サービス指向グルーピング機構を用いたユーザ主導ネットワークの構築

三村 和† 飛岡 良明†*1 森川 博之† 青山 友紀‡

†東京大学大学院工学系研究科
‡東京大学大学院情報理工学系研究科

あらまし 自分が所有する，特定の部屋に存在するなどの限られた端末グループだけにサービスを公開するよう制御できれば，インターネットはユーザにとってより利便性の高いものになろう．本稿では，ユーザがネットワークレベルで端末のグルーピングを行い，それに基づいてサービス間における通信制御を行う枠組みとしてMyNetSpaceシステムを示す．これによれば，IP層から分離した形でグルーピング機構を提供し，端末は複数のグループに同時に属することができる．

A User-controlled Network Construction using Service-orieted Grouping Mechanism

Nodoka MIMURA† Yoshiaki TOBIOKA† Hiroyuki MORIKAWA† Tomonori AOYAMA‡

†Graduate School of Engineering, The University of Tokyo
‡Graduate School of Infomation Science and Technology, The University of Tokyo

Abstract The Internet will become more convenient and friendly for users, if it can provide services, for example, to one's own devices only or to the devices in a specified room. It is effective for users to group those devices in the network level and to control the communication between services on the basis of the groupings. In this paper, we present the design and implementation of the MyNetSpace system which is a flexible grouping framework independent of the IP routing.

1. はじめに

インターネットはグローバル規模で任意のIPアドレスへの到達性を確保することを第一目的とし，それを達成した [1]．しかしながら，ユーザにしてみれば全世界の端末と平等に通信するだけでなく，その中の限られた端末とだけ通信したいという要求も存在する．たとえば，家の共有ファイルを公開する端末に世界中どこからでも，しかしいくつかの自分の携帯端末だけからアクセスを許可したい．このような要求は，現状のVirtual Private Network（VPN）ツール [2–4] やパーソナルファイヤウォールの利用からも覗える．

今後，身の回りのあらゆるものが計算能力をもち，ネットワーク上に遍在化するに伴い，ユーザはそれらの上で動作する多種多様なサービス（アプリケーション）をインターネットを介して利用するようになる．さらには，Webサービスのようにサービスのモジュール化が進めば，サービス同士の繋がりがより重要になるだろう．このような環境では，ユーザの要求を満たす制御機構を個別に実装するのではなく，共通の基盤として提供することが必要である．

そのような共通基盤を提供するにあたり，筆者らは端末のグルーピングに着目する．サービスを公開する，またはサービス間で連携する場合，自分が所有する端末，特定の部屋にある端末などのようにその通信範囲は限定される．また，通信範囲に含まれる端末が動的に変化したり，サービスが通信範囲そのものを変更したりするであろう．したがって，図1に示すように端末の閉域グループを作り出し，サービスの通信範囲をその閉域グループ内に限定することが効果的である．このとき，端末上では多くのサービスが動作するので，端末は複数のグループに参加できることが求められる．このようにサービス単位で端末のグルーピングを行うことを，筆者らはサービス指向グルーピングと呼ぶ．

本研究の目的は，ユーザが柔軟に端末のグループを作り出し，それに基づいてサービス間における通信制御を行うミドルウェアを提供することである．これに向けて，筆者らはMyNetSpace（MNS）システムの検討を進めている．MNSは各々のユーザが定義する閉域グループで，MNSに参加しない端末との通信を許可しない．また参加認証機能をもち，ユーザが所有する端末や特定の部屋にある端末といった条件を満たさなければ端末は参加することができない．一方で，端末は条件を満たせばこれら複数のMNSに同時に参加できる．サービスと各MNSを結びつけることによって，サービスは閉域グループ内で通信を行うことができる．

