情報処理学会研究報告
IPSJ SIG Technical Report

Vol.2010-DPS-145 No.10
Vol.2010-GN-77 No.10
Vol.2010-EIP-50 No.10
2010/11/25

t-Room 環境に適した VPN 接続手法の選定と QoS による動的通信帯域制御手法の提案

古屋徳彦[†]　田中睦也[†]
片桐滋[†]　大崎美穂[†]

遠隔コラボレーション支援システム t-Room にモバイル計算機から接続参加する利用形態が新たに検討されている．しかし，従来の固定的な VPN 接続は，こうした新しく柔軟な t-Room の接続には適さない．また一般に，モバイル計算機は通信帯域などに制約を持つ公衆回線を経由することが多く，そうした場合，t-Room が扱う大量の視聴覚データをそのままモバイル計算機との間で送受信することは通信品質を低下させる危険がある．これらの課題の解決を目指し，本稿は，IPSec や SSL-VPN などの 4 種の VPN プロトコルの適性を改めて比較調査し，さらに QoS を用いて動的に通信帯域の制御を行う方法の有用性の検討を行うものである．

Selection of VPN Protocols Suited for t-Room Connection and Development of QoS-based Dynamic Communication Bandwidth Control Method

Norihiko Furuya[†], Tokiya Tanaka[†],
Shigeru Katagiri[†], and Miho Ohsaki[†]

A new utility form has been investigated that connects a remote collaboration support system, “t-Room”, and a mobile computer. However, a conventional VPN connection procedure, which basically matches to a static network environment, is not necessarily suited for such new and flexible t-Room connection. Also, a mobile computer usually uses a public line of which attributes like communication bandwidth are often limited, and therefore it is undesirable to transmit a large amount of t-Room’s multimedia data between a small mobile computer and t-Room. To alleviate these problems, we re-visit four VPN protocols such as IPSec and SSL-VPN, find the best selection among them, and investigate the usefulness of a new QoS-based dynamic communication bandwidth control method.

1. はじめに

遠隔コラボレーションの支援を目指して，未来の電話“t-Room”に関する研究が盛んに行われている [1) 2) 3)]．t-Room は，複数のディスプレイやカメラ，スピーカ，マイク等のマルチメディア機器とそれらを制御する複数のサーバ計算機から構成されるコンピュータ制御のマルチメディア空間である．

t-Room どうしはインターネットを経由して接続され，サーバ間で視聴覚メディアデータの送受信が行われる．接続におけるセキュリティを確保し，多数のネットワークノード管理を効率的に行うために，その接続は VPN（Virtual Private Network）上で行われる．しかし，VPN には IPSec（IP Security protocol）や PPTP（Point-to-Point Tunneling Protocol）などの種々のプロトコルがあり，t-Room 接続に望ましいプロトコル選択の評価は必ずしも十分に行われてこなかった．また最近では，t-Room 開発当初に提案された大型で部屋形状を持つ t-Room のみならず，1 台のモバイル型計算機上で運用されるモバイル型 t-Room も登場し [4)]，VPN 接続の運用に求められる機能も変化しつつある．

こうした状況を踏まえ，本稿は，従来の部屋型 t-Room どうしの接続のみならず，部屋型 t-Room とモバイル型 t-Room の接続も包含した，より柔軟な t-Room 接続に適した VPN 構築法の確立を目指し，プロトコルの調査を行うものである．本稿ではまず，特にネットワーク接続の視点から t-Room の概要を紹介し，続いて t-Room 接続に用いる VPN プロトコルの評価結果を紹介する．評価対象は，部屋型 t-Room どうしの接続に用いてきた IPSec に加え，新たに候補として考える PPTP 及び L2TP（Layer 2 Tunneling Protocol），SSL-VPN（Secure Sockets Layer Virtual Private Network）の 4 種類である．IPSec から L2TP までは，そのクライアント機能が t-Room が走る OS である Microsoft Windows OS 上に標準で実装されており，SSL-VPN は，そのセキュリティや柔軟性の高さで近年注目を集めるようになったプロトコルである．多数のマルチメディア機器を擁する t-Room の通信では，本質的に大量のメディアデータのパケットがやりとりされる．従って，1 台のモバイル型計算機で大量のデータ処理を行わなければならないモバイル型 t-Room との通信には通信遅延やパケットロスなどの問題が生じやすい．こうした障害を回避するための対策として，本稿の最終部では，QoS（Quality of Service）による動的な通信帯域制御法の適用を試み，その効果の調査結果も報告する．

2. 柔軟な接続形態に適応する t-Room

2.1 t-Room の概要

t-Roomとは，モノリスと呼ばれる壁面から構成される，遠隔コラボレーション支援

[†] 同志社大学大学院　工学研究科
Graduate School of Engineering, Doshisha University

Vol.2010-DPS-145 No.10
Vol.2010-GN-77 No.10
Vol.2010-EIP-50 No.10
2010/11/25

用のマルチメディア部屋型空間である．図1にその実装例を示す．各モノリスには大型ディスプレイとカメラ，スピーカ，マイクのメディア機器が組み込まれており，それらの機器は対応するサーバ（例えば，ディスプレイを制御するディスプレイサーバ）によって制御される．図2には，部屋型t-Room内のサーバ群の接続の様子と部屋型t-Roomどうしのネットワーク接続の様子を図解している．多数のメディア機器の利用に伴い多数のサーバが用いられ（現状では，1台のメディア機器に対し1台のサーバが割り当てられている），t-Room内ネットワークはルータで外部と仕切られたLANを構成している．

t-Room は，視聴覚メディアに関して対称な空間，すなわち異なる t-Room にいる利用者どうしが同じ視聴覚メディアデータを共有する空間を作り，その空間どうしを通信接続することによって，その利用者にあたかも同一の部屋にいるような同室感を提供することを目指す．

図 1　t-Room の実現例.
Figure 1　An example of t-Room installation.

