

UPnP サービス連携を適用した複数経路統合による動的帯域拡大システム

藤野 信次[†] 塩内 正利[†] 福田 茂紀[†]
石原 進^{††} 水野 忠則^{††}

モバイルとユビキタスコンピューティングの混在環境ではローカルとリモートネットワーク上に分散する機器やサービスの発見・連携技術が重要である。IP ネットワーク上の機器やサービスの発見・制御の業界標準に UPnP がある。しかし、UPnP は一般にサブネットをまたぐサービス発見ができず、複数サービスを制御して動的に連携するには限界がある。そこで本論文ではシナリオに従って複数サービスを制御するエージェントをローカルとリモートの両ネットワーク上に配置することにより、両ネットワーク上に分散する複数サービスを連携制御するアーキテクチャを提案する。本アーキテクチャの有効性を検証するため、他の端末の通信経路を動的に自己の通信経路と統合してリモート通信帯域を拡大する複数経路統合システムを試作したので、その実装と評価を報告する。複数経路統合システムはローカルの中継ノードの発見とリモートの分配サーバとの動的連携が必要になるので、本アーキテクチャの応用として適している。リモートと通信中にローカルの中継端末を追加する実験を行い、端末数に応じた動的なスループット拡大を観測し、発見した端末情報を分配サーバに通知して動的に通信経路を制御するというローカル-リモート間の連携動作を確認した。さらにサービス発見と連携の所要時間を測定し、従来の端末追加から帯域拡大までの所要時間と比較し実用的な値であることを確認した。

Dynamic Bandwidth Enhancing System Using Multiple Paths Aggregation Applying UPnP Service Collaboration Architecture

NOBUTSUGU FUJINO,[†] MASATOSHI SHIOUCHI,[†] SHIGEKI FUKUTA,[†]
SUSUMU ISHIHARA^{††} and TADANORI MIZUNO^{††}

In the intermingled environment of mobile and ubiquitous computing, devices and services collaboration framework is important, because they are distributed over the local and remote networks. Universal Plug and Play (UPnP) framework is suitable for discovering and utilizing distributed devices and services on the IP network. However, the UPnP service discovery can not work beyond subnets. Also, multiple services collaboration is out of scope in the UPnP. In this paper, we propose an architecture where two agents, which control multiple services according to a written scenario, are allocated both on the local and remote networks. To evaluate this architecture, we prototyped the multiple paths aggregation system, that needs service-discovery and dynamic collaboration with multiple services over the local and remote networks. We confirmed that the bandwidth of the terminal increased dynamically when the relay terminal is added while communicating with remote server, and that the architecture worked. We measured needed time for service discovery and collaboration, and found it relatively shorter than conventional needed time from terminal adding to bandwidth enhancing.

1. はじめに

3G 携帯電話等の広域無線通信網の普及により、い

つでもどこでもオフィスや学校のサーバに接続して一定のサービスを受けられるモバイルコンピューティングが可能になってきた。一方、無線 LAN 等のローカル無線網の普及により、その場にある機器やサービスを必要に応じて利用するユビキタスコンピューティングの可能性が広がってきた。今後は両者が混在するモバイル-ユビキタス統合環境でのサービス連携技術が重要になると考えられる。このような環境ではリモ

[†] 株式会社富士通研究所
Fujitsu Laboratories Ltd.

^{††} 静岡大学創造科学技術大学院
Graduate School of Science and Technology, Shizuoka University

トとローカルのネットワーク上に分散した種々のサービスを動的に組み合わせ、ユーザの要求する仕事を、その場その場で実現する仕組みが必要になる。

IP ネットワーク上にあるサービスやデバイスとその場で発見し、煩雑な設定なしに使用するための仕組みとして UPnP (Universal Plug and Play)¹⁾がある。UPnP はネットワークデバイスだけでなくデジタル家電や AV 機器へも適用され²⁾、業界標準となりつつある。しかし、UPnP は基本的にデバイス/サービスと、それを使用する制御機器間の 1 対 1 のプロトコルであり、複数のサービス間の連携までは規定していない。たとえば、あるサービスの処理結果を順次他のサービスに受け渡して処理させる場合、UPnP では 1 つのサービスとの一般的な制御手順は規定しているが、各サービス間でどのようにデータを受け渡すかまでは、特定の AV 機器等のサービスのほかは規定していない。また、UPnP はサービス発見の手段としてマルチキャストを使用しているため、リモートとローカルネットワーク間のようなサブネットをまたぐサービス発見は一般には行えない。

そこで我々は、シナリオに従って複数サービスを制御するエージェントをローカルとリモートの両ネットワーク上に配置することにより、端末の周囲のサービス群とリモートにあるサービス群を柔軟に連携する汎用的なアーキテクチャを提案する。

さらに、本論文ではこのアーキテクチャの応用として複数経路統合を取り上げ、周辺にある端末の通信帯域を動的に統合して自己の通信帯域を拡大するシステムを試作したので、その実装と評価について報告する。提案する UPnP サービス連携アーキテクチャを適用することで、中継端末の発見、およびリモートとローカルネットワーク上の動的なサービス連携が柔軟かつ容易に実現できることを示す。

2. モバイル-ユビキタス統合環境における UPnP の課題

ここではまず、前提とするモバイル-ユビキタス統合環境について述べ、次にサービスの発見・連携手段として有用な UPnP の概要と、それをモバイル-ユビキタス統合環境に適用する場合の課題について述べる。

2.1 モバイル-ユビキタス統合環境

モバイルコンピューティングとユビキタスコンピューティングが混在する環境をモバイル-ユビキタス統合環境と呼ぶことにする。

ここでモバイルコンピューティングとはユーザの場所にかかわらずメールや Web ブラウズのような一定

のサービスを受けることと考える。このとき、ユーザの移動中は離れた場所にあるリモートネットワーク上のサービスにアクセスすることになる。

一方、ユビキタスコンピューティングとは、無線 LAN スポットで提供されるプリンタサービスのように、ユーザの周辺にある機器やサービスをその場で利用することであると考えられる。この場合のサービスはユーザの近くのローカルネットワーク上にあると考えられる。

したがって、モバイル-ユビキタス統合環境ではリモートネットワーク上のサービスとローカルネットワーク上のサービスの連携が重要になる。たとえば、離れたオフィスにある資料を出張先で閲覧し、その場のプリンタで印刷するような場合は、リモートネットワーク上にある文書提供サービスと、ローカルネットワーク上にある文書閲覧サービスと印刷サービスを有機的に連携する必要がある。

図 1 にモバイル-ユビキタス統合環境で想定するネットワーク構成を示す。ユーザは必要最低限の端末を持って移動するものとする。この端末を通じて周りにある機器やサービスと、リモートにあるサービスを利用する。

