VPN 環境の構築とその評価

2000MT010 浮池 崇　2000MT072 小川 貴史　2000MT103 山田 真弓

指導教員　石崎 文雄

1. はじめに

インターネットは現在では社会基盤の重要な位置を占めている。ネットワーク関連の費用は大きな負担になっており、リモートアクセスへのニーズは高まっている。地理的に離れた複数の拠点間に安全に安価にプライベートネットワークを構築するための技術として VPN が現在注目されている。これはパブリックネットワーク上に仮想的に構築されたプライベートネットワークである。VPN を使うとインターネットを経由しているにもかかわらず、まるで同じネットワーク上にいるかのような利便性が得られる。VPN はインターネットの利便性を生かしながら、盗聴･改ざんといったセキュリティリスクを抑えるために暗号化技術を使用している。

VPN を実装する方法の一つとして、OpenVPN[9]の利用が挙げられる。OpenVPN はインターネット上にトンネルを作り、安全にネットワーク通信することができる、使いやすく強健な VPN デーモンである。OpenVPN は SSL/TLS プロトコルを使用して、SSL/TLS プロトコルはセッション層で実装されている技術であるため、NAT やファイアウォールを通過することができ、動的な IP アドレスもカバーできる。

本研究ではこの OpenVPN に着目し、OpenVPN による VPN 環境を構築して Qcheck のスループットテストを用いて通信速度について性能評価をする。PC―ルーター―PC のプライベートなネットワーク環境で OpenVPN サーバとクライアント間で性能評価を行う。OpenVPN の利用の有無とデータサイズの変化によるスループットを調べ、ファイアウォールの通過による通信速度の変化も調べる。

2. VPN について[1][2][3]

2.1. VPN(Virtual Private Network)の定義

VPN とは「公衆回線を介して、私的なネットワークどうし、および私的なネットワークと端末機器が相互通信するために、仮想的に構築される私的なネットワーク」と定義する。ここで定義した公衆回線とは、一般電話網やインターネットに代表される公衆回線網を指す。この種のネットワークは個人や特定の組織が管理しておらず、様々なネットワークの集合体であることが特徴だ。この種のネットワークを総じてインターネットと呼ぶ。一方、私的なネットワークとは、社内 LAN に代表される私有回線網を指す。仮想的にとは、物理的にではなく、インターネット上で必要に応じて仮想的に構築されることを意味する。

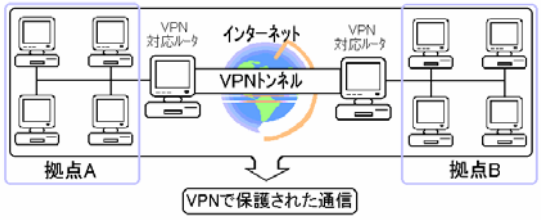

図 1：LAN 間接続 VPN

2.2. VPN の歴史

VPN は最初は遠隔地の会社を内線電話でつなぐという考えから始まった。そのため VPN という言葉はもとは内線電話網用 PBX(Private Branch Exchange)の相互接続を表すために使用されていた。そして、1995 年に閉域電話網上に構築された仮想ネットワークとして始まった。1996 年には米国の PSINet や UUNET などの ISP でサービスが開始され、1997 年になって日本で IIJ などでインターネットを利用した VPN サービスが提供された。1999 年頃になると VPN 機器や VPN 対応ソフトウェアが販売され、多くの企業やユーザに浸透し始めた。

2.3. 種類

VPN はインターネット VPN と IP-VPN の 2 種に大きく分類される。

･インターネットVPN

パブリックなインターネットを利用しつつ専用線のように閉域性を持たせた通信を確立する技術を指す。インターネットを通過するため遅延時間の発生や帯域の保証がないなどのデメリットもあるが、距離に依存せず低コストで 1 対多の接続が可能で拠点変更に対する柔軟性も備えているなどインターネットの持つメリットを享受することもできる。通信要件が、遅延や耐障害性の保証が無いベストエフォート型で要件が満たされるケースならば大きなコストメリットが得られる。

