

# 多様な接続手法に対応したデスクトップ画面共有システム

非会員 根岸 佑也\* 正員 河口 信夫\*\*

A Desktop Screen Sharing System based on Various Connection Methods

Yuya Negishi\*, Non-member, Nobuo Kawaguchi\*\*, Member

Recently it became very common to use information devices such as PCs during presentations and discussions. In these situations, a need arises for techniques that allow a smooth switch of presenters without changing cables, or an easy screen sharing in case of remote videoconferences.

In this paper, we propose a desktop screen sharing system that can be used for such purposes and situations. For that, we designed an automatic control of connections in the VNC system that can be operated remotely over the network. We also suggested an interface that assigns a role such as “Screen sender” or “Screen receiver” to each terminal. In the proposed system, while sharing a screen between multiple terminals, one can easily display and browse the screen without having to understand how the others are connected. We also implemented a “role card” using contactless IC card, where roles are assigned only by placing the card in the IC reader.

キーワード：会議支援，画面共有，VNC，グループウェア，実世界指向インタフェース

**Keywords:** Conference support, Screen sharing, VNC, Groupware, Real world oriented interface

## 1. はじめに

PCやPDAなどの普及により、情報端末を用いて会議やプレゼンテーション、ディスカッションなどを行う機会がますます多くなってきている。そのような場では、プレゼンテーションにおいて発表者が自分の端末画面を聴衆に提示したり、会議において自分の画面を他端末へ表示させたりすることが求められている。しかし、画面を他の人に見せるためには一般にプロジェクト等のケーブルの差し替えが必要となり、切り替えは必ずしも円滑とは言えない。また遠隔地間でのTV会議等においても手軽に画面共有を行う手法が求められている。

そこで、複数人が集まる多様な状況において、手軽に画面共有を行えるシステムの構築を考える。ケーブルの差し替えを不要にするアプローチの一つにネットワーク経由で画面を転送することが挙げられる。最近ではノートPCに

無線LANが標準的に搭載されているので、これを利用すればディスプレイケーブルの無線化を期待できる。ネットワーク経由で端末のデスクトップ画面転送を行う遠隔操作システムとしてはVNC<sup>(7)</sup>が存在する。しかし、VNCではビューワからサーバへの接続が一般的であり、画面を自分から他の端末に転送したい場合に不向きである。Liseten Modeを利用すればサーバからも接続可能であるが、いずれの場合も端末の利用方法が固定的である。また、接続時には相手のIPアドレスなどを事前に知る必要もある。

3台以上の複数端末間での画面共有を考慮した画面転送システムとしては三浦らのcomDesk<sup>(1)</sup>が存在する。comDeskはサーバとクライアントを区別しないP2P型の画面転送・遠隔操作システムである。参加端末の一覧をユーザにアイコンで提示し、アイコンのドラッグ&ドロップ操作によって手軽に接続指示を行えるインタフェースを備える。しかし、接続は1対1ごとに行う必要があり、端末の数が多くなればなるほど、その接続関係の把握や切替の手間が多くなることが予測される。複数端末の接続管理を考慮した画面転送システムとしてはMultiVNC<sup>(2)</sup>が存在する。しかし、MultiVNCはクライアントとサーバの関係が固定的であるので、複数端末を表示側と転送側に自由に切り替えるような利用方法には向いていない。

また、ユーザにIPアドレスを意識させない端末指定の方法として、名前を利用することが多い。しかし、あらか

\* 名古屋大学大学院 情報科学研究科 情報システム学専攻

〒464-8601 名古屋市千種区不老町 1

Department of Information Engineering, Graduate School of Information Science, Nagoya University

1, Furo-cho, Chikusa-ku, Nagoya, Aichi 464-8601

\*\* 名古屋大学 情報連携基盤センター

〒464-8601 名古屋市千種区不老町 1

Information Technology Center, Nagoya University

1, Furo-Cho, Chikusa-ku, Nagoya, Aichi 464-8601

じめ相手端末の名前を知っておかねば、正しい接続先を指定できない問題がある。このような問題に対して、実世界指向インタフェースにより接続させたい機器をユーザが物理的に指定可能にする研究がすでにいくつかなされてきた。例えば、綾塚らの TranStick<sup>(3)</sup> や Gaze-Link<sup>(4)</sup>、岩崎らの Touch-and-Connect<sup>(5)</sup> などが挙げられる。しかし、これらは複数の共有ディスプレイや閲覧者各自の端末にも表示しているような複雑な画面共有下において、どのように接続したら良いか分からないが画面を提示したいといったあいまいな接続要求を解決するまでは至らない。

我々は、会議やディスカッションなどの支援を目的とした多様な接続手法に対応するデスクトップ画面共有システムを提案する。本システムの特徴はネットワーク転送によりディスプレイケーブルの差し替えが不要なこと、IP アドレスなどの接続先端末の設定項目の事前調査が不要なこと、そして、任意の端末間の接続を明示的に指示できることに加えて、端末の役割決めによる自動接続を行うことが挙げられる。役割決めによる接続指示では、既存のシステムのように個々の端末間の接続に着目せず、端末、個々の役割に着目する。各端末ごとに「画面を転送する」、「画面の受信する」といった役割を与えることにより、ネットワーク内における画面転送の接続関係を自動的に構築する。このため、参加端末数に関係なく、役割変更を行う端末数の操作のみで画面共有を開始、切替できる。さらに、我々は各端末の役割関係を用いた接続ルール・解除ルールを定義することで、自動構築の手順を単純化することができた。これにより、ユーザに複雑な接続関係を意識させることなく、画面共有を実施できる。実装したシステムでは、Felica<sup>(9)</sup>を用いた役割カードを作成し実現した。

