

有線 LAN 上の PC 画面配信システム TreeVNC の改良

伊波 立樹¹ 河野 真治¹

概要：授業やゼミ等で、それぞれが PC 端末を持っている場合には、PC の機能を活かしたコミュニケーションが可能である。教員が操作する画面をそのまま学生に配信したり、ゼミなどで、発表する学生の画面を切り替えたりすることを可能にしたい。画面配信システム TreeVNC は参加したクライアントをバイナリツリー状に接続し、配信コストを分散させる仕組みを取っている。そのため、多人数が参加しても処理性能が下がらない。また、ツリーのルートが参照している VNC サーバーを変更することで、ケーブルの差し替えなしに画面の切替が可能となる。今研究では TreeVNC の改良として、WAN への対応、マルチディスプレイへの対応を行った。

キーワード：

1. 研究背景と目的

授業やゼミ等で、それぞれが PC 端末を持っている場合には、PC の機能を活かしたコミュニケーションが可能である。

通常の授業では先生の用意した資料、PC 画面を見ながら授業が進むことが多い。

ゼミでは発表者を切り替えながら発表を行う。

これらの画面を表示するためにプロジェクターが使用されている。しかし、プロジェクタでは通常の授業の際、参加者はプロジェクタを見ており、手元の PC をあまり使用できない。更に手元の PC を使う際はプロジェクタと PC を行き来するため、目に負担がかかってしまう。またゼミの際には発表者を切り替えるたびにプロジェクタにケーブルを差し替える必要がある。ケーブルの差し替えの際に発表者の PC によってアダプターの種類や解

像度の設定によって正常に PC 画面を表示できない場合もある。

画面配信システム TreeVNC[1][2] は参加者をバイナリツリー状に接続し、配信コストをクライアントにバランスさせる仕組みになっている。そのため、授業で先生の画面を表示する際、多人数の生徒が参加しても処理性能が下がらない。

また、ツリーのルートが参照している VNC サーバーを変更することで、共有する画面の切替が可能となる。これはゼミの際に発表者の画面切り替えを円滑に行うための機能で、プロジェクターなどのケーブルの差し替えの手間を省くことが出来る。

本研究では WAN、マルチディスプレイへの対応を行った。WAN への対応として、新しい接続方法を提案し、実装を行った。また、マルチディスプレイへの対応としては配信する際に、配信するディスプレイ情報を取得し、配信を行うことで、対応した。

¹ 琉球大学工学部情報工学科

2. 画面配信システム TreeVNC

2.1 VNC について

VNC(Virtual Network Computing) は、RFB プロトコルを用いて遠隔操作を行うリモートデスクトップソフトウェアである。VNC はサーバー側とクライアント (ビューア) 側に分かれている。サーバを起動し、クライアントがサーバに接続を行い遠隔操作を可能とする。

2.2 RFB プロトコル

RFB(remote frame buffer) プロトコル [3] とは、自身の画面を送信し、ネットワーク越しに他者の画面に表示するプロトコルである。ユーザが居る側を RFB クライアント側と呼び、Framebuffer への更新が行われる側は RFB サーバと呼ぶ。Framebuffer とは、メモリ上に置かれた画像データのことである。RFB プロトコルでは、最初にプロトコルバージョンの確認や認証が行われる。その後、クライアントに向けて Framebuffer の大きさやデスクトップに付けられた名前などが含まれている初期メッセージが送信される。RFB サーバ側は Framebuffer の更新が行われるたびに、RFB クライアントに対して Framebuffer の変更部分だけを送信する。更に RFB クライアントの FramebufferUpdateRequest が来るとそれに答え返信する。RFB プロトコルは、描画データに使われるエンコードが多数用意されており、また独自のエンコードを実装することもできるプロトコルである。

2.3 多人数で VNC を使用する時の問題点

VNC を使用すればクライアント側にサーバー側の画面を表示することが可能である。しかし、多人数のクライアントが1つのサーバーに接続してしまうと処理性能が落ちてしまうという問題点がある。

