

# Linux 版 t-Room の開発： デバイス制御システムの設計と実装

荻野 裕也<sup>1</sup> 杉本 直也<sup>2</sup> 片桐 滋<sup>1</sup> 大崎 美穂<sup>1</sup>

**概要：**遠隔地間の利用者同士に同室感を与え共同作業を支援することを目的とした「t-Room」では、Windows 特有のメディア処理遅延の大きさを軽減させるために新たに Linux を OS とするミドルウェアの開発が進められている。新ミドルウェアによって制御される Linux 版 t-Room では、先行研究により開発済みの映像伝送システムと音響サーバをカメラなどのデバイスを制御するデバイスサーバとして使用するが、現状ではそれぞれ独立しており t-Room として機能させることはできない。t-Room として動作させるために、これらのデバイスサーバを制御するデバイス制御システムが必要であり、本研究にてその設計と実装を行った。本稿では、デバイス制御システムの設計と実装、3 地点間通信などの実際に想定される状況における動作確認の結果について報告する。

**キーワード：**マルチメディアシステム、遠隔コラボレーション支援、t-Room、デバイス制御システム

## Development of t-Room on Linux: Design and implementation of device control system

YUYA OGINO<sup>1</sup> SUGIMOTO NAOYA<sup>2</sup> SHIGERU KATAGIRI<sup>1</sup> MIHO OHSAKI<sup>1</sup>

**Abstract:** To alleviate the processing delay of a remote collaboration support system called "t-Room" on Windows, we develop a new middleware on the Linux operating system. In previous research, video transmission system and acoustic server are developed on Linux and these work as device servers for video and sound transmission within the new middleware framework. To control these device servers as t-Room, we develop a device control system. In this paper, report its design and implementation result, and show its validation in various environment, like 3 points connection.

**Keywords:** Multi-media system, Remote collaboration support, t-Room, Device control system

### 1. はじめに

企業などを中心にテレビ会議システムのような遠隔地間の対話を支援するシステムが広く普及している。これらのシステムでは音声のみの対話と比べると視覚情報の交換が可能なので、より複雑なコミュニケーションを実現することができる。現在、これらをさらに発展させて新たな付加

価値を与え、共同作業を支援することを目的とした遠隔コラボレーション支援システムの研究開発が進められている [1], [2]。遠隔地の利用者同士に新たな付加価値として、あたかも同じ部屋にいるような感覚である同室感を与え、共同作業を支援することを目的として研究されているのが遠隔コラボレーション支援システム「t-Room」である。

t-Room は、カメラやディスプレイ、マイク、スピーカといったデバイスを用いて映像と音声の通信を実現している。それぞれのデバイスは、カメラサーバやディスプレイサーバなどのソフトウェアにより制御され、これらのソフトウェア群で構成されるミドルウェア「t-Room」によって

<sup>1</sup> 同志社大学大学院 理工学研究科  
Graduate School of Science and Engineering,  
Doshisha University

<sup>2</sup> 同志社大学 理工学部  
School of Science and Engineering, Doshisha University

t-Room 全体が制御され、遠隔地間での通信が実現される。本ミドルウェアは Microsoft 社の Windows OS 上で実装されているが、t-Room のようなリアルタイム通信において Windows のメディア処理速度は不十分である [3],[4]。

このような問題を軽減するため、著者らは、新たにオープンソース OS の Linux 上にミドルウェア「t-Room」を開発し、Linux 上で動作する Linux 版 t-Room を実現することを目指してきた。この中で、Linux 版 t-Room は、カメラやマイクなどのデバイスを制御して映像・音声の通信を行うデバイスサーバと、デバイスサーバを制御して t-Room 全体を機能させるデバイス制御システムとから成るものとして構想した。デバイスサーバには、先行研究で開発済みの映像伝送システムと音響サーバを使用できる [5], [6]。しかし、それらを制御するデバイス制御システムは未開発であった。こうした状況を受け、新たにデバイス制御システムの設計と実装を行ってきた。

本稿は、この設計と実装の詳細を紹介し、また実装したシステムを Linux 版 t-Room として動作させた結果を報告するものである。以下、まず Linux 版 t-Room にも共通する t-Room の基本概念について解説する。次に、Linux 版 t-Room に求められる機能について説明したあと、それを實現するデバイス制御システムの設計と実装を詳述する。最後に、2 地点通信と 3 地点通信を例に実際に想定される場面ごとの動作確認を行い、その結果を報告する。

## 2. 遠隔コラボレーション支援システム「t-Room」

### 2.1 ハードウェア構成

t-Room は、カメラとディスプレイ、マイク、スピーカから成るモニリスと呼ばれる装置を複数台組み合わせることで構成される。t-Room のハードウェア構成の例として、モニリスを 6 台使用した 6 面の t-Room の様子を図 1 (左) に示す。この例のディスプレイとカメラの関係を表したのが図 1 (右) で、図 1 (左) の t-Room を上から見た図となっている。図中で同じ番号の振られた対面するディスプレイとカメラがセットとなり、ディスプレイ周辺に存在するマイクとスピーカ (図では省略) を合わせて 1 台のモニリスが構成される。図 1 では 6 台のモニリスで周囲を囲むことで一種の部屋のような空間を實現し t-Room を構成している。

