

移動計算機情報発信環境のための Toolkit の設計と実装

田頭 茂明[†] 安田 修[†],
最所 圭三[†], 福田 晃[†]

移動計算機から情報発信を行う場合は、従来の固定計算機からの情報発信とは異なり、ネットワーク帯域の狭さや、ネットワークから分断する等の移動計算機特有の問題が生じる。このため移動計算機環境においては、(1) ネットワークとの様々な接続の状態（接続や分断等）においてサービスを安定して提供する機構、(2) ネットワーク帯域に応じて情報発信を抑制し、優先度に応じて情報を発信する機構、(3) 発信する情報や環境に応じた柔軟な通信方式を実現する機構、が必要となる。我々は、これらの機構を実現する移動計算機のための情報発信システムを構築している。本情報発信システムでは、移動計算機を考慮していない種々のアプリケーションを、移動計算機情報発信環境へ柔軟に対応させるための枠組みである情報発信 Toolkit を提供する。本論文では、情報発信 Toolkit の設計と実装について示した。さらに、本 Toolkit を用いて実際にいくつかのアプリケーションを移動計算機環境へ容易に適応できることを示すとともに、本 Toolkit により優先度に応じた効率的な情報発信を行うことができることを示した。

Design and Implementation of a Toolkit for Mobile Information Announcement Environment

SHIGEAKI TAGASHIRA,[†] OSAMU YASUDA,[†] KEIZO SAISHO[†],
and AKIRA FUKUDA[†]

In order to announce information from a mobile computer, the problems of narrow network bandwidth and disconnection from network have to be considered. In a mobile environment, the following three mechanisms are needed: (1) providing stable services under any network condition, (2) limiting announcing information with given network bandwidth, and announcing in order of priority of the information, and (3) transferring information using an appropriate protocol for any type of information and environment. Therefore, this paper proposes an information announcement system providing the mechanisms and a toolkit which can adapt a wide variety of network applications to the mobile environment smoothly. In this paper, the toolkit is designed and implemented. Moreover, we show that some applications can be constructed very easily and the mechanism provided by the toolkit can effectively announce information in order of its priority.

1. はじめに

計算機の小型化、高性能化により、携帯できる高性能計算機（移動計算機）が普及している。また、移動計算機を取り巻くネットワーク環境も変化し、様々な通信媒体を用いて移動先でネットワークと接続できる

ようになっている。これにより移動計算機を用いて、場所に依存しないで情報の獲得、編集、共有が可能な環境の構築が期待される。

移動計算機に関するネットワークの研究は、移動先からのネットワークを利用した情報取得に関するものが多い。しかしながら、情報取得ならびに情報編集の現場である移動先において、最新の情報を直接ユーザに提供することが可能となっており、移動先での情報発信も重要な検討課題である。移動先から情報発信できれば、生中継放送等のアプリケーションを、個人レベルで簡単に実現することが可能である。移動計算機に情報受信と情報発信を可能にする環境が整備され、さらにマルチメディアを扱うことができれば移動計算機を用いた移動型の遠隔医療システムや、携帯型のイ

[†] 奈良先端科学技術大学院大学情報科学研究科

Graduate School of Information Science, Nara Institute of Science and Technology

現在、富士通株式会社

Presently with FUJITSU LIMITED

現在、香川大学工学部信頼性情報システム工学科

Presently with Department of Reliability-based Information Systems Engineering, Faculty of Engineering, Kagawa University

ンターネット TV 電話システム等を実現できる。

我々は、文献 1) において WWW を基盤にした移動計算機からの情報発信システムを構築した。このシステムは、WWW に特化したシステム構造になっており、WWW のソフトウェア自体を修正している。このため他のアプリケーションはこのシステムへ容易に適用できない。本システムにおいては、移動計算機環境の問題点である通信の分断および不安定さに対処するシステムソフトウェアとして情報発信 Toolkit を提供する。この Toolkit を用いることにより、移動計算機の各状態に対応する処理を記述するだけで、既存の多くのアプリケーションを容易に移動計算機環境へ適用できる。情報発信 Toolkit は、(1) ネットワークとの様々な接続の状態（接続や分断等）においてサービスを安定して提供する機構、(2) ネットワーク帯域に応じて情報発信を抑制し、優先度に応じて情報を発信する機構、(3) 発信する情報や環境に応じた柔軟な通信方式を実現する機構、を提供している。また新しいアプリケーションを作成する場合には、本 Toolkit を用いることにより開発工期を短縮することが可能となる。

本論文では、移動計算機環境における情報発信の問題について議論し、提案システムがそれらの問題をどのように解決するかを示す。また、情報発信 Toolkit の設計と実装について示し、本 Toolkit を用いたアプリケーションの構築例について述べ評価を行う。

2. 移動計算機からの情報発信

2.1 研究目的

固定計算機を対象としたネットワークアプリケーションは数多く存在する。しかし、これらは移動計算機を対象としていないので、2.2 節で指摘する問題により、そのまま移動計算機環境で用いることができない。さらにこれらアプリケーションの中には、バイナリで配布され修正できないアプリケーションが多く存在する。したがって、既存の固定計算機を対象に作成されたアプリケーションを、そのまま移動計算機環境で利用できる環境が必要である。

本研究においては、以下のようなアプリケーションを容易に実現できる移動計算機からの情報発信システムの構築を目指している。

- WWW を用いた移動計算機からの情報発信：
移動計算機がネットワークと接続中は移動先で取得、編集した情報をすぐに発信する。また、分断中でもサービスを継続するために、バックボーンに存在する代替ホストが移動計算機の情報キャッチ