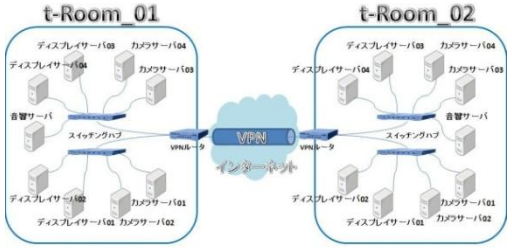

図 2　部屋型 t-Room どうしを接続するネットワーク構成.
Figure 2　Network configuration for connecting two t-Room LANs.

2.2　部屋型 t-Room とモバイル t-Room の接続

同室感の提供を目指すとき，基本的には，接続すべきt-Roomどうしは同一の空間構成を持つ部屋であることが望ましい．しかし，例えば電話という従来型の単一メディア型のコラボレーション支援システムに固定電話や携帯電話，電話会議システムなどの様々な形態があるように，t-Roomにも大型の部屋型のみならず携帯可能な形態が期待されるのは自然である．こうした小型のモバイル型t-Roomは，複数人が集まる部屋型t-Roomや，そうした部屋型t-Roomどうしを接続した仮想的な空間に，さらに個別に一人一人をつなぐことを可能にし，コラボレーションの枠を拡大する．

図 3 に，部屋型 t-Room とモバイル型 t-Room をネットワーク接続する様子を図解する．モバイル型は，機器の制約のために，多数の映像データや音データを 1 台で送受信し，処理を行う必要があることがわかる．

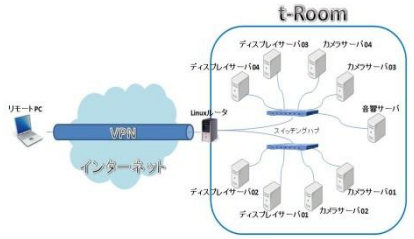

図 3　部屋型 t-Room とモバイル t-Room を接続するネットワーク構成.
Figure 3　Network configuration for connecting a room-type t-Room and a mobile t-Room.

3.　VPN プロトコル選択の必要性

基本的に部屋型 t-Room では，処理能力が高い計算機によってサーバが構成され，さらにそれら全てのサーバは 1 つの LAN 内で接続されることを前提としてきた．従って，そうした接続における VPN プロトコルとしては，セキュリティ確保に適し LAN どうしを VPN 接続することに優れた IPSec が選択されてきた．しかし IPSec は，これらの長所を持つ反面，接続手順がやや複雑であり，さらに，接続の柔軟性が低いなどの短所も持つ．接続数やアドレス等が予め固定的に設定されていることが多い部屋型 t-Room どうしの接続では深刻化しなかったこれらの IPSec の弱点は，アドレス等が動的に変化するモバイル型 t-Room 用のリモート PC を VPN 接続する場合には顕在化する可能性が大きい．従って，柔軟な接続を視野に入れたこれからの t-Room 接続を考える場合，これまで標準的に用いられてきた IPSec が最良の選択である保証はない．

上記の問題意識に基づき，本研究では，IPsec に加え，PPTP と L2TP，SSL-VPN を対象として，モバイル型 t-Room の VPN 接続における有用性の比較評価を行うこととした．

Vol.2010-DPS-145 No.10
Vol.2010-GN-77 No.10
Vol.2010-EIP-50 No.10
2010/11/25

4. VPN プロトコル評価実験

4.1 概要

図 3 に示す部屋型 t-Room とモバイル型 t-Room を接続する環境において，VPN ルータ上に IPsec と PPTP，L2TP，SSL-VPN を実装し，その 2 つの t-Room の通信において実現される通信帯域の測定や，複数の利用者による VPN 接続作業の難易度調査を行った．それぞれのプロトコルを採用した場合の，達成できる通信品質と使い勝手に関する目安を得ることを目指す実験である．

4.2 実験システムの構成

本実験で利用する部屋型t-Roomとモバイル型t-Roomについて説明する．

部屋型t-Roomは図1に示したものである．本t-Roomは4面のモノリスから構成されている（図1左にはそのうちの2面部分が，図1右には全体の様子が表示されている．）．映像データに関しては，モノリスの数に合わせて，それぞれ4台のディスプレイサーバとカメラサーバが用いられる．一方，音データの通信用には，便宜的に1台ずつのマイクとスピーカのみを用い，その双方を1台の音響サーバで制御する形態とした．こうして，1部屋のt-Roomを運用するために合計9台のサーバマシンを稼動させる構成とした．ディスプレイサーバとカメラサーバ，音響サーバのハードウェア仕様を表1に要約する．

