H-02

Linux ルータと SIP サーバを用いた t-Room 間接続環境の構築

Development of t-Room Connection Environment using Linux Router and SIP Server

古屋 徳彦† 米村 裕弘† 片桐 滋† 大崎 美穂†
Norihiko Furuya Yasuhiro Yonemura Shigeru Katagiri Miho Ohsaki

1. まえがき

情報通信技術を活用した遠隔地間のコラボレーションを支援するためのシステム開発に対する関心が急速に高まっている．そうした中で，未来の電話「t-Room」が提案され[1]，筆者らもそれを用いたコラボレーション支援技術の向上の研究を行ってきた[2][3].

t-Roomは，複数のディスプレイやカメラ，マイクなど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マシン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る．接続におけるセキュリティを確保し，多数のノード管理を効率的に行うために，その接続はVPN上で行われる．しかし，VPNにはIPSecやPPTP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の評価は必ずしも十分に行われてこなかった．本研究は，これまで研究されてきた大型のt-Roomのみならず，携帯型のt-Roomなども登場することを想定し，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，t-Room の概要を紹介し，続いて VPN プロトコルの評価実験結果を紹介する．評価対象は，これまでの t-Room 接続に用いてきた IPSec と，新たに候補として考える PPTP 及び L2TP である．また，評価結果を紹介するに先立ち，評価試験を行うために構築した実験的ネットワークの詳細も紹介する．大型かつ固定的な t-Room のみならず携帯型 t-Room の接続をも想定する場合，動的に変化する VPN ノードの効率的な情報管理が必要となる．続いて本稿では，この対策として，VPN 上でこの動的な管理を行う手法として SIP（Session Initiation Protocol）サーバを用いる方法も提案する．

2. t-Room 概要

2.1 t-Room とは

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用のマルチメディア空間である．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．視聴覚メディアに関して対称な空間を作り，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．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にも大型の固定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t-Roomは，複数人が集まるt-Roomや，そうしたt-Room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，コラボレーションの枠を大いに拡大する．

こうした，より進んだt-Roomの利用形態を考えるとき，接続数やアドレス等が予め固定的に設定されているVPNノードを前提とした現行のIPSecを用いたVPNの構築は柔軟性に欠け，携帯型t-Roomの動的な接続にも適したVPNの構成法を導入する必要がある．

2.2 実験システムの構成

本実験で利用するt-Room及び携帯端末について説明する．実験で用いるt-Roomの概要を図１に示す．本t-Roomは4面のモノリスから構成されている（図1にはそのうちの2面部分が表示されている．）．各モノリスにはディスプレイとカメラが組み込まれており，それぞれを制御するためのサーバ（OSはWindows），即ちディスプレイサーバとカメラサーバも接続されている．4面から成る本実験システムでは，各4台ずつのディスプレイサーバとカメラサーバが稼動することになる．一方，音メディアの通信用にも各モノリスはマイクとスピーカを付属する設計となっていたが，本実験システムでは便宜的に，一部屋のt-Roomに1つのマイクと1つのスピーカのみを接続し，その双方を1台の音響サーバで制御する形態としている．従って，1部屋のt-Room実験システムを運用するために合計9台のサーバマシンが稼動する．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1に，ディスプレイサーバとカメラサーバのハードウェア仕様を表2に，音響サーバのハードウェア仕様を表3に掲載する．

本実験では1部屋の上記のt-Room実験システムと追加的に携帯型t-Roomを接続する利用状況を想定する．そこで用いる携帯型t-Room端末は，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（OSはWindowsであり，以下ではリモートPCと呼ぶ）である．その諸元は表4に示すとおりである．

さらに，以上の大型及び携帯型の t-Room 実験システムどうしを接続するための VPN を制御する VPN ルータとして，Linux ルータを準備する．また，接続は，大型の t-Room どうしは LAN 内で，リモート PC はインターネット経由で行うものとする．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 LAN 内にある t-Room に接続される．ここで用いる Linux ルータのハードウェア仕様を表 5 に，商用光回線の仕様を表 6 に示す．

†同志社大学大学院,
Graduate School of Engineering Doshisha University

図 1　t-Room 環境

表 1　t-Room ソフトウェア環境

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

表 2　ディスプレイサーバとカメラサーバのハードウェア仕様

ディスプレイサーバ、カメラサーバ	
機種	Dell XPS720
CPU	Intel® Core2 Duo 2.66GHz
Memory	2.00GB RAM
OS	WindowsXP Professional(SP2)

表 3　音響サーバのハードウェア仕様

音響サーバ	
機種	Dell XPS720
CPU	Intel® Core2 Duo 2.66GHz
Memory	2.00GB RAM
OS	FedoraCore8(Kernel 2.6.26.8-56.fc8)

