社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2004－ＭＢＬ－29　(20)
2004／5／14

通信路管理を実現するセッション層アソシエーション

金子 晋丈 †　　森川 博之 ‡　　青山 友紀 †

† 東京大学大学院情報理工学系研究科　〒 113-8656 東京都文京区本郷 7-3-1
‡ 東京大学大学院新領域創成科学研究科　〒 277-8582 千葉県柏市柏の葉 5-1-5
E-mail: †‡{kaneko, mori, aoyama}@mlab.t.u-tokyo.ac.jp

あらまし　近年，1人が複数のコンピュータを同時並行して利用するようになってきた．このような環境では，ユーザのその時々の要求に応じて通信の終端点となる通信デバイスを選択することが期待される．しかしながら，現在のネットワークアーキテクチャでは，1人が複数の通信を組み合わせて通信サービスを構成することは困難である．筆者らは，ユーザの意図する通信サービスを実際の通信に対応付けるためのシステムフレームワークを構築した．本システムフレームワークはユーザが利用している通信を一括して管理制御できるだけでなく，アプリケーションからネットワークの下位層の情報を隠蔽し通信を抽象化することで，柔軟性の高い通信サービスを実現する機構である．さらに，本システムフレームワークに基づいた具体的な手法としてセッション層アソシエーションを設計した．
キーワード　セッション層 抽象化 階層 ネットワークアーキテクチャ one-to-many

Session Layer Association for Communication Channel Management

Kunitake KANEKO†　Hiroyuki MORIKAWA‡　and　Tomonori AOYAMA†

†Graduate School of Information Science and Technology, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656 Japan

‡Graduate School of Frontier Sciences, The University of Tokyo
5-1-5 Kashiwanoha, Kashiwa-shi, Chiba, 277-8582 Japan

E-mail: †‡{kaneko, mori, aoyama}@mlab.t.u-tokyo.ac.jp

Abstract: Recently, we use the several computers concurently. In this one-to-many computing environment, it is desired that we can choose the communication device and make a group of devices for more comfortable communication service according to our context. However, the current network architecture makes it difficult. We, therefore, discuss a new network architecture for realizing the above flexible communication service. With this architecture, we can control the all communication channel belonging to the user. And, it provides an interface which is independent to the lower layer details to application process. Finally, we show our system called session layer association based on the architecture.
Keywords: session-layer abstraction layering network-architecture one-to-many

1. はじめに

近年，コンピュータの利用形態が，1人のユーザがひとつのコンピュータを使うモデル（one-to-one モデル）から1人のユーザが複数のコンピュータを同時に並行して利用するモデル（one-to-many モデル）へと大きく変化しつつある．コンピュータシステムにおけるこのパラダイムシフトは，複数のユーザがタイムシェアリングで大型計算機を利用するモデル（many-to-one モデル）から one-to-one モデルへの変化と全く性質の異なるものである．many-to-one モデルから one-to-one モデルへの変化において，コンピュータシステムは，コンピュータの低廉化によりひとつのコンピュータを使うユーザ数が N から 1 に減っただけであり，変化していない．一方で，one-to-many モデルは1人のユーザが同時に並行して複数のコンピュータを統合して扱うことを意味しており，その時々に応じてユーザの目的とする処理に対しどのコンピュータを対応づけるかという全く新しいシステムフレームワークが必要となる．

このようなパラダイムシフトを背景にして生まれてきた，SETI@home[1] や distributed.net[2] などに代表されるグリッドコンピューティングは，コンピュータの計算資源に着目した one-to-many モデルにおけ

るコンピュータシステムの一形態である．例えば，SETI@home では，膨大な電波望遠鏡からのデータをワークユニットに分割し，それらをプロジェクトに参加するコンピュータに動的に割り当て計算を行うことで，地球外生命体の探索を行っている．すなわち，グリッドコンピューティングは，プロジェクトが目的とする処理を実際のコンピュータにおける計算処理に対応付け，それを管理制御することで最終目標を実現していると言えよう．

