多地点での動画配信及びスイッチング支援ミドルウェアの設計と実装

土井宏真
東京工業大学 情報理工学研究科
hiehie@uml.gsic.titech.ac.jp

望月祐洋
東京工業大学 学術国際情報センター
moma@gsic.titech.ac.jp

概 要

遠隔会議や遠隔授業等の動画配信を利用したアプリケーション開発には，時間や手間など多くのコストがかかる．本研究ではネットワークにマルチメディアデータソースとシンクが遍在する環境を想定し，端末間の動画配信スイッチング制御モデルを提案し，その制御を実現する通信ミドルウェア ID^{2}O の構築によって開発コストの低減を図った．本システムの特徴はユニキャストとマルチキャストを併用した通信モデルの採用及び NAT など実際の運用環境での使用を考慮した点にある．

おいて，多くの通信技術を “意識させない”プログラミングを実現させることである．そのために本研究では既存のビデオストリーミングを利用したシステムの分析に基づき，その開発支援を行う通信ミドルウェアを設計，提案する．

本研究で構築した通信ミドルウェア，ID^{2}O (Interacrive and Dynamic Device Organization) は，ビデオストリーミングアプリケーションの容易な開発を可能にする．またミドルウェア内部の機構で，ビデオストリーミングの NAT 越え機能を実現し，プログラマの負担を軽減する．

1 はじめに

近年，日本におけるブロードバンドサービスの普及が目覚ましい速度で進んでいる．そして，その急速な普及を背景にテレビ電話，遠隔授業，ビデオ会議など，広帯域な回線を生かしたビデオストリーミングを行う通信アプリケーションが多く普及している．今後，このようなソフトウェア開発の需要は一層高まると予想される．

しかし，既存のソフトウェア開発インタフェースは開発者の支援の役割を十分に果たしているとは言えない．

通信プログラムの実装のためには，まず，ネットワーク構造，ルーティングの仕組み，通信プロトコル，セッション管理など，ネットワークに関する知識が要求される．また，ビデオストリーミングを利用する場合，映像・音声などのマルチメディアデータのエンコード・デコード，データフォーマット，マルチメディアデータ通信プロトコルなどの知識も加えて必要となる．これらの要求される知識は，プログラマにとって大きな障壁となる．

また IPv4 アドレス枯渇問題により普及した NAT (Network Address Translation) 技術により，インターネットを介した動画配信などのサービスが正常に提供されない環境が存在する．NAT は，インターネットの基本理念である End-to-End 通信・双方向性を失わせていると同時に，この特殊な環境もプログラマによるアプリケーション開発の負担になっている．

本研究の目的は，動画配信アプリケーションの開発に

2 ミドルウェアへの要求仕様

2.1 既存の遠隔授業システムの分析

動画配信を利用したシステムの一例として，遠隔授業システムが挙げられる．これは，講師の映像と音声を遠隔地にいる受講者へ配信して講義を実現するシステムである．ここでは，この例からミドルウェアに必要な機能や仕様を考察する．

遠隔授業における講義の映像・音声データの配信は，その配信形式により大きく 2 つに分類できる．映像・音声データをリアルタイムに配信するライブ配信形式と，録画・録音され蓄積型データとなった映像・音声データを受講者の好きな時に受信するオンデマンド配信形式がある．また，蓄積型データを決められた時間に配信するスケジュール配信形式も存在する．

図 1 はライブ配信形式を用いた遠隔授業システムの概図である．受講者は講師の端末環境に接続要求を行い，講師側から受講者へデータ送信が行われる．

送信側には講師が存在し，その映像入力装置 (カメラ) と音声入力装置 (マイク) が存在する．受信側には受講者が存在し，映像出力装置 (ディスプレイ) と音声出力装置 (スピーカ) が存在する．

オンデマンド配信形式やスケジュール配信形式の場合には，送信側は講義の映像・音声データを蓄積しているサーバとなる．

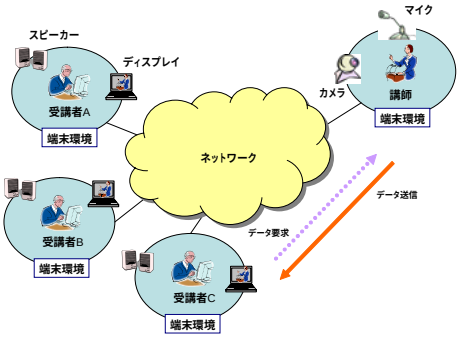

図 1: 遠隔授業システム (ライブ配信)