表 4　リモート PC のハードウェア使用

リモートPC	
機種	Sony VGN-TT90NS
CPU	Intel® Core2 Duo 1.20GHz
Memory	3.00GB RAM
OS	WindowsVista Business (SP2)

表 5　Linux ルータのハードウェア仕様

Linuxルータ	
機種	Dell PowerEdge SC440
CPU	Intel® Celeron 2.80GHz
Memory	3.50GB RAM
OS	Ubuntu10.04LTS(Kernel 2.6.32.11+drm33.2)

表 6　商用光回線の使用

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

図 2 に，実験システムのネットワークとしての構成を図解する．図には 4 面のモノリスからなる 1 部屋分の t-Room のサーバ群が LAN 接続されている様子がわかる．また図中には携帯型 t-Room（リモート PC）と固定型 t-Room がインターネット経由で接続される様子も図解されている．

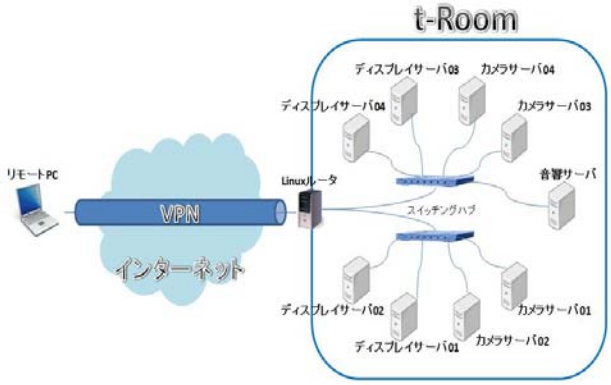

図 2　実験環境

3.　VPN プロトコルの評価

3.1　評価の方針と概要

t-Roomは基本的に，大型のディスプレイや比較的大型で計算能力も高いサーバマシンによって構成されることを前提としてきた．また1つのLAN空間に各制御サーバが集まり1組のt-Roomを形成している．これまでのVPN管理は，セキュリティ確保に適し，かつLAN空間どうしをVPN接続することに優れたIPSecを扱うことができるルータを用いて行ってきた．しかし，IPSecは，その優れた特長の反面，接続手順がやや複雑であるなどの短所もあり，大型かつ固定的なt-Roomのノード管理に加えて，（おそらくは多数に及ぶ）携帯型t-Roomに対応するノードの動的な管理を効率的に行うには最良の選択とは限らない．このVPNプロトコルの選択を行うために，本稿では，IPsec，PPTP，L2TPを取り上げ，Linuxルータ上に3種のプロトコルを実装し，同一のルータ環境（Linuxサーバ）においてそれぞれの性能や使い勝手等の比較評価を行う．

3.2　Linuxルータ概要

t-Room が VPN を必要とする理由としてセキュリティ確保と接続管理の簡便化が挙げられる．多くの機密性の高いデータが流れることが考えられる t-Room 通信はセキュリティで守られている必要がある．また t-Room は複数の PC と通信することが必要となるためネットワーク接続の管理が複雑になる．VPN はトンネル技術を用いることで，パケットをカプセル化しプライベート IP アドレスをグローバル IP アドレスで包む．このカプセル化によって遠隔地の PC があたかも同一 LAN に存在するかのように扱うことができる．このためネットワーク接続管理の簡便化は大きな役割を果たしている．

従来， t-Room では VPN ルータを用いてルータ間を IPsec で接続することで VPN を実現していた．しかしリモート PC からインターネットを経由した VPN 接続では，接続環境として VPN ルータが設置されていない状況が想定される．LAN を構成する拠点間の固定的な設定で，通信を

行う LAN 間接続 IPsec のみでは，VPN ルータが設置されていない場所からのアクセスには対応できない．またリモート PC では設定が動的に変化することが想定される点においても従来の VPN 接続方式では柔軟性に欠ける．よってリモート PC と t-Room との通信を想定した場合，これに対応した VPN 接続手法が必要となる．本稿では， t-Room 動作環境である Microsoft Windows OS 上で VPN クライアント機能が標準で実装されている IPsec，PPTP，L2TP を取り上げ，これらのサーバ機能を Linux 上（Linux ルータ）に実装する．

3.3 ルータ機能の実装

まず各 VPN プロトコルを実装する前段階として Linux 上にルータとしての機能を実装する．また Linux ディストリビューション Ubuntu では apt（Advanced Package Tool）を用いて適時必要となるパッケージの取得・管理を行う．

Linux マシンに NIC（Network Interface Card）を拡張し，インタフェース eth0 を外部ネットワーク（WAN）に接続（WAN）しグローバル IP アドレスを動的取得させ，インタフェース eth1 を内部ネットワーク（LAN）として使用し，プライベートアドレスを固定で設定する．使用する NIC のスペック表 7 に示す．