シュシ、代わりに情報を発信する。さらに移動計算機の通信帯域の狭さや不安定さに対処するために、通信における resume 機能を付加し、効率的な発信を可能にする。

- 移動計算機を用いた携帯 TV 電話：
移動計算機が分断中で通話できない場合、留守番電話サービスを提供する。また、通話中においては、ネットワーク帯域を考慮し帯域に応じた品質で通信する。
- アドホックネットワークを用いた会議システム：
移動計算機の新しいネットワーク形態として、移動先でネットワークを一時的に構築するアドホックネットワークが普及している。アドホックネットワークを利用し、移動先でプレゼンテーションのための OHP データや参考資料を互いに交換し、会議を行う。

2.2 移動計算機環境における情報発信

移動計算機からの情報発信においては、既存の固定計算機上での情報発信とは異なり、以下に示す問題が発生する。これらの問題に対処するための必要な機能を議論する。

(1) 接続の状態の管理：移動計算機は、ネットワークと接続する位置が不特定であり、ユーザの意図または環境によって、その接続の状態も著しく変化する。移動計算機から情報を発信する場合、移動計算機の接続先での位置と状態を管理する必要がある。移動計算機の位置の問題は、IETF により Mobile-IP²⁾ が提案されており、移動計算機の位置透過性が実現され解決されている。しかし、移動計算機のネットワークとの接続の状態は考慮されていない。接続の状態を考慮しなければ、分断等の移動計算機特有の状態において情報を発信できない。接続の状態にかかわらず移動計算機からの安定したサービスを実現するためには、移動計算機の接続の状態を管理し、その状態に対処する機構が必要である。また移動計算機の状態の管理において、移動計算機をサーバとする場合と、クライアントとする場合とでは管理する場所や機構が異なる。移動計算機をクライアントとする場合は、移動計算機を利用しているユーザに対して透過なサービスを実現する。このため移動計算機上における接続の状態の管理が重要である。しかし移動計算機をサーバとする場合では移動計算機の情報を利用する他の計算機のユーザに対して透過なサービスを実現する。この場合、他の計算機において移動計算機の状態を管理する機構が重要となる。

(2) 分断状態における代替処理：移動計算機における

ネットワークとの接続の状態として、大きく接続状態と分断状態が存在する。接続状態では移動計算機から情報を発信することができる。しかし、分断状態においては、移動計算機は情報を発信できない。移動計算機をサーバとしてとらえた場合、サービスを継続することは重要である。このため、接続の状態を考慮した処理が必要である。(1)で述べた接続の状態の情報を利用して実現できる。また分断時処理はアプリケーションに大きく依存するため、種々のアプリケーションに対応できる枠組が必要である。分断時処理の例として、WEBを利用した情報発信において、移動計算機上に存在する情報のコピーの提供や、携帯TV電話における留守番電話サービス等が考えられる。

(3) 通信帯域の有効利用：接続状態においても移動計算機の通信帯域は変化する。通信帯域は、移動計算機における情報取得の問題だけでなく、情報発信においても問題となる。通信帯域が狭い場合には、その帯域を有効利用するために、対象とする通信範囲、データの種類の種類、通信におけるアプリケーションの意図を考慮した柔軟な通信方式が必要である。たとえば、

(a) 一般に広域通信では、パケットロスが多い。このような場合、通信の保証に重点を置き、TCPを基盤にして resume 機能を組み合わせて通信する。

(b) 一般にLAN通信では、パケットロスが小さい。TCPは通信のオーバーヘッドが大きいので、このような場合UDPを基盤にして再送機能、resume機能を組み合わせて通信する。

(c) MPEG等の動画データを実時間で転送する場合は、通信の遅延をなくすために再送のないUDPを基盤にして通信する。そして同期のためにタイムスタンプを付加し、また重要な部分(たとえばIフレーム)に対しては、再送しなくてもパケット復元することが可能なパリティパケット³⁾を利用し通信する。

(d) (c)におけるデータの種類の同一でも、アプリケーションが意図する通信方式が異なる場合もある。MPEGの通信に対して、画質に重点を置くアプリケーションもある。

等が考えられる。このように、対象とする通信範囲やデータの種類の種類を考慮するアプリケーション指向の通信方式が必要である。

(4) 複数のアプリケーションからの同時発信：(3)と同様に情報取得と情報発信共通の問題である。複数のアプリケーションから同時に発信するとき、クライアントは移動計算機の性能や通信帯域以上のものを要求することがある。この場合、同時に処理するアプリケーションの数を制御する等の処置が必要である。また、

動画等の時間制約を持つデータを処理するアプリケーションを扱う場合、そのアプリケーションを優先して処理することも考えなければならない。

3章では以上の問題に対応できる情報発信システムの設計を行う。

2.3 関連研究

アプリケーションを移動計算機環境に適応(Adaptation)するために、様々なミドルウェアやToolkitが提案されている。また、文献4)、5)では、移動計算機環境に適応する様々なシステムを以下に示す3種類に分類している。

(a) Laissez-Faire Adaptation

(アプリケーションだけで適応)

(b) Application-Transparent Adaptation

(システムだけで適応)

(c) Application-Aware Adaptation

(システムとアプリケーションで適応)

(a) Laissez-Faire Adaptationは、ミドルウェア等のシステムを利用せずに、アプリケーションが独自に移動計算機環境へ適応する手法である。この手法では、OSを選ばず、アプリケーションが意図する、移動計算機環境への適応を実現できる。しかし、アプリケーションが移動計算機の資源を独自に管理し適応するため、複数のアプリケーションを同時に使用した場合、それらのポリシーの競合が発生し、個々のアプリケーションが意図した効果が現れない。このようなアプリケーション例としては、商用ソフトであるEudora⁶⁾等がある。

(b) Application-Transparent Adaptationは、アプリケーションを介在せずに、ミドルウェア等のシステム部分だけで移動計算機環境へ適応する。Application-Transparent Adaptationの利点は、アプリケーションを変更せずに利用できることである。また、システム部分が移動計算機の資源を管理するため、適応ポリシーの競合を回避し、複数のアプリケーションを同時に実行できる。しかし、適応において、アプリケーションを考慮しないために、アプリケーションはシステムが決定する適応ポリシーを利用するしかない。アプリケーションによっては自身のポリシーと異なる場合がある。