以下では、提案システムが想定する多様な画面共有の状況について議論し、提案システムの構築について述べる。

## 2. 多様な状況での画面共有

会議やプレゼンテーション、ディスカッションなどを行う場合には、プロジェクタやプラズマディスプレイを利用して多人数で情報を共有し合う状況や、広めの会議場においてそれら大型ディスプレイを複数利用した状況、共用可能なディスプレイが存在しないような状況、複数拠点間での遠隔ミーティングなど様々な状況が考えられる。我々はそれら画面共有における状況を以下のように共有ディスプレイでの表示、分散ディスプレイでの表示、遠隔ディスプレイでの表示と3つの利用方法に分類した。

**2.1 共用ディスプレイでの表示** 会議室などでは各自の端末に加えてプロジェクタ等の共用ディスプレイへ表示できる端末があることが想定できる。このような場合、転送元端末はこの共用端末へ接続することにより複数人と画面を共有することが可能になる。また、共用端末から他の端末へ画面を転送することにより、参加者への画面配信を実現すると同時に転送元端末の負荷を軽減し、閲覧者側の端末と共用端末との接続を保ったまま発表者の切り替え

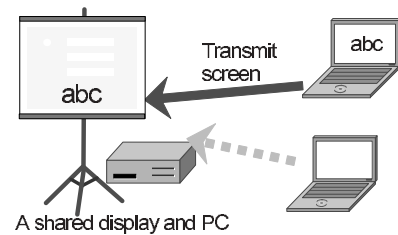


図1 共用ディスプレイでの表示

Fig. 1. A Screen sharing with a shared display.

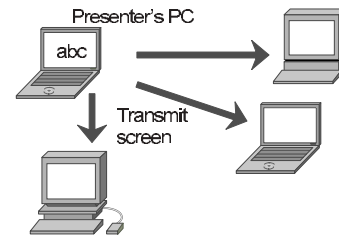


図2 分散ディスプレイでの表示

Fig. 2. A Screen sharing with multiple displays.

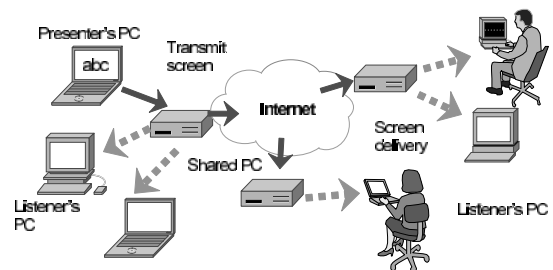


図3 遠隔ディスプレイでの表示

Fig. 3. A remote screen sharing.

ができる。この様子を図1に示す。

**2.2 分散ディスプレイでの表示** 共用のディスプレイがない場合は、参加者の各端末を転送先として接続すれば各端末の分散ディスプレイ間で画面共有が実現できる。この様子を図2に示す。しかしながら、前章で指摘したように、端末の数が増えれば増えるほど接続切替のための操作量が増えてしまう。そこで、この指示を一つの操作で行うことができるならば、個別に端末どうしを接続することなく瞬時に複数端末間で画面切り替えができるが期待できる。

**2.3 遠隔ディスプレイでの表示** テレビ会議と並行して、資料提示のために遠隔地間での画面共有も行えることが望ましい。そこで、先に挙げた共有ディスプレイや分散ディスプレイをインターネット経由で行うことを考える。この時に、まず各拠点の共用端末を中継局として利用し、参加者は自分の拠点の共用端末と接続すれば、各端末の負担を軽減することが可能である。なお、その場合、共用端末より先の接続状況は外の拠点から把握できない。そこで、資料提示者の交代による画面切替の際には、システムが自動的に共用端末を含む各端末の接続関係を再構築することが望ましい。これにより、ユーザは他の拠点での接

続関係を気にせず参加できる。この様子を図 3 に示す。図より遠隔ディスプレイでの表示は共用ディスプレイでの表示と分散ディスプレイでの表示を多段に組み合わせたものと見なすことができる。

### 3. 多様な接続を実現する画面共有システムの設計

**3.1 システムへの要求項目** 1 節で述べた既存システムでの問題点を踏まえ、2 節で述べた多様な接続状況を手軽に実現するためには、以下の性質を満たす画面共有システムを構築すれば良いと考えられる。

- (1) 端末画面をネットワーク経由で転送可能
- (2) 各端末が画面転送と表示の両機能を所有
- (3) IP アドレスなどの事前調査が不要
- (4) 任意の端末間の接続を指示可能
- (5) 複数人での接続・切替に配慮したインタフェース
- (6) 接続関係の管理に対するユーザの負担を軽減

まず、我々は画面共有における各端末の役割に着目し、役割を分類した。そして(1)(2)(3)を実現するために、システムがネットワーク内における各端末上の VNC サーバと VNC ビューワの存在を自動的に収集し、VNC セッションを制御する手法を提案する。現在、VNC サーバは様々なプラットフォーム上で提供されているため、多プラットフォーム向けの画面転送用に利用できる。(4)(5)(6)を実現するために、ユーザが画面共有における各端末に対して役割決めを行うインタフェースを提供する。システムは与えられた各端末の役割に従って自動的に端末間を接続する。我々はこのような手法に基づく画面共有システム Cvnc を構築した。以下、それぞれの手法について述べる。