表 7　NIC スペック

eth0	Broadcom NetXtreme BCM5754（10BASE-T/100BASE-TX）
eth1	BUFFALO LGY-PCI-GT（10BASE-T/100BASE-TX/1000BASE-T）

＜ネットワーク構築＞

Linux から直接 ISP に PPPoE 接続させるためにパッケージ pppoe を用いて設定を行う．また，より高速に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．PPPoE が開始されると新たなインタフェース ppp0 が生成される．

＜ルーティングテーブル＞

以下のポリシーによりルーティングテーブルを設定する．

（1） デフォルトのルーティングテーブルを削除する
（2） 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
（3） 宛先が LAN 向けパケットは eth1 にルーティングする

なお PC が再起動時にネットワーク設定が初期化してしまうため，上記（1）～（3）のポリシーに基づきスクリプトファイルを作成し，PC 起動時に自動読み込みが行われるネットワーク設定ファイル（/etc/network/interfaces）内で実行させることで PC 起動時の自動設定を行っている．

＜iptables＞

Linux カーネル 2.4 以降から実装されたファイアウォール機能である iptables を使用しパケットフィルタリングを行う．

[基本ポリシー]

（1） 外部からの INPUT・FORWARD は基本に全て拒否
（2） 内部からの OUTPUT は基本的に全て許可
（3） IP マスカレードの設定
（4） 各 VPN プロトコルで利用するポートの解放
（5） 各 VPN プロトコルで必要となるカーネルパラメータの無効化/有効化
（6） VPN クライアント間のパケットは全て通過

VPN クライアントからのコネクションが確立するとクライアントの数だけ ppp インタフェースが追加される（ppp1, ppp2，ppp3…）．そのため VPN クライアント同士のパケットは全て通過するというポリシーを追加するためには「ppp0 を除く全ての ppp インタフェース上のパケットは通過させる」というルールを作成することに注意する．

またネットワーク設定同様，PC 再起動時に iptables は初期化されるのでスクリプトファイルを作成し起動時に自動で実行させる．

＜DDNS（Dynamic DNS）＞

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．その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．今回は外部のサービスサイト（DynDns）を使用した．グローバル IP アドレス変化時にサービスサイトに自動更新させるためにパッケージ ddclient を使用する．

3.4 VPNサーバ機能の実装

3.4.1 PPTPの実装

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 Windows98 以降の Microsoft 製品には標準でクライアント機能が実装されている．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 では，まず PPTP 制御コネクションプロトコルを使用して，PPTP トンネルを確立する．そして，PPTP トンネル内で IP パケットを送信する場合，MPPE ヘッダの後半 2byte 分を構成するプロトコルフィールドが付加された IP パケット全体を RC4 で暗号化する．

加えて，暗号化したデータに MPPE ヘッダの前半 2byte 分と PPP ヘッダを付加し，PPP フレームを作成する．さらに GREv1 ヘッダと相手の VPN 機器まで運ぶためのトンネル IP ヘッダを付加することで，アクセス先拠点まで暗号化した状態で届けることができるという仕組みである．

Linux カーネルにおいて MPPE モジュールが無効状態であればカーネルの再構築を行う．また MPPE をサポートしない場合はカーネルパッチを当て，使用可能な状態にする．

PPTP の実装にはパッケージ pptpd を使用する．認証方式には MS-CHAP-v2 を使用，暗号化には MPPE-128（RC4 での鍵長 128bit を意味する）を使用する．認証用のユーザ名，パスワード，VPN クライアントへの割り当て IP などの設定を行い，サービスを開始させる（/etc/init.d/pptpd）．

3.4.2 IPsecの実装

IPsec（IP Security protocol）は VPN を構築するための標準プロトコルである．Windows2000/XP は IPv4,IPv6 で利用可能であるが，IPv6 は ESP 暗号化に対応していない．Windows Vista/7 は IPv4,IPv6 で利用可能である．認証ヘッダの「AH（Authentication Header）」，カプセル化の「ESP（Encapsulating Security Payload）」，鍵交換の「IKE（Internet Key Exchange）」などのプロトコルから構成される．

AH は，接続先認証，メッセージ認証，リプレイアタックの阻止などの機能を提供するプロトコルであり，IP プロトコル番号 51 を使用する．AH は暗号化の機能がないため現在ではほとんど利用されることはない．ESP は AH の機能に加えてデータの暗号化機能を提供するプロトコルであり，IP プロトコル番号 50 を使用する．IKE はこれらのプロトコルで使用するアルゴリズムのネゴシエーションや鍵のセットアップを行うためのプロトコルであり UDP の 500 番ポートを使用する．