一方で，コンピュータを通信の終端デバイスとして捉えると，通信に関しても同様に one-to-many モデルにおけるシステムフレームワークの構築が望まれる．one-to-many モデルにおいて実現される通信サービスとは，ユーザがその場その時に応じて利用可能な通信デバイスを選択して行う通信であり，また複数の通信デバイスを組み合わせて行う通信である．一例として，ユーザ A とユーザ B のテレビ電話について述べる．このテレビ電話は，A の近くのプラズマディスプレイ 10 面にそれぞれ異なる B の様子を表示し，壁に埋め込まれた 100 のスピーカに，B の周囲の 100 のマイクから拾われた音声をそれぞれに出力しながら，テレビ電話の様子をサーバで録画するというものである．このとき，ユーザ A と B の間で実現されている通信サービスはテレビ電話であるが，実際は複数の通信がこのサービスを実現するために利用されており，それらが異なる通信デバイス上で同時並行して動作している．すなわち，one-to-many モデルにおける通信サービスのシステムフレームワークとは，ユーザが意図する通信を，どのコンピュータ（通信デバイス）で終端するのかを適切に管理制御することに他ならない．

このような one-to-many モデルにおける通信の管理制御に関し，大きく 2 つのアプローチが考えられる．ひとつは，それぞれの通信デバイスで動作するアプリケーションプログラムを通して通信を制御する方法であり，もう一つは，一つ一つのアプリケーションプログラムとは独立に，すべてのアプリケーションに共通するネットワークシステムの基盤として通信を制御する方法である．筆者らはアプリケーション開発の容易性，アプリケーション間の互換性，様々なアプリケーションの基盤となりうる将来性を重視し，後者のアプローチを採る．通信の管理制御をネットワーク基盤として構築することにより，アプリケーションプログラムは現在のアプリケーションプログラムと大きく異なり，IP アドレスやポート番号などネットワークの通信識別情報を利用しない通信インタフェースを作成することになる．

本稿では，one-to-many モデルにおいて，ユーザ 1 人 1 人が複数のコンピュータを連携させながら通信を行う際のシステムフレームワークについて考察する．特に，ユーザが意図する通信サービスをネットワーク技術としてどのように具現化するか，またユーザの意図する通信サービスと実際の通信をどのように対応づけるかについて述べる．これは，one-to-many モデルにおけるシステムフレームワークを構築する上で必須な通信の抽象化に関する議論である．逆に，通信の抽象化の観点から one-to-many モデルにおける通信を捉えると，ユーザの意図する通信サービスを論理通信と，実際の通信を実通信とみなすことができる．

筆者らが検討するシステムフレームワークは管理機構，接続機構，そして制御機構から構成される．管理機構は，論理通信と実通信の対応付けを決定し，制御機構を通して接続機構に通知するとともに，対応付けを管理する．接続機構は，管理機構からの指示に従い，指示されたアプリケーションプログラム間で通信ができるように通信路を設定する．最後に制御機構は，管理機構から接続機構への指示を伝達するとともに，管理機構間のメッセージ交換を行う．

このようなシステムフレームワークを構築する事で，ユーザの意図する様々な形態の通信をサポートすることができる．これにより，通信デバイスに束縛されない通信サービスや，ネットワークトポロジを超えた通信サービス，さらに送受信されるデータの暗号化処理を統一したフレームワークの中で実現することができる．特に接続機構では，アプリケーションプログラムと実際の通信のインタフェースを接続する際に，一つの論理通信に対して常に同一の実通信を接続するだけではなく，様々な実通信を切り替えながら接続したり，常に複数の実通信に接続することで，ユーザの意図する論理通信をユニキャストだけではなく，モビリティサポートを含んだユニキャスト，マルチキャストやエニーキャストとして実現することができる．また，暗号化処理を行うことで，暗号通信をネットワーク技術の一部として実現することも可能である．

