

TCP を利用した分散ネットワーク環境のための 電子黒板システム

平原 貴行[†] 山之上 卓^{††}
安在 弘幸[†] 有田 五次郎^{†††}

本論文では、TCP/IP を用いた一般的な分散ネットワーク環境で利用できる電子黒板システムについて論じる。この電子黒板システムは、教師端末の画面の静止画像を LAN を用いて受講者端末に転送し表示するものである。本システムでは信頼性が高いがブロードキャストを行えない TCP により通信を行っていることから、転送時間の短縮を図るため、スイッチングハブを用い転送経路をツリー状に構成して並列かつ段階的に転送している。このシステムは TCP/IP を利用しているため一般的なパソコンネットワークを備える端末教室で利用可能なほか、インターネットによる遠隔授業への応用も可能である。

The Electronic Chalkboard System for Distributing Network Environment Using TCP

TAKAYUKI HIRAHARA,[†] TAKASHI YAMANOU,^{††} HIROYUKI ANZAI[†]
and ITSUJIROU ARITA^{†††}

In this paper, we discuss a electronic chalkboard system which is able to use in a typical distributed network environment. This is a system transmits a still image of a teacher's screen to a large number of student terminals through LAN. This system employs TCP for reliability. Therefore, in order to transmit image data in a short time using TCP, which uses point-to-point communication, the route of data transmission in this system is tree-shaped, and the data is transmitted in a parallel, step-by-step mode, using switching HUB. This system employs TCP/IP, therefore this system is able to use in a computer room which has a typical personal computer network system, and is able to apply to teleteaching with Internet, too.

1. はじめに

近年、急速な高性能化、低価格化を背景に、教育現場へのパソコンの普及が進んでおり、大学などにおいて、1人1台のパソコンまたは端末を備えた端末教室の設置が進んできている。さらに、ネットワークの高速化を受け、多数のパソコンやワークステーションなどを LAN で接続した分散環境の普及が進み、これを利用したさまざまな教育支援システムの研究、開発が行われている。

我々は、TCP/IP プロトコルに基づく分散ネットワーク環境で利用できる教育支援システムの開発を行っている^{6)~8)}。このシステムは、教師端末画面の静止画の一部を受講者端末に転送して再現、表示することにより、WYSIWIS (What You See Is What I See) 環境を実現する画面共有型のシステムである。これは、教師端末と各受講者端末が、特定のサーバを介さず、メッセージパッシングにより直接データを通信する、一種の P2P (Peer to Peer) 通信により実現されている。こうした画面共有ツールには、端末の遠隔操作やグループ内での共同作業といった、2台、あるいは比較的少数の端末を想定しているシステムもある。これに対し、我々のシステムは大学の端末教室のような、数十台以上の端末を接続した環境での一斉利用を想定している。

このシステムのねらいとして、(1) LAN やインターネットに接続された一般的なコンピュータネットワー

[†] 九州共立大学工学部
Faculty of Engineering, Kyushu Kyouritsu University

^{††} 九州工業大学情報科学センター
Information Science Center, Kyushu Institute of Technology

^{†††} 九州工業大学情報工学部
Faculty of Computer Science and Systems Engineering,
Kyushu Institute of Technology

クシステムに容易に導入でき、(2) 追加の機材の導入や特別な配線、設定などを行う必要なく、(3) UNIX, Windows, Macintosh といった OS や環境に依存せず導入、利用できる、といった点があげられる。

類似のシステムには、PC-SEMI (NEC) のように専用回線を用いて教師画面を受講者端末へ送信するものや、E-Chalk¹⁾ のようにネットワークに接続された黒板型入力装置に書かれた画像を受講者画面に転送、表示するもののように、専用の機材や配線を必要とするものがある。一方、こうした専用機材などを必要とせず、LAN やインターネットを通じて画像を転送し共有するものもある^{2)~5)}。我々の電子黒板システムはソフトウェアのみにより実現されており、さらに、インターネット標準プロトコルであり、一般的なパソコンネットワークにも多く利用されている TCP/IP を利用しているため、既存の教育用コンピュータネットワークシステムに容易に導入できる。

UNIX (X Window) や Windows NT の画面共有機能など、OS レベルの機能を利用する方法もあるが、この場合、システムが OS に依存し、汎用性が損なわれるという問題がある。たとえば、他の端末教室の環境へのシステムの移植や環境の変更 (機種更新など)、環境の異なる教室間での利用などに支障をきたすことが考えられる。我々のシステムは、こうした汎用性を考慮して一般的な TCP/IP により実現している。

一方、教師画面のイメージのようなサイズの大きい画像データを、可能な限り短時間で多数の端末に転送しなければならないという問題がある。データの一斉転送には一般的にブロードキャストやマルチキャストが用いられているが、これらに一般に用いられる UDP は転送データに対する信頼性が低く、動画像と異なりデータへの信頼性に対する要求がより厳しい静止画像の転送には適さない。また、TereDraw⁴⁾ の導入例があるマルチキャストでは、対応するルータを導入する必要性や設定作業などのコストの問題もある。こうした理由により、本システムは信頼性を保証している TCP により通信を行っているが、TCP は本来 1 対 1 通信向きのプロトコルであるため電子黒板のような放送システムではあまり利用されていない。そこで本システムでは、データの転送経路をツリー状に構成してデータを並列かつ段階的に転送することで転送時間の短縮を図っている。

