

Title	ATMマルチキャストを用いた大規模マルチメディアネットワークに関する研究
Author(s)	木村, 範彦
Citation	
Issue Date	2000-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1347
Rights	
Description	Supervisor:丹 康雄, 情報科学研究科, 修士

修 士 論 文

ATM マルチキャストを用いた
大規模マルチメディアネットワークに関する研究

指導教官 丹 康雄 助教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

木村 範彦

2000 年 2 月 15 日

要 旨

大規模なマルチメディアにおいてマルチキャスト通信を利用する場合の帯域や識別子といった資源を中間ノードにより有効利用する手法について検討する。また、実際のシステムに適用するために、従来のシステム間との間で用いるプロトコルを設計し、仕様記述言語により記述することでその仕様を明確化する。

本手法は ATM マルチキャストイングにおいて送受信者間で十分な帯域が確保できない場合に、中間ノードによりデータの圧縮やコマ落としといった処理を行ったり、1 コネクションあたり用意されたユーザを一意に識別するための識別子空間の枯渇といった問題に対し、複数のコネクションを中間ノードにより管理して識別子空間を拡張することで、様々な環境のユーザに対し、柔軟に対処することが可能となる。

目次

1	はじめに	1
2	マルチキャスト通信	3
2.1	通信方式	3
2.1.1	ユニキャスト方式	3
2.1.2	ブロードキャスト方式	4
2.1.3	マルチキャスト方式	4
2.2	ATMにおけるマルチキャスト通信	4
2.2.1	ATM	4
2.2.2	ATMの階層モデル	5
2.2.3	ATMにおけるコネクション設定	11
2.2.4	ATMにおけるマルチキャスト通信の効果	14
2.2.5	ATMマルチキャストにおける通信形態	14
2.2.6	VC meshとMCSの比較	15
2.3	マルチキャスト通信における問題点	17
3	中間ノード	19
3.1	中間ノードの性質	19
3.2	マルチキャストを利用するアプリケーション	20
3.3	中間ノードによるサービスの提供	21
3.4	中間ノードにより提供するサービス例	21
3.5	中間ノードを利用した配信形態	22
3.6	中間ノードが存在しない場合の各サービス	24
3.7	中間ノードの設計	25
3.7.1	中間ノードのモデル	25

3.7.2	複数の中間ノードの管理	25
3.8	中間ノードの有無による評価	27
3.8.1	コストによる比較	27
3.8.2	資源消費量による比較	27
3.8.3	価格と資源消費の関係	29
3.9	中間ノードの効果	29
4	実システムへの適用	31
4.1	JAIST VideoLAN システム	31
4.1.1	システム構成	31
4.2	JAIST VideoLAN システムのアーキテクチャ	32
4.2.1	ユーザと RMA 間のメッセージ	32
4.2.2	RMA と TS 間のメッセージ	33
4.3	中間ノードの組み込み	33
4.3.1	中間ノードと RMA 間のメッセージ	34
4.3.2	中間ノードの登録	34
4.4	中間ノードの利用	35
4.5	中間ノードにおける VC 接続機構	36
4.6	中間ノードを利用したサービスや機構	38
4.6.1	JAIST VideoLAN システムにおいて提供されるサービス	38
4.6.2	ATM マルチキャスト通信を拡張する機構と提案	39
4.7	インタフェースの拡張	40
4.7.1	従来のインタフェース	41
4.7.2	提案するインタフェース	42
4.8	形式記述法によるシステム設計	45
4.8.1	SDL による記述	47
4.9	実装	52
4.9.1	予備実験	52
4.9.2	実装システム	52
4.9.3	実験結果	52
4.10	評価	53
4.10.1	シグナリングコスト	53
4.10.2	複数の中間ノードによるサービスの提供	54

5	検討及び考察	66
5.1	中間ノードに対する検討	66
5.1.1	資源消費量	66
5.1.2	通信遅延時間	66
5.1.3	コネクションの集中度	67
5.1.4	シグナリングコスト	68
5.1.5	全体的な評価	68
5.2	JAIST VideoLAN システムにおけるアドレス体系	69
5.3	実装について	69
6	今後の課題	70
6.1	複数の中間ノードの配置における分散処理	70
6.2	異種ネットワーク間との結合	70
7	おわりに	71

第 1 章

はじめに

近年のネットワーク技術の目覚ましい発展により、動画像や音声といった、いわゆるマルチメディアデータが高速に転送されるようになってきている。このようなマルチメディアデータを扱うシステムの利用形態としては、単方向のデータ送信タイプであるニュースや天気予報のようなブロードキャスト型のシステムや、双方向性を持つビデオ会議やテレビ電話のようなシステムがあるが、現状のこれらのシステムの多くは画像の解像度が低かったり、動きがスムーズでないなどユーザの要求を満たしているとはいえない。また特別な機器の設置や専用線での利用など、利用条件も限定されているものもある。

一方、学校や特別な施設では FC(Fibre Channel)、GbE(Giga bit Ether)、ATM(Asynchronous Transfer Mode) のような広帯域の確保や、QoS(Quality of Service) の保証が可能となるネットワークが施設されており、画像や音声といったマルチメディアデータが比較的扱い易い。また、画像入出力装置にビデオカメラやテレビのような家電機器を利用することで、安価で操作に特別な訓練を必要としないシステムが構築可能となる。

このような条件を前提にしたビデオデータ配信システム JAIST VideoLAN [1] を本学で開発中である。この JAIST VideoLAN システムでは基幹のネットワークに ATM を用いており、複数のノードへのデータの配信はこの ATM の機能の一つである ATM マルチキャストにより行なわれる。マルチキャストとは複数の受信者を 1 つのグループとして把握、このグループに対して同じデータを送信する通信形態であるが、同じ内容を複数の人に送信するテレビ放送などのサービスにおいては、途中のスイッチやルータにより効果的にデータを複製して無駄なトラヒックの発生を抑えるこの方式は大変効果的であり、帯域といった資源を有効利用できるため、より多くの人とそのネットワークを利用できる。

グループとは、同じサービスを享受することを望む人々の集合であるが、グループ内の一部のメンバが他のメンバと多少でも異なる要求を欲した場合、そのためだけに物理的

接続の変更や、そのシステム自体からの除外を余儀なくされる。

また、受信者が要求するであろう機能は誰かが持つておかなければならないが、送信者、あるいは受信者になりうる全ての機器に対して、これを要求することは難しい。

そこで、ネットワーク中に様々なサービスを提供する特別なノードである「中間ノード」を配置し、同じデータに付加するサービスが異なるグループ毎の処理を行ったり、網側でサービスを提供することで使用する機器類に変更を加えないで様々なユーザの要求に対して柔軟に対処可能とする。

本論文では、この JAIST VideoLAN システムにおいて大規模な運用を行う際に有効であると考えられるサービスや機構について検討し、それを中間ノードにより実現する手法を示す。また、中間ノードと従来のシステム間のプロトコルを設計し、中間ノードが加わる際の効果や有効性について検証する。

第 2 章

マルチキャスト通信

本章では、ネットワークシステムにおける通信方式に注目し、各方式に対する比較を行う。また、有効であると考えられるマルチキャスト方式を利用する形態についても検討を行い、一般的なマルチキャスト通信における利用形態と、その効果について述べる。

2.1 通信方式

通信の方式には大きく分けて次の 3 つがある。

- ユニキャスト方式
- ブロードキャスト方式
- マルチキャスト方式

2.1.1 ユニキャスト方式

同じデータを複数のクライアントへ送信する場合、従来からの一般的な通信方式であるユニキャスト方式では送信者と受信者が 1 対 1 の関係にあるために、受信者が増えればそれだけ送信者からのデータの送信回数も増え、Fig.2.1(a) のように多くの部分で重複したデータが流れることになる。

また、静的な接続ツリーを形成させ、各サーバ毎にデータを複製するユニキャスト方式を拡張させた Fig.2.1(b) のような形式のものも存在する。この方式では直接的にユニキャスト方式を用いるよりは効率的ではあるが、必ずしもそのツリー形態が最良のものであるとは言えず、また手動でツリーを形成するため非効率的である。

2.1.2 ブロードキャスト方式

Fig.2.1(c) に表されるように、ブロードキャスト方式は全てのクライアントに対し、データを送信する方式である。しかし、データを要求していないクライアントにもデータを配送してしまうという欠点がある。そのため、ルータによってローカルネットワークまたは、サブネットワーク内に範囲を限定し、データの無駄な拡散を防ぐようになっている。そのため、大規模なネットワークには直接利用できない。

2.1.3 マルチキャスト方式

上記の二つの方式は、少数メンバー内での通信や、ある限定されたネットワーク内の全ての人にデータを送信したい場合には有効である。しかし今、「同じデータを複数のクライアントへ送信する場合」と限定した場合、うまく対応できない。そこで、Fig.2.1(d) に表すように、途中のスイッチやルータなどによりデータを効果的に複製し、送信者はデータを一度送信するだけで任意のノードにデータを送信することを可能としたマルチキャスト方式がある。これにより送信側のノードの負荷やネットワーク上のトラヒックが大幅に軽減することができる。

本論文では、大規模なマルチメディアネットワークにおいて、多人数の人が利用できる環境を想定しているため、最も効果的である、このマルチキャスト方式を用いる。

2.2 ATM におけるマルチキャスト通信

2.2.1 ATM

ATM(Asynchronous Transfer Mode:非同期転送モード)とは、CCITT が次世代の ISDN として標準化した B-ISDN のための基盤伝送システムで、「セル」と呼ばれる小さな固定長のパケットを転送単位とするセルネットワークの一つである。ATM ではデータを送信する際に、予め受信者に対してコネクションを確立し、予約した帯域幅を仮想的な一つの回線として利用することができる。ATM ネットワークは、ATM インタフェースボード、光ファイバ、ATM スイッチの3つから構成される(図 2.2)。データはセルに分割されて送信される。中間の ATM スイッチでは、セルヘッダに基づいてハードウェアで経路選択が行われる。受信側に到達したセルは再びデータに再構成される。ATM では固定長の小さなセルをハードウェアで制御することにより、高速なネットワークの実現を可能としている。

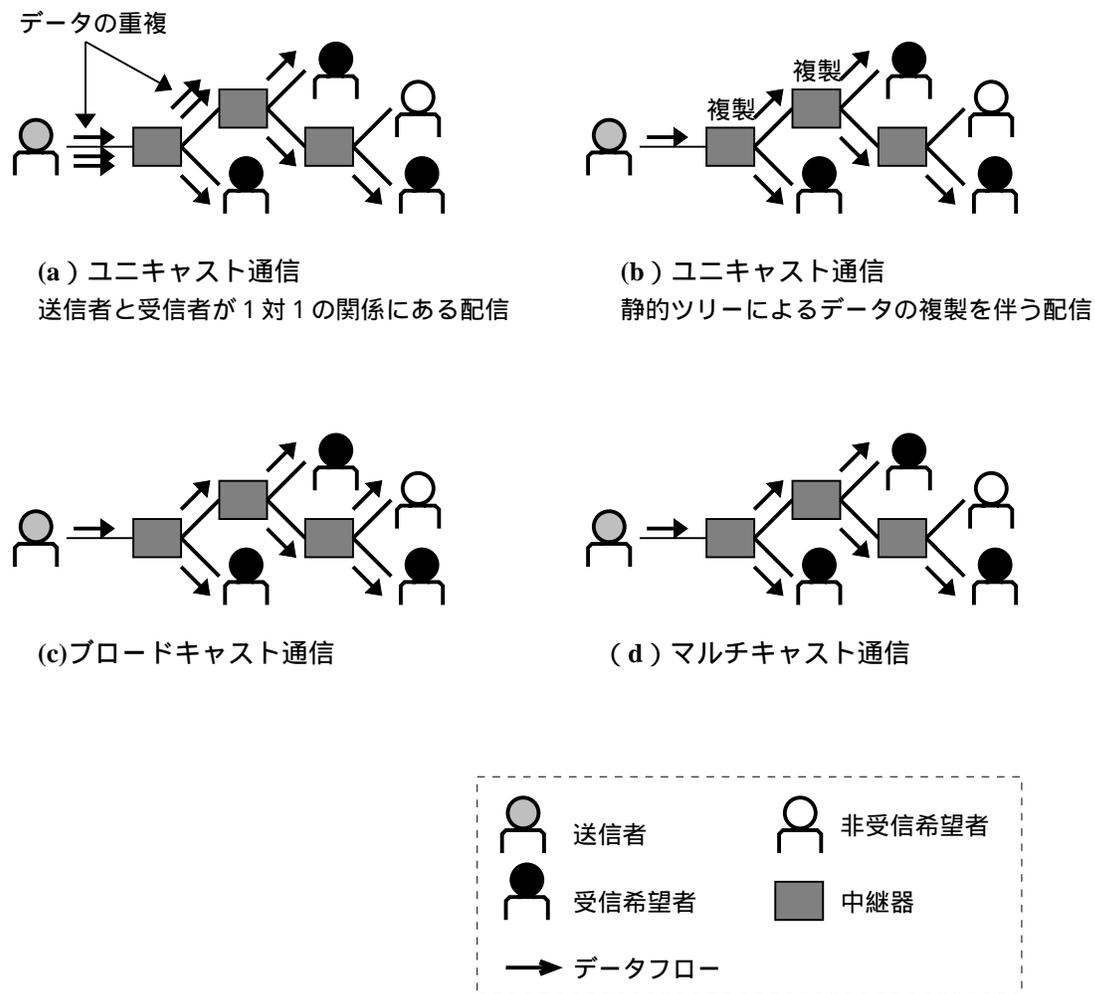


図 2.1: それぞれの通信方式による多人数へのデータ配信

2.2.2 ATM の階層モデル

ATM の階層モデルを表 2.1 に示す。ここでは各階層について説明する。

物理層

ATM の伝送媒体としては光ファイバを用いるのが一般的である。伝送速度は 155.52Mbps(SDH¹ STM-1) のものが普及しているが、622.08(SDH STM-4)、2488.32Mbps(SDH STM-16) のものも使用されている。

¹SDH(Synchronous Digital Hierarchy:同期デジタルハイアラキー) というデジタル伝送速度の階層構造のことで、物理インタフェースの国際標準となっている。

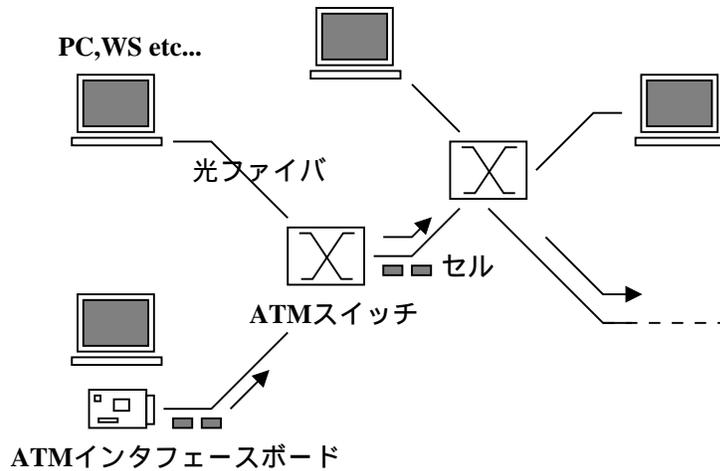


図 2.2: ATM ネットワークの構成

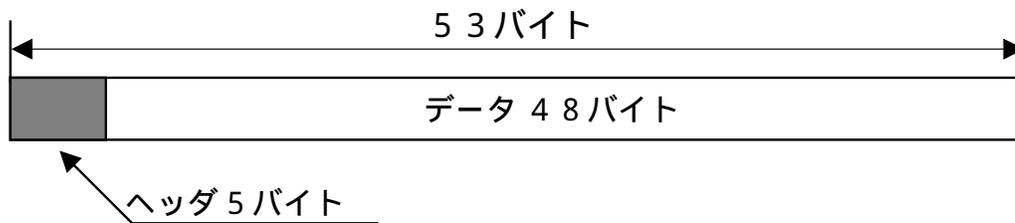


図 2.3: ATM セルの構成

ATM 層

ATM ではすべてのデータを「セル」にして転送する。ATM 層ではこのセルを取り扱う。セルは 48 バイトのペイロードと 5 バイトのヘッダを合わせた 53 バイトを単位として構成されている (図 2.3)。セルヘッダの構成は図 2.4 に示した通りである。

- GFC(Generic Flow Control:一般的フロー制御)

GFC は UNI(User Network Interface) で使われる。この 4 ビットフィールドは ATM を DQDB(Distributed Queue Dual Bus) リングのような共有アクセスネットワークで使用する際の多重化方法を調整するために確保されている。現在このフィールドは利用されていない。

- VPI(Virtual Path Identifier:仮想パス識別子)

AAL	コンバージェンス副層
	SAR(セル分割・再構成) 副層
ATM 層	GFC(一般フロー制御)
	セルヘッダ生成・抽出
	VCI/VPI 書き換え機能
	セル多重化・逆多重化機能
物理層	伝送コンバージェンス副層
	物理媒体

表 2.1: ATM 階層モデル

VPIは8ビット、もしくは12ビットの仮想パス識別子である。この識別子は、コネクションの集合で形成された仮想的なパスの識別に用いる。

- VCI(Virtual Channel Identifier:仮想チャネル識別子)

VCIは16ビットの仮想チャネル識別子である。これは同じVPIで表現される仮想パスの中で異なるコネクションを識別するために用いる。

- PT(Payload Type:ペイロードタイプ)

PTはセル情報を示す。最初のビットが0ならユーザデータセル、1なら管理用セルである。

- CLP(Cell Loss Priority:セル損失優先表示)

CLPはセルの優先度を示す。ATMスイッチなどでバッファがあふれた場合、この優先度に従ってセルを破棄する。

- HEC(Header Error Control:ヘッダ誤り制御)

HECは8ビットのヘッダエラーチェックフィールドで、5バイトのセルヘッダに対するCRC計算値が入る。ヘッダは各ホップで変化するので、各ホップ毎にチェックされ再計算される。

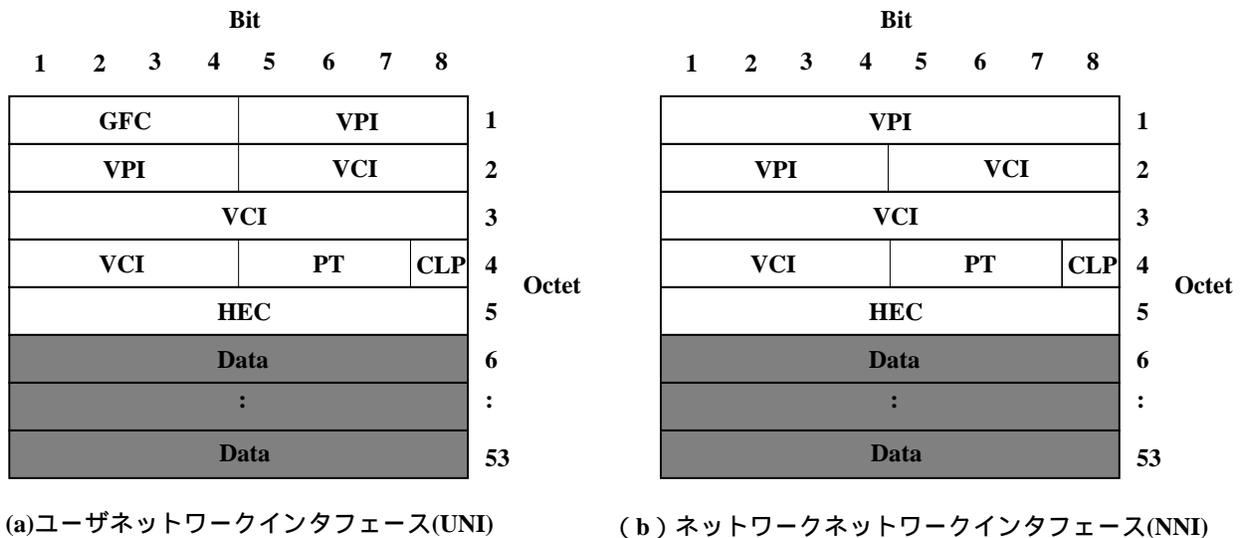


図 2.4: セルヘッダの構成

AAL(ATM Adaptation Layer:ATM アダプテーションレイヤ)