**3.2 画面共有における役割の定義** 画面共有に参加する際に転送、表示以外に 2 章で述べたように共用端末を中継局として扱う場合を考慮し、画面共有における役割を以下のように定義した。

役割	説明
Sender	自端末の画面を転送
Receiver	画面を受信し表示
Repeater	画面を受信すると同時に、その画面を転送 (共有ディスプレイ)

**3.3 VNC セッションの制御** 一般的な VNC の利用による画面転送開始の流れは(1)まず画面を転送する端末上で VNC サーバを起動し(2)表示先端末上のビューワからその VNC サーバと接続するのに必要な各情報をユーザが入力し(3)認証などの通信を経て完了する<sup>(8)</sup>。必要な情報は相手端末の IP アドレス、使用するポート番号、ディスプレイ番号、パスワードである。提案手法では、それらの情報をあらかじめ端末間で通知し合っておくことによりユーザが入力する手間を省く。通知するパスワードのみは初回起動時にユーザに入力してもらい、他は自動取得する。この様子を図 4、5 を用いて説明する。

図 4 中の Cvnc は本システム、Viewer は Cvnc と連携可

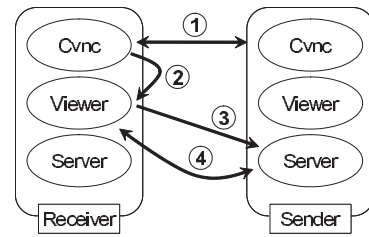


図 4 VNC 接続の制御 (ビューワから接続)  
Fig. 4. VNC session control (Request from Viewer).

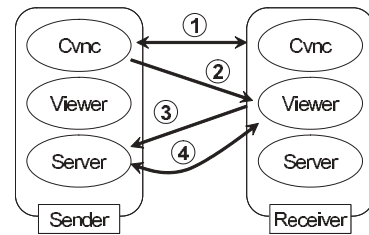


図 5 VNC 接続の制御 (サーバから接続)  
Fig. 5. VNC session control (Request from Server).

能な VNC ビューワで、Server は VNC サーバである。(1)まず Cvnc 起動時に、自己の VNC サーバとの接続に要する情報を他端末の Cvnc にブロードキャストして通知する。(2)そして、ユーザから図 4 中の Receiver 端末から Sender 端末の画面を表示するよう指示があった場合、Receiver 端末の Cvnc は取得済みの Sender 端末の VNC サーバに関する情報より Viewer を生成し、接続を試みる。(3)認証が成功すれば、VNC セッションが確立する。なお、接続開始を要求する端末は表示先である必要はない。

同様の流れで図 5 のように(2) Sender 端末上の Cvnc が Receiver 端末上の Cvnc に Viewer を生成して自己端末に接続するように通知すれば、Sender からも VNC セッションを開始できる。このように VNC サーバとビューワの接続をユーザの代わりに制御することにより、VNC でもサーバとクライアントを区別することなく、また煩雑な入力作業なしに画面転送を開始可能となる。

**3.4 役割決めによる接続インタフェース** 前節で述べた VNC セッションの制御において、画面共有を開始するにあたり、どの端末どうしを接続するのかユーザが指示できるインタフェースが必要である。会議などを行っている際のユーザの要求は、手間をかけることなく素早く画面を他の端末上に転送することである。そこで我々は、一般ユーザの操作を、画面を送ると受け取るの 2 種類に限定し、接続する端末間の指定を Cvnc が自動的に行うことで、そのような要求を満たせると考えた。これを、役割決めインタフェースと呼ぶことにする。

**3.4.1 接続ルール** ユーザにより、端末に 3.2 節で定義した役割が与えられたならば、端末はそれぞれの役割をブロードキャストによって相互に通知する。そして、以下のルールに基づき各端末が自動的に VNC セッションを

構築していく。

- Sender : 何もしない
- Receiver : Repeater > Sender の優先度でそれらを発見したら画面を受信
- Repeater : Sender を発見したら画面を受信し、画面を受信中ならば他の Repeater に画面を転送

Repeater が存在する場合は、Sender 端末にかかる負荷を軽減させるために Repeater への接続を優先する。なお、画面を接続する際、Sender を特定できる一意の ID を共有し合うことにより、各端末上で Viewer を生成する際にその ID をキーとして重複接続を防ぐ。

**3.4.2 画面転送の停止・切替** 画面共有を終了する際には、各端末の役割を解除し、解除前の役割ごとに以下の解除ルールに基づき接続を停止する。

- Sender : 自分との接続を切断するよう受信端末に通知
- Receiver : 自端末上の Viewer を全て閉じる
- Repeater : Sender より切断通知を受けたら対応する Viewer を閉じ、他の Repeater に対してもその Sender に対応する Viewer を閉じるよう通知

Sender と対応する Viewer の特定には、接続ルールと同様の ID を用いる。発表者の交代など転送画面の切替時には、Sender の役割を解除後、次の端末に Sender の役割を与えることで行う。この時に、役割変更のない Repeater や Receiver 端末に対して、ユーザが操作を行う必要は一切ない。これにより円滑な切替を実現する。

**3.5 役割決めによる接続アルゴリズム** 3.3 節と 3.4 節で述べた手法に基づき、自動的に VNC セッションを構築するアルゴリズムを設計した。各端末はユーザより与えられた役割に従い、以下の動作手順に基づく処理と通信を行う。

ここで myRole は自己の役割を保持する変数、myVncServerInfo は自端末上の VNC サーバと接続するのに必要な各種情報、serverList は、収集した各端末上の VNC サーバとその接続情報および各端末の myRole 変数を保持するリスト、forwardingServerInfo は Repeater が転送元としている Sender の VNC サーバ情報とする。