Application-Transparent Adaptationの研究例として、Codaファイルシステム^{7),8)}が提案されている。Codaファイルシステムは、移動計算機の長期間の分断のための分断時処理(Disconnected Operation)をサポートしている。分断中に使用されるとされるサーバ上のファイルをあらかじめ移動計算機上にキャッシュ

することにより、分断中においてもネットワークと接続中の環境と同等の環境をできる限り実現する。また、移動先でキャッシュを編集することを考慮し、元のサーバ上のファイル間の整合性を保つ機構が提供される。しかし、Coda ファイルシステムにおける分断時処理はファイルシステムを提供するだけであり、様々な分断時のための適応ポリシーを支援できない。

(c) Application-Aware Adaptation では、アプリケーションごとに用意される適応部分とシステム部分が協調し、アプリケーションを移動計算機環境へ適応する。これにより種々のアプリケーションの意図に応じた適応を実現できる。システム部分はネットワークの環境や移動計算機の資源を管理し、その情報を適応部分に伝える。適応部分は、その情報に基づいてアプリケーションを移動計算機環境に適応する。

Application-Aware Adaptation の研究例として、Odyssey⁵⁾が提案されている。Odyssey では、移動計算機における資源を管理する Viceroy と各アプリケーションの適応を実現する Warden で構成される。Viceroy は移動計算機の現在の資源情報を Warden に通知し、移動計算機の環境に応じて、Warden がアプリケーションに最適となるようにデータの品質やシステムの構成を柔軟に適応する。同様のシステムとして Rover⁹⁾が提案されている。Rover では、クライアントとサーバの機能を動的に配置できる RDO (Relocatable Dynamic Object) と、移動計算機が分断時も継続処理を可能にするために、non-blocking なリモート手続き呼び出しを実現する QRPC (Queued Remote Procedure Call) の 2 つの機構を提供することにより、アプリケーションを移動計算機環境へ適応する。RDO を用いて、サーバの機能をクライアントに配置できるので、種々のアプリケーションに対して、分断状態に対応する機能を提供している。しかし、Odyssey や Rover の RDO を利用するには、アプリケーションの変更が必要となる。

従来のアプリケーションを変更せずに利用するために、Proxy を用いる Application-Aware Adaptation の研究が提案されている。BARWAN プロジェクト¹⁰⁾は、通信するデータの種別を考慮し、そのデータの品質を通信帯域に応じて変化させ、移動計算機環境へ適応する。データの品質を変化させる機構は、データの種別によって異なる。Proxy において、様々なデータの種別に対して Transcoders と呼ばれる品質を変更する機構を用意し、各データに最適に適応している。しかし、このシステムでは移動計算機の分断状態を考慮していない。

以上で述べたシステムでは、我々の目的とする移動計算機をサーバとしてとらえ、アプリケーションを変更なしに移動計算機情報発信環境へ適応することを実現できない。文献 1) において、移動計算機からの情報発信システムを提案した。このシステムでは移動計算機の分断時処理を提供しているが、アプリケーションとして WWW だけを支援している。本論文で提案するシステムで重要な点は、移動計算機をサーバとしてとらえ、アプリケーションの意図に応じたサーバのための分断時処理を実現する機構と、従来のアプリケーションをそのまま利用する機構を提供することである。

3. 情報発信システムの設計

本章では移動計算機からの情報発信システムの設計について示す。さらにそのシステムを構築するための情報発信 Toolkit について述べる。情報発信 Toolkit は、ネットワークの接続の状態を考慮し、2.2 節で述べた移動計算機環境における問題を解決する枠組みをアプリケーションに提供するシステムソフトウェアである。アプリケーション開発者は、接続の状態に対処する処理を記述した MAPM (Mobile Application Processing Module) を追加するだけで、移動計算機環境の問題をほとんど意識することなく、既存のクライアント・サーバアプリケーションをそのまま利用できる。

3.1 移動計算機のための情報発信システム

我々の情報発信システムでは、既存のネットワークアプリケーションを変更しないで利用するために、現在最も広く普及しているクライアント・サーバモデルに添ったシステムモデルを採用する。提案情報発信システムの構成を図 1 に示す。このモデルでは、移動計

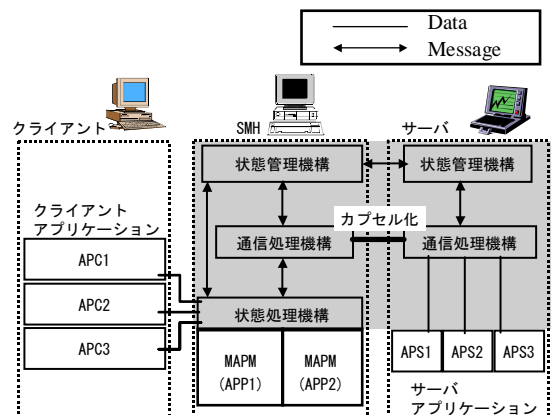


図 1 情報発信システムの構成

Fig. 1 Overview of information announcement system.

