMD) :
動的移動処理選択支援機構において移動体計算機の状態情報の監視及び適応処理の選択を行う部分．
- Mobile Communication Optimizing Procedure(MCOP) :
MMDによって選択，実行される移動体通信技術及び移動体通信支援技術．移動体通信技術に特化した処理だけでなく，通信の効率化を行うもの全てが対象となる．
- Policy Data Base(PDB) :
MMD内で保持するUIn．状態情報とMCOPの対応を持つ．

6.2 前提とする移動体計算機の環境

本節では，動的移動処理選択支援機構の説明を行う上で前提とする移動体計算機の環境について述べる．移動体計算機での通信に用いる移動体通信技術であるMIPv6に関して処理の流れを解説する．

本章では以後，以下の条件を満たす移動体計算機を想定して解説を行っていく．

- 複数の通信インターフェースを有している：
移動体計算機はIEEE 802.11b 無線通信デバイスやIEEE 802.3 Ethernet 通信デバイス，そして携帯電話などの通信デバイスを複数持っている．

- MIPv6 を用いた通信を行う：
移動体計算機は，MIPv6 を用いて一意の IP アドレスで移動中も通信を行う．
- 動的移動処理選択支援機構を用いて通信の管理を行う：
移動体計算機は，MIPv6 の処理の管理や通信インターフェースの選択など，全て動的移動処理選択支援機構を用いて管理されている．

6.3 設計概要

第 5.2 章にて動的移動処理選択支援モデルについて概要を述べた．本節では，本モデルを基に動的移動処理選択機構 (MMD) がどのように動作するかを述べる．

第 3 章にて単一の移動体通信技術では多様化する移動体通信環境において常に理想とされる通信品質を提供することは困難であることを述べた．動的移動処理選択モデルでは，この問題を複数の MCOP を組み合わせて利用することによってこの問題の解決を試みる．また，MCOP を組み合わせて利用する際に，理想的な移動体通信の定義が必要である事も既に述べた．動的移動処理選択モデルでは，MMD に利用者の UIn 情報を管理させる機構を持たせ，移動体通信における柔軟な処理の管理を実現する．

OSI 参照モデルにおける MMD の位置付け及び本機構の構成を図 6.1 示す．

MMD は，以下の 3 つの機構に大別することができる．また，MMD を用いた処理の流れもこれらの機構に基づいて構成される．

以下に処理の流れに沿って各機構の概要を説明する．

1. 状態情報監視機構：
移動体計算機上で，通信の状態を示すものを各階層から取得するための機構．MMD の処理は移動体計算機の状態情報を取得することから始まる．
2. 適応処理判断機構：
取得した状態情報を基に実行されるべき処理を選択する．
3. 設定実行機構：
適応処理判断機構にて選択された処理の実行を行う．処理実行の対象となる通信プロトコルやユーザーランドアプリケーションとの間のインターフェース．

以下に MMD における第 5.2 節で述べた機能に関して具体的に解説する．

6.4 状態情報監視機構

状態情報監視機構は，適応処理判断部に対して移動体計算機の情報を提供する機構である．情報取得機構より得ることができる移動体計算機の状態情報の例を図 6.1 に示す．

MMD は，上記の様な情報を全て取得できなければならない．状態情報に変化が起きた場合，図 6.1 で示したように適応処理判断機構にその情報を提供する．

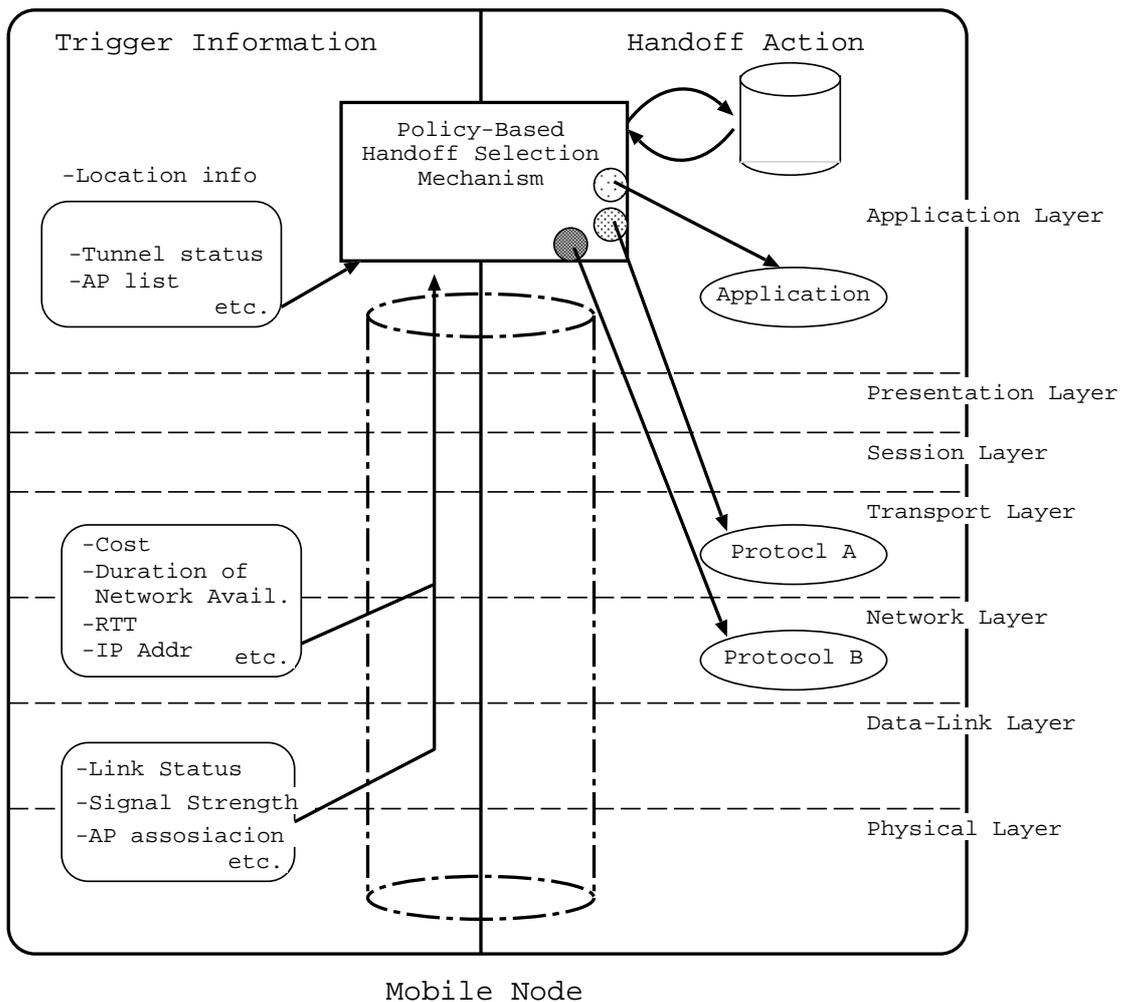


図 6.1: MMD 動作概要

表 6.1: 各モデルにおける処理の正確さの比較

階層	情報の種類
アプリケーション層	位置情報, 要求帯域, 通信コスト
トランスポート層	利用帯域, パケットロス率, 遅延
ネットワーク層	IP アドレス, 通信経路
データリンク層	リンク情報, 無線電波強度, 利用可能な通信デバイス

状態情報監視機構は, 階層に関係なく通信の効率化に利用することが可能な状態情報全てを取得する.

第 4.1 節に本論文における情報の取得手法に関して具体的に述べる.