部屋型t-Roomに用いられる複数のサーバ群は全てLAN内に接続し，部屋型t-RoomどうしはVPN接続をする．従って，このLANと外部ネットワークとの間にはVPNを制御するVPNルータを設ける必要がある．本研究では，操作性を考慮し，Linux上で稼動するLinuxルータを構築し，実験に利用した．

モバイル型t-Roomは，カメラとディスプレイ，マイク，スピーカを搭載したノートPC（以下，リモートPCと呼ぶ）である．その諸元は表2に示す通りである．

部屋型 t-Room どうしの接続は LAN 内で，モバイル型 t-Room と部屋型 t-Room の接続はインターネット経由で行った．モバイル型 t-Room が走るリモート PC は商用の光通信回線上に張られた VPN を通して部屋型 t-Room に接続される．ここで用いる VPN ルータのハードウェア仕様を表 2 に示す．また t-Room を稼動させるための OS などのソフトウェア環境を表 3 に，本実験で使用する商用光回線の仕様を表 4 に示す．

表 1　ディスプレイサーバ，カメラサーバ，音響サーバのハードウェア仕様．
Table 1　Hardware specification of display server, camera server, and sound server.

ディスプレイサーバ、カメラサーバ		音響サーバ	
機種	Dell XPS720	機種	Dell XPS720
CPU	Intel® Core2 Duo 2.66GHz	CPU	Intel® Core2 Duo 2.66GHz
Memory	2.00GB RAM	Memory	2.00GB RAM
OS	WindowsXP Professional(SP2)	OS	FedoraCore8(Kernel 2.6.26.8-56.fc8)

表 2　リモート PC，Linux ルータのハードウェア仕様．
Table 2　Hardware specification of remote PC and Linux router.

リモートPC		Linuxルータ	
機種	Sony VGN-TT90NS	機種	Dell PowerEdge SC440
CPU	Intel® Core2 Duo 1.20GHz	CPU	Intel® Celeron 2.80GHz
Memory	3.00GB RAM	Memory	3.50GB RAM
OS	WindowsVista Business (SP2)	OS	Ubuntu10.04LTS(Kernel 2.6.32.11+drm33.2)

表 3　t-Room ソフトウェア環境．
Table 3　Software environment for t-Room.

t-Room動作環境
・Windows XP(SP2,SP3) Windows Vista(SP2) ・Ruby 1.86 ・FXRuby 1.6.16

表 4　商用光回線の仕様．
Table 4　Specification of employed FTTH.

契約回線名	スペック
OCN光「フレッツ」ファミリータイプ	下り最大100Mbps 上り最大100Mbps

4.3 ルータの実装

各 VPN プロトコルを実装する準備として Linux 上にルータ機能を実装した． Linux 計算機の NIC（Network Interface Card）を拡張し，一方を外部ネットワーク（WAN）へ，もう一方を内部ネットワーク（LAN）へ接続するデバイスとして使用した．使用した NIC はいずれも 100 Base のものである．

外部ネットワークへの接続には PPPoE 接続を使用する．ここではより高速な通信を行うためにカーネル空間で動作するカーネルモード PPPoE を採用した．続いてルーティングテーブルの設定を行う．デフォルトのルーティングテーブルを削除し，PPPoE 接続によって新たに生まれたインタフェース ppp0 をデフォルトルートに設定する．これらのルールを記述したスクリプトファイルを作成する．また，パケットフィルタリングには Linux カーネル 2.4 以降から実装された iptables を使用しスクリプトファイルを作成する．VPN プロトコルで利用するポート解放や必要となるカーネルパラメータの有効化等の設定も iptables を用いてスクリプトに記述する．最後に DDNS（Dynamic DNS）の設定を行う．本実験環境で用いる契約回線は動的グローバル IP アドレスを使用するため，固定ホスト名を動的グローバル IP アドレスに適応させるために DDNS を用いる．

4.4 VPN サーバの実装

4.4.1 PPTP

PPTP（Point-to-Point Tunneling Protocol）は Microsoft 社によって提案された暗号通信のためのプロトコルである． PPTP 自体には認証や暗号化の機能を有していないが MS-CHAP による認証と MPPE による暗号化を組み合わせたものが Microsoft 製品には標準で搭載されている．PPTP では，PPTP トンネルを制御するための「PPTP 制御コネクションプロトコル」（1723 / TCP）と，PPP フレームを IP ネットワーク上で送信す

Vol.2010-DPS-145 No.10
Vol.2010-GN-77 No.10
Vol.2010-EIP-50 No.10
2010/11/25

るための「PPTP トンネルプロトコル」（IP プロトコル番号 47）の 2 種類のプロトコルを利用する． PPTP の実装には sourceforge が提供するオープンソース pptpd[5] を使用する．認証方式には MS-CHAP-v2 を使用し，暗号化には MPPE-128 を使用する．

4.4.2 IPsec

IPsec（IP Security protocol）は VPN を構築するための標準プロトコルである．カプセル化の ESP（Encapsulating Security Payload），鍵交換の IKE（Internet Key Exchange）等のプロトコルから構成される．

