インターネットにおけるパーソナルネットワークの構築

光来 健一 * 千葉 滋 †‡

* 東京大学大学院 理学系研究科 情報科学専攻
† 東京工業大学 情報理工学研究科 数理・計算科学専攻
‡ 科学技術振興事業団 さきがけ研究 21

要 旨

近年、ユーザは複数のネットワークを使うことが多くなっているが、位置情報に基づいてアクセス制御を行っている現在のネットワーク環境では、ユーザがリモートネットワークを十分に利用するのは難しい。ユーザ中心のネットワークを実現するために、我々は仮想プライベートネットワークにユーザの概念を取り入れた**パーソナルネットワーク**を提案する。パーソナルネットワークはユーザが専用の仮想ネットワークを使えるようにするために、通信路などのネットワーク資源、ルーティングなどのネットワーク設定、ファイアウォールなどのゲートウェイ設定をパーソナライズする。我々はパーソナルネットワークの機能の一部を実装し、ファイアウォールなどのために直接通信することができないホスト同士のプライベート通信を可能にした。

Constructing Personal Network on the Internet

Kenichi Kourai* Shigeru Chiba†‡

* Dept. of Information Science, Graduate School of Science, University of Tokyo
† Dept. of Mathematical and Computing Sciences,
Graduate School of Information Science and Engineering, Tokyo Institute of Technology
‡ PREST, Japan Science Technology Corp.

Abstract

Recent users are often working in multiple network environments. Nevertheless, since today's network environments perform a location-centric access control, it is difficult to sufficiently use the abilities of remote network environments from local network environment. Toward constructing user-centric networks, we propose *personal network*, which introduces the notion of users into virtual private network. For each user to use his/her own virtual networks, personal network personalizes network resources like communication channels, network settings like IP routing, and gateway settings like firewall rules. We have implemented a part of the functions that personal network needs and enabled private communication between hosts that cannot directly communicate each other due to firewall.

1 はじめに

近年、ユーザは複数のネットワークを使うことが多くなってきている。大学では研究室のネットワークを使い、自宅では ADSL やケーブルテレビで常時接続されたネットワーク、出張先ではダイアルアップ接続されたネットワークを使う、などである。しかし、それぞれのネットワークはセキュリティ上の問題を防ぐために閉鎖的であり、異なるネットワークの間での通信は非常に不便である。自宅からメールを読むためだけに、わざわざ研究室のホストにログインしたりする必要がある。また、異なるユーザ同士でサーバを介さずにピア・ツー・ピアの通信を行うことも増えてきている。しかし、2 つのホストが異なるネットワーク上にあり、それぞれのネットワークのファイアウォールで守られている場合やプライベートアドレスしか持っていない場合には、通信を行うことができない。

このような問題は各ネットワークのサーバやファイアウォールが位置情報、つまり、IP アドレスに基づくアクセス制御を行っているために生じる。様々なサイトにアカウントを持つユーザはそれぞれのネットワークの IP アドレスを持ったホストを使っており、その IP アドレスでの他のネットワークへのアクセスは著しく制限される。異なるネットワークに属するホスト同士では、ファイアウォールのために、許可されたごくわずかのサービス以外に関しては、互いにネットワーク接続をすることができない。そのため、位置情報に代わりユーザ情報を中心にしたネットワークが必要とされている。ネットワーク環境がユーザという概念をもっと利用できれば、異なるネットワークに属するホストを使用していても、ユーザ認証により安全なプライベート通信が可能になる。

そこで、我々は、ユーザ中心のネットワークを実現するために**パーソナルネットワーク**を提案する。パーソナルネットワークとは、インターネットを介したホスト間に構築される仮想プライベートネットワーク（VPN）にユーザの概念を取り入れたものである。パーソナルネットワークでは、ユーザ専用の仮想ネットワークを張れるようにするために、通信路などのネットワーク資源、ルーティングなどのネットワーク設定、ファイアウォールなどのゲートウェイの設定をパーソナライズする。例えば、リモートホストとの間の通信路をユーザ毎に暗号化したり、ゲートウェイでユーザ情報に基づいてパケットの中継先を変更できるようになる。