### 2.2 通信形態

図 1 (左) の 6 面の t-Room を例に実際の通信を行ったときの様子を図 2 に示す。t-Room は基本的に、遠隔地と同じ構成のシステムを配置し通信を行う。図 2 の t-Room1 と t-Room2 は遠隔地に存在し、t-Room1 に赤色で示した人物が、t-Room2 に青色で示した人物がいるものとする。t-Room1 の赤色で示した人物は対面のカメラで撮影され、

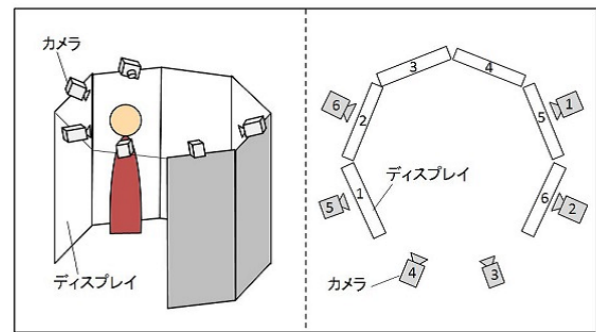


図 1 t-Room のハードウェア構成 (左), およびカメラとディスプレイの関係 (右).

Fig. 1 Hardware structure of t-Room (left), and relation between cameras and displays (right).

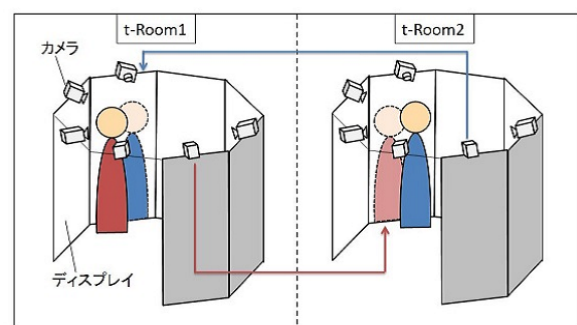


図 2 t-Room の通信の例.

Fig. 2 An example of t-Room communication.

t-Room2 の対応するディスプレイ上に等身大で表示される。t-Room2 の青色で示した人物に関しても同様である。t-Room ではこのような特殊なハードウェア構成と通信形態によって仮想的に部屋を重ね合わせ、遠隔地にいる利用者同士に同室感を提供することを目指す。ここで、人物映像の通信を行うモニリスに注目すると、t-Room 同士の通信は実際には対応するモニリス同士の 1 対 1 の通信の集合であることがわかる。図 2 の例では、t-Room1 の赤色で示した人物の背後にあるディスプレイと対面に位置するカメラは同じモニリスに属している。同様に t-Room2 でも青色で示した人物の背後のディスプレイと対面のカメラは同じモニリスに属しており、通信はこの 2 台のモニリス間で完結している。それぞれの t-Room において残りの 5 台のモニリスにも同様のことが言え、t-Room 同士の通信はモニリス同士の 1 対 1 の通信が組み合わせられて實現していることがわかる。

## 3. Linux 版 t-Room に求められる機能

### 3.1 t-Room 単位での接続機能

前節でも述べたように t-Room 間通信は実際にはモニリス間通信の集合により實現される。しかし、実際に t-Room 間接続を試みる際には、モニリスを意識せず t-Room 単位

で接続相手の指定を行い、通信を実現させるのが望ましい。t-Room を構成するモノリスやモノリスを構成するデバイスサーバの存在を意識することなく t-Room 単位で接続を行った結果、各モノリスとそれを構成するデバイスサーバまで自動的に接続が確立され t-Room 間通信が実現するのが理想である。そのための t-Room 単位での接続制御が必要である。

### 3.2 t-Room 情報の管理機能

t-Room 単位で接続するとき、接続相手となる t-Room の情報が必要である。t-Room の情報とは、t-Room を構成するモノリスの数や各モノリスを構成するデバイスサーバの IP アドレスやポート番号などである。t-Room 間通信を行う際には、各 t-Room が自身を構成しているモノリスやデバイスサーバの情報を保持しておき、接続相手となる t-Room とその情報を交換することで通信に必要な情報のやり取りを行う。このような流れで t-Room 間通信を行うためには、各 t-Room が自身を構成するサーバ群の情報を保持していなければならない。その情報を作成するためには t-Room 内のサーバ構成を管理する機能が必要である。

### 3.3 多地点接続機能

上記の機能により t-Room 間接続が実現した場合、次に考慮すべきなのは接続形態である。Linux 版 t-Room では、最も基本的な 2 地点間通信のみならず 3 地点以上の多地点間通信も想定している。また、2 地点間通信に他の地点が加わり 3 地点間通信に遷移する場合や 3 地点間通信から 1 地点離脱して 2 地点間通信に遷移する場合など様々な状況が考えられる。これらの状況に対応することのできる多地点接続機能が Linux 版 t-Room には求められる。