AAL はデータのセルへの分割・セルからの再構成を取り扱う。AAL の目的は受信側でセルを適切にデータに再構成できるように、効率良くセルにデータを格納することである。上位層が扱うデータの種類に応じてクラス A から D までのサービスクラスが考えられ(図 2.5)、それぞれのクラスに応じた AAL が規定されている。

- AAL1

AAL1 は映像や音声などを実時間性のある固定ビットレートのコネクション指向サービス(クラス A)を提供することを目的としている。

- AAL2

AAL2 は実時間性のある可変ビットレートのコネクション指向サービス(クラス B)を提供することを目的としている。最適化アルゴリズムにより圧縮された映像や音声などがこれにあたる。現在、標準化は行われていない。

- AAL3/4

AAL3 では X.25 のようなコネクション指向データプロトコル通信サービス(クラス C)を、また AAL4 で IP のようなコネクションレスデータプロトコル通信サービス

ストリーム配送		パケット配送	
固定ビットレート	可変ビットレート		
コネクション指向			コネクションレス指向
クラスA	クラスB	クラスC	クラスD

図 2.5: サービスクラスの分類

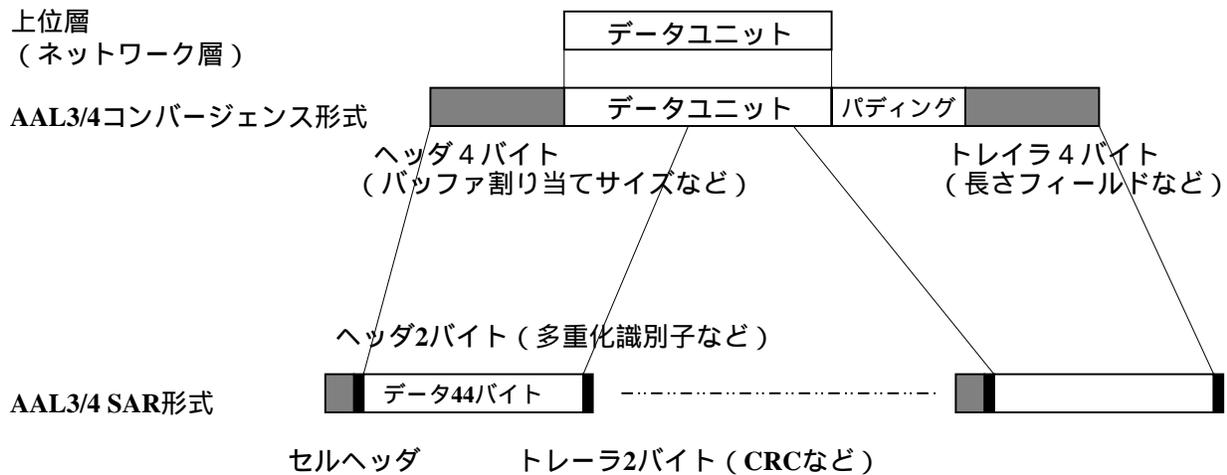


図 2.6: AAL3/4

(クラス D) を提供しようとしていた。しかし提案されたフレームはほとんど差異がなかったため、AAL3/4 として1つにまとめられた。

- AAL5

AAL5 は可変ビットレートのコネクション指向サービスを提供する。IP を用いたコンピュータネットワークでは、当初 AAL3/4 を使うことが想定されていた。しかし AAL3/4 では、送受信においてセル毎の処理が必要であり、またセルのペイロード部 48 バイトのうち 44 バイトしかユーザデータを格納することができない (図 2.6)。従って、よりオーバーヘッドの少ない AAL を求める意見から、以下のような要求を満たす AAL5 が設計された。

- ペイロード部 48 バイトは全てユーザデータで占める
- セルに対する計算機の処理は最小限に留める

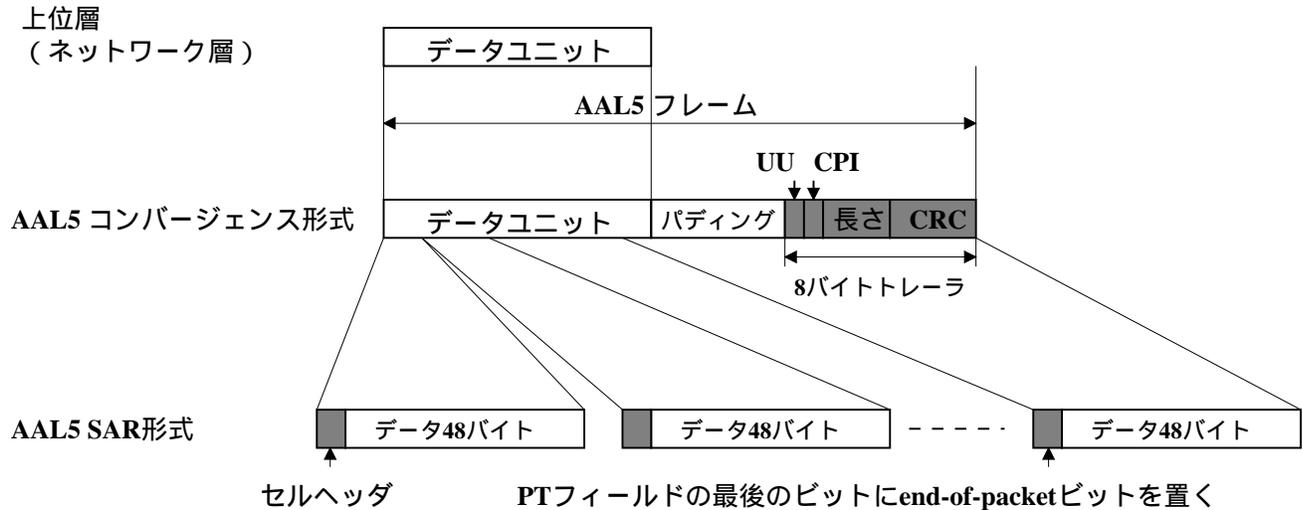


図 2.7: AAL5

- イーサネットや FDDI のような既存データ通信インタフェースと同様の振舞いをする

現在、ATM におけるデータ通信のほとんどが AAL5 を用いて行われている。本研究でも AAL5 を用いることを前提とする。そこで AAL5 についてのみ詳説することにする。

AAL5 の構成

AAL は図 2.1にあるように 2 つの副層から成る。データのセルへの分割、セルからデータの再構成を行う SAR 輻輳と、SAR 層と上位層の間でデータの流れを管理するコンバージェンス層がある。図 2.7に AAL5 SAR 形式と AAL5 コンバージェンス形式を示す。

● コンバージェンス層

コンバージェンス層では、上位層(ネットワーク層)で扱うデータユニットと、AAL5で扱うデータユニット(AAL5 フレーム)の変換を行う。上位層から渡されたデータはパディング(48 バイトのセルペイロードに収めるため)と 8 バイトのコンバージェンス層トレーラが負荷されて、AAL5 フレームになる。また逆に SAR 層から渡された AAL5 フレームは、パディング、トレーラが取り除かれ、パケット長及び、CRC チェックの後に上位層へ渡される。トレーラはユーザ-ユーザ(UU) 指示フィールド、共通パート指示子(CPI) フィールド、長さフィールド、CRC から成る。この中で UU と CPI は現在使われていない。

- SAR 層

SAR 層では、AAL5 フレームからセルへの分割・セルから AAL5 フレームへの再構成を行う。AAL5 の SAR 形式では、セルのペイロード部分 48 バイトはすべてデータ領域である。これによりヘッダ・トレーラによるオーバヘッドを最小に抑えることができる。また SAR 層では、セルヘッダのペイロードタイプフィールドの 1 ビットのみを使用する。AAL5 フレームからセルに分割した際、最後にあたるセルのセルヘッダの部分に end-of-packet ビットをセットして、そのセルがパケットの最後のセルであることを示す。これにより受信側では end-of-packet ビットがセットされたセルが来るまで、単にセルを受信バッファに溜めるだけでよいことになり、大幅な性能向上が図れる。最後のセルが到着すると、バッファにあるセルのペイロード部を継ぎ合わせてパケットを再構成する。

2.2.3 ATM におけるコネクション設定

ATM のコネクションは以下の 3 つの形態が規定されている。[7]

- 一対一単方向コネクション
- 一対一双方向コネクション
- 一対多単方向コネクション

コネクションをシグナリングプロトコルと呼ばれる帯域外データにより確立される。コネクションの確立には VPI と VCI を組み合わせた 24 もしくは 28 ビットの識別子が用いられる。これらの識別子は、コネクション確立時に中間にある ATM スイッチが容易に経路選択できるようにホップ毎に選ばれる。コネクション確立後、セルが ATM スイッチに到着すると、ATM スイッチはコネクション確立時に決めたテーブルに従ってセルヘッダの VPI と VCI を書き換え、その後セルヘッダを用いてハードウェアによる経路選択を行う (図 2.8)。これを繰り返して、セルが送信者から受信者へと転送されることになる。

ATM のシグナリングプロトコルは ITU-T 勧告により Q.2931 を基本としており、現在も開発中である。ここではユーザとネットワーク間でのコネクション確立設定手順を示す。

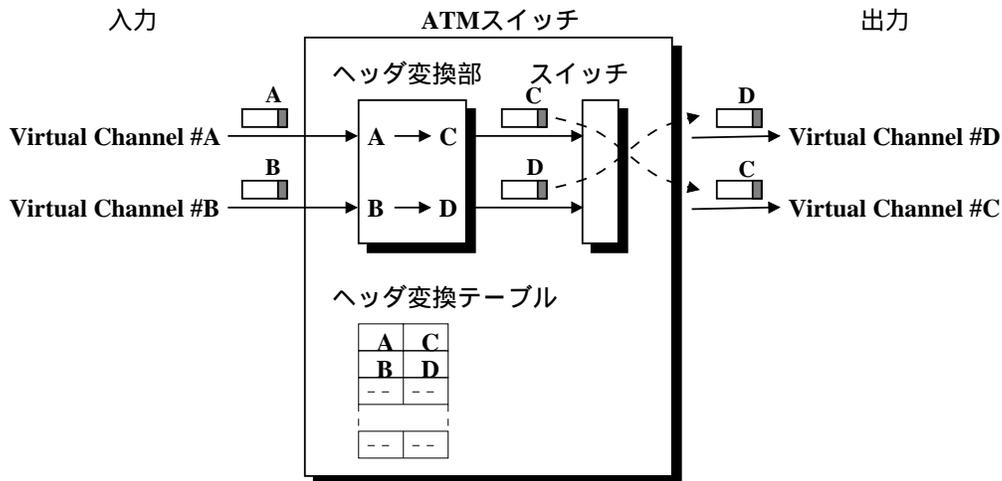


図 2.8: ATM スイッチにおけるセル交換

一対一単方向コネクション

一対一の単方向コネクションの確立手順は以下の通りである (図 2.9)。

1. 送信者はネットワークに対し、FlowSpec、受信者のアドレス、使用する AAL タイプ等を含んだセットアップメッセージを送信する。
2. ネットワークは CALL PROCEEDING メッセージを使って、SETUP メッセージの確認応答を送信者に伝える。同時に、このコネクションで使われる VPI/VCI が割り当てられる。
3. ネットワークは受信者に対して SETUP メッセージを送り、コネクション確立要求を伝える。
4. 受信者は同じく CALL PROCEEDING を送り、SETUP メッセージに含まれる内容によってコネクションを受け入れるかどうかを決める。
5. 受け入れる場合には、受信者はネットワークに対し CONNECT メッセージを送る。
6. ネットワークは CONNECT ACKNOWLEDGE を使って CONNECT メッセージの確認応答を受信者に伝える。
7. ネットワークは送信者に対して CONNECT メッセージを送り、コネクション確立が完了する。このとき発信者が CONNECT ACKNOWLEDGE を返すこともある。

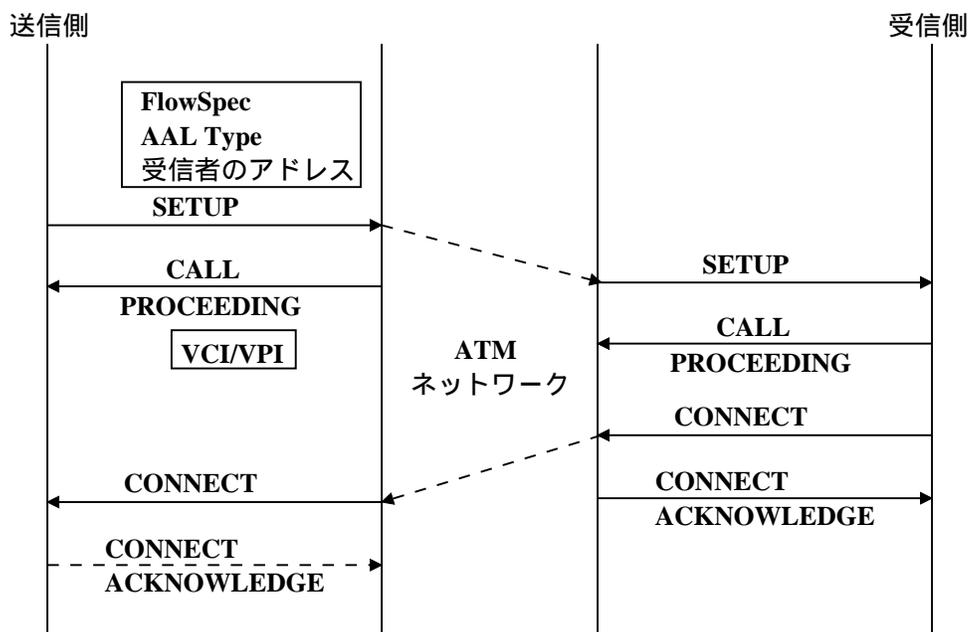


図 2.9: ATM のコネクション確立の手順

もしコネクションが受け入れられない場合は RELEASE COMPLETE メッセージを返す。メッセージを受け取った送信者及び中間のスイッチは、このコネクションに関するすべての情報を破棄する。

一対多単方向コネクション

ATM のコネクション確立に使われる ATM アドレスは、基本的に一つの ATM インタフェースを示すユニキャストアドレスであり、特定の集合を表現するようなマルチキャストアドレスは提供されていない。このため一対多の単方向コネクションを確立する場合には、まず送信者が受信者の一つと前述した手順で一対一の単方向コネクションを確立する。そして残りの受信者に対して、送信者は ADD PARTY というメッセージを用いて、最初のコネクションに枝を追加していく形で行われる。また受信者のコネクションからの解放は DROP PARTY メッセージで行われる。

本論文において対象とするシグナリングプロトコル UNI のバージョン 3.1 においては、このマルチキャストコネクションの確立、解放を行えるのは送信者のみである。つまり受信者が新たに参加したり、脱退したりする場合はすべて送信者に依頼しなければならないことになる。

2.2.4 ATM におけるマルチキャスト通信の効果

ATM におけるマルチキャスト通信では、光ファイバによるデータ転送の高速性や、QoS による品質保証が可能であるため、動画や音声といった高速性や信頼性を要求する、いわゆるマルチメディアデータが扱いやすい。

しかし、このようなデータはデータ量が大きいため、帯域といった資源を大きく消費してしまう。また、大規模な利用を目的とした利用が多くなされるため、そのようなアプリケーション形態の場合、一対一の接続を複数張ることは現実的ではない。そこで、本稿においては、ATM の機能の一つであるマルチキャスト機能を用いて、多人数通信を実現する。

2.2.5 ATM マルチキャストにおける通信形態

前節で示したように ATM のシグナリングプロトコルにおいて現在提供されているマルチキャスト接続は、一対多の単方向接続のみである。従って、多対多の双方向通信を行う場合は前述した 3 つの接続タイプを組み合わせる。またそれらを行う場合の接続形態として、VC mesh と MCS(MultiCast Server) を用いる方法の 2 つが提案されている。[9]

図 2.10,2.11 にそれぞれの形態を示す。

VC mesh

VC mesh とは、マルチキャストグループに対してデータの送信を希望するホストが、自分のホストを除くすべてのグループの参加者に対して、一対多の単方向接続を確立する方法である。

MCS(MultiCast Server)

MCS(MultiCast Server) とは、マルチキャストサーバを配置してマルチキャスト通信を行う方法である。マルチキャストグループ内の各送信者は、マルチキャストサーバに対して一対一の単方向接続を確立し、データを送信する。マルチキャストサーバは、マルチキャストグループに参加するすべてのホストに対して、一対多の単方向接続を確立する。そして各ホストから受信するデータを一対多の単方向接続に送信することにより、マルチキャストグループに対するデータの送信を代行する。

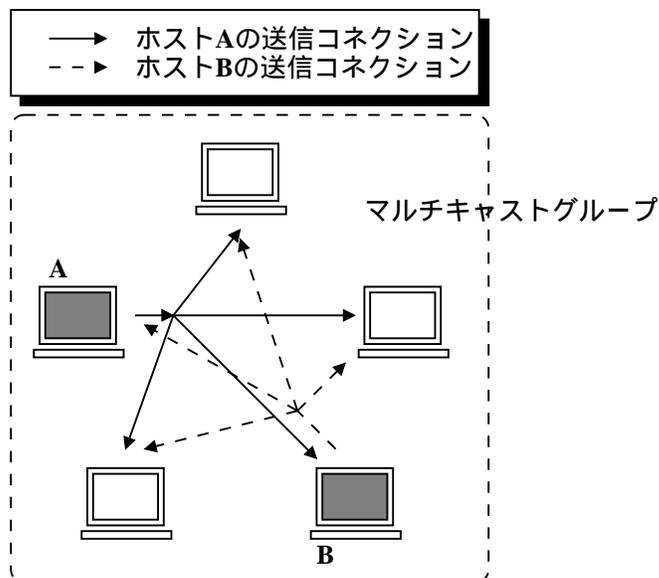


図 2.10: VC Mesh

2.2.6 VC mesh と MCS の比較

VC mesh は MCS において発生する中間の AAL_SDU の再構成や中間での処理を行わないために、MCS に比べスループットは高く、エンドノード間でのレイテンシは低くなる。また、輻輳に関してデータパスに注目して考えると、VC mesh のほうがよい。なぜなら、MCS では明らかに一つのポイントに輻輳が発生しがちになってしまうからである (図 2.12)。

しかしながら、各グループのノードは、それぞれ送信者あたりの VC を確立しなければならないため、VC mesh は資源消費は高くなる。つまり、接続を確立する ATM ポート上における状態情報、AAL サービスインスタンス、ベンダ依存のアーキテクチャによるバッファリングされるバッファなどのボード上メモリの消費である。一方、MCS では送信者の数に依存せず、入力と出力の 2 つの VC のみで実現できる。また、VC の割り当てに関連した資源も、MCS を用いた手法のほうが消費を抑えることになる。これは ATM プロバイダが VC 一本あたりに課金するようなネットワークを利用する場合などに特にメリットを持つ。また、大きな帯域を消費しがちなメディアを利用する場合にも効果が大きいと考える。

また、シグナリングの負荷に関して考えると、MCS は受信者が動的に変化する集合であるという利点がある。マルチキャストグループのメンバシップは常に変化する (リーフ

