

卒業論文 2004年度 (平成16年度)

計算機資源に応じた映像配信機構に関する研究

慶應義塾大学 環境情報学部

氏名：千代 佑

指導教員

慶應義塾大学 環境情報学部

村井 純

徳田 英幸

楠本 博之

中村 修

南 政樹

平成16年12月29日

## 計算機資源に応じた映像配信機構に関する研究

本研究では、計算機資源に基づく ALM(Application Layer Multicast) 映像配信機構の設計を行い、映像配信におけるエンドユーザの取得機会の拡大と配信の効率化を実現するシステムを構築した。

既にインターネット上において広域へのコンテンツ配信を実現するための配信の効率化手法として、IP マルチキャストを利用するモデルが存在する。しかし、IP マルチキャストを利用するためには、配信元とエンドユーザの間のネットワーク構成がマルチキャスト経路制御に対応している必要がある。IP マルチキャストは個々のユーザ環境に応じてコンテンツの品質を制御することができないため、ユーザ環境の多様化に対応できない問題がある。特にコンテンツが映像ストリームの場合、ユーザ環境による取得機会の制約は著しい。

本研究では、ネットワーク構成に依存せずに配信網を構築することが可能であるオーバーレイマルチキャスト技術と、エンドユーザが所有する計算機資源における余剰資源に注目した。計算機資源の大きさをメトリックとし、アプリケーションレイヤで配信網を構成する機構と、余剰資源を利用してユーザの要求に応じた品質の映像ストリームを複製する機構を組み合わせたシステムの設計、実装、及び性能評価を行った。

本研究により、配信元はネットワーク構成に依存する配信の効率化技術を利用せずに、広域へのコンテンツ配信を行うことが可能になった。また、配信元が要求するユーザ環境に満たないエンドユーザによるコンテンツの閲覧が可能になった。

### キーワード

1. コンテンツ配信, 2. ALM(Application Layer Multicast), 3. 計算機資源

<p>A Research of Video Distribution System Based on Computer Resources</p>
--

In this research, the ALM (Application Layer Multicast) video distribute system that is based on the computer resource is designed, and the system provides more opportunity to the end user in video distribution and improves the efficiency of the distribution is constructed. There are some models, which uses IP multicast as an efficient way to deliver contents in wide area. However, in order to utilize the IP multicast, entire network from contents distributors to the end user must be corresponded to multicast routing protocol.

IP multicast is not capable for controlling the quality of the contents according to individual user's environment. Because of this, it cannot correspond to the diversification of user environment. Especially when the contents are video streams, the restriction with user environment is considerable.

This research focuses on overlay multicasting technology, which can build a distribution network without dependent on user's network composition, and surplus computer resources which end users own.

The mechanism of constitutes the application layer multicast network which sets size of computer resources to metric, and the mechanism of coping video stream which is controlled by the user's demand which is based on their surplus resources are focused. In this research, the system, which is combined by those two mechanisms, is designed and implemented. Also, the performance evaluation was performed.

Through this research, contents distributors are able to distribute contents in wide area without using distribution technologies, which depend on network composition. In addition, this research allows end users who don't have the environment, which contents distributors require to view the contents.

Keywords :

1. contents distribution, 2. ALM(Application Layer Multicast), 3. computer resources

Keio University , Faculty of Environmental Information

Tasuku Chiyo

# 目次

第1章	序論: インターネットにおける コンテンツ配信モデル	1
1.1	背景	1
1.2	インターネットにおけるコンテンツ配信モデルの現状	2
1.3	研究の目的	2
1.4	論文の構成	3
第2章	本研究における要素技術	4
2.1	IP ネットワークにおける映像・音声配信	4
2.1.1	ストリーミング	4
2.2	配信の効率化	5
2.2.1	IP マルチキャスト	5
2.2.2	オーバーレイマルチキャスト	6
第3章	関連研究	8
3.1	XCAST : Explicit Multicast	8
3.1.1	XCAST の特徴	8
3.1.2	XCAST の問題点	9
3.2	RelayCast	10
3.2.1	RelayCast の特徴	10
3.2.2	プロキシ型ミドルウェア	10
3.2.3	RelayCast の問題点	12
3.3	PCN : PointCast Network	12
3.3.1	PointCast の特徴	12
3.3.2	プッシュ型配信技術	12
3.3.3	PointCast の問題点	13
3.4	既存の配信モデルにおける問題点	13
第4章	コンテンツ配信モデルの提案	14
4.1	本研究の要求事項	14
4.2	本研究の対象とするモデル	14
4.2.1	配信形態	15
4.2.2	ノード環境	15