MNSシステムの設計指針の1つは，現状のインターネットアーキテクチャとの親和性を考慮し，サー

*1 現在，NTTコミュニケーションズ株式会社

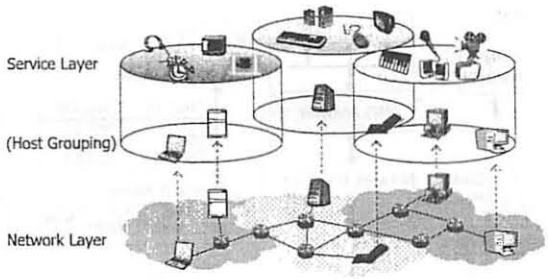

図 1: 端末のグルーピングによるサービス管理

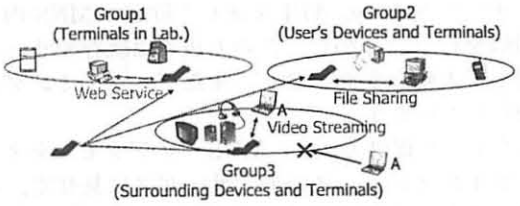

図 2: 複数のグループへのサービスの公開

ビス指向グルーピング機構をネットワークレベルで提供することである．サービスは，MNS ごとに専用の仮想ネットワークインタフェース（VNI : Virtual Network Interface）を介して，MNS 内部との通信を行う．VNI を通ったパケットには各 MNS に固有な識別子（MNSID）がラベル付けされ，それを解析することによって適切な VNI へと転送される．このようにネットワークレベルで実現することにより，サービスは TCP などのトランスポート層でのプロトコルを利用することができる．

もう 1 つの設計指針は，通信を許可する MNS をサービスが明示的に選択することである．既存手法 [5,6] ではルーティングによって暗黙的にグループを選択する．グループごとに新たに仮想 IP アドレス空間を割り当て，サービスからは相手の仮想 IP アドレスを指定する．これは既存のサービスに対して変更を要さない反面，いくつかのインターネットアーキテクチャ上の制約を生む．端末が複数のグループに属する際に，割り当てたアドレス空間が衝突したり，サービスが意図しないパケットを受信したりする可能性がある．また，ユーザが設定ミスを行った際に他の実通信に影響を与える可能性もある．将来的にユーザが銘々に多数のサービスを扱うようになるならば，ユーザ主導なグルーピング機構と管理者主導な IP ネットワークの機構とは完全に分離するべきであろう．本手法では，サービスが通常の物理インタフェース間での通信に加えて，明示的に VNI に対してバインドすることによって MNS 内での閉域な通信を実現する．

以上を踏まえ，MNS システムを MNS Analyzer, MNS Server, MNS Manager という 3 つの要素から構成した．MNS Analyzer は MNS 内におけるサービス間の通信を実現する．VNI を通過したパケットに対応する MNSID を付加することで通信の識別を行う．MNS Server と MNS Manager は MNS の参加端末の管理を実現する．MNS Server は MNS ごとに 1 つ存在して参加認証などを行う．MNS Manager は個々の端末に 1 つずつ存在して MNS への参加要求を出し，また VNI や MNS Analyzer の設定などを行う．

本稿の残りの構成は以下の通りである．2. ではサービス指向グルーピングと設計指針について説明する．3. では MyNetSpace システムの構成について述べる．4. では MyNetSpace システムのプロトタイプ実装について述べる．5. では関連研究との比較を示し，最後に 6. でまとめとする．

2. サービス指向グルーピング

2.1 グルーピングに基づくサービス管理

1 台の端末上で複数のサービスを動かしているときに，サービスごとに端末をグルーピングを行う様子を図 2 に示す．ここでは，研究室にある端末に対して Web サービスを，ユーザが所有するデバイスや端末に対してファイル共有を，周辺のデバイスや端末に対してビデオストリーミングを公開する．この場合，それぞれの端末群に対応した 3 つのグループを作り，サービスの公開先を選択するという操作を行う．公開されたサービスは，それぞれのグループに参加した端末から利用されることになる．たとえば，端末 A がサービスを公開する端末の周辺に存在する場合，端末 A は Group3 に参加してビデオストリーミングを受信することができる．逆に，端末 A が離れていった場合には Group3 への参加権を失うため，ビデオストリーミングを受信できなくなる．

このようなサービス指向グルーピング機構を実現するためには，各グループ内部の端末間通信を識別してサービスに振り分ける機構と，認証を通して端末をグループに参加させる機構が要求される．前者では，端末が複数のグループに属することになるため，物理的には同一端末宛の通信であってもそれぞれの通信がどのグループにおけるものなのかを識別しなければならない．後者では，ユーザが定義した参加条件にしたがって認証や監視を行い，端末の参加・脱退を管理しなければならない．参加条件とは，具体的には，‘研究室にある端末’ や ‘ユーザが所有するデバイス’ などのメタ情報である．実際には，RFID を用いた物理位置情報の取得やユーザ ID の入力などの外部機構と連携するよう設定される．また，グループ内のサービス間で通信を行う際，端末がグループに参加していることを証明できなければならない．