算機上でサーバアプリケーションが稼働し情報を発信する。本システムモデルは、集中管理型モデル（移動計算機上の情報を、固定ホスト上に蓄積し、固定ホストから情報を発信するモデル）に比べ、以下の利点を有する。

- 集中管理型モデルでは中央のサーバによって情報発信が制約される場合があるが、本モデルでは移動計算機がサーバとなるため自由な情報発信が可能である。
- 移動計算機上のサーバが単独で情報発信する機能を持つため、バックボーンネットワークに接続しないアドホックネットワーク環境においても、本システムを利用できる。

また、提案するシステムは、クライアントとサーバ間に移動計算機を管理する Proxy を置く。この Proxy を本システムでは、System Management Host (SMH) と呼ぶ。SMH の具体的な機能は次節で示す。Proxy を基盤にすることで以下の利点を有する。

- SMH が移動計算機環境に対応する機能を提供することで、種々のアプリケーションを変更なしに利用できる。
- 移動計算機が分断しているときに、SMH をサーバの分断時処理をする代替ホストとして利用するため、分断時処理を支援できる。
- 種々のアプリケーションの分断時処理をサポートするため、SMH に MAPM を用意することで、様々なアプリケーションに応じた分断時処理が実現できる。これは、アプリケーションを考慮し移動計算機環境へ適応する Application-Aware Adaptation を実現している。

このように SMH を用いることで、我々の目的とする従来のアプリケーションを変更なしに利用できる。しかし、2.3 節で述べた Odyssey や分断時に対応する Rover 等の、移動計算機環境を前提にアプリケーションを変更するアプローチと比較し、以下の欠点がある。

- 通信のプロトコルが公開されていないアプリケーションにおいては、細かな制御を行う機構を提供することができない。
- アプリケーションを変更しないために、システムの実装は冗長になる。たとえば、通信のプロトコル変換では、カプセル化により実現するため、そのオーバーヘッドが生じる。

3.2 情報発信 Toolkit の構成

図 1 の網掛け領域が提案する情報発信 Toolkit である。これと MAPM を組み合わせることにより情報発信システムを構築できる。APC はクライアントアプ

리케이션を示し、APS はサーバアプリケーションを示す。

SMH は、2.2 節で述べた、(1) 接続の状態の管理、(2) 分断状態における代替処理、(3) 通信帯域の有効利用、および、(4) 複数のアプリケーションからの同時発信を支援する機構を提供する。具体的には、移動計算機の状態を管理することで (1) を解決する状態管理機構、分断状態等の移動計算機の接続の状態に応じて処理することで (2) を解決する状態処理機構、通信帯域を考慮し複数のアプリケーションの通信を制御することで (3) と (4) を解決する通信処理機構を提供する。

SMH はクライアントから要求を受けると、状態管理機構から移動計算機の状態を取得し、その状態に応じて状態処理機構で MAPM に記述された処理を実行する。たとえば、移動計算機が分断状態なら、分断時の処理として SMH 上のコピーを提供する等がある。接続状態なら、通信処理機構を通じて要求を移動計算機上のサーバアプリケーションに転送し、アプリケーションで選択された、移動計算機と SMH 間のプロトコルを用いてデータを効率的に通信する。最後にクライアントはデータを取得する。通信中に通信品質が悪化し通信帯域が狭くなると、通信処理機構は、各アプリケーションの通信を制御し通信帯域に応じて負荷を下げる。

3 つの機構の詳細を以下に示す。

状態管理機構 移動計算機のネットワークとの接続の状態を保持する。状態管理機構は SMH と移動計算機上に存在し互いに協調して、3.3 節で説明する移動計算機の接続の状態を示す管理ステートを決定する。

状態処理機構 移動計算機の管理ステートに従った処理を行う。移動計算機の状態によりどのような処理を行うかはアプリケーションに依存する。このため、アプリケーション依存する処理はアプリケーション開発者に提供してもらう。この部分を MAPM と呼ぶ。MAPM は、状態管理機構で保持される移動計算機の管理ステートに対応する処理が記述されている。同一のアプリケーションプロトコルを用いる複数のアプリケーションで、MAPM を共有することも可能である。

通信処理機構 通信処理機構では、通信における以下に示す 2 つの機能を提供する。

- ・ 移動計算機と SMH 間の通信管理：

この 2 点間のプロトコルは、独自のプロトコルを構築することが可能であり、各アプリケーションは 3.4 節で提案するプロトコルを用いて、通常の

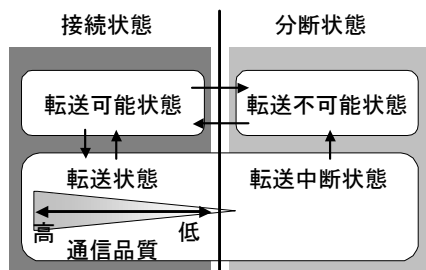


図2 管理ステートの概要
Fig. 2 Overview of administration states.

プロトコルをカプセル化し、アプリケーションの意図に添った通信方式を実現する。

・過度の発信の抑制：

過度の発信を抑制するために発信する情報数（コネクション数）を制限し通信を制御する。これには、3.5 節で説明するコネクション数の動的制御機構¹¹⁾を用いる。

3.3 管理ステート

情報発信 Toolkit では、移動計算機の接続の状態を、管理ステートを用いて表現する。図2に管理ステートを示す。管理ステートは、接続状態と分断状態だけでなく、通信品質も管理する。これにより無線 LAN のような通信品質が不安定な通信媒体において、そのときの通信状態に最適な処理を行うことができる。

- (1) 転送不可能状態
移動計算機がネットワークから分断している状態で、クライアントからの要求を受け付けることができない状態を示す。
- (2) 転送可能状態
移動計算機がネットワークに接続している状態で、クライアントからの要求を待っている状態を示す。
- (3) 転送状態
移動計算機がデータを転送中の状態であることを示す。通信品質が属性として与えられる。
- (4) 転送中断状態
データ転送が中断している状態を示す。この状態は転送状態からだけ遷移する。

3.4 Flexible Application Protocol

アプリケーションで最適な通信を実現するために、FAP (Flexible Application Protocol) を提案する。FAP はアプリケーションで想定されていない状況（移動計算機の分断等）に対処するためのプロトコルで、規定の入出力 API を持つ複数のモジュールから構成されており、アプリケーションはそれらを組み合わせ