6.5 適応処理判断機構

適応処理判断機構は、状態情報監視機構より得た情報を基に、利用者があらかじめ設定した UIn の検索を行う。検索の結果として、次に行うべき処理を適応処理実行機構に渡す。

適応処理判断機構には、2通りの抽象化の種類がある。一つは、デバイスや通信プロトコルから直接取得できるプリミティブな情報の抽象化。そしてもう一つは、利用者に対する情報及び MCOP の抽象化である。

以下にそれぞれの抽象化と実際に利用される場合、どのようにして対応付けられるかについて考察を行う。

6.5.1 利用環境の状態情報の抽象化

状態情報監視機構から取得できる情報は、デバイスからの情報や、各プロトコル独自の状態情報などプリミティブな情報であり、そのままの状態では意味をなさない。既存の移動体通信技術はデバイスから上がってくるプリミティブな情報を基に設計されておらず、情報が持つ意味を基に設計されている。そのため、状態情報に基づいて効率的な通信を実現するためにはプリミティブな情報をそれぞれの意味に基づいて抽象化できている必要がある。

MIPv6 を想定した通信に利用する情報と、抽象化を行った後の意味の対応を表 6.2 に示す。

表 6.2: プリミティブな情報の抽象化

情報の意味	情報の種類
ネットワークの移動	IP アドレスの変化
移動する可能性が高い	パケットロス率の上昇, 遅延の増長 無線電波強度の低下, 位置情報の変化
移動した可能性が高い	リンク状態の変化, default router の変化

MIPv6 における処理と情報の関連付けを表??示す。

以上のように、抽象化された情報と MIPv6 の処理は対応付けられている。ネットワークの移動を検知した場合、MIPv6 における MN は新しいアドレスの通知を行うために binding update を送信する。また、MN が移動する可能性が高いことを示す情報を取得した場合、あらかじめ Router Discovery や Neighbor Unreachability Detection を頻繁に行うことで実際の移動を早期に発見することができるようになる。さらに、移動した可能性が高いことを示す情報を取得した場合は、Neighbor Solicitation や Router Solicitation を行うことで実際に移動したのかを確認することができる。

MIPv6 における処理は、このように抽象化された情報を基に実行されるように設計

表 6.3: MIPv6 における処理と状態情報の対応

情報	処理
ネットワークの移動	binding update 送信
移動する可能性が高い	Router Discovery を頻繁に行う Neighbor Unreachability Detection を頻繁に行う
移動した可能性が高い	Neighbor Solicitation を行う Router Solicitation を行う

されている．よって，移動体計算機は取得する情報の抽象化を行い，MIPv6 の処理に結びつけることをしなければならない．

しかし，実際に通信を行っている最中に上記のような抽象化のプロセスが入っている場合は処理オーバーヘッドが大きくなる．そこで，MIPv6 における実際の通信では，表 6.4 みたたく一度抽象化して対応付けを行った結果を基に，プリミティブな情報と MIPv6 の処理を直接対応付ける．

表 6.4: MIPv6 における処理と状態情報の対応

情報	処理
IP アドレスの変化	binding update 送信
パケットロス率の情報，遅延の増長 無線電波強度の低下，位置情報の変化	Router Discovery を頻繁に行う Neighbor Unreachability Detection を頻繁に行う
リンク状態の変化 default router の変化	Neighbor Solicitation を行う Router Solicitation を行う

6.5.2 利用者に対する高度な抽象化

利用者に対する抽象化は，利用者によって希望する設定粒度が異なるため，一意に最適なものを決定するのは非常に困難である．

本論文では，利用者にとって最も効率的な抽象化を一意に決定することを目的としていないため，設定手法に一貫性を持たせるところまでを実現する．

利用者に対する抽象化は，移動体計算機が通信ができなくなった，遅延が激しくなった，パケットロス率が高くなってきた等の抽象的な情報を扱えるように抽象化しなければならない．

また，実行されるべき設定の変更に関しても，繋ぎなおしたい，同時に利用する通信インターフェースの数を増やしたい，移動時にできるだけ早く通信を再開といった

抽象的な定義を基に実際の処理を行えるようにしなければならない。

6.6 適応処理実行機構

適応処理実行機構は、適応処理判断機構にて選択された MCOP の呼び出しや実行を行う。

MMD を用いて通信を行っている移動体計算機では、状態情報は全て MMD によって監視される。MCOP は MMD からの呼び出しや処理の実行命令が送信されるまで情報の監視などは一切行わない。各 MCOP は、自分で状態情報の取得を行わないため、MMD が適応処理実行機構から呼び出しと共に付加する情報を基に処理を行わなければならない。

以下に MIPv6 を例にどのような情報が MMD から MIPv6 に渡されるか述べる。付録 A に MIPv6 について解説する。

- 実行したい処理：
MMD は MCOP に対して具体的にどの処理を実行するのかを指定しなければならない。例えば、MIPv6 における binding update や homeagent address discovery 等である。
- 処理に必要な情報：
特定の処理を実行させる場合、MMD はその処理に必要な情報を提供しなければならない場合がある。例えば、利用者が意図的に binding update を送信したい場合の binding update の送信先のアドレスなどである。

このように、MMD は呼び出しまたは処理の実行を行わせたい通信プロトコルに対して必要な情報を提供しなければならない。提供する情報はプロトコルによって異なるため、一意に定義することはできない。

第7章 動的移動処理選択支援機構の実装

本章では，動的移動処理選択支援機構の実装について述べる．

7.1 実装環境

動的移動処理選択支援機構の実装は FreeBSD[17]4.6.2-Release 上に行った．MIPv6 及び Mobility Socket の実装は SFCMIP[18] の draft-ietf-mobileip-ipv6-18 版を用いた．また，全ての機能は C 言語を用いて実装した．

7.2 実装概要

本機構の実装は，第 6.3 節で述べた 3 つの機構の実装から成る．以下に各機構の実装に関して概要を述べる．

7.2.1 状態情報監視機構

kernel 空間から情報の変化を取得してくる．本実装では，SFCMIP 内で利用されている情報監視機構である Mobility Socket を用いた．Mobility Socket は MIBsocket の機能を情報の取得に限定したものである．Mobility Socket は本論文で新たに実装された機能ではないため，詳しい説明は省略する．

7.2.2 適応処理判断機構

適応処理判断機構は，以下の 2 つの要素からなる．

- 設定のパーズング：
利用者によって設定された内容をパーズングし，PDB を生成する．第 7.4.1 節にて詳しく説明する．
- 適応処理の検索：
状態情報監視機構から取得した情報を基に次に実行すべき適応処理の検索を行う．第 7.4.2 節にて詳しく解説する．

利用者によって設定された内容のパーズングを行うには、特定の書式を利用者に対して提供する必要がある。本機構では、第 6.5.1 節で述べた抽象化に基づいて設定ファイルの書式を定める。

適応処理の検索は、適応処理判断機構が内部に保持している state と、状態情報監視機構から取得した状態情報に基づいて検索を行う。結果として実行すべき適応処理への関数ポインターを取得する。

7.2.3 適応処理実行機構

適応処理判断機構から取得した関数ポインターの先の関数を実行させる。また同時に、適応処理判断機構内部の state の更新を行う。

適応処理判断機構から取得する関数ポインターは、必要に応じて設定ファイルにて追加することができる。また、関数ポインターに対応する関数は、実行する適応処理のインターフェースに合わせてプログラムされている必要がある。

7.3 状態情報監視機構：Mobility Socket の利用

Mobility Socket からの情報の取得は、raw socket を Mobility Socket 独自のプロトコル・ファミリーである、PF_MSOCK を用いて open し、read することで行える。

MSOCK から取得している情報を図 7.1 に示す。MSOCK を用いて kernel 空間から取得する情報は現在これらに限定している。よって、msock からこれらに関する情報が read された場合、全ての情報を適応処理判断機構へ渡す。

```
#define MSOCK_ADD          0x01
#define MSOCK_DELETE      0x02
#define MSOCK_NOTFOUND    0x03
#define MSOCK_CHANGE      0x04
#define MSOCK_UP          0x05
#define MSOCK_DOWN        0x06
#define MSOCK_POOR        0x07
#define MSOCK_REFERER     0x08
```

図 7.1: msock から取得する情報-net/mobilitysocket.h

MSOCK から取得できるこれらの情報は、適応処理判断機構にて新たに意味付けを行うことが可能である。適応処理判断機構での意味付けは、第 7.4.1 項で解説を行う。

7.4 適応処理判断機構

適応処理判断機構には、設定のパーズングと適応処理の検索の2つの機能が必要であることは第7.2.2節で述べた。以下にそれぞれの実装に関して述べる。

7.4.1 設定のパーズング

第6.5.2節にて、利用者に対する情報の抽象化に関して説明した。

本節では、利用者に対する設定に関して実装を基に説明する。設定を行う項目は、以下のような4項目に大別することができる。

1. 情報の意味とプリミティブな情報の対応付け
2. 処理の種類と処理を行う関数の対応付け
3. 通信の状態と処理の種類に対応付け
4. 情報の意味と処理の目的の対応付け