また、ゼミ等の発表で画面配信者が頻繁に切り替わる場合、配信者が替わる度にアプリケーションを終了し、接続をし直さないといけないという問題がある。

2.4 TreeVNC の構造

TreeVNC は Java を用いて作成された TightVNC(Tight Virtual Network Computing)[4] を元に作成されている。

TreeVNC は クライアント同士を接続させ、画面描画のデータを受け取ったクライアントが次のクライアントにデータを流す方式を取っている。また、サーバへ接続しに来たクライアントをバイナリツリー状に接続する (図 1)。バイナリツリー状に接続することで、 N 台のクライアントが接続しに来た場合、画面配信の画像データをコピーする回数は従来の VNC ではサーバー側で N 回する必要があるが、TreeVNC では各ノードが 2 回ずつコピーするだけで済む。

バイナリツリーのルートのノードを Root Node と呼ぶ。Root Node は子ノードにデータを流す機能に加え、各ノードの管理と VNC サーバから流れてきた画像データの管理を行う。

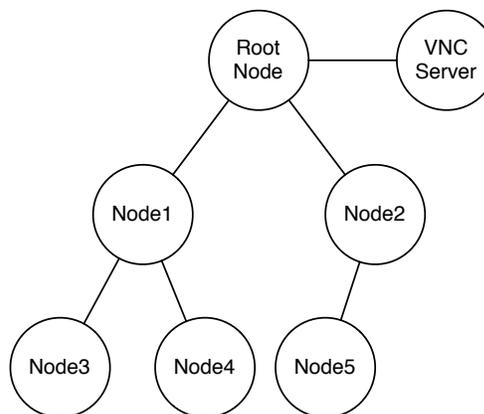


図 1 構成される木構造

2.5 node 間で行われるメッセージ通信

RFB プロトコルで提供されているメッセージに加え、TreeVNC 独自のメッセージを使用している。TreeVNC で使用されるメッセージの一覧を表 1 に示す。

図 2 は TreeVNC で Node が Root Node に接続し、画像データを受信するまでのメッセージ通

表 1 通信経路とメッセージ一覧

通信経路	message	説明
send direct message (child to root)	FIND_ROOT	TreeVNC 接続時に root を探す。
	WHERE_TO_CONNECT	接続先を root に開く。
	LOST_CHILD	子 node の切断を root に知らせる。
send direct message (root to child)	FIND_ROOT_REPLY	FIND_ROOT への返信。
	CONNECT_TO_AS_LEADER CONNECT	左子 node として接続する。接続先の node が含まれている。 右子 node として接続する。接続先の node が含まれている。
message down tree (root to child)	FRAMEBUFFER_UPDATE	画像データ。EncodingType を持っている。
	CHECK_DELAY	通信の遅延を測定する。
message up tree (child to root)	CHECK_DELAY_REPLY	CHECK_DELAY への返信。
	SERVER_CHANGE_REQUEST	画面切り替え要求。
send message (root to VNCServer)	FRAMEBUFFER_UPDATE_REPLY	画像データの要求。
	SET_PIXEL_FORMAT	pixel 値の設定。
	SET_ENCODINGS	pixel データの encodeType の設定。
	KEY_EVENT	キーボードからのイベント。
	POINTER_EVENT	ポインタからのイベント。
send message (VNCServer to root)	CLIENT_CUT_TEXT	テキストのカットバッファを持った際の message。
	FRAMEBUFFER_UPDATE	画像データ。EncodingType を持っている。
	SET_COLOR_MAP_ENTRIES	指定されている pixel 値にマップする RGB 値。
	BELL	ビーブ音を鳴らす。
	SERVER_CUT_TEXT	サーバがテキストのカットバッファを持った際の message。