教育用ネットワークツールとしては、X Window で動作する XTV²⁾、XMX³⁾ や、製品化されているものでは MS Windows 上で動作する Campus ESPer (Compaq) など、特定 OS 上で組まれているシステ

ムも多い。我々のシステムも、当初は特定 OS での利用を想定したものであった^{6),7)}。一方、実際の教育用コンピュータネットワークでは Windows や UNIX などのさまざまな環境が存在し、さらに、遠隔授業の場合、異なる環境が同時に利用されることが考えられる。我々の電子黒板システムでは、このような場合に対応するため、OS や環境を問わず動作するマルチプラットフォーム言語である Java を用いて開発を行っている。

本論文では Java 版を中心に電子黒板システムの構成と機能、実際のシステムに実装した場合の評価について述べる。本論文の構成は以下のとおりである。次の 2 章で電子黒板システムの概要、構成について説明する。3 章で電子黒板システムを実際の教育用パソコンネットワークに実装して評価を行った結果について述べ、4 章でまとめを述べる。

2. 電子黒板システムの概要

我々の開発している電子黒板システムは、教師端末の画面の静止画の一部を、既存のコンピュータネットワークを介しソフトウェアにより受講者端末へ転送して表示させるものである。教師画面を受講者に示す方法として、プロジェクタで投影する方法や専用機器を用いて別モニタなどに表示する方法などが多く用いられている。しかし、プロジェクタで投影する方法では遠くの席から画面が見えにくく、別モニタなどに表示する方法ではモニタや機材などを設置するスペースや配線などにかかるコストなどの問題がある。ソフトウェアを用いて画面を転送することで専用の機器・端末や配線を必要とせず、現在のシステムをほぼそのまま利用でき、導入にともなうコストを抑えることができる。特に我々の電子黒板システムは TCP/IP を利用しているため、一般的なコンピュータネットワークシステムで容易に導入できる。

我々は電子黒板システムをまず UNIX オペレーティングシステムおよび X ウィンドウを搭載するワークステーションシステム向けに開発し⁷⁾、次いで Microsoft Windows NT を搭載したパソコンによるネットワークシステム上で動作する仕様を開発した⁶⁾。前者はすでに実際の授業で用いられており、後者は実験運用中である。また、これらのシステムは特定の OS に依存しており、あらゆるシステムで利用可能なように、またこうしたシステム間を結んで行うことが想定される遠隔授業向けには OS などに依存しないシステムが必要であるため、新たに、マルチプラットフォーム言語である Java を用いてシステムの開発を行っている。以下、この Java により開発された電子黒板システムを

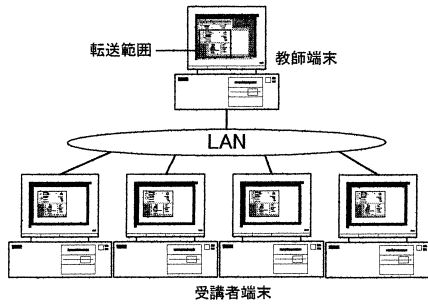


図1 電子黒板の構成

Fig. 1 Configuration of the electronic chalkboard.

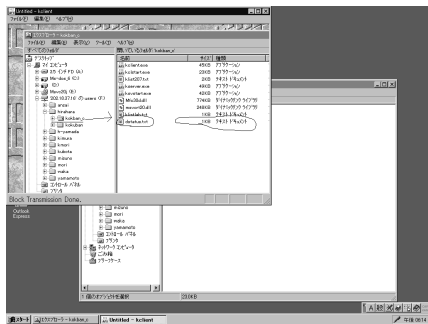


図2 送信された教師画面の例

Fig. 2 An example of the image of teacher's screen.

Java 版と呼ぶことにする。

2.1 構成と機能

我々の電子黒板システムは、1つの教師側プロセスと複数の受講者側プロセスからなり、それぞれの端末上で互いに独立に動作する(図1)。電子黒板システムの機能には、

- 教師画面の受講者端末への転送・表示、
- 教師画面上に書いた線画、テキストなどの受講者端末への転送・表示、

がある。

教師端末側プロセスは教師が画面上の範囲を指定する、画面上に線画などを書くなどすると、その範囲の静止画のイメージデータまたは描画コマンドを各受講者端末に送信し、受講者側プロセスでこれらを解析してその画面上に再現する(図2)。これらの教師端末から各受講者端末へのデータの送受信はメッセージパッシングによる並列プログラミングの手法により実現されている。

多数の端末へデータを一斉転送する手法として、ブロードキャストやマルチキャストがよく知られている。しかしながら、これらに用いられているUDPは信頼性が低く、転送データの欠落を引き起こす恐れがある。そのため、多少のデータ欠落が許される動画像や音声

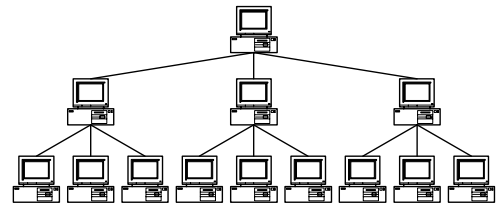


図3 ツリー状のデータ転送経路

Fig. 3 An data transmitting route that is tree-shaped.