･IP-VPN

MPLS や VR 技術を用いてキャリアや ISP が独自にサービスを展開する IP-VPN 網を利用した VPN を指す。機器やコストの関係からエンドユーザが独自にこの方式で VPN を構築することは少ないと考えている。キャリアによっては回

線品質や遅延について SLA を保証する場合もあるので信頼性を求める通信にも使える。通信事業者の閉鎖的な IP-VPN 網を介して通信を行っているため、特に暗号化や認証といった手法は用いずに、必要なルータを経由させないことでセキュリティを高めていることが特徴と言える。

2.4. VPN の特徴

VPN と専用線を使用したネットワークを4つの面から比較したものを以下に述べる。

まず、コストの面で比較すると、専用線の回線使用料は非常に高額である。初期投資は、通信量の最大値を基準にして回線を敷設する必要があるだけでなく、物理的距離や接続するネットワークの数にともないそのコストは増大する。VPN 導入により、初期投資は小規模なＶＰＮを構築する場合、フリーのソフトウェアを利用することができ、通信コストはインターネットに接続するためのコストだけであり、高額な専用線を必要としないので大幅なコスト削減になる。特に国内での長距離通信や国際通信に専用線を利用している企業などは、インターネットを利用した VPN に置き換えることでコスト削減が可能となる。

次にセキュリティ面については、専用線は完全に閉鎖的な回線であるため、途中経路で第三者によるデータの盗聴や改ざんの脅威が排除される。VPN ではインターネットを介して通信を行うので、セキュリティは最も重要とされる。VPN は暗号化、認証といった技術を組み合わせることで、高度なセキュリティ対策を実現できる。データの盗聴に対しては、共通鍵暗号方式による暗号化、改ざんに対しては、防ぐことは難しいが、電子署名を利用することで、改ざんの痕跡を検知することが出来る。

パフォーマンスの面では、専用線はいったん敷設してしまえば、一定の帯域を利用することが完全に保証されているため、一貫したパフォーマンスを発揮することができ、高い信頼性もあるので、最も優れていると言える。VPN はインターネットを介した通信という性質上、通信速度や通信帯域に保証はない。そのため、帯域保証が要求されるネットワークが必要な場合、VPN でそれを解決するのは難しい。また、暗号化や鍵生成などのたくさんの処理をする必要があるため、ホスト上の負荷が増大することになる。よって、VPN が普及するに従い、インターネット上に大量のトラフィックを発生させることにもなる恐れがある。

柔軟性や拡張性の面について、専用線は、基本的に 2 点間を物理的に接続しているため、決められた場所からしかサービスを利用できず、利用環境の柔軟性に欠けている。さらに、新規に通信相手を追加する場合、新しい回線を追加しなければならず、拡張性にも乏しい。VPN は場所的な制約はなく、インターネットに接続できさえすればどこからでも利用できる。ランダムな遠隔地からの利用だけでなく、ネットワーク構成を再構築する必要があるときなどに、柔軟に対応できる。また、同時接続数は VPN ハードウェアやソフトウェアの仕様に依存するが、一般的に豊富な接続数を処理する機能を備えており、拡張性にも富んでいる。

表1：用件別比較

	コスト	セキュリティ	パフォーマンス	柔軟性
専用線	×	◎	◎	×
VPN	◎	×	×	◎

2.5. VPN で使用される技術

VPN を実現するための核となる技術は大きく分けて3つある。

1 つは、トンネリングと呼ばれる仮想的な面で役割を果たす技術である。もとのパケットをほかの配送ヘッダでカプセル化することにより、異なるプロトコルやアドレス体系のネットワーク間で通信を行うことが可能になる。ユーザは、データを送る側も受け取る側も、トンネリングされていることを意識することなく、使用中のシステムを変更せずそのまま利用できる。プライベートアドレスやマルチプロトコル通信を実現する VPN の最も重要な機能である。

