

## t-Room における接続先実行環境を考慮した 動的制御の提案と開発

田中睦也<sup>†</sup> 古屋徳彦<sup>†</sup> 片桐滋<sup>†</sup> 大崎美穂<sup>†</sup>  
平田圭二<sup>††</sup> 原田康徳<sup>††</sup> 高田敏弘<sup>††</sup>  
青柳滋己<sup>††</sup> 白井良成<sup>††</sup>

t-Room の能力を拡張する一環で、従来型の部屋型 t-Room とモバイル計算機を接続する利用形態が試みられている。しかしこの場合、モバイル計算機は、部屋型 t-Room から送られる大量の視聴覚データを自身 1 台で処理しなければならず、通信の質を保つためのなんらかのデータ制御を導入することが求められる。本稿は、この問題に対する対策の 1 つとして、SNMP を用いてモバイル計算機側のネットワーク情報と CPU 使用率などの情報を取得し、その結果を反映した動的な通信制御を行う手法を提案するものである。

## Dynamic Data Transmission Control Based on Mobile Computer Performance Measurement for t-Room

TOKIYA TANAKA<sup>†</sup> NORIHIKO FURUYA<sup>†</sup>  
SHIGERU KATAGIRI<sup>†</sup> MIHO OHSAKI<sup>†</sup>  
KEIJI HIRATA<sup>††</sup> YASUNORI HARADA<sup>††</sup>  
TOSHIHIRO TAKADA<sup>††</sup> SHIGEMI AOYAGI<sup>††</sup>  
YOSHINARI SHIRAI<sup>††</sup>

Aiming at the enhancement of the t-Room's utility, a new idea has been investigated that connects a conventional t-Room and a mobile computer. However, in this case, the mobile computer has to solely handle a considerable amount of multimedia data that is transmitted from the large-scale t-Room, and thus to maintain the quality of communication, any contrivance for controlling the data amount should be incorporated. In this paper, we then propose a solution that enables one to dynamically control the data amount by measuring the network and computer status with SNMP.

### 1. はじめに

同室感を高めることによって遠隔コラボレーション支援を行う t-Room の研究開発が行われている<sup>1)2)</sup>。同室感の実現は、主に視聴覚メディアの対称性の確保に拠る。しかし、例えば映像ディスプレイによる完全な 3 次元映像表示が困難なように、少なくとも現状においては完全な同室感の達成は容易ではない。同室感を高めつつ、t-Room を構成する情報通信技術を駆使した支援機能の追加によって、その不足を補う試みは現実的な遠隔コラボレーション支援システムとしての t-Room の効用を増加させるように思われる。

実際、t-Room どうしを接続する利用法の研究に加え、t-Room (特にことわらない限り大型の部屋型 t-Room を t-Room と呼ぶ) と小型のモバイル端末を接続する利用形態の研究も行われるようになった<sup>3)</sup>。しかし、本質的に多チャンネルの映像信号や音響信号を扱う t-Room の通信データを、一般に貧弱なモバイル端末によって高速に処理することは容易ではない。映像のコマ落ちや通信遅延などの問題を生じ、望ましい通信ができない恐れがある。本研究は、こうした問題を回避するために、モバイル端末の処理能力や通信帯域を測定し、その測定結果に応じて一定の通信品質を確保するために通信量を動的に制御する手法を提案し、その有効性を調査するものである。

### 2. t-Room 通信の概要

#### 2.1 デバイスサーバを中心とした基本動作

図 1 に t-Room の実装例を図解する。t-Room は、モノリスと呼ばれるマルチメディア機器 (デバイス) が搭載された壁で部屋を構成する、部屋型の通信支援システムである。モノリスの配置等に応じて部屋の構成には自由度がある。また基本的に、各モノリスには、ディスプレイとカメラ、マイク、スピーカの 4 種のマルチメディアデバイスが搭載されている。各デバイスは、t-Room 全体を制御するミドルウェアが実装されたサーバ計算機 (例えば、ディスプレイを制御するサーバはディスプレイサーバと呼ばれる) によって制御される。各デバイスサーバの機能は以下のように要約される。