ESP（IP プロトコル番号 50）は接続先認証，メッセージ認証，リプレイアタックの阻止，データの暗号化機能等を提供するプロトコルであり，その配送方法にはトランスポートモードとトンネルモードの 2 種類のモードがある．トランスポートモードは元の IP データグラムのデータ部だけを認証・暗号化し，トンネルモードは元の IP データグラム全体を認証・暗号化する．本稿における IPsec を用いた VPN 接続では公衆回線上で VPN 接続することを想定しているためデータ全体を暗号化するトンネルモードを使用する．

また IPsec と後述する L2TP over IPsec では NAT/NAPT を使用するネットワーク環境において，IPsec の拡張技術である NAT-Traversal を利用することで拠点間の通信を確立する．

実装には FreeS/WAN プロジェクトより派生し Xelerance 社によってサポートされている openswan[6] を使用する．ソフトウェアは GPL（GNU General Public License）よりリリースされている．今回は設定の簡略化のため事前共有鍵を利用し接続先認証を行う．IKE/ESP ともに暗号化には 3DES，メッセージ認証には SHA-1 を使用し，その他のパラメータは WindowsOS デフォルト値に統一する．

4.4.3 L2TP

L2TP（Layer 2 Tunneling Protocol）は Microsoft 社等が推進してきた PPTP と CiscoSystems 社の L2F を統合し IETF が標準化したプロトコルである．前述した PPTP では制御用のプロトコルとデータ用のプロトコルにはそれぞれ異なるプロトコルを使用する．一方 L2TP では「L2TP 制御メッセージ」と「L2TP データメッセージ」の両方に UDP1701 番ポートを使用する．しかし，L2TP にはセキュリティを確保する機能が存在しないため IPsec と組み合わせる L2TP over IPsec が一般的となっている．

まず L2TP 制御メッセージを使用して L2TP トンネルを確立する．この上で，L2TP トンネル内で IP パケットを送信する場合，送信する IP パケットの直前に PPP ヘッダを付加し，PPP フレームを作成する．

この PPP フレームに，相手の VPN 機器まで運ぶためのトンネル IP ヘッダ，UDP ヘッダ，L2TP ヘッダを加えることで，PPP フレームでカプセル化した L2TP パケットを作成する．さらに L2TP over IPsec では，この作成した L2TP パケットに対しトランスポートモードの ESP が適用され，アクセス拠点まで暗号化した状態で届けられる [7]．

本研究でも L2TP と IPsec を組み合わせた L2TP over IPsec を使用する．L2TP トンネリングを実装するために openswan 同様 Xelerance 社でサポートされている xl2tpd[8] を使用する．xl2tpd は openswan で実装されたレイヤー3 で動作する IPsec とレイヤー2 で動作する L2TP とを組み合わせて L2TP over IPsec を実現する．L2TP over IPsec ではレイヤー2 でトンネル IP ヘッダが加わるためトランスポートモードを使用する．認証方式は MS-CHAP-v2 を使用し，IPsec のその他のパラメータは前述した仕様と統一する．

4.4.4 SSL-VPN（OpenVPN）

SSL-VPN はトランスポート層とアプリケーション層の間で動作する SSL（Secure Sockets Layer）技術を用いた暗号化により VPN 通信を実現する．特徴の一つとしてファイアフォールや NAT を意識せずに通信することが可能である．前述した IPsec では NAT-Traversal を利用しポート番号の書き換え問題を解決しているが SSL-VPN では SSL で暗号化するため，これらの問題を意識せず通信することが可能である [9]．

今回はオープンソースプロジェクトである OpenVPN を利用し SSL-VPN の機能を実装した．OpenVPN にはルーティングとブリッジという 2 つの通信方式がある．ルーティング方式では OpenVPN サーバとクライアントに別セグメントのネットワークが割り当てられ，ブリッジ方式ではサーバとクライアントが同一セグメントに仮想的に接続する．本実験環境ではブロードキャスト等の無駄なパケット転送を除外するためルーティング方式を採用する．ただしルーティング方式を利用する場合クライアントからサーバの LAN 内へのルーティングテーブルを新たに設定する必要がある．本実験では UDP プロトコルを使用し（TCP，UDP のどちらかを選択可能），ポートはデフォルトの 1194 を使用する．今回は設定の簡略化のため事前共有鍵による接続先認証を行う．

4.5 VPN クライアントの実装

VPN クライアントとして動作させる WindowsPC において設定を行う．WindowsOS で NAT-Traversal を使用するにはレジストリを有効にする [10]．

PPTP，IPsec，L2TP，これら 3 種の VPN プロトコルは携帯型端末として使用するリモート PC の動作環境である WindowsVista にて，標準でクライアント機能が実装されているためこちらを利用する． 特にレイヤー3 で動作する IPsec では IP セキュリティポリシーにおけるセキュリティの設定として VPN サーバとクライアント PC 間でのパケットをフィルタリングすることで IPsec を実現する．

OpenVPN を使用した VPN 接続ではアプリケーションを用いて接続を行う．今回はオープンソースアプリケーションである OpenVPN GUI for Windows[11] を使用する．

4.6 評価

4.6.1 通信帯域の比較

t-Room では画像データ，音声データを同時に通信するため多量のトラフィック（通

Vol.2010-DPS-145 No.10
Vol.2010-GN-77 No.10
Vol.2010-EIP-50 No.10
2010/11/25

信量）が発生する．そこで実装した各 VPN プロトコルにおける接続時の通信帯域と t-Room 通信時に発生するトラフィックを測定する．この二つの結果から t-Room に対する各 VPN プロトコルの適応性を検証する．