2.2 分析に基づいた要求仕様の考察

各端末環境間で行われる通信は 2 種類に分類できる．すなわち，映像・音声データの通信と，その接続の開始・終了や通信先の指定・選択を行うための通信である．前者の通信を行う機能を**ストリーミング機能**，後者の通信を行う機能を**スイッチング機能**と呼ぶことにする．

2.2.1 ストリーミング機能

遠隔授業システムにおける端末環境には，共通して**データ送受信機能**が備えられている．同時に**データ入出力機能**を持った装置が設置されている．

遠隔授業システムのように，一般的にはデータ入力機能とデータ送信機能，データ受信機能とデータ出力機能はそれぞれ同じ端末環境で行われる．よって，本ミドルウェアの設計ではデータ入力・送信する端末とデータ受信・出力する端末に分類する．

2.2.2 スイッチング機能

ここで定義するスイッチング機能とは，送信パケットのルーティングではなく，端末間の接続を切り替える操作等の機能を指す．その機能の実現のために，各端末環境間で操作命令等の通信を行う必要性がある．

本ミドルウェアでは各端末間の接続方式として Server-Client 方式を双方向に適用することで，Peer-to-Peer 方式の通信を実現する．映像・音声データ通信を行う端末をクライアントとし，それらを集中的に管理し制御するサーバを新たに用意する．

3 ミドルウェアの設計

3.1 データ入出力装置の抽象化

ネットワーク上に散在するカメラ，マイク，ディスプレイ，スピーカ，データファイルなど，データの入出力機能を有する装置を抽象化して AVClient (Audio/Video Client) と呼ぶ．それらのデータ入出力装置は，それ自体が送受信・計算機能を有する場合と，計算機に接続されている場合がある．ここで指す計算機には PC (Personal Computer) だけでなく，PDA (Personal Digital Assistance)，携帯電話などの携帯型端末も含まれる．また，ここで想定するカメラには “ネットワークカメラ” や “PC に接続された USB カメラ”，“携帯電話に付属しているカメラ”などが含まれる．

さらに，様々な端末環境を想定して，機能ごとに AVClient を分類・コンポーネント化した．データ入力機能と送信機能を AVSource，データ受信機能と出力機能を AVSink，スイッチングのクライアントとしての機能を AVController としてそれぞれ抽象化した．

また，AVClient に対して，スイッチング機能を実現するために，AVClient の情報を集中的に管理するコンポーネントを導入した．これを AVConnector (Audio/Video Connector) と呼び，AVClient は AVConnector に自身の情報を登録する (この作業をログインと呼ぶ)．AVConnector 自身はデータ入出力機能を持たない．

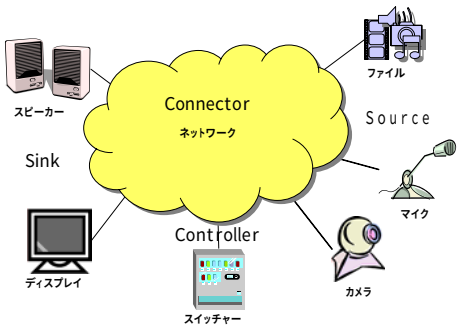

図 2: 抽象化

以上の抽象化を図 2 に示す．

3.2 コンポーネント間の連携

3.2.1 データ送受信開始プロセス

AVSource から AVSink へのマルチメディアデータ送受信は，AVController が AVSource と AVSink を指定し，AVConnector へ命令を送信するだけで開始される．ミドルウェアが，セッション管理等のネットワーク操作を全て代行することで，プログラマにネットワーク操作を意識させないスイッチング機能を提供する．