- ディスプレイサーバ  
映像コンテンツを有するサーバ (カメラサーバなど) に接続要求を行い、映像データを取得し、ディスプレイに映像表示を行う。

<sup>†</sup> 同志社大学大学院 工学研究科

Graduate School of Engineering, Doshisha University

<sup>††</sup> 日本電信電話株式会社 NTT コミュニケーション科学基礎研究所

NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation

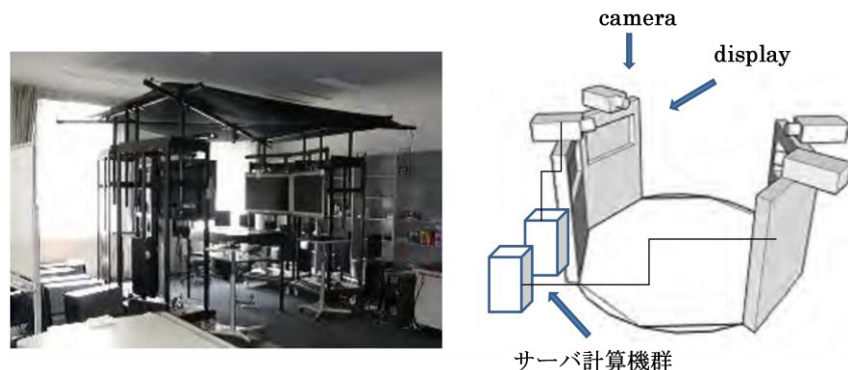


図1 t-Roomの実装例。

((左) 概観, (右) 左図に対応するモニリスとデバイス (特に映像系) 配置)

Figure 1 An example of t-Room implementation.

- ・カメラサーバ  
カメラ映像を配信するデバイスであり、ディスプレイサーバなどの映像表示機能を持つクライアントや映像コンテンツを編集するクライアントからの接続要求に応じて配信を行う。
- ・スピーカサーバ  
音響信号を再生するサーバであり、マイクサーバなどの音データを有するサーバに接続要求を行い、音データを取得し、それをスピーカから出力する。
- ・マイクサーバ  
マイクから音データを収集し、スピーカサーバや音響コンテンツを編集するクライアントからの接続要求に応じて配信を行う。

図2に、図1に示した4面のモニリスを持つt-Roomのサーバ群の構成を図解する。camera01からcamera04までのカメラサーバと、display01からdisplay04までのディスプレイサーバ、そしてウェブサーバ (web svr) と制御サーバ (control svr)、プロキシサーバ (proxy svr) から構成されている。なお、サーバ間のデータ通信プロトコルは映像データに関してはTCPであり、音声データに関してはUDPである。

t-Roomミドルウェアにおいて、各種のデバイスサーバは、その動作開始時にデバイス定義ファイルを読み込むことによって自身が持つべき動作要件を把握するように設計されている。デバイス定義はいわば対応するデバイスの基本的属性を記述するものであり、動作開始時にのみ引用される。また、各デバイスファイルは、対応するデバ

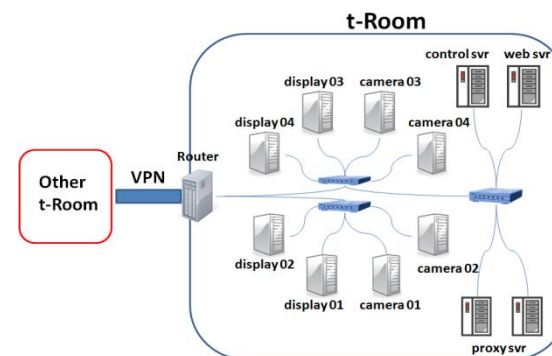


図2 t-Roomのサーバ及びネットワーク構成。

Figure 2 Servers and network configuration of t-Room.

イスサーバが扱うメディアデータコンテンツをさらに詳細に条件づけするコンテンツ定義ファイルを引用する。このコンテンツ定義ファイルには、例えば複数の映像を1枚の映像に統合するようなデータ編集機能や画像の変形処理機能を記述する。編集を動的に行うような柔軟性を実現するため、デバイス定義ファイルと異なり、このファイルは動作開始以降に自由に呼出されるように設計されている。このように、サーバ動作を階層的に定義することによってt-Room全体の安定かつ柔軟な通信の実現が図られている。

図3に、これらの定義ファイルを用いて、ディスプレイサーバが4台のカメラサーバの映像を1枚の映像に編集して表示するまでの基本的な処理の流れを図解する。ディスプレイサーバがt-Room内にある場合、この処理の流れをそのまま利用することができる。一方、ディスプレイサーバがモバイル端末の場合、その処理能力の低さのために、このままの流れを利用することは困難となる。後述するプロキシサーバを用いる解決策との対比のために、まずここではプロキシサーバを用いない基本的な処理の流れを解説する。

- ① ディスプレイサーバの利用者がサーバを立ち上げる。但し、カメラサーバは既に立ち上がっているものとする。
- ② ディスプレイサーバはまず最初にウェブサーバ上に置かれているディスプレイ定義ファイル (ディスプレイサーバ用のデバイスファイル) を読む。
- ③ ディスプレイサーバは、ディスプレイ定義ファイルが引用しているコンテンツ定義ファイルを読む。