信の様子である。図 2 の手順として

- 接続を行う Node (以下 Client Node) は Multicast 通信で Root Node に対して FIND_ROOT を送信する (1:findRoot())
- Root Node が FIND_ROOT を受信し、FIND_ROOT_REPLY を送信する (2:findRootReply())
- Client Node 側で、どの Root Node に接続するかを選択するパネルが表示される
- Client Node はパネルで接続する Root Node を選択し、Root に対して接続先を要求する WHERE_TO_CONNECT を送信する (3:whereToConnect())
- 受信した Root Node は Client Node の接続先を CONNECT_TO で送信する (4:connectTo)
- Client Node は Root の指定した接続先に接続しに行く
- Root Node, Client Node 間の接続が確立後、Root Node から Client Node に対して定期的に画像データ FRAME_BUFFER_UPDATE を送信する (5:framebufferUpdate())
を行っている。

2.6 配信画面切り替え

ゼミでは発表者が順々に入れ替わる。発表者が入れ替わる度に共有する画面の切り替えが必要となる。ゼミを円滑に進めるために、画面の切り替えをスムーズに行いたい。

画面の共有にプロジェクタを使用する場合、発表者が変わる度にケーブルの抜き差しを行う必要

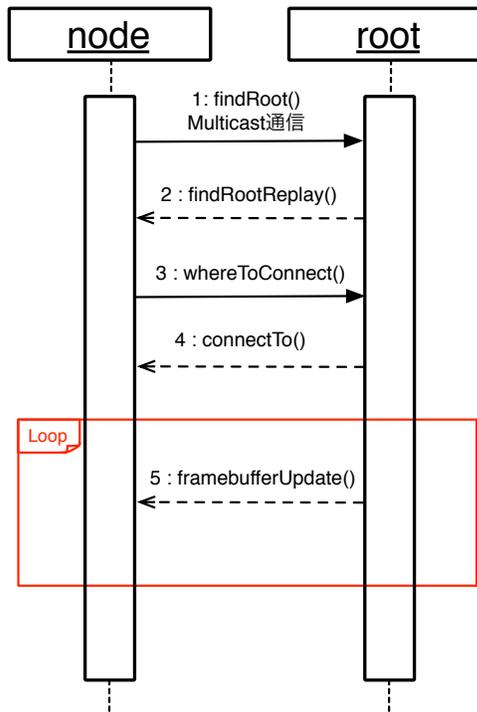


図 2 node 間で行われるメッセージ通信

がある。その際に、ディスプレイ解像度を設定し直す必要が出たり、接続不良が起こる等の煩わしい問題が生じることがある。

従来の VNC を使用する場合、画面の切り替えの度に一旦 VNC を終了し、発表者の VNCServer へと再接続を行う必要がある。

TreeVNC は、配信者の切り替えの度に生じる問題を解決している。TreeVNC を立ち上げることで、ケーブルを使用する必要なしに、各参加者の手元の PC に発表者の画面を共有することができる。画面の切り替えは、ユーザが VNCServer への再接続を行うことなく、ビューワの“Share Screen”ボタンを押すことによって、配信者の切り替えを行うことができる。

TreeVNC の Root Node は配信者の VNCServer と通信を行っている。VNCServer から画面データを受信し、そのデータの子 Node へと送信している。配信者切り替え時に“Share Screen”ボタンが押されると、Root Node は“Share Screen”ボタンを押したクライアントの VNC サーバー と通信を始める。そのため TreeVNC は配信者切り替

えの度に VNC を終了し、再接続する必要がない。

3. TreeVNC の新機能

3.1 QUALITY モードと SPEED モード

高解像度のまま拡大・縮小の処理を行うと、PC のスペックによっては描画処理に時間がかかってしまうことがある。配信者の画面をリアルタイムに取得するため、描画処理に時間のかからないモードを追加する。

画像描画処理には、高画質優先の QUALITY モードと描画速度優先の SPEED モードがある。今まで TreeVNC は QUALITY モードで使用していた。