データの転送には向いているが、本システムのような静止画像の転送には適さない。そこで本システムでは、エラー発生時に再送を行う、信頼性の高いTCPによりデータ転送を行っている。

しかし、TCPは基本的には1対1通信向きのプロトコルであり、さらに再送を行うのでUDPに比べ低速でブロードキャストには向かない。そのうえ、本システムは取り扱うデータ量も送信対象となる端末数も非常に大きいため、一斉転送をいかに実現するかが問題となる。そこで本システムは起動時にTCPのコネクションをツリー状に張り、複数のコネクションの通信を同時に行う並列転送によりデータ転送時間の短縮を図っている(図3)。

転送経路をツリー状に構成することにより、ツリーの構造にもよるが、教師端末に直接接続する端末はただか数台となり、すべての受講者端末が直接教師端末に接続する場合と比べ教師端末の負担が軽減される。また、ツリー構造により中継する受講者端末に負荷が分散されるため、多数の端末へのデータ転送にも対応できる。

従来のシェアードハブでは、回線を共有するため同時に複数コネクションの通信を行うことは不可能であったが、近年では高速LAN技術の1つとして広帯域の専用型ハブであるスイッチングハブ(10 Base TX, 100 Base TX^{10),11}など)が普及してきており、これを導入することでTCPを用いた並列通信などが可能となった。スイッチングハブはコストも低いため幅広く導入されている。

2.2 受講者接続状況管理サーバ

実際の授業においては、端末に途中からログインするか、または途中でログアウトする受講者が出ることもある。従来の電子黒板システム⁶⁾では起動時に転送経路を決定してそのまま固定しているため、後から電子黒板の利用を開始した人は黒板を再起動しない限り黒板の画面を受け取ることができないという問題があった。これに対処するため、教師側プロセスおよび各受講者側プロセスは、新たな接続を受け付けるため

のスレッドを生成し本体と平行して処理するようにした(図4)。これを接続受付スレッドと呼ぶことにする。

新たに黒板の使用を開始しようとする受講者端末は、接続先端末の接続受付スレッドに要求を送って接続を行うが、このとき経路構成などの問題から、どの端末に接続するべきかという問題が生じる。また、受講者が途中で電子黒板の使用を終了する場合、データの転送経路がツリー上のその受講者端末に対応するノードで断ち切れ、このノードに接続する受講者端末のノードやその子ノード、孫ノード以降の受講者がその後のデータを受信できなくなる恐れがある。

これらの問題に対処するため、Java版電子黒板では受講者の電子黒板への接続状況に応じたデータ転送経路の再構成を行うため、教師側、受講者側のいずれのプロセスからも独立した、受講者の接続状況を管理するプロセスを設けた。以降このプロセスを管理サーバ(Connection observer)(図4、図5、図6)と呼ぶことにする。管理サーバは受講者の接続状況を管理し、受講者が途中で電子黒板の使用を開始または終了する場合、必要に応じて転送経路の再構成を行って関係する端末に通知する役割を持つ。

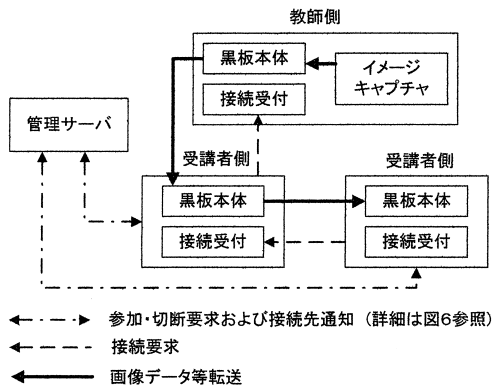


図4 Configuration of the electronic chalkboard by Java.

管理サーバは現在接続されている端末とその経路構成をテーブルに記録している。また転送経路の構成および切断する場合における再接続のためのルールを記録している。管理サーバは起動の際、この構成ルール(転送経路構成)の設定を対話的に行った後、受講者端末からの要求待ちの状態に入る。

新たに電子黒板の使用を開始しようとする受講者端末は、まず管理サーバに黒板参加要求信号を送る。管理サーバはこの信号を受信し、この端末をテーブルに追加するとともに、設定されている転送経路構成に基づいてこの受講者端末の接続先を決定して通知する。受講者端末はこれを受けて指定された端末の接続受付スレッドへ接続要求信号を送り、接続を試みる(図6(a))。

すでに接続している受講者端末が電子黒板の使用を終了する場合も同様に、まず管理サーバに切断要求信号を送る。管理サーバはこの信号を受信するとテーブルからこの端末を削除する。その際、この端末が子ノードを持っていた場合、これらの子ノードの再接続先を設定されている再接続ルールに基づいて決定し、これらのノードへ通知する。それぞれの子ノードはこの信号を受信して指定された端末の接続受付スレッド