通信帯域測定には Iperf[12] を用いる．Iperf では t-Room の通信方式である TCP で動作させ，測定用ウインドウサイズを 64Kbyte（WindowsXP のデフォルト値）に設定する．Iperf の実行環境を図 4 に示す．リモート PC からインターネットを経由し，Linux ルータへ各 VPN プロトコルで VPN 接続させる．この状態で t-Room を構成する LAN 内の 1 サーバ（t-Room サーバ 1）間の通信帯域を測定する（これをグローバル接続とする）．また参考までに t-Room を構成する LAN 内のサーバ 2 台（t-Room サーバ 1 と t-Room サーバ 2）を同じく VPN 接続させ，その間の通信帯域も測定する（これをローカル接続とする）．ローカル接続時，グローバル接続時において各 5 回帯域測定を行う．この測定結果より各平均値を算出した結果を図 5 に示す． また各測定結果より標準偏差を算出し表 5 に示す．

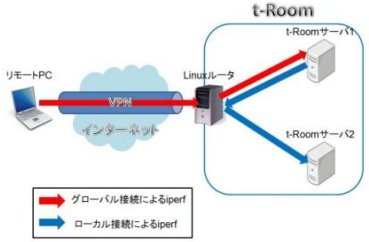

図 4　Iperf 実行環境．
Figure 4　Settings for measuring communication bandwidth.

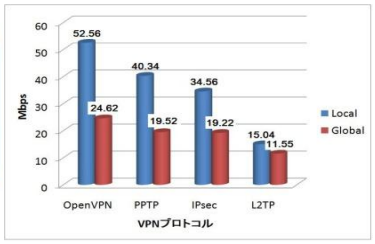

図 5　通信帯域測定結果．
Figure 5　Bandwidth measurement results.

表 5　各通信帯域の標準偏差．
Table 5　Standard deviation of measured bandwidth values.

VPNプロトコル	ローカル接続における標準偏差(Mbps)	グローバル接続における標準偏差(Mbps)
OpenVPN	0.75	0.75
PPTP	2.75	0.05
IPsec	1.15	0.1
L2TP	7.4	2.325

続いて t-Room 動作時のトラフィックを測定する．トラフィックの測定では公衆回線やリモート PC の処理能力等の制約のない状態を測定する．これの t-Room 通信時に発生するトラフィックに影響を及ぼすと予想される要素を除くために 4 面から構成される 2 組の t-Room を用意する．この t-Room を同一 LAN 上で動作させた際のトラフィックを測定する．

トラフィック測定には RRDtool のフロントエンドとして動作するグラフ作成ツール Cacti を用いる． t-Room 通信が開始されてからカメラサーバ，ディスプレイサーバ，音響サーバの Input，Output のトラフィックを測定する．30 分間 t-Room を動作させ，トラフィック平均値を測定した結果を図 6 に示す．（O は Output，I は Input を表す）

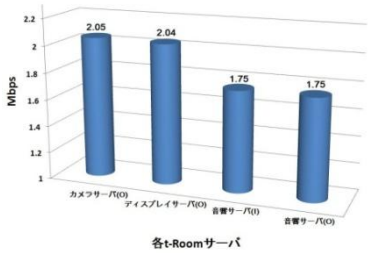

図 6　各サーバのトラフィック測定結果．
Figure 6　Results of traffic measurement.

4.6.2　利用難易度の比較

次にリモート PC を用いてユーザが各 VPN プロトコルに対して接続が完了するまでにかかる所要時間を計測する．計測は 10 名の実験参加者を得て行い，各参加者は日常的に WindowsOS を利用している情報系学科の大学生である．設定項目を記述した設定手順書を作成し，実験参加者はこれを見ながら接続設定を行う．設定手順書では 1 画面上で設定可能な項目を 1 ステップと規定し作成した．実験中，設定についての不明点や問題が発生した場合は質問を受け，それに対して説明を施す形で実験を行った．設定手順書を渡すと同時にタイマーをスタートさせ，接続が完了した時点までの時間を測定した．接続にかかる所要時間の平均値とその標準偏差，実験中に発生した質問回数，接続にかかるステップ数を表 6 に示す．

表 6　利用難易度測定結果．
Table 6　Evaluation results of difficulty in VPN connection procedures.

VPNプロトコル	平均接続時間	標準偏差	質問回数	ステップ数
IPsec	6分22秒	118.5	4	36
OpenVPN	4分14秒	61.27	0	17
L2TP	4分02秒	101.9	0	16
PPTP	2分55秒	90.42	1	8

Vol.2010-DPS-145 No.10
Vol.2010-GN-77 No.10
Vol.2010-EIP-50 No.10
2010/11/25

4.6.3 性能の総括

t-Room は TCP で通信を行っているため毎回のトラフィックのばらつきはあるもののカメラ，ディスプレイサーバでは約 2Mbps，音響サーバでは約 1.7Mbps のトラフィックで動作している．本実験環境である 4 面のモノリスで構成された t-Room に対してリモート PC から参加することを想定すると，リモート PC が 1 つのモノリスを使用する．この場合のリモート PC から見た各サーバ間の通信経路を図 7 に示す．

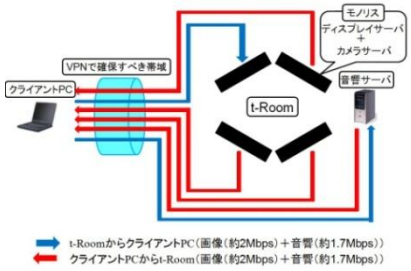

図 7　リモート PC 接続時の t-Room 通信経路．
Figure 7　Transmission path between mobile t-Room and room-type t-Room.