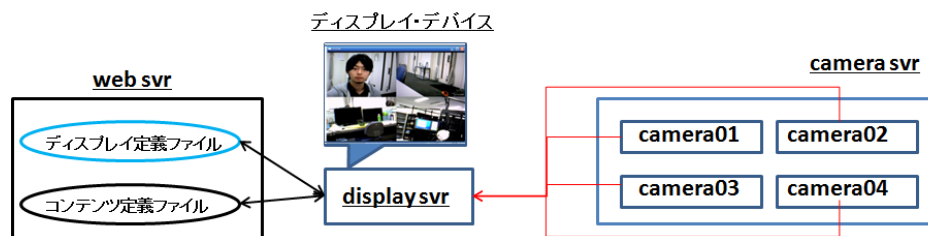


図3 4画面を結合・表示する処理の流れ。

Figure 3 Data processing flow for combining 4 camera data streams.

- ④ ディスプレイサーバは、コンテンツ定義ファイルに記述されている camera01 から camera04 までの映像データを結合する指示に従い、各カメラサーバからデータを取得し、さらに取得したデータを結合して表示用の映像データを生成する。
- ⑤ ディスプレイサーバは、映像再生時に必要となるディスプレイ制御パラメータと共に生成データをディスプレイに送出する。

## 2.2 プロキシサーバを用いる柔軟性の向上

処理速度やメモリ容量が不足しがちなモバイル端末と t-Room との通信を容易にするために、t-Room にはプロキシサーバが導入されている<sup>4)</sup>。プロキシサーバは、ディスプレイサーバ（ここではモバイル端末）に送るべき映像データの画像サイズと動画再生時間間隔（以下、interval と呼ぶ）を変更することと、映像を結合する処理を行うことでディスプレイサーバの処理能力の低さを補うことができる。

図4にプロキシサーバの効用を図解する。図4は、4枚の映像を1枚の映像に統合編集する処理をプロキシサーバが行う点を除いて、他の処理は図3と同じである。この時、プロキシサーバはプロキシ定義ファイルに記述されている命令に従って処理を行い、ディスプレイサーバからの接続要求を待ち受ける。図5と図6に、図4の構成で画像サイズと interval の変更を行った場合の、ディスプレイサーバにおける CPU 使用率と通信量（ディスプレイサーバとプロキシサーバの間における双方向の映像データと音データの合計）の変動を示す。ディスプレイサーバは、Windows XP Professional (SP2) を OS とし、Intel Core2 Duo (1.2GHz) の CPU と、1.99GB RAM のメモリを用いた、カメラやディスプレイ、マイク、スピーカを持つ一般的なモバイル計算機である。計測は、ローカルネットワーク上で、かつセキュリティソフトなどの CPU 使用率に影響のあるバックグラウンドプロセスを全て停止して行った。また、画像サイズは解像度 1280×960 を 100% とし、100%、90%、80%、70%、60% の 5 通りに変化させ、interval は 25 (fps=40)、50 (fps=20)、75 (fps=13)、100 (fps=10) の 4 通りとした。

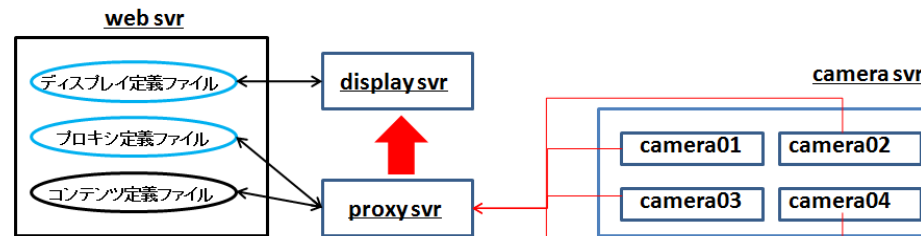


図4 プロキシサーバの役割。

Figure 4 Schematic description of proxy server functions.