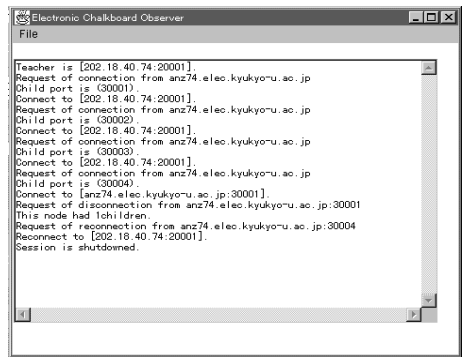
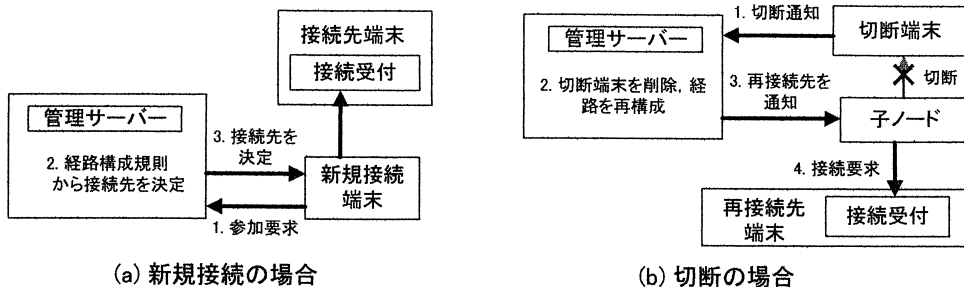


図5 管理サーバの画面
Fig. 5 A window of connection observer.



(a) 新規接続の場合

(b) 切断の場合

図6 管理サーバの動作

Fig. 6 A behavior of connection observer: new connection and disconnection.

へ接続要求信号を送り、再接続を試みる(図6(b)).

教師(あるいは管理者)は、図8(次章に記述)に示すようないくつかの転送経路構成のうち1つを管理サーバ起動時に選択できる。管理サーバはこの転送経路を構成ルールとしてテーブルに記録し、このルールに基づいて新たに加わるノードが接続すべきノードを決定する。一方削除については、現在は削除したノードに接続していたすべてのサブツリーを、その削除したノードの親ノードに接続するようにしている。

なお、管理サーバは独立した1つのプロセスであるため、任意の端末、たとえば教師端末などに置いて使用することができる。また、この管理サーバは受講者端末が黒板の使用を開始または終了する際の経路構成の管理にのみ用いられる。教師画面などのデータ転送は教師端末および各受講者端末間で直接行われ、管理サーバはこれらのデータ転送には関わっていない。

3. 実装および評価

本システムを実際のパソコンネットワークに導入して行った評価について述べる。評価実験を行った環境を図7に、実験に用いた端末などの仕様を表1に示す。この環境にJDK 1.0.2を用いて開発したJava版電子黒板システムを導入して実験を行った。またMicrosoft Visual C++ 2.0を用いて開発した同システム⁶⁾(以下C++版とする)との比較を行った。なお、Java版の実行はJIT(Just In Time)コンパイラを含むJDK 1.1.8のインタプリタを使用している。

ここでは図7、表1の環境において、いくつかの観点から転送完了時間を測定して評価を行った。ここでいう転送完了時間とは、教師端末が画面データの転送を開始してからすべての受講者端末がこのデータをすべて受信するまでの所要時間を示す。

3.1 転送完了時間と転送経路

本システムでは転送完了時間を短くするため転送経路をツリー状に構成して並列に転送を行っているが、この転送経路の構成にはさまざまなパターンが考えられ、TCPのような1対1通信ではこの経路構成が転送完了時間に大きく影響する^{8),9)}。そこでいくつかの経路構成を提案し、転送完了時間の比較を行う。

ここでは3種類の転送経路構成を提案する。1つは完全木(Complete Tree)(図8(a))と呼んでいるもので、葉以外のいずれのノードもすべて一定数の子ノードを持つツリー構造である。次はMBT(Minimum Broadcast Tree)^{12),13)}(図8(b))と呼ばれるツリー構造で、 n 個のノードに対し $O(\log n)$ の時間で転送を完了できるような構造である。最後は直列木(Linear

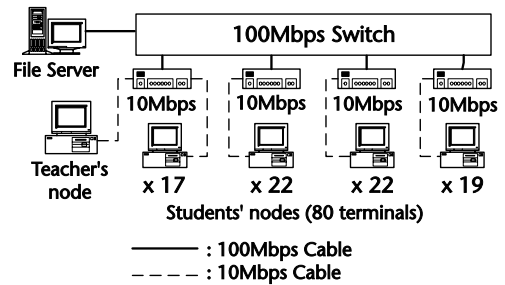


図7 実験で用いた端末教室の構成
Fig. 7 Configuration of a terminal room for experiments.

表1 実験で用いた環境
Table 1 Environment for experiments.