我々は、ファイアウォールなどのために直接通信することができないホスト同士の間でプライベート通信ができるように、パーソナルネットワークの最低限必要とされる機能を実装した。まず、カーネルレベルで Secure Socket Layer (SSL)[1] をサポートすることにより、ユーザ認証および通信路の暗号化をアプリケーション透明に行い、ユーザ専用の仮想ネットワークを張ることができるようにした。さらに、ゲートウェイでユーザ認証に基づいてプライベート通信を中継できるようにし、安全にリモートネットワーク内部のホストと通信することを可能にした。

2 ユーザ中心のネットワークの必要性

OS のアクセス制御にはユーザという概念がよく用いられているが、従来のネットワーク環境ではあまりユーザという概念が使われてこなかった。例えば、ゲートウェイでのファイアウォールによるパケットフィルタリングでは、必要に応じてパケットを通すようにしておき、内側のサーバでユーザを認証している。これはサーバプログラマの負担になり、パケットをネットワークの内側に入れることにより危険度が増す。また、サーバが提供するサービスをユーザ毎に変える機能もサーバアプリケーション次第であり、OS によるサポートはない。さらに、OS のネットワークの設定に関してもユーザ毎にカスタマイズすることはできない。一つの OS は一つのホスト名、ドメイン名、ルーティングテーブルしか持たず、DNS サーバなどの設定も固定である。

従来のネットワーク環境でのアクセス制御にはホストの位置情報、つまり、IP アドレスが用いられることが多い。IP アドレスは組織に対応するように割り当てられるので、IP アドレスによるアクセス制御は通常はうまく働く。例えば、ある組織に対応する`131.112.40.128/26` の IP アドレスを持つホストにだけローカルウェブページの閲覧を許可することなどができる。しかし、ユーザが複数の組織に所属している場合には位置情報によるアクセス制御はうまく働かない。ユーザがアカウントを持っているサイトであっても、それがリモートにあると十分な

サービスを受けることができなくなる。例えば、リモートサイトのローカルウェブページを閲覧するためには、その内部のホストにログインする必要があるが、テキストベースのブラウザしか使えないなどの不便がある。このような場合には特定のリモートホストにも許可を与えるという方法が取られることが多いが、リモートホストの IP アドレスが DHCP で動的に割り当てられる場合や、プライベートアドレスしか持たない場合にはうまくいかない。

このように、ユーザが一つの組織にだけ属するのではなく、複数の組織にまたがって属している場合を考えると、従来の位置中心のネットワークではなく、ユーザ中心のネットワークが望まれる。ユーザ中心のネットワークでは、ネットワークトポロジに縛られず、ユーザがアクセスを許可されているネットワークの持つ能力を存分に使うことができる。

ユーザ中心のネットワークを使えば、リモートサイトにログインせずに、リモートサイト内にいれば利用できるサービスを享受することができる。例えば、リモートサイトのローカルウェブページの閲覧やリモートサイトから閲覧が許可されている他のサイトのローカルウェブページの閲覧が可能である。さらに、リモートサイト内でだけ使えるメールサーバや高速で通信できるネットワーク経路を利用することもできる。

また、ユーザ中心のネットワークでは、サイト同士の信頼関係ではなく、ユーザ同士の信頼関係に基づいてプライベート通信を行うことができる。例えば、特定のユーザからのパケットだけはファイアウォールを越えてネットワーク内部に入ってこられるように設定することができる。これによりホスト同士のピア・ツー・ピアのプライベート通信が容易になる。さらには、複数のユーザが集まって一つの新しいサブネットを形成することもできる。この仮想的なサブネットでは NFS や NIS などを使うことも可能である。そのため、頻繁な組織改変にも簡単に対応することができる。

3　パーソナルネットワーク

ユーザ中心のネットワークを実現するために、我々は**パーソナルネットワーク**を提案する。パーソナルネットワークとは、仮想プライベートネットワーク（VPN）にユーザの概念を取り入れたものである。ここで、VPN とは、インターネットを介するホスト間に構築される、外部から干渉されない仮想的なネットワークを指す。パーソナルネットワークでは、ユーザが他のユーザから干渉されない自分専用の仮想ネットワークを張れるようにするために、リモートホストとの間の通信路をユーザ毎に暗号化する。また、ユーザ毎に複数の異なる仮想ネットワークを使うことができるため、ルーティングや IP アドレスなどのネットワーク設定もユーザ毎に設定することができる。さらに、ゲートウェイなど仮想ネットワークの経路上にあるホストでも、ユーザ情報に基づいてパケットを中継するかどうか、どこに中継するかなどを変更することができる。パーソナルネットワークの利用例を図 1 に示す。