## 4. デバイス制御システム

### 4.1 階層構造による制御

3.1 節で述べた t-Room 単位での接続機能を実現するために、デバイス制御システムを階層構造で構成し、Linux 版 t-Room 全体を階層的に制御する。Linux 版 t-Room を階層的に制御するときの論理的な全体構成を図 3 に示す。図 3 中の映像サーバと音響サーバは Linux 版 t-Room でのデバイスサーバを示しており、実態は先行研究により開発済みの映像伝送システムと音響サーバである。赤字で記されたモノリス制御サーバ、t-Room 制御サーバ、セッション制御サーバの 3 種類のサーバが本研究で設計と実装を行ったデバイス制御システムである。デバイス制御システムを構成する 3 種類のサーバとデバイスサーバを含めた計 4 種類のサーバは階層構造になっており、モノリス制御サーバ、t-Room 制御サーバ、セッション制御サーバはそれぞれ自身より 1 つ下の階層に存在するサーバをまとめて制御する機能を持つ。再帰的にすべてのサーバが制御し合

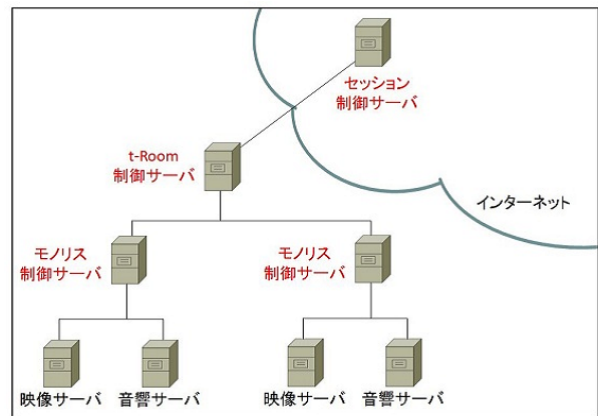


図 3 Linux 版 t-Room の論理的な全体構成。

Fig. 3 Logical structure of t-Room on Linux.

うことで Linux 版 t-Room としての制御を行う構成となっている。

具体的に、モノリス制御サーバは映像サーバと音響サーバを 1 台ずつ制御し、1 台のモノリスとして動作する役割を果たす。t-Room 制御サーバは、複数台のモノリス制御サーバを制御し 1 つの t-Room として動作する役割を果たす。セッション制御サーバは、すべての t-Room 制御サーバをまとめ t-Room 制御サーバ間接続の補助を行う。セッション制御サーバはインターネット上に 1 台のみ存在し、すべての t-Room 制御サーバからアクセスできる。各 t-Room 制御サーバは、セッション制御サーバを介して他の t-Room 制御サーバと情報の交換を行い、セッション制御サーバの指示に従ってお互いの接続を確立する。t-Room 制御サーバ同士を接続する過程で最下層にあるデバイスサーバ同士まで接続され、t-Room 間通信が実現する。実際に操作を行う利用者からすると、t-Room 制御サーバを接続しただけで t-Room 間通信が実現するので、t-Room 制御サーバが t-Room 本体であるかのように見える。このことから、利用者は t-Room 制御サーバの配下に存在するモノリス制御サーバやデバイスサーバを意識する必要がない。このように全体を階層構造で制御することで、利用者にモノリスやデバイスサーバの存在を意識させない t-Room 単位での接続を実現することができる。

### 4.2 コンテンツツリー

3.2 節で述べた t-Room 情報の管理機能を実現するために、Linux 版 t-Room ではコンテンツツリーという概念を導入する。前述したように t-Room 間通信を実現する際、t-Room 同士でお互いの持つ t-Room 情報を交換する必要がある。Linux 版 t-Room では、t-Room 情報をツリー構造で表現したものをコンテンツツリーと呼び、それぞれの t-Room 制御サーバが XML 形式で保持している。コンテンツツリーの論理的な概念を図 4 に示す。

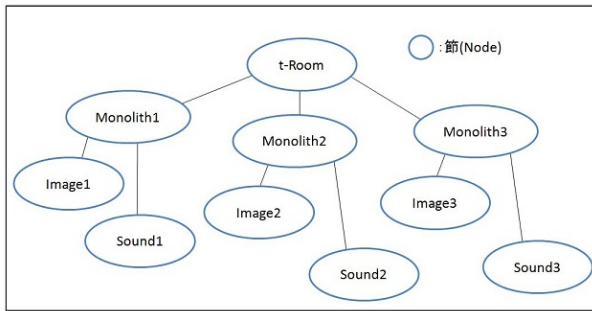


図 4 コンテンツツリーの概念。  
Fig. 4 Concept of contents tree.

図 4 の青色の丸はツリー構造の節 (Node) を表わしており、Linux 版 t-Room におけるサーバを表している。t-Room と記述された節は t-Room 制御サーバを表わしており、同様に Monolith はモノリス制御サーバを、Image は映像サーバを、Sound は音響サーバを表わしている。モノリス制御サーバには明示的に番号が振ってあり、各モノリスの配下に属する映像サーバと音響サーバにも同様の番号が振ってある。図 4 は、映像サーバと音響サーバを 1 台ずつ持つモノリスが 3 台ある 3 面の t-Room のコンテンツツリーを表わしている。コンテンツツリー情報は XML 形式でツリー構造を保ったまま保存され、各節の情報として IP アドレスや通信に必要なポート番号などの情報を格納することができる。