トンネルの主な役割は、プライベートなアドレスの隠蔽、IP 以外のペイロードの伝送、データ伝送の容易化、組み込みセキュリティ機能である。トンネリングは、プライベートなパケットとそのアドレスを公開されたアドレスを持つパケットの内部に隠蔽し、そのプライベートなパケットを公衆ネットワークを介して伝送できるようにする。例えば、組織が正式に登録されていない IP アドレスをプライベートネットワークで使用している場合、トンネリングを使用すると、IP アドレスの割り当て方法を変更しなくても、公共のネットワーク上での通信機能を利用することができる。

2つめは、通信パケットを暗号化する機能である。トンネリングだけではデータの内容は見えてしまうので機密性を保証し、トンネリングされたパケットの盗聴や改ざんなどを防止するために、パケットを暗号化して伝送するための仕組みが必要になる。暗号化はセキュリティ対策技術のひとつとして、機密性の保証、正当性の検証、完全性の検証の3つの目的で利用されている。

3つめは認証である。認証とは送信（受信）者の正当性と送信されるデータの完全性を証明することを指す。

2.6. VPN 構築方法

VPN を構築するには、ファイアウォールやルータ、アクセス・サーバーなどに組み込まれた VPN 機能を利用するほか、VPN 専用装置も各種登場している。ファイアウォール、暗号化、ユーザ認証などの機能をパッケージ化したインターネット VPN 構築ツールも数多く提供されている。このように VPN 製品やソリューションは数多くあり、何百という数の製品が市場に出ている。IPSec に加え、これらのトンネリング

プロトコルをサポートするルータやアクセス･サーバーは多く、VPN を構築しやすい環境が整っている。また、ルータや LAN スイッチ内部のハードウェア(ASIC)で暗号処理を行ない、高速スループットを可能にする VPN 対応のネットワーク機器もあり、VPN はさまざまな通信形態に対応することができる。

最良のソリューションを得るには、それぞれのビジネス形態に適したソリューションを適切に組み合わせることである。

3. VPN でよく使用されるプロトコル

第 3 層のトンネリングは、企業のイントラネットサイト間の接続にも使用されることもあるように、IP ネットワーク内に VPN を構築する上でモットの有益なトンネリングだ。第 3 層のパケットをカプセル化すれば、アドレス指定やプロトコルの違いを覆い隠してくれるほか、複数のイントラネットサイトを結合することもできる。

トンネリングには、第 2 層と第 3 層にまたがるものもある。これは、第 2 層と第 3 層のヘッダにラベルが付加されることからラベルスイッチング方式と呼ばれている。

トンネリングプロトコルのなかで、成熟度、長い実用期間、機能、および柔軟性の面から全体的に最も構築に成功する可能性が高いと思われるもので代表的な IPSec、PPTP、L2TP と、実際に構築した OpenVPN で使用されている SSL を取り上げる。TLS については 4. 2 章で詳しく取り上げる。

3.1. IPSec

IPSec は、IP パケットの完全性と機密性を保証する機能を持っている。これらの機能を提供するための方法として、IPSec は 3 つの基本的な要素で構成されており、そのすべてによって IPSec は VPN プロトコルとして有効になっている。それらの 3 つの要素は、IP 層での認証（ユーザレベルではなくパケットレベル）、暗号化、および鍵管理である。

IPSec の特徴は、認証と暗号化のアルゴリズムや鍵管理の仕組みを IPSec プロトコルから切り離していることである。IPSec で通信を行うホスト同士は通信にさきだって、何らかの方法で認証･暗号化のアルゴリズムや使用する鍵を決定してその情報を共有する。この関係を SA(Security Association)という。

3.2. PPTP[4][5]

PPTP(Point to Point Tunneling Protocol)は、マイクロソフト(Microsoft)社がアセンド(Ascend)社、3Com 社などと共に設計した第 2 層のトンネリングプロトコルである。元々は, サーバに対してクライアントがリモート接続する際, 暗号化で安全な通信を行うために考え出されたものだ。IPSec と同様に LAN 間を結ぶためにも利用されることが多い。