4.2.3	計算機資源	15
4.3	配信の効率化手法	21
4.4	取得機会の拡大化手法	22
4.5	システムモデル	23
<b>第5章</b>	<b>計算機資源に基づく</b>	
	ALM 映像配信機構の設計	<b>25</b>
5.1	設計概要	25
5.1.1	全体構成	25
5.1.2	動作概要:Root Mode	27
5.1.3	動作概要:Edge Mode	28
5.1.4	動作概要:Leaf Mode	29
5.2	ALM 機構の機能	30
5.2.1	ノード情報取得モジュール	30
5.2.2	ノード間リンクモジュール	30
5.2.3	ノード情報管理モジュール	31
5.2.4	映像配信機構制御モジュール	32
5.3	映像配信機構の機能	32
5.3.1	配信データの複製	32
5.3.2	配信データの品質制御	33
<b>第6章</b>	<b>計算機資源に基づく</b>	
	ALM 映像配信機構の実装	<b>35</b>
6.1	実装概要	35
6.2	実装環境	35
6.3	DVTS	35
6.4	ALM 機構	36
6.4.1	ノード情報取得	36
6.4.2	ノード間リンク	37
6.4.3	ノード情報管理	40
6.4.4	DVTS 制御	40
6.5	映像配信機構	40
6.5.1	DV データの複製	40
6.5.2	DV データの品質制御	40
<b>第7章</b>	<b>評価</b>	<b>41</b>
7.1	評価概要	41
7.2	本機構の実現した機能	41
7.3	ALM 機構の性能	43
7.3.1	評価環境	44

7.3.2	ノード数によるマッチング時間 . . . . .	45
7.3.3	検索条件数によるマッチング時間 . . . . .	47
7.3.4	再構成時のデータ欠損量 . . . . .	47
<b>第8章</b>	<b>結論</b>	<b>48</b>
8.1	まとめ . . . . .	48
8.2	今後の展望 . . . . .	49

# 目次

1.1	センタライズされた配信モデル	2
2.1	ダウンロード型再生方式	4
2.2	ストリーミング型再生方式	4
2.3	IP マルチキャスト概要	6
2.4	オーバーレイマルチキャスト概要	7
3.1	RelayCast のアーキテクチャ	11
3.2	PointCast 専用クライアント	13
4.1	資源の再配分モデル	16
4.2	処理と資源の関係	17
4.3	既存の Unicast 配信システム	21
4.4	再配信による効率化	22
4.5	単一品質による配信システム	22
4.6	品質制御による取得機会の拡大	23
4.7	システムモデル	24
5.1	全体概要図	26
5.2	動作概要:Root Mode	27
5.3	動作概要:Edge Mode	28
5.4	動作概要:Leaf Mode	29
5.5	1 : 1 の複製	33
5.6	1 : n の複製	33
5.7	品質制御:無	33
5.8	品質制御:有	33
6.1	get_bogomips 関数	37
6.2	show_basic_mii 関数	38
6.3	rbm_node_t 構造体	38
6.4	ノードの生存確認	39
7.1	ノードの初期状態	42
7.2	エッジノードの参加状態	42

7.3	ノードの初期状態 . . . . .	43
7.4	リーフノードの参加状態 . . . . .	44
7.5	検索回数とマッチング時間の変化 . . . . .	45
7.6	ノード数 (~100) とマッチング時間の変化 . . . . .	46
7.7	ノード数 (~1000) とマッチング時間の変化 . . . . .	46
7.8	条件数とマッチング時間の変化 . . . . .	47



# 表 目 次

1.1	コンテンツの変化 . . . . .	1
4.1	資源量と閲覧可否の関連性 . . . . .	15
4.2	BogoMips 値 . . . . .	19
4.3	CPU 種別と CPU 係数 . . . . .	19
4.4	演算能力と推定消費資源 . . . . .	20
5.1	構成モジュール . . . . .	26
5.2	品質制御の方法 . . . . .	34
6.1	実装ソフトウェア環境 . . . . .	35
6.2	実装ハードウェア環境 . . . . .	36
7.1	評価環境 . . . . .	44