デバイス制御システムでは、モノリス制御サーバと t-Room 制御サーバがコンテンツツリーを作成する機能を持っている。モノリス制御サーバは自身の配下にあるデバイスサーバの情報を集め、それぞれが自分自身のコンテンツツリーを作成する。t-Room 制御サーバは、配下に存在するモノリス制御サーバから各モノリスのコンテンツツリーを集め、それらを統合することで t-Room 全体のコンテンツツリーを作成する。このようにして作成された t-Room のコンテンツツリー情報を見ると、その t-Room に接続されているモノリスの数に加え各モノリスの配下に存在するデバイスサーバの IP アドレスや通信用ポート番号など、t-Room 間通信を実現する上で必要な情報がすべてわかるようになってきている。t-Room 間通信を実現する際には、接続予定のすべての t-Room とコンテンツツリー情報を交換することによって、お互いのサーバ構成と接続時に必要な情報をすべて把握することができる。このようにして各 t-Room で t-Room 情報を管理し、それを t-Room 間で交換することで t-Room 間通信を実現する。

## 5. 2 地点間通信での処理の流れ

本節では、t-Room 間通信で最も単純な 2 地点間通信を例に実際の処理の流れを解説する。いくつかの処理のまとまりに区切って 3 つの小節に渡って解説する。

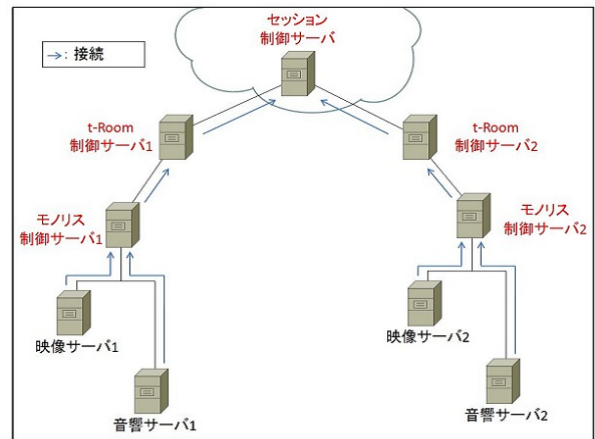


図 5 サーバ間接続の処理の流れ。  
Fig. 5 Process flow chart for connection between servers.

### 5.1 サーバ間接続

Linux 版 t-Room は、まず最初に、各階層のサーバ同士の階層間接続の処理から始まる。その時の処理の様子を図 5 に示す。図 5 は、左右に異なる 2 つの t-Room を示している。明示的に t-Room ごとにそれぞれ t-Room 制御サーバ 1 のような番号が振ってある。同じ番号が振られているサーバが同じ t-Room に属しており、スペースの関係上モノリスは 1 台のみとしている。本来ならばモノリスは複数台存在し、その配下のデバイスサーバである映像サーバと音響サーバも複数台存在する。図の青色の矢印は t-Room 制御サーバ、モノリス制御サーバ、デバイスサーバが自身を制御する上の階層のサーバに対して接続しに行く様子を表わしている。接続により命令のやり取りをするための制御用ソケットが確立されることはすべてのサーバで共通しているが、その他の細かい処理が異なるので処理の流れに従って順に説明していく。

まず、t-Room 制御サーバがセッション制御サーバに接続していく際の処理の流れを説明する。セッション制御サーバは原則として、常時、t-Room 制御サーバからの接続を待つ受付状態にいと仮定する。t-Room 制御サーバは、このセッション制御サーバに接続を試みる。t-Room 制御サーバから接続を受け付けたセッション制御サーバは、接続してきた t-Room 制御サーバの情報を自身に登録する。t-Room 制御サーバ情報には、以下の表 1 の項目が含まれる。

表 1 t-Room 制御サーバ情報。

Table 1 t-Room control server information.

項目	詳細
IP アドレス	t-Room 制御サーバの IP アドレス
制御用ポート番号	制御用ソケット確立に用いるポート番号
通信用ポート番号	通信用ソケット確立に用いるポート番号
t-Room 名	各 t-Room 固有の名前

表1の各項目について解説する。IPアドレスはt-Room制御サーバのIPアドレスである。制御用ポート番号は、セッション制御サーバとt-Room制御サーバ間で確立される制御用ソケットを確立する際に使用するポート番号である。通信用ポート番号は、t-Room制御サーバが他のt-Room制御サーバと通信を行うための通信用ソケットを確立する際に使用するポート番号である。通信用ソケットの使い道については後の節で詳述する。t-Room名は、各t-Roomに固有に割り当てられる名前でも重複は許されない。電話番号に相当する概念で、将来的にt-Room名を使用して接続したいt-Roomを指定し、t-Room間通信を確立できるようにする。

制御用ポート番号と通信用ポート番号に関しては、各t-Roomで重複しないようにセッション制御サーバが自動的に決定する。セッション制御サーバは接続してきたt-Roomに対してIDに相当する固有の値を割り当てる。制御用ポート番号と通信用ポート番号にはあらかじめ最小値が決められている。これらの最小値に先ほど割り当てた固有の値を加算し、それぞれのt-Room制御サーバの制御用ポート番号と通信用ポート番号とすることでt-Room制御サーバ間で使用するポート番号の重複を回避できる。t-Room制御サーバは、セッション制御サーバへの接続が完了するとモノリス制御サーバの接続受付に移る。

モノリス制御サーバは、接続受付状態のt-Room制御サーバに対して接続を行う。t-Room制御サーバとセッション制御サーバの接続処理と基本的な流れは同じで、t-Room制御サーバは接続してきたモノリス制御サーバの情報を自身に登録する。モノリス制御サーバ情報には、以下の表2の項目が含まれる。

表2 モノリス制御サーバ情報。

Table 2 Monolith control server information.