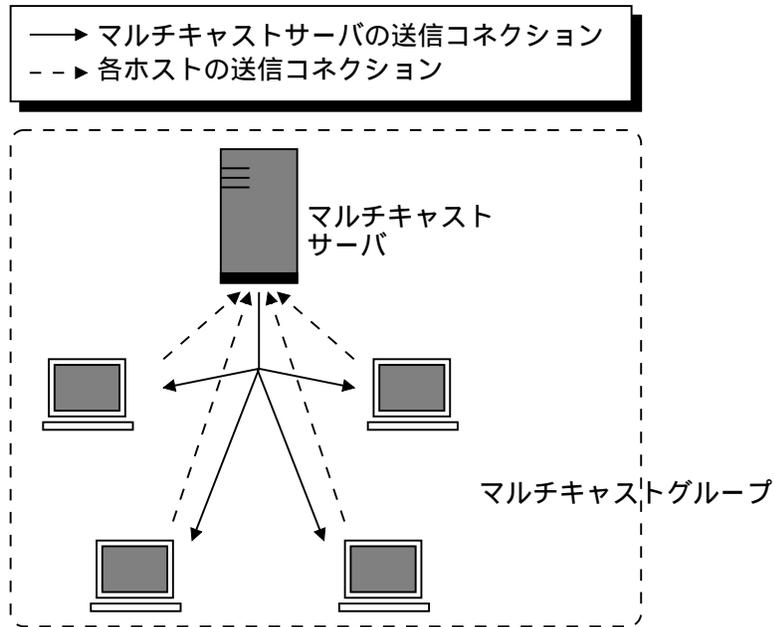


図 2.11: MCS(MultiCast Server)

ノードは付け加えられたり、はずされたりする必要がある)。MCS を利用するとき、MCS へシグナリングイベントを発してエンドノードと中間ノード間の一对多 VC だけを変更することによりメンバシップが変更可能となる。しかし、VC mesh では、シグナリングイベントは各トラヒックの送信元のノード側へ送られる。このため、シグナリングイベントに対する負荷をどちらにするかはトレードオフとなる。また、変更後の安定をするためのグループの接続性のための時間もレイテンシとして細かく分類して考えることもできる。

このように、VC mesh と MCS では、各手法とも長短所があり、一概にどちらがよいとは言えない。例えば、利用する人々によってレイテンシという意味でも、エンドノード間のパケット時間であったり、メンバシップの変更によって、あるメンバがあるグループへ移動するまでにかかる時間であったりする。そのため、最終的な選択はマルチキャストトラヒックを生成しているアプリケーションの性格に左右される。表 2.2 に VC mesh と MCS の比較を示す。

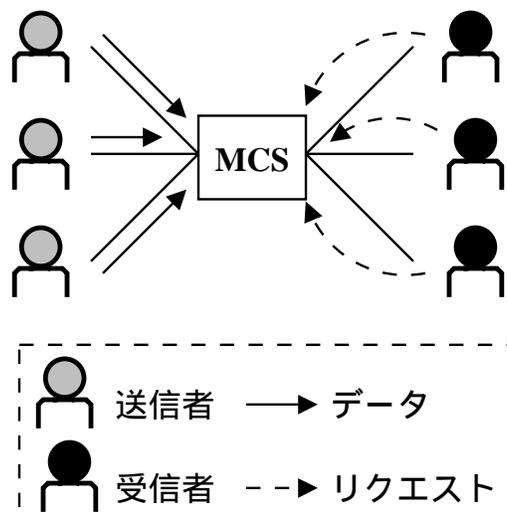


図 2.12: MCS への負荷集中による輻輳

	VC mesh	MCS
スループット	比較的高い	中間での処理などにより低くなる
レイテンシ	小さい	同上
輻輳	エンドノード付近で発生しがち	中間ノード付近で発生しがち
資源消費量	比較的多い	アプリケーションによっては小さくなる

表 2.2: VC mesh と MCS の比較

2.3 マルチキャスト通信における問題点

マルチキャスト通信は、全員に同じデータを送信する通信方式であり、マルチキャストメンバに属するメンバが多少でも異なるサービスやデータを要求した場合には、そのグループから脱退して、新たにグループを作成する必要がある。しかし、その場合には、グループからの脱退、新たなグループの設立といった新たな段階のための負担がかかってしまう。また、グループ内のメンバを識別する識別子が枯渇した場合、同じ要求をするメンバでさえあってもグループを分けるか、グループへの加入を断念するしかない。

このように、マルチキャストにより効率的な多人数配信が可能となる反面、それら参加者増加による細かなサービスが提供できなくなるといった問題点も発生するようになる。

次章からはこれらの問題点などに対し、柔軟に解決するための機構として MCS を用いた手法を示す。また、MCS を本稿では中間ノードと呼ぶ。

第 3 章

中間ノード

これまで、多人数配信のための効果的な手段であるマルチキャスト通信を利用するシステムにおいて効率よく運用するためには、適切なコネクション形態を用意し、その場合の資源消費やスループットなどの要素を熟慮する必要があることを述べてきた。また、それら資源消費やスループットなどの要素はそこで用いられるアプリケーション、そこで提供されるサービスに左右されることも述べた。本章では、必要とされるアプリケーションやそこで提供され得るサービスについて考え、また、その場合の効果的な実装形態についても考えることとする。

3.1 中間ノードの性質

IP マルチキャスト通信では、UDP と共に利用されることから、その信頼性の確保が重要な課題となっている。その場合、信頼性を確保するための手段として ACK あるいは NACK といった確認応答を行う手法が多く提案されている。

ACK はデータが届いた場合に送信者に向けて送られるデータであるが、マルチキャストされたデータに対して必ず応答するようにしてしまうと、一度に送信者にデータが返ってしまい、大変な負荷になる。そのため、NACK のみを用いたり、ACK と NACK を併用する手法も提案されている。

しかし、いずれの手法においても確認応答のためのメッセージが多く氾濫し、余計な輻輳の発生や輻輳の拡大を招くという恐れがある。そのため、これらの確認応答のメッセージを送信者と受信者の間で集約したり、メッセージの発生を抑制する機構が必要とされる。

このような機構を組み込む特別なノードは、送受信者間の中間に存在して適切な処理を

行うため、中間ノードと呼ばれている。

また、一方データフォーマットを変換したり、映像に処理を加えたりするなどのサービスを提供する場合に、ネットワーク中に変換装置を用意して、適切なコネクションを確立することにより実現する。この場合の変換装置も中間ノードと呼ばれる。

前述した MCS もネットワーク中でサービスを提供するノードであるから、中間ノードの一つであると考ええる。

このように中間ノードは、ある利用法における従来のプロトコルの問題点を解決するための機構と、その利用法におけるサービスの追加の 2 つの側面を提供するノードであるといえる。

3.2 マルチキャストを利用するアプリケーション

マルチキャスト機能が有効なアプリケーションとしては以下のようなものがある。

- ビデオ会議システム
- テレビ電話システム
- VoD システム

本稿では、このようなリアルタイム性や高品質な映像などが要求されるマルチメディアデータを扱うこととする。

ビデオ会議システムなどでは遠隔地間の会話や状況をより分かりやすく表現するために、複数の画面を合成して表示したり、ポインタや手書文字などの任意のパターンや時間表示やロゴ等を画面にインポーズするなどのサービスは有効である。また、VoD システムではサーバが対応していないフォーマットへも変換する機構などは利便性を高める事となる。

1 つのセッション中でやりとりされるデータに、一部のメンバがこのようなサービスを要求する場合、一般的なマルチキャスト通信では別々の通信として扱われるようになるが、それでは多くの部分で同じデータが無駄に重複して流れることとなる。

そこで、「中間ノード」という特別なノードを用いて、マルチキャスト通信を拡張し、ユーザの様々な要求に対してより柔軟に対応する機構を考える。

3.3 中間ノードによるサービスの提供

中間ノードは、マルチキャストグループへ参加することにより、データ通信を仲介し、データにサービスを付加した後、要求するクライアントへ再送信するサービス提供ノードである。データを中継するため、ネットワーク的に遅れが発生してしまうことや、データが集中してしまう為、広い帯域を確保しなければならないという問題が考えられるが、汎用的なサービスを提供するためには、中間ノードを用いる手法が効果的である。

また、中間ノードを用いない場合では各ノードが要求され得るすべての機能を持っておかなければならず、機器自体が高価になってしまったり、汎用性が失われてしまうことになる。

終端の装置には必要最低限のサービス提供機能のみを要求し、それ以外を中間ノードで提供することにより、ユーザには安価なシステム作りが可能となる。中間ノードに機能を集中すると、中間ノード自体は高価になるが中間ノードの機能を分け、それを1つのものとして管理することにより各中間ノードパーツは比較的安価になる。サービス提供者は必要な機能のみをシステム内に組み込むことで、全体のシステムとしてのコストを下げ、ユーザには安価な装置とサービスを提供することが可能となる。

3.4 中間ノードにより提供するサービス例

動画や音声を扱うシステムにおいて有効であると考えられるサービスを以下に列挙する。

- 画面分割サービス

ビデオ会議などで、会議に参加している全拠点の画面をモニタ上に分割して一覧表示するサービス。等分割に表示したり、話者を協調して表示するなど可能となる。

- トランスコードサービス

リアルタイムエンコーダにより、MPEG2 や Real 形式にエンコードしデータ量を小さくしたり、対応するクライアントのプレイヤーに適合させるサービス。

- データ記録サービス

会議のデータなどを大容量の記憶装置を用いて保存するサービス。

- 映像効果付加サービス

画面切替の際の特殊効果や、時間表示、スポンサーなどのロゴマークなどの挿入など画像に効果を加えるサービス。

- 一時停止サービス

映像の配信途中で一時的に停止を行いたい場合に、その間のデータを記録するサービス。停止解除により、続きを受信したり、元のタイムシーケンスに戻って受信することが可能となる。

- ビデオオンデマンドサービス

VoD サーバなどを中間ノードの機能として組み込むことによりビデオオンデマンドを実現するサービス。

3.5 中間ノードを利用した配信形態

中間ノードを用いた場合、上記のようなサービスによってその配信形態が異なる。その場合の配信形態と最も適すると考えるサービスを併記する (図 3.1)。

- 一般的な配信形態 (該当サービス : エンコード・デコードサービス, 映像付加サービス, 一時停止サービス)

各送信者と受信者の接続を中間ノードを介した形で確立し、送信されたデータに何かしらの処理を施し、受信者へ送信しなおす。

この形態では、参加するノード間で全て接続を確立させて利用するようになる。そのため、中間ノードへ負荷が集中してしまうために、多くの接続を確立することは難しくなる。また、中間ノードを経由するための送信遅延及び処理遅延が発生するため、そこで提供される付加サービスとのトレードオフになる。

- 選択的配信形態 (該当サービス : データ記録サービス, VoD サービス)

この形態では、必要とするノード間で適宜、配信を選択するため、無駄な接続を確立する必要がなく、そのため中間ノードへの負荷は一般的な配信形態と比較して多くはない。シグナリングによって中間ノードを介したエンドノード間の接続確立をその度毎に行う必要があるため、そのための遅延が発生するが、帯域といった資源を効率よく利用するには、この形態が望ましい。

ただし、全てのノード間を選択した場合には一般的な配信形態と同じものになってしまう。また、VoD サービスにおいては対象とする送信者は主に中間ノードとなる。

この形態では、必要な接続のみ確立するため、送信者と中間ノード間で確立する接続数が減少し、また、同様に中間ノードと受信者間でも接続

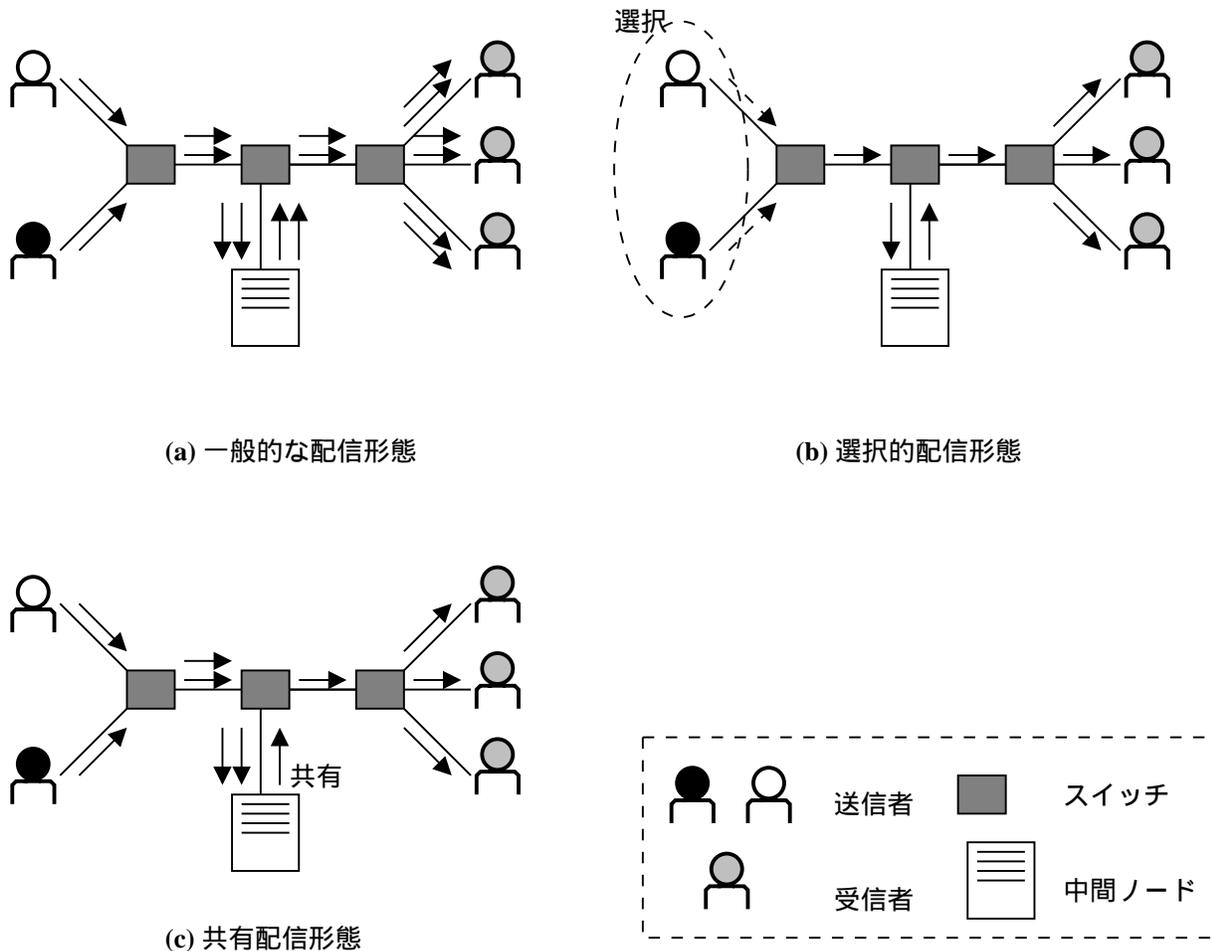


図 3.1: 配信形態

ン数が減少するため、上記の形態に比べ、帯域といった資源を比較的、有効に扱うことができる。

- 共有配信形態 (該当サービス : 画面分割サービス)] 送信者から送られたデータを中間ノードで集約し、セルレベルあるいは画像レベルで個別のものと変換し、受信者へ送信しなおす。共有させるデータは選択的な配信形態となるが、中間ノードと受信者間で確立されるコネクション数は少なくなる。

この形態では、中間ノードと受信者間で確立するコネクションが少なくなり、アプリケーションによっては中間ノードと受信者間は1つのVCだけの確立で済むことができる。

3.6 中間ノードが存在しない場合の各サービス

前述したサービスを中間ノードを用いずに実現する場合には、送信者側、あるいは受信者側でなんらかの処理を行わなければならない。

- 画面分割サービス

本サービスは、どこかで複数のデータを一つのものとしなければならないので、送信側で処理する場合には、まずどのノードにより処理するかを決定した後、そこへデータを集約させ、一つのデータとした後、各ノードへマルチキャストにより転送される。このサービスの場合、まず、どの送信者の元へデータを集約するか、どのノードがこの機能を実現できるのかなどの判断をする必要があり、大変困難である。

また、受信側で処理する場合には全てのデータを一度受信した後、画面を構成し直して表示する機構が必要となる。この場合もこのサービスを利用し得る全てのノードがこの機構を保持している必要があり、大変困難である。

- トランスコードサービス、データ記録サービス、一時停止サービス

これらのサービスでは、各送受信ノードが実現機構を持っていれば実現できる。トランスコードサービスに関しては、最近では1ボードでエンコード・デコード機能を持ったハードウェアもあり、また計算機的能力が向上したため、ソフトウェアエンコード・デコードも十分に可能となってきた。そのため、中間ノードがない場合にも比較的实现は容易である。しかし、大規模なネットワークにおいて、このサービスを必要とするであろう大多数のユーザ全員がこの機能を持つことを要求することは難しい。

また、データの記録や一時停止サービスに関しても、近年の計算機環境の低価格化で、比較的安価に大容量の記憶装置を入手することが可能となっている。しかし、これもユーザ全員に要求することは難しく、また共有可能なデータであっても、個別に保存されると再利用性が薄れてしまう。

- 映像効果付加サービス

本サービスも、付加する映像や特殊効果にもよるが、このサービスも実現機構を持ったノードにデータを一度集約させるか、また送受信ノード全てにこの機能を持たせなければならないことになる。

- ビデオオンデマンドサービス

本サービスは、ユーザの要求により発生するサービスのために、常に送信者の立場になる。また、ユーザからいつ要求が起こるか分からないために、常にネットワークに接続されていなければならない。また、多くのユーザが多くの記憶領域を消費して、コンテンツを保持しておかなければならない。また、ビデオ配信のための機構や要求の集中による負荷など一般のユーザに要求することは難しい。

これらを見ると、大規模なネットワークにおいて、そこに存在するユーザの様々な要求に対して柔軟に対処するため、網側でサービスを提供する中間ノードの存在は重要である。

3.7 中間ノードの設計

一般的に中間ノードは単体で動作し、その多くは単一のサービスしか行なわない。複数存在する場合においても同様であるが、中には各中間ノードが協調し合い、また、なんらかのメッセージにより相手を制御する [10]。

しかし、大規模なネットワークにおいて多くのユーザの要求に対し、柔軟に対応する機構を構築するには前述したような多くのサービス提供機構が必要となってくる。また、単一の中間ノードに全ての機構を持たせることは、処理能力や利用可能帯域といった面で問題となる。そこで、中間ノードの設計には複数の存在を許容し、且つ複数のサービスを提供できる複数の機構の組み込みが必要となる。

3.7.1 中間ノードのモデル

中間ノードは前述したサービスの他にも暗号化サービスや認証サービス、異種間ネットワークとのインタフェース機能など多くの部分でシステムを拡張可能である。そのために中間ノードの設計としては各機能をモジュール化して単純化した構成が必要となる。

図 3.2 に中間ノードのモデルを示す。

3.7.2 複数の中間ノードの管理

提供されるサービスが多くなると、一台の中間ノードに要求が集中してしまい処理に変な負荷がかかることや、帯域が不足するなどの問題が発生する。そこで複数の中間ノードに分散して処理を行わせることで、負荷の分散を計ることができる。