今回、どちらのモードを使用するかをビューワから変更出来るようにした。これにより、描画処理の遅延を解決することができた。

3.2 マルチディスプレイ対応

画面配信側の PC がマルチディスプレイの場合、VNCServer からは複数の画面全体の画像データが送信されてしまう。

授業やゼミ等で TreeVNC を使用する場合、複数画面の表示は必要ない。そこで、画面を共有する際、ディスプレイを選択させ、画面共有を行う機能を追加した。

ディスプレイの情報は個々のクライアントでしか取得ができない。そのため、配信側は画面の切替を行う際に、ディスプレイを選択し、そのディスプレイの左上と右下の座標を取得する。その座標を Root Node への画面切り替えを要求する SERVER_CHANGE_REQUEST message に付加させる。Root Node は 配信側の VNCServer に画像データを要求する FRAMEBUFFER_UPDATE_REPLY message に送信された座標を付加する。VNC サーバーは要求された座標内の画像データを FRAMEBUFFER_UPDATE message で Root Node に送信する。これにより、一画面のみの表示が可能となる。

図 3 は Display1 のみを画面共有する例を示している。

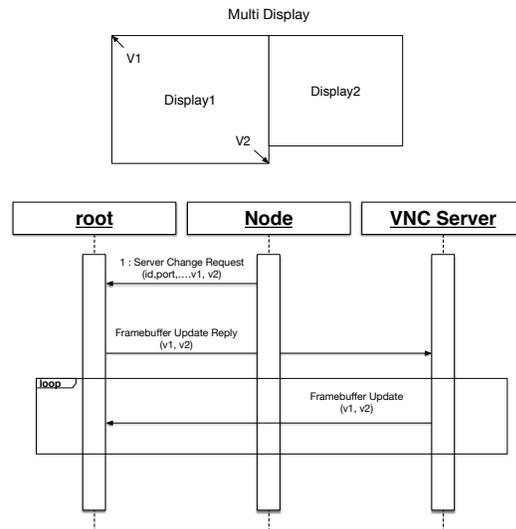


図 3 マルチディスプレイへの対応

3.3 無線 LAN への対応

授業で TreeVNC を使用する場合、有線を使用するか否かは学生によって違う。TreeVNC を有線・無線の両方からの接続に対応したい。

従来の TreeVNC は、クライアントの接続する木構造が単一であった。そのため、単一のネットワークインターフェースでしか使用することができなかった。

この問題を解決するために、図 4 の様に、ネットワークインターフェース別に 木構造を形成するように設計した。

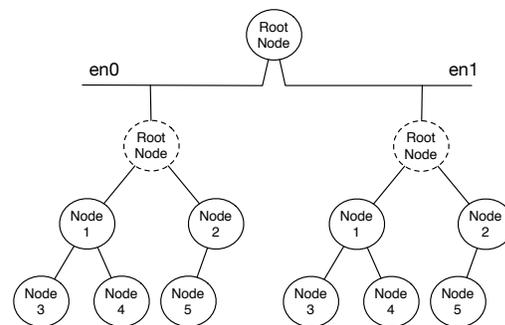


図 4 Multi Network Tree

TreeVNC は Root Node が TreeManager というオブジェクトを持っている。TreeManager は

TreeVNC の接続部分を管理している。TreeManager では木構造を管理する nodeList が生成される。この nodeList を元に、新しい Client Node の接続や、切断検出時の接続の切り替え等を行う。

Root Node の保持しているネットワークインタフェース毎に TreeManager を生成する様に変更した。新しい Client Node が接続してきた際、interfaces から Client Node のネットワークインタフェースと一致する TreeManager を取得する。その TreeManager に、Client Node 接続の処理を任せる。こうすることによって、TreeVNC を複数のネットワークインタフェース別に木構造を構成することができる。

3.4 WAN への対応

遠隔地からでもゼミや授業に参加できるよう、別ネットワークから TreeVNC への接続を可能にした。

図 5 に別ネットワークからの接続を示す。別ネットワークから TreeVNC に参加する際、直接配信側のネットワークの Root Node に接続を行う。この接続を Direct Connection と呼ぶ。