3.1　ネットワーク資源のパーソナライズ

パーソナルネットワークでは、ホスト間に特定のユーザにしか使えない仮想ネットワークを張ることができる。仮想ネットワークを張ったユーザにだけ使用を許可するために、双方のホストでユーザのチェックを行う。ユーザのチェックにはプロセスのユーザ ID やパケットの署名が用いられる。また、他のユーザに仮想ネットワークを流れるデータを盗み見られないようにするために、仮想ネットワークに対してユーザ毎に異なる暗号化を行う。VPN ではインターネットを流れるパケットに対して暗号化を行うが、パーソナルネットワークではユーザ単位での機密性を保持するためにインターネットを通らない場合、つまり、組織内のネットワークを通る場合であっても通信の暗号化を行う。

また、サーバにおけるサービスの入口であるネットワークポートに対して、通信相手のユーザによって異なるサービスを割り当てることができる。この機能により、ウェブサービスは常に同一のポート（80 番）で提供しつつ、アクセスしてくるユーザ毎に別々のサーバを動かすことができる。例えば、デフォルトでは CGI の実行を許さないウェブサーバが動いていても、許可されたユーザは CGI の実行を許すウェブサーバにアクセスすることができる。

3.2　ネットワーク設定のパーソナライズ

パーソナルネットワークでは、ユーザまたはプロセス毎に異なるネットワーク設定を行うことができ

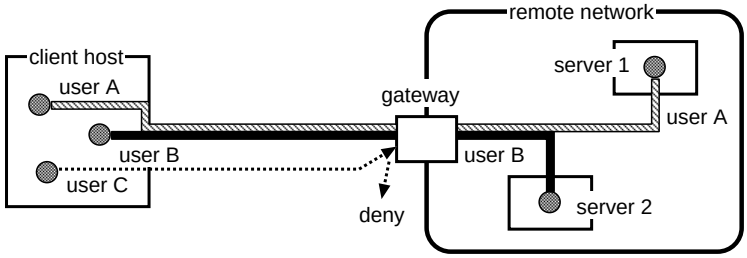

図 1: パーソナルネットワークの利用例

る。一例を挙げると、以下の設定を変更することができる。

ルーティングテーブル 複数の経路がある時に、どの経路を優先的に選択するかをユーザが決めることができる。また、特定のネットワークに向かうパケットをルーティングしないようにすることもできる。

IP アドレス 特定のユーザやプロセスにだけ別の IP アドレスを割り当てることができる。これは IP エイリアスの機能に似ているが、全てのプロセスに複数の IP アドレスが見えるようになるわけではない点が異なる。

ホスト名・ドメイン名 ホストが複数のネットワークに接続されている時などに、どのネットワークにサービスを提供しているかによって、プロセスに割り当てられるホスト名やドメイン名を変えることができる。

サーバ設定（DNS、NIS 等） ホストが複数のネットワークに接続されている時などに、それぞれのネットワークに対して各種サーバを設定することができる。

ドメイン名の検索順序（`/etc/resolv.conf`） ユーザが複数のサイトにアカウントを持っている場合、全てのサイトのドメイン名を検索する対象にすることにより、FQDN の長い名前で指定しなくてよくなる。

ホスト名データベース（`/etc/hosts`） ユーザがホスト名のニックネームを登録することができ、長い名前を持つホストや無意味な数字からなる DHCP ホストにアクセスしやすくなる。

変更されたネットワーク設定は基本的に親プロセスから子プロセスへと継承される。ただし、プロセスが`setuid` された場合にはそのユーザの設定へと変更される。

3.3 ゲートウェイのパーソナライズ

パーソナルネットワークでは、ユーザが自分のネットワークのゲートウェイに専用のフィルタリングルールや中継ルールを設定することができる。これらのルールがそのユーザに関する通信にだけ適用されるようにするため、ゲートウェイで通信相手を認証する。これにより、通信相手のホストがファイアウォール内部にある場合やプライベートアドレスしか持たない場合であっても、ゲートウェイでユーザを認証し、認証に成功したユーザからの通信だけを内部ホストに転送するように設定することができる。また、通信相手のユーザによってパケットの中継先を変えることも可能である。