て利用する。

3.5 コネクション数の動的制御機構

通信処理機構ではコネクション数を制御することにより過度の情報発信を防ぐ機構を持っている。コネクション数の動的制御機構では、情報の転送において、同時に使用されるコネクション（TCP コネクション等）の数を制御する。複数のコネクションを用いて同時に通信することにより、全体的な転送効率は各コネクションのプロトコルオーバーヘッド（コネクションの確立や ACK を待つ等）の削減等により向上する。しかし、コネクション数が過度に多くなると、個々のコネクションに割り当てられる帯域は減少するだけでなく、全体の通信効率も低下する。これは最適なコネクション数が存在することを示す。通信処理機構は最適なコネクション数になるように制御する。

さらに情報発信に優先度を設けることで、その優先度に従った情報発信を行う。時間制限を持つ情報に対して高い優先度をつけることにより、その情報を高速に送ることができ、時間制限を満足できる。さらに優先度を低く設定された情報は、通信量が少ない時間に発信され（遅延発信）、全体の通信効率が向上する。また、コネクション数の制御は、ネットワーク媒体やネットワークプロトコルに依存しないため、様々なネットワークを利用する移動計算機環境に適している。

4. 情報発信 Toolkit の実装

本章では情報発信 Toolkit を構成する、状態管理機構、状態処理機構、通信処理機構の実装について示す。

4.1 状態管理機構

状態管理機構は移動計算機の接続の状態を検出し、それに対応した管理ステートを保持する。まず移動計算機のネットワークとの接続の状態をハードウェアレベルでの検出を試みる。ハードウェアで接続の状態を検出できない場合、ソフトウェアで検出する。

移動計算機が通信可能になった（DHCP サーバから IP アドレスを割り当てられた等）ときに、移動計算機上の状態管理機構は SMH 上の状態管理機構に接続登録のメッセージを通知する。このとき移動計算機から提供するサービスの情報を同時に登録する。SMH 上の状態管理機構はそのメッセージを受け取り、移動計算機と通信可能であることを認識する。その後、SMH 上の状態管理機構は、移動計算機上の状態管理機構に接続確認のメッセージを定期的を送信する。移動計算機上の状態管理機構はそのメッセージに対して ACK を返答する。SMH は移動計算機との通信遅延として、このメッセージの RTT (Round Trip Time) 値を計

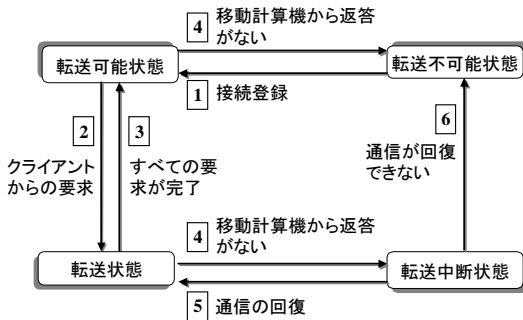


図3 管理ステートの遷移

Fig. 3 Transition of administration states.

測し、その平均値を保持する。また、コネクション数制御機構で利用するために転送状態においては、移動計算機との通信スループットを計測する。管理状態における、状態の遷移を図3に示す。以下において状態の遷移を図中の番号順に説明する。ここで、 α および β は、接続確認メッセージの消失を許容する回数で、 α はシステムが決定し、 β はアプリケーションが与える。初期状態は転送不可能状態である。

- 1 SMH 上の状態管理機構は、接続登録のメッセージを受信すると、管理ステートを転送可能状態にする。
- 2 SMH は、クライアントからの要求を移動計算機に伝える。SMH と移動計算機との間で通信を開始したとき転送状態へ遷移する。
- 3 クライアントからのすべての要求を処理したとき、転送可能状態へ遷移する。
- 4 接続確認メッセージに対して平均 RTT 時間を超えても返答がないとき、接続確認メッセージが消失したもとする。メッセージ消失が連続して α 回を超えたなら、現在の状態が転送状態なら転送中断状態へ遷移し、転送可能状態なら転送不可能状態へ遷移する。
- 5 メッセージの連続消失回数が β 回までに移動計算機との通信が回復したとき、転送状態へ復帰する。
- 6 メッセージの連続消失回数が β 回までに移動計算機との通信が回復しなかった場合、転送不可能状態へ遷移する。

4.2 状態処理機構

状態処理機構では、状態管理機構から移動計算機の管理ステータを取得し、そのステータに応じて MAPM に記述された処理を行う。MAPM は、Dynamic Loadable Module の形で本システムに組み込まれる。MAPM は表 1 に示すエントリポイントを用

表 1 状態処理機構におけるエントリポイントの一覧
Table 1 List of entry points.

connect_c2m	転送可能状態における処理が記述される。クライアントから移動計算機に対して通信するときに利用される。
connect_m2c	転送可能状態における処理が記述される。移動計算機からクライアントに対して通信するときに利用される。
suspend	転送中断状態における処理が記述される。
reconnect	転送中断状態から転送状態に遷移するための再要求の処理が記述される。
disconnect	転送不可能状態における処理が記述される。

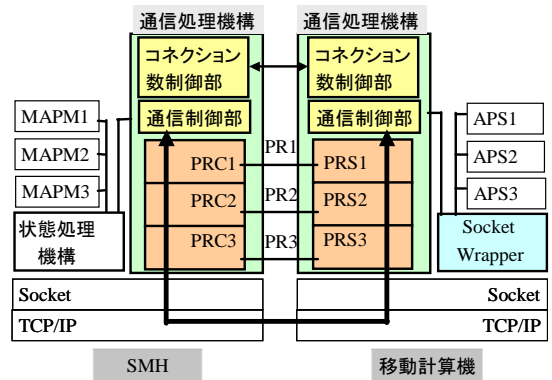


図4 通信処理機構の詳細

Fig. 4 Detail of communication processing part.