また、以下の命令と関数を定義する。引数のデータ集合をネットワーク内の端末にブロードキャストする Broadcast 命令。自端末上の VNC ビューワを全て閉じる CloseAllViewer 命令と、引数が示す VNC サーバと接続するビューワを閉じる CloseViewer 命令。第 2 引数にしたがって、第 1 引数のリストを追加・削除・更新を行う Update 命令。第 1 引数の VNC サーバのビューワを第 2 引数の端末上に生成する Connect 命令。引数のリスト内にそれぞれ Repeater と Sender の役割を持つ端末情報を含むかどうかの真偽値を返す ContainRepeater 関数と ContainSender 関数。引数のリストから、Repeater と Sender の要素を取り出す GetRepeater 関数と GetSender 関数。他端末より受信したメッセージが存在するならば、メッセージを返す GetMessage 関数。me は自端末を示す識別子である。

```
(1) //初期化
myRole=prevRole=None;
serverList=[];
forwardingServerInfo=NULL;
set(myVncServerInfo) //自端末 VNC サーバ情報を収集

(2) //ユーザより役割を変更された場合
IF (myRole!=prevRole) THEN
//解除ルールを処理
SWITCH prevRole
CASE "sender"
Broadcast(type="disconnect",
           info=myVncServerInfo)
CASE "receiver"
CloseAllViewer()
CASE "repeater"
CloseAllViewer();
Broadcast({type="disconnect",
           info=myVncServerInfo,
           fwd=forwardingServerInfo})
END SWITCH
//役割変更を通知
prevRole=myRole;
Broadcast({type="update role",
           info=myVncServerInfo,
           role=myRole})
END IF

(3) //更新・切断通知を受け取った場合
msg=GetMessage();
SWITCH msg.type
CASE "update role"
Update(serverList,msg);
//接続ルールを処理
SWITCH myRole //自端末の役割
CASE "receiver"
IF ContainRepeater(serverList)
THEN
Connect(
    GetRepeater(serverList),me)
ELSE-IF ContainSender(serverList)
THEN
Connect(
    GetSender(serverList),me)
END IF
CASE "repeater"
IF ContainSender(serverList)
THEN
Connect(
```

```

        GetSender(serverList),me)
    END IF
    IF ContainRepeater(serverList)
    THEN
        Connect(
            GetRepeater(serverList),me);
        forwardingServerInfo=
            GetSender(serverList);
    END IF
    END SWITCH

CASE "disconnect"
    CloseViewer(msg.info);
    IF (msg.fwd!=NULL)
        CloseViewer(msg.fwd)
    END IF
END SWITCH
END IF

GOTO (2)

```

**3・6 議長ツールの提供** さらに、付加的なツールとして議長ツールについて述べる。会議のような場においては、発表者の交代などを取り仕切る議長が存在する。この時、会議の進行を円滑に行うために、議長が共有端末や発表者の端末に対して転送開始・停止・切替を管理したいという要望が考えられる。そこで議長のような管理者のために、端末間の接続関係を表示し、各端末の役割を一元的に設定可能なツールがあると利便性の向上が見込まれる。

**3・7 多様な接続状況の実現** ここまで述べてきた VNC セッションの制御と、役割決めによる自動接続に基づき、2 節で分類した共有、分散、遠隔ディスプレイでの表示が実現可能であることを図 6 の例で示す。この例では遠隔ディスプレイでの表示として、分散、共有ディスプレイでの表示も含んでいる。図中のように点線で区切られた各拠点の端末に役割を与える。各拠点間では Repeater 端末しか存在を知りえないとする。そして接続ルールを適用した結果、Repeater 端末は実線の矢印の接続を、Receiver

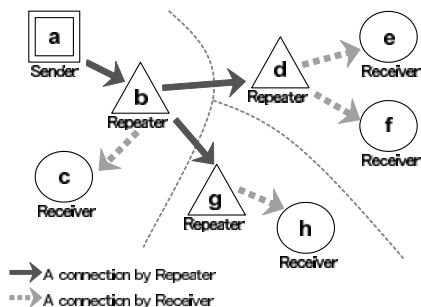


図 6 接続ルールの適用例

Fig. 6. Application of a connection rule.

端末は点線の矢印の接続をそれぞれ確立し、画面共有が開始される。

**4. 多様な接続を実現する画面共有システムの構築**

**4・1 役割カードを用いた実装** 以上の設計に基づき、多様な接続を実現する画面共有システム Cvnc を開発した。ネットワーク経由での画面転送には各プラットフォーム用の既存の VNC サーバを利用し、VNC セッションの制御部分は機器間連携ミドルウェア cogma<sup>(6)</sup> 上に Java を用いて実装した。cogma は、河口らが開発した動的なネットワークにも対応する機器間連携ミドルウェアである。Cvnc の実装にあたっては cogma のコンポーネントの一つである EmbCodget の枠組みを利用することで、ネットワーク内の Cvnc 起動端末の発見と IP アドレスなど VNC セッション開始に要する情報を動的に自動収集する。なお、Repeater をインターネット経由で利用するために、VNC セッションを行うのに必要な相手端末の情報を静的にユーザが定義できるようにもしてある。

役割決めインタフェースの実装には、非接触 IC カード Felica<sup>(9)</sup> を用いた。カード型を選択した理由は、GUI が不要なので共有画面の妨げにならないため、カードを置くだけという手軽で直感的な操作を実現できるためである。画面転送の役割カードを Sender カード、表示の役割カードを Receiver カードと定めた。端末を Sender, Receiver にするには対応するカードをリーダ・ライタ上に置くだけである。Repeater は共有端末という性質から固定的な利用方法になると考え、カードではなく GUI により RepeaterMode オプションを指定できるようにした。