周囲の機器とは無線 LAN のような狭域無線網により接続されるものとする。たとえば、出先のオフィスにあるプリンタやディスプレイと無線 LAN アドホックモードで接続したり、街中の無線 LAN スポットで提供されるプリンタに無線 LAN インフラモードで接続したりして、それらを利用する。ここで、周囲の機器と接続されるネットワークをローカルネットワークと呼ぶ。すなわち、本論文中で、ローカルネットワー

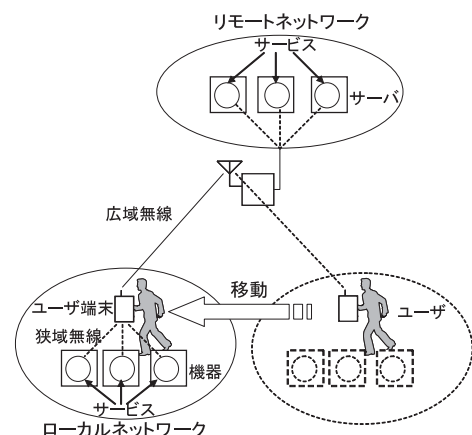


図 1 モバイル-ユビキタス統合環境のネットワーク構成
Fig. 1 Network configuration of mobile-ubiquitous combined environment.

クとは端末が現在、物理的に直接接続されているサブネットワークをいう。これには端末どうして構成するアドホックネットワークも含む。端末とローカルネットワークは狭域無線で接続される。

リモートにあるサービスを提供するサーバとはセルラ網のような広域無線網を通じて接続されるものとする。たとえば、移動先で、自分のオフィスにセルラのデータ通信モードで接続し、サーバ上の文書を閲覧する。このようなサーバが接続されているネットワークをリモートネットワークと呼ぶ。すなわち、本論文で、リモートネットワークとは、端末が現在アクセスしている、離れた場所にあるサーバが接続されるサブネットワークをいう。また、リモートネットワークはユーザが通常、移動していないときにいるオフィスのような、本拠としているホームネットワークと考える。

ローカルネットワークとリモートネットワークは端末の持つ広域無線網を通じて接続される。

2.2 UPnP の概要

UPnP は IP ネットワークを前提とし、ネットワーク上のサービスを発見し、煩雑な設定なしにすぐ使えるようにするためのフレームワークであり、サービスの連携手段として有効と考えられる。

図 2 に UPnP デバイス・アーキテクチャ³⁾を示す。サービスを提供する UPnP デバイスとそれを使用するコントロールポイント (CP) から構成される。UPnP デバイスは 1 つ以上のサービスを含む。UPnP は SSDP (Simple Service Discovery Protocol)⁴⁾ と呼ばれるサービス発見プロトコルを使用している。

UPnP プロトコルは (0) Addressing, (1) Discovery, (2) Description, (3) Control, (4) Eventing, (5) Presentation の各ステップからなる。ここでは Eventing と Presentation の説明は本論文の議論の本質には関係しないので省略する。

(0) Addressing : UPnP デバイスは IP アドレスを取得する。動的にアドレスを取得する場合は DHCP や

AutoIP を使用するが、その方法までは規定していない。

(1) Discovery : UPnP デバイスはその存在をマルチキャストパケットによりネットワーク上で広告する。それを受信した CP はそのデバイスを発見する。また、ネットワークに接続された CP はマルチキャストによりデバイスを検索する。検索メッセージと合致した UPnP デバイスは CP に応答し、発見される。以降の通信はユニキャストにより行われる。

(2) Description : デバイスを発見した CP はデバイス記述を UPnP デバイ스에 要求し、それを受け取る。デバイス記述にはデバイス ID, デバイスの構成, サービスの URL 等が含まれる。CP は必要に応じ UPnP サービスの URL に対してサービス記述を要求する。要求されたサービスはサービス記述を返答する。サービス記述にはサービス ID, サービス名, 制御のための URL 等が含まれる。デバイスおよびサービス記述は XML により記述される。

(3) Control : CP はサービス記述に従い、定義されたアクションの要求と、実行結果の受け取りをする。

2.3 モバイル-ユビキタス統合環境における課題

以上で述べた UPnP のサービス発見や制御メカニズムはユビキタスコンピューティング環境では有効と考えられる。しかし、リモートとローカルネットワーク上にサービスが分散するモバイル-ユビキタス統合環境に適用するには以下のような課題がある：

(1) リモート-ローカルネットワーク間のサービス発見

UPnP ではサービス発見のためにマルチキャストを利用しているため、一般にサブネットワークをまたぐサービス発見ができない。したがってリモートとローカルネットワーク間でのサービス発見のための対策が必要になる。

(2) リモート-ローカル間のロバストなサービス連携

モバイル-ユビキタス環境では、リモート-ローカルネットワーク間の接続が一時的に途切れるような状況でも、各ネットワーク上のサービス間の連携が、できるだけ安定して継続的に実行される必要がある。すなわち、各サービスの呼び出しが中断せず、サービス間のデータの受け渡しが正しく行われ、また一連のサービスの処理時間の変動をできるだけ少なくする必要がある。

たとえば、ローカルの端末上の CP からリモートの複数サービスを順番に制御する場合、UPnP では CP がデバイスの IP アドレスとサービスの URL を知っていれば、リモートのサービスの制御は可能である。しかし、2.1 節で述べたように、リモートネットワー

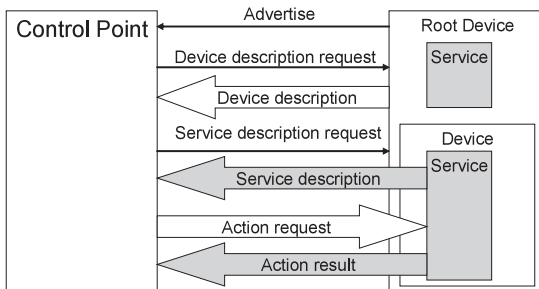


図 2 UPnP デバイス・アーキテクチャ

Fig.2 UPnP Device architecture.

クと端末は無線網で接続されるので、つねに安定した接続が得られるとは限らない。誤り訂正や再送制御等、通信レイヤの機能により、データは正しく受け渡し可能だが、サービス制御の中断、処理時間の大幅な遅延を生じる可能性がある。したがって、できるだけ、これらの問題が生じないような対策が必要になる。

(3) サービス連携実行の汎用的な仕組み

UPnP は基本的にはサービスと CP 間の 1 対 1 の制御プロトコルであり、複数のサービス間の連携までは規定していない。CP で複数の UPnP サービスを連携制御するようにコードを開発することは可能だが、実現したいサービス連携ごとにコードを開発しては効率が悪い。そこで開発が容易でコードの再利用が可能な汎用的な仕組みが必要になる。

(4) サービスの抽象化

一般的に UPnP で定義されるサービスは比較的粒度の小さい単機能のサービスが多い。リモートとローカルネットワーク上のそれらのサービス群が各々独立に連携しては、両ネットワーク間をまたがるサービス間の通信パス数が多くなり、通信オーバーヘッドが大きくなる。たとえば、一方のネットワーク上に m 個のサービスがあり、他方のネットワーク上に n 個のサービスがあるとすると、両ネットワークをまたぐ通信パスは最大 ($m \times n$) となり、通信オーバーヘッドが大きくなる。リモートとローカルネットワーク間はずねに安定して接続されているとは限らないので、できるだけ通信を減らす必要がある。