以下では，one-to-many モデルにおけるシステムフレームワークの全体像を管理機構，接続機構，制御機構に分けて述べる．さらに，具体的な実現手法として，セッション層アソシエーションを用いたシステムフレームワークについて述べる．

2. 通信の抽象化

2.1 システムフレームワークの概要

本稿では，図 1 に示すように，通信の抽象化に基づくシステムフレームワークを論理通信と実通信の対応付けを管理する機構，管理機構の指示に従い，実際にアプリケーションデータのやりとりを実現する実通信を設定しアプリケーションプログラムと接続する接続機構，管理機構と接続機構間，管理機構間を繋ぐ制御機構に分けて説明する．まず，それぞれの機構における検討事項を述べる．

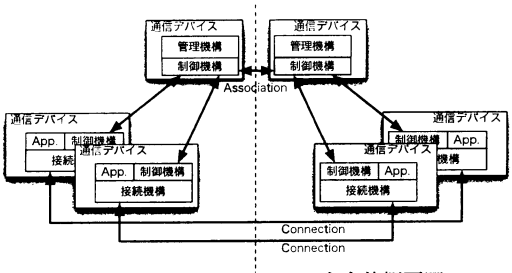

図 1: システムフレームワーク全体概要図

（1）管理機構

実通信は，アプリケーションプログラム間でアプリケーションデータをやりとりするための通信である．したがって，管理機構が実通信を把握するためには，実通信を行う際に必要な通信識別情報を管理する必要がある．すなわち，管理機構は，どのアプリケーションプログラムがネットワーク上のどこで動作しているのか，そのプログラムが保持する通信インタフェースは何を目的として使われるのかを知らなければならない．

一方，論理通信に関しては，まず，one-to-many モデルにおける通信の利用の観点から，論理通信と実通信の対応付けにおける要求事項を明確にする必要がある．その上で，論理通信と実通信の対応付けはどのように実現されなければならないか，また，論理通信をどのように識別するのかについて検討を行う必要がある．

（2）接続機構

管理機構で実現される論理通信と実通信の対応付けは，実際にアプリケーションプログラムが扱うアプリケーションデータの送受信に反映される必要がある．管理機構が論理通信と実通信の対応付けを解除した場合には，その実通信は終了されなければならない．すなわち，アプリケーションデータの送受信を制御するために，接続機構は，管理機構からの指示に従いアプリケーションプログラムと実通信路を接続，解除する機能が必要となる．

上記の結果，接続機構が実通信路を設定し実通信のインタフェースを保持するため，アプリケーションプログラムが保持する通信インタフェースがネットワークに依存する通信識別情報を持たない通信インタフェースとなり，アプリケーションプログラムは通信の状況を容易に把握することができない．したがって，接続機構は，アプリケーションプログラムが通信状態を把握できるように，実通信の状況をアプリケーションプログラムに伝達する必要がある．

（3）制御機構

one-to-many モデルでは，どの通信デバイス上のアプリケーションプログラムも利用可能である．また，通信を管理するデバイスに関しても同様である．その結果，管理機構の動作する通信デバイスとアプリケーションプログラムの動作する通信デバイスが物理的に異なることが想定される．そのため，管理機構の指示を接続機構に通知する必要がある．さらに，管理機構の状態が常に接続機構に反映されているように，これら二つの機構の間で通信状態に関する情報の一貫性が保たれていなければならない．