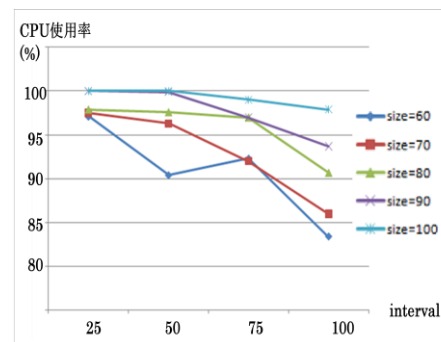


図5 パラメータ操作時の CPU 使用率の挙動

Figure 5 Movements of CPU usage

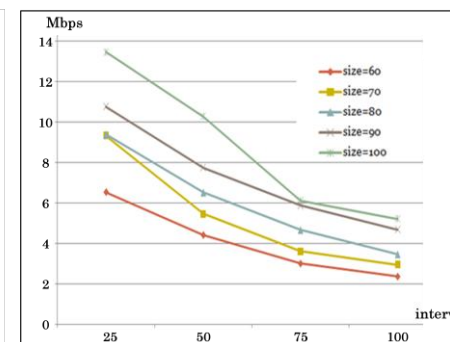


図6 パラメータ操作時の通信量の挙動

Figure 6 Movements of traffic

結果的に 20 通りの計測条件となった。各条件における計測時間は全て 10 分間とした。

図5から、interval の値が大きくなるにつれて CPU 使用率が低下することがわかる。また、画像サイズの減少とともに、CPU 使用率は interval 値にかかわらず全体的に低下するが、interval 値の増加に伴う使用率の下降傾向が強まることを読み取ることができる。

図6からは、やはり interval の値が大きくなるにつれて通信量が低下し、画像サイズの減少とともに通信量は interval 値にかかわらず全体的に低下することがわかる。

以上の、interval 値の増加や画像サイズの減少が確実にディスプレイサーバの CPU 使用率と通信量を低減する関係は、通信品質の決定要因である CPU 使用率や通信量を制御するための操作変数として interval や画像サイズを用いることができることを示している。

### 3. CPU 使用率と通信量の interval を変数とする動的制御手法

#### 3.1 概要

前節において interval と画像サイズがモバイル端末の CPU 使用率（以下、特にことわらない限りモバイル端末の CPU 使用率を指す）や通信量を制御するための有効な変数になりえることを示したが、画像サイズは利用者が利用目的に応じて自由に設定できるものであるべきように思われる。そこで、本研究では、interval を制御変数として CPU 使用率と通信量を動的に制御する手法を検討した。

interval 値は、動画表示におけるフレームの時間間隔であり、t-Room の通信開始時点に設定されていなければならない。従って、通信開始時に参照されるデバイス定義ファイルに記述されており、通信中に動的にその値を変更することはできない。本研究では、この問題を解決し、interval を動的に書き換える手段として複数のプロキシサーバを利用することを考案した。

本実験で実際に構成した 2 式のプロキシサーバを用いる構成図を図 7 に示す。モバイル端末が通信を開始する当初、t-Room との通信は interval にデフォルトの初期値が設定されている proxy1 サーバを経由して行われる。通信が始まると同時に、control サーバがモバイル端末から CPU 使用率や通信量を測定し、予め定められた判定条件に沿った新しい interval 値を含むプロキシ定義ファイルを proxy2 サーバに送付、そしてモバイル端末の接続先を proxy1 サーバから proxy2 サーバに切り替える。この時、proxy1 サーバ用のプロキシ定義ファイル中で引用されるコンテンツ定義ファイルは proxy2 サーバ用に送付しておく。こうして、これ以降、CPU 使用率などの評価パラメータが基準を満たすまで 2 台のプロキシサーバを交互に起動と停止を繰り返しながら望ましいプロキシサーバの状態を生成する。

元々 t-Room では interval 値は 1~1000 の interval 値（最低値はサーバの処理能力に依存する）を扱うことができるが、構成した control サーバでは interval 値を 40~200 (fps 値で 25~5 に相当) の範囲に制限した。これは、制御に伴う interval 値の増減量を ±10 としたこと、コミュニケーションに必要な最低限の映像品質の確保を考慮したことによる。

#### 3.2 SNMP を用いる CPU 使用率と通信量の測定

t-Room ミドルウェアは Windows 上で動いており、またモバイル端末 OS を Windows と想定することは現実的と思われる。しかし、Windows 標準の SNMP Agent 機能では CPU 使用率と通信量の取得ができない。この不足を補うために、拡張された SNMP 用管理情報データベースである標準化拡張 Management Information Base をモバイル端末とプロキシサーバに導入した。なお、SNMP Manager 機能は Ruby のフリーソフトウェアを用いて実現した<sup>5)6)</sup>。