AH や ESP にはその配送方法によって「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・暗号化する．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．トランスポートはエンド・ツー・エンドで認証や暗号化を行う場合に，トンネルモードは LAN 間の通信に対して認証や暗号化を行う場合に使用するのが一般的である．

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．NAT-Traversal では ESP パケットを UDP パケットでカプセル化する．この UDP ヘッダは暗号化の対象にならないことから NAT/NAPT 機器によるポート番号の書き換え（500 番→4500 番）を可能にする．

Linux カーネル 2.4 系では IPsec の機能を利用するためにはカーネルパッチを当てる必要がある．Linux カーネル 2.6 系から IPsec のサポートが標準カーネルに取り込まれている．実装にはパッケージ openswan を使用する．

[基本設定]
- nat traversal の有効化
- トランスポートモードの使用
- 事前共有鍵の使用
- PFS（Perfect Forward Secrecy）・ReKey は使用しない
- 接続リトライ回数：3，IKELifeTime：8 時間，KeyLife：1 時間（全て Windows デフォルト値に統一する）
- IKE/ESP ともに暗号化には 3DES，メッセージ認証には SHA-1 を使用する

3.4.3 L2TPの実装

L2TP（Layer 2 Tunneling Protocol）は Microsoft 社などが推進してきた PPTP と CiscoSystems 社の L2F を統合し IETF が標準化したプロトコルで Windows2000 以降の Microsoft 製品には標準でクライアント機能が実装されている．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．

L2TP を使用する場合には，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[4].

本研究でも L2TP と IPsec を組み合わせた L2TP over IPsec を使用し以後特に断りがなければ L2TP という表現は L2TP over IPsec を意味するものとする．

L2TP トンネリングを実装するためにパッケージ xl2tpd を使用する．これと openswan で実装した IPsec を組み合わせることで L2TP over IPsec を実現している．認証方式は MS-CHAP-v2 を使用し，認証用のユーザ名，パスワード，VPN クライアントへの割り当て IP などの設定を行い，サービスを開始させる（/etc/init.d/xl2tpd）．

3.4.4 クライアント・リモートPCの設定

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必要がある．表 8 に Windows レジストリのサブキーを示す.

表 8　Windows レジストリ

Windows XP	HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Services/Ipsec
Windows Vista	HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Services/PolicyAgent

PPTP，L2TP を使用した VPN 接続では Windows「コントロールパネル」→「ネットワークと共有センター」→「接続またはネットワークのセットアップ」より接続先 IP アドレス，ユーザ名，パスワード（L2TP の場合は「オプション」より事前共有鍵の設定も必要）を設定する．

IPsec を使用した VPN 接続では「コントロールパネル」→「管理ツール」→「ローカルセキュリティ ポリシー」でセキュリティの設定として VPN ルータとクライアント 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．「IP セキュリティポリシー（ローカル コンピュータ）」にセキュリティポリシーを追加し有効化する．

3.5 性能評価

3.5.1 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 を用いる．iperf では t-Room の通信方式である TCP で動作させ，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 3 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それぞれ 5 回測定を行う．この測定結果より各平均値を算出した結果を図 4 に示す． また各測定結果より標準偏差を算出し表 9 に示す．

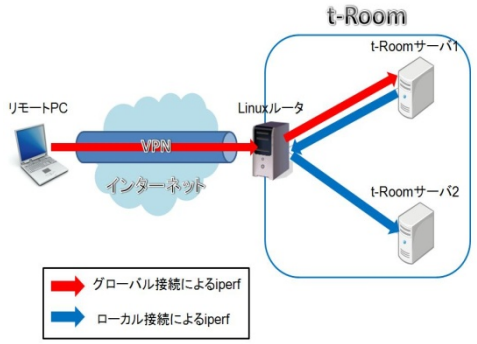

図 3　iperf 実行環境

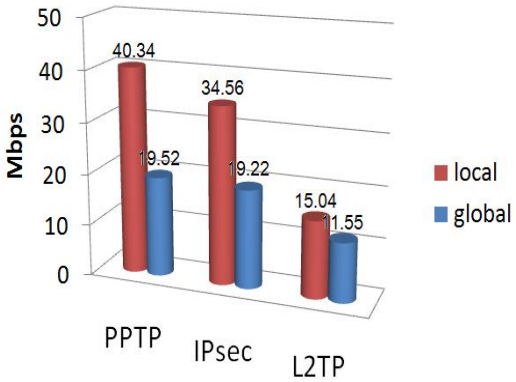

図 4　通信帯域測定結果

表 9　標準偏差

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)
PPTP	2.75	0.05
IPsec	1.15	0.1
L2TP	7.4	2.325

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．測定用サーバとして 1 台の Linux サーバを用意し，Cacti を導入した．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 5 に示す．（O は Output，I は Input を表す）