状態処理機構に提供する。状態処理機構はクライアントからの要求を受けると移動計算機の状態に応じた MAPM のエントリを呼び出す。

4.3 通信処理機構

通信処理機構では、FAP とコネクション数制御機構を提供する。通信処理機構の詳細を図4に示す。

FAP は、規定のインタフェースを持つモジュール (PR) で構成される。PR は、移動計算機側の PR (PRS) と SMH 側の PR (PRC) の組で構成され、複数個存在する。どの PR を用いるかはアプリケーションによってあらかじめ設定されている。移動計算機と SMH 間のプロトコルが既存のアプリケーションプロトコルと異なる場合、カプセル化し通信する。カプセル化するために移動計算機側に Socket Wrapper を導入する。Socket Wrapper はアプリケーションの通信を通信処理機構に渡すものでアプリケーションに対し通常のソケット API¹²⁾ と同一の API を提供する。通信処理機構は PR と通信処理の核となる通信制御部で構成される。通信制御部はアプリケーションによって設定された PRS にデータを渡し、実際に SMH と通信する。対応する PRC は、受け取ったデータを通信制御

部に渡す．そして通信制御部，状態処理機構，MAPM を経由してクライアントにデータが提供される．

状態管理機構は移動計算機と SHM との間のスループットを計測している．コネクション数制御機構は計測データを用いて，移動計算機と SMH 間のスループットの予測，そのスループットに応じたコネクション数の決定，アプリケーションへのコネクションの割当てを決定する．コネクションの割当ては，アプリケーション間とアプリケーション内の優先度に従い決定される．このとき，飢餓状態になるアプリケーションが発生しないように考慮する必要がある．

5. 評価

5.1 アプリケーションの構築

情報発信 Toolkit のプロトタイプを作成し，実際にアプリケーションを構築することにより，提案している Toolkit の評価を行う．移動計算機とクライアントは Windows 上で実装され，SMH は FreeBSD 上で稼働する．Windows におけるソケット API はダイナミックリンクライブラリ `wsock32.dll` で与えられる．移動計算機における Socket Wrapper 部は，`wsock32.dll` として実現し，元のものと同じ置き換えている．

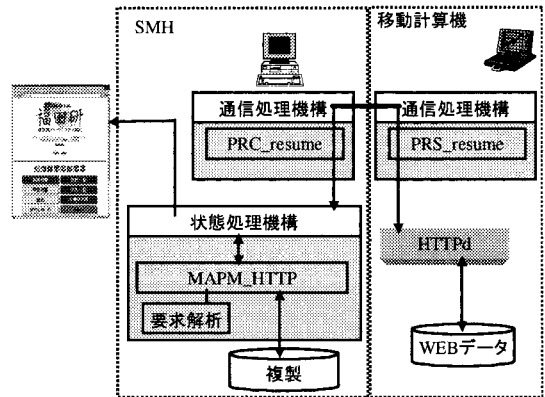
以下にアプリケーション例の詳細を示す．

(a) WWW

文献 1) で構築した WWW システム (旧システム) と同様の機能を持つ WWW システム (新システム) を，本 Toolkit を用いて実現し評価する．本アプリケーションにおける処理の流れを示す．移動計算機がネットワークと接続中は，移動計算機から WWW データが発信される．また同時に WWW データの複製が作成される．移動計算機が分断中においては，接続中に作成された複製が発信される．

新システムの構成を図 5 に示す．個人用途向けの WWW サーバを移動計算機上で利用し，クライアントには通常のブラウザを使用する．FAP として，`PR_resume` を利用する．`PR_resume` は，通信途中でデータの転送が中断しても，次回の要求時には中断地点からデータの通信を再開するものである．図 6 に本アプリケーションにおける MAPM のソースの一部を示す．`p_sd` は，移動計算機へのソケットを示し，`c_sd` はクライアントへのソケットを示す．

旧システムとコード量で比較する．旧システムは既存の WWW システムに機能を追加する形で実装されている．追加されたコード量は，固定ホスト側で 2930 行，モバイル側で 4655 行である．新システムでは，提案 Toolkit で用いて作成した MAPM のコード量は



<code>connect_c2m</code>	要求された URL を解析する．
<code>connect_m2c</code>	WEB データの複製を作成，中継する．
<code>disconnect</code>	複製を提供する．
<code>PRC_resume</code>	バッファを用意し，受信データをバッファに蓄積する．データの転送が完了すればバッファから消去する．バッファに存在するデータがクライアントから再要求されたとき，前回，中断した場所から通信する．
<code>PRS_resume</code>	送信データをあらかじめ spool し転送する．中断地点からの要求があれば，それに応える．

図 5 アプリケーション例 (1) : WWW
Fig. 5 Sample application (1): WWW.

346 行であった．プログラムの容易さを単純にプログラムの行数で比較はできないが，それでも全体のコード量で約 5% になっており，本 Toolkit を用いることでアプリケーションが容易に作成できることが確認できる．

(b) インターネット TV 電話

移動計算機上にインターネット TV 電話システムを構築する．移動計算機からクライアントへの一方方向の通信を，本アプリケーションでは提供している．これを双方向で利用することでインタラクティブなシステムを構築できる．