利用者は全ての項目に対して設定を行うことができる。しかし、似たような環境を持っている計算機に対しては、1-3は既存の設定を組み合わせ、4だけ利用者が設定を行うことも可能である。

情報の意味とプリミティブな情報の対応付け

図7.1で挙げたような情報と、その情報が移動体計算機上で持つ意味の関連付けを示す。この関連付けに基づいて、適応処理判断機構は最適な処理の選択を行う。

図7.2に設定例を示す。状態を示すプリミティブな情報は、情報の種類と情報の内容の2つと関連付けることによって抽象化を行う。例えば、図中のリンクダウンを意味する情報 "0x06" では、link_status が情報の種類を、そして down_trig が情報の内容を指す。

本実装では、利用者の設定でこれ以上情報の意味に関する抽象化を行わない。しかし、さらに抽象化することで、link_status の down_trig 情報を "通信が切れた" といったように表現することも可能である。

処理の種類と処理を行う関数の対応付け

情報と同様、各適応処理に対しても処理を実行するための関数と、それらの処理が持つ意味について対応付けを行う必要がある。

図7.3に設定例を示す。適応処理判断機構より選択される処理は、処理の種類と処理の名前を指す func_name で表される。例えば、図中の通信インターフェース切り替え

```

link_status:
    !down_trig=0x06
    !up_trig=0x05
;

elvnb_signal:
    !weak_trig=0x07
;

ifsw:
    !next_if=0x30
    !next_if_fail=0x31
;

```

図 7.2: 情報の意味とプリミティブな情報の対応付け

```

+DEF_CONNECT_11b:
    !fnc_name=set_elvnb
    !interfaces=(1:wi0)
;

+DEF_IFSW:
    !fnc_name=ifsw
    !interfaces=(1:wi0, 2:fxp0, 3:tun, 4:gif0)
    !search_opt=try_all
;

```

図 7.3: 処理の種類と処理を行う関数の対応付け

を行わせる処理である ifsw では、DEF_IFSW が処理の種類を、そして func_name が処理の名前に関する情報であることを指す。

また、処理の種類に対して特定の情報を負荷したい場合は、利用者が自由に追加することが可能である。例えば、DEF_IFSW の例では、切り替え優先順位を示す情報として interfaces という項目が追加されてる。

通信の状態と処理の種類の対応付け

適応処理に関係する関数と、その処理の種類の対応付けに関して説明した。処理の種類は、更なる抽象化を行うために、その処理を実行する目的で分類する。

図 7.4 に設定例を示す。処理の種類は、その内容に基づいて、実行されるべき通信の状態に抽象化される。IFSW を例にして考える。IFSW は移動体計算機の状態に基づいて、通信に利用するインターフェースを切り替えることによって、常時接続性や必要な帯域の確保を行うための技術である。よって、図中で移動体計算機がネットワークへの接続性を失った状態を示す LOST_CONNECTION 内に関連付けることができる。

```
+LOST_CONNECTION:
    !mod_priority=(1:DEF_IFSW, 2:DEF_GPS_APL, 3:DEF_CONNECT_11b,
        4:DEF_CAR_APL, 5:DEF_CONNECT_11b, 6:DEF_SCAN_11b,
        7:DEF_CONNECT_11b, 8:DEF_WAIT, 9:DEF_SHTDWN)
    !mod_opt=(1:interfaces=(1:fxp0, 2:wi0, 3:tun, 4:gif0))
    !ifsw=(next_if:DEFAULT, next_if_fail:NEXT)
    !gps_ap_lookup=(ap_info_gps:NEXT, fail_gps:DEF_CAR_APL)
    !set_elvnb=(PRI:3, next_ap:DEFAULT, fail_11b_connect:NEXT)
    !car_ap_lookup=(ap_info_car:NEXT, fail_car:DEF_SCAN_11b)
    !set_elvnb=(PRI:5, next_ap:DEFAULT, fail_11b_connect:NEXT)
    !elvnb_ap_scan=(ap_info_scan:NEXT, fail_scan:DEF_WAIT)
    !set_elvnb=(PRI:7, next_ap:DEFAULT, fail_11b_connect:NEXT)
    !wait_timer=(wait_done:LOST_CONNECTION, wait_over:NEXT)
;
```

図 7.4: 通信の状態と処理の種類の対応付け

移動体計算機がネットワークへの接続性を失った状態において、実行する処理が IFSW だけでは十分ではないと利用者が判断した場合、図中の mod_priority 内に定義することで必要な処理を追加することができる。mod_priority 内に指定する項目としては、LOST_CONNECTION 内の処理における優先順位とそれに対応する処理の種類である。

mod_priority 以外にも、設定を行える項目として mod_opt がある。ここでは、mod_priority 内で指定された優先順位ごとに付加情報を設定する。

最後に、LOST_CONNECTION 内で移動体計算機が取得する情報に基づいた処理の流れを情報ごとに指定する。例えば、IFSW を実行している最中に移動体計算機から取得する情報として、next_if と next_if_fail がある。もし IFSW に成功した場合は、next_if として通信に利用するインターフェース名を取得する。この時、コロンの分割された値の DEFAULT は次の挙動を示す。この場合、IFSW が成功し、通信を再開することができたら DEFAULT として定義されている状態に状態を遷移させることを指す。

情報の意味と処理の目的の対応付け

本実装では，利用者に対して最も抽象度の高い設定手法が情報の意味と処理の目的の対応である．利用者は，特定の情報に対して通信設定の目的を指定することで通信プロトコルの制御を行わせる．

図 7.5 に設定の例を示す．ここでも，LOST_CONNECTION の例と同様にコロンで分割された値に次の挙動として LOST_CONNECTION のような通信の状態を示す値が入る．

```
+DEFAULT:
    !link_status=(PRI:1, down_trig:LOST_CONNECTION)
    !gps_info=(PRI:2, edge_trig:GPS_EDGE)
    !elvnb_signal=(PRI:2, weak_trig:11b_WEAK)
    !vsix_addr=(PRI:3, no_v6addr_trig:NO_V6ADDR)
;
```

図 7.5: 通信の状態と処理の種類に対応付け

7.4.2 適応処理の検索

適応処理の検索は，図 7.6 の流れに沿って行われる．

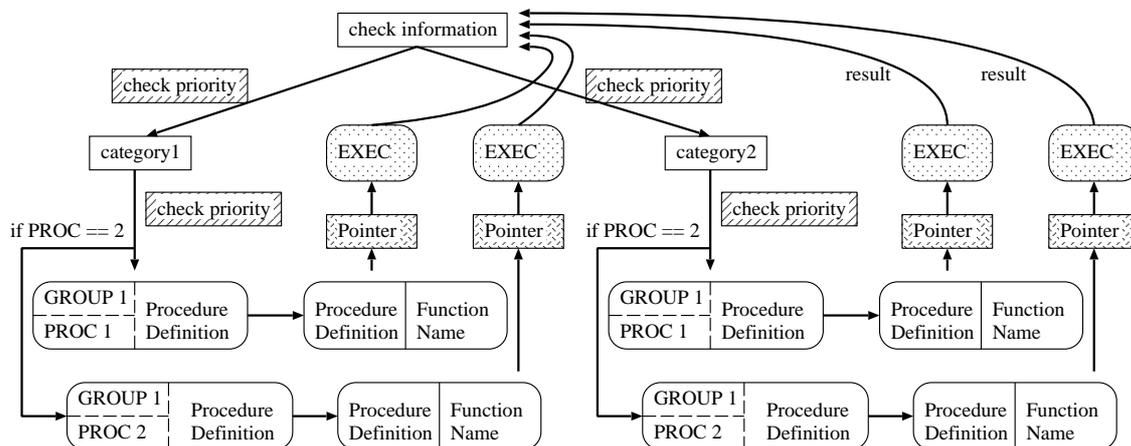


図 7.6: 情報の抽象化と処理の流れ

本システムは，常に処理の内容に基づいた state を管理している．

実行すべき設定処理は，移動体計算機が状態情報取得部から取得した情報と本システム内部で管理を行っている state を基に検索を行う．この手法により，検索に必要な情報は state と取得した状態情報のみで行えるようになる．