# 第1章 序論: インターネットにおけるコンテンツ配信モデル

## 1.1 背景

インターネットの利用目的は、利用者の増加と共に多様化してきた。ネットワーク・エンドノード環境の性能向上によって、ユーザは様々なコンテンツの制作・公開・配布が可能になった。また、それらのコンテンツは定常的にインターネット上に流通している。

インターネットに接続可能なエンドノードは、デスクトップPC、ノートPC、PDA、携帯電話等、種類が増加している。これらの機器はそれぞれ計算機としての処理能力や処理可能なデータ形式が異なる。また、エンドノード上で動作するOS（オペレーティングシステム）やアプリケーションの種類も増加している。

インターネット上で扱われるコンテンツの絶対量、種類、コンテンツの利用形態、インターネット上で扱われるコンテンツの種類と情報のライフサイクルの変化を表 1.1 に示す。ただし、情報のライフサイクルの長さはコンテンツに対して一定ではなく、ユーザのコンテンツに対する要求に基づいて流動する。

表 1.1: コンテンツの変化

過去		現在
少ない ex. HTTP, FTP, RealMedia[1]	絶対量	多い ex. HTTP, FTP, WMV, VoIP DVTS, MPEG2-TS
少ない	種類	多い
ダウンロード型	利用形態	ダウンロード型 プログレッシブダウンロード型 ストリーミング型
長期 ex. 蓄積型配信, 統計データ	ライフサイクル	短期 ex. ライブ配信, ニュース速報

## 1.2 インターネットにおけるコンテンツ配信モデルの現状

現状のコンテンツ配信モデルは、WWW(World Wide Web) を介したサーバ・クライアント型を用いるものが多い。このようにセンタライズされた配信モデルを図 1.1 に示す。ユーザは、ネットワーク・エンドノード環境、コンテンツの性質に関わらず、スター型のセンタライズされたサーバを介して、受動的な情報取得を行っている。コンテンツ配信において、任意の送信元からエンドノードまでの環境には様々な構成が考えられる。

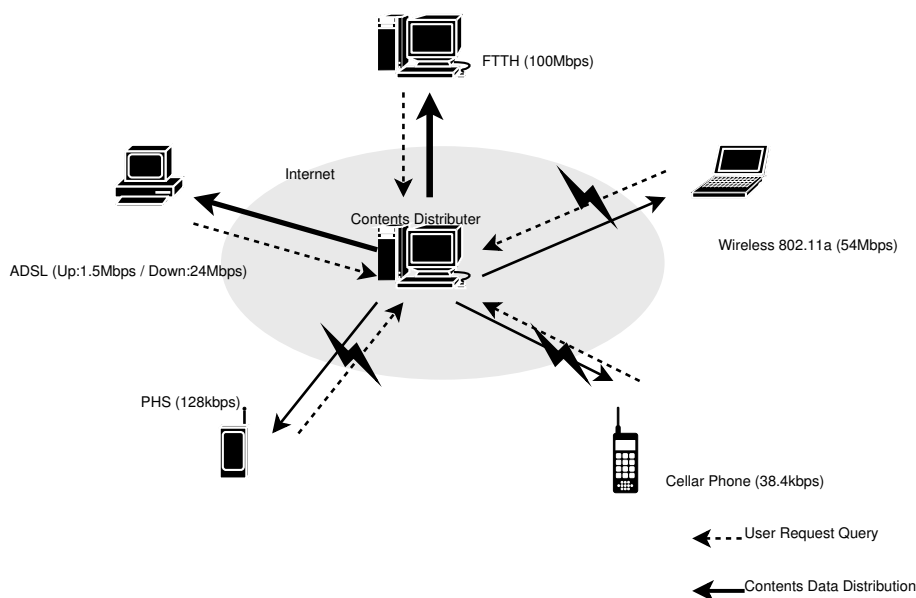


図 1.1: センタライズされた配信モデル

配信元がコンテンツのデータ形式や品質といった個々のユーザ要求に基づいて配信を行うことは技術的に可能である。しかし個々のユーザ要求に対応するために要求別のサーバを用意し、配信網を構築するにはコストがかかる。単一の送信元が全ての受信者に対し、一括して配信を行う現在の方法はサーバへの負担が大きい。また、一括配信を行うモデルでは受信対象環境がサーバに依存するため、ユーザの情報取得機会の制約となる可能性がある。

以上のことより、広域に分散したユーザに対しインターネットを介してコンテンツ配信を行う場合、1) ユーザ要求に基づいた配信コンテンツタイプとユーザ環境のマッチング、2) 配信元を誘導するコンテンツ配信の 2 つが必要となる。