リモート PC 上で動作するカメラからの Output が 1 つ，各モノリスのカメラからの Input が 3 つ，マイク・スピーカの Output/Input がそれぞれ 1 つずつとなる．カメラ，ディスプレイサーバが約 2Mbps，音響サーバが約 1.7Mbps のトラフィックが必要であることから以下の式が成り立つ．

Camera Output（2Mbps） × 1 ＋ Sound Output（1.7Mbps） × 1 = Total Output（3.7Mbps）
Camera Input（2Mbps） × 3 ＋ Sound Input（1.7Mbps） × 1 = Total Input（7.7Mbps）
Total Output（3.7Mbps） ＋ Total Input（7.7Mbps） = Total Traffic（11.4Mbps）

この式からリモート PC1 台が本実験環境である 4 面のモノリスで構成された t-Room と通信するためには少なくとも合計 11.4Mbps 程度の通信帯域が必要であると予測される．本実験環境では 4 面のモノリスを使用しているが，モノリスは 6 面，8 面と拡張することも可能である．このような拡張に比例しトラフィックは増加するため，通信帯域の確保は t-Room 通信を行う上で重要なパラメータであると言える．

ローカル接続，グローバル接続共に各 VPN プロトコルおいて条件（11.4Mbps）を満たすが，t-Room 通信時に予測される通信帯域の占有率は高い．結果から算出すると各 VPN 接続を確立した状態で t-Room 通信を行うと 11.4Mbps のトラフィックが発生すると仮定すると OpenVPN（46.3%），PPTP（58.4%），IPsec（59.31%），L2TP（98.7%），それぞれが通信帯域を占有することになる．

トラフィックの標準偏差は，通信帯域を確保する際の安定性を示す数値と考えることができる．標準偏差が小さい程，安定性は高い．L2TP の標準偏差値は他の VPN プロトコルに比べ高く，このプロトコルが安定した通信帯域の確保ができていないことが分かる．

利用難易度の測定では接続にかかるステップ数に比例して接続までにかかる平均時間が計測された．IPsec では実験中の質問回数が他のプロトコルに比べ多かった．この接続にかかる所要時間は初期接続にかかる時間であり，設定を保存しておけば以後接続にかかる時間は大幅に縮まり各プロトコルの時間差はほぼなくなるが，実験参加者（10 人）の 4 人が質問しているという事実は設定事項が複雑であることを示している．

4.6.4 考察

初期接続にかかる所要時間の差が 1〜2 分程度であることからユーザビリティよりも，確保できる通信帯域やセキュリティの強度でプロトコルを選定すべきであると考える．PPTP でユーザ認証に用いる MS-CHAP-v2 では MD4 と DES が使用される．MD4 はハッシュ衝突を作成することが可能であると報告されており，また DES に関しても実際に解読できることが証明されている．また暗号化方式である MPPE の暗号鍵の交換にもこの MS-CHAP-v2 が使用されている．このように PPTP はセキュリティレベルが低いため長時間の使用や機密性の高い情報をやり取りには不向きである．

L2TP は確保できる通信帯域は低く，この上で t-Room 通信を行うことは難しい．残る OpenVPN と IPsec を比べた場合，通信帯域，利用難易度共に OpenVPN の方が適していると判断できる．さらに今回使用したクライアントアプリケーションの OpenVPN GUI for Windows ではサーバ側のパラメータ等の変更に対して conf ファイルの記述を変更するのみで対応できるため，決して複雑な設定を必要としない．以上の点から OpenVPN での t-Room 接続が現在最も推奨されるべきであると考える．

5. QoS による動的通信帯域制御

5.1 概要

3 節のトラフィックの測定結果からもわかるように t-Room 通信では多量のトラフィックを必要とする．さらに本実験環境では 4 面のモノリスを使用しているが，環境によってモノリスの拡張は可能であり，音響通信に関しても今回は 4 面のモノリスに対して 1 台の音響サーバを使用したが，位置感覚の同期を遠距離地点でとるために音響サーバの拡張も考えられる．このようなハードウェアの拡張は t-Room 全体のトラフィックの増加の要因となり得る．t-Room 通信においてトラフィックを制御・運用する手法の必要性があることは，各 VPN プロトコルに対して約 5 割以上の帯域を占有するという事実から明らかであると考えられる．この解決策として本稿では QoS を用いて