2.2 ネットワークレベルでの提供

MNS システムは，2.1 のようにユーザが柔軟に端末の閉域グループを作り出し，それに基づいてサービス間における通信制御を行うミドルウェアである．上記の例では端末のグループが MNS に相当する．

このような機構をサービスに提供するには 2 つの方法が考えられる．1 つは，ライブラリの中で新たなアプリケーションプログラムインタフェース（API : Application Program Interface）を定義する方法であ

る．サービスはこの API を通して特定の MNS 内部での通信を行う．しかし，この方法では既存のサービスに対して大幅な変更を要し，またプログラミング言語を限定してしまう．

もう 1 つの提供方法は，MNS へのアクセスをネットワークインタフェースの形でサービスに見せて，ネットワークレベルで提供する方法である．サービスは，MNS ごとに専用の VNI : Virtual Network Interface を介して MNS 内部との通信を行う．VNI を通ったパケットは各 MNS に固有な MNSID が付加され，それの解析によって適切な VNI へと転送される．VNI としてサービスに見せるので Socket API をそのまま利用でき，サービスの作成において既存のプログラミングフレームワークを崩すことなく，グルーピング機構を導入できる．筆者らはなるべく現在のインターネットアーキテクチャと親和するように新たな機構を導入することが重要であると考え，後者の方法を採用する．

2.3 グループの明示的な選択

端末が複数のネットワークに属する場合，インターネットには強モデルと弱モデルという 2 つの考え方がある [7]．強モデルはインタフェースについたアドレスで待ち受けを行い，弱モデルではホストについたアドレス（すなわち *INADDR_ANY*）で待ち受けを行う．一般にリンク層は強モデル，ネットワーク層は弱モデルで設計されている．後者の理由は，マルチホーム環境においても任意の IP アドレスへの到達性を柔軟に確保したいからである．そのため，現在のサービスも基本的には弱モデルで設計されるものが多い．しかし本研究で想定するのは，サービス自身が通信範囲を指定する環境である．筆者らは強モデルに基づいたサービスの構築が今後は重要であると考える．したがって，サービスが通信範囲となる閉域グループ（すなわち MNS）を明示的に選択するという方針を採る．

MNS システムでは具体的には，サービスが VNI に対してバインドすることによってグループの明示的選択を行う．これにより，ユーザ主導な枠組みと管理者主導な IP ネットワークの機構とを分離して考えることができ，機構を簡単化することができる．既存手法 [5,6] のように VPN 技術を用いて閉域グループを構築する場合，それぞれのグループに対して新たに仮想 IP アドレス空間を割り当てる．サービスは端末上のルーティングテーブルにしたがって暗黙的にグループを選択することになる．これらは既存のサービスに対して変更を要さない反面，いくつかのインターネットアーキテクチャ上の制約を生む．端末が複数のグループに属する際に，割り当てた仮想 IP アドレス空間が衝突したり，弱モデルであるためにサービスが意図しないパケットを受信したりする可能性がある．また，ユーザが設定ミスを行った際に他の実通信に影響を与える可能性もある．本手法では，サービスが物理インタフェースに付いた実 IP アドレスでの通信に加えて，明示的に VNI に対してバインドすることによって MNS 内での閉域な通信を実現する．