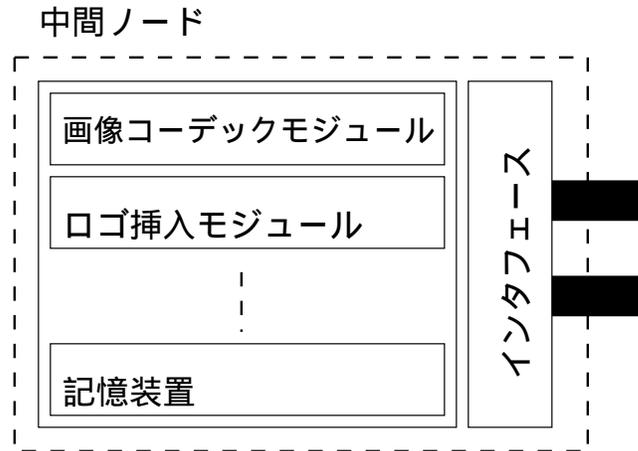


図 3.2: 中間ノードのモデル

各中間ノードが協調しあう設計

複数の中間ノードが通信しあって動作する場合、別の中間ノードのネットワークアドレスや提供できるサービスを知っておかなければならないために、情報のやりとりを適宜行なわなければならない。また新たに中間ノードが加わる場合、全ての中間ノードに対してその情報を伝達し、各中間ノードはその情報を常に保持しておかなければならない。さらに、サービスの提供要求が発生する度にその負荷に関して計算し、適切なサービスの振り分けなどを行う必要がある。

この場合、情報伝達の遅延や各中間ノードの処理負荷が大きいなどの問題が発生する。

また、ユーザが独自に判断して中間ノードへの要求を振り分けることも考えられるが、誰が、いつ、どのような理由で用いるか分からない大規模なネットワークにおいて、これは効果を成さないと考える。

集中管理による設計

JAIST VideoLAN システムにおいては、家電機器を対象としていることもあって、各ノードがストレージやメモリといった情報資源を保持しないで、管理や情報の共有を計ることが可能な集中管理法を用いている。中間ノードに関しても、他のノードと同様に集中管理ノードにより管理することとした。これにより、複数の中間ノードが存在しても情報伝達の遅延による問題が発生しにくくなり、処理の振り分けや各中間ノードの情報を保持する必要がなくなる。

3.8 中間ノードの有無による評価

3.8.1 コストによる比較

送受信者の数を n とした場合、各送受信者に何かしらの機器を必要とするならば、その機器の価格を $Cost$ として、システム全体でこのサービスを利用しようとするとその機器だけで、

$$\text{サービス付加にかかるコスト} = Cost \times n$$

かかることになる。一方中間ノードを用いる場合は中間ノードに 1 つ配置すればいいだけなので、コストは単純に $Cost$ となる。

3.8.2 資源消費量による比較

中間ノードを加えない場合、中間ノードを加える場合、中間ノードの機能を全て各ノードが持っている場合の 3 つの場合について考え、以下のように条件を付けて、トランスコードサービスを利用する場合を例に、それぞれにかかる資源消費量の違いを計算する。

- 送信者の数は 1、受信者の数は n 、中間ノードの数は 1 とする
- VC が終端スイッチまで分岐せずに、通過するパスは最小
- 終端スイッチから受信者までは 1 ホップ分の資源消費をする
- パスは 1 ホップに対し資源消費量 5 がかかるが、中間ノードまたは各ノードがトランスコードを持っており、そこを通過すると資源消費量は 1 になるとする
- 送信者から受信者までのホップ数は $hop_{end-end}$
- 中間ノードと送信者間のホップ数は $hop_{sender-in}$
- 各ノードがトランスコードを持っている場合は送信者間のホップ数は 0 とする

中間ノードを用いない場合では、終端のスイッチまでのホップ数に対する資源消費と受信者までの資源消費の合計となり、

$$RC_{without-in} = 5 \times (hop_{end-end} + n)$$

となる。一方、中間ノードを用いると、最も良い結果が得られるのは送信者のところに中間ノードがあり、送信者と中間ノードの間では資源を消費しない場合で

$$RC_{with-in_{best}} = hop_{end-end} + n$$

また最悪の場合は、送信者と受信者の間のパスとは違う方向へ新しいパスを作成する場合で、その場合には中間ノードまでの往復ホップ分、余計に資源を消費することになる。

$$RC_{with-in_{worst}} = 5 \times hop_{sender-in} + hop_{sender-in} + hop_{end-end} + n$$

各ノードがトランスコーダを持つ場合は、

$$RC_{every} = hop_{end-end} + n$$

となる。

まず、スケーラビリティの面から比較するために、受信者数を 1000 人、送信者数は 1 人として、中間ノードを利用した場合の送受信者間のホップ数の変化による資源消費の変化を調べ、また、中間ノードの配置位置による資源消費の変化を調べる。

図 3.3 に示すように、送受信者間のホップ数を 1 から 1000 まで変化させるとき、中間ノードを用いない場合と、用いる場合の最高と最悪の場合の資源消費は変化はこのようになる。ここで中間ノードを用いる場合の最悪の場合、つまり中間ノードの存在がホップ数を多く増加させてしまうことについては、中間ノードがない場合にかかる送受信者間のホップ数分を最大とした。

この図から、資源消費量の面から、大規模なネットワークにおいては特に中間ノードが有効であることが分かる。また、各ノードがトランスコーダを持つ場合、送信者が 1 なので、送信者と中間ノード間の距離が等しく計算されるため、中間ノードを用いた場合の最高の場合と同じ結果となる。しかし、複数の送信者が存在する場合には中間ノードを用いる場合の方が資源消費は多くなる。

また、図 3.4 では、送信者と中間ノード間のホップ数の変化による資源消費の変化を示している。ここでは、送信者と中間ノードのホップ数は少ない方がよい結果になる。これは中間ノードによりデータ量を小さくして資源消費量を抑えるためであり、トランスコードサービスのようなサービスにおいては特に中間ノードまでの距離が重要となる。しかし、各ノードがトランスコーダを持つ場合には、これらに影響されないために、全体の資源消費量は少なくなる。

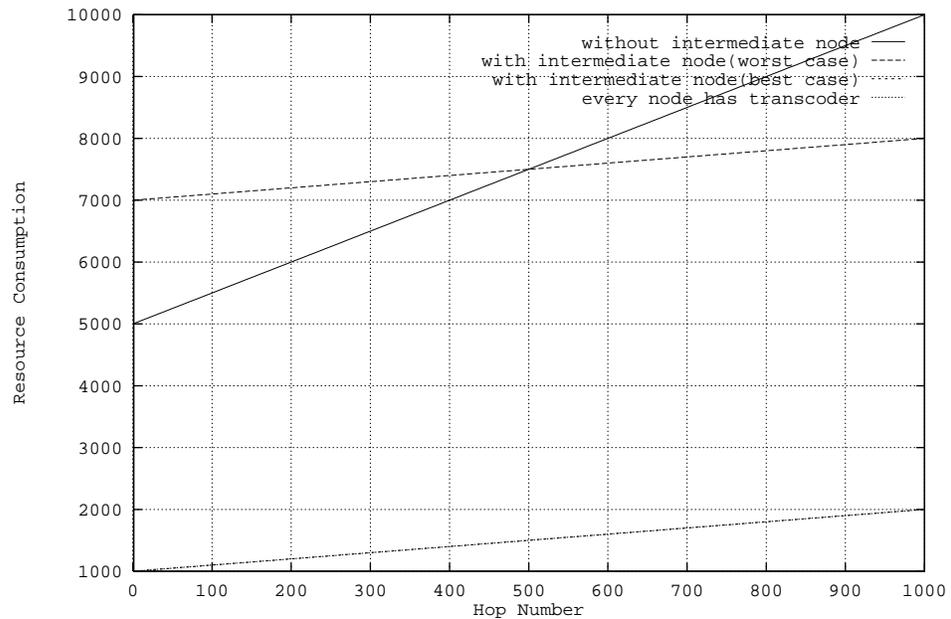


図 3.3: 中間ノードの有無に対するネットワーク規模と資源消費量の関係

3.8.3 価格と資源消費の関係

資源消費量だけを注目すると、各ノードが中間ノードが提供する機能を持つことで大幅に消費量を抑えることができることが分かる。しかし、全てのノードがこの機能を持つと、価格が大幅に増加する。例えば、送信者が 1 の場合はそれほど影響を及ぼさないが、大規模なネットワークにおいて、受信者者となり得るノードは送信者にもなり得る。また、トランスコードされたデータなどは両端でこのような機能を持つことを要求される場合もある。

3.9 中間ノードの効果

中間ノードはネットワーク中でデータにサービスを付加することで、そのサービスを利用しようとするノードに特別な機構を持つことを要求せずに実現できるため、新しいサービスが加わる際にも中間ノードだけの変更だけで実現できる。また、中間ノードが複数ある場合でも、いずれかの中間ノードがその機能を持つことで実現可能で、その拡張性も高い。そのため、マルチキャスト機能により得られるスケーラビリティや資源の効果的

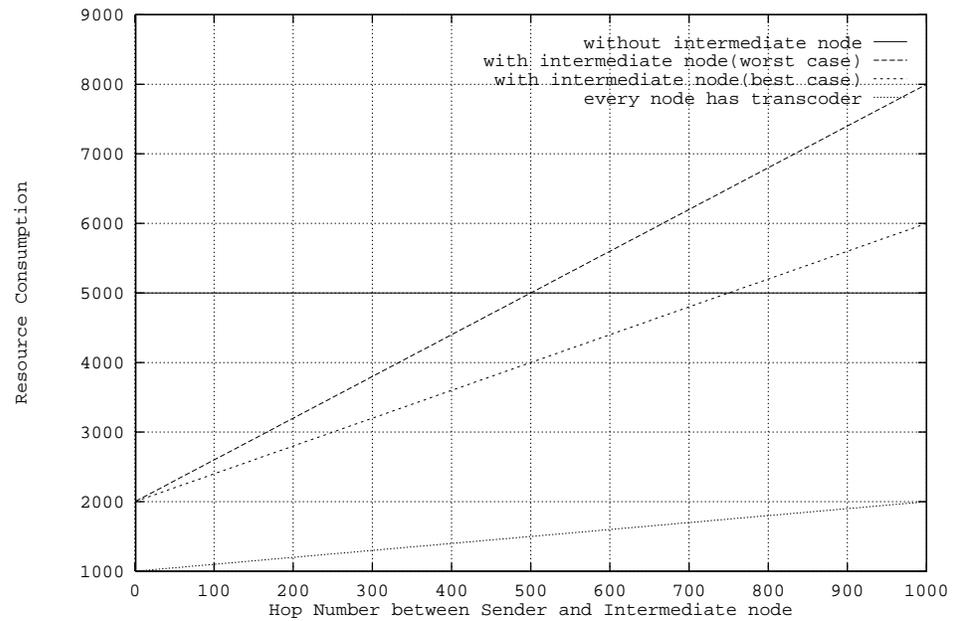


図 3.4: 中間ノードの有無に対する送信者と中間ノードと距離による資源消費量の関係

な利用といった長所をスポイルすることなく多くのユーザの要求に答えることが可能となる。特に大規模なネットワークを対象とし、さらに動画や音声といった、いわゆるマルチメディアデータの利用は広い帯域を消費しがちで、このような環境において中間ノードによるサービスの提供はあらゆる面で効果を発揮する。

次章からは、このような環境の実例として JAIST VideoLAN システムに有用なサービスを挙げ、それを中間ノードにより提供する実現方法について述べる。

第 4 章

実システムへの適用

これまで、大規模なシステム運用をマルチキャスト通信を用いて行う際に発生し得る問題に対する解決機構や、そこで有用なサービス提供機構などについて中間ノードを用いる手法を述べてきた。本章では、本学で稼働中の JAIST VideoLAN システムにこれらの機構を組み込み、適用させる手法について述べる。また、本システムに対して有効であると考えられる機構やサービスについても追記することとする。

4.1 JAIST VideoLAN システム

4.1.1 システム構成

JAIST VideoLAN システムの構成は図 4.1 のようになっている。送信者がビデオカメラにより映した映像を受信者がテレビ等で受信して視聴することができる。

基幹に ATM ネットワークを用いており、これとテレビカメラやテレビのような IEEE1394 コネクタを持つ端末とをターミナルシステム (以下、TS:Terminal System と表す) を利用して接続する。

本システムの特徴はユーザ側には TS 及び、ビデオカメラあるいはテレビのような家電機器しか必要としないで、容易にテレビ会議やテレビ電話のようなシステムが構築可能となることである。TS の接続状況などの情報は管理マシンである資源管理エージェント (以下、RMA:Resource Management Agent と表す) が集中管理する。また、ビデオカメラで撮った映像を DV データとしてそのままネットワークに流すことが可能で、より家電機器と親和性の高いシステムになっている。

操作はブラウザあるいは専用のアプリケーションによるセレクト GUI が用意されてい

る。画面には各機器と番号の組のウィンドウとそれに対する番号のマトリックスが表示されており、オペレータはこのマトリックスの交点を指定することで任意の送信者、受信者を選択することが可能となる。このように一般的なマトリックススイッチャの形式を採用することで、小さなウィンドウスペースでも分かりやすく、操作が容易な画面構成をとる。また、今回追加される機構を扱うためには従来の GUI 環境だけでは十分ではない場合がある。そこで追加機構に関して、それぞれ専用の GUI を用意することで、より扱い易い設計を行う。

DV カメラにより受像された映像を IEEE1394 と ATM を用いて DV データのまま無圧縮で高速に送受信するこのシステムは画像の劣化が少なくリアルタイムでデータをやりとりするため、ビデオ会議などの利用の他に遠隔モニタリングやテレビ電話などの利用など様々な用途に利用できる。このような汎用的な利用が期待できるシステムであるため、様々なユーザの要求に対して柔軟な対応を取る機構を必要とする。

これまで述べた通り、複数の人に効率的にデータを配信するマルチキャスト通信において、多くのユーザにサービスを付加して提供しようとするのに中間ノードを用いる手法は有効である。そこで、本章では JAIST VideoLAN システムに中間ノードを加えた構成を示し、その利用法や中間ノードを利用したサービスの種類や実現方法について述べる。

4.2 JAIST VideoLAN システムのアーキテクチャ

JAIST VideoLAN では、システムをユーザ (セレクト GUI)、TS、そして RMA で構成する。図 4.2 に各構成要素間関係の概念図を示す。この 3 つの構成要素は図中太線で示すように物理的に接続され、構成要素間で通信を行うために定義された各要素間プロトコルで制御メッセージを交換する。

また、各制御メッセージは ATM ネットワーク中を CLIP(Classical IP over ATM) 技術を用いて各ノード間で交換される。

4.2.1 ユーザと RMA 間のメッセージ

ユーザが RMA へサービスを要求するために用意されているメッセージを表 4.1 に示す。これらのメッセージはユーザが GUI を操作することにより、GUI が要求を解釈し、RMA へと転送される。また、RMA はユーザに対しては、なんの要求も出さないために、現行ではこの場合のメッセージは規定されていない。

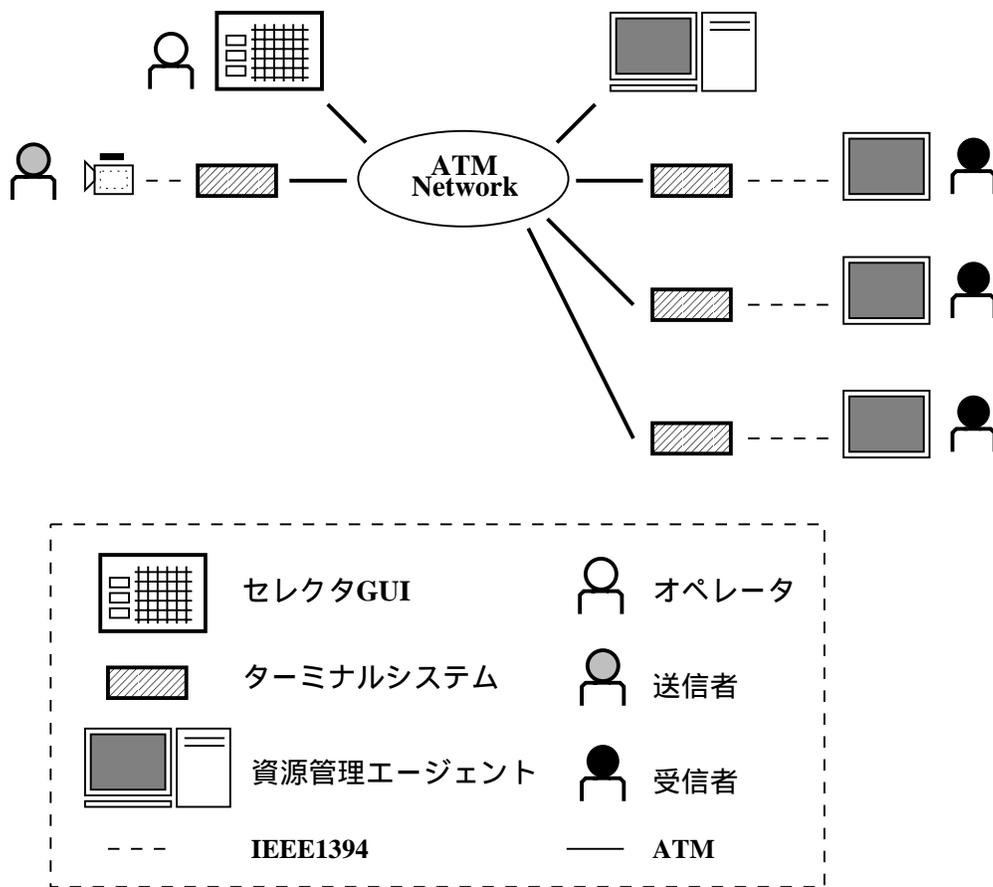


図 4.1: JAIST VideoLAN System

4.2.2 RMA と TS 間のメッセージ

RMA はユーザからの要求を受けると、必要に応じて TS へメッセージを送信する。また、TS はシステムに接続されると、自分自身を RMA へ登録する必要がある。そのため、TS 側からもメッセージが送信される。

表 4.2に RMA が TS に出すメッセージを、表 4.3に TS が RMA に出すメッセージを示す。

4.3 中間ノードの組み込み

JAIST VideoLAN システムでは全ての機器の管理を RMA が行なっている。よって中間ノードの情報を RMA に登録して、そこで管理される必要がある。JAIST VideoLAN

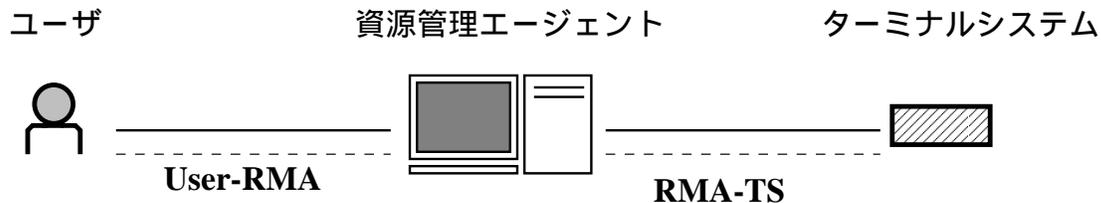


図 4.2: 各構成要素間関係の概念図

にとっての中間ノードは RMA に管理されるノードの一つでしかない。もし複数の機能を分散して持つサービス提供ノードが存在しても RMA により一元管理されているためユーザからは一つの間接ノードとして見えるのである。また中間ノードは RMA に管理されることにより、物理的な位置の制限を受けないという利点もある。さらに中間ノードに負荷が集中しないように要求を分散させるなどの処理も可能となる。

4.3.1 中間ノードと RMA 間のメッセージ

前述の通り、中間ノードと RMA でやりとりをするメッセージは中間ノードの登録のためのメッセージとさらに RMA が行なっていた TS への接続要求の代行要求である。RMA が中間ノードを介した接続の要求を出すことも考えられるが、それには TS(送信側) から中間ノードへの接続要求と中間ノードから TS(受信側) の 2 度の接続要求を出す必要があり、さらにそのコネクションに対して、どのようなサービスを要求するかメッセージも別に、あるいは合わせて送付する必要がある。一方、RMA が中間ノードへ処理を依頼する場合には、送受信側の TS のアドレスとサービスを送るだけで可能となる。

このようにメッセージの単純化と処理区分の明確化によって多くのデータを一元するための高負荷のかかりがちな RMA への処理を軽減する。また、メッセージは従来より定義されている名前を利用して決定した。