本システム内の state は，図 7.6 の矢印の流れに沿って遷移する．

7.5 適応処理実行機構

適応処理実行機構は，適応処理判断機構より取得した，関数ポインタを実行する．

第8章 評価

本章では，本システムの定性的評価及び定量的評価を行う．

8.1 定性的評価

本システムは，移動体計算機の状態に基づいて適切な設定を実行する．これらの機能が正常に動作しているかを検証するために，動作確認を行い，利用者があらかじめ設定しておいた内容通りに処理が実行されることを評価した．

8.1.1 実験環境

本システムの定性的評価を行った環境を表 8.1.1 に示す．本システムでは MSOCK を利用するため，OS に SFCMIP の 2002 年 10 月 15 日版を用いて評価を行った．

表 8.1: MIPv6 における処理と状態情報の対応

項目	仕様
OS	FreeBSD 4.6.2-RELEASE with SFCMIP-Oct15
CPU	AMD Athlon 650Mhz
RAM	512MB
Interfaces	IEEE802.3 100M-Ethernet device, IEEE802.11b WLAN device

8.1.2 実験内容

実験では，以下のような状態変化を移動体計算機に対して発生させ，適切な処理が行えているかを確認する．

- リンク情報の変化
- IEEE802.11b の無線電波強度の変化

尚，利用者があらかじめ設定した内容は以下の通りである．

1. IEEE802.11b 無線通信デバイスの電波強度が弱まったら IEEE802.3 100M-Ethernet 通信デバイスが利用可能か確認する
2. IEEE802.3 100M-Ethernet 通信デバイスが利用可能な場合，通信 IF の切り替え処理を行う
3. IEEE802.3 100M-Ethernet 通信デバイスに切り替えが成功した場合は，そのまま通信を行う．しかし，失敗した場合は次の処理へ移る
4. IEEE802.3 100M-Ethernet 通信デバイスが利用不可能だった場合，近隣 Access Point の検索を行う
5. 接続可能な Access Point を発見した場合，IEEE802.11b 無線通信デバイスを用いて通信を行うように設定を行う
6. IEEE802.3 100M-Ethernet 通信デバイスへの切り替えに失敗した場合，もう一度 IEEE802.11b 無線通信デバイスを用いて通信を行えるか試す
7. 両方の通信デバイスで通信を行うことができなかった場合，30 秒待ってどちらかの通信デバイスが利用可能になるか待つ
8. 所有している通信デバイスが利用可能になった場合，それを用いて通信を行う
9. 30 秒経過しても利用可能なデバイスが登場しなかった場合，計算機をシャットダウンする

以上の処理を本システムにおける設定ファイルとして記述した場合，図 8.1 のようになる．上記の処理番号との対応を，ファイル中の括弧内に示す．

また，本システムは MSOCK を用いて移動体計算機の状態情報を取得するだけでなく，TCP ソケットを用いたユーザーランドアプリケーション及びリモート・ホストからの状態情報の取得を前提に実装されている．本システムに対して情報を提供するユーザーランドアプリケーションとして，GPS や CARD が考えられる．

今回の評価では，この特性を用いて本来なら MSOCK 経由で取得する情報をユーザーランドアプリケーションでエミュレートした．情報のエミュレートも含めた今回の評価の流れを図 8.2 に示す．図中各処理における番号は，前述の利用者が行った設定の番号を示す．

各処理は，実行された結果としてランダムに選ばれた結果を本システムに対して送信する．

上記のような評価システムにおいて，本システムは第 2.2 節にて述べた機能要件を満たしているかを評価した．

```

+DEFAULT:
    !elvnb_signal=(PRI:1, weak_trig:11b_WEAK)

+11b_WEAK:
    !mod_priority(1:DEF_CHK_AVAILIF, 2:DEF_IFSW, 3:DEF_SCAN_11b,
        4:DEF_IFSW, 5:DEF_WAIT, 6:DEF_CHK_AVAILIF, 7:DEF_IFSW,
        8:DEF_SHTDWN)
    !avail_if(if_avail:NEXT, if_navail:DEF_SCAN_11b)----- (1, 5)
    !ifsw=(next_if:DEFAULT, next_if_fail:NEXT)----- (2, 3)
    !elvnb_ap_scan=(ap_info_scan:NEXT, fail_scan:DEF_WAIT)-(4)
    !ifsw=(next_if:DEFAULT, next_if_fail:NEXT)----- (6)
    !wait_timer=(wait_done:NEXT, wait_over:DEF_SHTDWN)----- (7)
    !avail_if(if_avail:NEXT, if_navail:DEF_SHTDWN)----- (8, 9)
    !ifsw=(next_if:DEFAULT, next_if_fail:NEXT)----- (8)

;

```

図 8.1: 通信の状態と処理の種類に対応付け

8.1.3 実験結果

第 5.1 節にて、本システムが満たさなければならない条件は以下の 3 つであることを述べた。

- 移動体計算機の状態情報の監視を行える
- 通信状態に基づいた最適な設定の判断が行える
- 選択された設定への切り替え及び処理の実行が行える

これらの条件が全て満たされているかを検証するためには、本システムで第 8.1.2 節で述べた処理実行条件に基づいて設定の変更が行われている必要がある。

設定の変更が忠実に行われていれば、適応処理判断機構にて正しい設定の検索が行えている事を示す。また、実行されるべき設定の変更が正しく選択されているということは本システムが取得した情報も正しいことを示す。

今回は、TCP ソケットを用いて MSOCK からの情報をエミュレートしたため、MSOCK 自体の評価は行っていない。しかし、MSOCK は SFCMIP において移動検知などに用いられており、正しく情報を取得できている。よって、本システムにて利用した場合も正確に情報が取得できるはずである。また、MSOCK は本論文にて新たに実装された機構ではないため、評価の対象としなかった。

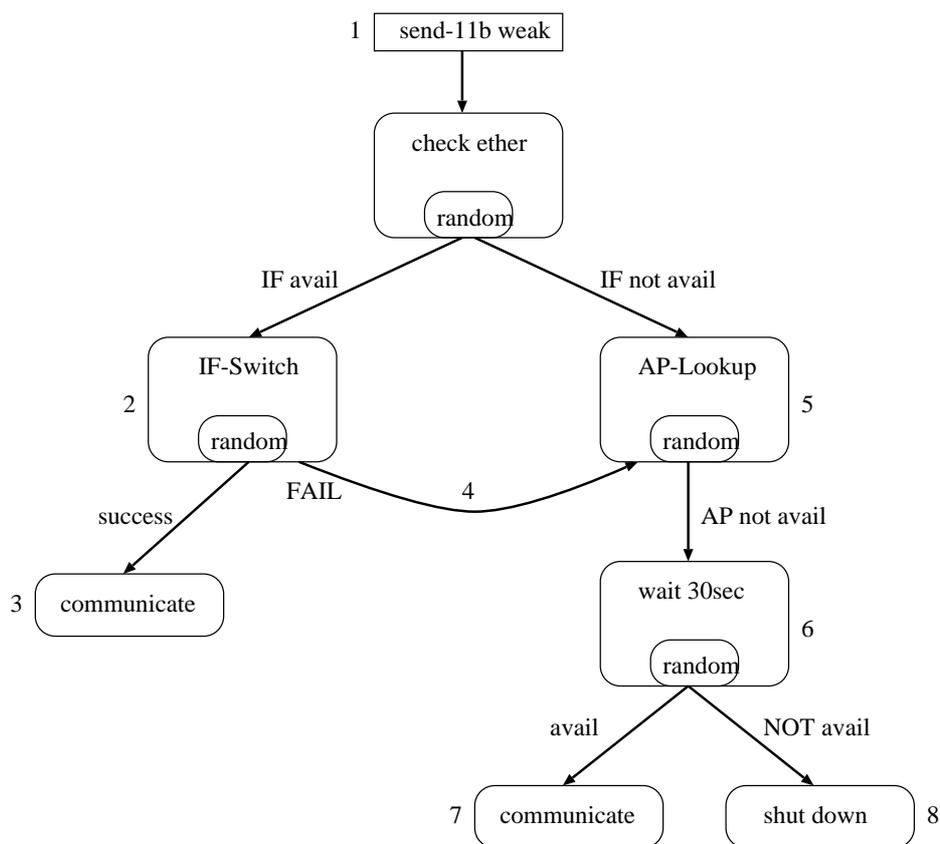


図 8.2: 評価の流れ

以上のことを踏まえ、評価を行った。評価方法は、第 8.1.2 節で述べた処理実行条件を本システムに与え、移動体計算機上で実行する。そこへ、無線電波強度が弱くなっていることを意味する情報を本システムの TCP ソケットに対して送信することで処理を始める。