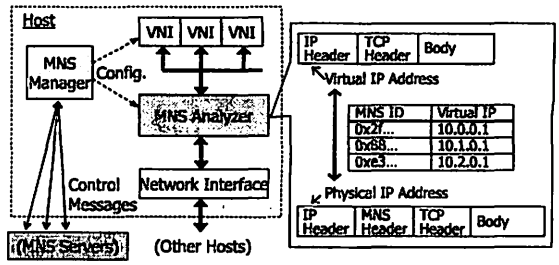

図 3: システム構成要素の関係

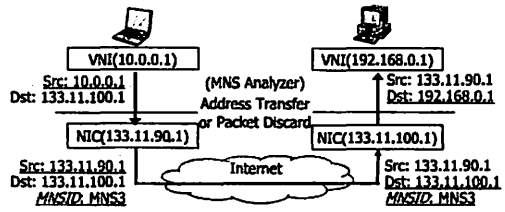

図 4: MyNetSpace 内部におけるパケット送受信の動作詳細

3. MyNetSpace システム

3.1 システム構成

MNS システムは，内部通信機構と端末管理機構の 2 つから構成される．前者は，TCP などの既存通信機構に影響を与えずに VNI 間で正しくパケット送受信を行う機構であり，MNS Analyzer によって実現される．後者は，参加認証や監視を行うことによって端末が MNS に所属していることを保証する機構であり，MNS Manager と MNS Server によって実現される．

これら 3 つの要素の関係を図 3 に示す．MNS Analyzer は，VNI と物理ネットワークインタフェースの間を仲介し，適切なパケットの受け渡しを行う．MNS Server は MNS ごとに 1 つ存在し，参加の認証や MNSID の配布などを行う．MNS Manager は各端末ごとに 1 つ存在し，MNS Server と制御メッセージのやり取りを行って端末を MNS へ参加させる．また，各端末内における VNI や MNS Analyzer の設定を行う．

3.2 内部通信機構

図 4 に同一の MNS に参加する端末間におけるパケット送受信の動作詳細を示す．それぞれの端末は，MNSID が 'MNS3' であるという情報だけを共有し，お互いの VNI に付けられる仮想アドレスの情報は共有しない．

MNS 内部の通信は以下の要領で実行される．MNS Analyzer にはあらかじめ，自端末の VNI の仮想 IP アドレスと MNSID との対応表が設定されている．送信端末では，VNI に割り当てた仮想 IP アドレス (10.0.0.1) を送信元に設定し，通信相手の実 IP アドレス (133.11.100.1) を宛先に設定する．MNS Analyzer は，VNI を通過したパケットがインターネットへ送出される前に，送信元を仮想 IP アドレスから自端末の実 IP アドレス (133.11.90.1) へと変換し，パケッ

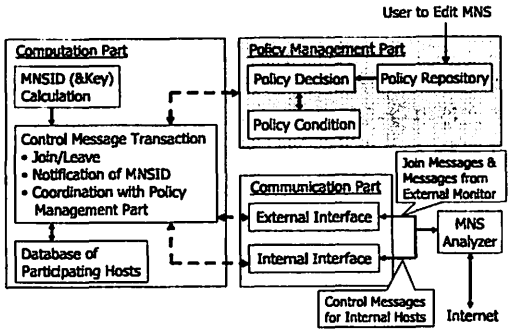

図 5: MNS Server のフレームワーク

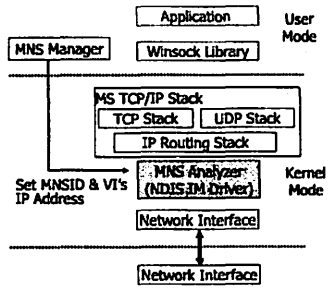

図 6: Windows XP SP1 における実装概念図

トに MNSID (MNS3) を付加する．したがって，インターネット上では通信を行う双方の実 IP アドレスと MNSID がパケットに記載されることになる．受信端末では，MNS アナライザが MNSID をもとに対応表を参照し，宛先を該当する VNI の仮想 IP アドレス (192.168.0.1) に変換する．その後，MNSID を削除して該当の VNI へとパケットを転送する．MNSID がない，あるいは MNSID が対応表と一致しないパケットは VNI へと転送されないので，仮想的に閉域ネットワークが実現される．

3.3 端末管理機構

ユーザが端末を MNS に参加させる場合，端末上の MNS Manager を通して MNS Server との間で制御メッセージを交換して認証を行う．認証後，MNS Server から MNSID と共通鍵が配られる．この鍵は，MNS から脱退した端末が既知の MNSID を使って通信を行うことを防いだり，通信の暗号化を行うのに用いられる．MNS Manager は参加した MNS に対応した VNI を作成して仮想 IP アドレスを割り振る．その後，MNS Server から受信した MNSID と鍵を VNI との対応情報とともに MNS Analyzer へ通知する．