2.1 節で述べたように、リモートネットワーク上のサービス群はモバイルコンピューティングで使用され、ローカルネットワーク上のサービスはユビキタスコンピューティングで使用される。したがって、各ネットワーク上で、それぞれモバイルまたはユビキタスコンピューティングのサービスどうしをまとめることが容易と考えられる。これらサービス群をより粒度の大きい、抽象度の高いサービスにまとめる仕組みがあれば、両ネットワーク間のサービス連携が簡素になり、通信オーバーヘッドの削減が期待できる。

3. エージェントによる UPnP サービス連携

3.1 UPnP エージェントによるリモート-ローカル連携アーキテクチャ

前章で述べた課題を解決するため、シナリオに従って複数のサービスを制御するエージェント⁵⁾を用いた UPnP サービス連携アーキテクチャを提案する。図 3 にそのアーキテクチャを示す。図で、端末はローカルネットワークとは狭域無線網で、リモートネットワー

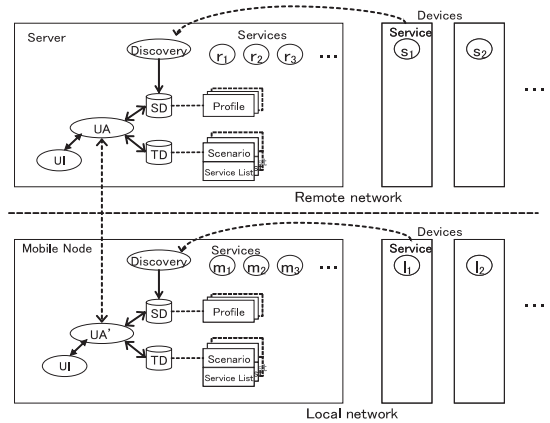


図 3 UPnP エージェントによるサービス連携アーキテクチャ
Fig. 3 Service collaboration architecture using UPnP Agents.

クとは広域無線網で接続されるものとする。

ローカルネットワークに接続される端末とリモートネットワーク上のサーバに UPnP Agent (UA) と呼ぶエージェントを配置する。UA は CP として他のサービスを制御すると同時に、自身が UPnP サービスとして他の CP または UA から制御される。CP としては複数の UPnP サービスを連携制御するサービスコーディネータとして動作する。

端末側の UA は端末上のサービスとローカルネットワーク上で発見したサービスの管理と制御を行う。リモート側の UA も同様である。

端末上の UA はローカルネットワーク上の他の端末の UA やリモートネットワーク上の UA と通信する。ローカルネットワーク上の UA どうしは UPnP の発見機構により互いに発見するが、端末上の UA はリモートの UA の IP アドレスをあらかじめ知っており、互いに通信できる。2.1 節で述べたように、リモートネットワークはユーザにとってホームネットワークだからである。UA は UA 間通信により、自身が管理するサービス記述を他の UA に通知できる。通知を受けた UA はそれらのサービスも制御できるようになる。

UA はサービス発見機構、ユーザインタフェース (UI)、サービス・ディレクトリ (SD)、タスク・ディレクトリ (TD) を持つ。SD には UA が管理するサービスのリストとサービス記述がプロファイルとして保持される。通常は SSDP により発見された UPnP サービスのサービス記述が SD に保持される。UA と同一デバイス上のサービスは必ずしも UPnP サービスである必要はない。非 UPnP サービスの場合には、あらかじめ UA に UPnP と同形式のサービス記述を登録する。UA はそれらのサービスの代理としてサービ

ス広告を行い、他の CP には UPnP サービスとして認識される。

TD にはタスク⁶⁾の実行に必要なサービスリストと、それらの実行順序や実行条件を記述したシナリオが保持される。ここでタスクとはユーザが実行する、複数の UPnP サービスから構成されるメタサービスのことをいう。シナリオに従って複数のサービスを適宜制御することによりタスクが実行される。

シナリオには UPnP サービスが提供するアクションの実行と、実行結果による分岐、UPnP イベントの待ち合わせ、等を記述する。これにより、複数サービスの多様な利用と連携ができる。詳細は 5.3 節で述べる。

また、UA は自身の実行コードといえるシナリオを他の UA に送って実行すること、すなわち、一種の移動コードをサポートする。これにより、リモートの UA や他の端末のサービス制御が容易にできる。

3.2 本アーキテクチャによる課題解決

以上で述べたアーキテクチャにより、先に述べた課題を解決する。以下に 2.3 節に示した課題とその解決手段を示す。

(1) リモート-ローカルネットワーク間のサービス発見

端末とサーバ上の UA がそれぞれローカルとリモートネットワーク上のサービス発見を行い、互いにそれを通知することにより、両ネットワークをまたぐサービス発見が可能となる。

(2) リモート-ローカル間のロバストなサービス連携

リモートネットワークとの通信が安定している間に、端末の UA がシナリオをリモートの UA に送ることにより、リモートネットワークとの通信が一時的に切断されたとしても、リモートの UA は送付されたシナリオに従い、サービス制御を中断することなく実行できる。また同様に、シナリオにより両ネットワーク上のサービス間連携に必要なパラメータも送ることができる。したがって両ネットワーク間のサービスの、よりロバストな連携が可能となる。

(3) サービス連携実行の汎用的な仕組み

シナリオによるサービス連携実行により、様々なタスクやサービスに柔軟に対応可能である。またシナリオの記述を変更することで他のタスクへの対応も容易になる。また、シナリオを再利用することで類似したタスクの実現が効率的に行える。

(4) サービスの抽象化

リモートとローカルネットワーク上の UA が、それぞれのネットワーク上の複数サービスをまとめることにより、より抽象度の高いサービスにすることが可能

である。たとえば、出張中に会議設定して、その結果を印刷する場合を考える。リモートの UA 上で、複数の予定表サービス、閲覧サービス等を連携制御することにより“会議設定”という抽象化されたサービスが構築できる。ローカルの UA 上でレンダリングサービス、プリンタへのファイル送信サービス等を連携制御することで“印刷”サービスが構築できる。“会議設定”サービスの結果を“印刷”サービスに送る方が元のサービスを個々に連携制御するよりも全体のサービス連携が簡素になり、両ネットワークを行き来する通信量を削減できる。

たとえば、 n 人の会議設定をする場合に、リモートネットワーク上の“会議設定”サービスを利用せず、端末側から予定表サービスにアクセスしたとすると、各人の予定の要求と応答のために少なくとも $2n$ 回の通信が発生する。これに対して“会議設定”サービスを利用すればシナリオの送信と会議設定結果の受信の 2 回の通信で済む。仮に予定表の要求に 100 バイト、応答に 3k バイト、シナリオに 10k バイト、会議設定結果の応答に 1k バイト要するとすると、10 人の会議設定時、サービスを抽象化しない場合の通信量 31k バイトに対し、抽象化した場合の通信量は 11k バイトと約 3 分の 1 に削減できる。抽象化した場合の通信量は一定なので、人数が増えるほど削減率は増す。なお、後述する試作で使用したシナリオは 5k バイト以下である。