3.2.2 AVConnector の連携

ネットワーク上にあるすべての AVClient を 1 つの AVConnector で管理することは，膨大な負荷が予想される

ため現実的ではない．そこで 1 つの AVConnector が小規模のネットワークセグメント内のクライアントを管理するような運用を想定した設計が考えられる．そして異なるネットワークセグメントに存在する AVClient 間の接続を実現するために，AVConnector 間で連携を行う．本ミドルウェアではこのようにしてスケーラビリティを考慮した設計を行った．

3.3 マルチキャスト通信モデルの包括

ネットワークトラフィックは AVSink の数に正比例して増大する．マルチメディアデータの送受信時には，これが原因で回線の帯域限界を超えてしまうことがある．一般的に，この問題の解決策としてはマルチキャスト技術が利用される．マルチキャスト技術により無駄なパケット送信を減少させ，AVSource 側の回線の負担も最小限に抑えられる．

本ミドルウェアではマルチキャスト技術を導入するために，仮想的な Source と Sink である MulticastSource と MulticastSink を導入した．これらは 1 つのマルチキャストアドレスとポート番号を有しており，これらと AVSource などを指定することで簡単にマルチキャストが実現できる．

3.4 NAT 越え問題

3.4.1 NAT 技術とその問題点

NAT とは，一般に 1 つのグローバルな IP アドレスを複数のコンピュータで共有する技術を指す．NAT 内部のプライベート IP アドレスを外部から取得する方法は一般的に存在しない．よって，NAT 内向きの通信を実現できないことが多い．これはインターネットの双方向性を喪失させてしまっているが，この一方向性がセキュリティ保全上都合がよかったという点も，NAT 技術の普及の一因となっている．

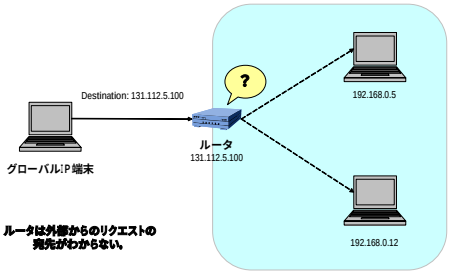

図 3: NAT 越え問題

NAT 越え問題の概図を図 3 に示す．NAT 越え問題とは，NAT 外部と NAT 内部で双方向性を確立するためにアドレス変換に特別な処理を行う必要が生じることを指す．

3.4.2 NAT 越え問題の解決策

まず，NAT の内側に AVConnector を 1 つ設置する．次に，NAT の入り口のルータの設定をユーザ自身で変更し，特定のポートを NAT 内部の AVConnector へポートフォワーディングさせる．これにより，外部の AVConnector と NAT 内部の AVConnector の通信の双方向性が確立する．その後，NAT 内部の AVClient を NAT 内部の AVConnector へログインさせ，NAT の外との通信を，全てその AVConnector がフォワーディングすることにより，外部に存在する AVSource や AVSink とのデータ送受信を実現させる．

この時，NAT 内部の AVConnector は，外部からのデータパケットを内部の AVSink へルーティングする必要性がある．そのためにデータパケット内部の情報 (RTP パケット内部の SSRC/Synchronization Source という識別子) を利用する．この情報は通信開始時のプロセスで AVSource から受け取る仕組みになっている．

4 ミドルウェアの実装

実装環境として OS は Windows XP，プログラミング環境には J2SDK 1.4.2 を使用し，API として Java Media Framework 2.1.1e を利用した．

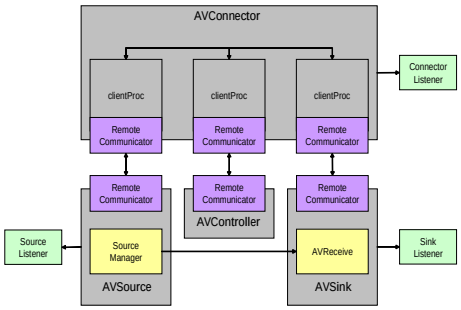

図 4: 各クラスの相関図

構成した各クラス間の相関を図 4 に示す．スイッチング命令通信機能を RemoteCommunicator クラスとして実装し，ログインしている AVClient ごとに生成する AVConnector の内部クラス ClientProc と，AVClient クラスがこれを継承しているクラスになる．AVClient クラスを継承した AVSource，AVSink，AVController クラスを実装し，AVSource にはマルチメディアデータ送信機能として内部クラス SourceManager を持ち，AVSink はマル