項目	詳細
IPアドレス	モノリス制御サーバのIPアドレス
制御用ポート番号	制御用ソケット確立に用いるポート番号
通信用ポート番号	通信用ソケット確立に用いるポート番号
モノリス名	各モノリス固有の名前

表2の各項目の詳細はt-Room制御サーバ情報と基本的に同様のため詳しい説明は省略する。t-Room制御サーバに割り当てられたIDはt-Room制御サーバによってモノリス制御サーバにも通知され、その値を制御用ポート番号と通信用ポート番号の最小値に加算してそのt-Room内の最小値とする。t-Room内には基本的に複数台のモノリスが存在するため、接続してきた順番を記憶しておきその順番を先ほどのt-Room内のポート番号の最小値に加算することで、すべてのモノリスとの間でポート番号の重複を回避する。モノリス制御サーバは、t-Room制御サーバへの接続が完了するとデバイスサーバからの接続受付に移る。

デバイスサーバは、接続受付状態のモノリス制御サーバに対して接続を行う。t-Room制御サーバとセッション制御サーバ、モノリス制御サーバとt-Room制御サーバ間の接続と基本的な流れは同じで、モノリス制御サーバは接続してきたデバイスサーバの情報を自身に登録する。デバイスサーバ情報には、以下の表3の項目が含まれる。

表3 デバイスサーバ情報。

Table 3 Device server information.

項目	詳細
IPアドレス	デバイスサーバのIPアドレス
制御用ポート番号	制御用ソケット確立に用いるポート番号
通信用ポート番号	通信用ソケット確立に用いるポート番号
遅延測定用ポート番号	遅延測定時に使用するポート番号
デバイスの種類	映像サーバ or 音響サーバ

表3で、t-Room制御サーバ情報やモノリス制御サーバ情報と重複している項目の詳しい説明は省略する。遅延測定用ポート番号に関して、先行研究の映像伝送システムと音響サーバではすべての接続相手との間の通信遅延を測定する必要がある。遅延測定用ポート番号は、遅延測定の際に遅延測定用ソケットを確立するために使用する。t-Room制御サーバに割り当てられたIDはモノリス制御サーバを介してデバイスサーバにも通知される。各ポート番号は、モノリス制御サーバに重複しないようにポート番号を割り当てたのと同様の流れで、すべてのデバイスサーバ間で重複しないように通知されたIDを用いて割り当てられる。デバイスの種類は、映像サーバなのか音響サーバなのかを判断するための項目である。モノリス制御サーバは、映像サーバも音響サーバも同じデバイスサーバとして受け付けているため、デバイスの種類の項目を使用して各デバイスサーバの実態が映像サーバなのか音響サーバなのかを判断する。

以上のように、各階層に存在するサーバ同士が階層間接続を完了し、それぞれのサーバ内に配下のサーバの情報が登録されるまでがサーバ間接続の流れである。この時点で、t-Roomを構成するt-Room制御サーバ、モノリス制御サーバ、デバイスサーバはすべて間接的に接続されて1つのt-Roomとしてまとまっており、セッション制御サーバはそれらのt-Roomの情報をt-Room制御サーバ情報として保持している状態である。

## 5.2 コンテンツツリー情報の作成と交換。

サーバ間での接続が完了すると、t-Room制御サーバとモノリス制御サーバはコンテンツツリー情報の作成に移る。コンテンツツリー情報の作成はモノリス制御サーバから行われる。モノリス制御サーバ内には自身と接続されているすべてのデバイスサーバ情報が保存されている。その情報

```

1 <Monolith>
2   <address>192.168.1.18</address>
3   <mainPort>10000</mainPort>
4   <Image>
5     <address>192.168.1.18</address>
6     <mainPort>10000</mainPort>
7     <delayPort>30000</delayPort>
8   </Image>
9   <Sound>
10    <address>192.168.1.15</address>
11    <mainPort>10001</mainPort>
12    <delayPort>30001</delayPort>
13  </Sound>
14 </Monolith>

```

図 6 XML 形式で書かれたモノリスのコンテンツツリー情報。

Fig. 6 Contents tree information for monolith in XML format.

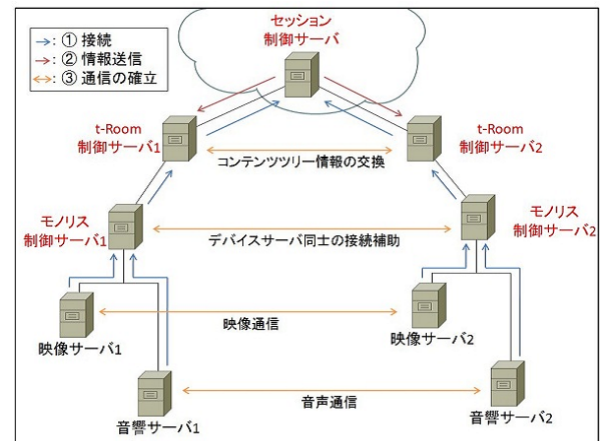


図 8 各階層間での通信の確立。

Fig. 8 Connection establishment for every server layer .