このプロトコルはトンネリングを行う部分だけのものなのでその他の認証機能や暗号化機能は PAP（Password Authentication Protocol）、CHAP (Challenge Handshake Authentication Protocol)、MS－CHAP (Microsoft CHAP)など他のプロトコルと組み合わせて利用することになる。

3.3. L2TP

L2TP は PPTP と L2F を合わせたものである。L2TP はダイヤルアップユーザに対してプライベート網へのアクセスを行わせる。

例で、L2TP を使用する場合、自宅からインターネットにアクセスしたいユーザはパソコンの PPP 接続ツールでプロバイダ網に入って、そこからインターネットに抜ける形になる。認証サーバで認証されたら、そこから契約プロバイダのゲートウェイまで、L2TP のトンネルができあがり、自宅の電話回線経由で会社のネットワークに入ることとなる。

3.4. SSL[6][7][8]

SSL はインターネットで広く使われている暗号化プロトコルで、安全に盗聴されることなくデータを通信する暗号化の方式である。SSL-VPN は Microsoft Internet Explorer や Netscape Navigator の標準機能として提供されている SSL（Secure Sockets Layer）を利用して、VPN を実現する技術である。TCP/IP の上位層に位置し、TCP/IP 通信が確立されている場所であれば、簡単に暗号化通信を実現できるのが特徴である。特に、HTTP などの特定プロトコルのみの通過を許可するファイアウォールや NAT を介した通信などで威力を発揮する。この点が、ファイアウォールや NAT を透過しての通信が難しい IPSec と異なる。

4. OpenVPN について

4.1. OpenVPN

OpenVPN について説明する。[9]

OpenVPN はインターネット上にトンネルを作り、安全にネットワーク通信することができる、使いやすく強健な VPN デーモンである。

OpenVPN と IPSecVPN の違いはまずユーザスペースデーモンとして動作するので、IPSec や IKE ではなく SSL/TLS プロトコルを使用していることになる。IPSec プロトコルは、カーネルスペースの IP スタックへの修正として実装されるように設計されている。したがって、各オペレーティングシステムは、IPSec のそれ自身の独立した実装を要求する。OpenVPN と IPSecVPN の大きな違いは OpenVPN で使用される SSL/TLS プロトコルはセッション層で実装されている技術であるため、IPSecVPN とは違い NAT やファイアウォールを通過することができ、動的な IP アドレスもカバーできることである。

他の相違点として OpenVPN はセッション認証用の

OpenSSL PKI、鍵交換用の TLS プロトコル、暗号化するトンネルデータのための OpenSSL の独立した暗号 EVP インターフェース、認証するトンネルデータ用の HMAC アルゴリズム、またシングル UDP ポートの至る所のこれを多重化するためにサポートする唯一の open source VPN である。また、OpenVPN は受動的な攻撃やアクティブな攻撃に対して保護することを目指した産業の強さのセキュリティモデルを使用する。OpenVPN のセキュリティモデルは IPSec のセキュリティモデルに似ているがカーネルや IP の修正必要条件を積み重ねない。

OpenVPN はポータビリティのために構築されているので、OpenVPN が実装された Linux、Solaris、OpenBSD、FreeBSD、NetBSD、Mac OS X、Windows2000/XP 上で動作し、クロスプラットフォームのトンネルを作成できる。OpenVPN は IP 層へのカーネルモジュールでも、複雑な修正でもなくユーザスペースデーモンとして書かれているので、運用は IPSec などの他の VPN に比べて単純化されている。

OpenVPN を使用することは容易で、一般にトンネルは 1 つのコマンドで(要求された配置ファイルなし)で作成し形成することができる。

OpenVPN は速く、Redhat 7.2 で TLS ベースのセッション認証、Blowfish 暗号、トンネルデータ用の SHA1 認証、トンネリング、FTP セッション、圧縮ファイルを使用して OpenVPN は送受信 1.455Mbps のトランスファーレイトを達成している。

また、OpenVPN は柔軟であり、同じマシンへの、またはそのマシンからのトンネルの数を増やすことができる。帯域幅使用法を制限するためにトンネルにトラフィックシェーピングを適応することができる。