## 1.3 研究の目的

本研究の目的は、ユーザの要求に基づいて配信するコンテンツとユーザ環境とのマッチングを行い、配信元を誘導するようなコンテンツ配信モデルを構築することによって、

ユーザの情報取得機会の拡大を実現することである。本研究では、エンドノードの計算機資源に応じてネットワークや計算機資源の負荷分配を行う。これにより、送信者は配信のためのネットワークや計算機資源の消費量を抑えることができる。受信者は送信者が提供するサーバの能力に制約を受けずにコンテンツの閲覧を行うことができる。本研究ではこれらを実現するマルチキャスト配信システムを提案する。

## 1.4 論文の構成

本論文は 8 章で構成される。第 2 章ではインターネットにおける映像・音声配信の要素技術を述べた後コンテンツ配信の効率化技術について検討する。第 3 章では、本研究の関連研究として既存の配信モデルについて言及し、これらの問題点を明らかにする。第 4 章において本研究の提案する映像・音声配信モデルについて述べ、配信の効率化とコンテンツの取得機会の拡大について検討する。第 5 章において本研究の設計について述べる。第 6 章では設計に基づいた実装について述べ、第 7 章で本システムの実装に対する評価を行う。最後に本システムのまとめと応用の可能性及び今後の課題について述べる。

## 第2章 本研究における要素技術

本章では, 本機構を構築するために必要である IP ネットワークを用いた映像・音声転送技術, 及びそれらの効率化に用いられる技術に関して述べる.

### 2.1 IP ネットワークにおける映像・音声配信

本節では IP ネットワーク上における映像・音声転送に用いられる, ストリーミング技術に関して述べる.

#### 2.1.1 ストリーミング

映像・音声データ全体を受信してからローカルで再生する方式が図 2.1 のダウンロード型再生方式である. これに対し, データの受信と再生を平行して行う方法が図 2.2 のストリーミング型再生方式である. リアルタイム性の高い映像・音声転送だけでなく, Real Systems[2] に代表される蓄積型映像・音声配信にも用いられる.

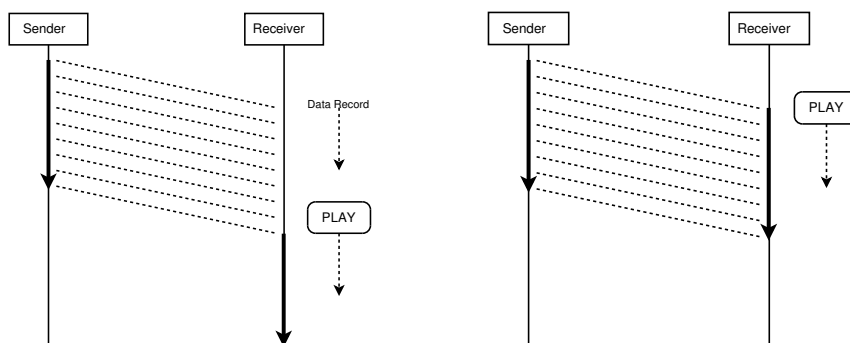


図 2.1: ダウンロード型再生方式 図 2.2: ストリーミング型再生方式

## 2.2 配信の効率化

コンテンツ配信を効率化する試みとして、IP マルチキャスト技術 [3] を応用した CDN (Contents Delivery Network) が挙げられる。CDN は、ネットワーク上の様々な場所に配布ポイントを用意し、ユーザのネットワーク位置に応じた最適な配布ポイントを指定することで、配信の効率化を図るものである。

本節では、配信の効率化の手法である IP マルチキャスト技術とマルチキャストを上位レイヤで実現するオーバーレイマルチキャスト [4] について技術的要素と問題点を述べる。

### 2.2.1 IP マルチキャスト

単一の送信ノードから複数の受信ノードに同一内容のパケットを送信する場合に利用するのがマルチキャスト技術である。送信ノードから同一内容のパケットを複数の受信ノードに送信する方法には、ユニキャストを用いる方法とマルチキャストを用いる方法がある。

複数のユニキャスト (1 対 1) 通信を用いた場合、送信元のノードが送信すべきデータサイズが受信ノードの数に比例して増加する。このためデータの送信処理を行うための計算機資源の消費量が増加する。また、送信ノードが必要とするネットワーク帯域も同様に増加する。