表 4.4,4.5 に中間ノードと RMA、そして TS における関連したメッセージを示す。

4.3.2 中間ノードの登録

中間ノードを RMA により管理するためには以下の情報が必要となってくる。

- IP アドレス
- ATM アドレス

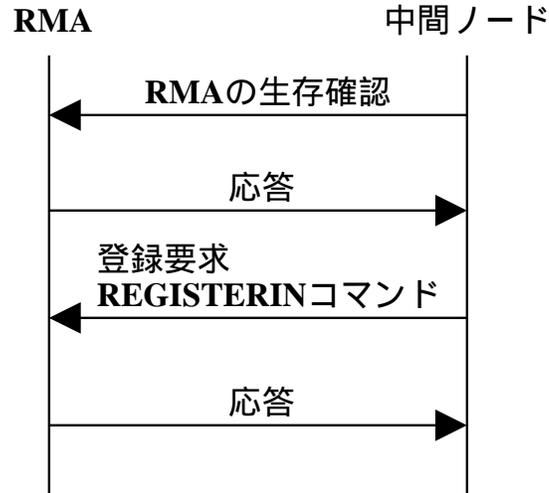


図 4.3: 中間ノードの RMA への情報登録動作フロー

- 名前
- サービス

そこで、中間ノードが JAIST VideoLAN システムに接続されると RMA に対して自身の情報登録を行う。

図 4.1に RMA への情報登録動作フローを示す。RMA へは RMA の理解する RMA コマンドを発行することにより行う。RMA コマンドを受理した RMA はそのデータベース中に情報を格納し、ユーザへ提供すべき情報を変化させる。また登録が完了した中間ノードは RMA からの要求により動作を起こす。

4.4 中間ノードの利用

中間ノードを用いない場合ではユーザの要求は RMA を経て 2 地点の TS へと渡り、各 TS 間でポイントツーポイントの通信コネクションが形成されるか、あるいは受信する側の TS がマルチキャストグループへ追加されるようになる。ここでマルチキャストグループが形成される場合は、いわゆる VC mesh の形式となる。一方、あるサービスを通信データに付加する場合には図 4.4のように RMA から中間ノードへ要求がなされ、中間ノードを経て各 TS へ要求が渡される。

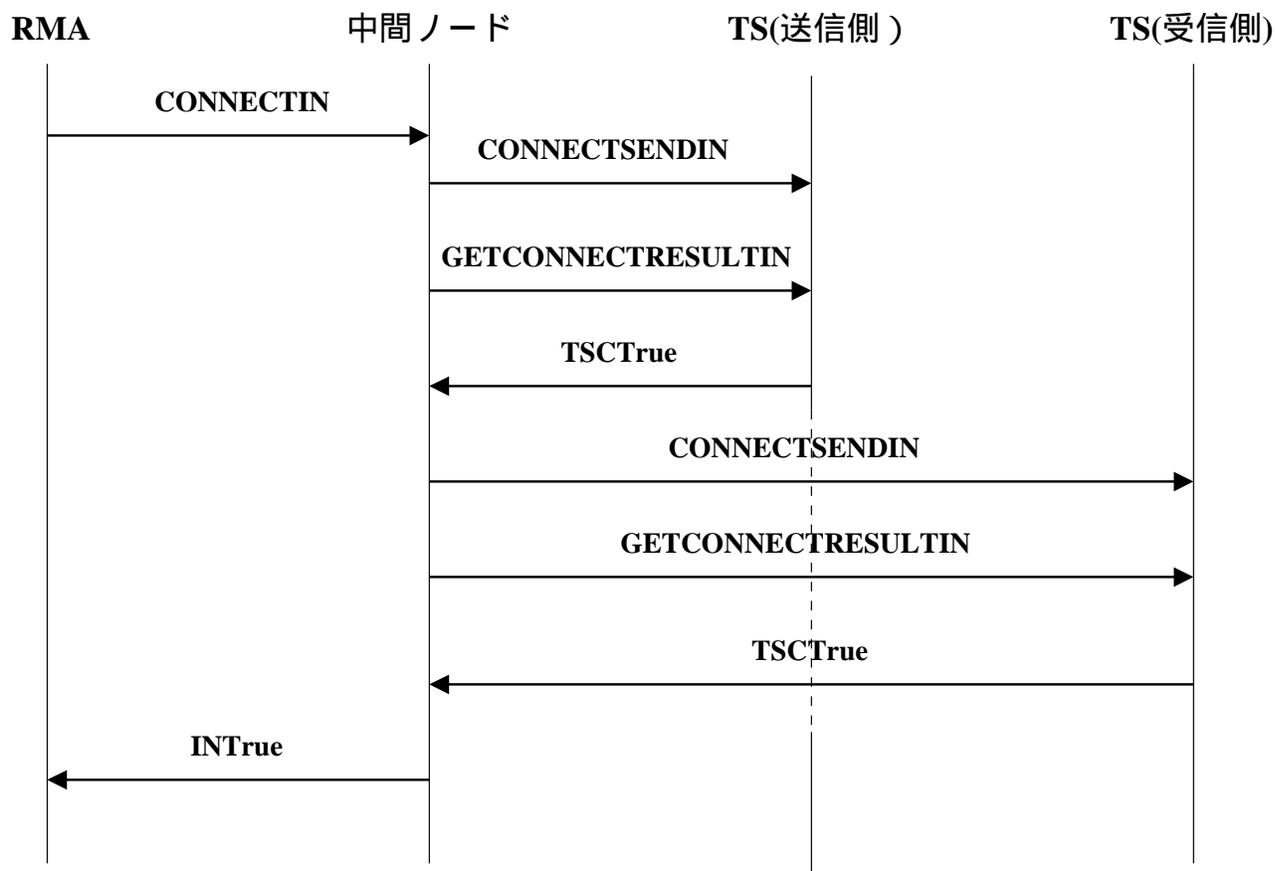


図 4.4: 中間ノードを介した送受信 TS 間コネクション設定における動作フロー

4.5 中間ノードにおける VC 接続機構

中間ノードは2つのコネクションを継ぎ合わせてエンドノード間でデータを送受信させる。そのため、送信者から中間ノードまでのコネクションと中間ノードから受信者までのコネクションを継ぐ接続機構が必要となる。(図 4.5)

そのために必要な動作は以下の段階に分けられる。

1. 到着するセルを AAL5 フレームに戻す
2. 宛先を決定
3. 送信コネクション毎のバッファのキューに入れる
4. AAL5 フレームをセルに分割し、送信

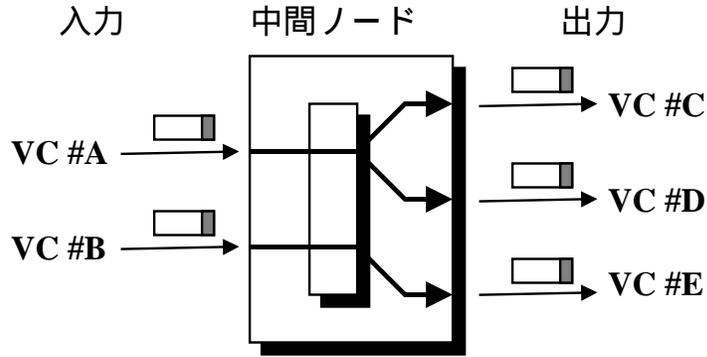


図 4.5: 中間ノードによる VC 接続機構

1,4 は主に ATM インタフェースボード上のハードウェア、及びデバイスドライバの役割であるので、中間ノード内のモジュールは 2,3 を行う。このモジュールに必要な機構を以下に示す。

- バッファ管理機構

送信コネクション毎のバッファ管理を行う。コネクション確立時に使用帯域幅に合わせてバッファを確保する。またバッファがオーバーフローを起こした場合にはフレーム破棄も行う必要がある。

- 経路選択機構

確立された送信者と中間ノード間の VPI/VCI と中間ノードと受信者間の VPI/VCI を対応付けるテーブルを用意する。

- スケジューリング機構

バッファを監視し、オーバーフローが起きないように送信フレームのスケジューリングを行う。

- シグナリングモジュール

RMA からの接続要求に応じて各エンドノード間とのコネクションを確立・解放を行う。この時、経路制御機構で用いるテーブルの登録、更新を行う。

4.6 中間ノードを利用したサービスや機構

大規模マルチメディアネットワークにおいて、特に多人数が利用する場合に強く要求されると考えるいくつかの点を挙げ、これらの項目に対して JAIST VideoLAN システムに対して中間ノードを用いて提供する手法を示し、その実現方法について述べる。ユーザへ提供されるサービスは中間ノードの一機構としてモジュール毎に利用できるようになるため、モジュールの追加により他の機能においても容易に拡張を行うことができるようになる。

また、多人数が利用する際に ATM における各 ER(Endpoint Reference) の枯渇問題が発生し得る。そこで、これを拡張する機構を中間ノードにより実現する手法を示し、JAIST VideoLAN システムに対して適用する手法についても述べる。また、この ATM の機能を拡張する手法を用いて効果的なグループの運用管理機構について提案する。

4.6.1 JAIST VideoLAN システムにおいて提供されるサービス

多人数が参加するアプリケーションは前述した通りだが、汎用的な利用法が考えられる画像圧縮サービスとビデオ会議など特別な用途に用いる画面分割サービスの2点について、中間ノードにより JAIST VideoLAN システムに提供する手法を示す。

トランスコードサービス

従来の JAIST VideoLAN システムでは DV のデータのみを扱っていた。それにより高品質の画像や音声を提供していたが、より多くのユーザに対して利用してもらうためには帯域を広くとってしまうために非効率であるという問題もある。そこで、MPEG2 フォーマットによりデータを圧縮するサービスを提供する。これにより、多くのユーザが回線を利用できるようになり、さらに MPEG といった汎用データフォーマットに変換することにより、対応する再生用クライアントが増えることとなる。また圧縮することで同じ記憶領域でも格納するデータ量が増える。

画面分割サービス

画面を分割して複数の話者を表示することで、より臨場感を持った場が提供できる。さらに物理的に接続された画面分割装置では、その装置の能力により提供できる分割数が限定されてしまうことや、表示する場所が限定されることもあるが、ネットワーク的に接続された環境でこのような装置を利用するという事は、例えば、4画面分割された映像を

再度、4画面分割装置に入力し直して16分割を行ったり、出力先を変更して表示する位置を容易に変更することが可能となる。

4.6.2 ATM マルチキャスト通信を拡張する機構と提案

ERの拡張

UNI3.1における一対多の通信では、そのコネクションの確立にSETUPメッセージを用いるのは前述した通りである。表4.6と表4.7にSETUPメッセージとERメッセージのIE(Information Element:情報要素)を示す。表4.7からUNIの仕様ではERの識別子は最大15ビット(=35767ノード)しかないことが分かる。大規模な運用を考えた場合、これでは不十分であることは明らかである。この値は同一のマルチキャストコネクションVCにつき付られた各マルチキャストリーフノードの識別子のことであるから、このマルチキャストコネクションを複数用意して一つのものとして中間ノードにより管理してやれば容易にこの識別子空間を拡張することができる。記憶する値はVPI/VCIの組と総合した識別子である。図4.6にこの機能の概要図を示す。

図4.6(a)では、35767-1のノードが既に存在しているマルチキャストセッションに対して新たなノードが加入しようとしている。あと1つでERが枯渇する場合、本手法では中間ノードにそれを割りあて、その中間ノードから新たなVCを新規参加要求者間で確立する。そして、中間ノードは従来のマルチキャストセッションの情報を新規加入者にフォワードすることによりこの識別子空間を拡張する。(図4.6(b))

中間ノードはこのVP/VCの識別子の組とこの新規に確立したVP/VCの組を記憶しておくことで、これらが実現可能となる。つまり、VCの接続機構を利用することにより実現できる。

複数のVCを用いた同一グループ管理の提案

前節では、中間ノードにより複数のVCを管理することにより識別子空間を拡張した。この手法を応用して複数のグループをVC毎に管理する機構を考える。その場合、次のような利点が生まれる。

- コネクションの切替が早い

従来の手法ではコネクションの切替には、切断、確立の2つの手順が必要で、その度毎にシグナリングが必要であった。これに対し、既にコネクションが確立されていれば中間ノード内のルーティングテーブルを変更するだけで送信者が変更可能と

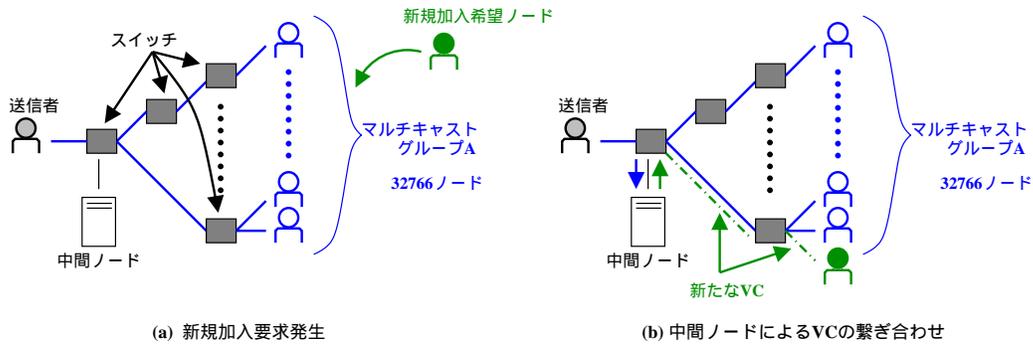


図 4.6: ER の拡張

なる。また、情報の停止に関しても、中間ノードによる情報のフォワードを止めるだけで実現できるため、グループ単位の情報の切断を容易に扱うことができる。グループからの脱退のためのメッセージのやりとりなどの遅延に対し、迅速にユーザの要求が反映できる。また多くのリーフノードが同時に脱退をするときの負荷が中間ノードで吸収できる。さらに課金される放送の切断に対しても有効である。

- 同一目的を持つ集合毎に分けることによるグループ管理の明確化

それぞれのサービスや課金システム、地域によって中規模のグループに分割することにより、管理対称を明確化させる。RMA や中間ノードに論理的なデータベースを保持して、それによりグループの管理は可能ではあるが、余計な情報を持たずにコネクション毎に管理することにより、記憶容量といった資源を効率化できる。

4.7 インタフェースの拡張

現在の JAIST VideoLAN システムにおいては、比較的小規模なビデオ会議やキャンパスレベルの講義映像の配信などの利用が対象目的として挙げられている。これらに対しては、中間ノードを用いることにより、前述した付加サービスが効果的に利用可能となる。また、本稿においては大規模な利用を想定しているため、より帯域や識別子といった資源を効果的に扱う必要があり、そのために ER の拡張や複数の VC を用いてコネクションの切替の効率化などを行う手法を加えた。

これらにより、大規模な利用が可能となるが、そのために多くのユーザの扱いが従来のマトリックススイッチャ形式のインタフェースでは難しくなる。また、サービスは様々で

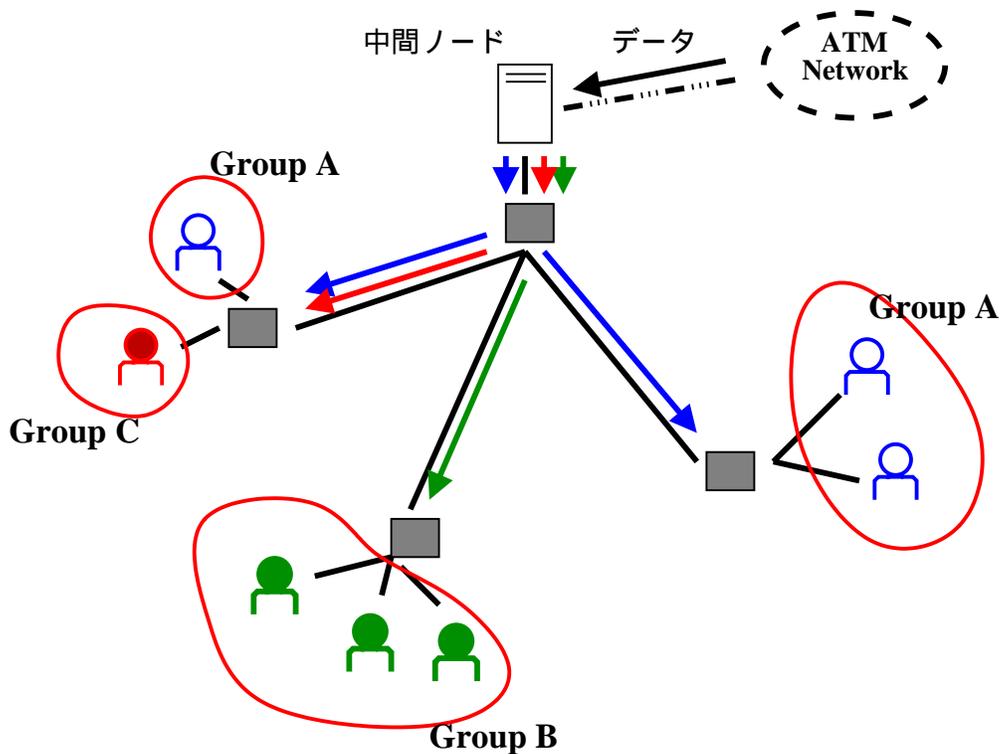


図 4.7: 複数の VC によるグループ毎の分割

あるため、単一の表現方法では多くのサービスを表現することは難しい。そこで JAIST VideoLAN では、GUI をウェブブラウザベースで行い、提供されるサービス毎に URL を指定することで対応する。

以下に本稿で簡易的に作成したインタフェースを挙げ、従来のインタフェースと比較した場合の効果を示す。

4.7.1 従来のインタフェース

従来のインタフェースは特別なサービスを付加することを想定していないために、送信者、受信者を表現するアイコンのみから構成され、またユーザが分かりやすいように、格子状に並べられて、送信者と受信者の交点のボタンを指定することで、コネクションが確立され、データが流れるようになっている。

言語は Tcl/Tk 版と Java 版が作成されており、Java 版は本稿で挙げるウェブブラウザベースで流用可能であるため、特別なサービスを利用しない場合にはこのインタフェースを利用する。

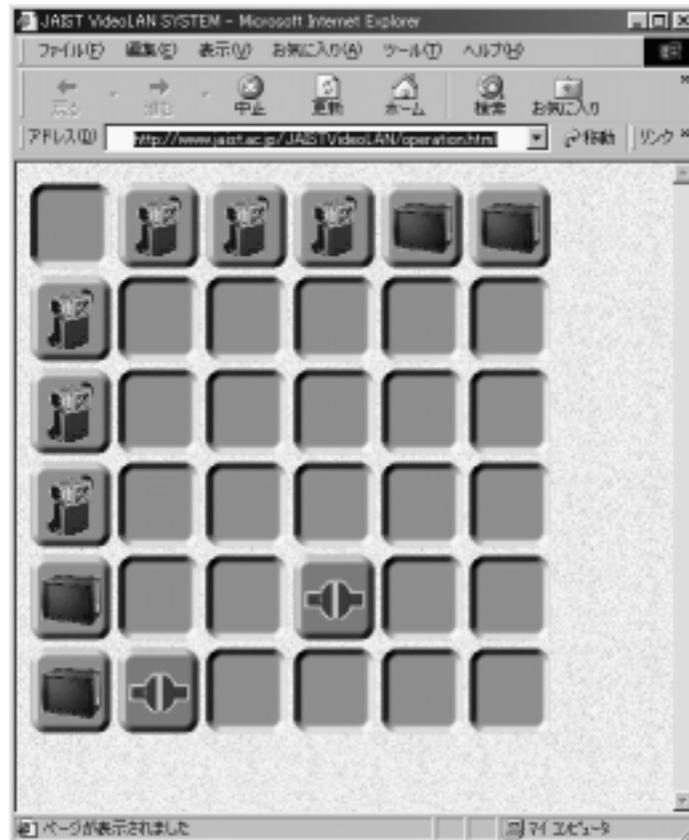


図 4.8: ブラウザを用いた GUI 例

4.7.2 提案するインタフェース

単純なインタフェースとしては図 4.10のような形式も考えられる。このような形式のインタフェースは一般的なスイッチャ機構にも見られる。また、従来からの図 4.11のような格子状に表現するインタフェースもある。