	教師側端末	受講者側端末
CPU	Pentium 166 MHz	486DX4 100 MHz
Memory	96 MB	27.3 MB
NIC	3Com 3C905B-TX 100 Mbps	Intel EtherExpress PRO 10 Mbps
OS	Windows NT Workstation 4.0	
Switching HUB	MACNICA Grand Junction Fastswitch 10/100	

NIC: Network Interface Card

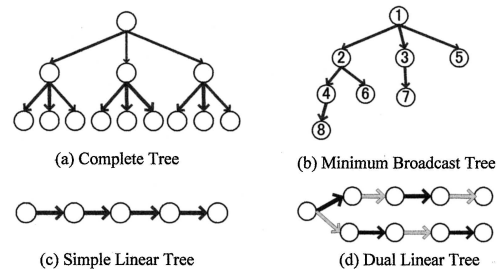


図8 種々の転送経路構成
Fig. 8 Examples of tree structures as data transmission routes.

Tree)と呼んでいるもので、どのノードもただか1個の子ノードしか持たないツリー構造である。なお、単純な直列木(図8(c))の場合、通信路が送受信を同時に行えない半二重の場合にデータ衝突の恐れがあるため、図8(d)のように根のみ子ノードを2個とする2列直列木(DLT: Dual Linear Tree)を導入した。この木構造では、2つの列におけるノード数の差はただか1個である。

以上の経路構成を電子黒板システムに実装して転送完了時間の測定を行った。測定を行った木構造は完全2分木(BIN)、完全3分木(TRI)、MBT、2列直列木(DLT)であり、転送する画像のサイズは640×480ピクセルである。しかし、このシステムでは、Java版およびC++版それぞれAPI(Application Programming Interface)が直接扱うことのできる画像データ形式を

そのまま利用しており、その形式は表 2 に示すようにそれぞれ異なる。そのため、同じ画像サイズながら実際に転送するデータのサイズは異なる。Java 版では 1 ピクセルを表す 4 Bytes 中、色表現に影響しないと考えられる 1 Byte を省略して転送しているが、これを考慮してもデータサイズは $640 \times 480 \times 3 = 921600$ Bytes となり、C++ 版 (実際はヘッダサイズが含まれる) の約 3 倍となる。

本実験では、転送する画像データを複数のデータブロックに分割して連続転送を行っている。端末台数を

表 2 実験で転送した画像データのサイズ (640 × 480 Pixels)
Table 2 A size of image data for experiments.

Format	Windows BMP	Java 画像形式
Bite/Pixel	1 (256 色)	4 (フルカラー)
Total Size (Bytes)	307200	1228000

20, 40, 78 (C++ 版は 75) 台として測定を行った結果を図 9 に示す。各グラフの横軸はデータブロック (分割) 数を示す。転送完了時間の実用性について考えると、Java 版は 78 台の場合で転送経路を完全 2 分木 (BIN) とした場合に、15 分割で最短の転送完了時間 (約 4.8 秒) を示している。C++ 版の 75 台における最短転送時間 (BIN, 40 分割で約 1.5 秒) に比べ約 3 倍の時間を要しているが、Java 版と C++ 版の転送データサイズの違いを考慮すると、JIT コンパイラを用いることで十分実用に耐えうる転送完了時間が得られたと考えられる。

C++ 版の測定結果には各経路構成における転送完了時間の理論値を併記している。これらは以下の式で表される。 t 個のデータブロックを連続して転送する場合、木の深さ i の完全 b 分木における転送完了時