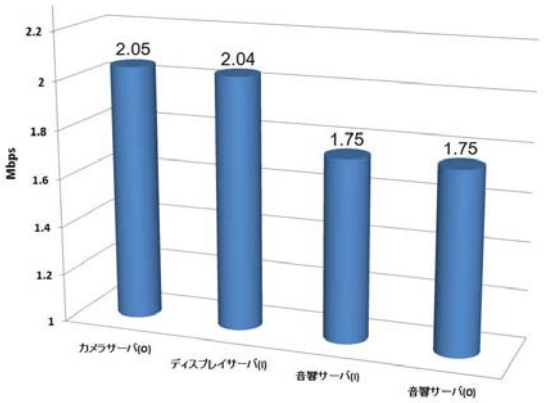

図 5　各サーバのトラフィック測定結果

3.5.2　利用難易度の比較

次にリモート PC を用いてユーザが接続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った．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 画面における設定を 1 ステップと規定し作成した．実験中，設定についてわからないこと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を算出し図 6 に示す．またそれぞれの設定ステップ数と実験中に発生した質問回数を図 7 に示す．設定手順書に記載した PPTP 接続手順を例として図 8 に示す．

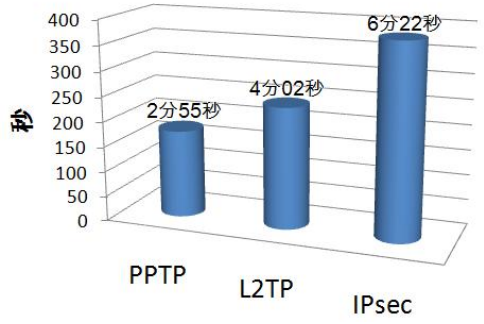

図 6　接続所要時間

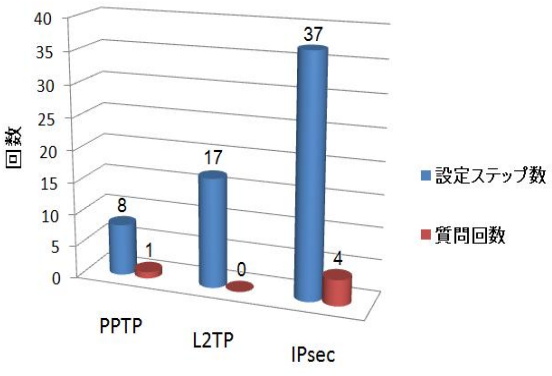

図 7　設定ステップ数と実験中の質問回数

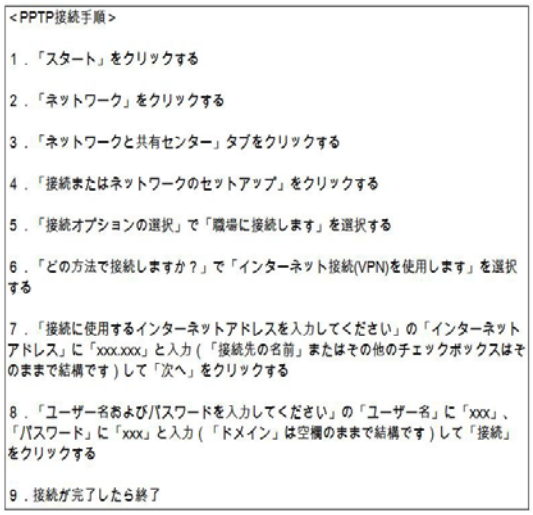

図 8　PPTP 設定手順書

3.6　評価

トラフィック測定結果より t-Room 通信時の各サーバでのトラフィックが得られた．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，どのサーバも 2Mbps 前後のトラフィックで動作していることがわかる．

本実験環境で 4 面のモノリスで構成された t-Room を運用することを想定した場合，これにリモート PC から参加することを想定すると，そのリモート PC が 1 つのモノリスを使用する．つまりリモート PC1 台が t-Room 通信を行うと 3 つのモノリスと通信することになる．この場合リモート PC から見ると，リモート PC 上で動作するカメラからの Output が 1 つ，各モノリスのディスプレイからの 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．各サーバが 2Mbps 程度の通信量を必要とすると考えるとリモート PC1 台が t-Room に参加するためには Output：4Mbps（カメラ・マイク），Input：8Mbps（ディスプレイ・スピーカ），合計 12Mbps 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．このように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となっている．

ローカル接続においては各 VPN プロトコル共にこの条件（4 モノリス通信時）を満たすが，グローバル接続において L2TP 接続は条件を満たしていない．また L2TP はローカル接続では 7.4，グローバル接続では 2.325 と他の VPN プロトコルに比べ標準偏差の値が高く安定した通信帯域を確保できなかった．これに対し PPTP，IPsec はどちらも条件である通信帯域を確保している．標準偏差の値も PPTP はローカル接続で 2.75，グローバル接続では 0.05，IPsec ではローカル接続で 1.15，グローバル接続では 0.1 と安定した数値である．