しかし、これらのインタフェースの問題点としては以下のようなものがある。

- ユーザ数が多くなった場合に全てのユーザを表示することが困難
- 特別なサービスが提供されないために、GUI が一つしか用意されていない
- 中間ノードへの明示的なデータコネクション確立機能がない

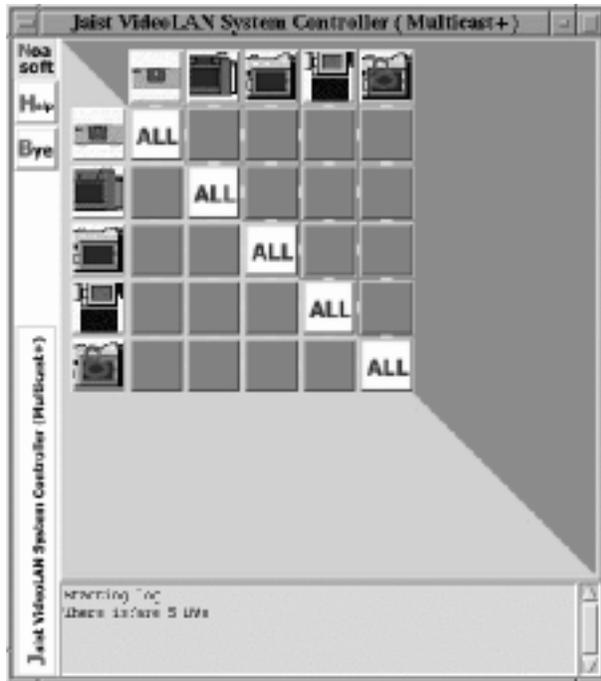


図 4.9: ウィンドウズシェルを用いた GUI 例

多数のユーザの表現

ユーザ数が多くなった場合の表示方法はなかなか難しいが、全てのユーザが別々のサービスを受けることは考えにくい。その場合、あるマルチキャストグループに属していると考えられる。そこで、マルチキャストグループを1つのノードとして表示することにより、多くのユーザを表現できる。また、その中のノードはマルチキャストグループを示しているノードを指定することにより、別の場所へ表示されることとする。

また、注目している一部のみを表示して、それがどこを示しているのかをまた別の場所に表示するなど、表示にも工夫が必要となってくる。(図 4.13,4.12)

サービス毎の GUI の提供

様々なサービスがあるため、それら全てを一つのインタフェースだけで表現することは難しい。そこで、各サービスに対してそれぞれ個別の GUI を用意する。これにより、そのサービスを分かりやすく表現することができ、利用しやすくなる。例えば、複数の画面入力がある画面分割サービスでは、図 4.14のように、中間ノードを表示して、明示的に中間ノードへ入力させる。この場合、入力として得られたデータの流れは、再び送信側の中

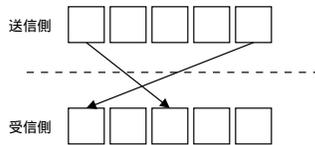


図 4.10: 並列配置インタフェース

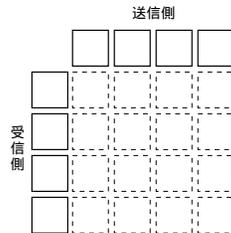


図 4.11: 格子状配置インタフェース

間ノードへと送られ、そこから出力先を指定することにより画面分割サービスが実現される。しかし、並列配置インタフェースにおいては、データフローが重なったり、見えない部分との接続をどのように表現するかなどの問題もある。そのため、格子状に配置されたインタフェースのほうが効果的である。

格子状に配置したインタフェースでは画面分割サービスは、図 4.15 のようになる。例えば、中間ノードが 4 分割機能を 1 つしか実装していなくても、その出力を入力に戻してやることで、16 分割機能が実現可能である。しかし、その場合にはユーザの要求により、あるいは自動的に新たな中間ノードを出現させてるなどの表現が必要となってくる。また、グループのメンバを別の場所に表現することによって、グループとして 1 つに表現される前述の表示方法を応用して、中間ノードと、その内部に入力される入力との関係をそれぞれ別の場所へ表示する方法も考えられる。

ユーザには見えないサービス

このように、中間ノードを明示的に表示することにより、ユーザにその利用を促す方法は中間ノードからの出力をまた入力として扱うなどの特殊な場合に有効である。しかし、多くのサービスは、暗黙の内に提供され、ユーザの要求をよりよい形で実現する。例えば、識別子の枯渇問題に対する ER の拡張などはユーザはまったく意識せず、多くのユーザが本ネットワークシステムを利用できるようになる機構である。また、MPEG へのエ

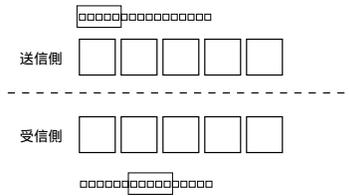


図 4.12: 並列配置インタフェースの拡張 1

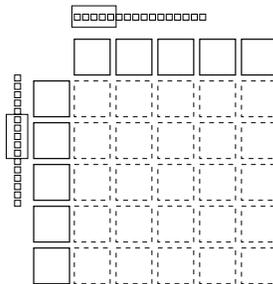


図 4.13: 格子状配置インタフェースの拡張 1

ンコード/デコード機能においても、MPEG を受信できるノードが表示されるように変化
するだけで、従来の GUI とは大きく変わる必要性は強くない。

このように、多くの部分で従来の GUI の形態がユーザに対して効果的であると考え
る。また、サービス毎に GUI を用意するにしても、それらの仕様が異なりすぎると、操作
の為の知識をユーザに強要してしまい、あるいは、使いにくいものになってしまう。このた
め、基本形態である格子状の表示及び、操作系は従来のものを利用し、追加された機能が
それでは表現が難しい場合のみ、特別な操作系を用意することがよいと考える。

4.8 形式記述法によるシステム設計

中間ノードと RMA の制御手順については状態数が多いことから、システムの動作や仕
様を明示的に表現することが重要である。そこで OSI の 7 層モデルのサービスとプロト
コルを記述することに主眼を置いた形式記述法 FDT(Formal Description Techniques) を
用いた。FDT の採用により記述の曖昧性を除くことができ、明晰で簡潔な仕様が記述可
能となる。この結果、仕様の完全性、矛盾性等の検証、仕様に基づくシステムの機能分析
が可能となる [14]。また要求仕様からコードを自動生成することにより汎用プログラミン

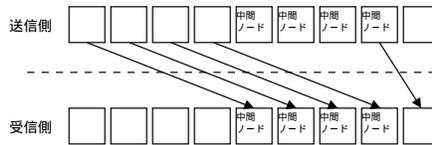


図 4.14: 並列配置インタフェースの拡張 2

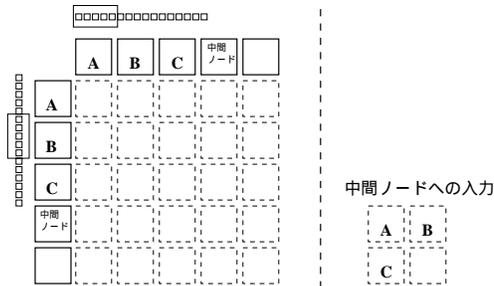


図 4.15: 格子状配置インタフェースの拡張 2

グ言語で記述するのに比べ、記述者の負担は格段に少なくすむという利点もある。

ATM マルチキャストを用いた大規模マルチメディアネットワークにおいて、RMA という管理ノードにより中間ノードを運用するシステムを以下のような手順で設計した。

1. プロトコル開発 (要求の詳細定義)
2. 詳細の仕様を作成 (設計)
3. 実装用プログラムのコーディング (作成)
4. 仕様を検査 (試験)
5. 通信ソフトウェアを運用 (保守)

1 と 2 では FDT の機能を備えており、ITU-T において交換機の仕様を規定するために開発され勧告として標準化された SDL (Specification and Description Language) を用い、3. ではプロトタイプ言語として C 言語を用いる。SDL の記述には NEC 社製の RDSPACE を使用した。4 では RDSPACE により自動的に生成可能なシミュレータによって動作を試験し、その仕様に基づいて、実際に VC++ を用いて WindowsNT 上で動作するアプリケーションを作成した。

4.8.1 SDL による記述

SDL では、システム全体の構造を「システム仕様」として記述する。そして、その中の機能を大きく分割した機能ブロックを「ブロック仕様」として記述する。また、ブロック仕様で記述されたブロックを実際に活動を行うプロセス単位に分けて、各プロセスについて「プロセス仕様」として記述する。プロセス仕様は状態遷移図として表記される。

SDL では、グラフィカルな図形モデルとして表記する SDL/GR(Graphical Representation) と、テキスト形式で表記される SDL/PR(textual Phrase Representation) のどちらかで記述可能であるが、RDSPACE では SDL/GR で記述して、SDL/PR に変換することが可能なので、ここでは記述の負担を避けるために SDL/GR 形式で記述した。しかし当然ながら、どちらの記法で記述しても全くの差異はない。

システム仕様

JAIST VideoLAN は大きく 3 つのブロックから構成される。つまり、RMA、TS、そして今回設計した中間ノードである。それぞれは互いに協調しあいシステムを構成する。

Fig.4.16に SDL/GR により記述をした JAIST VideoLAN システムの system 仕様を表す。ここでは、Fig. 4.1に示した JAIST VideoLAN の実装構造に沿った形で仕様化してある。また、中間ノードから見た場合、データのやりとりを行うのは TS と RMA なので、GUI の部分はブラックボックスとして、それ以外のノードについてブロック仕様を表した。そして、中間ノードに関してのみプロセス仕様を表した。

ブロック仕様

図 4.17に中間ノード部分のブロック仕様を示す。中間ノードは内部に interface と制御用の control、そして実際に機能する各モジュールである function プロセスを持つ。ここでは function プロセスは要求を完全に満たすブラックボックスプロセスとして規定した。

プロセス仕様

- interface プロセス

interface プロセスは TS の制御用プロセスである TSC と通信したり、RMA プロセスとのやりとりを行う。

- control プロセス

system JaistVideoLAN

```

newtype Data
  literals DVdata,MPEGdata,blank;
  operators value:Data->Character;
endnewtype Data;
newtype Function
  literals DV2MPEG,MPEG2DV,VC_Connect;
  operators value:Function->Integer;
endnewtype Function;
dcl src_ATM,dst_ATM,target_ATM,ATM Charstring;
dcl IP,TSC_name,src_TSC,dst_TSC,IN_name,Message CharString;
dcl src_nuid,con_ID,leaf_ID,nuid,offset,value,Service Integer;
signal
  REGISTERBRIDGE(target_ATM,IP,TSC_name),TSCNOTIFY(IP,TSC_name),CONNECTSEND(src_nuid,src_ATM,dst_ATM),
  DISCONNECTSEND(src_nuid,src_ATM,dst_ATM,conID,leafID),GETCONNECTRESULT(dst_ATM,conID,leafID),
  OVERLAY(conID,dst_ATM),GETNODES,PANIC,WRITEQUADLET(target_ATM,nuid,offset,value),VERSION,
  VACUUM,REGISTERIN(IN_name,IP,ATM,Service),DELETEIN(IN_name),INTrue,INFalse,TSCOK,TSCNG,
  CONNECTIN(src_nuid,src_ATM,dst_ATM),DISCONNECTIN(src_nuid,src_ATM,dst_ATM,conID,leafID),
  RMATrue,RMAFalse,TSCTrue,TSCFalse,Raw_Stream(Data),CONNECTSENDIN(src_nuid,src_ATM,dst_ATM),
  DISCONNECTSENDIN(src_nuid,src_ATM,dst_ATM,conID,leafID),GETCONNECTRESULTIN(dst_ATM,conID,leafID),
  CONNECT(src_nuid,dst_nuid),DISCONNECT(src_nuid,dst_nuid),GETCONF,GETCONNECTION,DOPOST(Message),
  OVERLAYIN,OUT_Stream(Data)CONNECTSEND(src_nuid,src_ATM,dst_ATM)

```

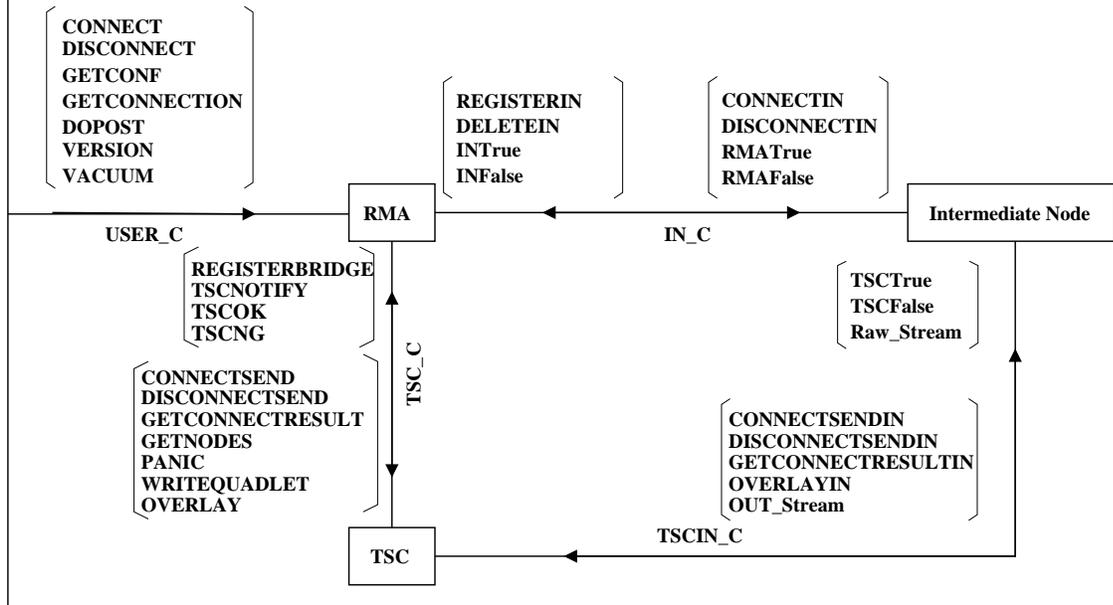


図 4.16: JAIST VideoLAN のシステム仕様

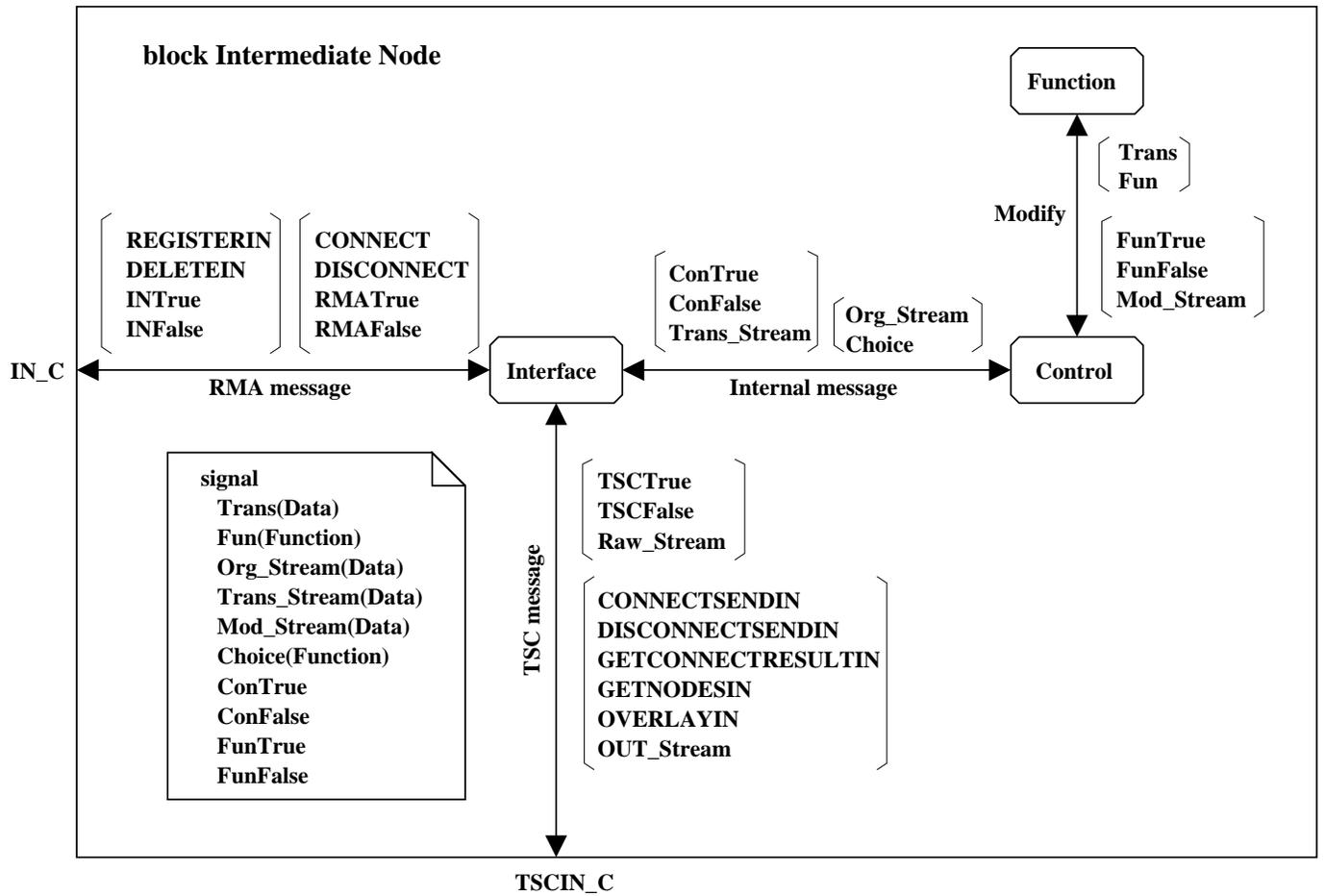


図 4.17: 中間ノードのブロック仕様

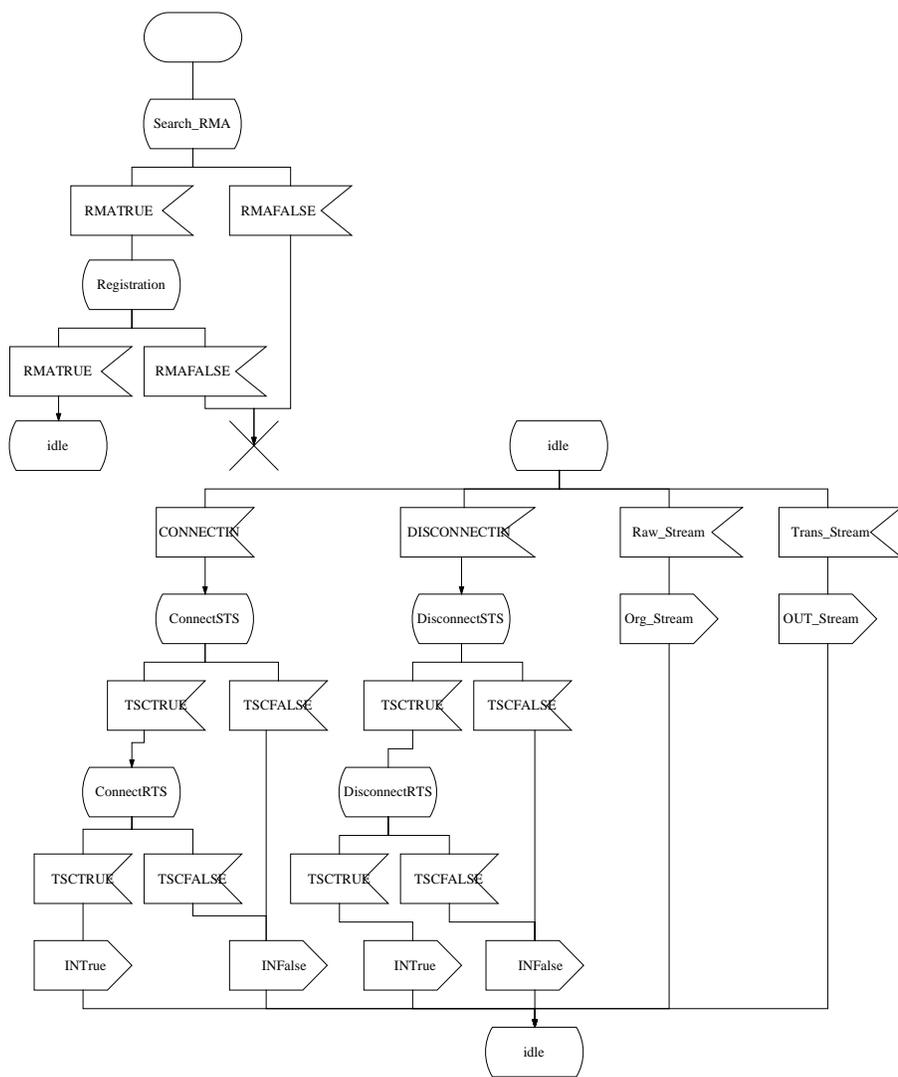


図 4.18: interface プロセス

control プロセスは RMA から受けたユーザの要求する付加サービスを各通信毎に記憶して適切な機能を選択して function に受け渡し、サービス付加後のデータを interface プロセスへ戻す機構を持つ。