IP マルチキャストの概要を図 2.3 に示す。マルチキャスト通信を用いた場合、送信ノードは単一のパケットのみを送信する。パケットは、途中の適切なルータによって複製され、受信ノードに対して行われる。経路上のルータは、マルチキャスト受信ノードが異なるインタフェースに存在している場合のみ、パケットの複製を行う。

IP ネットワークにおいてマルチキャスト通信を行う IP マルチキャストは、送信ノードで受信ノードの数のパケットを複製せずに、分散しているルータを用いてパケットの複製を行うので、負荷が分散される。しかし、IP マルチキャストを行うにはマルチキャスト経路制御 (Multicast Routing) [5] をサポートするノードとスイッチ・ルータなどのネットワーク環境が必要である。

マルチキャストパケットの配送は、マルチキャストアドレスに基づいたグループに参加している全てのノードに配送される。このため、IP マルチキャストの経路制御は複数ホストを対象として行わなくてはならないため、非常に複雑になる。グループに参加しているノードが拡散している場合、経路上の全てのルータが保持する必要がある経路情報が膨大になる。また、現状ではコンテンツやネットワークの管理を考慮した効率的なアプリケーション体系が無いため、同一 ISP 内のような一部の限定されたネットワークを除いて IP マルチキャストの普及は難しい。

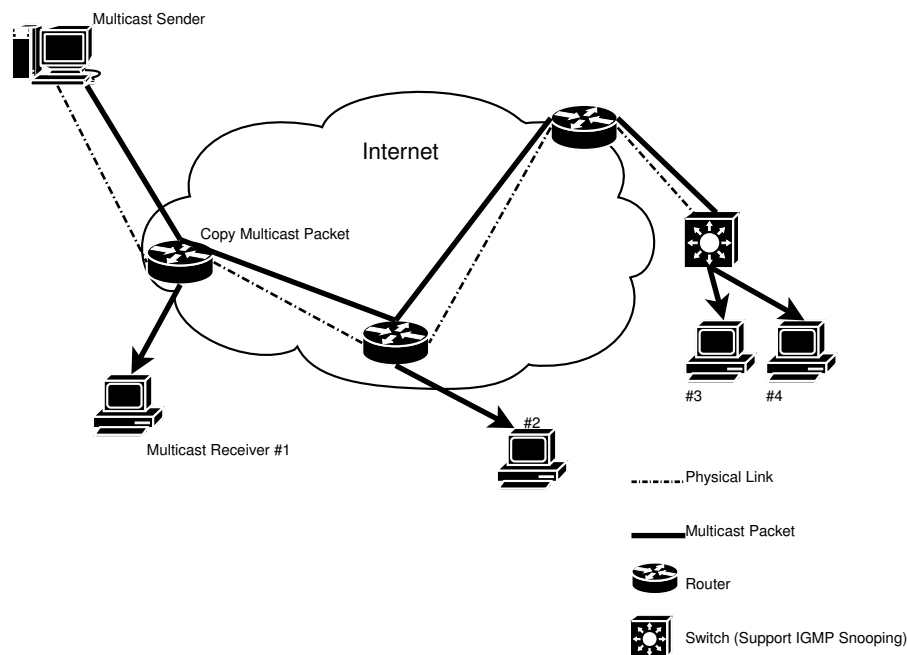


図 2.3: IP マルチキャスト概要

### 2.2.2 オーバーレイマルチキャスト

オーバーレイマルチキャストとは IP マルチキャストの代替手法である。IP マルチキャストではマルチキャスト通信をネットワークレイヤで行うが、オーバーレイマルチキャストはこの機能をアプリケーションレイヤなどの上位レイヤで実現する。

オーバーレイマルチキャストの概要を図 2.4 に示す。具体的には、ルータで行っていたパケットの複製処理をエンドノードで行う。オーバーレイマルチキャストでは、マルチキャストグループに属するエンドノード同士で論理的なリンクを構成し、それぞれのノードがユニキャスト通信を行う。

またオーバーレイマルチキャストでは、マルチキャストグループに参加しているノードをコントロール・トポロジとデータ・トポロジに二分し体系化している。コントロール・トポロジは各ノードが相互に生存確認を行い、トポロジの構成を維持する。データ・トポロジは実データの流れを規定する。