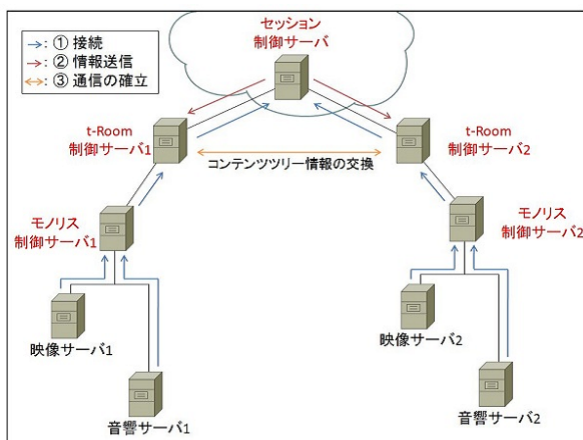


図 7 t-Room 間でのコンテンツツリー情報交換。

Fig. 7 Exchange of contents tree information between t-Rooms.

を読み込みモノリス単位でのコンテンツツリー情報を作成する。コンテンツツリー情報はツリー構造を保持したまま XML 形式のファイルに保存される。その例を図 6 に示す。図 6 中の Monolith はモノリス制御サーバ、Image は映像サーバ、Sound は音響サーバを表しており、address は IP アドレス、mainPort は通信のメインとなる通信用ポート、delayPort は遅延測定用ポートである。このコンテンツツリー情報を読み込むことで、モノリスの配下に存在するデバイスサーバの数と種類、通信に必要な IP アドレスやポート番号などの情報がわかるようになっていく。

モノリス制御サーバはコンテンツツリー情報の作成が完了すると、それを記述した XML ファイルを t-Room 制御サーバに対して送信する。t-Room 制御サーバでは、自身に接続されているすべてのモノリスからコンテンツツリー情報を受信し、それを統合することで t-Room 単位のコンテンツツリーの作成を行う。

t-Room 単位でのコンテンツツリー情報の作成が完了したあとの処理の流れを図 7 に示す。この後、t-Room 制御サーバ間での接続が行われるがそのためにまず、セッショ

ン制御サーバから各 t-Room 制御サーバに対して接続相手の t-Room 制御サーバの情報を送信する必要がある。図 7 の赤色の矢印は、その処理を表している。セッション制御サーバから接続相手の t-Room 制御サーバの情報を受信した t-Room 制御サーバは、セッション制御サーバからの命令を待つ。セッション制御サーバの命令に従って、すべての t-Room 制御サーバ同士が接続受付側と接続側に分かれて処理を行うことで t-Room 制御サーバ間での接続が完了する。このとき使用するのが通信用ポート番号で、ここで通信用ソケットが確立される。通信用ソケット確立後、t-Room 制御サーバはそのソケットを使用してお互いに自身の t-Room のコンテンツツリー情報を送受信しあう。ここまでで、お互いに接続相手となる t-Room の構成や通信に必要な情報を取得できた状態となる。

### 5.3 階層ごとの通信の確立

t-Room 制御サーバ間でコンテンツツリー情報の交換が完了した後の処理の流れについて解説する。各階層ごとに通信の確立が行われた図 8 の状態を目指す。t-Room 制御サーバは、接続相手の t-Room の情報が記述されたコンテンツツリー情報をモノリスごとに分割し、自身の配下のモノリスに対して分配する。t-Room 間通信では、モノリスの数が異なる場合も考えられるが、本研究では最も単純な例としてモノリスの数や各モノリスの配下のデバイスサーバの数がすべて等しい場合のみ扱う。t-Room 同士でモノリスの数などの構成が異なる場合は、接続相手のコンテンツツリー情報を分割する前にモノリス同士の対応付けが必要である。

モノリス制御サーバは t-Room 制御サーバから自身が接続すべきモノリスのコンテンツツリー情報を受信し、接続相手の情報を取得する。セッション制御サーバの命令で t-Room 制御サーバ間接続が行われたのと同様に、t-Room 制御サーバの命令でモノリス制御サーバ間の接続が確立

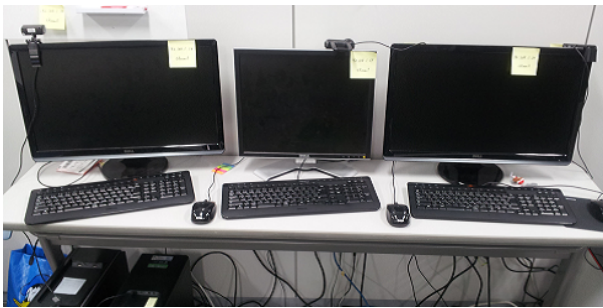


図 9 動作確認の環境.

Fig. 9 Experimental environment for evaluating servers.