本アプリケーションの構成を図 7 に示す．本アプリケーションは UDP を基盤にしている．転送状態では，動画と音声进行中継する．SMH が移動計算機の通信状態が不安定であることを検出すると，転送中断状態に遷移し，通信が不安定であることを利用者に示す画像および音声を配信する．通信が分断すると SMH において留守番サービスが機能し用件を記録する．移動計算機が再接続したとき，記録した用件を移動計算機に転送する．本アプリケーションを実現するための MAPM のコード量は 138 行であった．WWW の約 80% になっており，本 Toolkit がアプリケーション作


```

/* MAPMのための include ファイル */
#include "mapm.h"

int connect_c2m(void)
{
    /* クライアントから要求を取得 */
    GetRequestByHTTP(c_sd,buffer,&readlen);
    /* 要求を解析 */
    GetURL(buffer);
    /* 要求を移動計算機に転送 */
    SendDataByHTTP(p_sd,buffer,readlen);
}

int connect_m2c(void)
{
    /* 移動計算機からデータを取得 */
    GetDataByHTTP(p_sd,buffer,&readlen);
    /* データの複製を作成 */
    WriteReplication(buffer,readlen);
    /* クライアントにデータを提供 */
    sendlen=SendDataByHTTP(c_sd,buffer,readlen);
}

int disconnect(void)
{
    /* クライアントから要求を取得 */
    GetRequestByHTTP(c_sd,buffer,&readlen);
    /* 要求を解析 */
    GetURL(buffer);
    /* 複製を取得 */
    ReadReplication(buffer,&readlen);
    /* クライアントに複製を提供 */
    SendDataByHTTP(c_sd,buffer,readlen);
}
    
```

図6 WWWアプリケーションにおけるMAPMの記述
Fig.6 Example of MAPM in WWW.

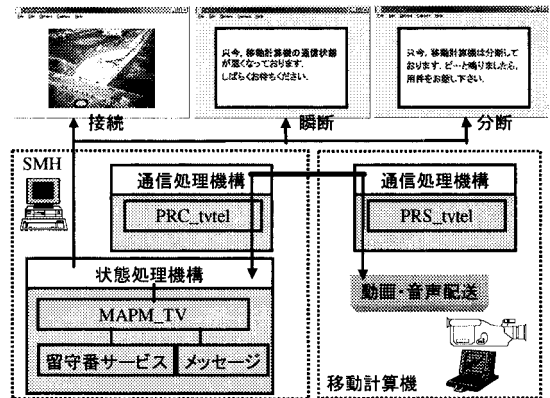
成を容易にしていることが分かる。

5.2 通信処理機構の評価

FAPの一例としてマルチメディアデータの転送するために一部の packets 落ちを復元できるパリティパケットモジュールのプロトタイプを実装し、実際の情報発信する状況を想定した環境で通信処理機構の評価を行う。

パリティパケットモジュールはマルチメディアデータにおける実時間通信や、衛星通信等の packets の再送による遅延が問題になる場合に有効なパリティパケット方式を用いた通信を提供する。パリティパケット方式は、 N 個の packets で構成されるグループに1つのパリティパケットを転送する。これにより通信に冗長性を持たせ packets の消失が発生しても、再送を行わずにパリティパケットから消失 packets の復元を行う。しかし、通信量が $N + 1/N$ 倍に増加する問題がある。

実験条件について示す。移動計算機からマルチメ



connect_c2m	クライアントから移動計算機へ中継する。
connect_m2c	移動計算機からクライアントへ中継する。
disconnect	あらかじめ用意されたメッセージと、留守番サービスを提供する。
suspend	あらかじめ用意されたメッセージを提供する。
PRC_tvte1	バッファが存在する。PRS_tvte1の送信したデータを受信しバッファリングする。
PRS_tvte1	通信帯域に応じて音声と動画の優先度を変化させ発信する。

図7 アプリケーション例(2): インターネット TV 電話
Fig.7 Sample application (2): internet TV phone system.

ディアデータ(ストリームデータ)と 50 KByte の固定長データを 40 個転送する。このとき、マルチメディアデータを優先して転送する。マルチメディアデータはパリティパケットモジュールを用いて転送する。グループ内の packets の個数を、なし、4、3、2 とする場合(NO, G4, G3, G2)について実験した。またマルチメディアデータとして 11,025 Byte/s 用いた。通信メディアとしては無線 LAN (1 Mbps) を用いた。

結果を図8に示す。横軸(コネクション数)における1から10は、コネクション数を固定にした結果を示し、NO, G4, G3, G2は、コネクション数数制御機構を用いた結果を示す。コネクション数を固定にした結果から、実験環境において最適なコネクション数とそのときの転送時間、packet 損失率の最適値が分かる。本実験における最適値は、転送時間が63秒、packet 損失率が0.7%であった。

次にコネクション数制御機構を用いた場合であるが、NOは、パリティパケットモジュールを用いずにコネクション数制御機構だけを動作させた結果である。最適値に近い結果を示しており、通信処理機構が有効に動作していることが分かる。最適値を若干上回る原因は、最適なコネクション数に遷移するのに時間がかかるためである。

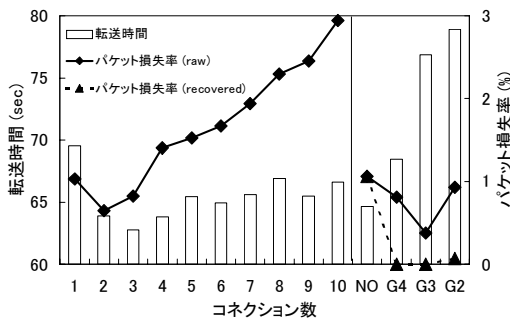


図 8 通信処理機構の評価

Fig. 8 Evaluation of communication processing part.

G4, G3, G2におけるスループットの結果では, パケットが損失した“パケット損失率 (raw)”と, パケットは損失したがパリティパケットで復元した“パケット損失率 (recovered)”の2種類を示す. G4においては, すべてのスループットにおいて, パケット損失率 (recovered) がなくなり良好な結果を得ることができた. このとき転送時間が増えているのは, 通信処理機構がマルチメディアデータを優先して転送したためであり, 通信処理機構が設計どおりに働いていることを示す.