OpenVPN は完全にユーザスペースで動作し、TUN/TAP トンネルを使用する他のアプリケーションと平穏に共存する。

OpenVPN は、動的オンデマンドの VPN 接続を構築するスクリプトとより高いアプリケーションをうまく構築することを目指している。これらのアプリケーションについては、OpenVPN が不活発をコントロールするためにさまざまなオプションを提供する。

OpenVPN には以下の2つのトンネルモードがある。

- Routed IP Tunnels―常にブロードキャストのないポイントトゥポイント IP トラフィックを送るために良く使用される。Bridged Ethernet Tunnels よりもわずかに効率的で、構築することが簡単である。
- Bridged Ethernet Tunnels―IP と非 IP プロトコルの両方にトンネルを作るために使用できる。このトンネルは Windows ネットワークと LAN ゲームのようなブロードキャストによって通信するアプリケーションに適切で、わずかに構築するのが複雑である。

今回の実験では Routed IP Tunnels を構築し評価を行った。後に構築方法について詳しく記述する。

4.2. TLS

OpenVPN で使われる TLS について説明する。[10]

TLS プロトコルの主な目的は、通信を行う 2 つのアプリケーション間に、通信プライバシーとデータ保全機能を提供することであり、プロトコルは 2 つの層から構成されている。それは TLS レコードプロトコルと TLS ハンドシェイクプロトコルである。信頼のおける伝送プロトコル(例えば、TCP)のすぐ上、最も低いレベルに位置するのが TLS レコードプロトコルである。TLS レコードプロトコルは、より上位に位置するさまざまなプロトコルのカプセル化を行う。

TLS ハンドシェイクプロトコルでは、アプリケーションプロトコルの最初のデータを送信または受信する前に、サーバとクライアントの相互認証、暗号化アルゴリズムと暗号鍵のネゴシエーションを行うことができる。

TLS の利点の 1 つは、アプリケーションプロトコルから独立していることである。TLS プロトコルを透過的なものとした、より上位のプロトコル層を形成することができる。OpenVPN の構築

OpenVPN を実際に構築したときの環境を示す。

4.3. ネットワーク仕様

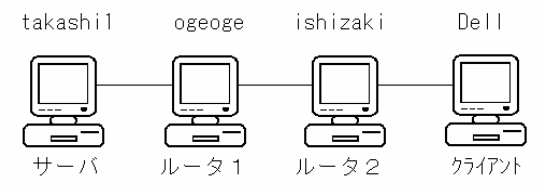

図 2:接続状態

スペック

Dell : Intel®
Pentium® 4 CPU 2.66GHz
512MB RAM
ネットワークアダプタ
Intel(R) PRO/1000 MT Network Connection

ルータ 1,2、サーバ : Genuine Intel
Inter® Celeron™ Processor
126.0MB RAM
ネットワークアダプタ
Intel 8255x-based PCI Ethernet Adapter(10/100)
I-O DATA ETX-PCI Fast Ethernet Adapter

OpenVPN ネットワークアダプタ
TAP-Win32 Adapter

表2:設定アドレス

		eth0	eth1
ogeoge	ルータ1	10.0.0.1	1.2.3.4
ishizaki	ルータ2	10.0.1.1	1.2.3.5

		eth0	tup
takashi1	RedHat9	10.0.0.2	10.1.0.1
Dell	WindowsXP	10.0.1.3	10.1.0.3

4.4. インストール

4.4.1. ライブラリの用意

まず、LZO ライブラリをインストールする。[11]から"lzo-1.08.tar.gz"をダウンロードし、インストールした。

OpenSSL はすでにインストールされていたので、行わなかった。

4.4.2. OpenVPN のインストール

OpenVPN のダウンロードページ[9]から"openvpn-1.5-beta6.tar.gz"をダウンロードし、インストールした。

Windows ではインストーラが用意されているのでその通りにやっていく。インストールが終わってネットワーク接続(ネットワークとダイヤルアップ接続)を確認すると、新しい「ローカル エリア接続 2」などが出来ているのが確認できる(番号は場合によって異なる)。この名前は、分かりやすく「OpenVPN-TAP」などに変更しておく。