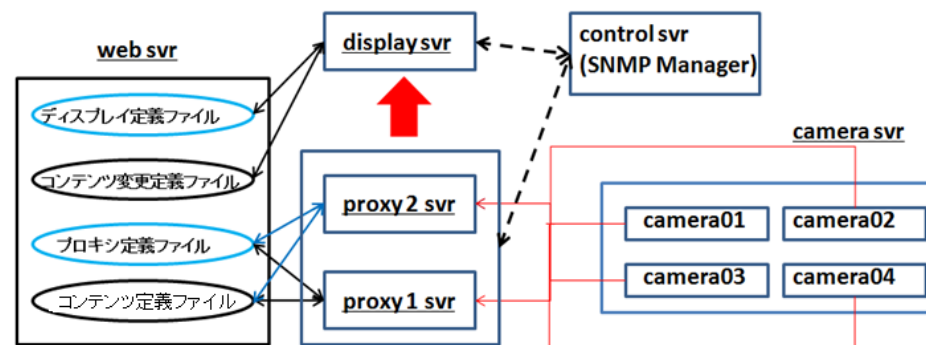


図 7 動的制御システムの構成

Figure 7 Schematic diagram of Dynamic Traffic Control

動的制御を実現するために、CPU 使用率などを監視する対象は、モバイル端末と交互に起動・終了を繰り返す 2 台のプロキシサーバである。従って、この 3 台の計算機上に SNMP Agent を走らせ、監視用に作成した SNMP Manager を制御サーバ上で走らせた。監視に用いる管理情報、即ち、モバイル端末における CPU 使用率と outbound 通信量、そしてプロキシサーバにおける outbound 通信量が、これらの Manager と Agent 間でやりとりされる。

#### 3.3 モバイル端末における最適状態

制御が目指すべきモバイル端末の通信に関する最適な状態を定義する。これは、モバイル端末において画像サイズや通信帯域に関する種々のパラメータの制御に伴う 72 通りの場合分けを行い、それぞれにおける通信品質を事前に調査した結果に基づいて決定した。

##### 【定義】

CPU 使用率が 80~85% に収まるか、あるいはモバイル端末の outbound 通信量とプロキシサーバの outbound 通信量の合計が、初期値として設定する最大帯域の 85~90% に収まっている状態を最適状態とする。ここで、CPU 使用率等に一定の余裕を持たせたのは、コマンド送信や SNMP 通信などの t-Room 以外のアプリケーション実行を可能とする余剰帯域を確保するためである。

### 3.4 interval 値変更の手順

上記の最適状態の達成を目指し、interval 値の変更は以下の手順で行うように定義した。手順は、interval 値の自動的な増減と t-Room 管理者への画像サイズ変更推奨メッセージ発信の 2 つのステージに大別される。後者のステージは、interval 値を上限まで大きくしても CPU 使用率あるいは通信量に余裕がある場合と、interval 値を下限まで小さくしてもそれらの量を最適状態に収めることが困難な場合のための処理である。なお、以下 4 項目における通信量とは、モバイル端末の outbound 通信量と proxy サーバの outbound 通信量の合計である。

1. 以下の条件のいずれかを満たす場合、interval 値を 10 増やす（動画再生時間間隔が長くなるため通信量も処理量も減少する）。
  - interval 値が 200 より小さく CPU 使用率が 85% より大きい。
  - interval 値が 200 より小さく通信量が最大帯域の 90% より大きい。
2. 以下の条件のいずれかを満たす場合、interval 値を 10 減らす（動画再生時間間隔が短くなるため通信量も処理量も増大する）。
  - interval 値が 40 より大きく CPU 使用率が 80% より小さい。
  - interval 値が 40 より大きく通信量が最大帯域の 85% より小さい。
3. 以下の条件を満たす場合、画像サイズの拡大を推奨するメッセージを管理者に提示する（処理量と帯域ともに増大する）。この場合、モバイル端末ではより大きな画像を利用することで通信品質を向上させることができる。
  - interval 値が 40 で CPU 使用率が 80% より小さく、かつ通信量が最大帯域の 85% より小さい。
4. 以下の条件を満たす場合、画像サイズの縮小を推奨するメッセージを管理者に提示する（処理量と帯域ともに減少する）。
  - interval 値が 200 で CPU 使用率が 85% より大きいか、あるいは interval 値が 200 でモバイル端末の通信量が最大帯域の 90% より大きい。

上記の条件を参照し、実験においては interval 値の変更は 15 秒間隔で行った。目標である最適条件を満足したか否かの判定は、CPU 使用率と通信量のいずれか一方が条件を満たした場合に最適状態を満足したものと定義した。また、動作環境は常に動的に変化するため、制御によって最適状態が達成された後も、制御サーバによる監視と interval 値変更による通信制御は引き続き行うものとした。なお、最大帯域値は t-Room 通信の開始前に予め測定しておくものとする。