3.3 先行研究との比較

サブネットをまたぐ UPnP サービスの連携については、サービスのあるサブネットと CP のあるサブネットの間にゲートウェイ (GW) を設置して橋渡しする方法が提案されている^{7)~9)}。我々の提案もユーザの端末が一種の GW になっていると考えてもよい。

文献 7) では、インターネットと家庭内ネットワークの間に設置したホーム GW 上に、インターネット上のサービスプロバイダから制御される仮想的な UPnP MediaServer¹⁰⁾ を実装することで、家庭内の UPnP 機器からのサービス発見とコンテンツへのアクセスを可能としている。文献 8) でも同様の AV 機器サービスの Proxy を実装する手法が述べられている。これらは、提供するサービスが単なるメッセージ転送でない点が我々の提案する UA と同じだが、提供サービスが特定の AV 機器サービスに固定されている点が異なる。我々の提案するエージェントはシナリオで動作を定義できるので、提供するサービスは限定されず汎用的である。

文献 9) では、発見したデバイスの情報と制御メッ

ページの転送機能を持つ UPnP Bridge を、インターネットと家庭内ネットワークの間に設置する Internet Gateway Device (IGD)¹¹⁾ 上に実装している。デバイス情報を取得したインターネット上の CP は IGD 経由でデバイスの制御が可能となる。この UPnP Bridge は、メッセージを転送するだけで自身がサービスを制御しない点が我々の提案する UA と異なる。また、家庭内のデバイスや外部の CP の数が増大すると、家庭内外を通る通信量の増大が懸念される。

類似の手法として、UPnP ではないが、JINI¹²⁾ の Look Up Service (LUS) がある。JINI では各サブネット上に配置された LUS どうしがサービスリストを通知することにより、サブネットを越えたサービス発見を可能としている。この場合もサービスやそれを使うものの数が多くなるとサブネット間の通信量が増大する。

以上、いずれの先行研究もサブネット内の複数のサービスを連携制御してまとめることをしない点と、安定した有線環境を前提とし、無線環境で起こりうるオフラインでのサービス連携を考慮していない点が我々の提案方式と異なる。我々の提案ではリモートとローカルの両サブネットに配置したエージェントが各々のサブネット内の複数サービスを連携制御し、より抽象度の高いサービスにしてから互いに通信できるので、両サブネット間の通信量を削減可能である。これはシナリオの記述により実現されるので、より柔軟なサービス連携が可能である。また、我々の提案では、通信安定時にサービス実行シナリオを送ることにより、オフライン時のサービス連携処理の継続を可能にしている。

4. UPnP サービス連携による複数経路統合

3章で述べたアーキテクチャの応用として複数経路統合システムへの適用を行う。複数経路統合はローカルネットワーク上の中継サービスの発見と、その中継サービスとリモートネットワーク上の分配サービスとの連携により動作するので本アーキテクチャの応用として適している。以下に複数経路統合システムの概要と提案アーキテクチャによる実現について述べる。

4.1 複数経路統合システムの概要

複数経路統合は端末が自身と他の端末の通信経路を同時に使用することにより、自己の通信帯域や通信の接続性を向上させるものである^{13),14)}。図4に複数経路統合システムの基本構成を示す。複数経路統合システムは複数のモバイル端末 (MN: Mobile Node) とリモートネットワーク上の相手サーバ (CN: Correspondent Node)、分配サーバから構成される。

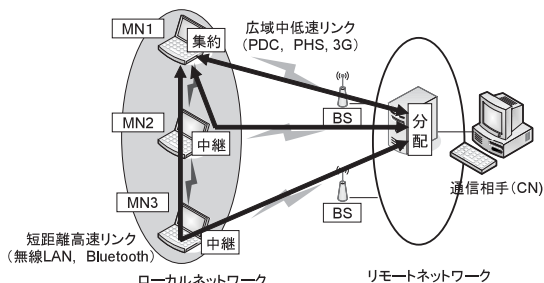


図4 複数経路統合システムの基本構成

Fig. 4 Basic configuration of multiple paths aggregation system.

端末は各々リモートネットワークに接続するための通信手段と、端末どうしのローカルな接続を行うための通信手段を持ち、同時に2つのネットワーク接続ができるものとする。ここでローカル接続の通信帯域はリモート接続の通信帯域に対して十分高速とする。このとき、ある端末と相手サーバとの通信経路は、自己のリモート接続ネットワークを使用してサーバに接続する経路と、ローカル接続ネットワークを経由して他の端末のリモート接続ネットワークからサーバに到達する2種類の経路が存在する。これらの経路を同時に使用することで、端末は自己のリモート接続の帯域を仮想的に増大できる。また、複数の経路を同時に利用することでネットワークへの接続性を高め、通信の信頼性を向上させることができる。

このように1つの端末に経路を統合するためにリモートネットワーク側にはトラフィックを分配するサーバが必要になる。CNからMNへのトラフィックはすべて分配サーバを経由する。また集約する端末にはトラフィックを集約するための機能が、他の端末には中継機能が必要になる。ここで分配/集約という言葉はサーバから端末へのダウンロード方向を基準にする。トラフィックの分配/集約方法には様々なものが提案されている。Web Proxyを利用する方法¹⁵⁾は、アプリケーションがWebに限定されるが、実装が比較的簡単である。またMobile IPを利用する方法¹⁶⁾は、IPレイヤで実現しているためアプリケーションに依存せず、既存のサーバに影響を与えずに実現できるので実用的である。

複数経路統合システムにおいて通信リソースを共有する端末どうしの関係をアライアンスと呼ぶ¹⁷⁾。そして通信リソースを統合して使用する端末をアライアンス・リーダー (AL: Alliance Leader)、通信リソースを提供する端末をアライアンス・メンバ (AM: Alliance Member) と呼ぶ。

ユースケースとしては、たとえば、同僚と出張中に

無線 LAN 付きデュアル無線携帯電話のセルラ回線に会社に接続し、資料の閲覧を開始したが帯域不足で時間がかかるため、途中で同僚の同種携帯電話を追加してダウンロード時間を短縮する等が考えられる。

4.2 UPnP サービス連携アーキテクチャの適用

複数経路統合システムでは AM となる中継端末を発見し、その情報をリモートネットワーク上の分配サーバに通知する仕組みが必要になる。中継端末の発見には UPnP の SSDP が利用できる。また、分配サービスへの通知には 3 章で提案した UPnP サービス連携アーキテクチャの UA 間通信が利用できる。複数経路統合システムはリモートネットワーク上の分配機能と、ローカルネットワーク上の集約、中継機能を実現する各サービスの連携により成立しているといえる。したがって、提案アーキテクチャの応用として適している。