ゲートウェイで通信の中継を行う際に、送信元の IP アドレスをゲートウェイの内部アドレスに変換することにより、ゲートウェイが仮想的にリモートユーザの役割を果すことができる。さらに、ユーザ情報に基づいて、中継する仮想ネットワークに異なる IP アドレスを割り当てることにより、ネットワーク内部のホストはどのユーザからの通信であるか判断することができる。これにより、従来のネットワークで行われている位置情報によるアクセス制御にも対応することができる。もちろん、内部ホストの OS がパーソナルネットワークのユーザ認証に対応していれば、直接ユーザ情報を用いてアクセス制御を行うことができる。

ゲートウェイはユーザを認証した後、アプリケーションとゲートウェイ、ゲートウェイとサーバの間の 2 つの暗号化通信路を中継する。ゲートウェイで通信内容を復号できるようにすることで、ユーザ情報に基づいて、アプリケーションデータによるフィ

ルタリングを行うことが可能になり、通信記録をログに残すこともできる。ゲートウェイで通信内容を解析する必要がない通信に関しては、アプリケーションとサーバの間で 1 つの暗号化通信路を作り、ゲートウェイではパケットの中継だけを行うようにして、通信の高速化を図ることも考えられる。

3.4 異なるユーザ同士の安全な通信

異なるユーザ同士がパーソナルネットワークを用いてプライベート通信を行う場合、互いに通信相手のユーザを十分に信頼できなければ、さらなる安全性を考慮しなければならない。パーソナルネットワークでは、通信相手に攻撃されて許可していないサービスを使われたり、他のプロセスに干渉されたりするのを防ぐために、プライベート通信を行っているプロセスは他のプロセスから分離され、ネットワークへのアクセスも制限される。例えば、プライベート通信を行っているユーザ A とユーザ B のプロセスからは、それぞれのユーザのその他のプロセスにはアクセスできなくなり、他のホストへのネットワーク経路も見えなくなる。

また、パーソナルネットワークは仮想ネットワークを張るユーザの組み合わせ毎に別々の暗号化を行い、通信の機密を保持する。1 人のユーザが専用の仮想ネットワークを使ってプライベート通信する場合と違い、異なるユーザ同士の通信では仮想ネットワークは 2 人のユーザに共有される。ユーザ A とユーザ B の間の仮想ネットワークとユーザ A とユーザ C の間の仮想ネットワークが異なる暗号化を行えば、ユーザ B とユーザ C はユーザ A とのそれぞれのプライベート通信を盗み見ることはできない。

4 実装

我々は直接通信できないホスト間のプライベート通信を可能にするために、パーソナルネットワークの必要な機能を Linux 上に実装した。実装した機能は、ユーザ毎に仮想ネットワークを張れるようにするための SSL のカーネルサポートと、ゲートウェイでクライアントのユーザ認証に基づいて通信を中継するプロキシサーバである。

4.1 SSL のカーネルサポート

アプリケーションに意識させずにユーザ認証や通信路の暗号化を行えるようにするために、カーネルレベルで SSL をサポートするようにした。SSL サポートにより証明書（公開鍵暗号）を用いたユーザ認証を行うことができ、様々な暗号方式で通信路を暗号化することができる。SSL を利用するにあたっては、OpenSSL [4] を Linux カーネル内で動かせるように移植した。具体的には、ファイル・ディスクリプタの代わりにカーネル内のソケットを使えるように変更した。SSL のプロトコルには変更を加えていないので、OpenSSL ライブラリを使って開発されたアプリケーションとも通信することが可能である。ただし、SSL の制約により TCP/IP 通信のみサポートしている。

アプリケーションが通信に SSL を使えるようにするために、2 種類の方法を提供している。1 つはプログラムの中で`setsockopt` システムコールでソケットに対して`SO_USE_SSL` ソケットオプションをセットする方法である。この方法はプログラムに変更を加える必要があるので、外部から proc ファイルシステムを通して、特定のリモートホストとポートの組み合わせに SSL を使うように指定する方法も用意されている。