また，論理通信についても同様に，通信相手と通信状態に関する情報の一貫性が保たれていなければならない．

以上をまとめると，それぞれの管理機構における検討課題は，以下のようになる．

（1）管理機構
- ・実通信環境の把握
- ・論理通信の定義
- ・論理通信と実通信の対応付け

（2）接続機構
- ・アプリケーションプログラムと実通信路の接続
- ・アプリケーションプログラムが感知できない実通信に関する情報の扱い

（3）制御機構
- ・通信デバイス間の通信状態情報の一貫性維持

次節以降では，機構ごとに，それぞれの検討事項について述べる．

2.2 管理機構

実通信環境の把握

現在検討が進められているサービス発見技術 [3, 4, 5] に様々な特徴があることから明らかなように，実通信環境は非常に多様性に富んでおり，単一の手法ですべての実通信環境を把握できるとは考えられない．そこで，実通信環境の把握に関しては本稿では詳細を述べず，なんらかの手法に基づき利用するアプリケーションプログラムがどの通信デバイス上で動作しているか知り得るものとする．なお，ユーザはユーザ自身が利用するアプリケーションプログラムに関しては把握できるが，通信相手がどのアプリケーションプログラムを選択するかについては知り得ないものとする．

論理通信の具現化

まず，論理通信と実通信の対応付けに関する要求事項を明確にする．one-to-many モデルにおいて最も重要な点は，ユーザがその場その時に応じてどのような実通信を利用するかを決定することにある．論理通信と実通信の対応付けは，実通信の終端点と実通信の数において，ユーザの要求により時間軸上で動的に変化するものであり，その変化は即座に反映される必要がある．すなわち，論理通信と実通信の対応付けは柔軟に変更できなければならない．

ここで，"通信"を再考してみると，どのような通信であっても通信の終端点は 2 点以上あり，すべての終端点が状態情報を同期していなければ通信は実現されない．すなわち，論理通信に対応付ける実通信を

動的に変更する場合，その変更情報は必ず通信相手に通知されなければならない．そこで，筆者らは，論理通信に通信としての実体をもたせ，その通信によって通信相手とネゴシエーションを行い通信状態を同期することで，実通信との対応付けを動的に行う機構を実現することにした．このとき，すべての通信は，論理通信を終端する通信デバイスから制御されることに注意されたい．

したがって，本管理機構における論理通信と実通信は以下のように対応付けられる．

- 論理通信を開始し論理通信によって実通信の識別情報を交換
- 論理通信で得られた情報に基づき実通信を開始
- 実通信を終了しても論理通信は継続
- 論理通信の終了により実通信も終了
- 実通信の変更情報は論理通信によって交換

以下では，論理通信の通信としての実体をアソシエーションと呼び，実通信をコネクションと呼ぶ．また，一つのアソシエーションで管理されたコネクションの集合をセッションと呼ぶ．

次に，アソシエーションをどのように実現するかについて述べる．アソシエーションは通信によって実現されるため，通信を終端する通信デバイスが必要となる．ここで，コネクションを終端する通信デバイスとアソシエーションを終端する通信デバイスは本質的に独立なものであることから，アソシエーションは，コネクションとは独立に構築される．具体的なアソシエーションの終端デバイスとしては，携帯電話のような常に身につけている端末やホームサーバなどを検討している．

最後に，アソシエーションを用いて論理通信をどのように識別するかについて述べる．アソシエーションの意義は，論理通信と実通信の対応付けに関する情報を通信相手と共有することにある．したがって，アソシエーションは，論理通信の開始から終了まで通信路を維持する必要はなく，情報の更新が生じたときのみ通信相手に通信路を構築し通知すればよい．そのため，アソシエーションによる更新情報には論理通信を識別する情報が必要になるとともに，論理通信と実通信の対応付けにより通信が開始したり終了したりするため，アソシエーションに用いる論理通信の識別情報は高い安全性を持たなければならない．そこで，筆者らは，アソシエーションにおける論理通信の識別情報として，暗号学的な識別情報を用いることとした．暗号学的な識別情報により，暗号学に基づいた安全性を論理通信と実通信の対応付けに用いることが可能になる．なお，暗号学的な識別情報を用いるため，この識別情報がグローバルで一意であることの保証はない．

2.3 接続機構

論理通信と実通信は管理機構において対応付けがなされるが，通信サービスの実体は，通信デバイス上でアプリケーションデータの送受信によって実現される．接続機構は，それぞれの通信デバイス上で動作し，アプリケーションプロセスからアプリケーションデータを受け取り，管理機構で決定された通信相手にデータを送信し，逆に通信相手から受け取ったデータをアプリケーションに渡す機能を保持している．