図 5 に UPnP サービス連携を応用した複数経路統合システムの構成を示す。システムはローカルネットワーク上にある端末 MN1 と MN2、リモートネットワーク上にあるコンテンツサーバ CN、分配サーバからなる。MN1 上に集約サービス、MN2 上に中継サービス、分配サーバ上に分配サービスを配置する。また分配サーバ、MN1、MN2 上にそれぞれ UPnP エージェント UA1、UA1'、UA2' を配置する。各端末は UPnP デバイスとして、またサービスは UPnP サービスとして定義する。中継サービスの発見は UPnP を使用し、システムへの組み込みは UA1 および UA1' のシナリオにより実現する。UA2' は中継サービスの制御と UA1' との認証とアライアンスへの参加許可を行う。

本アーキテクチャを適用することにより、中継端末

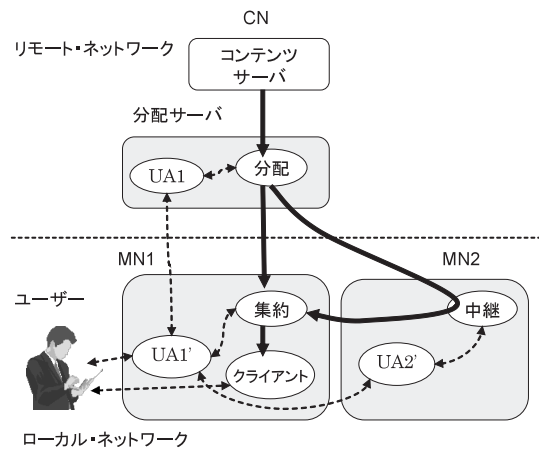


図 5 UPnP サービス連携による複数経路統合

Fig. 5 Multiple paths aggregation applying UPnP service collaboration.

の発見は UPnP のサービス発見機構をそのまま使用でき、システムの構成が簡単になる。そして発見した中継サービスの分配サーバへの通知には UA 間通信が利用できる。また、中継端末のシステムへの組み込みは UA のシナリオにより実行するので、端末や中継サービスの種類変更に対応しやすいという利点がある。

5. 試作システム

UPnP サービス連携アーキテクチャによる経路統合システムの実用性を検証するために、通信中に他端末を中継ノードとしてシステムに組み込み、動的に帯域を拡大するシステムを試作した。以下に試作システムの実装と動作概要について述べる。

5.1 実装

図 6 に試作したシステムの構成（実験系）を示す。ローカルネットワーク側は携帯端末（MN1~3）、無線 LAN アクセスポイント（AP）からなる。端末には PHS と無線 LAN の同時使用が可能な IP 携帯電話端末を使用した。ローカルネットワークはインフラモードの無線 LAN により構成した。リモートネットワークとは PHS 網により接続される。リモートネットワーク側はコンテンツサーバ（CN）、分配サーバ、リモートアクセスサーバ（RAS）から構成される。経路分配には VPN¹⁸⁾ による方式を採用し、分配サーバには PPTP¹⁹⁾ を使用した。

VPN による経路分配方式は Mobile IP による方式

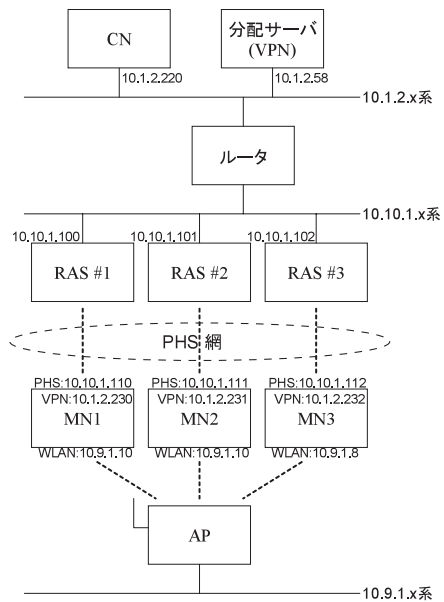


図 6 試作システム構成（実験系）

Fig. 6 The prototype system configuration (experimental system).

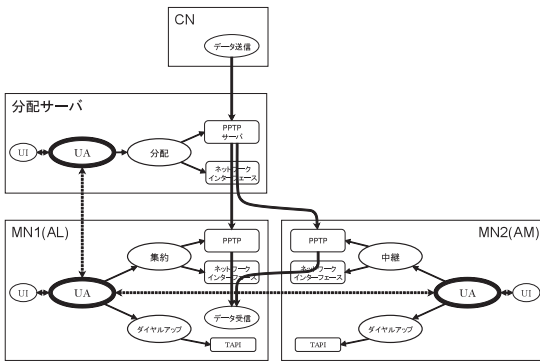


図 7 試作システムのソフトウェア構成

Fig. 7 The software configuration of the prototype system.

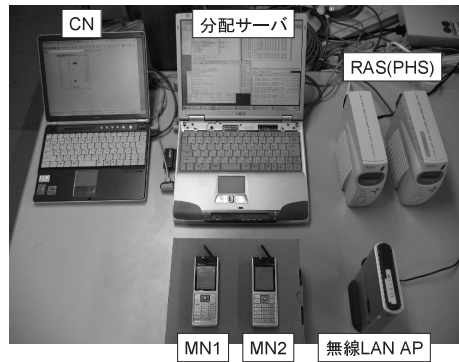


図 8 試作システムの外観

Fig. 8 Appearance of the prototype system.

と同様に、トラフィックの分配を IP レイヤで実現するため、アプリケーションに依存せず、既存のサーバに影響を与えずに実現できるという特長を持つ。加えて、もともと企業等の外部ネットワークからのアクセス時に使用される VPN サーバを利用するため、セキュリティ上の問題が少なく、適用しやすい。VPN による方式の動作概要については 5.4 節で述べる。

図 7 に試作システムのソフトウェア構成を示す。図で MN3 (AM) は MN2 と同一であるので省略している。分配サーバは分配サービス, PPTP サーバ, UPnP エージェント (UA) で構成される。分配サービスはあらかじめ UA に登録される。分配サービスは PPTP サーバと、ネットワークインタフェースのルーティングを制御する。分配サーバの OS は Linux で、PPTP サーバは Poptop Ver. 1.3.0²⁰⁾ を改造したものをを使用した。

端末は UA, 集約または中継サービス, ダイアルアップサービス, PPTP クライアント, データ受信ソフトから構成される。集約または中継サービスと、ダイアルアップサービスはあらかじめ UA に登録される。集約サービスおよび中継サービスは PPTP クライアントと、ネットワークインタフェースのルーティングを制御する。端末の OS は Windows CE.NET4.2 で、集約、中継の各サービスは同 OS 上で動作する独自開発のルータ機能を用いて実装した。PPTP クライアントは OS 標準装備のものを使用した。図 8 に試作したシステムの外観を示す。