Felica リーダ・ライタには Sony RC-S310 を使い、状態監視を行う Java Native Interface ライブラリの実装には FelicaSDK Lite と Microsoft Visual C++ を用いた。

各端末上の Cvnc システム構成図を図 7 に示す。Cvnc は図中のように以下の 3 つのコンポーネントから構成され、ネットワーク内の Cvnc 間で連携動作を行う。

- CvncServer: ローカルの VNC サーバへの接続に必要な IP アドレス・ポート番号などを他の PC 上の Cvnc-Selector に通知するコンポーネント
- CvncViewer: CvncSelector より指示された VNC サーバと接続する VNC ビューワ

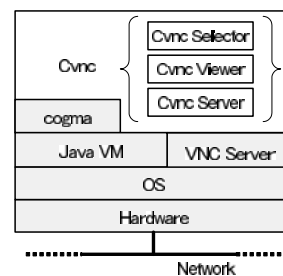


図 7 システム構成図

Fig. 7. The structure of Cvnc system.

- CvncSelector : cogma を通じてネットワーク上の他のコンポーネントの存在と情報を管理するコンポーネント、役割カードの状態も監視

RepeaterMode の CvncServer と CvncViewer は、Repeater であることも通知する。

例として、2 台の端末にそれぞれ Sender カードと Receiver カードを置くことで画面転送を開始するまでのシーケンスを図 8 に示す。図中では (1) ユーザによって役割カードが置かれたことを検知し、VNC セッションを制御している。なお (5) どちらかの役割カードが離されるとセッションが閉じる。実際に動作中のスクリーンショットと役割カードを図 9 に示す。このようにユーザは役割カードを Felica リーダ・ライタ上に置くだけで画面転送を開始できる。

#### 4.2 利用シーン

4.2.1 分散ディスプレイ表示 具体的な利用シーンとして、ミーティングテーブルを囲み、持ち寄ったノート PC 間で画面共有を行なうことを挙げる。この場合 (1) 参加者は、それぞれの PC に取り付けた Felica リーダ・ラ

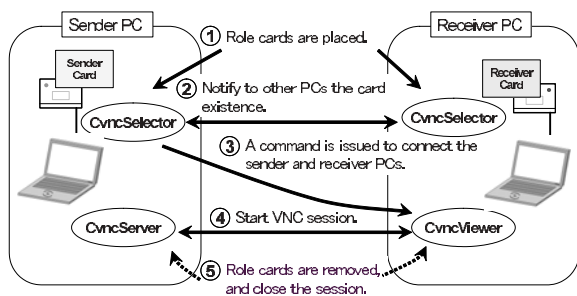


図 8 役割カードを用いた接続シーケンス

Fig. 8. Session establishment flow with role cards.

イタ上に役割カードを載せ (2) 自己の画面を提示したい参加者間で Sender カードの受け渡しを行う。(3) さらに、途中からの参加者もネットワークに接続後、Receiver カードを置くだけで、他の参加者の端末に影響を与えることなく、画面共有に加わることができる。そのような、分散ディスプレイ表示の様子を図 10 に示す。図中では、3 台のノート PC を用いた。動作の様子は cogma project のページ<sup>(11)</sup>を参照されたい。

4.2.2 共用・遠隔ディスプレイ表示 共用・遠隔ディスプレイ表示の利用シーンとして、H.323<sup>(10)</sup> 上のテレビ電話システムと Cvnc 併用した遠隔ディスプレイでの表示による画面共有を行った。その接続構成を図 11 に示す。現時点の Cvnc は、同一ネットワーク内の端末しか発見できないので、部屋間のネットワークを SoftEther を利用して仮想的に同一セグメントにした。各 PC の OS は WindowsXP である。この場合 (1) 参加者は、それぞれ Sender

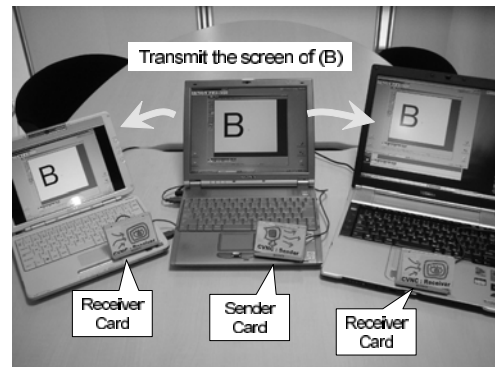


図 10 分散ディスプレイの利用シーン

Fig. 10. Example of using Cvnc with multiple PCs.