4.5. OpenVPN(Linux)の設定

まず、共通鍵を作り、鍵を ftp で、takashi1 と Dell が共有するようにした。そして、root 以外のユーザが読めないように設定する。

次に、ルート設定スクリプトと設定ファイルを書く。基本的に Web ページ[4]に書かれていたものと同じだが、少し違う点がある。takashi1 の/etc/openvpn/takashi1.up に以下のスクリプトを書いた。

```
#!/bin/sh
/sbin/ifconfig $1 10.1.0.1 netmask 255.255.255.0 mtu $2
```

また、takashi1 の takashi1.conf には次のような点に気をつけて書いた。まずサーバ側なので proto tcp-server と書き、tap デバイスを使うので tun ではなく dev tap と書いた。

4.6. OpenVPN(Windows)の設定

サーバ側と同様に設定ファイルを編集し設定する。インストールパス配下に config ディレクトリが作成されているのでそこに config.ovpn ファイルを作成した。サーバ側と異なるところは以下のことを書く点である。

・remote 10.0.0.1 と書きサーバ側にリモートするようにする
・クライアント側なので proto tcp-client と書く
・dev tap の下に dev-node OpenVPN-TAP と書く
・verb 9 の下に mute 10 と書く

4.7. 接続と確認

Linux での VPN を起動する方法は、

```
#  /usr/local/sbin/openvpn -config
   /etc/openvpn/takashi1.conf
```

を実行するとトンネルが作成される。

Windows での起動方法はまず config.ovpn のあるディレクトリに移動して、

```
C:¥Program Files¥OpenVPN¥config>..¥bin¥openvpn
-config config.ovpn
```

を実行するとトンネルが作成される。

Ping で互いに接続しているか確認する。

5. 性能評価

今回は性能評価をするにあたって、トラブルシューティング向けの Qcheck[12]というソフトウェアを使用した。Qcheck はネットワークにおけるレスポンスタイム、スループット、ストリーミングパフォーマンスをテストすることができる。

実験の内容は Qcheck のスループットテストを用いてインターネット環境での通信速度と OpenVPN による VPN 環境での通信速度を比較した。

5.1. 結果

データサイズは 100kBytes から 100kBytes ずつ増やしていき 1MBytes まで評価した。なおファイアウォールはなしで実験を行った。結果は各 10 回テストを行ない、データの平均をとる。インターネット環境の場合は図3に、OpenVPN による VPN 環境の通信速度の変化は図4に示す。
縦軸に通信速度、横軸にデータサイズをとる。

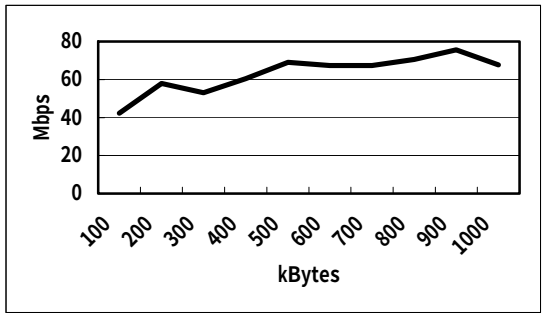

図3:インターネット環境

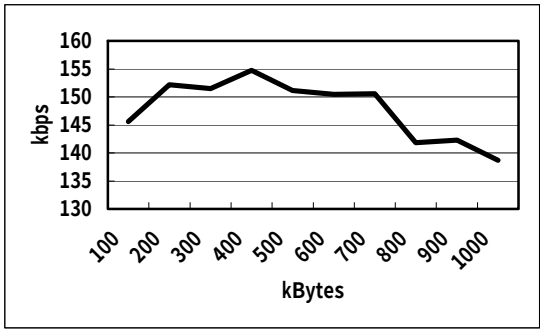

図4:OpenVPN による VPN 環境

次にファイアウォールを通過させるとどのように通信速度が変化するかを調べるため、ルータ1のファイアウォール1つを起動した場合とルータ1とルータ2の両方のファイアウォールを起動した場合を評価した。その結果を図 5 に示す。縦軸に通信速度、横軸にデータサイズをとる。