5.2 動作シーケンス

図 9 にシステムの動作シーケンスを示す。図は端末 MN1 がサーバと 1 対 1 で通信中に、新たに他の中継端末 MN2 が接続されてから、システムに追加されて経路統合が完了するまでの動作シーケンスである。図で (0) は MN1 が VPN 経由で CN と 1:1 で通信中

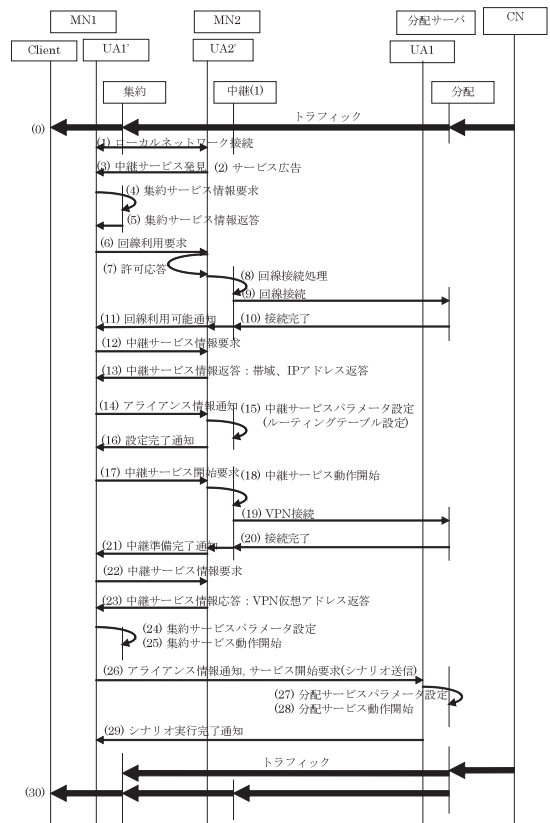


図 9 システムの動作シーケンス

Fig. 9 The system sequence diagram.

の状態, (30) が MN1 と MN2 の経路統合が行われている状態である。

以下、UPnP の各ステップに対応させて説明する。(1) で MN2 がローカルネットワークすなわち、無線 LAN に接続され、IP アドレスが割り当てられる (Addressing)。 (2) で MN2 の UA (UA2') が自身のサービスを広告する (Advertise)。 (3) で MN1 の UA

(UA1') がそれを発見する (Discovery). ここで図では明らかに示していないが UA1' と UA2' の間でサービス記述の要求と応答が行われる (Description). 以降, Control のフェーズに移行する. ここから UA はシナリオに従い CP として UPnP サービスのアクションの実行を行う. たとえば (4) は UA1' から自身の端末 (MN1) 上の集約サービスに対する Action request で, (5) はそれに対する Action result である.

以下は特に UA に特徴的な部分を説明する. (7) では UA1' からの回線利用 (アライアンス参加) 要求 (6) に対する許可を, UA2' が端末のユーザインタフェースを使用してユーザに判断させている. (26) では UA 間通信によりローカルネットワーク上の UA1' からリモートネットワーク上の UA1 にシナリオを送信している. このとき, 追加された MN2 へのパケット分配に必要な IP アドレス等のパラメータもシナリオ中で送っている. UA1 がそれによって (27), (28) の処理を実行し, 分配サーバに MN2 へのパケット分配を開始させている.

5.3 シナリオ

以上で述べたシーケンスは UA のシナリオで記述される. 図 10 に今回用いたシナリオの一部を示す.

シナリオは XML 形式で記述される. 最上位の要素はタスクで, タスクはサービス定義, アクション定義, プロセス定義, イベント定義から構成される. サービ

```

<?xml version="1.0" encoding="Shift_JIS" ?>
<task name="start">
  <service-mappings>
    <service name="service1" type="urn:schemas:fujitsu-com:service:AllianceLeader:1" id="AL_1-0" />
    <service name="service2" type="urn:schemas:fujitsu-com:service:AllianceMember:1" id="AM_1-0" />
  </service-mappings>
  <action-mappings>
    <action name="action1" service="service1" cmd="GetAddresses" />
  </action-mappings>
  <action name="action1" service="service1" cmd="GetAddresses" />
  <params />
  <results>
    <var name="SAM_REMOTES" relative="Remote" />
    <var name="SAM_LOCALS" relative="Local" />
    <var name="SAM_GLOBALS" relative="Global" />
    <eval forward-to="action2" />
  </results>
</action>
</task>
  
```

タスク定義

サービス定義

アクション1

アクション6

アクション10

図 10 UA のシナリオの一例 (抜粋)
Fig. 10 Example of the UA script (extract).

ス定義は, サービスタイプとサービス ID によって特定される UPnP サービスと, タスク中で使用される名前とをマッピングする.

アクション定義とプロセス定義はいずれも実行する動作を定義するが, 前者は UPnP サービスに行わせる動作, 后者は自端末のコマンド等で行う動作を定義するという違いがある. いずれの定義も, それぞれ 1 つ以上のアクションかプロセスで構成される.

アクション定義では, サービス名と UPnP アクション名を指定する. 下位要素としてパラメータ定義と結果判定記述を含む. 前者は指定したサービスへ渡すパラメータを記述する. 結果判定記述はアクションの実行後に返り値を評価し, それによって選択される次の動作としてアクションかプロセスを指定する. アクション実行後のパラメータの変数への格納も記述できる. また “forward-to-remote” 属性としてシナリオを指定し, リモートの UA に送る指定もできる.

一方, プロセス定義では起動するローカル環境のコマンドとその引数を記述する. “wait-for-terminated” 属性を用いてコマンドの終了を待つこともできる. アクション定義と同様に, 下位要素として結果判定記述を加えることができる.

イベント定義はタスク中で待ち合わせる UPnP イベントの指定と, イベント発生時の動作を記述する. シナリオは独自開発したシナリオエンジンにより処理される. シナリオの転送と転送先での実行はこのシナリオエンジンにより実現している.

5.4 トラフィック分配・集約

ここで, 本試作で用いた VPN によるトラフィック分配と集約の動作概要について述べる.

図 11 に VPN によるトラフィック分配の動作概要を示す. AL (MN1) に対しては送信元: CN, 宛先: AL の仮想 VPN アドレスのパケットを, 送信元: VPN

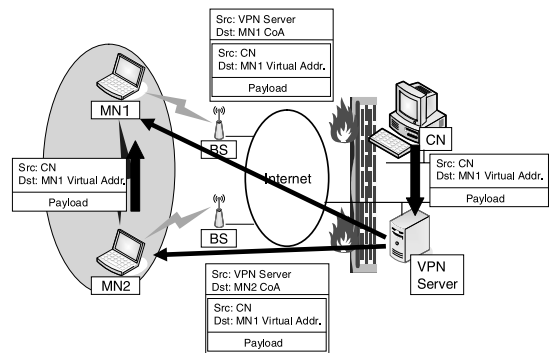


図 11 VPN によるトラフィック分配
Fig. 11 Traffic distribution using VPN.