### 3.5 動的制御の動作手順

図 7 に示す動的制御の動作手順について詳細に説明する。

#### 【動作手順】

- ① t-Room とモバイル端末が通信を開始する時、通信は interval にデフォルトの初期値が設定されている proxy1 サーバを経由して行われる。通信開始と同時に control サーバは SNMP Manager 機能を用いて、モバイル端末の CPU 使用率とモバイル端末の outbound 通信量、proxy1 サーバの outbound 通信量の監視を開始するために SMMP Agent との通信を開始する。
- ② SNMP Manager は、合計の通信量と CPU 使用率を 1 秒毎に監視し、15 秒間のそれらの平均値を評価パラメータとして算出する。
- ③ 3.3 節で定義したモバイル端末における最適な状態に向けて interval を更新するために、3.4 節で述べた判定条件と SNMP 取得平均値と照合する。
- ④ interval 値変更の判定条件に合致した場合、新たに interval 値を変更したプロキシ定義ファイルと、そのコンテンツを生成するコンテンツ定義ファイルを自動生成し、それらの定義ファイル web サーバに転送する。
- ⑤ proxy2 サーバは web サーバの新しい定義ファイルを読み込む命令を control サーバから受信し、新しい interval 値を用いて t-Room を起動する。control サーバは、proxy2 サーバが起動したことを確認した後に、display サーバのコンテンツを生成するコンテンツ変更定義ファイルを生成、web サーバに送信する。
- ⑥ コンテンツ変更定義ファイルには新しい interval 値を用いて起動された proxy2 サーバに映像データを要求する命令が記述されているため、display サーバにて受信される映像データは interval 値が変更されたものとなる。そこで control サーバは、データの配信が確認された後に proxy1 サーバに終了命令を送信し、proxy1 サーバは次の起動命令を待ち受ける。
- ⑦ 動的制御はこれ以降、CPU 使用率などの評価パラメータが基準を満たすまで②～⑥の処理を繰り返すことで、2 台のプロキシサーバを交互に起動と停止を繰り返しながら望ましいプロキシサーバの状態を生成する。
- ⑧ 動的制御の動作は逐一 control サーバの管理者に提示され、SNMP の評価パラメータはグラフとして視覚的に提示される。
- ⑨ t-Room 通信終了時には、control サーバから各制御サーバに終了命令が送信され、SNMP Manager は SNMP Agent との通信を切断する。

動的制御を実行するため本研究で構築した実験システムには、配信データ量の動的制御に加えて、t-Room における動的な帯域制御に必要な SNMP の管理情報を VPN ルータに送信する機能も実装されている。この帯域制御と合わせて interval 値の動的更新を行うことで、t-Room を実行するために必要な帯域を確保しつつ、他のアプリケーションが用いるポートにも最適な帯域を振り分けることが可能になる<sup>7)</sup>。



表1 t-Room 動作のソフトウェア環境.

Table 1 Software environment for t-Room.

t-Room 動作環境	
• WindowsXP (SP2, SP3)	
• WindowsVista (SP2)	
• Windows7	
• Ruby 1.86	

表2 デバイスサーバの仕様.

Table 2 Specification of device servers.

display サーバ, camera サーバ	
機種	Dell XPS720
CPU	Intel Core2 Duo 2.66GHz
メモリ	2.00GB RAM
OS	WindowsXP Professional (SP2)

表3 プロキシサーバと制御サーバの仕様.

Table 3 Specification of proxy and control servers.

プロキシサーバ, 制御計算機	
機種	Dell studio XPS 8100
CPU	Intel Core i7 2.80GHz
メモリ	4.00GB RAM
OS	Windows7 Home Premium

表4 モバイル端末の仕様.

Table 4 Specification of mobile PC.

モバイル端末	
機種	Sony VGN-TT90NS
CPU	Intel Core2 Duo 1.20GHz
メモリ	1.99GB RAM
OS	WindowsXP Professional (SP2)

## 4. 動的制御手法の評価

### 4.1 実験システムの構成

提案手法の評価を, 表 1~表 4 に示す性能を持つサーバやモバイル端末からなる t-Room 実験システムを用いて行った. なお, モバイル端末の台数は 1 とした.

t-Room とモバイル端末を接続するために, VPN (Virtual Private Network) を制御する VPN ルータを Linux ルータを利用して構築した. この Linux ルータは, Windows OS に標準でクライアント機能が備わっている PPTP と IPsec, L2TP の 3 つのプロトコルによる VPN 接続を可能とするものである. なお本実験では, 通信帯域が広く, ユーザビリティが優れていると報告されている PPTP を利用して VPN 接続を行った<sup>8)</sup>. t-Room の LAN 上の機器は全て 1000BASE-TX で接続されており, PPTP はグローバル接続で約 20Mbps の帯域を持つことを確認した.