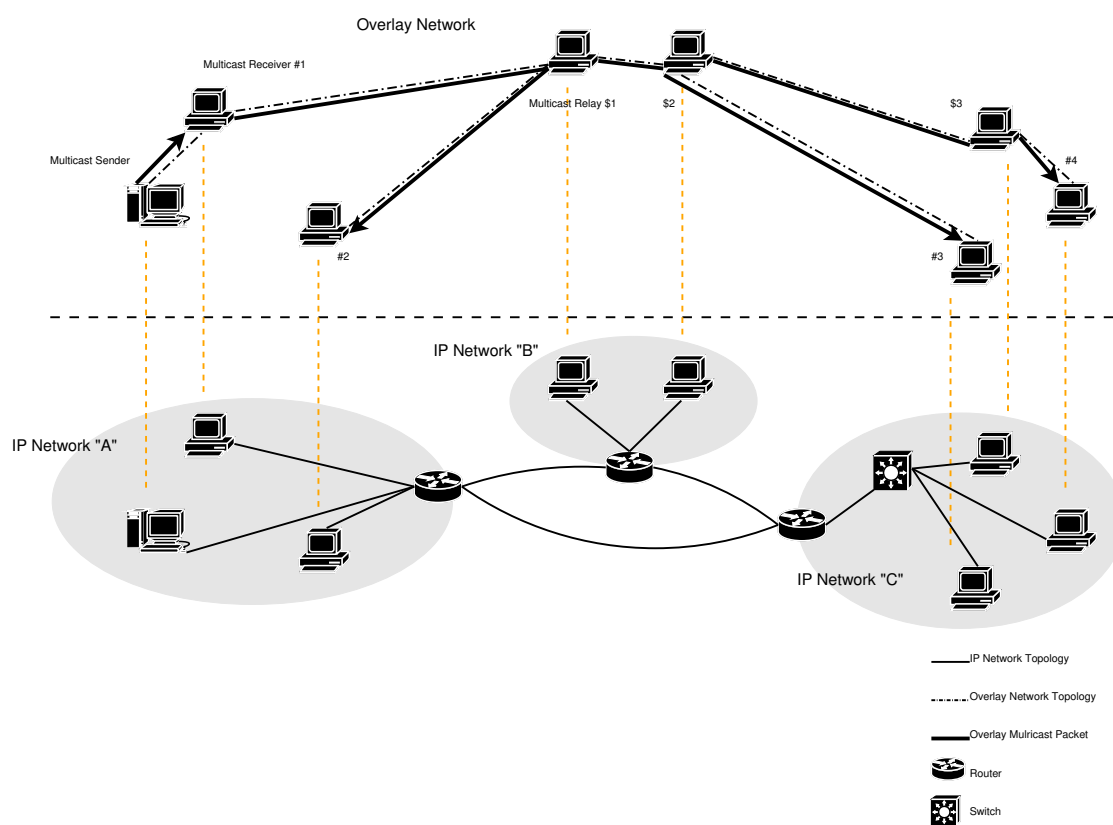


図 2.4: オーバーレイマルチキャスト概要



## 第3章 関連研究

本章では従来のユニキャストやIPマルチキャストに対してコンテンツ配信の効率化を考慮した研究について述べる。その上で本研究の目的との関連性と技術的問題点を述べる。またそれらと本研究における関連性と、技術的問題点の検証を行う。

### 3.1 XCAST : Explicit Multicast

XCAST(Explicit Multicast)[6]とはIPヘッダの拡張ヘッダ内に宛先を全て羅列することにより、IPマルチキャスト技術を用いずに、1対nの転送を実現する。XCASTはIPマルチキャストと異なり、新たな経路制御プロトコルやアドレスの管理、割り当てを必要としない。

XCASTではIPマルチキャストにおけるマルチキャストグループに対して、マルチキャストセッションという用語を用いる。また、ビデオ会議のように複数の送信元があるセッションにおいて、1つの送信元によって送信されるフローはチャンネルと定義される。

マルチキャストのホストグループモデルにおけるパケットは、全てのグループメンバーの論理的な識別子として、マルチキャストアドレスを送受信する。XCASTにおいて送信元ノードは、宛先へパケットを送信しようとするマルチキャストチャンネル内の宛先を追跡する、これは Keeps Track と呼ばれる。

送信元はXCASTヘッダで宛先のリストを符号化し、ルータにそのパケットを送信する。経路上のルータはXCASTヘッダを解析し、それぞれの宛先の次ホップに基づいた宛先毎に分割する。そして、それぞれの次ホップに適切なXCASTヘッダをつけてパケットを転送する。