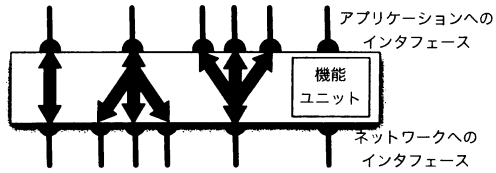

図 2: 接続機構機能ユニット

一方で，接続機構はアプリケーションプロセスがデータを送受信しようとしているかどうかに関わらず，管理機構の切断要求を受け取ったときには，コネクションを閉じ，データをネットワークに流さないような機構が必要になる．すなわち，接続機構は，アプリケーションプロセスと通信相手の接続機構に繋がる少なくとも二つのインタフェースを持っており，管理機構の要求に応じてこれを接続，切断するｓ機構である．

最も基本的な接続機構は，アプリケーションプロセスへのインタフェースとネットワークへのインタフェースをそれぞれ一つずつ保有するが，接続機構を以下のように拡張することで，様々な実通信を実現することが可能になる（図 2）．例えばマルチプレクス処理では，ネットワークから流れてきたデータを複製して，複数のアプリケーションプロセスへ渡すことが可能である．逆に，スプリット処理では，アプリケーションプロセスから渡されたデータを複製して複数のネットワークインタフェースに送出することも可能となる．このような接続機構における処理を接続機構における機能ユニットだと考えると．上記の仕組みは，複製・分割機能ユニットであるといえる．

接続機構における機能ユニットとして他にも様々な機能を設けることが可能である．例えば，アプリケーションプロセスへのインタフェースに対して，動的にネットワークへのインタフェースを切り替えて接続するという機能はモビリティサポート [6] やマルチホーミングサポートになる．暗号・復号処理を行うことで，暗号通信機能を保有することが可能となる．

接続機構はアプリケーション空間からネットワーク空間を完全に分離する機構であるため，アプリケーションはネットワーク情報について把握することが困難である．上記のように，アプリケーションはデータを送信しているものの，接続機構によってデータの送受信が止められている場合，アプリケーションプロ

セスは期待している応答を通信相手から得られなくなる．また，アプリケーションプロセスは直接のネットワークインタフェースを持たないために，ネットワークの輻輳状況や遅延などについて完全に隠蔽されている．そのため，接続機構は，データを送受信するインタフェースだけではなく，接続状況，およびネットワーク状況についてアプリケーションプロセスに通知するメッセージ用のインタフェースを設ける．このインタフェースを用いることで，アプリケーションプロセスはより柔軟な通信サービスを構築する事ができる．

2.4 制御機構

one-to-many モデルにおいては，アソシエーションを構築する通信デバイス（アソシエーション終端デバイス）とコネクションを構築するデバイス（コネクション終端デバイス）が異なる通信デバイスであることが考えられるため，管理機構は接続機構と通信によって状態の一貫性を常に保つ必要がある．制御機構は，管理機構の要求を実際に接続機構に通知することで，実通信の開始や終了を実現するために不可欠な機構である．

例えば，コネクションを確立したものの，その後ユーザの都合により，そのコネクションを終了しようとしても，アソシエーション終端デバイスとコネクション終端デバイスが通信できない場合，通信状態に関する情報の一貫性が失われる．このとき，アソシエーション終端デバイスからコネクション終了の指示を出しても，コネクション終端デバイスでは通信が継続したままになってしまう．同様の問題は，アソシエーション終端デバイス間においても発生しうる．

このような問題を解決するには，アソシエーション終端デバイスとコネクション終端デバイス，もしくは，アソシエーション終端デバイス間で常に通信できる状態を維持することが考えられるが，実現は困難でありコストもかかる．そのため，通信ができなくなったときのために，これらの通信デバイス間で一定間隔のビーコンを送信し，ビーコン応答がない場合制御不能と見なし，自動的にコネクションを切断し通信サービスの安全性を保つ．