一度処理を開始すると、各処理にてランダムに生成される情報を基に移動体計算機の状態が推移して行く。よって、移動体計算機における処理の流れを確認することで本システムが全ての条件を満たしているか評価した。

同様の処理を 10 回に渡って実行した結果、本システムは全ての状況において利用者によって設定された処理実行条件に基づいた設定の制御を行った。よって、本システムは全ての機能要件を満たすシステムであると言える。

8.2 定量的評価

定量的評価として、本システムを用いて SFCMIP が提供している複数インターフェース支援機構の制御を行い、通信再開までにかかる時間を測定した。

8.2.1 実験環境

本システムの定量的評価を行った環境を図8.3に示す。移動体計算機(MN)はIEEE802.3 100M Ethernet 通信インターフェース及びIEEE802.11b 無線通信インターフェースを所有している。また、それぞれのインターフェースはForeign Linkに接続している。

MNのIEEE802.3 100M Ethernet 通信デバイスが接続されているネットワークでは、Router Advertisementの送信間隔は最短で3秒、最長で4秒、そしてlifetimeは6秒に設定されている。

それに対してIEEE802.11b 無線通信インターフェースにて接続を行っているネットワークでは、Router Advertisementの送信間隔、及びlifetimeは全てdefaultの値を用いられているため、送信間隔の指定は最長の600秒のみである。そして、lifetimeも標準の259200秒(30日)となっている。

このような環境にて、MNはIEEE802.3 100M Ethernet 通信インターフェースとIEEE802.11b 無線通信インターフェースを切り替えながら通信を行う。

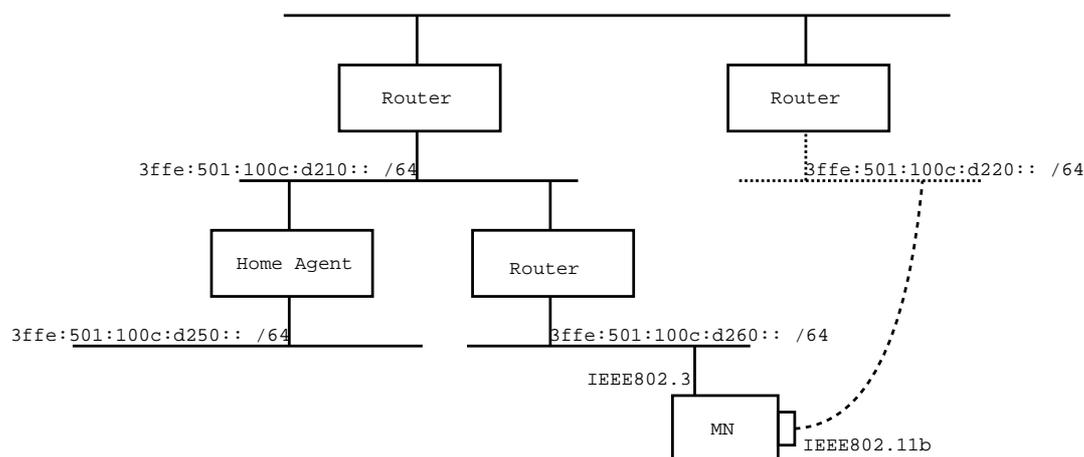


図 8.3: 評価を行ったネットワークの構成

また、MN及びホームエージェントの構成を表8.2.1に示す。

表 8.2: MN及びHAの構成

	CPU	Memory	OS
MN	Celeron 500Mhz	192MB	FreeBSD 4.6.2-RELEASE with SFCMIP-Oct15
HA	Pentium 166Mhz	128MB	FreeBSD 4.6.2-RELEASE with SFCMIP-Oct15

8.2.2 実験内容

本システムのように，カーネル空間の情報を一度ユーザーランドで取得し，状況を判断してから設定の変更を行う機構が，通信コストに及ぼす影響を評価する．

評価として SFCMIP に実装されている複数インターフェース支援機構を利用して，通信インターフェースの切り替えを行い，通信再開までにかかった時間を比較する．測定は，以下の2通りの通信インターフェース切り替え手法にて行った．

1. SFCMIP にあらかじめ実装されている通信インターフェース切り替え機構によって切り替える手法
2. 本システムを用い，SFCMIP に対して通信インターフェース切り替え命令を出し，切り替えを行わせる手法

両手法共に優先する通信インターフェースに，IEEE802.3 100M Ethernet 通信インターフェースを指定する．よって，IEEE802.3 100M Ethernet 通信インターフェースが利用可能な状況では，そちらを使うように通信インターフェースの切り替えを自動で行う．また，利用できない状況では，他の利用可能な通信インターフェースへの切り替えを試みる．今回の測定は，IEEE802.11b 無線通信デバイスは常に利用可能な状態にして行った．

以下に測定の手順を示す．

1. ping6 を用い，ICMP6 Echo Request(56byte) のフラッディングを HA に対して送信を開始する
2. IEEE802.3 100M Ethernet 通信デバイス利用不可能にする
3. IEEE802.11b 無線通信インターフェースを用いた通信の再開を確認し，再度 IEEE802.3 100M Ethernet 通信インターフェースを利用可能にする．通信が再開されたこと目安として，HA にて IEEE802.11b 無線通信インターフェースのアドレスを CoA とする BU が新たに登録されたかを監視した．
4. HA にて，IEEE802.3 100M Ethernet 通信インターフェースのアドレスを CoA とする BU が再び登録されたのを確認し，フラッディングを終了する

通信インターフェースの切り替えが完了するまでの時間は，フラッディングに対する ICMP6 Echo Reply を受信した時間を pentium counter を用いて取得することで算出可能にした．ICMP6 Echo Reply にて，sequence 番号が途中抜けている部分が通信インターフェースの切り替えを行っている時間であるとし，再びパケットを受信した時間から最後にパケットを受信した時間の差分を求め，通信インターフェース切り替えにかかった時間とした．

8.2.3 実験結果

第 8.2.2 項で述べた実験を，SFCMIP による通信インターフェース切り替えと本システムを用いた通信インターフェース切り替えに対し，それぞれ 30 回ずつ行った．

それぞれが通信再開までに要した時間の平均を表 8.2.3 に示す．

平均より本システムを用いた場合，どちらの切り替えに関してもほぼ同様の結果が得られていることが分かる．しかし，無線通信インターフェースから Ethernet 通信インターフェースへの切り替えを行った場合，その反対と比較して若干切り替えにかかる時間が長くなっていることが分かる．

また，SFCMIP における Ethernet 通信インターフェースから無線通信インターフェースへの切り替えに要する時間は，本システムを用いた場合と比較して長い．

一方，無線通信インターフェースから Ethernet 通信インターフェースへの切り替えに要する時間は SFCMIP，本システム共に短い．しかし，本システムを用いた場合の平均の方が若干長い．

これらの結果に関する考察は，第 8.2.4 項にて述べる．

表 8.3: 通信インターフェース切り替えにかかった時間の平均

処理 / 環境	SFCMIP[標準偏差]	SFCMIP with MMD[標準偏差]
IEEE802.3 to IEEE802.11b(msec)	1.502[0.260]	0.205[0.019]
IEEE802.11b to IEEE802.3(msec)	0.196[0.040]	0.292[0.019]

また，実験にて取得した結果から一つ抜粋し，図 8.4 に示す．SFCMIP を用いた通信では，Ethernet 通信インターフェースから無線通信インターフェースへの切り替えを行う際に約 1.22msec と，他と比較して大きな遅延が発生している．これは，平均を見ても分かるようにほぼ全ての測定において同様の結果が見られた．