サーバ、宛先：AL の CoA (Care of Address) のパケットでカプセル化して送る．AM (MN2) に対しては送信元：CN，宛先：AL の仮想 VPN アドレスのパケットを，送信元：VPN サーバ，宛先：AM の CoA のパケットでカプセル化して送る．パケットを受信した AM (MN2) はデカプセル化したパケットの宛先が AL (MN1) であるため，パケットを AL に転送する．以上により VPN サーバで分配したパケットが AL で集約される．

なお，試作では各端末へのトラフィックの配分は最も単純なラウンドロビンを使用し，均等配分とした．

6. 評価

本試作を用いて基本的な動作確認実験と実環境での評価を行った．表 1 に実験・評価環境を示す．

6.1 静的評価

複数経路統合の動作確認のため，1～3 経路時のスループットを測定した．測定は 1k バイトの UDP パケットを連続で各 320 回送信し，MN1 でのデータ受信記録から平均スループットを求めた．

図 12 に測定結果を示す．1 経路時のスループット 26.4 kbps に対し，2 経路時 51.7 kbps，3 経路時 76.5 kbps とスループットが向上している．これより，VPN による経路統合が機能していることが分かる．

表 1 実験・評価環境

Table 1 The experiment and evaluation environment.

	分配サーバ	端末
使用機器	FMV-BIBLO NEB/1000H	WipCom 1000
OS	Linux (Fedore core 2)	Windows CE.NET 4.2
VPN	Poptop Ver 1.30 改	Windows CE 標準搭載
リモート通信	—	PHS 32kbps 回線交換
ローカル通信	—	無線 LAN IEEE802.11b

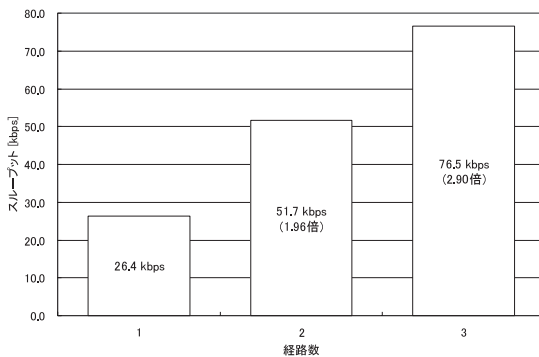


図 12 経路数に対するスループットの測定結果

Fig. 12 The measurement result of throughput versus numbers of paths.

6.2 動的評価

UPnP による中継端末の発見と動的追加時の動作を確認するため，1 経路で通信中に中継端末を追加する実験を行った．CN より 1k バイトの UDP パケットを MN1 に連続送信し，途中で中継端末 MN2 を立ち上げ，MN1 でのデータ受信量を時系列に測定した．

図 13 に測定結果を示す．図は経過時間に対するデータ転送量と，その微係数より求めたスループットを示す．図より 104 秒付近で転送量が増加し，スループットが高くなっている．これより，ローカルネットワーク上の中継端末の発見と，リモートネットワーク上の分配サーバへの動的な追加が行われたことが分かる．

6.3 サービス発見・連携処理時間の評価

次に，本アーキテクチャによるサービス発見と連携実行処理時間を評価するために，動作シーケンスの各ステップの所要時間を測定した．表 2 に中継端末 MN2 がローカルネットワークに接続されてから，経路統合が完了するまでの所要時間の測定結果を示す．ここで，ローカルネットワーク接続の所要時間は，MN2 の無

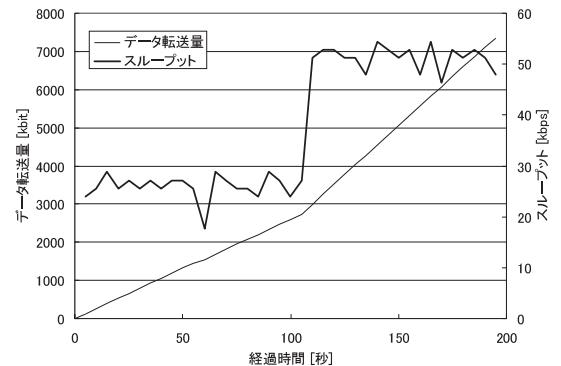


図 13 転送データ量とスループットの時間変化

Fig. 13 Time sequential variation of transmitted data amount and throughput.

表 2 各動作ステップの所要時間

Table 2 Required time for each step.

ステップ	所要時間 [msec]
中継端末のローカルネットワーク接続	3,436
中継サービス開始～発見	2,810
集約サービス情報取得	38
中継回線接続指示～完了	23,348
中継サービス情報取得	593
中継サービス設定～完了	981
中継端末の VPN 接続指示～完了	3,720
中継端末の VPN 情報取得	514
集約サービス設定～完了	50
シナリオ転送～完了	1,830
分配サーバ設定～完了	3,483
合計	40,803

線 LAN 設定インタフェース (ZeroConfig) への接続指示から、無線 LAN インタフェースに IP アドレスが割り当てられるまでの時間を計測した。表は 5 回の測定値から極端な値を除いて平均化した値を示す。ただし、ユーザによるアライアンス参加許可の応答時間は除いた。

表より中継端末のローカルネットワーク接続から経路統合の完了までの 40.8 秒のうち、集約端末上の UPnP エージェントによる中継サービスの発見に 2.8 秒、分配サーバへのシナリオ転送に 1.8 秒を要したことが分かる。

また、中継端末 MN2 の無線 LAN への接続に 3.4 秒、PHS での回線接続に 23.3 秒、VPN 接続に 3.7 秒、分配サーバの VPN 再設定に 3.4 秒を要した。これらは回線および使用した VPN ソフトの性能に依存する。

したがって、以上を除いた残りの所要時間の合計は 2.2 秒となる。これはほぼ一連のサービス連携での UPnP エージェントの処理時間の合計と考えられる。この値は本アーキテクチャを使用しない場合の中継端末追加から帯域拡大開始までの所要時間 34.0 秒と比較して小さいといえる。

なお、サービス発見の時間は使用した UPnP ライブラリの性能に依存するが、SSDP でやりとりされる XML メッセージの通信とそのパースに時間がかかっているものと考えられる。シナリオ転送の時間は PHS の通信速度とサーバ側 UA でのシナリオ展開時間と XML のパースによるものと考えられる。UPnP エージェントの処理時間は XML で表記されたシナリオのパースと実行処理の時間と考えられる。

7. ま と め

モバイルユビキタス統合環境におけるサービス連携の課題を解決する UPnP エージェントによるサービス連携アーキテクチャについて述べた。本アーキテクチャにより、業界標準である UPnP を使用しながら、リモートとローカルネットワーク上に分散するサービスの柔軟な連携が可能となる。