Vol.2010-DPS-145 No.10
Vol.2010-GN-77 No.10
Vol.2010-EIP-50 No.10
2010/11/25

t-Room 通信時において動的にトラフィックを制御する手法を提案する．

5.2 キューイング規則を利用する帯域制御

5.2.1 TBF（Token Bucket Filter）

本稿では Linux カーネルにおいて qdisc（Queueing Discipline）と呼ばれるキューイング規則を利用することにより通信帯域制御を行う．qdisc には優先制限用，帯域制限用，またはその両用を管理する機能が存在する．その中でも本稿では，帯域制御用 qdisc である TBF を利用する．TBF は，バッファと，バッファに格納されるトークン（仮想的なパケット群）によって帯域を制御することで，設定した通信量を超えない範囲でパケット出力を行う．各トークンはデータキューから受信データパケットを拾いバッファに格納される．そしてバッファは一定の割合（トークン速度）で満たされていく．データの処理速度とトークン速度とを監視し，これを制御することで帯域制御を実現している．ここで注意する点は qdisc における制御は受け取ったパケットを送信する際にのみ動作するということである [13) 14)]．

5.2.2 SNMP（Simple Network Management Protocol）

トラフィックを動的に制御するためには常にトラフィックの状況を知る必要がある．提案手法ではトラフィックの監視を SNMP を用いて行う．SNMP では管理者（SNMP Manager）と管理対象（SNMP Agent）の間を MIB（Management Information Base）と呼ばれる管理情報データベースの情報をやり取りすることで CPU 使用率，メモリ使用率，といったハードウェア情報を取得することが可能となる [15)]．しかし，トラフィック情報は標準 MIB である MIB-2，RFC1514 には定義されていない．よって，監視対象であるサーバに標準化拡張 MIB を定義する．この標準化拡張 MIB とは RFC で標準化のために定義された製品ベンダーに依存しない，計算機やソフトウェア管理のための MIB 定義である．これは WTSC（Williams Technology Consulting Services）社が提供する SNMP informant というフリーソフトウェアによって拡張されトラフィック情報の取得が可能になる．

5.3 実装

4 節で構築した Linux ルータ上で TBF を実行する．TBF を使用するためにはカーネルパラメータを有効にする必要があるが，今回使用するカーネル 2.6 系では標準でカーネルパラメータがモジュール化されているためカーネルの再構築は行わなかった．開発環境は ruby1.8.6 で，SNMP Manager の実装はフリーソフトウェアである SNMP ライブラリを利用する．また Linux 上で QoS 機能である TBF を実行するために tc コマンドを使用する．tc コマンドによって入力するパラメータ等を作成し実行することで帯域制御を実現する．本稿では 1 組の大型 t-Room と携帯型のリモート PC とが t-Room 通信を行う際に最も大きいトラフィックを発生させる大型 t-Room からリモート PC に向かう送信パケットに対して帯域制御を行い，提案手法の有用性を確かめる．

今回は動作フローを明確にするため，また今後の機能追加等を想定し，t-Room のトータルトラフィックが流れるよう Linux ルータと t-Room サーバ群の間にコンテントサーバとして新たにサーバを導入する．このコンテントサーバを SNMP Agent としてトラフィック情報を SNMP Manager に送信する．SNMP Manager となる監視サーバはその情報を Linux ルータに対して TCP Connection を確立し通知する．SNMP Manager と Linux ルータは Ruby プログラムのサーバ･クライアントの関係で動作している（Linux ルータがサーバ，SNMP Manager がクライアントとして動作する）．図 8 に全体体系図，図 9 に動作フローを示す．

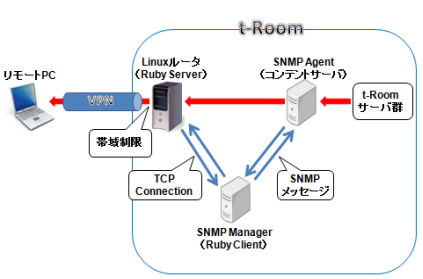

図 8 提案手法の全体体系図．
Figure 8 Overview of dynamic bandwidth control procedure.

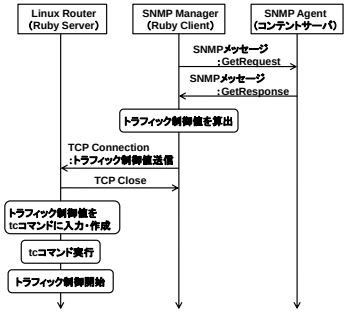

図 9 動的通信制御手法の動作フロー．
Figure 9 Process diagram of dynamic bandwidth control procedure.

まず Linux ルータ上を流れるトラフィックを監視し，その値に応じたトラフィック制御値を設定する．1 秒間隔でトラフィックを取得し 10 秒間の平均値を出す．得られた平均値に対して 10%間隔で 10～100%の間でトラフィック制御値を設定する．これによって算出された値をトラフィック制御値とする．トラフィック制御値は Ruby サーバ（Linux ルータ）に TCP コネクションを確立させた後送信される．Ruby サーバは受け取ったトラフィック制御値を tc コマンドに入力することで制御コマンドを作成する．作成された tc 制御コマンドは Linux カーネル上で上書き処理ができないため，ルールの初期化が必要である．よって tc 制御コマンドを実行する前にルールの初期化を行う．コマンド実行の応答を確認後，tc 制御コマンドを実行しトラフィック制御を開始する．

5.4 動作確認

本実験環境である 4 面のモノリスから構成される t-Room に対してリモート PC より VPN 接続（OpenVPN）を確立した状態で t-Room 通信を行う．ここでは動作確認のため音響サーバは使用せずカメラサーバとディスプレイサーバのみを使用した．t-Room 通信開始後，トラフィック制御値をトラフィック検知値の 50%に設定し，構築したシ