一方，本システムを用いた測定結果は約 0.203msec と短かい．その結果として，SFCMIP と比較して短時間に多くの ICMP6 Echo Request を送受信でき，スループットが向上していることが分かる．

また，無線通信インターフェースから Ethernet 通信インターフェースへの切り替えに関しては，どちらも高速に切り替えが行えていることが分かる．図では本システムを用いた切り替えの方が SFCMIP よりも早く切り替えが行われていることが分かる．これは，平均とは反対の結果であり，これに関しても第 8.2.4 項にて考察を行う．

8.2.4 定量的評価に関する考察

第 8.2.3 項にて得られた結果を基に，本システムが通信インターフェース切り替えに及ぼす影響に関して考察を行う．

本項では，以下の項目に関して考察を行う．

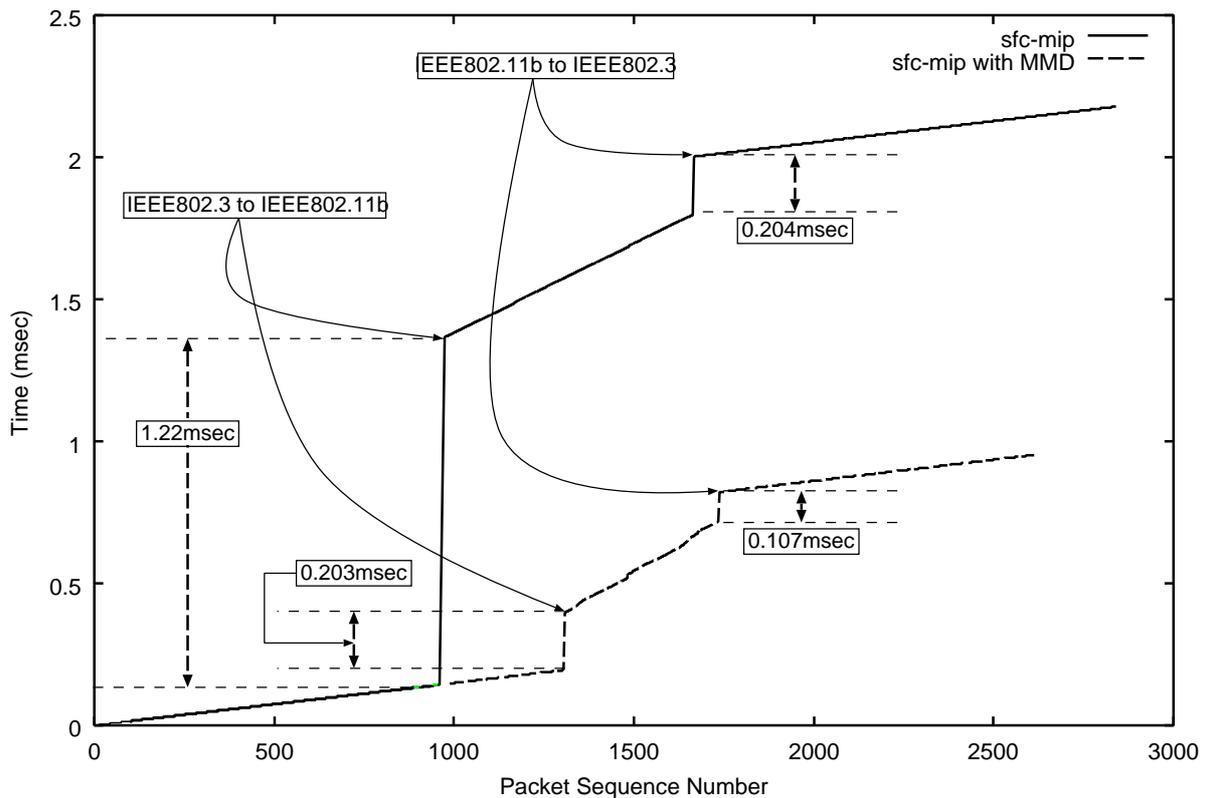


図 8.4: インターフェース切り替えに要する時間の比較

- SFCMIP において，Ethernet 通信インターフェースから無線通信インターフェースへの切り替えに要する時間が長いことについて
- 本システムを用いた場合，SFCMIP と比較して無線通信インターフェースから Ethernet 通信インターフェースへの切り替えに要する時間が長くなることについて
- 本システムを用いた場合，Ethernet 通信インターフェースから無線通信インターフェースへの切り替えにかかる時間よりも，無線通信インターフェースから Ethernet 通信インターフェースへの切り替えにかかる時間の方が若干長いことについて

SFCMIP における Ethernet 通信インターフェースから無線通信インターフェースへの切り替えについて

表 8.2.3 や図 8.4 から分かるように，SFCMIP では Ethernet 通信インターフェースから無線通信インターフェースへの切り替えにかかる時間が他と比べ長い。

これは，現在の SFCMIP の実装では Router Advertisement の受信を移動検知に用いているため，RA を受信するまでの時間によって大幅に左右されるためである。

今回の実験を行った環境では，無線通信インターフェースを用いて接続を行っている

るネットワークの Router Advertisement 送信間隔が 600 秒と非常に長い．そのため，SFCMIP は現在利用している通信インターフェースが利用不可能になったことを検知した瞬間に，無線通信インターフェースに対して Router Solicitation を送信するが，送信のタイミングによっては必ずしも

本システム利用時における無線通信インターフェースから Ethernet 通信インターフェースへの切り替えについて

本システムを用いた場合，SFCMIP による通信インターフェースの切り替えと比較して無線通信インターフェースから Ethernet 通信インターフェースへの切り替えにかかる時間が長い．

SFCMIP は，カーネル空間でリンクダウン情報を取得し，利用可能な通信インターフェースへの切り替えを行う．しかし，本システムでは一度情報をユーザーランドで取得し，ポリシーの検索を行ってから通信インターフェースの切り替えを行う．そのため，通信インターフェースの切り替えを実行させるまでに行わなければならない処理が多くなってしまい，結果として切り替え完了までにかかる時間が長くなってしまふ．

図 8.5 にて具体的に差が生じて居る点を示す．図中 (a)，(b)，(c) で表されている部分は本システムを用いた場合の処理オーバーヘッドとなり得る部分である．

(a) はユーザーランド空間にて情報を取得するのにかかる時間を示す．(b) は利用者によって設定された処理実行条件の検索にかかる時間を示し，(c) は (b) にて取得した処理をユーザーランド空間にて実行するのにかかる時間を示す．

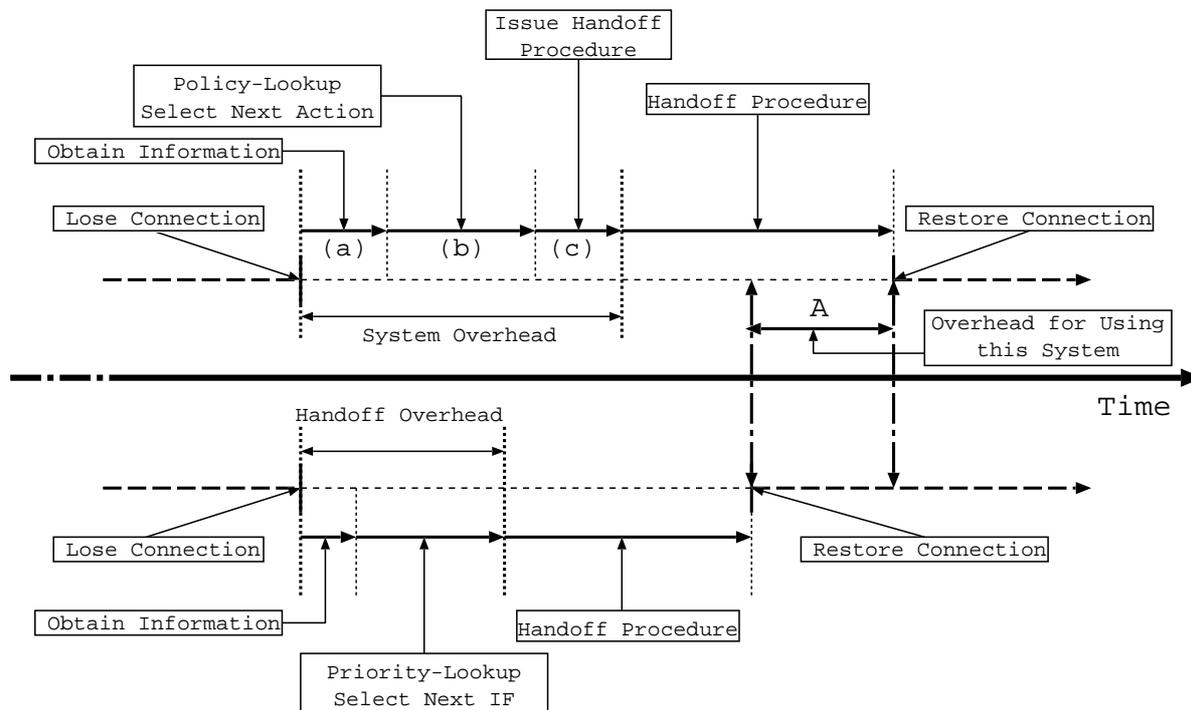
今回評価用に実装を行った通信インターフェースの切り替えを行う関数は，まず内部で現在通信に用いられているインターフェースの情報を取得する．次に現在利用可能な通信インターフェースの中から優先順位が最も高いものを選択し，切り替え命令を発行する．