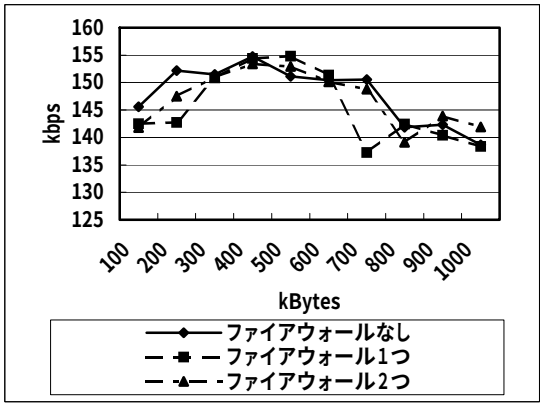

図5:ファイアウォールがある場合の通信速度変化の比較

5.2. 考察とまとめ

この実験結果からインターネット環境での通信速度は平均63.120Mbps に対して OpenVPN による VPN 環境では平均147.90kbps であった。約43分の1の通信速度になった。

図3からインターネット環境での通信では徐々に通信速度は増していき、1MBytes で少し速度が下がるぐらいである。それに対し、図4から OpenVPN による VPN 環境での通信速度の変化は 400kBytes でピークになり 800kBytes のところで通信速度はかなり下がっている。800kBytes 程度の負荷で性能が落ちてくることがわかる。

図5からファイアウォール1つの場合も2つの場合もファイアウォールがない場合と同じように変化していると感じられる。データサイズが 400kBytes 辺りでもっとも速度が速くなり、800kBytes 辺りで急激に速度が下がっている。したがってファイアウォールによる通信速度への影響はそれほどないものと考えられる。

6. おわりに

本研究では VPN 環境を容易に構築できるパッケージとして OpenVPN に注目し、OpenVPN を用いた VPN 環境を構築した。また PC―ルーター―PC のプライベートなネットワーク環境で OpenVPN サーバとクライアント間で性能評価を行った。著者の調べた限り、公式 Web ページ[9]以外には OpenVPN の通信速度を測定した資料がなく、実際評価をした結果を比較できるものがなかった。その結果、OpenVPN の公式 Web ページ[9]では、通信速度 1.455Mbps を達成したと報告があったが、実際に測定してみるとその約 10 分の1の速度までしか出なかった。

今後の課題としては、OpenVPN の特徴である NAT を通過させたときの通信速度の変化なども確認し、IPSec-VPN である FreeS/WAN も実装して OpenVPN と比較評価することが挙げられる。複数のトンネルの構築と現実的な通信環境での場合の性能評価をスループットだけでなく、クライアント数の増加や、同時アクセス時の通信速度の評価も必要と思われる。

参考文献

[1] 矢次弘志:IPsec による VPN 構築ガイド 基礎と実践,技術評論社(2003)
[2] 小早川知昭:IPsec 徹底入門,翔泳社 (2002)
[3] ブルース・パールムッター,ジョナサン・ザルコウワー :VPN 入門,ピアソンエデュケーション(2001)
[4] エンタープライズ http://www.itmedia.co.jp/help/howto/linux/vpn/
[5] IP Network Skill http://xai.nu/ipnet/stack/0073.txt
[6] VPN の仕組み http://www.netone.co.jp/doc/reference/im199808/im199808.html
[7] ITSquare 技術用語ミニ解説 http://www.sw.nec.co.jp/lecture/tech_word/sslvpn/
[8] @IT:トレンド解説 http://www.atmarkit.co.jp/fnetwork/trend/20030725/sslvpn.html
[9] OpenVPN http://openvpn.sourceforge.net/
[10] TLS http://www.ietf.org/html.charters/tls-charter.html
[11] LZO http://www.oberhumer.com/opensource/lzo/
[12] Qcheck http://www.ixiacom.com/enterprise/Qcheck.php