Direct Connection した Client Node はそのネットワークの Root Node になる。そのネットワークの他の Client Node はそのネットワークの Root Node に接続し、木構造を生成する。

配信側の Root Node は Direct Connection で接続された Root Node に対して Framebuffer Update で画像データを送信する。Framebuffer Update が送信された Root Node はそのネットワークの Client Node に対して Framebuffer Update を送信する。

これにより、別ネットワークでの画面共有が可能となる。

4. TreeVNC の評価

4.1 木の深さによる画像データ伝達の遅延

VNC サーバー から受信する画像データ、TreeVNC で扱われるメッセージ通信は構成された木を伝って伝達される。接続する人数が増える毎に木の段数は増えていく。そこで Root Node か

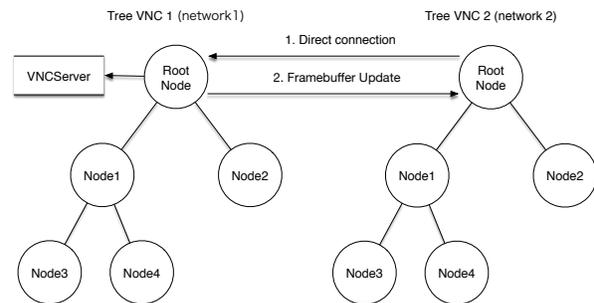


図 5 遠隔地 Node からの接続

ら木の末端の Client Node までの画像データ伝達の遅延を検証する実験を行った。

4.2 実験環境

授業を受講している学生が TreeVNC を使用した状態で実験を行った。TreeVNC には最大で 17 名が接続していた。

4.3 メッセージを使用した実測

TreeVNC を伝搬するメッセージに、CHECK_DELAY・CHECK_DELAY_REPLY を追加した。CHECK_DELAY は Root Node から末端の Client Node まで伝達するメッセージと画像データ、CHECK_DELAY_REPLY は各 Client Node から Root Node まで伝達するメッセージである。

CHECK_DELAY メッセージは送信時刻を付けて送信する。Root Node から CHECK_DELAY 送信し、末端の Client Node まで各 Node を伝いながら伝達して行く。

CHECK_DELAY_REPLY には CHECK_DELAY から受け取った送信時刻に画像データのサイズを付けて送信する。CHECK_DELAY を受け取った各 Client Node は CHECK_DELAY_REPLY を接続している親 Node に送信する。

CHECK_DELAY_REPLY を受け取った Root Node はメッセージと画像データの伝達にどれだけの時間がかかったかを計算する。データ計算方法を以下のソースコード 1 に記述する。この変数 “time” は CHECK_DELAY_REPLY に付いて

いる送信時刻である。

```
1 Long delay = System.currentTimeMillis()
   - time;
```

Code 1 遅延時間の計算方法

4.4 depth 毎の遅延結果

バイナリツリーで木を構成した場合、Node 数が 17 台だと深さが 4 となる。各木構造の階層毎に、画像データの伝搬にかかった時間を測定した。

図 6 は遅延の分布を示した散布図である。X 軸はメッセージ伝達にかかった秒数 (ms)、Y 軸は画像データのサイズ (Byte) である。

- 大体 1 秒以内
- 大容量の画像の送信の後の Delay が残っているため、容量が小さいところでも時間がかかる場合がある
 - Depth3 に極端に遅い場合がある → 1 つの node がネックになっている
 - 極端に遅いやつを下に持っていくアルゴリズムが必要 (これはまとめにも書く)

5. まとめ

本研究では画面配信システム TreeVNC をマルチディスプレイ、WAN に対応させた。

マルチディスプレイに対応したことで、配信者が配信したいディスプレイを選択し、画面配信することが可能となった。

WAN に対応することで別ネットワークにいるユーザーが TreeVNC に参加することが可能となった。

今後の課題として、画面切り替えの安定化、WAN での画面切り替え、ユーザビリティの向上、共有機能の追加を行う。

現在の TreeVNC では、share button を押すと、その時配信されている画面から、自動的に画面が切り替わってしまうという問題がある。それを防ぐために share button が押されるとその時の配信者に切り替え確認を行う処理を追加する。