切り替え命令の発行を行う部分は，SFCMIP が用意している外部コマンドである `mipconfig` を `system` 関数を用いて呼び出す．そのため，利用可能な通信インターフェースから優先順位の高いものを選択するためのオーバーヘッドだけでなく，内部で `system` 関数を呼び出すことによるオーバーヘッドや，`mipconfig` 内で利用されている `ioctl` を用いるオーバーヘッドなど，カーネル空間で切り替えを行った場合と比較して処理オーバーヘッドが高い．

よって，カーネル空間で全てを処理している SFCMIP では，ユーザーランド空間で全てを管理している本システムと比較して，(a) 及び (c) に関してそれぞれ処理オーバーヘッドが小さいと言える．また (b) の処理実行条件の検索に関しては，SFCMIP が次に利用する通信インターフェースの選択しか行わないのに対し，他の様々な処理から一つの処理を選択し，実行しなければならない本システムの方が処理コストが高くなることが予想される．

このように，(a)，(b)，(c) それぞれにおける処理オーバーヘッドの差が図中 A で表されている通信再開までの時間の差となる．今回の評価にて，無線通信インターフェースから Ethernet 通信インターフェースへの切り替えにかかった平均時間の差は，図中

MMD:



SFCMIP:

図 8.5: 本システムと SFCMIP の処理オーバーヘッドの比較

A で表すことができる。

まとめると、本システムを用いた場合、SFCMIP のようにカーネル空間で全てを処理している機構よりも、処理オーバーヘッドが大きくなってしまふのは、ユーザーランド空間で全ての処理を行っていることが一番大きいと思われる。次に、管理しなければならない処理が多岐に渡るため、実行すべき処理の検索にかかる処理オーバーヘッドも大きくなる。

無線通信インターフェースから Ethernet 通信インターフェースへ切り替えを行った際に、その反対と比較して切り替えにかかる時間が長いことについて

本システムを利用して通信インターフェースの切り替えを行った場合、無線通信インターフェースから Ethernet 通信インターフェースへの切り替えにかかる時間の方がその反対と比較して長い。

これは、Ethernet 通信インターフェースから無線通信インターフェースへの切り替えを行う場合、無線通信インターフェースは常時利用可能であり、アドレスを保持しているからである。それに対して無線通信インターフェースから Ethernet 通信インターフェースへの切り替えを行う際には、Ethernet 通信インターフェースのリンクが利用

可能な状態になった後にアドレスの取得を行わなければならない。そのため、リンク情報を用いて通信インターフェースの切り替えを行っている本システムでは、切り替えを行ってからアドレスの取得が完了するまでの時間分、無線通信インターフェースから Ethernet 通信インターフェースへの切り替えに時間がかかる。

8.2.5 定量評価まとめ

定量評価では、ユーザーランド空間にて通信インターフェースの切り替えの管理を行っている本システムと、カーネル空間にて管理を行っている SFCMIP の、通信インターフェース切り替え時に要する時間について評価を行った。

結果、本システムを用いて通信インターフェースの切り替えを行った場合、Ethernet 通信インターフェースから無線通信インターフェースへの切り替えを行った場合と、無線通信インターフェースから Ethernet 通信インターフェースへ切り替えを行った場合との差は約 0.1 ミリ秒であった。よって、本システムでは、切り替えを行う内容によって処理速度に偏りが生じることはないことが分かった。

また、SFCMIP との比較では、無線通信インターフェースから Ethernet 通信インターフェースへの切り替えにおいて SFCMIP の方が約 0.1 ミリ秒と小さい。よって、ユーザーランド空間で切り替えの処理を行わせた場合でも通信に大きな支障をきたすことなく設定の制御を行えることが可能であることが分かった。

8.3 評価まとめ

定性的評価及び定量的評価を行い、本システムは利用者によって設定された内容と移動体計算機の状態情報に基づいて通信の設定を制御することが可能であることと、通信の設定の制御を行う際に、カーネル空間で処理した場合と処理オーバーヘッドにおいて大きな差がないことが分かった。よって、本システムは移動体通信環境において移動体計算機の通信の制御を行うために必要な条件を満たしていると言える。

本システムを用いることによって、利用者は多様化する移動体通信環境において状態情報に基づいて自動で通信の設定を行えるようになるだけでなく、意志に基づいた処理を実現することが可能となる。

第9章 結論

9.1 まとめ

本研究により，移動体計算機の通信の設定を，移動体計算機の状態情報に基づいて制御することが可能となった．通信の設定の制御は，利用者があらかじめ指定しておいた処理実行条件に沿って選択される．本システムを有効に活用するためには，利用者が制御の対象としている通信技術に対して，本システムから制御を行うためのインターフェースを用意する必要がある．しかし，既に外部の機構から処理を制御できる通信技術に関しては本システムを拡張することで，容易に対応することができる設計とした．

本システムにより，移動体計算機での通信において，利用者が理想とする通信技術の設定を自動で行えるようになった．これにより，既存の移動体通信技術やトランスポート層の技術を自由に組み合わせることが可能となり，移動体通信環境における効率的な通信を実現する．

9.2 今後の課題

今後の課題として，より利用者に対して設定のしやすい抽象化手法の模索，処理オーバーヘッドの軽減，リモート・ホストとの協調動作の評価，リモート・ホストとの情報交換におけるセキュリティの向上等が挙げられる．

より高い次元での抽象化を実現することによって，現在大まかに定義されている通信の効率化に関して一定の法則を見つけ出す．これにより，通信の効率化を行う処理実行条件を自動で生成できるようになることを目指す．

また，処理オーバーヘッドを軽減することによって通信の再開までにかかる時間を短縮を図る．リモート・ホストとの協調動作に関する研究を行うことで，移動時の通信の最適化の可能性も模索する．

最後に，リモート・ホストとの情報交換におけるセキュリティの向上を行い，既存の移動体通信に対するデメリットを排除する．

謝辞

本論文の作成にあたり，ご指導頂きました慶應義塾大学環境情報学部教授 村井純博士並びに同学部教授 徳田英幸博士，同学部助教授 楠本博之博士，同学部助教授 中村修博士，同学部専任講師 南政樹氏，同学部専任講師 重近範行氏に感謝致します．

絶えずご指導とご助言を頂きました慶應義塾大学政策・メディア研究科特別研究専任講師 植原啓介氏，同大学大学院政策・メディア研究科後期博士課程 湧川隆次氏，同大学院修士課程 三屋光史朗氏，若山史朗氏に感謝致します．

本論文の作成にあたり，多大なるご協力を頂いた株式会社三菱総合研究所 佐藤雅明氏，慶應義塾大学大学院政策・メディア研究科後期博士課程 今泉英明氏，同大学環境情報学部の久松剛氏，成瀬大亮氏，岡田耕司氏，久松慎一氏に感謝致します．