XCASTヘッダに1つの宛先だけが残ったとき、XCASTパケットは通常のユニキャストパケットに変更される。そして、それは残った経路上をユニキャストによって転送される。(X2U:XCAST to Unicast)

#### 3.1.1 XCASTの特徴

XCASTは従来のIPマルチキャストと比べて、以下のような特徴がある。

- ルータはセッションまたはチャンネル毎の状態を維持する必要がない。これにより、ネットワークのノードがセッションのマルチキャストルーティング情報を伝播し

たり記憶したりする必要が全く無くなる。このためサポート可能なセッションの数に関して、XCAST は容易に大規模化できる。

- マルチキャストアドレス割当が不要である。
- マルチキャストルーティングプロトコルが不要である。XCAST パケットは、通常のユニキャストルーティングプロトコルによって決められた正しい経路へ転送される。
- コアノードが不要なので、その故障によるシステムダウンはない。配信ツリー構成と違って、XCAST はネットワーク遅延を最小にし、ネットワーク効率を最大にする。
- 対称のパスは必要ない。従来のマルチキャストルーティングプロトコルは、パスが対称 (対称=A から B までの最短パスが B から A までの最短パスと同じ) でないなら、非最短パスツリーを作成する。
- ユニキャストへの新ルートに自動的に反応する。XCAST は、ユニキャストのルートの変化に、即座に反応できる。従来のマルチキャストルーティングプロトコルにおいて、ユニキャストおよびマルチキャストルーティングプロトコル間の通信は接続確立が必要である。多くの実装ではポーリングを基本としており、例えばリンクの故障に対してより遅い反応となる。多数のグループがリンクの故障に反映する必要があるれば、従来方式のマルチキャストルーティングプロトコルは、これを処理するのに時間がかかる。

### 3.1.2 XCAST の問題点

XCAST にはコンテンツ配信の効率化を考慮した上で以下の問題点がある。

- IPv4 環境において経路上のルータが、XCAST に対応していない場合経路が冗長になり、ルータに負荷がかかる。
- それぞれのパケットは、全ての未到達宛先の情報を含むため、オーバーヘッドが生じる。解決には宛先アドレスのリストを圧縮する方法などが必要である。
- パケットにあるそれぞれの宛先に対して、ルーティングテーブルを検索する必要がある。それで、 $n$  個の宛先を持つ XCAST パケットは、 $n$  個のユニキャストヘッダと同じだけルーティングテーブルの検索を必要とする。そのうえ、異なるヘッダをホップ毎に再構成する必要がある。このため経路上のルータのルーティングテーブルの参照回数が増加する。

- 拡張ヘッダ内に宛先を含めるという設計上, マルチキャストセッションあたりの受信できるノード数の上限が低い. このため, 広域に同じ内容を一括配信するといった目的には向いていない.

## 3.2 RelayCast

RelayCast[7] はノードのみを構成要素とするオーバーレイネットワーク上にマルチキャストツリーを形成する ALM(Application Layer Multicast) である. RelayCast はこの機能をミドルウェアで提供する.

RelayCast はミドルウェアによって ALM の抽象化を行い, 既存のユニキャスト配信アプリケーションを ALM 対応アプリケーションへ移行させることが考えられている.

### 3.2.1 RelayCast の特徴

RelayCast の基本機能は, オーバーレイネットワーク構築とマルチキャストツリー構築の 2 つに大きく分けられる. これらのミドルウェアとしての基本機能を実現するために 2 つの機能は, それぞれインタフェースを定めた 2 層に分けられている. この他に, 対向ホストと通信を行う部分, メトリックを測定する部分, 対向ホストの情報やリストを保持する部分が RelayCast には必要となる.

### 3.2.2 プロキシ型ミドルウェア

RelayCast のアーキテクチャを図 3.1 に示す. 点線で囲まれた部分がプロキシ型ミドルウェアとなる. アプリケーションとミドルウェアの間, およびミドルウェアと他ホストのミドルウェアの間は, Socket API により接続・通信される.

各ブロックのそれぞれの役割は以下の通りである.