される。その後、モノリス制御サーバは接続相手のモノリスのコンテンツツリー情報をデバイスごとに分割し、自身の配下のデバイスサーバに対して分配する。モノリス制御サーバ間接続が行われたのと同様の方法で各デバイスサーバ間接続がなされる。このようにしてすべての階層ごとに通信が確立し、各デバイスサーバが映像と音声の通信を行うことにより、Linux 版 t-Room の 2 地点間通信が実現する。

## 6. 想定される場面ごとの動作確認

### 6.1 概要

開発した Linux 版 t-Room を用いて実際に想定される場面ごとに動作確認を行った結果について述べる。動作確認の環境を図 9 に示す。動作確認では、t-Room 間通信が行われていることを確認するためにカメラ映像を用いる。図 9 の 3 台のディスプレイはそれぞれ異なる PC に接続されている。動作確認では、t-Room 制御サーバとモノリス制御サーバと映像サーバをそれぞれ同じ PC 上で起動し、3 台の PC を用いることで 3 地点の t-Room の環境を模擬的に実現している。ディスプレイに貼られている黄色の付箋には、ディスプレイが接続されている PC の IP アドレスと t-Room 名 (tRoom0, tRoom1, tRoom2) が記載されている。各 PC に接続されたカメラは、これと同じ内容が書かれた付箋を撮影しており、カメラ映像から、接続されている PC および再生されているカメラ映像の t-Room がいずれのものかがわかる。音響サーバは、図 9 とは別の PC 上に実装されており、2 台のみ確保ができた\*1)ので 2 地点間通信の場合のみ使用する。セッション制御サーバは、図 9 の 3 台の PC とは別の PC 上に起動するほうがわかりやすいが、今回はこの 3 台の PC のうち tRoom0 と同じ PC 上にセッション制御サーバも起動する。Linux 版 t-Room の概念としては、セッション制御サーバはインターネット上に存在するものであるが、動作確認を行う上ではすべての t-Room 制御サーバからアクセスできる場所に起動していればいいので同じ PC 上でも問題ない。

\*1) これは、技術的制約ではなく機材調達都合による。

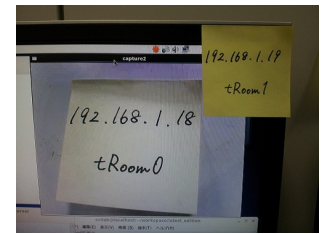
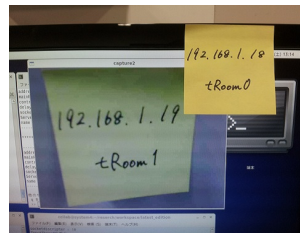


図 10 2 地点間通信 (tRoom0). 図 11 2 地点間通信 (tRoom1).

Fig. 10 2-point connection (tRoom0). Fig. 11 2-point connection (tRoom1).

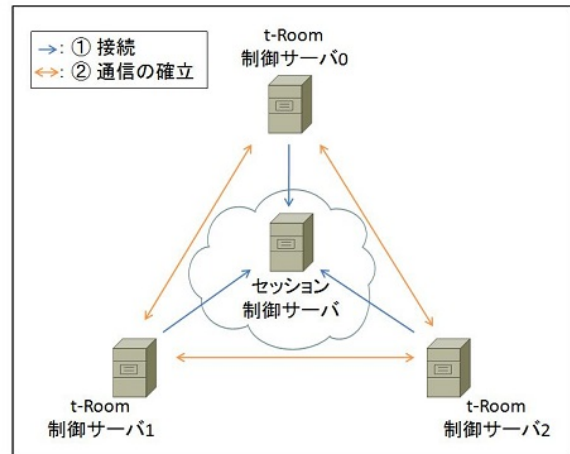


図 12 3 地点間通信.

Fig. 12 3-point connection.

### 6.2 2 地点間通信

2 地点間通信では 5 節で解説した手順で t-Room 間通信が行われる。2 地点間通信での動作の様子を図 10 と図 11 に示す。3 台ある PC のうち t-Room 名が tRoom0 と tRoom1 の PC を使用して動作確認を行った。図 10 は tRoom0 のディスプレイを写した写真で、ディスプレイ上に tRoom1 と記載された付箋の映像が映っていることがわかる。図 11 も同様に、tRoom1 のディスプレイ上に tRoom0 と記載された付箋が映っている。これはお互いに接続相手のカメラ映像を表示していることになるため、2 地点間での t-Room 間通信がうまく行われていることが確認できる。

### 6.3 3 地点間通信

次に、3 地点間通信での動作確認を行う。3 地点間通信では、図 12 に示すように 3 つの t-Room 制御サーバがそれぞれセッション制御サーバに接続を行い、セッション制御サーバの命令によりすべての t-Room 制御サーバ同士が接続する形となる。3 地点間通信の場合、1 台の t-Room 制御サーバは他の 2 台の t-Room 制御サーバと接続され、コンテンツツリー情報の交換をそれぞれの t-Room 制御サーバと行う。結果的に、すべての t-Room 制御サーバは自身以外の 2 地点の t-Room のコンテンツツリー情報を取得することとなり、モノリス制御サーバとデバイスサーバにも

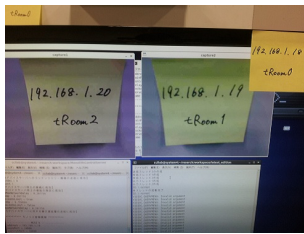


図 13 3 地点間通信 (tRoom0).

Fig. 13 3-point connection (tRoom0).