Vol.2010-DPS-145 No.10
Vol.2010-GN-77 No.10
Vol.2010-EIP-50 No.10
2010/11/25

ステムを実行する．その時のシステム実行前後のトラフィックの変動を図 11 に示す．図 11 の青線は Linux ルータからリモート PC に送られる t-Room パケットの Output を表し，赤線はリモート PC でこの t-Room パケットを受け取った Input を表す．

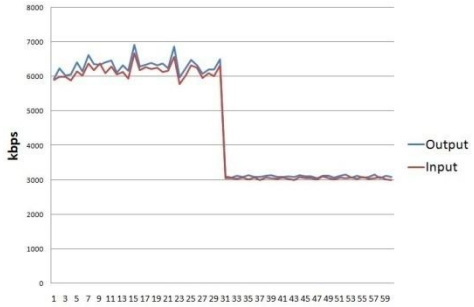

図 10　システム実行前後のトラフィックの変動．
Figure 10　Traffic change produced by dynamic bandwidth control procedure.

システム実行前のトラフィックは 6000～7000Kbps の間で変動している．システム実行直前の 10 秒のトラフィックの平均値は 6302Kbps であった．このトラフィック検知値に 50%の制御がかかり 3151Kbps のトラフィック制御値が得られ，tc コマンドにより実行される．システム実行後はトラフィック制御値に従い安定したトラフィックが得られていることがわかる．

5. まとめ

本稿前半部では通信帯域と利用難易度の観点からモバイル t-Room 接続に適した VPN プロトコルの選択を行い，総合的に OpenVPN が適していると判断した．VPN 接続時では帯域が制限されるため，今後の部屋型 t-Room の拡張，及びモバイル t-Room 接続数の増加を考慮すると現状の通信帯域では限界がある．この解決策として本稿後半部では QoS を用いた動的なトラフィック制御システムを構築した．しかし，トラフィックの変動は確認されているが，実際の t-Room 通信における影響（通信遅延，パケットロス）の調査にまでは至っていない．このため，今後は詳細な評価実験が必要であると考えられる．また，Linux カーネルで利用できるキューイング規則による通信帯域制御に関する機能は TBF の他に，多数のキューにパケットを格納し公平に処理することで特定の通信だけが帯域を占有しないよう処理を行う SFQ（Stochastic Fairness Queueing），様々な条件でクラスフルな設定が可能な CBQ（Class Based Queueing）等が存在する．これらの機能を組み込むことで詳細な QoS 制御設定を行いシステムの拡張を図ることが必要である．動的な通信帯域制御では，制御が目指すべき帯域幅の目標値を設定する必要がある．現行の手法では経験的に目標設定をすることとしているため，この目標値の決定も今後の課題となる．

謝辞

本研究を進めるにあたって様々な御助言を頂いた NTT コミュニケーション科学基礎研究所の平田圭二氏をはじめとする t-Room 研究グループの皆様に感謝申し上げます．

参考文献

1) Keiji Hirata, Yasunori Harada, Toshihiro Takada, Shigemi Aoyagi, Yoshinari Shirai, Naomi Yamashita, and Junji Yamato, The t-Room: Toward the Future Phone, NTT Technical Review, Vol.4, No.12, pp.26-33（2006）.
2) 清水康史，清田康介，片桐滋，青井真希，大崎美穂，平田圭二，原田康徳，高田敏弘，青柳滋己，梶克彦：未来の電話「t-Room」の機能強化に向けて，情報処理学会インタラクション 2009 C07 （2009. 3）.
3) 小寺晋平，片桐滋，原田康徳，平田圭三，大崎美穂，映像フィードバックに伴うエコーのキャンセリング法に関する実験的評価，信学技報，Vol.109, No.306, PRMU2009-133，pp.291-296，2009.
4) 梶克彦，平田圭二，原田康徳，白井良成：大型ビデオコミュニケーションシステムとモバイル端末の接続手法，情報処理学会研究報告（2009）
5) pptpd：http://sourceforge.net/projects/poptop/files/pptpd/pptpd-1.3.4/pptpd-1.3.4.tar.gz/download
6) openswan：http://openswan.org/
7) ASAHI INTERACTIVE, Inc. 高崎達哉 ：VPN の仕組みを探る（2005.8）
http://japan.zdnet.com/sp/feature/netsecurity1/story/0,2000056696,20086051,00.htm
8) xl2tp：http://www.xelerance.com/software/xl2tpd/
9) OpenVPN Technologies, Inc. http://openvpn.net/index.php/open-source/documentation/howto.html
10) Microsoft Corporation. TechNet オンライン　http://technet.microsoft.com/ja-jp/default.aspx
11) OpenVPN GUI for Windows 日本語版：http://www.plum-systems.co.jp/techinfo/openvpn/21.html
12) Iperf：http://sourceforge.net/projects/iperf/files/iperf-2.0.5.tar.gz/download
13) Linux Advanced Routing & Traffic Control HOWTO（2003） Bert Hubert（日本語訳 中野武雄）
http://www.linux.or.jp/JF/JFdocs/Adv-Routing-HOWTO/
14) オーム社，戸田巌：ネットワーク QoS 技術（2001）
15) CQ 出版社，斗光佳輝：SNMP ツール開発テクニック（2003）