チメディアデータ受信機能として内部クラス AVReceive を持つ．AVSource と AVSink，AVConnector は非同期メソッドを持つため，コールバック用のインタフェース SourceListener，SinkListener，ConnectorListener をそれぞれ用意した．

5　アプリケーション構築例

実装したシステム例として，ビデオストリーミングのスイッチングを行う単純なシステムを挙げる．

図 5: ビデオスイッチングシステムの実行画面

図 5 はアプリケーションのスクリーンショットである．上部のウィンドウが GUI プログラムであり，これは内部に AVController インスタンスを持つ．AVConnector にログインしている AVClient の情報を用いて横方向に AVSource，縦方向に AVSink のパネルを構成し，格子状にそれぞれの接続を表すパネルを描いている．黄色いパネルは接続状態を表している．

ディスプレイとカメラのペアが 2 つあり，それぞれ相手を映している．ディスプレイは AVSink と，カメラは AVSource とそれぞれ関連付けられていて，すべてが同一の AVConnector にログインしている．AVController の AVConnector 越しのスイッチング命令に従って，AVSource と AVSink はデータ送受信を行っている．

AVSink と関連付けられたディスプレイを増やせば，遠隔講義の形態とすることができる．また，受講者側に AVSource と関連付けられたカメラやマイクを追加すれば，双方向の講義へと簡単に変更することもできる．

6　まとめ

本研究の目的は，ビデオストリーミング・スイッチングを利用したアプリケーションの容易な開発環境を提供することであった．問題解決のために既存のシステム例の分析を行い，機能をストリーミング機能とスイッチング機能に分類し，さらにストリーミング機能をデータ入力・送信機能とデータ受信・出力機能に分類した．そしてそれに対応して AVConnector，AVSource，AVSink，AVController の 4 つの基本構成要素から成るミドルウェア ID^{2}O の設計・実装を行った．

設計では，ユニキャストとマルチキャストの併用が可能な送受信モデルを提案・採用した．また実装において NAT 越え問題に着目し，AVConnector にポートフォワーディング機能を組み込むことで，それを解決した．これらはスイッチング機能を提供するミドルウェアとして重要な機能であり，ID^{2}O の大きな特徴でもある．

ミドルウェアの評価のために，ID^{2}O の提供する API を利用して 2 つのストリーミング・スイッチングを利用したシステムを試作した．ビデオスイッチングシステムの例において実装した Java プログラムは 50 数行と短く，提供された API により容易にビデオストリーミングアプリケーションを構築できることが分かる．

今回のミドルウェアの実装では携帯端末のサポートを行っていない．今後その対応も行うことで，よりミドルウェアの完成度を高め，また有用性について検証を進める予定である．

参考文献

[1] Sun Microsystems, http://java.sun.com/products/java-media/jmf/, Java Media Framework API(JMF)

[2] H.Schulzrinne, S.Casner, R.Frederick, and V.Jacobson, “RTP:A Transport Protocol for Real-Time Applications, ” RFC1889, IETF, Jan.1996

[3] 大谷哲夫，遊佐博幸，三沢雅一，木内舞，“分散リアルタイムネットワークアーキテクチャにおけるミドルウェアの概念設計，” 電子情報通信学会研究報告，2002 年 9 月号

[4] 永原崇範，鹿島拓也，猿渡俊介，川原圭博，南正輝，森川博之，青山友紀，篠田庄司，“ユビキタス環境に向けたセンサネットワークアプリケーション構築支援のための開発用モジュール U 3 の設計と実装，” 電子情報通信学会研究報告，2003 年 3 月号

[5] 岩井将行，中澤仁，西尾信彦，徳田英幸，“分散コンポーネントによる即興的アプリケーション構成機構の実現，” 情報処理学会論文誌，2002 年 6 月号

[6] 松浦宣彦，松本敏宏，清末悌之，菅原昌平，正木茂樹，“簡易型多地点テレビ会議システム NetForum の開発と評価，” 情報処理学会論文誌，2000 年 01 月号