TCP コネクションが確立した後、クライアント側は`connect` システムコール、サーバ側は`accept` システムコールの中で SSL セッションを確立するためのネゴシエーションを行う。ただし、サーバ側のコネクションの確立は`accept` システムコールとは非同期に行われるので、`accept` システムコールが呼ばれてからネゴシエーションを行うとデッドロックする可能性がある。例えば、サーバはクライアントからのデータを受け取ってから新たなコネクションに対して`accept` しようとする一方で、クライアントは`connect` が完了してからサーバの必要なデータを送ろうとしている場合にデッドロックする。サーバ側で`accept` が呼ばれないために、クライアント側の`connect` の中の SSL ネゴシエーションが完了しないためである[1]。この問題に対処するためには、サーバ側では別のカーネルスレッドを作ってネゴシエーションを行わせ、完了した後で`accept` システムコールを終了するようにすべきであるが、まだ未

[1]実際、コマンド用コネクションとデータ用コネクションを用いる ftp でこの問題が発生する。

実装である。

カーネルで SSL をサポートすることにより、ユーザ単位での SSL セッションの再利用が可能になった。従来の SSL ライブラリを用いたアプリケーションでは、アプリケーション内でしかセッションを再利用することができなかった。ユーザ単位でセッションを再利用することにより、2 回目以降のコネクションを確立する時には、証明書を用いたオーバヘッドの大きい完全な認証を行うことなく、ユーザを認証することができる。ただし、このように簡略化した認証ではセキュリティが弱まるので、一定時間毎に証明書を用いた認証をやり直す。

4.2 ゲートウェイのプロキシサーバ

ゲートウェイでホスト間のプライベート通信を中継できるように、SSL 通信のプロキシサーバを開発した。このプロキシはクライアントから SSL を使った通信のみを受けつけ、ネットワーク内部のサーバとも SSL を使って通信を行い、各プライベート通信に対して 2 つの SSL セッションを中継する。ネットワークの内部ユーザは、どのポートへのアクセスをどのホストのどのポートに転送するかという中継ルールをプロキシに登録することができる。単純な SSL プロキシとの違いは、通信相手のユーザを認証して中継を許すかどうかを決定するだけでなく、認証したユーザ情報に基づいて異なる中継ルールを適用することができる点である。ユーザ認証には SSL ネゴシエーション時にクライアントから送られてくる証明書を用いる。一方、ゲートウェイのどのポートにどのホストのどのようなサービスが割り当てられているかは、別のプロトコルを使って通信相手に伝える。

図 2 のように、プロキシで 2 つの異なる SSL セッションを中継すると、通信内容を解析できるという利点はあるものの、復号化と暗号化を行う必要があり効率が悪い。そこで、クライアントとプロキシ、プロキシとサーバの間の 2 つのセッションを、クライアントとサーバの間の 1 つのセッションで置き換えられるようにした。そのために、クライアントとプロキシの間の SSL セッションをクライアントとサーバの間に拡張するというアプローチを取った。これにより、中継のオーバヘッドを減らし、かつ、プロキシでも通信内容を解析することができる (図 3)。プロキシは、まず、クライアントとの間の

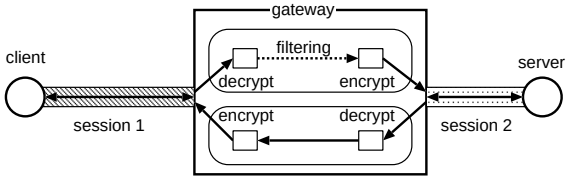

図 2: 2 つのセッションを中継するゲートウェイ

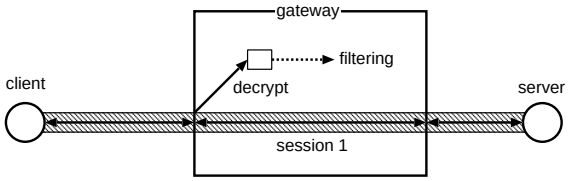

図 3: 拡張した 1 つのセッションで通信するゲートウェイ

SSL セッション情報をサーバとの間の暗号化通信路を使ってサーバに送る。SSL セッション情報として必要なのは、

- セッション ID
- 暗号方式
- 圧縮方式
- マスター鍵（共有鍵）
- サーバとクライアントで交換した 2 つの乱数
- 読み書きのシーケンス番号
- 暗号アルゴリズム依存の情報（DES の初期ベクトルなど）