記述された仕様に対する検証

RDSPACE では SDL で記述された仕様の状態遷移図に対して、以下の項目をチェックする。

- 属性
- 重複
- コネクタ
- 接続性
- 到達性
- ループ

これらのチェック機構は作成した仕様に対する状態遷移図に構文的な誤りがないかを検査し、仕様を明確化する。

図 4.20 と図 4.21 にそれぞれ interface と control のプロセス仕様に対する全ての項目に関してチェックした時の結果を示す。

まず、interface プロセス仕様に関しては、多くの FATAL エラーが発生している。これは STOP によってプロセスが終了した場合に、復帰できないことを示している。しかし、STOP により終了する場合とは、RMA を発見することができず、自分の情報を登録できない場合である。本稿においては、この場合に中間ノードが機能しないとし、STOP になると処理が停止して終了する仕様としているため、このエラーに関しては問題がない。また、この部分に関しては登録要求プロセスを何度か試行する機構を加えることにより、より柔軟性の高い仕様とすることができる。

control プロセス仕様に関しては、STOP が存在しないために警告が发っせられているが、動作に関して全く問題はない。

これらから今回設計した中間ノードの仕様は正常に動作することが確認できる。また、このことから、中間ノードと RMA 及び TS 間のプロトコル仕様が明確化されたといえる。

4.9 実装

これまで仕様化したプロトコルについて実装を行う。中間ノード自体の実装を行うには、メッセージを行う部分と、提供されるサービス機構の実装の2つが必要であるが、本稿においては、メッセージ部分のみの実装を行い、サービス機構については利用するためのインタフェース部分のみの設計を行った。また、実験はSDLからの自動生成が可能なシミュレーションプログラムによる予備実験と実際に実装したシステム実験の2つを行った。

4.9.1 予備実験

RDSPACEにより、仕様が明確化されたSDLのプロセス仕様について、それぞれからイベント駆動型のシミュレーションプログラムを生成し、各個に実験を行った。

4.9.2 実装システム

中間ノードとRMA及び、TS間のメッセージ交換プロトコルの実装はWindows NT上で行った。また、使用言語としてVC++ version6.0を用いた。

Windows NTを用いた理由は、MPEGのコーデックなどを利用する場合、Windows用のAPIが用意された製品が多いことから開発を簡易化するためである。また、VC++を用いた理由が、オブジェクト指向型の言語であることから、ソースコードの再利用性といった利点を生かすためと、前述のAPIを扱うために有効であるからである。

本来はATMネットワーク上をCLIPを用いて、IPによりメッセージを交換するが、本稿における実験では、LAN環境においてTCP/IPを用いたメッセージの交換を行なった。また、実験はRMAのIPアドレス及び、ポート番号が既知であると前提づけ、以下の項目について行った。

- 中間ノードのRMAへの登録
- RMAからのメッセージに対する応答

4.9.3 実験結果

予備実験、及び実装システムでの実験はいずれも正常に機能した。予備実験に用いたシミュレータはC言語で生成されるため、VC++との互換性はほぼないといっているが、

メッセージの受信における正常な動作を確認できた。また、実装システムにおいても同様の事が確認できた。

4.10 評価

ここでは、実際の JAIST VideoLAN における仕様を元に、中間ノードを用いる場合と用いない場合のシグナリングコストの違い、そして、中間ノードと利用可能なサービスの数についての関係について考え、評価を行う。

4.10.1 シグナリングコスト

リアルタイム性が要求される JAIST VideoLAN システムにおいては、メッセージの授受にかかるコストも無視できない。そこで、トポロジやネットワークの規模の変化に対する中間ノードの影響を調べるため、コネクション確立の場合を例にシグナリングコストを計算する。

まずネットワークトポロジとして、線形接続されたネットワーク (図 4.22(a)) と 2 分木により構成されたネットワーク (図 4.22(b)) を考え、各トポロジにおける中間ノードの効果を調べる。他にもスター型ネットワークトポロジが考えられるが、規模拡張性の点で除外することとした。

まず、シグナリングコスト $C_{signaling}$ はメッセージハンドリングコスト C_{mh} と各ノード間の平均距離 $Dist$ の積で表し、

$$C_{signaling} = C_{mh} \times Dist$$

となる。また、メッセージハンドリングコスト C_{mh} は、各メッセージのコストを 1 とし、JAIST VideoLAN における各ノードのメッセージのハンドリング数を MH_{JVL} 、ATM シグナリングにおけるハンドリング数 MH_{ATM} の和として計算を行う。

$$C_{mh} = MH_{JVL} + MH_{ATM}$$

従来のコネクション確立は、ユーザ、RMA、TS の 3 つのノード間のメッセージのやりとりにより行われる。この時、やりとりされるメッセージの数は 5 つとなり、メッセージハンドリングコスト MH_{JVL} は 1×5 となる (図 4.23)。また、TS 間の ATM シグナリングは 1 度しか行われず、メッセージ数は 10 なので、ATM のメッセージハンドリングコスト MH_{ATM} は 1×10 である (図 2.9)。よって、全体のメッセージハンドリングコスト C_{mh} は 15 となる。

一方、中間ノードを介した場合は、ユーザ、RMA、TS、中間ノードの4つのノード間でメッセージがやりとりされ、この時のJAIST VideoLANにおけるメッセージ数は10となり、メッセージハンドリングコスト MH_{JVL} は 1×10 で、10となる(図4.24)。また、中間ノードを介した形でTS間接続が行われるためにATMシグナリングは2度必要となる。そのため、ハンドリングコスト MH_{ATM} は $1 \times 10 \times 2$ で20となり、中間ノードを介した場合の全体のメッセージハンドリングコスト C_{mh} は30となる。

また、各ノード間の平均距離は表4.8のように表される[15]。

そのため、中間ノードの有無による全体のシグナリングコストとノードの数の関係は図4.25のようになる。この図は各トポロジにおける中間ノードがある場合とない場合の接続確立を行うユーザの増加に対する全体のシグナリングコストの増加を示している。ここでは、トポロジの形態は線形に比べて2分木の方が有効であることが分かる。

この図では中間ノードが一樣に分布し、各ノードからの平均距離が全て等しい場合を示しており、中間ノードを用いる場合のほうがシグナリングコストが多くかかっている。一方、図4.26は、中間ノードを用いた場合の平均距離を半分にした場合を示している。この場合、中間ノードを用いた場合の方がシグナリングコストが低くなる。このように中間ノードの配置においては、適切な配置を行うことでシグナリングコストを小さくできる。

4.10.2 複数の中間ノードによるサービスの提供

1つの中間ノードでは、帯域やそこでの処理能力によって、利用できるサービス数は左右される。サービスに関する処理能力に関しては、ハードウェアで処理することにより、ある程度無視できる場合が多い。しかし、マルチメディアデータのような広い帯域を必要とするデータを扱うネットワークでは、データの集中するノード周辺で輻輳の原因にもなり、問題である。例えば、JAIST VideoLANにおける例を考えると、利用するデータはDVで、約35Mの帯域を必要とする。このDV接続サービスの場合、OC-3(155Mbps)で4サービスしか提供することができない。そのため、複数の中間ノードにより効果的に利用依頼の分散を行う必要がある。

図4.27にサービス要求が1000個発生した場合の中間ノードの数と接続の集中度合の関係を示す。1つの中間ノードでは4サービスを処理することとし、その負荷を適切に分散されたとすると、この関係は以下のように表すことができる。

$$\frac{1000}{n \times 4}$$

ここで、1000はサービス数、4は一つの中間ノードで提供可能なサービス数、そして n は中間ノードの数である。この図から、複数の中間ノードの配置は負荷分散に関して有効

であることが分かる。これに関しては資源管理エージェントが JAIST VideoLAN におけるネットワークの状況を監視している状態にあるため、この機能を利用して適切に振り分けることで実現可能となる。

メッセージ	意味
GETCONF	VideoLAN 内のデバイス情報の一覧を得る
GETCONNECTION	DV 機器についてのコネクション情報を得る
CONNECT	ユーザから要求される送信側と受信側の DV 機器間のコネクションを設定
DISCONNECT	ユーザから要求される送信側と受信側の DV 機器間のコネクションを解放
DEBUG	デバッグレベルの再設定
PANIC	全ての TS にリセット命令を出し、データを再収集
VERSION	バージョン情報を提供
VACUUM	データベースの履歴を消去

表 4.1: ユーザから RMA へのメッセージ

メッセージ	機能
CONNECTSEND	DV 機器間のコネクション設定を要求
DISCONNECTSEND	DV 機器間のコネクションの解放を要求
GETCONNECTRESULT	コネクション設定の結果を要求
GETNODES	TS に接続されている DV 機器の情報を要求
OVERLAY	P-MP のコネクションに対して参加要求
PANIC	TS にシステムのリセットを要求
WRITEQUADLET	DV 機器に対して制御命令の送信を要求
CONNECTIN	中間ノードを介して 2 つの TS 間のコネクション確立を要求
DISCONNECTIN	確立されている TS 間のコネクション解放を要求

表 4.2: RMA から TS へのメッセージ

メッセージ	機能
REGISTERBRIDGE	RMA に TS の登録を要求
TSCNOTIFY	TS の更新を通知

表 4.3: TS から RMA へのメッセージ

メッセージ	機能
CONNECTIN	中間ノードを介して 2 つの TS 間のコネクション 確立を要求
DISCONNECTIN	確立されている TS 間のコネクション解放を要求

表 4.4: RMA から中間ノードへのメッセージ

宛先	メッセージ	機能
RMA	REGISTERIN	RMA へ中間ノードの情報を登録
	DELETEIN	RMA へ登録してある情報を消去
TS	CONNECTSENDIN	TS と自身との接続を要求
	DISCONNECTSENDIN	TS と自身との接続の解放
	GETCONNECTRESULTIN	コネクション設定の結果を要求
	OVERLAYIN	一対多のコネクションの確立要求

表 4.5: 中間ノードからのメッセージ

Information Element	Direction	Type	Length
Protocol discriminator	both	M	1
Call reference	both	M	4
Message type	both	M	2
Message length	both	M	2
AAL parameters	both	O	4-21
ATM traffic descriptor	both	M	12-30
Broadband bearer capability	both	M	6-7
Broadband high layer information	both	O	4-13
Broadband repeat indicator	both	O	4-5
Broadband low layer information	both	O	4-17
Called party number	both	M	(5)
Called party subaddress	both	O	4-25
Calling party number	both	O	4-26
Calling party subaddress	both	O	4-25
Connection identifier	both	M	9
QoS parameter	both	M	6
Broadband sending complete	both	O	4-5
Transit network selection	both	O	4-8
Endpoint reference	both	O	4-7

M : Mandatory O : Optional

表 4.6: SETUP message content

Bits								Octets
8	7	6	5	4	3	2	1	
Endpoint reference								
0	1	0	1	0	1	0	0	1
Information element identifier								
1	Coding Standard		IE Instruction Field					2
Length of endpoint reference contents								3
Length of endpoint reference contents(continued)								4
Endpoint Reference Type								5
0	Endpoint Reference Identifier Value						6	
Endpoint Reference Flag								
Endpoint Reference Identifier Value(continued)								7