MNS Server は，ユーザが MNS を作成するごとに 1 つ起動されるプロセスである．その構成は図 5 に示すように，計算部 (Computation)，ポリシ管理部 (Policy Management)，通信部 (Communication) の 3 つに分かれる．計算部は，MNS に参加する端末の情報保持，参加・脱退などの制御メッセージ処理，それに応じた MNSID と鍵の作成・配布を行う．この計算部は MNS の設定に関わらず，共通の処理を行う．

ポリシ管理部は，ユーザが定義した MNS への参加ポリシと現在の状態を照合しながら参加の許否を決定し，計算部に対して通知する．ユーザが MNS を作成する際には，このポリシ管理部の動作を決めることになる．たとえば，ある部屋にいる端末が MNS へ参加できるようにするには，外部の端末監視機器から情報を取得して現在の状態を記録し，その情報がない端末からの参加要求を受け付けない，あるいは脱退させるという動作を決定する．

通信部は，外部と内部のインタフェースをもち，それぞれで通信の待ち受けを行う．内部インタフェースを通しては，MNS 内部の端末と制御メッセージの交換を行う．外部インタフェースを通しては，端末から MNS への参加要求を受け付け，またポリシ管理のための外部機構から情報を取得する．この外部インタフェースに特定の VNI を指定することで，ある MNS に参加する端末だけから参加要求を受け付けるという動作を容易に実現することができる．

4. 実装

4.1 実装の概要

MyNetSpace システムのプロトタイプの実装を Windows XP SP1 上にて行った．図 6 に実装概念図を示す．MNS Analayzer は，Windows ネットワークスタックの最下層に位置する NDIS (Network Driver Interface Specification) 中間ドライバに実装され，カーネルモードで動作する．MNS Server と MNS Manager はユーザモードで動作するプログラムとして実装される．

4.2 内部通信機構の実装

MNS Analyzer は OS の TCP スタックよりも下位層でパケットの書き換えを行う必要があるため，NDIS 中間ドライバとして実装した．NDIS とは Microsoft 社が定めたネットワークドライバの標準仕様であり，その中間ドライバはネットワークドライバにミドルウェアとして機能を追加できるように準備されている．MNS Analyzer は，ドライバレベルでパケットを捕まえて 3.2 で述べた操作を行い，加えて IP，および TCP/UDP チェックサムの再計算を行う．また，MNS Manager から MNS Analyzer に対して VNI と対応する MNSID，鍵の情報を通知するために，ドライバ間の通信のために規定された I/O Request Packet を利用した API を用意した．

既存サービスから MNS の内部通信を利用できるようにするため，VNI へはローカル IP アドレスを割り振る．サービスは VNI の IP アドレスに対してバインドし，通信する MNS を選択する．ただし，理想的には新たなアドレス空間を定義して，既存 IP 層から完全に分離させることが望ましいことを付け加えておく．また，バインドのオプションをもたない既存サービスも多く存在するので，Winsock ラッパライブラリを用意した．これは，バインドさせたい MNS を MNS Manager からあらかじめ選んでおくと，TCP 接続を行う，すなわち Connect 関数を呼び出す際に適切な VNI に対して

Bind 関数を呼び出す.

4.3 端末管理機構の実装

端末管理機構の実装では，MNSID の計算，鍵の変更，MNS 作成時の設定項目の 3 つを特に考慮する．MNSID の計算については，ユニークな値を算出する必要がある．本実装では 128 ビットの乱数として計算し，確率的に MNSID の衝突を回避するようにした．もし衝突した場合には，両方の MNS に参加した端末の MNS Manager にて検出し，MNS Server に対して MNSID を変更するように要求を出す．

鍵の変更については，厳密には端末の脱退が生じる度に更新することが要求される．しかし，この方法は端末の MNS への出入りが激しい環境においてはあまり効率的ではない．そのため，本実装ではより汎用的な利用を意図し，定期的に鍵を更新する方法を採った．

MNS 作成時の設定項目については，作成者は名前や MNS Server のアドレスなど以外に，参加認証のために MNS Server のポリシ管理部の設定を行う必要がある．しかし，ポリシ管理において用いる情報は位置情報やユーザ情報など多種に渡り，さらには端末の物理位置監視やユーザ情報データベースなどの外部機構との連携も必要になる．そのため，汎用的なルールやメッセージフォーマットの記述方式が課題となる．これに向け，筆者らは RFID 基地局，およびタグリーダ付携帯電話との連携を検討した [8]．今後はより詳細に検討を行う予定である．