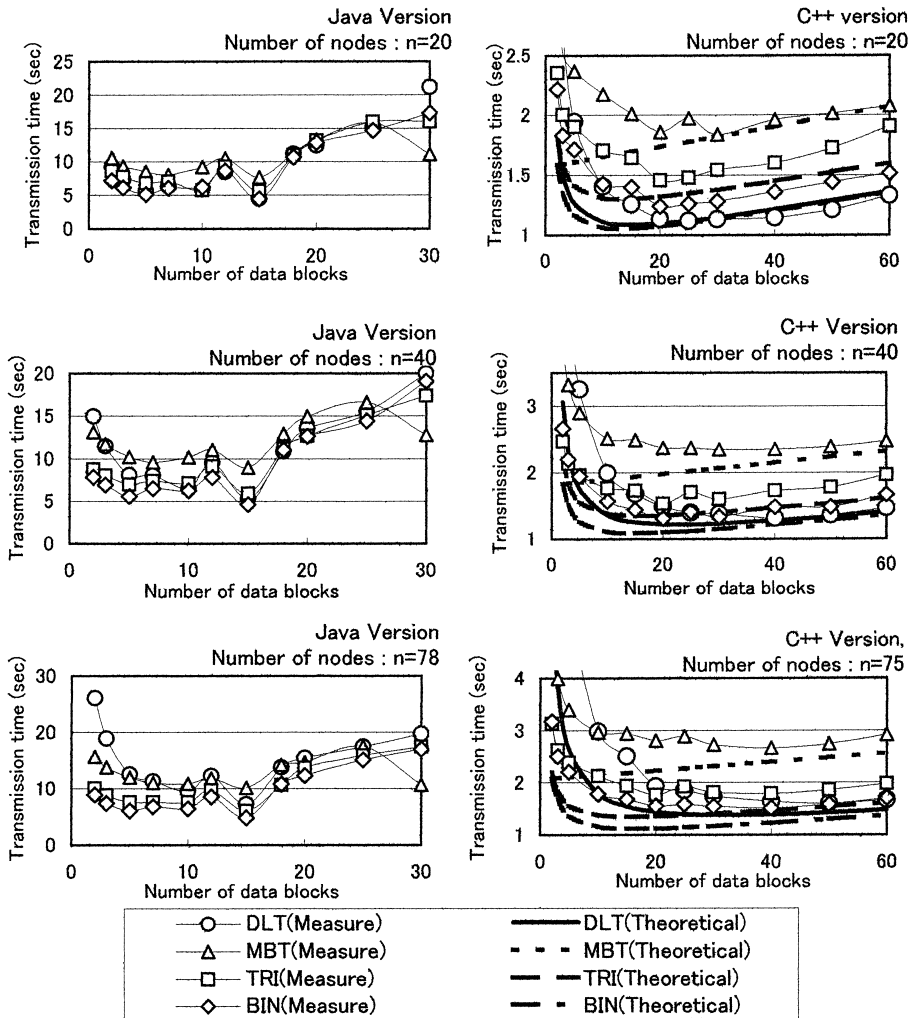


図 9 転送経路、分割数と転送完了時間

Fig. 9 Relation of transmission time, transmission route and number of data partitions.

間は

$$C_B(tb + b(i-1)) + d(t). \quad (1)$$

で示される, また, ノード数が n の場合, MBT では転送完了時間は

$$C_B t [\log_2 n] + d(t). \quad (2)$$

DLT では

$$C_B [2t + [(n-1)/2] - 1] + d(t). \quad (3)$$

となる. ここに C_B は単一コネクションにおける 1 データブロックの転送時間を示し, この実験の場合, 元の画像のデータサイズ s [Bytes], ハブおよびケーブルのバンド幅 w [bps], ハブの遅延時間 d_H [sec] より

$$C_B = \frac{8s}{wt} + d_H. \quad (4)$$

と推定される. 本実験においては $w = 10^7$, $d_H = 3 \times 10^{-5}$ となる. また, $d(t)$ は転送経路に依存しない, アプリケーション側での処理時間である. これは主に教師側ノードにおける, 画像データを分割するための前処理に必要な時間であると考えられ, 分割されたデータブロック数 t に対して

$$d(t) = a + ct. \quad (5)$$

の形となることが予想される. ここに, a, c はパソコンの性能やアプリケーションの処理に依存する任意定数であり, $d(t)$ を無視した理論値と実測値との差分から $a \approx 0.3400$, $c \approx 8.269 \times 10^{-3}$ と推定している.

Java 版と C++ 版を比較すると, C++ 版の転送完了時間の実装値はほぼ理論に近い傾向を示しているのに対し, Java 版の実測値の傾向は C++ 版および理論値と異なり, 分割数を増やした場合に転送時間は一般的な増加ではなく増減を繰り返す傾向や, Java 版では 30 分割で MBT が他構造に対して最短の転送時間を示す一方, C++ 版では多分割数で最短の転送完了時間を示した DLT が最も遅くなるなど, C++ 版と異なる傾向が見られた. この原因の解明については今後の検討課題とするが, 転送データサイズと Java 版, C++ 版の違い以外はほぼ同一条件で実験を行っているため, この傾向の違いは Java VM の内部構成に起因するものと推測される.

3.2 ファイルサーバからの一斉データ転送との比較

他の転送手法として, 切り取った教師画面のイメージをファイルに保存してファイルサーバに置き, コマンドにより受講者端末がこのサーバにアクセスしてファイルを開き, イメージを復元する手法が考えられる. 今回, 我々の電子黒板システム (Java 版) にこの手法を実装して転送完了時間の測定を行った. その結果と, 我々の提案方式との比較を図 10 に示す.

ファイルを介した転送手法では, 台数に比例して読

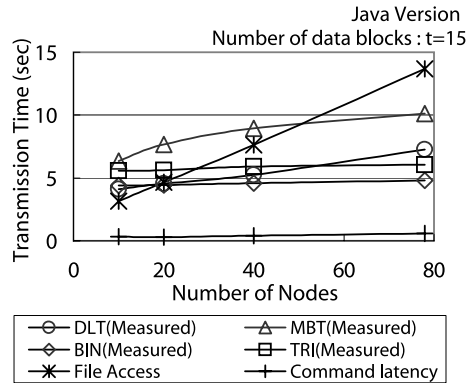


図 10 ファイル読み込みとツリー状経路方式の転送完了時間の比較 (Java 版)

Fig. 10 Comparison of transmission time between the case of our method and file reading.

み込み完了時間が增大していることが図 10 (File Access) から読み取れる. すべての端末が同時に読み込みを開始するのが理想であるが, 実際は読み込み開始を指示するコマンドの伝播の時間差により, 各端末の読み込み開始に遅延が生じる. 図 10 には, 最初のファイル読み込み開始から最後のファイル読み込み開始までの時間 (Command Latency) も示しているが, これによる遅延を差し引いても読み込み完了時間が台数に比例していることは明らかである. これは台数の増加により, ファイルサーバにアクセスが集中してサーバの負荷が増大し, ファイル読み込みの速度が低下しているためと考えられる.