また，本論文の作成期間中に楽しい話題を提供して下さった慶應義塾大学大学院政策・メディア研究科 石原知洋氏，同大学環境情報学部の千代佑氏，中村懿妹氏に感謝致します．

本研究を進めて行く上でご支援下さった慶應義塾大学徳田・村井研究室並びに同研究室 NACM 研究グループの皆様には感謝の念を表します．

参考文献

- [1] D. Johnson and C. Perkins. *Mobility support in IPv6*, 2001. draft-ietf-mobileip-ipv6-19.txt.
- [2] LIN6 Working Group. *LIN6 WWW page*. <http://www.lin6.net/>.
- [3] Rajeev Koodli. *Fast Handovers for Mobile IPv6*, 2002. INTERNET DRAFT, draft-ietf-mobileip-fast-mipv6-05.txt.
- [4] Hesham Soliman, Claude Castelluccia, Karim El-Malki, and Ludovic Bellier. *Hierarchical MIPv6 mobility management (HMIPv6)*, 2001. INTERNET DRAFT, draft-ietf-mobileip-hmipv6-04.txt.
- [5] Zygmunt J. Haas. *Mobile TCP: An Asymmetric Transport Protocol Design for Mobile Systems*, 1997.
- [6] UCLA CS Department. *TCP Westwood Home Page*. <http://www.cs.ucla.edu/NRL/hpi/tcpw/>.
- [7] IEEE802.11WG. *IEEE 802.11 WIRELESS LOCAL AREA NETWORKS*. <http://grouper.ieee.org/groups/802/11/>.
- [8] 湧川隆次. *MobileIPv6を用いた複数インタフェース通信機構の設計と実装*, 2000.
- [9] Internet ITS Project. *Internet ITS WWW page*. <http://www.internetits.org/>.
- [10] IPv6 Promotion Council. *IPv6 Promotion Council WWW page*. <http://www.v6pc.jp/>.
- [11] 三屋光史朗. *インターネット自動車における車載ルータの設計と実装*, 2001.
- [12] 財団法人 道路交通情報通信システムセンター. *VICS Home Page*. <http://www.vics.or.jp/>.
- [13] Thierry Ernst, Alexis Olivereau, Ludovic Bellier, Claude Castelluccia, and Hong-Yon Lach. *Mobile Networks Support in Mobile IPv6 (Prefix Scope Binding Updates)*, 2002.

- [14] Peter H. Dana and Department of Geography. *Overview of the Global Positioning System (GPS)*. <http://www.fsfeurope.org/coposys/index.en.html>.
- [15] IEEE 1073 Medical Device Communications. *IEEE 1073 WWW page*. <http://www.ieee1073.org/meetings/minutes/1998-04-Baltimore/1998-04-Baltimore.html>.
- [16] C. Rose and R. Yates. Ensemble polling strategies for increased paging capacity in mobile communications networks. *C. Rose and R. Yates. Ensemble polling strategies for increased paging capacity in mobile communications networks. ACM Wireless Networks, 1996. (in press).*, 1996.
- [17] The FreeBSD Project. *FreeBSD WWW page*. <http://www.freebsd.org>.
- [18] NEMO (Network in Motion). *SFC-MIP6 WWW page*. <http://neo.sfc.wide.ad.jp/mip6/>.
- [19] Internet society. *IETF Home Page WWW page*. <http://www.ietf.org/>.
- [20] 山本和彦. Ip トンネルのモデル化と実装. コンピュータ・ソフトウェア、第 15 巻、第 2 号、pp. 38 - 47, 1998.

付録A Mobile IPv6(MIPv6)

MIPv6 は、インターネット通信技術の標準化を行っている機関である Internet Engineering Task Force(IETF)[19] にて標準化に向けて研究が行われている移動体通信支援技術である。

A.1 MIPv6 で解決される問題

以下に本技術で解決することができる移動体通信における問題点とそれぞれに関する説明を示す。

- 移動透過性：
インターネット上の一意の識別子である IP アドレスの変化を IP 層で隠蔽し、トランスポート層等の上位層では一意の IP アドレスでの通信を実現する。よって、移動体計算機のネットワーク間移動に伴う IP アドレスの変化による通信セッションの遮断は発生しない。
- 着信可能性：
ネットワーク間の移動は、移動体計算機の IP アドレスの変化を伴うので、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。よって、一意の IP アドレスにて移動体を識別できるようにすることで通信相手は移動体との通信を継続させることが可能となる。

A.2 MIPv6 の用語

以下に MIPv6 で定義されている用語の解説を行う。本論文では、以後それぞれの用語を括弧内の形で利用する。

- Mobile Node(MN)：
リンクからリンクへ移動するノード
- Correspondent Node(CN)：
MN と通信する相手の固定ノード及び MN。
- Care of Address(CoA)：
MN が移動先リンクで取得する IP アドレス。

- Home Link(ホームリンク) :
MN のホームアドレスと同じプレフィックスを持つネットワーク
- Foreign Link(移動先リンク) :
MN のホームリンク以外のリンク
- Home Address(ホームアドレス) :
MN がホームリンクにて取得する IP アドレス .
- Binding(binding) :
MN のホームアドレスと CoA の対応とその対応の有効期間を示すもの .
- Home Agent(HA) :
MN の binding が登録されているホームリンク上のルーター . MN がホームリンクから離れている場合 , HA は MN のホームアドレス宛のパケットを傍受し , MN の CoA に向けてトンネル [20] を用い転送する .

A.3 MIPv6 の動作概要

図 A.1 に MIPv6 の大まかな処理の流れを示す . MIPv6 では , MN はホームアドレスを用いて CN と通信を行う . よって , MN は移動先リンクへ移動し新たに CoA の取得を行った時点で HA 及び移動前に通信を行っていた CN に対して binding の更新を行う . しかし , CN が移動前に MN と通信を行っておらず , MN の binding を所有していない場合は図中 Packet A の用に CN からのパケットは処理される . ここで HA は MN のホームアドレス宛に送信されたパケットを傍受し , MN の CoA にトンネルを用いて転送する . トンネル越しにパケットを受信した場合 , MN は CN に対して binding を送信する . CN が binding を受信した後の通信は Packet B の処理と同様になる . Packet B は , CN が MN の binding を所有していた場合を示す . CN は MN の CoA に対して直接パケットを送信する .

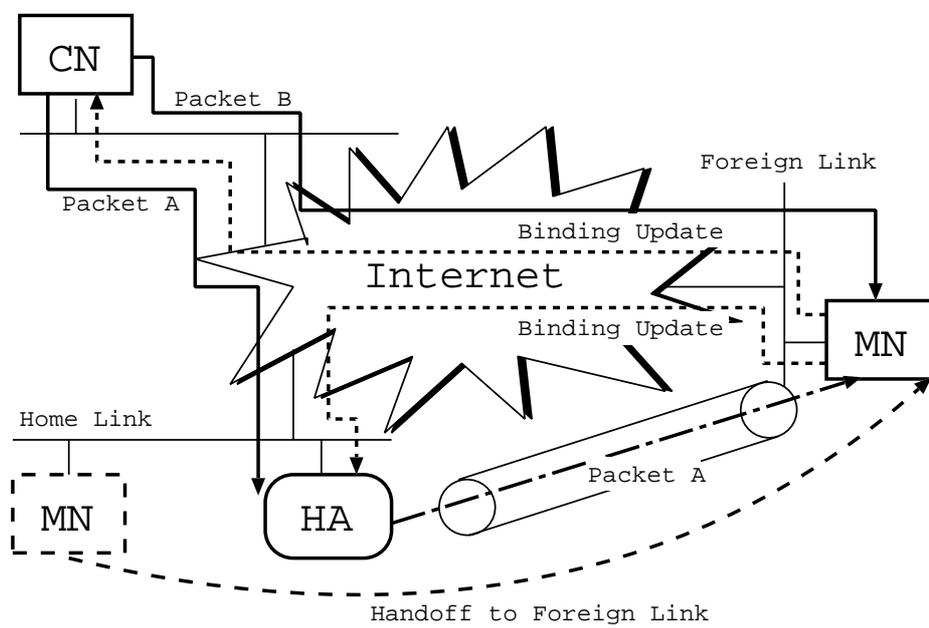


図 A.1: MIPv6 における移動時の処理の流れ