今回追加した Direct Connection などの一部の

機能はコマンドラインオプションで指定する必要があるため、一般ユーザーでは操作するのが困難である。そこで、今までコマンドラインオプションで指定していた機能を ビューワ で操作するように変更を行う。

共有機能の追加としては、音声、講義中の質問・意見 等が挙げられる。

- 新機能の評価方法、評価
- 極端に遅いやつを下に持っていくアルゴリズム

参考文献

- [1] Miwa OSHIRO: 授業やゼミ向けの画面配信システム TreeVNC の拡張機能, 琉球大学工学部情報工学科平成 26 年度学位論文 (学士) (2014).
- [2] Yu TANINARI and Nobuyasu OSHIRO and Shinji KONO: VNC を用いた授業用画面共有システムの設計・開発, 情報処理学会 (2012).
- [3] Tristan Richardson: The RFB Protocol, <http://www.realvnc.com/docs/rfbproto.pdf>.
- [4] : TightVNC Software, <http://www.tightvnc.com>.

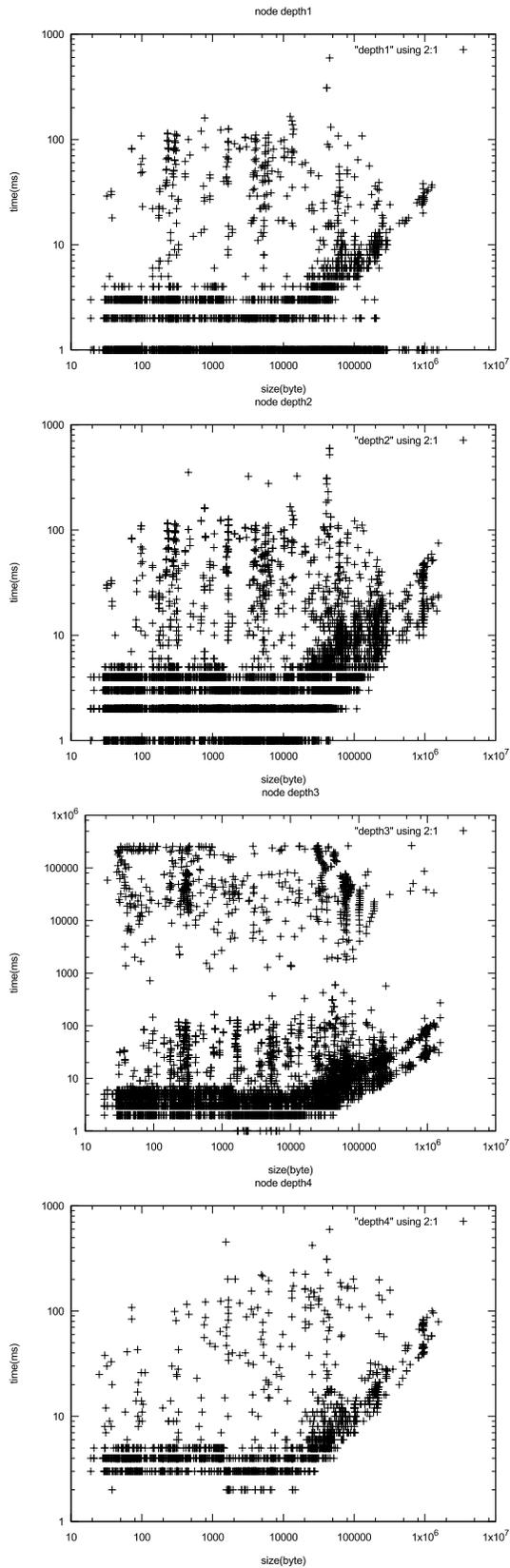


図 6 深さ毎のデータサイズと遅延の関係