本アーキテクチャを応用した複数経路統合による動的帯域拡大システムの実装を行い、実環境でその評価をした。その結果、UPnP サービス化した中継ノードが分配サーバに動的に追加され、経路数に応じて帯域が拡大されることを確認した。また、UPnP エージェントによるサービス発見と処理の所要時間は本アーキテクチャを使用しない場合の中継端末追加から帯域拡大開始までの所要時間と比較して小さい値であることを確認した。以上より、本アーキテクチャの実用性を

実証した。

今後は UA による認証/アクセス制御と、サービス数の増大に備えたサービスの自動選択等の実装と評価をしたい。また、UPnP エージェントのシナリオ処理系の改善を行い、実用化を目指したい。また他のデバイスやサービスとの連携、異なるシステムとの連携等、本アーキテクチャの特長を活かした応用を行いたい。

謝辞 本試作および実験にご協力いただいた皆様に、つつしんで感謝の意を表する。

参 考 文 献

- 1) UPnP. <http://www.upnp.org/>
- 2) Ritchie, J. and Kuehnel, T.: UPnP AV Architecture, 0.83 v.1.0, UPnP Forum (June 2002).
- 3) UPnP Device Architecture 1.0 Ver.1.0.1, UPnP Forum (Dec. 2003).
- 4) Goland, Y.Y., Cai, T., Leach, P. and Gu, Y.: Simple Service Discovery Protocol/1.0 (Oct. 1999). INTERNET DRAFT, draft-cai-ssdp-v1-0.3.txt
- 5) 西ヶ谷岳, 栗田敏彦, 飯田一朗, 村上孝三: エージェント指向ネットワークアーキテクチャ DUET の提案, 電子情報通信学会論文誌, Vol.J79-B-I, No.5, pp.216-225 (1996).
- 6) Masuoka, R. and Labrou, Y.: Task Computing — Semantic-web enabled, user-driven, interactive environments, *WWW Based Communities for Knowledge Presentation, Sharing, Mining and Protection (The PSMP workshop) within CIC* (June 2003).
- 7) Kang, D.-O., Kang, K., Choi, S. and Lee, J.: UPnP AV Architectural Multimedia system with a Home Gateway Power by the OSGo Platform, *IEEE Trans. Consumer Electronics*, Vol.51, No.1, pp.87-93 (Feb. 2005).
- 8) Vilei, A., Convertino, G. and Crudo, F.: A New UPnP Architecture for Distributed Video Voice over IP, *ACM MUM'06* (Dec. 2006).
- 9) Kim, K.-S., Park, C. and Lee, J.: Internet Home Network Electrical Appliance Control on the Internet with the UPnP Expansion, *2006 International Conference on Hybrid Information Technology (ICHIT'06)* (Nov. 2006).
- 10) Ritchie, J.: MediaServer: 1 Device Template Version 1.0.1, UPnP Forum (June 2002).
- 11) Iyer, P. and Warrior, U.: InternetGatewayDevice: 1 Device Template Version 1.0.1, UPnP Forum (Nov. 2001).
- 12) JINI. <http://www.jini.org/>
- 13) Mineno, H., Ishihara, S., Ohta, K., Aono, M., Ideguchi, T. and Mizuno, T.: Multiple paths protocol for a cluster type network,

International Journal of Communication System, Vol.12, pp.391-403 (1999).

- 14) 林 孝典, 山崎真一郎, 森田直人, 相田 仁, 武市正人, 土居範久: インターネットを用いた複数経路データ伝送方式の性能評価, 信学論 (B), Vol.J84-B, No.3, pp.523-533 (2001).
- 15) 小西洋裕, 石原 進, 水野忠則: 通信回線共有方式を用いた Web アクセスの実現, 情報処理学会論文誌, Vol.43, No.6, pp.1968-1976 (2002).
- 16) Koyama, K., Ito, Y., Mineno, H. and Ishihara, S.: Evaluation of Performance of TCP on Mobile IP SHAKE, *IPJS Journal*, Vol.45, No.10, pp.2270-2278 (2004).
- 17) Ito, Y., Mineno, H. and Ishihara, S.: A scheme encouraging mobile nodes to forward packets via multiple wireless links aggregating system between the Internet and mobile ad hoc networks, *Lecture Notes in Computer Science*, Vol.3544, pp.110-123 (Aug. 2005). (Revised Selected Papers of OPODIS 2004).
- 18) Gleeson, B., Lin, A., Heinanen, J., Armitage, G. and Malis, A.: A Framework for IP Based Virtual Private Networks, RFC2764 (Feb. 2002).
- 19) Hamzeh, K., Pall, G., Verthein, W., Taarud, J., Little, W. and Zorn, G.: Point-to-Point Tunneling Protocol (PPTP), RFC2637 (July 1999).
- 20) Poptop. <http://www.poptop.org/>

(平成 18 年 11 月 2 日受付)

(平成 19 年 4 月 6 日採録)



藤野 信次 (正会員)

1960 年生。1986 年大阪府立大学大学院工学研究科電子工学専攻博士前期課程修了。同年 (株) 富士通研究所入社。デジタル無線の研究開発の後、モバイルコンピューティングの研究に従事。主管研究員。2003 年「第 3 世代移动通信網に適用するモバイルインターネットプロトコルの研究開発と標準化」で情報処理学会業績賞受賞。



塩内 正利 (正会員)

1962 年生。1987 年九州大学大学院総合理工学研究科修了。同年 (株) 富士通研究所入所。以来、機械翻訳、自然言語処理、知識処理、マルチエージェントシステムの研究に従事。



福田 茂紀 (正会員)

1974 年生。1999 年東京大学大学院工学系研究科電子情報工学専攻修士課程修了。同年より (株) 富士通研究所に勤務。パーソナルエージェント技術の研究開発に従事し、特にモバイル機器や遠隔機器、周辺機器をユーザ視点で連携させる技術に関わる。



石原 進 (正会員)

1972 年生。1994 年名古屋大学工学部電気学科卒業。1999 年同大学大学院工学研究科博士後期課程修了。1998 年日本学術振興会特別研究員。1999 年静岡大学情報学部助手。2001 年同大学工学部助教授。2006 年同大学創造科学技術大学院助教授。2007 年同准教授。工学博士。1997 年電気通信普及財団テレコムシステム技術学生賞受賞。モバイルコンピューティング、無線環境用 TCP/IP、アドホックネットワークに関する研究に従事。電子情報通信学会、IEEE、ACM 各会員。



水野 忠則 (フェロー)

1945 年生。1969 年名古屋工業大学経営工学科卒業。同年三菱電機 (株) 入社。1993 年静岡大学工学部情報知識工学科教授。1996 年情報学部情報科学科教授。2006 年より創造科学技術大学院院長。工学博士。情報ネットワーク、モバイルコンピューティング、放送コンピューティングに関する研究に従事。著訳書としては『コンピュータネットワーク』(日経 BP)、『モダンオペレーティングシステム』(ピアソン・エデュケーション) 等がある。電子情報通信学会、IEEE、ACM 各会員。情報処理学会フェロー、監事。