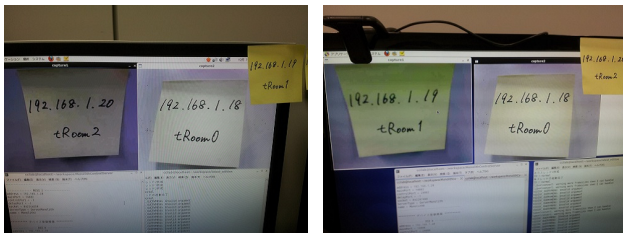


図 14 3 地点間通信 (tRoom1). 図 15 3 地点間通信 (tRoom2).

Fig. 14 3-point connection Fig. 15 3-point connection (tRoom1).

2 地点分の情報が引き渡される. 図 12 のように t-Room 制御サーバ同士が接続したのと同様に, モノリス制御サーバ, デバイスサーバもそれぞれ同じ階層に属するサーバと 2 地点分の接続を行う. 3 地点間通信での動作の様子を図 13, 図 14, 図 15 に示す. 図 13 は tRoom0 のディスプレイの様子である. ディスプレイ上には tRoom1, tRoom2 と記載された付箋の映像が映っていることが確認できる. これは, tRoom0 の映像サーバが他の 2 地点の映像サーバと通信を行い, 他地点の t-Room の映像を表示していることになる. 図 14 は tRoom1 のディスプレイの様子で, tRoom0 と同様に他地点の t-Room の映像が表示されている. 図 15 は tRoom2 のディスプレイの様子で, こちらも同様に他地点の t-Room の映像が表示されている. これにより, 3 地点でもデバイスサーバ同士の通信まで正常に行われていることが確認でき, t-Room 間通信がうまく行われていることがわかる.

#### 6.4 2 地点間通信と 3 地点間通信での遷移

3 地点以上の多地点間通信を行う場合, リアルタイムで地点数が増える状況が考えられる. Linux 版 t-Room では, そのような状況も想定して任意のタイミングで接続地点数を変更することができる. その最も単純な例として, 2 地点間通信と 3 地点間通信での遷移を対象に動作確認を行った. 図 16 はそのときの様子を示すもので, 2 地点間通信から 3 地点間通信へ遷移し, その後再度 2 地点間通信に遷移してから通信が終了するまでの様子を表している. 図 16 の縦軸はネットワーク通信量で単位は byte, 横軸は時間で単位は秒である. 通信量が 0 付近のときには通信は行われておらず, 2 地点間通信と 3 地点間通信に遷移する際に

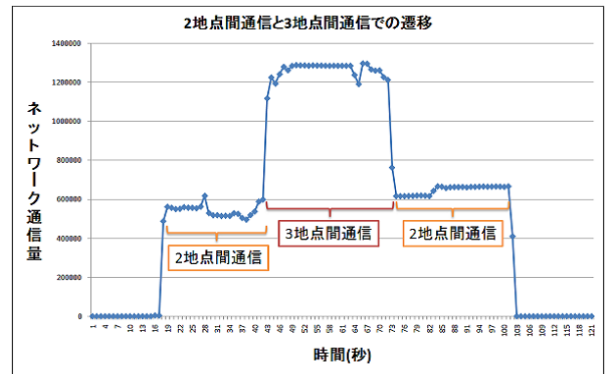


図 16 2 地点間通信と 3 地点間通信での遷移.

Fig. 16 Transition between 2-point connection and 3-point connection.

通信量が増えていることがわかる. 図 16 を見ると, 通信中に動的に接続形態を遷移できていることが確認できる.

## 7. おわりに

Linux 版 t-Room の実現を目指し, 新ミドルウェアの一部であるデバイス制御システムの設計と実装, その動作確認を行った. その結果, 2 地点通信と 3 地点通信においてデバイス制御システムがデバイスサーバの制御を行い, 新ミドルウェアとしての機能を設計通りに果たしたことを確認できた. また, 先行研究の結果である映像サーバなどと共に運用することによって, 実装システムが映像と音声の通信が可能な Linux 版 t-Room として動作することも確認できた.

謝辞 本研究を進めるにあたり, システムの設計から開発に至るまで数多くの御助言を頂きました山口毅氏に深く感謝致します.

## 参考文献

- [1] Hirata, K., Harada, Y., Takada, T., Aoyagi, S., Shirai, Y., Yamashita, N., and Yamato, J.: *The t-Room: Toward the Future Phone*, NTT Technical Review, Vol. 4, No. 12, pp. 26-33 (2006).
- [2] 森川治: 超鏡: 魅力あるビデオ対話方式を目指して, 情報処理学会誌, 41, 3, pp.815-822 (2000).
- [3] 井ノ口浩平: 遠隔コラボレーション支援システム「t-Room」におけるメディア同期について: 音声遅延量について, 同志社大学 理工学部 情報システムデザイン学科 卒業論文 (2012).
- [4] 李榮宰: 遠隔コラボレーション支援システム「t-Room」におけるメディア同期について: 映像遅延量について, 同志社大学 理工学部 情報システムデザイン学科 卒業論文 (2012).
- [5] 村上昂: 遠隔コラボレーション支援のための映像伝送システムの開発, 同志社大学大学院 工学研究科 情報工学専攻 修士論文 (2013).
- [6] 岩原正典: ローカル・ラグ制御機能と同期機能を持つ音響サーバの構築, 同志社大学大学院 工学研究科 情報工学専攻 修士論文 (2013).