5. 関連研究

Regions プロジェクト [9] では，将来の大規模，広分散，かつヘテロなネットワークのためのアーキテクチャ設計で鍵となるのはグルーピング機構であると主張する．ネットワーク上のエンティティ（ルータや端末）は必ず，Region と呼ぶ何かしらのメタ的な意味をもつグループに属する．このプロジェクトでは，グローバル規模で汎用的にネットワークを区切ることにより IP ネットワークを管理することを目的とする．一方，本研究ではエンド端末の構成に着目し，ユーザが柔軟に端末の閉域グループを作り出し，その特性をサービスから利用できる枠組みの構築を目指す．

既存の VPN 技術 [2,3] は，物理的に離れた場所にある端末や LAN をトンネリングで接続してプライベートネットワークへと仮想的に参加させ，その内部への資源に対して安全にアクセスさせることを主目的する．これらはルーティングにその機能を埋め込むためにアプリケーションへの変更を要せず，現実のインターネットに即した非常に有効な手法である．P2PCUG [5] や ELA [6] では，この VPN 技術を用いて閉域なユーザグループを構築する．しかし，VPN 内部で用いられる仮想インタフェースに割り当てられる IP アドレス（およびアドレス空間）は VPN 管理者が決めるものであるため，ユーザが自由にグループを定義することは難しい．本研究では，IP ネットワークの管理とユーザによるグルーピングを完全に分離することでインターネットの利便性を向上することを目指す．

JXTA [10] で提供される PeerGroup では，オーバレイネットワークで論理グループを形成し，JXTA 上で提供される専用インタフェースと API を通してその内部のリソースやサービスへのアクセスを許可する．しかし，JTXA プラットフォーム上で作成されたプログラムでしか動作せず，既存のものに対してはプログラミングフレームワーク自体の変更を強いることになる．

6. おわりに

本稿では，ユーザが柔軟に端末のグループを作り出し，それに基づいてサービス間における通信制御を行う枠組みとして MNS システムを示した．MNS はネットワークレベルでユーザが構築する仮想的な閉域グループである．端末を複数の MNS に同時に参加させ，端末上で動くサービスを各 MNS と結びつけることによってサービス指向グルーピングを実現する．また，本稿では Windows 上での実装について述べたが，現在では多様化する端末環境を考慮して Linux 上での実装も行っている．今後は，端末管理機構の詳細化や新たな適用シナリオなどの検討を進める．

謝辞

本研究は，総務省からの委託研究の成果である．

参考文献

[1] D. D. Clark. The design philosophy of the DARPA internet protocols. *Proc. SIGCOMM 1988*, Aug. 1988.

[2] K. Hamzeh, G. Pall, W. Verthein, J. Taarud, W. Little, and G. Zorn. Point-to-Point Tunneling Protocol (PPTP). RFC 2637, IETF, July 1999.

[3] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, and B. Palter. Layer Two Tunneling Protocol (L2TP). RFC 2661, IETF, Aug. 1999.

[4] OpenVPN. http://openvpn.sourceforge.net/

[5] 藤田範人, 小出俊夫, 石川雄一, 塚本明. ピアツーピア型レイヤ 2 仮想ネットワークの提案. 信学ソ大, B-6-59, Sept. 2004.

[6] S. Aoyagi, M. Takizawa, M. Saito, H. Aida, and H. Tokuda. ELA: A Fully Distributed VPN System over Peer-to-Peer Network. *Proc. SAINT 2005*, Jan. 2005.

[7] R. Braden, Editor. Requirements for Internet Hosts – Communication Layers. RFC 1122, IETF, Oct. 1989.

[8] 平井璽, 三村和, 森川博之, 青山友紀. タグリーダ付携帯電話を用いた近傍仮想ネットワークの検出・参加機構. 信学ソ大, Sept. 2005.

[9] K. R. Sollins. Designing for Scale and Differentiation. *Proc. FDNA-03*, Aug. 2003.

[10] JXTA. http://www.jxta.org/