6. おわりに

移動計算機から情報発信するシステムを提案した. 提案システムは, 移動計算機環境を考慮し, 移動計算機からの情報の発信を可能にする. さらに情報発信システムを実現するための情報発信 Toolkit の設計および実装を行った. 情報発信 Toolkit は, ネットワークの接続の状態の管理, その接続状態に応じた処理, 柔軟な通信方式等の枠組みをアプリケーションに提供する. 利用者は, 接続の状態に対処する機構である MAPM を追加するだけで, 移動計算機環境の問題を意識することなく, 既存のクライアント・サーバアプリケーションをそのまま利用できる. また, 提案 Toolkit の評価を行い, Toolkit 上で実現されるコネクション数制御機構の有効性を示した. 今後の課題を以下に示す.

- 提案した Toolkit は, アドホックネットワークに関する具体的な機能を持たない. 今後は, 本 Toolkit を用いてアドホックネットワークにおける移動計算機からの情報発信環境の実現を検討する.
- 移動計算機の通信環境, 使用する通信媒体等に応じて動的に FAP (Flexible Application Protocol) を構成できるシステムに拡張する.
- 様々なアプリケーションに提案 Toolkit を適用し, 特殊なアプリケーションについて検討する.

- 状態管理機構において, ネットワークの資源の状態を考慮するだけでなく, バッテリ等のその他の移動計算機の資源を考慮できる機構に拡張する. またそれに応じて FAP による通信方式や状態処理機構を適応させる.

参考文献

- 1) Tagashira, S., Nagatomo, K., Saisho, K. and Fukuda, A.: An Information Announcement System Based on WWW for Mobile Computers, *IEICE Trans. Fundamentals*, Vol.E81-A, No.7, pp.1387-1395 (1998).
- 2) Perkins, C.: IP Mobility Support, RFC2002 (1996).
- 3) Saisho, K.: Highly Reliable Multimedia Data Transmission with Redundancy, *Advanced Database Systems for Integration of Media and User Environments '98*, Kambayashi, Y., et al. (Eds.), *World Scientific*, pp.61-65 (1998).
- 4) Jing, J., Helal, A.S. and Elmagarmid, A.: Client-Server Computing in Mobile Environments, *ACM Computing Surveys*, Vol.31, No.2, pp.117-157 (1999).
- 5) Noble, D.B., Satyanarayanan, M., Narayanan, D., Tilton, E.J., Flinn, J. and Walker, R.K.: Agile Application-Aware Adaptation for Mobility, *Proc. 16th ACM symposium on Operating System Principles*, pp.276-287 (1997).
- 6) Qualcomm Inc.: Eudora Email. <http://www.eudora.com>.
- 7) Kistler, J.J. and Satyanarayanan, M.: Disconnected Operation in the Coda File System, *Operating System Review*, Vol.25, No.5, pp.213-225 (1991).
- 8) Kistler, J.J.: Disconnected Operation in a Distributed File System, *ACM Distinguished Theses* (1996).
- 9) Joseph, D.A., Tauber, A.J. and Kaashoek, F.M.: Mobile Computing with the Rover Toolkit, *IEEE Trans. Computers, Special issue on Mobile Computing*, Vol.46, No.3, pp.337-352 (1997).
- 10) Brewer, E., Katz, R., Chawathe, Y., Gribble, S., Hodes, T., Nguyen, G., Stemm, M., Henderson, T., Amir, E., Balakrishnan, H., Fox, A., Padmanabhan, V. and Seshan, S.: A Network Architecture for Heterogeneous Mobile Computing, *IEEE Personal Commun.*, Vol.5, No.5, pp.8-24 (1998).
- 11) 田頭, 稲田, 最所, 福田: 通信量が制限された環境における移動計算機情報発信方式, 情報処理学会コンピュータシステムシンポジウム論文集, pp.33-40 (1998).

- 12) Stevens, R.W.: *UNIX Network Programming*, Vol.1, Second Edition, Prentice Hall PTR (1998). 篠田陽一(訳): UNIX ネットワークプログラミング, 第2版, トッパン (1999).

(平成 11 年 12 月 13 日受付)

(平成 12 年 4 月 6 日採録)



田頭 茂明(学生会員)

1973 年生。1996 年龍谷大学理工学部卒業。1998 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了。1998 年同大学博士後期課程入学。現在に至る。モバイルコンピューティング, マルチメディアシステムの研究に従事。1999 年第 14 回電気通信普及財団賞テレコムシステム技術学生賞入賞。



安田 修

1972 年生。1996 年同志社大学工学部電気工学科卒業。ユニデン(株)を経て, 1999 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了。1999 年富士通(株)入社。現在に至る。移动通信システムのソフトウェア開発に従事。



最所 圭三(正会員)

1959 年生。1982 年九州大学工学部情報工学科卒業。1984 年同大学院工学研究科修士課程修了。同年同大学工学部助手。1991 年同大学工学部講師。1993 年同大学大型計算機センター助教授。1994 年奈良先端科学技術大学院大学情報科学研究科助教授, 2000 年香川大学工学部信頼性情報システム工学科教授, 現在に至る。工学博士。高信頼性システム, 並列/分散処理, モバイルシステム, 並行処理等の研究に従事。1998 年情報処理学会全国大会大会優秀賞受賞。電子情報通信学会会員。



福田 晃(正会員)

1954 年生。1977 年九州大学工学部情報工学科卒業。1979 年同大学院工学研究科修士課程修了。同年 NTT 研究所入所。1983 年九州大学大学院総合理工学研究科助手。1989 年同大学助教授。1994 年より奈良先端科学技術大学院大学情報科学研究科教授, 工学博士。オペレーティング・システム, 並列化コンパイラ, 計算機アーキテクチャ, 並列/分散処理, 性能評価等の研究に従事。本学会平成 2 年度研究賞, 平成 5 年度 Best Author 賞受賞。著書「並列オペレーティングシステム」(コロナ社), 訳書「オペレーティングシステムの概念」(共訳, 培風館)。ACM, IEEE Computer Society, 電子情報通信学会, 日本 OR 学会各会員。