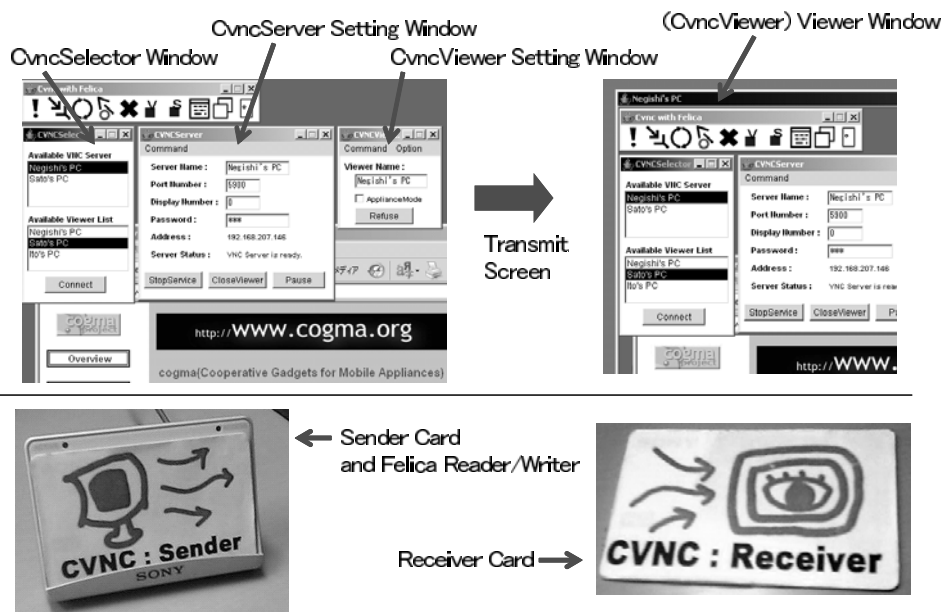


図 9 Cvnc スクリーンショットと役割カードの写真

Fig. 9. Screenshots of Cvnc and photos of role cards.

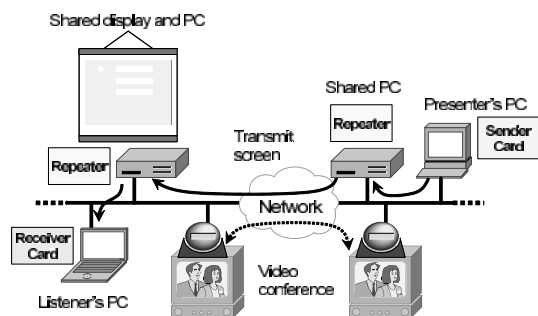


図 11 テレビ会議との併用シーン

Fig. 11. Example of using Cvnc and videoconference.

カードと Receiver カードを持ち、画面を提示したい側が Sender カードを使用する。(2) Repeater には共用ディスプレイが接続されているので、複数人の閲覧も可能であるが、Receiver 端末を追加することで、手元のノート PC でも閲覧できる。

他にも有用な利用シーンの例として、共有ディスプレイを多数使った学会でのプレゼンテーションにより、後部席の人でも資料画面を見やすくすることが挙げられる。

## 5. 評価

**5.1 画面切替時における操作量と時間の比較** 提案システムでは、役割決めインタフェースにより、複数端末間でのスムーズな画面切替を実現している。実際に、1 台の Sender 端末と 3 台の Receiver 端末による分散ディスプレイ表示において、画面転送を行う端末の切替にかかるユーザの操作量を VNC 単体による実現と Cvnc とで比較する。また、一連の操作にかかる時間も計測し、比較する。

**5.1.1 タスク内容** 図 12 にタスクの流れを示す。このタスクでは (2) 図中の A 端末の画面を B, C, D 端末に表示させ (3) 一度、接続を解除し (4) B 端末の画面を A, C, D 端末に表示させる。

今回は、VNC 単体による実現のために、あらかじめ各端末に Liseten Mode の VNC ビューワを起動し、各端末上の VNC サーバと接続するのに必要な IP アドレスなどの情報を事前に調査済みであるとする。Cvnc の場合においても、すでに各端末上で起動済みとする。経過時間は、最初の (2) の操作開始から (4) における全ての VNC セッションが確立するまでを、それぞれ 3 回計測した。

**5.1.2 比較結果** 操作に要する操作内容を表 1 に示す。結果より、切替操作である (3)(4) において、VNC 単体の利用では 4 ステップの作業があるのに対して、Cvnc 利用時では 3 ステップで済む。このように、通常の VNC 操作における個別の再接続作業を、Cvnc ではシステムが他端末の役割に応じて自動的にを行うため、ユーザの操作量を軽減し、スムーズな画面切替を実現できる。

一般に、転送元端末 1 台に対して表示先端末数を  $n$  台とすると、画面切替タスクは、VNC 単体利用では 1 回の解除作業と  $n$  回の接続作業が必要となるが、Cvnc では表示

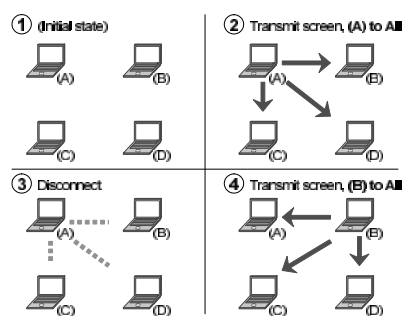


図 12 画面切替タスク

Fig. 12. Task of changing a screen sender.

表 1 ユーザ操作数の比較

Table 1. Comparison of user's operations.

Step	VNC only	Cvnc
(1)	-	-
(2)	(A, Add New Client to B) (A, Add New Client to C) (A, Add New Client to D)	(A, Place a Sender Card) (B, Place a Receiver Card) (C, Place a Receiver Card) (D, Place a Receiver Card)
(3)	(A, Kill All Clients)	(A, Remove a Sender Card)
(4)	(B, Add New Client to A) (B, Add New Client to C) (B, Add New Client to D)	(B, Remove a Receiver Card) (B, Place a Sender Card)

表 2 平均所要時間の比較

Table 2. Comparison of the average time required.