である。サーバはこのセッション情報を持つ仮想的なセッションを作り、以後、そのセッションを使ってプロキシと通信を行う。一方、プロキシは、以後、クライアントとサーバの間の通信を復号化・暗号化せずにそのまま中継するので、クライアントとサーバは 1 つのセッションで通信を行うことができるようになる。また、プロキシもそのセッション情報を保持しているので、通信内容を解析することができる。

5 実験

実装上の工夫による SSL 利用のオーバヘッド軽減を調べるために、2 種類の実験を行った。実験に用いた SSL プロトコルはバージョン 3 であり、サー

表 1: セッションを確立するまでにかかる時間

	msec
セッションを再利用しない	60
セッションを再利用する	1

表 2: セッション中継の有無によるスループットの差

	Mbps
2 つのセッションを中継	3.9
1 つのセッションで通信	6.5

バとクライアントの認証とセッション鍵の交換には 1024 ビットの RSA（公開鍵暗号）、セッションの暗号化には 3DES（共有鍵暗号）、メッセージ認証には SHA1（ハッシュ関数）を用いた。実験に使用したマシンは、100Mbps のイーサネットで繋がれた PentiumIII 733MHz 3 台であった。SSL ライブラリには OpenSSL 0.9.6 を使用し、全てユーザレベルで実験を行った。

5.1 SSL セッションの再利用による性能改善

TCP コネクションを確立する度に一からセッションを作り直す場合と、セッションを再利用する場合との性能差を測定した。それぞれの場合について、`connect` システムコールを発行してから、セッションを確立するまでの時間を表 1 に示す。セッションを再利用するとはセッションの確立にかかる時間が大幅に減っていることが分かる。ウェブサーバのようにコネクションの確立と切断を頻繁に行うような場合には特に、セッションの再利用が有効である。

5.2 SSL セッションの拡張による性能改善

ゲートウェイで復号化・暗号化を行い 2 つのセッションを中継する場合と、一方のセッションを拡張してゲートウェイでは暗号化されたデータの転送のみを行う場合との性能差を測定した。それぞれの場合についてのスループットを表 2 に示す。この実験では高速なネットワークを使用したので、2 つのセッションを中継する場合には、ゲートウェイでの復号化・暗号化が大きなボトルネックになっており、セッションを 1 つにした時には 2 倍近い性能向上が見られた。実際はクライアントからゲートウェイまではもっと遅いネットワークであることが多いので、差は縮まると考えられる。

6 関連研究

IPsec [3] を使った VPN ではリモートサイトとの通信は IP レベルで暗号化され、ホストは仮想的にリモートネットワークの一員になる。そのためリモートホストにログインした時と同様に、リモートネットワークの能力を自由に使うことができる。しかし、ホスト全体がリモートネットワークに参加することになり、ユーザ単位で参加を制限するということはできない。ホストが参加する時にユーザを認証することも可能だが、一旦参加するとホストのそのユーザ以外のユーザもリモートネットワークを使えてしまう。リモートネットワークで提供されているサービスの利用に関しては、アプリケーションレベルでのユーザ認証に依存する。

PPTP [2] や L2PT [5] では IPsec とは違い、データリンク層でトンネリングを行う。従来ダイアルアップ接続などで用いられてきた PPP フレームを暗号化して IP パケットにカプセル化することにより、インターネット経由で安全にリモートネットワークに接続することができる。IPsec と同様、リモートネットワークに参加する時に PPP ネゴシエーションを行い、ユーザを認証することができるが、その後は誰でも使えるネットワークになってしまう。

一般ユーザでも利用できるアプリケーションレベルの暗号化として、ネットワークの両端で SSL プロキシを使う方法や ssh のポートフォワーディングを使う方法がある。これらは指定したポートにアプリケーションがデータを送ると、暗号化された通信路を使ってリモートホストへ転送し、目的のポートにリダイレクトする。暗号化通信路を確立する時にはユーザ認証することができるが、その後はどのユーザでも使えるため、特定のユーザだけが使えるプライベートネットワークにすることはできない。また、特定のリモートホストのポートへ転送するだけなので、リモートネットワークの他の能力を使うことはできない。つまり、ユーザはリモートホストの決められたポートにしかアクセスできない。