続いて接続所要時間を検証する．図 7 の接続ステップ数と比例して図 6 のように PPTP，L2TP，IPsec の順に接続所要時間が増加する順当な結果となった．質問の回数からも IPsec では 4 回と他の接続よりも接続の複雑さがうかがえる．今回は L2TP，IPsec ともに簡易な事前共有鍵で設定を行っている．認証方式としてコンピュータレベルでの証明書を利用することが本来推奨されているため，もしこれを利用するとなると所要時間，質問回数はさらに増加するものと考えられる．

また WindowsOS による IPsec 接続の設定では相手先のドメイン指定ができず固定グローバル IP を指定する必要があり，サーバ側が本実験環境のように動的 IP である場合，IP 変化毎に設定変更が生じてしまう．これらの点から IPsec は接続設定の面では他の VPN プロトコルに比べユーザへの負担が大きなものになると考えられる．

3.7　考察

通信帯域，接続の簡易性の二点を考慮すると PPTP 接続が t-Room 環境での VPN 接続には適していると言える．しかし，セキュリティレベルを考えると必ずしもそうとは言えない．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 IPsec に比べセキュリティレベルが低いため長時間の使用や，より機密性の高い情報をやり取りする際の通信には不安要素もあることを念頭に置かなければならない．

4.　SIPサーバの構築

4.1　導入の目的

SIP とは双方向リアルタイム通信を構築するためのセッション制御プロトコルであり，テキスト形式の制御情報を送受信することにより呼制御を行うことができる．SIP はセッションの開始，変更，終了を行うだけのものであり，セッションの内容には関知することがない．そのため他の技術との組み合わせが容易であり，拡張性の高いプロトコルだと言える．主に IP 電話などの呼制御に応用されているが，その拡張性の高さにより今後様々なシステムへの応用が期待されている．また，SIP には呼制御だけでなく情報登録機能，情報通知機能がある．本研究では上述した VPN プロトコル PPTP を接続した上で SIP を動作させることで認証機能の役割を果たすことで，セキュリティを確保している [5][6][7]．

t-Room サービスを起動させるためには通信相手の IP アドレス情報が必要となる．加えて通信相手がどのデバイスを制御するサーバであるのかという情報も必要である．リモート PC などの携帯型端末と t-Room 通信を想定した場合，従来の固定型設定の LAN 間接続に比べネットワーク体系が複雑になる．さらに接続ノード数の増加に伴いノード情報の管理が複雑になることが懸念される．そのため動的に変化するノード情報を取得し適切に管理する手法が求められる．本論文は SIP の情報登録機能，情報通知機能を利用することで t-Room 通信における通信ノード間の情報取得手法を提案する [8]．

4.2　構成要素

図 9　SIP の構成

SIP は図 9 に示すように VPN ノード，SIP サーバ，Location サーバから構成される．

・VPN ノード
VPN での接続機能を有する．VPN は UAC（User Agent Client）として機能し，SIP サーバと通信を行うことで自身のノード情報の登録，または他の VPN ノードの情報を取得することが可能である．

・SIP サーバ
SIP サーバは VPN ノードからの要求を受け取る．また VPN ノードに情報を通知するためのサーバである．Location サーバに対して情報の参照，登録，変更，削除を行い，VPN ノードの情報管理も行う．

・Location サーバ
Location サーバは SIP サーバから送られた VPN ノード情報を登録するためのサーバである．Location サーバ自体はデータベースサーバであり，SIP ではデータベースへのアクセス方法が規定されていないため SIP サーバ間との通信には SIP が用いられていない．Location サーバは SIP サーバ上で同一サーバとして動作させることも可能である．

VPN ノードと SIP サーバの間では SIP メッセージを用いることで通信を行う．SIP メッセージとは複数行のテキストから構成される．

4.3　実装と動作手順

プログラミング言語として C 言語（gcc 4.3.2）を使用した．

・REGISTER
VPN ノードは自身のノード情報を Location サーバに登録するために REGISTER リクエストを送信する．ヘッダフィールドには VPN ノードの情報が記述されており，SIP サーバはノード情報を Location サーバに登録する．REGISTER リクエストは図 9 の(1)，(2)に相当する．
REGISTER リクエストを受け取った SIP サーバはレスポンスコード 200 OK を VPN ノードに返す．これは HTTP プロトコルで定義されているレスポンスコードと同様である．受け取った REGISTER リクエストのヘッダ情報を SAVE_MSG 構造体に保存する．SAVE_MSG 構造体には全てのヘッダ情報が保存されているその中から表に示したヘッダ情報を抽出し Location サーバに登録する．Location サーバのデータベースには MySQL（Ver5.0.51a）を使用する．データベース上にテーブルを作成し，表 10 のノード情報を格納する．Location サーバへの登録は文字列 buff に INSERT 構文を格納し，mysql_real_query()により登録を行う．REGISTER リクエストの動作手順を図 10 に示す．