	VNC only	Cvnc
average time	77.8sec	31.1sec

端末数に関係なく、2 端末の役割入れ替え作業で完了する。図 6 のような遠隔・共用ディスプレイにおいても、図中の a 端末を Receiver に、e 端末を Sender に切り替える場合、既存システムの接続変更ではより複雑な 4 回の再接続操作を要するが、Cvnc では接続関係に関係なく、2 端末に対する操作で完了する。

また、一連の操作に要した平均所要時間を表 2 に示す。結果より、VNC 単体での利用に比べて、Cvnc の方では約半分の時間で操作を完了できる。これは、Cvnc は表 1 中の 1 ステップにおいて、役割カードをリーダー・ライタ上に置くだけに対して、VNC 単体の場合は接続先端末の IP アドレスを入力する作業が必要となるためである。このように個々の操作に関しても、Cvnc の方が少ない操作量で接続を指示できる。

これらの結果より、本手法は切替操作の円滑化に有効であると言える。

**5.2 Repeater による負荷分散性能の測定** 会議やディスカッションの場では、プレゼンテーション用アプリケーションなどを動作させながら画面を提示するので、Sender 端末にかかる通信による負荷を可能な限り低減させたい。そのため、提案システムでは Repeater を中継局として、他の Receiver 端末は Repeater から画面を受信する

ようにしている。この設定は、同一ネットワーク内に Repeater の役割を与えた PC を設置するだけで自動的に完了する。Sender と Receiver の使用方法に変更点はない。

実際に、どの程度、負荷を分散可能であるかを検証するため、多数の Receiver 端末を画面共有に参加させた場合における、Repeater 利用時と非利用時の Sender 端末にかかる負荷を比較した。

5・2・1 実験方法 会議における利用を想定し、

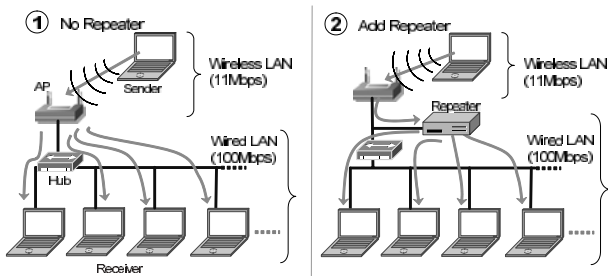


図 13 負荷測定実験のネットワーク構成

Fig. 13. Network composition of the measurement experiment.

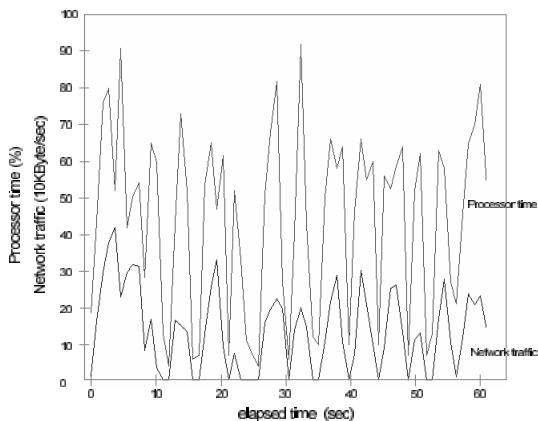


図 14 Sender にかかる負荷の測定結果 (Repeater なし)

Fig. 14. Load on a sender (without a repeater).

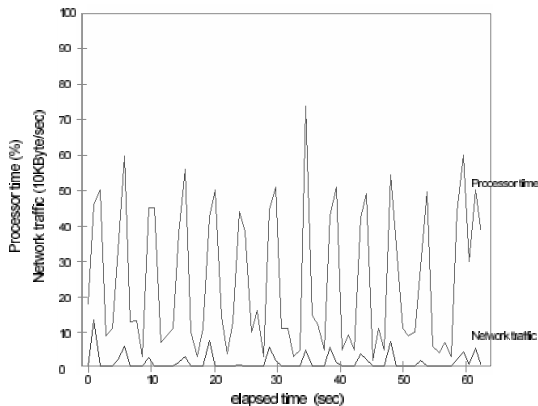


図 15 Sender にかかる負荷の測定結果 (Repeater 利用時)

Fig. 15. Load on a sender (with a repeater).

Sender 端末として使用したノート PC は 11Mbps 無線 LAN (IEEE802.11b) 経由で通信し、CPU は PentiumM(1.0GHz)、解像度は一般的な 1024 × 768 ドットである。VNC サーバには Ultr@VNC (v1.0.0RC18) を使用した。同時に Microsoft PowerPoint2002 を Sender 上で動作させた。Repeater と各 Receiver の通信環境は 100Mbps 有線 LAN を用意した。全ての端末の OS は WindowsXP である。実験手順は以下の通りである。接続構成の詳細を図 13 に示す。

- (1) Sender 端末 1 台と Receiver 端末 10 台を用意し、Sender 上で、5 秒おきに PowerPoint のスライドを次に進める。この時の CPU 使用率 (%) とネットワーク通信量 (Byte/sec) をパフォーマンスモニターより 1 秒おきに測定する。測定時間は 1 分である。
- (2) Repeater を 1 台追加し、Sender に対して (1) と同様の測定を行う。

5・2・2 測定結果 測定結果を図 14 と図 15 に示す。

結果より、Repeater を中継局とすることで、Sender 端末にかかる負荷を CPU 使用率において約 54%、ネットワーク通信量において約 9%まで軽減できていることが分かる。

6. 考 察

6・1 役割カードの利点 本稿で提案した役割決めインタフェースにより、手軽な接続指示を実現することができた。その実装方法には多様な方法が考えられる。例えば、GUI にて「転送ボタン」と「表示ボタン」だけの 2 種類のボタンをユーザに提供したり、F7 キーと F8 キーをショートカットとして利用する方法などが挙げられる。本稿では役割カードを作成したが、Felica リーダ・ライタを利用できない環境ではそのような実装の検討も必要である。一方、役割カードが持つ物理的制約による利点もいくつか存在する。例えば、学会や会議などにおける発表者は基本的に一人であるので、役割カードの枚数を制限することにより画面転送可能な端末数を制限できる。また、役割カードを持っている人しか画面共有に参加できないという制限も設けられる。