図 10 には, 図 9 を基に, BIN で最短の転送完了時間を示したと見られる 15 分割における, 台数と転送完了時間の関係も示している. これらの比較から, 本実験の場合のようなスイッチングハブに接続された多数の端末に同じデータを一斉転送する場合は, 経路をツリー状にしてメッセージパッシングで転送する手法が適しているといえる.

3.3 UDP によるデータ転送との比較

次に, 本システムと UDP との比較のため, 本システムの画像転送プロトコルを UDP に変更して転送完了時間の測定を行った. この測定では以下の環境を用いた. OS は Windows 2000, 教師端末は CPU: Celeron 500 MHz, メモリ 64 MB, 受講者端末は CPU: Celeron 733 MHz, メモリ 128 MB とし, 最大 20 台で測定を行い, また, 比較のため同環境で本来の TCP を用いた場合の測定を行った.

図 11 にはそれぞれ, データ分割数 15 の場合の Star (すべての受講者端末を教師端末に直接接続) と BIN での測定結果を示す. ここで, Star において UDP の

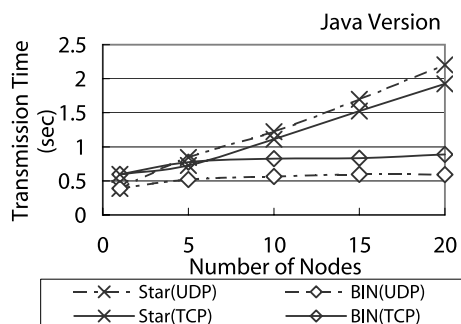


図 11 TCP と UDP の転送完了時間の比較 (Java 版)

Fig. 11 Comparison of transmission time between the case of TCP and UDP.

転送完了時間が TCP を上回る現象が生じているのは、教師端末の負荷の増大に加え、TCP を前提とした本システムに UDP を実装したため、受信画像処理などの部分がボトルネックになっていると考えられる。また、UDP が TCP を下回っている BIN では、データ損失がツリーの末端に達するにつれ増大し、実転送量が減少している影響が考えられる。いずれの場合も UDP と TCP の転送完了時間の差は比較的小さい。ブロードキャストやマルチキャストを用いれば、これよりさらに短く、かつ台数にほとんど依存しない転送完了時間が得られると予想されるが、UDP では実際に少なからぬデータ損失が生じており、実用上問題となる。図 11 より、静止画の送信に限って言えば TCP による転送は本実験の範囲内では十分実用に耐えうると考えられる。

3.4 評価

図 9 から、本実験で用いたノード数 80 台以内の範囲では、本電子黒板システムはおおむね実用に耐えうる転送完了時間を示しているということがいえる。Java 版については仕様上、データの転送量が C++ 版で同サイズのデータを転送する場合の約 3 倍となっていることから、転送する画像データについて圧縮などの処理を行う必要があると考えている。図 9 はさらに、転送経路の構成やデータの転送(分割)方法が転送完了時間に影響していることを実証している。この結果からは最適な転送経路を特定することはできず、教室の規模などにより適切な転送経路を選択することが必要であるといえる。図 10 より、本システムが採用しているデータ転送の手法および経路により特定サーバへの負荷の集中が避けられ、より速いデータ転送を実現できることがいえる。また、図 11 は本システムの方法が UDP と比較しても実験の範囲内では実用に十分耐えうることを示している。

本システムは、UNIX や Windows のような特定 OS

に頼る機能は使わず、汎用性のある TCP で実用に耐えうる性能を目指したが、少なくとも本実験の範囲内ではある程度達成されたと考えられる。他の環境での性能比較、および複数環境の並存が想定される遠隔授業などでの性能測定は今後の課題となる。

4. おわりに

本論文では、一般的な分散ネットワーク環境で利用可能な電子黒板システムについて論じた。教師画面を受講者端末へ転送、表示する本システムは、データの信頼性を確保するため TCP を用い、これにより生じる画面表示の遅延を最小限に抑え、かつ通信により各ノードにかかる負荷を軽減するため、データの転送経路をツリー状に構成して並列かつ段階的に転送を行っている。教師画面の転送など黒板本体の機能は P2P により実現されているが受講者の端末起動状況の動的変化に対応するため、受講者の起動状況を管理して転送経路の再構成を行うための管理サーバを導入している。

今後の課題として、さらにデータ転送の高速化を図るため、転送データをより軽くするためのフォーマットや圧縮などの検討、また、コネクションレス型ながらデータの信頼性を付加している Reliable Broadcast^{(14),(15)}の利用などを検討している。また、管理サーバにおいては、受講者が途中で電子黒板の使用を終了することによる経路切断に対するより最適な経路組替えの手法の確立が課題となる。また、現在は指定の経路構成に従って新規参加端末の経路への追加を行っているが、今後は転送完了時間面での最適化を考慮した手法、たとえば、接続先の回線の混み具合などを判断してより最適な接続先を判断する方式などを検討している。さらに、たとえば教師の出題に対する受講者の回答の回収、集計や、受講者の個別質問に対する回答の全受講者への配信などといった教師と受講者間の双方向のデータ転送を追加するなどして使い勝手の向上を図るよう検討している。