システムとネットワークの全体構成は, 基本的に図 2 と同様であり, 図中, “Other t-Room” がモバイル端末である点のみが異なる.

### 4.2 評価手法

本実験では提案手法を評価するために, 以下の 3 点の検証を行った.

- (1) 最大帯域を設定した時に, 動的制御によって通信量が最適状態の範囲内に収束するか否かの検証. なお, 最大帯域は 10Mbps とした.
- (2) CPU 使用率の測定に通信量が影響を及ぼさない程十分な通信帯域が確保さ

れている時に, 動的制御によって CPU 使用率が最適状態の最適値幅に収束するか否かの検証. 実験には十分な通信帯域として 20Mbps を確保した.

- (3) CPU 使用率と通信量のいずれかが最適状態の範囲内に収束した場合に, モバイル端末において再送要求や遅延などのコミュニケーション障害が起こっているか否かの検証.

伝送データが通常の t-Room 利用状況を反映するように, 実験では実際に t-Room 内に 3 人の利用者を配置し, モバイル端末は 1 人の利用者が利用する状況を準備した. 結果的に, t-Room 内の 1 面のモニタにモバイル端末利用者の映像を映し出し, モバイル端末ディスプレイには, その利用者自身を含む 4 人の利用者を映し出すような通信が行われた. 音声はモバイル端末と t-Room 内のサーバで双方向通信を行った. 通信されるデータは, モバイル端末から t-Room に送信される視聴覚データと, t-Room 内のプロキシサーバからモバイル端末に送信される 4 画面結合データと音声データであり, 描画領域は t-Room 側では 1280×720 とし, モバイル端末側ではディスプレイ全面画とした. この描画領域で実行する時, 本実験で用いたモバイル端末の性能では, interval 値が最小の 40 の時には CPU 使用率が 95% を超えて通信量は 18Mbps となり, interval 値が最大の 200 の時には CPU 使用率が 60% で通信量は 5Mbps となった (数値は全て 10 分間計測時の平均値). また, モバイル端末においてコミュニケーション障害の要因となる再送処理や処理遅延が起きていないかを確認するために, 受信パケットのシーケンス数/時間を計測した.

図 8 に, 最大帯域を 10Mbps として通信を行った際の通信量の変化を示す. 縦軸が t-Room 通信量 (kbps), 横軸が時間である. 通信量の最適値範囲 (8500~9000kbps) を赤の枠で囲っている. 通信開始時の interval 値を 200 と設定したため, 最適値に収束するために動的に interval 値を減らすことで通信量が増大していることが確認できる. また通信量が最適値に達した場合には最適値を示す赤の枠に収束するために interval の値を動的に増減することで配信データ量を制御していることが確認できた. しかし, 動的制御が行われていると確認できる一方で, 通信量が設定した最大帯域を超えてしまっている様子も見てとることができる.

図 9 には, 20Mbps の広帯域な環境において通信を実行した時の CPU 使用率の変化を示している. 縦軸が百分率, 横軸が時間である. 図中, 黄色線は CPU 使用率を示し, 青線は最大帯域 (20Mbps) に占める通信量の比率である. CPU 使用率の最適値範囲 (80~85%) を赤い枠で囲っている. 同様に通信開始時の interval 値を 200 と設定したため, CPU 使用率の最適値に収束するために interval 値を減らすことで通信量が増大している. また通信量と同様に, CPU 使用率が最適値に達した場合には最適値を示す赤の枠に収束するために, interval の値を動的に増減することで配信データ量を制御している様子が確認できる. また, 前述した 2 つの動作実験時に, プロキシサーバにて送信パケットのシーケンス数/時間を計測したが, シーケンス数と時間が比例して直線を描い

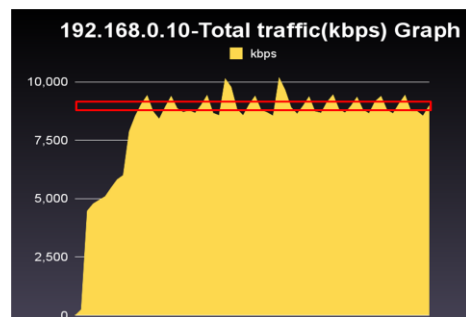


図8 帯域制限時の通信量の変化。  
Figure 8 Traffic change under limited bandwidth (10Mbps).