表 4.7: Endpoint Reference Information Element

トポロジ	平均距離
linear	$\frac{(n+1)}{3}$
binary tree	$\frac{2n \log_2 n - (n-1)}{n+1}$

表 4.8: 各トポロジにおける平均距離

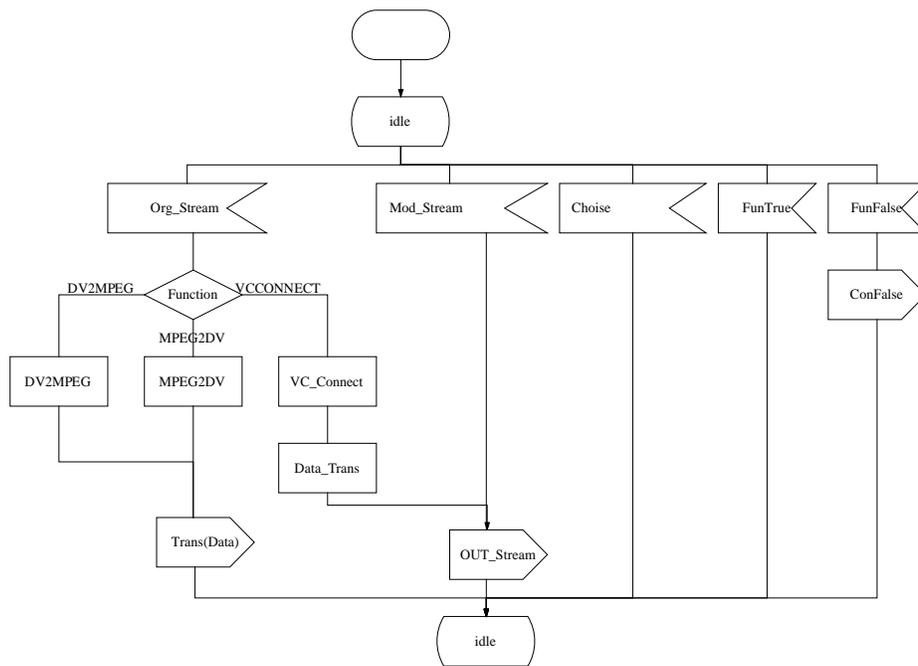


図 4.19: control プロセス

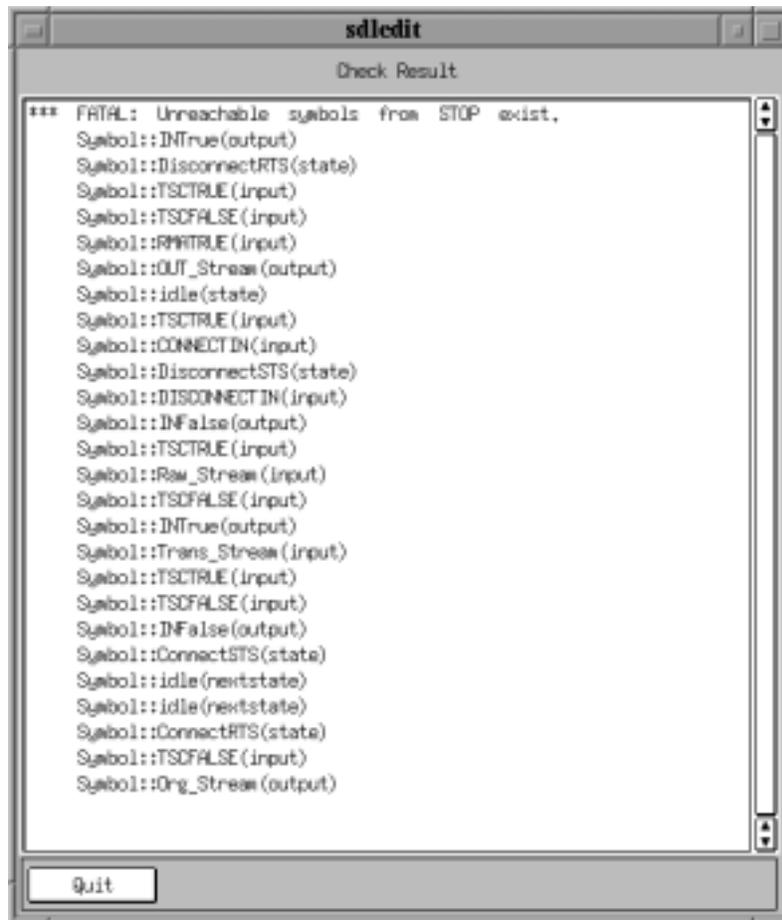


図 4.20: interface プロセス仕様に対するチェック

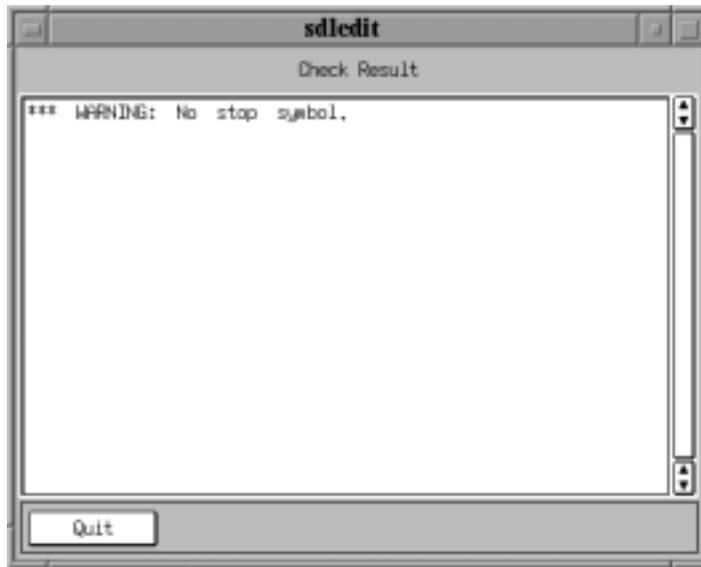


図 4.21: control プロセス仕様に対するチェック

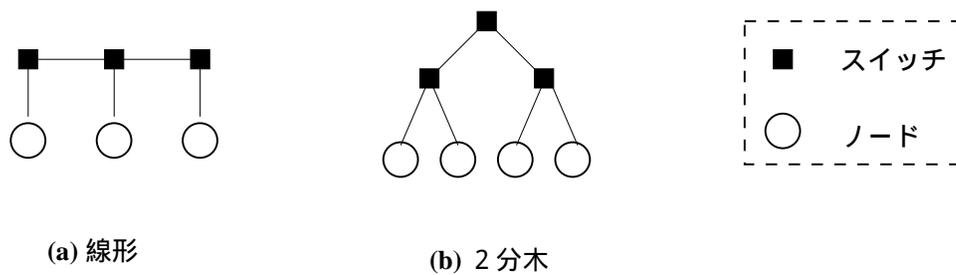


図 4.22: トポロジ

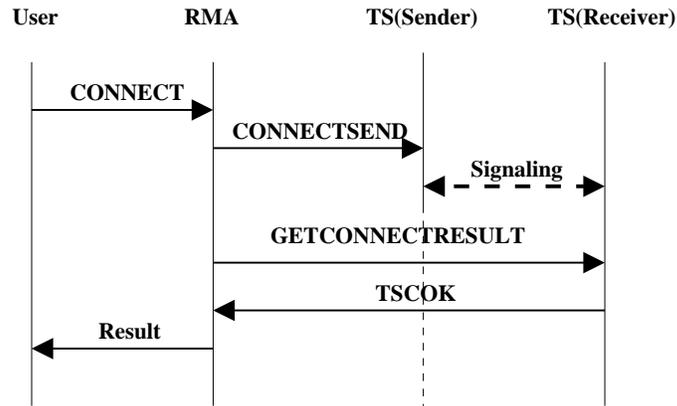


図 4.23: 中間ノードを用いない場合のコネクション確立時のメッセージハンドリング

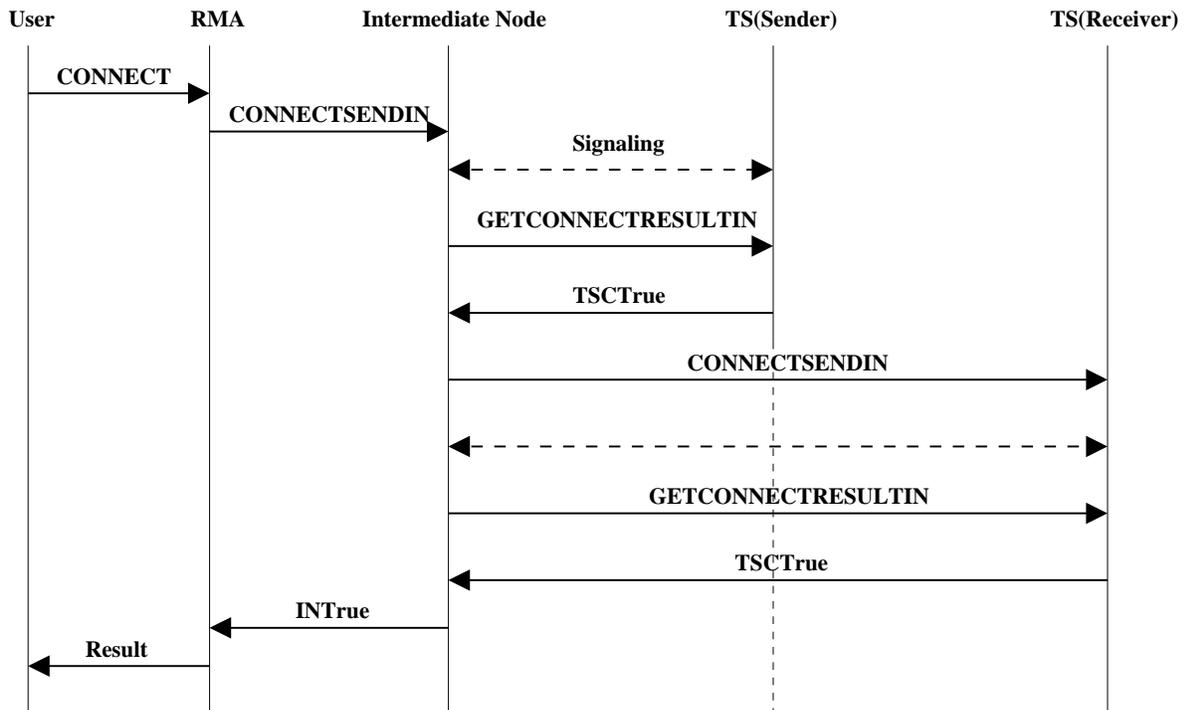


図 4.24: 中間ノードを用いる場合のコネクション確立時のメッセージハンドリング

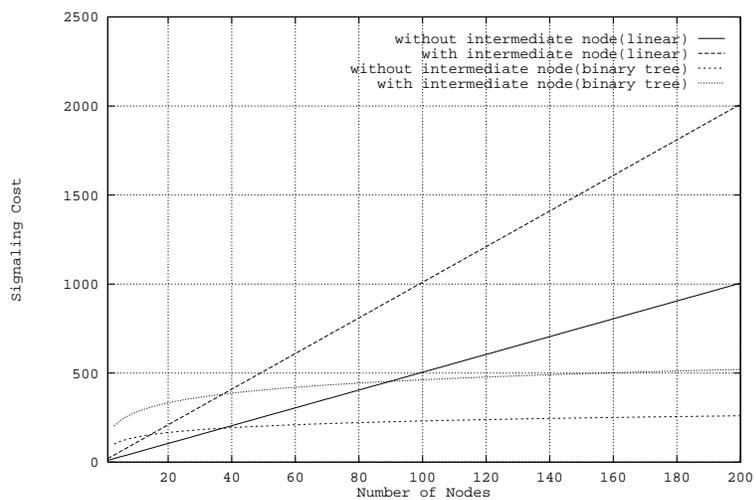


図 4.25: トポロジに対するメッセージの送信者と全体のシグナリングコストの関係

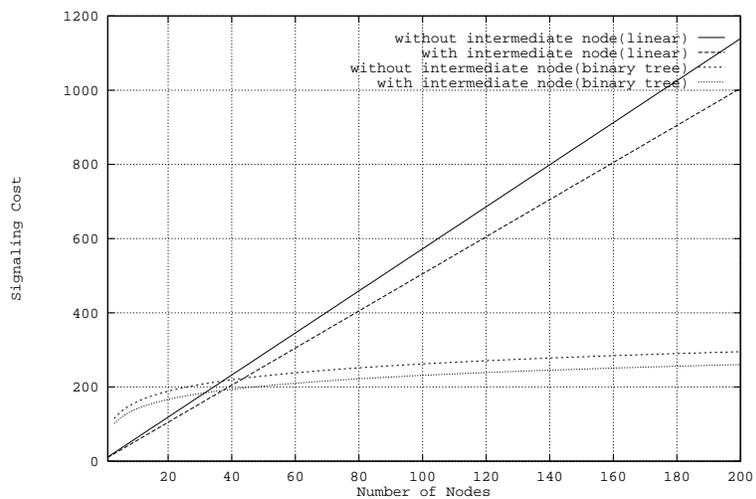


図 4.26: 各ノードから中間ノードへの距離を小さくした場合の全体のシグナリングコストの関係

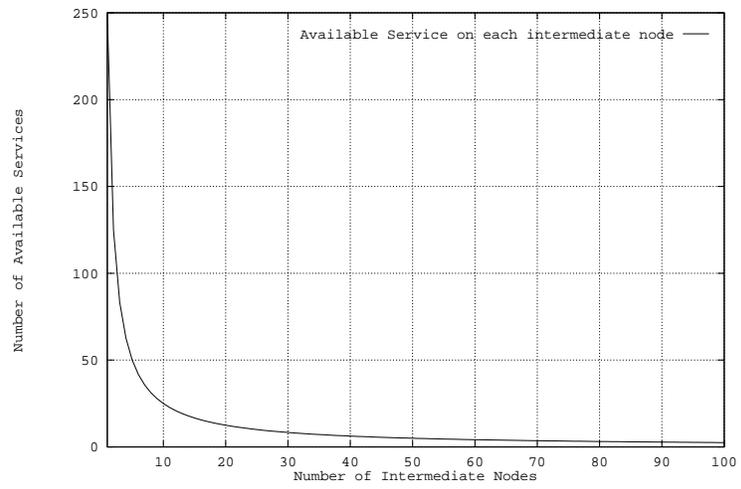


図 4.27: 中間ノードの数と接続の集中度の関係

第 5 章

検討及び考察

本章では中間ノードを利用する際の効果や評価に対し検討を加え、提案する中間ノードによる拡張機構に対する考察を行う。

5.1 中間ノードに対する検討

5.1.1 資源消費量

中間ノードを用いた場合、近隣のノード同士の通信においても直接コネクションを確立するのではなく中間ノードを介したコネクションを確立するため、その経路分多くの資源を消費してしまうのは明らかである。ただし、画面分割サービスなどはいくつかの映像を一度中間ノードでデータを集約して一つのものとして送出するため資源消費を抑えるようになるサービスもある。そのデータの圧縮の程度が大きければ大きい程、大規模なネットワークにおいて中間ノードの存在が重要となってくる。

5.1.2 通信遅延時間

中間ノードを用いることの最大の欠点は通信遅延時間を生じることである。特にマルチメディアアプリケーションのように実時間性のあるデータを取り扱うものは遅延時間や遅延時間のばらつきによるジッタはサービスの質を悪化させるものである。例えば画面分割サービスなどは一度データを復元して適切な大きさに変形させ、複数の画像を合成させたあと、再びセルレベルまで分割して送る。また画像フォーマットのエンコードあるいはデコードなどは、現在の性能の高いハードウェアで処理をしたとしても数百 ms 程度の遅延が発生してしまう。エンコードされたデータはデコードされなければならないので、実質

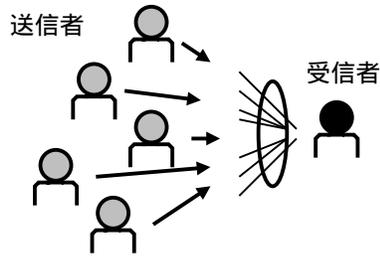


図 5.1: 受信者付近でのデータの集中

倍の遅延を考えなければならない。さらに中間ノードによる経路制御に要する時間も十分に考慮しなければならない。特に ATM ネットワークの特徴の要素であるセルヘッダによるハードウェアスイッチングを中間ノードを用いるために、それを一度 AAL 以上のレイヤまで持ち上げて処理することが要求されるため、ATM における利点を一部無駄にってしまう。

5.1.3 コネクションの集中度

中間ノードを用いる場合、中間ノード付近で送受信のコネクションが集中することも考慮することが重要となる。特に複数の画像を集約する画面分割サービスなどはその分割数分は最低でも集中する。ただし、このサービスの場合、中間ノードを用いない場合に、図 5.1 のように送信者、あるいは受信側付近で集中するため、一方的に問題があるとは言えない。ただ、いずれにしても広い帯域を占有してしまうために、JAIST VideoLAN で通常用いる DV データなど 30M なども帯域を必要とするフォーマットでは利用が難しい。これに対してはポイントツーポイントで中間ノードへ接続して MPEG フォーマットへ変換を行なった後、そのデータを再び中間ノードに戻して画面分割サービスを利用するなどの対処により回避できるのではないかと考える。また、コネクションの集中に対する問題と同様、中間ノードの能力も限界があるため、複数の中間ノードを設けることでコネクションを分散化する必要がある。これに対しては RMA により効率的に中間ノードの利用を分散させることが可能である。

しかし、複数の中間ノードを多段的に利用すると、さらに伝達遅延時間が累積する結果となる。またシグナリングコストの面に関しても増加するといったオーバーヘッドも伴うことになる。従って複数の中間ノードを利用する形態を提供する場合、そのトレードオフも十分に考慮しなければならない。

5.1.4 シグナリングコスト

中間ノードを用いる場合、中間ノードを介したコネクションが送受信側に必要となるために、単純に倍のシグナリングが必要となる。しかし、既存のグループ同士の結合といった場合、送出経路データの変更だけで済むという利点もある。ある巨大なマルチキャストグループを統合する場合、どちらかのグループが一度解散した後、再び1つのグループとして確立するコストを考えると、中間ノード内の経路選択のみで変更可能で、シグナリングは必要ないとなると中間ノードは大変有効である。特に中間ノード内の経路選択モジュールが ATM スイッチと同等の性能を持てばこのデータの受け渡しに関する遅延は、シグナリングを繰り返す場合に比べ大幅に軽減することができる。

また同様に、既存のグループに対し別の映像を送信しようとする場合、中間ノードによりグループが管理されていれば中間ノードへポイントツーポイントのコネクションを確立するだけで中間ノードがそのグループヘデータをフォワードすることでシグナリングコストを大幅に軽減することができる。

5.1.5 全体的な評価

本来、ATM は送信側から受信側までコネクションを確立して、その後は ATM スイッチのみで高速にセルの経路選択をすることにより、高速で遅延時間の少ないネットワークを目指したもので、その点から言って、中間ノードを介したコネクション確立はその長所をスポイルしていると言える。

しかし、中間ノードを用いることによって多くの付加的サービスが容易に提供できるようになり、またシグナリングコストのような動的なメンバシップの構成に関する点やコネクションの集中を分散させる機構など、中間ノードがない場合は良いとは言えない。中間ノードを用いない場合、規模が大きくなるにつれて新規参加者のコネクション確立の付加は増大する。また既存の参加者においても、メンバシップが変更する度にコネクションの追加、削除を強いられる。これら UNI3.1 では送信者からしかコネクションが確立できないことによるものである。中間ノードを用いるとメンバシップの変更を中間ノードが行うことができる場合があり、コネクションの追加や削除に関して参加ノードに負担を与えない。従って、中間ノードに十分な能力があれば個々の参加者の負担を極めて低く抑えることが可能となる。

また、今回設計した管理ノードとの連携については、複数の中間ノードが配置されるような環境において、その効果が発揮されると予想される。中間ノードを複数用いる場合は、それらが連携しあって動作する場合も考えられるが、メッセージのやりとりにかかる

コストやそのための資源消費に比べて非常に効果的であると考え。特に多人数でコネクションが集中しがちなサービス形態においては、複数の中間ノードを一元管理する本手法はメッセージのやりとりを行わず、RMA の処理だけでコネクションの重複度やメッセージの集中度が計れるため、適切なコネクションの振り分けが可能となる。

5.2 JAIST VideoLAN システムにおけるアドレス体系

JAIST VideoLAN システムでは管理や制御に関するメッセージは CLIP(Classical IP over ATM) を用いて ATM ネットワーク上で IP データグラムにより通信を行っている。ここで中間ノードが RMA との間で管理や制御に関するメッセージを交換するためには、御互いの IP アドレスを知っておく必要がある。本稿においてはこの部分は明確に規定せず、固定で既知のものとした上で実装を行っている。この部分に関しては、Well-known な IP アドレスとして設定する方法や ATM スイッチなどで提供される ILMI 機構と同様にスイッチなどの拡張機構によって IP を動的に割り当てる機構を用意することや、システム全体のメッセージなどの帯域外データを IP ではなく別の独自アドレス体系を持って利用するなどが考えられる。

5.3 実装について

今回の実装においては管理ノードである RMA と、シグナリングを行い、ATM ネットワークで DV データをやりとりする端末である TS とのメッセージのやりとり部分のみを行い、具体的な経路交換や映像フォーマットの変換などのモジュール類の実装は行わなかった。しかし、管理ノードである RMA とのプロトコルを設計し、その検証を行うことでモジュール類の多くは既存のハードウェア製品により容易に実現でき、ユーザの要求は容易に満たすことができる。

第 6 章

今後の課題

6.1 複数の中間ノードの配置における分散処理

本論文においては、中間ノードが複数存在した場合、RMA により適切に処理して負荷を分散させることを示した。また、このことにより、より ATM マルチキャストにおけるスケーラビリティが向上する。しかし、その分散させる手法は中間ノードの処理能力や中間ノードまでの経路の持つ帯域幅、ネットワーク的な距離による遅延など複雑な要素が絡みあう。そのため、中間ノードの適切な配置、及び、その選択機構を十分に考慮しなければならない。

6.2 異種ネットワーク間との結合

現在、Jini や HAVi、Universal Plug&Play などの家電製品や AV 機器のための相互接続規格が多く提案されている。また、相互接続されるネットワークシステムとしてはビデオ会議のための H.320 といった規格やインターネットなども対象とされる。

それぞれの利用形態毎には多くの利点が存在し、各メディア毎に適した利用をすることが望ましい。しかし、多くのユーザが利用するためには1つの共通のフレームワークを提供する必要がある。そこで、中間ノードを外部ネットワークとのインタフェースを用意して異種ネットワーク間を接続する。これにより、それぞれのシステムのアーキテクチャは変更せずに外部ネットワークから JAIST VideoLAN における資源を利用できるようになり、また逆も可能となる。

第 7 章

おわりに

最初にマルチキャスト通信を用いた多人数参加型システムに対して発生し得る問題点や、そのような通信形態におけるサービスについて述べ、それらに対する効果的な解決手法として中間ノードを用いる手法を検討した。また、中間ノードを利用する際の 1 手法としてネットワーク内に管理ノードを配置して、そのノードにより複数の中間ノードを用いることが可能となる構成について検討を行なった。これらの検討にあたって、多人数参加型会議中におけるグループの存在という概念を挙げ、それぞれを中間ノードにより効果的に管理することによりメンバ管理の負荷を減少させる手法について深く考慮した。また、実際に本学で稼働中の JAIST VideoLAN システムにおける管理ノードに対して存在する RMA との連携についても重点を置き、RMA と中間ノード間のプロトコルを明確に規定する点も熟慮した。さらに、JAIST VideoLAN システムに中間ノードを実装した際の構成やそこで提供されることによりユーザに提供される有効なサービスについて検討し、その仕様を作成した。

最後に、この仕様を実際のシステム上で利用する際の問題や本システムの有効性について考察を行なった。

謝辞

本研究をまとめるにあたり、終始寛容なご指導を受け賜りました、丹康雄助教授に心より感謝致します。また、適切なお助言をして頂きました篠田陽一助教授に深く感謝致します。

さらに、苦勞を共にした、倉岡貴志君、谷口雅幸君、中田潤也君に感謝致します。

その他、貴重なご意見、ご討論頂きました丹研究室の皆様をはじめ、ソニー株式会社関係者の方々に対し、厚く御礼申し上げます。

参考文献

- [1] 丹 康雄, "JAIST Video LAN - 実世界指向マルチメディアネットワーク", 人工知能学会 FAI 研究会, SIG-FAI-9802, 1999
- [2] 丹 康雄、野村 隆、田守 寛文, "家電的ユーザーインタフェースを有する大規模マルチメディア LAN システム", 情報処理学会 Interaction'99, 1999.
- [3] Yasuo Tan, "Scaling up IEEE 1394 DV Network to an Enterprise Video LAN with ATM Technology", IEEE International Conference on Consumer Electronics '98, 1998.
- [4] Takashi Nomura, Tetsuaki Kiriya, Hiroshi Yamamoto, Atsushi Maruyama, Hiroshi Takizuka, "New Protocol Architecture ASEL", IEEE International Conference on Consumer Electronics '98, 1998.
- [5] H. Tamori, "The AV Controller on PC Using AV Plug", IEEE International Conference on Consumer Electronics '98, 1998.
- [6] S.Deering, Host Extensions for IP Multicasting. Internet RFC1112, Aug. 1989
- [7] The ATM Forum, User-Network Interface(UNI) Specifications Version 3.1, Prentice Hall PTR, 1995.
- [8] Yasutaka Miwa, "A Study for Multicast Communications on ATM Network", master thesis, Faculty of Science and Technology, KEIO UNIVERSITY, 1995.
- [9] G.Armitage, Bellcore, "Support for Multicast over UNI 3.0/3.1 based ATM Networks", Internet RFC2022, Nov.1996

- [10] 三重野勤
「大規模なネットワーク上での信頼性のあるマルチキャスト」大分大学 工学部 修士
論文 1996 年 2 月
- [11] S.Deering
RFC1112,Host Extentions for IP Multicasting
Stanford University August 1989
- [12] 木村範彦, 丹 康雄
「マルチキャスト通信プロトコルにおける中間ノードによる資源の有効利用法」
情報処理学会, 第 59 回 全国大会,1999
- [13] 倉岡貴志, 丹 康雄
「AV 系ネットワークシステムにおける資源管理法に関する一手法」
情報処理学会, 第 59 回 全国大会,1999
- [14] NEC,RDSPACE 利用の手引き ,pp.151-171,1994
- [15] S.Shenker, D.J.Mitzel.” Asymptonic Resource Consumption in Multicast Reservation
Styles”
In Proceedings of SIGCOMM '94,1994.