従来、外部からのネットワークパケットはゲートウェイで単純に内部ホストに転送されたり、NAT やプロキシによって中継されていた。この場合、ゲートウェイは誰からのアクセスでも中継してしまい、通信内容に関与できなくなるので、セキュリティ上好ましくない。プロキシはユーザ認証する場合もあるが、プロキシを使えるユーザであるかどうかを確かめるだけである。パーソナルネットワークにおけるゲートウェイは、これらの中継技術にユーザという属性を付け加える。ゲートウェイでユーザを認証し、許可されたユーザからのパケットだけを中継する。さらに、中継先も通信相手のユーザによって変更することができる。

パーソナルネットワークで行っているネットワーク環境の分割は、文献 [6] で提案されている多重通信クラスによっても行うことができる。多重通信クラスは、ネットワークトポロジの観点からネットワークを分割し、複数の仮想ネットワークを構築することを可能にする。各仮想ネットワークにリソーススペースと呼ばれる OS 内の処理モジュール群を自由に割り当てることにより、仮想ネットワーク毎に異なるネットワーク環境を実現する。ただし、多重通信クラスはトポロジ、つまり、IP アドレスなどの位置情報の違いによってネットワークを分割しており、ユーザ情報によりネットワークを分割するパーソナルネットワークとは異なる。

また、FreeBSD で実装されている`jail` システムコールを使うことによっても、ネットワーク環境の分割を行うことができる。`jail` は`chroot` システムコールの改良版であり、プロセスに対してファイルシステムを制限するだけでなく、一種の VM を作ることができる。それぞれの VM には別々の IP アドレスやドメイン名を割り当てたり、異なるネットワーク設定ファイルを使うことができ、VM 同士は干渉することができない。この VM に各ユーザや各サービスを割り当てることで、ネットワーク環境をユーザやサービスの単位で分割することができる。しかし、VM を作る度にファイルシステムのセットアップが必要になり、オーバヘッドが大きい。

7　まとめと今後の課題

本稿では仮想プライベートネットワークにユーザの概念を取り入れたパーソナルネットワークを提案した。パーソナルネットワークはユーザが専用の仮想ネットワークを使えるようにするために、ネットワーク資源、ネットワーク設定、ゲートウェイをパーソナライズする。我々はパーソナルネットワークの一部の機能を実装し、直接通信できないホスト同士のプライベート通信を可能にした。実装した機能は、ユーザ単位でのセッション再利用を可能にするカーネルレベルでの SSL サポートと、ゲートウェイのプロキシでのユーザ認証に基づく SSL セッションの中継である。

今後の課題は、まず、ネットワーク設定のパーソナライズなど、まだ実現できていない機能を実装することである。その際に、従来、管理者しか設定できなかったネットワーク設定をいかにして安全に、整合性の取れた形でユーザが設定できるようにするかが解決すべき問題である。また、同様の問題はサーバのポートのパーソナライズに関しても生じ、ユーザが自由にサービスを起動するのを許すのは問題がある場合がある。これらの問題はネットワーク環境の分割をうまく行うことによって、周りに被害が及ばないようにすることが可能かもしれない。

暗号化通信に関して、現在は SSL を使い、TCP/IP ベースのネットワークを考えている。TCP 層は IP 層よりソケット層やプロセスに近いため、ユーザを扱いやすいという利点がある一方、IP ベースのネットワーク上できめ細かいことをするのにはあまり向いていない。今後は IP ベースでパーソナルネットワークを構築していくか、TCP と IP をうまく使い分けていくかを十分に検討する必要がある。

参考文献

[1] Freier, A. O., Karlton, P. and Kocher, P.: The SSL Protocol Version 3.0, Internet Draft (1996).

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

[3] Kent, S. and Atkinson, R.: Security Architecture for the Internet Protocol, RFC 2401 (1998).

[4] The OpenSSL Project, : OpenSSL, `http://www.openssl.org/`.

[5] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G. and Palter, B.: Layer 2 Tunneling Protocol “L2TP”, RFC 2661 (1999).

[6] 廣津登志夫, 福田健介, 明石修, 佐藤孝治, 山崎憲一, 菅原俊治：仮想データリンクを用いた多重通信クラスに関する一考察, 第 3 回インターネットテクノロジーワークショップ (WIT2000) (2000).