表 10　Location サーバへの登録情報

ノード情報	目的
To	ノードの論理アドレス
Contact	ノードのIPアドレス
UserID	ノードの名前
GroupIP	ノードの所属名
datetime	ノード情報がデータベースに登録された日時

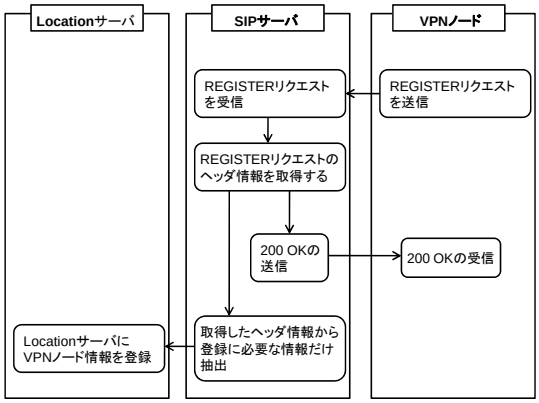

図 10　REGISTER の動作手順

・SUBSCRIBE，NOTIFY
VPN ノードは自身と同じグループに属する VPN ノード情報を取得するため SUBSCRIBE リクエストを SIP サーバに対して送信し，情報の要求を行う．SUBSCRIBE リクエストを受け取った SIP サーバは VPN ノードに 200 OK レスポンスを返し，SAVE_MSG 構造体に受け取った SUBSCRIBE リクエストのヘッダ情報を格納する．そして，データベースに格納されているデータの中から GroupID がクライアントと同じものを SELECT 構文を用いて参照する．参照されたデータは Location サーバから SIP サーバへ送信され XML 形式に変換してメッセージボディに記述し，要求元の VPN ノードへ NOTIFY メッセージとして通知する．SUBSCRIBE リクエストは図 9 の(3)，(4)に，NOTIFY メッセージは(5)，(6)に相当する．SUBSCRIBE リクエスト，NOTIFY メッセージの動作手順を図 11 に示す．

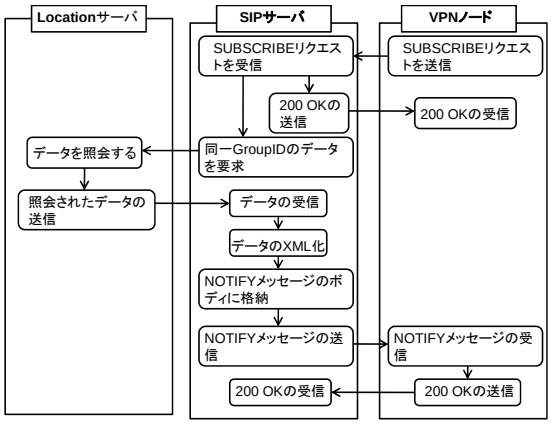

図 11　SUBSCRIBE，NOTIFY の動作手順

4.4 動作確認

コンパイル後，動作を確認する．図 12 に REGISTER リクエストによって登録された MySQL データベースの出力結果を示す．また図 13 は SUBSCRIBE リクエストの実行結果，図 14 は SUBSCRIBE リクエストに対する NOTIFY メッセージの出力結果を示す．

```
mysql> select * from sip_msg;
+--------------------------------+--------------------------+-----------+--------------+---------------------+
| save_To                        | save_Contact             | save_User | save_GroupID | save_Time           |
+--------------------------------+--------------------------+-----------+--------------+---------------------+
| user1 <sip:user1@VPN_Router>   | <sip:user1@192.168.0.1>  | VPN_Router| t-Room       | 2010-07-23 16:21:18 |
+--------------------------------+--------------------------+-----------+--------------+---------------------+
| user2 <sip:user2@remotePC>     | <sip:user2@192.168.0.10>| remotePC  | t-Room       | 2010-07-23 16:23:12 |
+--------------------------------+--------------------------+-----------+--------------+---------------------+
2 row in set (0.00 sec)
```