6・2 グループ分け 現時点での Cvnc は閉じられたネットワーク内での利用を想定しており、Sender カードを複数の端末上に置いた場合、複数の VNC ビューワのウィンドウが開かれる。このような利用方法とは別に、ネットワーク内に別々の画面共有グループを設けたい場合も想定される。そこで、役割決め時に通知する情報にグループ ID を加える対応策が考えられる。役割カードにグループ ID を付与する方法は、会議室の入り口付近に参加受付用の端末を設置しておき、そこにカードをかざすことでその会議で使用するグループ ID を書き込む。もしくは、議長ツールにて端末のグループを直接作成する方法も挙げられる。

6・3 スケーラビリティ 遠隔ディスプレイでの表示において共用端末を中継局として利用する場合、回線の帯域制限などにより全てのの中継局間を直接結ばずに、マルチ



ホップで多段接続する可能性がある。このような場合でも、3・4・1 節の接続ルールで対応可能である。

一方、分散ディスプレイでの表示は全 VNC 接続が転送端末に集中してしまうので、ネットワークが非常に混雑してしまうことが想定される。11Mbps の無線 LAN を利用する場合は、表示端末数も 3,4 台が快適な利用の限界である。そこで、分散ディスプレイでの表示時に表示端末の数が一定台数を超えると、自動的に一部の表示端末を中継局にする解決策が考えられる。すなわち、自動的にマルチホップでの接続を構築することで、通信負担の分散化が期待できる。

## 7. まとめと今後の課題

本稿では会議やディスカッションでの利用を想定した、共有、分散、遠隔ディスプレイでの表示に対応するデスクトップ画面共有システム Cvnc の構築について述べた。そのために、VNC セッションの制御と役割決めインタフェースを提案した。特徴は、ユーザは接続相手の IP アドレスなどを事前に調査することなく、また端末間の接続関係を気にすることなく、各端末に役割カードを置くだけで画面共有に参加できることである。それらにより、複数端末間での転送画面の切替作業の簡易化、自動化を実現できた。

今後は、考察に挙げたグループ ID 機能とスケラビリティ向上の導入を検討したい。他にも、システムの初期設定を簡易化することで一般ユーザが導入しやすいように改良し、議長ツールの整備を行うことで、システムの完成度をより高めたい。さらには、プロジェクトなど共有ディスプレイなどに接続可能な小型ディスプレイサーバの開発を行う予定である。

(平成 17 年 3 月 16 日受付, 平成 17 年 9 月 12 日再受付)

## 文 献

- (1) 三浦元喜・志築文太郎・田中二郎:「P2P 技術を適用した画面転送・遠隔操作システムの開発」, 情報処理学論誌, Vol.45, No.1 (2004)
- (2) Alpha Systems Inc: MultiVNC (<http://www.alpha.co.jp/business/products/multivnc/>)
- (3) 綾塚祐二・厩本純一: tranStick:「空間を越えて仮想的に繋がったメディア」, WISS 2003, pp.93-102 (2003)
- (4) 綾塚祐二・松下伸行・厩本純一: Gaze-Link:「実世界指向ユーザインタフェースにおける「見ているものに接続する」というメタファ」, 情報処理学論誌, Vol.42, No.6 (2001)
- (5) 岩崎陽平・河口信夫・稲垣康善: Touch-and-Connect:「無線機器間の接続手法」, WISS 2001, pp.201-202 (2001)
- (6) 河口信夫・稲垣康善: cogma:「動的ネットワーク環境における組み込み機器間の連携用ミドルウェア」, 情報処理学会コンピュータシステム・シンポジウム, pp.1-8 (2001)
- (7) T. Richardson, Q.S. Fraser, K.R. Wood, and A. Hopper: Virtual Network Computing, IEEE Internet Computing Vol.2, No.1 (1998)
- (8) T. Richardson, RealVNC Ltd (formerly of Olivetti Research Ltd/AT&T Labs Cambridge): The RFB Protocol Version3.8, (2004), (<http://www.realvnc.com/documentation.html>)
- (9) Sony Corporation: Felica (<http://www.sony.co.jp/Products/felica/>)
- (10) International Telecommunications Union: Recommendation H.323

(<http://www.itu.int/home/index.html>)

(11) Cvnc Demo Video:

(<http://www.cogma.org/press/cvnc/index.html>)

根 岸 佑 也 (非会員) 1982 年生。2005 年 3 月名古屋大学工学部電気電子・情報工学科卒業。同年 4 月より名古屋大学大学院情報科学研究科博士課程前期課程情報システム学専攻入学, 現在に至る。主として, コピキタス・コミュニケーションと機器間連携に関する研究に従事。情報処理学会会員。



河 口 信 夫 (正員) 1990 年名古屋大学工学部電気電子工学科卒業。1995 年同大学大学院情報工学専攻博士課程了。同年同大学工学部助手。同大学講師, 助教授を経て, 2002 年より同大学情報連携基盤センター助教授。モバイルコミュニケーション, マルチモーダルユーザインタフェース, コピキタスコンピューティングの研究に従事。2004 年より大学発ベンチャー企業(有) ユビグラフ取締役兼務。工博。ACM, IEEE, 情報処理学会, ソフトウェア科学会, 人工知能学会, 日本音響学会各会員。