3. セッション層アソシエーション

3.1 概要

セッション層アソシエーションは，2. で述べたシステムフレームワークをセッション層において構築したシステムである．本システムでは，現在，広範に利用されているインターネットをネットワーク基盤に用いることとし，識別情報に IP アドレスとポート番号，トランスポート層のプロトコルを用いている．

セッション層アソシエーションは，セッションマネージャとソケットマネージャによって構成されている．セッションマネージャは，2. で述べた管理機構と制御機構を保持しており．アソシエーション終端デバイスで動作する．ソケットマネージャは，接続機構と制御機構を保持しており．コネクション終端デバイスで動作する．なお，携帯電話端末をアソシエーション終端デバイスにする場合など，セッションマネージャとソケットマネージャが両方搭載している通信デバイスも存在する．

3.2 セッション層アソシエーション

セッション層アソシエーションは以下のように動作する．まず，アプリケーションプログラムはセッション層アソシエーションに対応した機能を利用するため，通信インターフェースとしてセッション層インタフェースである slsocket を利用する．セッション層インタフェースは通信路情報に依存しないインタフェースであり，ソケットマネージャによって管理されている．ソケットマネージャは，通信デバイス内のすべての slsocket を管理しており，セッションマネージャからの指示に応じて，接続先 IP アドレス，ポート番号，トランスポート層プロトコル，接続種別から，socket を作成し対応する slsocket と接続する．なお，接続種別は，socket を待ち受けモードで作成するか接続モードで作成するかを示す．一方，セッションマネージャには，同一セッションとしてまとめられたすべてのコネクションに関して，slsocket と socket の対応が登録されている．

3.3 slsocket 利用手順

本節では，セッション層アソシエーションを用いる際のプロトコル手順を示す．セッション層アソシエーションでは，2. で述べたように論理通信を確立することで通信を開始する．以下では，ユーザ α とユーザ β の通信を例に手順を述べる．それぞれのセッションマネージャを A，B とし，コネクション終端デバイスを a，b とする．なお．A と B の間ではすでに公開鍵の交換が行われており，互いを確実に認証できるものとする．また，a，b ではすでにアプリケーションプログラムが起動しており slsocket をソケットマネージャに登録しているものとする．まず，A は，B に接続するために，接続要求を出しアソシエーションを確立する．アソシエーション確立後，A は利用したい a を指定して B にコネクション開始要求を送信する．B は，a と通信できる近隣のアプリケーションプログラムを検索し，b を選択し，b にコネクション待ち受け準備要求を送信する．b においてコネクションの待ち受け準備ができると b は B に b の IP アドレスとポート番号を ACK とともに返し，B は A に通知する．A は a にコネクション開始要求をだし，a は b に接続する．

4. 関連研究

1970年代後半から異種コンピュータのネットワーク接続のために，ネットワークの通信モデルの検討が開始された．とくにISOで1977年に始まった検討は，1979年にOpen Systems Interconnection（以下，OSI）参照モデルとして通信の7階層モデル[7]が標準化される．OSIでは，セッション層を分散的に行われている処理を論理的なつながりでまとめるもの[8]としている．本稿が扱うone-to-manyモデルはOSIで明確に想定されていないが，本システムフレームワークのユーザの通信活動における様々なコネクションをひとつのアソシエーションで束ねるという思想は，広義のセッション層の機能として捉えることができるであろう．

以下では，通信の抽象化の観点から，これまでの研究と本研究との関連性について述べる．ユーザを通信の主体として捉え，抽象化を行ってきた研究としてパーソナルモビリティ[9]が挙げられる．パーソナルモビリティでは，メールアドレスなどユーザを示す名前空間を定義し，その名前空間を名前解決することで，利用通信デバイスを決定する仕組みである．単なる名前空間の解決は，通信先ノードの抽象化であり，通信の抽象化は実現できない．通信先ノードを抽象化する研究としてはネットワーク層におけるターミナルモビリティ[10, 11]が挙げられる．