図 12　データベース

```
<----- send Subscribe request

----------------------------------------
SUBSCRIBE sip:VPN_Router SIP/2.0
Via: SIP/2.0/UDP 192.168.0.10:5060;branch=z9hG4bK-243668316-9384
Max-Forwards: 70
From: user2 <sip:user2@remotePC>;tag=s30886
To: user2 <sip:user2@remotePC>
Contact: <sip:user2@192.168.0.10>
Event: grouplist
UserID: Client2
GroupID: t-Room
Call-ID: 7793-6915-2777@remotePC
CSeq: 9384 REGISTER
Content-Length: 0

----------------------------------------

-----> received OK response

----------------------------------------
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.0.10:5060;branch=z9hG4bK-243668316-9384
Via: SIP/2.0/UDP 192.168.0.10:5060;branch=z9hG4bK-243668316-9384
Max-Forwards: 70
From: user2 <sip:user2@remotePC>;tag=s30886
To: user2 <sip:user2@remotePC>;tag=r38335
Call-ID: 7793-6915-2777@remotePC
```

図 13　SUBSCRIBE リクエスト実行結果

```
-----> received NOTIFY request

----------------------------------------
NOTIFY sip:VPN_Router SIP/2.0
Via: SIP/2.0/UDP 192.168.0.1:5060;branch=z9hG4bK-243668345-5387
Max-Forwards: 70
From: user2 <sip:user2@remotePC>;tag=s30886
To: user2 <sip:user2@remotePC>
Contact: <sip:user1@192.168.0.1>
Event: grouplist
UserID: SIP Server
GroupID: t-Room
Call-ID: 1421-6649-492@VPN_Router
CSeq: 5387 REGISTER
Content-Type: text/xml
Content-Length: 391

<?xml version="1.0" encoding="UTF-8" ?>
<Grouplist>
<AoR="user1 <sip:user1@VPN_Router>">
      <ContactAddr><sip:user1@192.168.0.1></ContactAddr>
      <UserID>VPN_Router</UserID>
      <GroupID>t-Room</GroupID>
</AoR>
<AoR="user2 <sip:user2@remotePC>">
      <ContactAddr><sip:user2@192.168.0.10></ContactAddr>
      <UserID>remotePC</UserID>
      <GroupID>t-Room</GroupID>
</AoR>
</Grouplist>

----------------------------------------
get_HeaderOK

<----- send OK response

----------------------------------------
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.0.1:5060;branch=z9hG4bK-243668345-5387
Via: SIP/2.0/UDP 192.168.0.1:5060;branch=z9hG4bK-243668345-5387
Max-Forwards: 70
From: user2 <sip:user2@remotePC>;tag=s30886
To: user2 <sip:user2@remotePC>;tag=r2362
Call-ID: 1421-6649-492@VPN_Router
----------------------------------------

<?xml version="1.0" encoding="UTF-8" ?>
<Grouplist>
<AoR="user1 <sip:user1@VPN_Router>">
      <ContactAddr><sip:user1@192.168.0.1></ContactAddr>
      <UserID>VPN_Router</UserID>
      <GroupID>t-Room</GroupID>
</AoR>
<AoR="user2 <sip:user2@remotePC>">
      <ContactAddr><sip:user2@192.168.0.10></ContactAddr>
      <UserID>remotePC</UserID>
      <GroupID>t-Room</GroupID>
</AoR>
</Grouplist>

Store NOTIFY OK
done
>>
```

図 14　NOTIFY メッセージ出力結果

図 12 に示したデータベースに登録された同一 GroupID（t-Room）を持ったノードは，SUBSCRIBE リクエストによるレスポンスである NOTIFY メッセージによって VPN ノード情報を取得している様子が図 14 からわかる．図 14 の赤線で囲まれた部分が同一 GroupID を持つ VPN ノード情報である．

5.　まとめ

Linux ルータと SIP サーバの導入により，t-Room へセキュアにアクセスし，その上で動作している t-Room マシンのノード情報を取得する環境を構築した．これにより動的にノード情報が変化する携帯型端末からでも t-Room へ参加することが可能となり，接続の柔軟性は向上したと言える．

しかし現状の PPTP 接続では VPN 接続では十分な帯域を確保できているとは言えない．今後はリモート PC によるクライアント数の増加に伴うネットワークへ負荷を調査し，これに耐えうる対策を考える必要がある．また SIP サーバの動作環境は現状 Linux であるため t-Room の動作環境である Windows へ移植することも今後の課題となる．

6.　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] 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

[5] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E.: SIP:Session Initiation Protocol, RFC 3261 (2002).

[6] Rosenberg, J.: A Presence Event Package for the Session Initiation Protocol (SIP), RFC3856 (2004).

[7] マッキーソフト: 基礎からわかる TCP/IP SIP による VoIP プログラミング(2004).

[8] 岩崎哲弥：“SIP を用いた VPN 確立手法の提案と評価”同志社大学工学部情報システムデザイン学科卒業論文 (2008)

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．また，本研究の SIP 技術について同志社大学在学中の岩崎哲弥氏に様々な御助言，御協力を頂きました．深く御礼申し上げます．