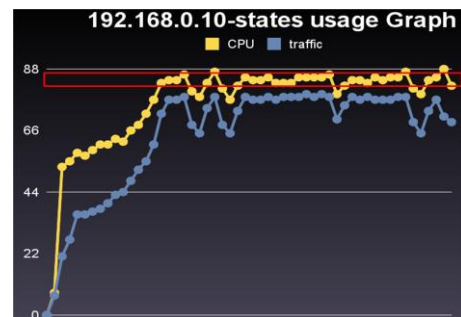


図9 通信帯域確保時のCPU使用率の変化。  
Figure 9 CPU usage rate under wide bandwidth (20Mbps).

ていたことから、パケットロスや遅延は起きていないことが確認できた。以上の結果から、動的制御は適切に動作したものと考えられる。

## 5. 考察

図8中の通信量は、全体としては interval 値の増減とともに最適値範囲に収束しているものの、最大帯域 (10Mbps) を超えてしまうことがあった。この原因としては、t-Room の通信が TCP を用いていることが考えられるが、加えて、提案制御手法が interval 値の増減量を±10に固定したことも考えられる。図6から、描画領域と fps ( $1000/\text{interval}$ ) が大きくなるにつれて、interval 値の変更に伴う通信量の増減が大きくなるのが分かる。その結果、図8では interval 値の増減量が大きいためにかえって収束できず、無駄な制御が多くなってしまったように思われる。interval 値の増減幅を小さくすることでこうした問題は軽減できる可能性はあるが、一方で目的値範囲までの収束が遅くなるのが懸念され、この点は今後さらに検討を要するものと考えられる。

通信量の収束と比べ、図9のCPU使用率の収束は遅い。これは、モバイル端末の処理能力の影響がより強く現れたためと考えられる。このような制御回数が増える問題を解決するために、interval 値の増減量も動的に制御する必要がある。また、一度動的制御を実行したモバイル端末における最適な画像サイズと interval 値は、余程の外的要因が影響しない限りは前回実行時と同じになるはずである。よって、一度動的制御を実行した端末において最適値に収束したのであれば、そのデータを保存すること

で次に制御する時に参照し制御回数を抑えることができるため、何らかの SNMP の不変の情報 (例: Mac アドレス) から端末を特定するような機能も効果的であると考えられる。こうした機能を用いることで SNMP の管理情報の属性として定義される Object Identifier を端末ごとに検証し、SNMP Manager に設定する作業も省くことができる。今後解決すべき課題も残されているが、実際の接続端末の性能を反映して最適な interval 値を自動的に検出できることから、その基本的な有用性が示されたと思われる。

## 6. おわりに

モバイル端末を用いて部屋型 t-Room と通信を行う際に、用いるモバイル端末の性能によって制約される通信量を動的に最適化する制御法を提案し、評価実験を通してその基本的な有効性を明らかにした。しかし、提案手法には接続先実行環境によって効率が悪化するという問題があり、SNMP で一旦取得した計算機属性の記憶や画像サイズと interval の関係性を見出すなどして、interval のパラメータを適切に決定する制御方法の確立が求められる。また、本稿における実験ではモバイル端末が1台であることを想定しており、複数台の端末との同時通信を行う場合の運用については新たな工夫が求められる。具体的には、最も性能が低い端末に合わせて制御条件を設定する方法などが考えられるが、こうした使用状況への対策が今後求められる。

**謝辞** 本研究を進めるにあたって、多くの御助言を頂いた t-Room 研究グループの皆さまに感謝申し上げます。

## 参考文献

- 1) 平田圭二: 未来の電話を考える-遠隔コミュニケーションシステム t-Room, NTT 技術ジャーナル, vol.19, No.6 (2007年6月号), pp.10-12 (2007)
- 2) 入江洋介, 青柳滋己, 高田敏弘, 平田圭二, 梶克彦, 片桐滋, 大崎美穂: t-Room のための遠隔合奏システムの構築, 情報処理学会研究報告 (2009-GN-73)
- 3) 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)
- 4) 土田由里子: t-Room におけるプロキシサーバの有効性の検討 同志社大学工学部情報システムデザイン学科卒業論文 (2009)
- 5) IT TEXT 社, 山居正幸: 実践 SNMP 教科書—ネットワーク管理ツールの開発と活用 (2007)
- 6) CQ 出版社, 斗光佳輝: SNMP ツール開発テクニック (2003)
- 7) 日下真士, 三浦康之, 渡辺重佳: ローカルエリアテレビ会議ネットワークにおける SNMP を用いた帯域制御, 情報処理学会研究報告 (2008-DPS-137)
- 8) 古屋徳彦, 米村裕弘, 片桐滋, 大崎美穂: Linux ルータと SIP サーバを用いた t-Room 間接続環境の構築, 情報処理学会関西支部 支部大会 (2010-H02)