一方，SIP[12]は，INVITEリクエストによって論理通信を開始し，ダイアログによって通信を識別し，BYEリクエストで論理通信を開放している．すなわち，SIPは，論理通信を構築するフレームワークである．しかし，SIPでは，本来全く別の機構である通信相手の位置解決と通信の確立を一体的に扱っている．そのため，論理通信を確立するのに必ずSIP URLが必要となり，柔軟性の低い抽象化となってしまっている．また，SIP自体は，実通信の構築に関する機構を持たない．

実通信の構築は，SIPと併せて用いられるSDP[13]によって実現されている．しかし，SDPはマルチメディア通信のみを扱うプロトコルであり，実通信を変更するたびにすべての通信を記したSDPを再度作り直し，re-INVITEリクエストにより新しいSDPを送受信しなければならない．さらに，SDPは，マルチキャストの配信情報を示すプロトコルであるため，情報受信のためのアクセス先しか記載されていない．そのため，SIPの論理通信の終端デバイス以外を受信用通信デバイスにすることはSIPとSDPの機構だけではできない．

また，トランスポート層で実現するターミナルモビリティサポート[14]は，TCPコネクションにCookieを与えることで通信の抽象化を行っているが，単一コネクションしかサポートしないため，one-to-manyモデルでは動作しない．

5. おわりに

本稿では，ユーザが同時に複数のコンピュータを同時並行して利用するone-to-manyモデルにおける通信サービスを実現するためのシステムフレームワークを示し，あわせて，実現手法であるセッション層アソシエーションについて述べた．one-to-manyモデルでは，様々な通信をユーザの意図にあわせてどの通信デバイスで通信を終端させるのかを適切に管理・制御することが重要であり，そのための機構としてアソシエーションを導入した．アソシエーションにより通信相手と同期しながら，通信の接続・終了を制御することが可能になる．すなわち，通信の抽象化により，ユーザが享受する通信サービスと実際の通信の対応付けが可能になったといえる．

参考文献

[1] SETI@home, http://setiathome.ssl.berkeley.edu/.

[2] distributed.net, http://www.distributed.net/.

[3] The Jini Community, http://www.jini.org/.

[4] J. Veizades, E. Guttman, C. Perkins, and S. Kaplan, "Service Location Protocol (SLP)," RFC 2165, IETF, Jun. 1997.

[5] Universal Plug and Play Forum, http://www.upnp.org/.

[6] 金子，森川，青山，中山，"多様化するインターネット環境におけるエンドツーエンド型モビリティサポート，" 信学技報，MoMuC2002-8, May 2002.

[7] H. Zimmerman, "OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection," IEEE Transactions on Communication, COM-28 4, Apr. 1980.

[8] C. Bachman and M. Canepa, "The Session Control Layer of an Open System Interconnection," In: Proc. of COMPCON 78. IEEE: New York (Fall, 1978).

[9] P. Maniatis, M. Roussopoulos, E. Swierk, K. Lai, G. Appenzeller, X. Zhao, and M. Baker, "The Mobile People Architecture," ACM Mobile Computing and Communications Review, Jul. 1999.

[10] C. Perkins, "Mobile IP," IEEE Communications Magazine, pp. 84-99, May 1997.

[11] M.Kunishi, M.Ishiyama, K.Uehara, H.Esaki, and F.Teraoka, "LIN6: A New Approach to Mobility Support in IPv6", In Proc. of International Symposium on Wireless Personal Multimedia Communication (WPMC) 2000, Nov. 2000.

[12] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: session initiation protocol," RFC 2543, IETF, Mar. 1999.

[13] V. Jacobson and M. Handley, "SDP: Session Description Protocol", RFC 2327, IETF, Apr. 1998.

[14] A. Snoeren and H. Balakrishnan, "An end-to-end approach to host mobility. In Proc. of 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking, pp.155-166, Aug. 2000.