謝辞 本実験を行うにあたり、ご協力をいただきました九州工業大学情報科学センター山之上研究室の山根真人君、澤田崇君、堤宏智君、および九州共立大学情報処理教育研究センターの方々へ感謝の意を表します。本研究の一部は、科学研究費補助金基盤研究(C) 2009680401 の援助を得て行われています。

参考文献

- 1) Rojas, R., Friedland, G., Knipping, L., et al.: *Electronic Chalk* (2001). <http://kazan.inf.fu-berlin.de/echalk/index.e.html>

- 2) Abdel-Wahab, H.M. and Feit, M.A.: XTV: A Framework for Sharing X Window Clients in Remote Synchronous Collaboration, *Proc. Tri-comm '91*, pp.159–167 (1991).
- 3) John Bazik: *XMx—An X Protocol Multiplexor*. <http://www.cs.brown.edu/software/xmx/home.html>
- 4) Rozek, A.: *TeleDraw*. <http://www.uni-stuttgart.de/rus/Projects/MERCI/MERCI/TeleDraw/Info.html>
- 5) Kawai, T., Ikeda, M. and Okada, M.: Point-to-Multipoint Communication Protocol on Window-Based Network Presentation System, *Trans. IEICE INF. & SYST.*, Vol.E80-D, No.2 (1997).
- 6) 平原貴行, 山根真人, 山之上卓, 安在弘幸, 澤田崇, 堤 宏智: TCP/IP プロトコルを利用したパソコンネットワークのための電子黒板システム, 分散システム運用技術シンポジウム '98 論文集, pp.79–85 (1998).
- 7) Yamanoue, T., Shimizu, M. and Fujiki, T.: Development of an Electronic Chalkboard for a Large Classroom by Parallel Programming and Its Application to English Classes, *Proc. APITITE94*, Vol.2, pp.651–656 (1994).
- 8) 平原貴行, 山之上卓, 山根真人, 安在弘幸: 電子黒板システムにおけるデータ転送経路構成及び配信方法の比較, 信学論 (DI), Vol.J82-D-I, No.8 (1999).
- 9) Hirahara, T., Yamanoue, T., Anzai, H. and Arita, I.: Sending an Image to a large number of nodes in short time using TCP, *IEEE International Conference on Multimedia and Expo* (2000).
- 10) Stelzer, G.: Versatile 10/100 Megabit/sec Ethernet connection combines 10-base T and 100-base TX performance, *Elektronik* (Germany), Vol.45, No.13, pp.54, 56, 58–60 (1996).
- 11) van Gelder, T.: The aims of 100 Base-T, *Elektronica* (Netherlands), Vol.42, No.16, pp.35–37 (1994).
- 12) Farley, A.: Minimum broadcast network, *Network*, Vol.9, pp.313–332 (1979).
- 13) Shastri, A.: Broadcasting in General Network I: Trees, *Lecture Notes in Computer Science*, Vol.959, pp.482–489 (1995).
- 14) Kaashoek, M.F., Tanenbaum, A.S., Hummel, S.F. and Bal, H.E.: An efficient reliable broadcast protocol, *Oper. Syst. Rev.* (USA), Vol.23, No.4, pp.5–19 (1989).
- 15) Hong-Yi, T. and Kai-Yeung, S.: On the message and time complexity of protocols for re-

liable broadcasts/multicasts in networks with omission failures, *IEEE J. Sel. Areas Commun.* (USA), Vol.13, No.7, pp.1296–1308 (1995).

(平成 13 年 3 月 12 日受付)

(平成 13 年 10 月 16 日採録)



平原 貴行 (正会員)

平成 5 年九州工業大学工学部電気工学科卒業。平成 7 年同大学大学院工学研究科博士前期課程修了。同年より九州共立大学工学部助手。平成 10 年より九州工業大学大学院情報工学研究科博士後期課程在籍中。電子情報通信学会会員。



山之上 卓 (正会員)

昭和 59 年九州工業大学大学院修士課程修了。昭和 62 年同大学院博士後期課程単位取得退学, 同年九州工業大学工学部助手, 平成 2 年, 九州工業大学情報科学センター講師, 現在同助教。博士 (工学)。教育支援システムや言語処理系等の研究に従事。電子情報通信学会, ACM, IEEE 各会員。



安在 弘幸 (正会員)

昭和 39 年九州大学工学部電子工学科卒業。昭和 41 年同大学大学院工学研究科修士課程修了。工学博士。昭和 42 年九州大学中央計数施設講師。昭和 47 年九州工業大学工学部助教授, 教授を経て, 平成 5 年より九州共立大学工学部教授, 情報処理教育研究センター所長。電子情報通信学会, 人工知能学会会員。



有田五次郎 (正会員)

昭和 38 年九州大学工学部電子工学科卒業。昭和 40 年同大学大学院修士課程修了。同年九州大学講師 (中央計数施設)。昭和 59 年九州工業大学工学部教授を経て, 昭和 63 年同大学情報工学部教授 (知能情報工学科)。工学博士。計算機アーキテクチャ, 並列処理システム, 計算機ネットワークの研究に従事。電子情報通信学会, ソフトウェア科学会会員。