- P2PCom  
指定されたホストに対して, 制御メッセージ及びデータの送受信を行う. 他ホストに対して向けられたソケットを持つ.
- MetricEstimator  
オーバーレイネットワーク最適化のためのメトリックを測定, 評価する. 基本的には, バックグラウンドで実行される. 測定には, ping や pchar などの外部プログラムを利用することも可能であるが, OS によっては引数や出力表示が異なる場合があるので, 機種依存性が生じる. このような場合は, 同等なメカニズムを用意する必要がある. 測定結果は LogicNet に渡される.
- HostList

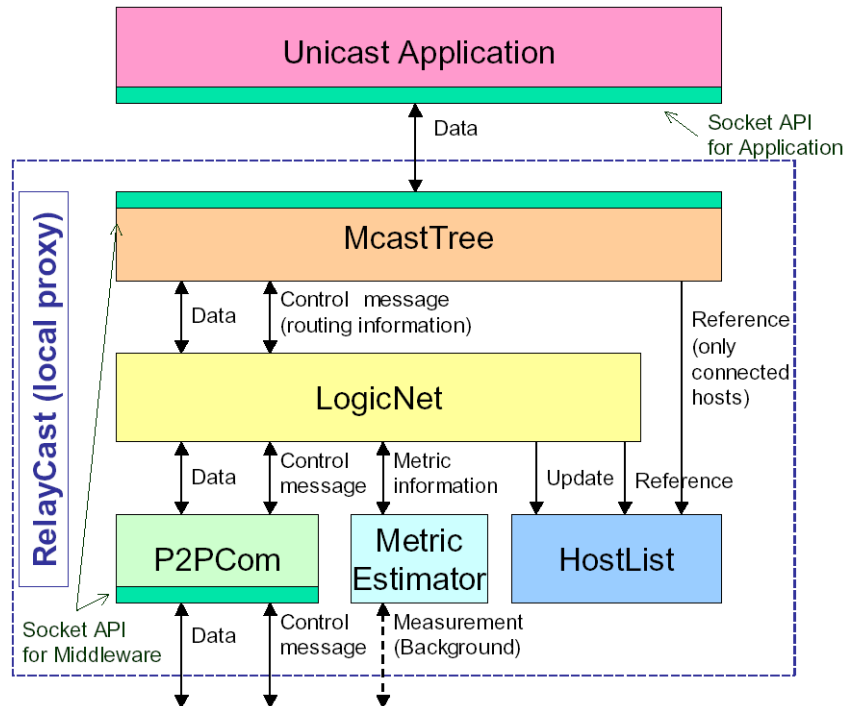


図 3.1: RelayCast のアーキテクチャ

リスト交換により得られたホストリストとホストの情報を格納する。ホストリストには、論理リンクの対向ホストやマルチキャストツリー上の下流ホスト（対向ホスト）が含まれる。ここに保存された情報は、LogicNet や McastTree から参照されることになる。

- LogicNet

オーバーレイネットワーク構築に関する機能をもつ。オーバーレイネットワーク構築の制御メッセージ（ホストリスト、セッション参加・脱退、論理リンクの接続・切断）を、HostList から選択した対向のホストを指定して P2PCom に渡す。

また、MetricEstimator から得られる測定結果に基づいて、オーバーレイネットワークの最適化を行う。なお、データや制御メッセージの受信時のフローは上記とは逆になる。

- McastTree

マルチキャストツリー構築に関する機能をもつ、アプリケーションから受信したデータ及びマルチキャストルーティングの制御メッセージ（ルーティング情報）を、HostList から選択した下流のホストを指定して P2PCom に渡す (LogicNet 経由)。このとき、パケットのループを防ぐためにシーケンス番号を書いたヘッダを付加

する。データや制御メッセージの受信時のフローは上記とは逆になる。受信時には、パケットに付加されているシーケンス番号をチェックすることにより、ループを防止する。

### 3.2.3 RelayCast の問題点

RelayCast はノード間でネゴシエーションを行い、帯域と遅延 (RTT:Round Trip Time) というメトリックを用いてマルチキャストツリーを構成し、接続環境のより良いノードへの接続を行う。しかし、RelayCast ではマルチキャストツリーを構成するためのメトリックが帯域と遅延 (RTT) のみである。また、計算資源の再配分による効率化は考慮していないため、ノードの処理能力がボトルネックとなる可能性がある。

## 3.3 PCN : PointCast Network

PCN(PointCast Network)[8] とは、EntryPoint 社 (旧 PointCast 社)[9] が 1996 年から 2000 年にかけて、インターネット上で提供していた情報提供サービスである。ニュースや天気予報などのコンテンツがあり、専用ビューワで閲覧する。ユーザがいくつかあるチャンネルの中から好きなチャンネルを選んで登録すると、サーバから自動的に最新の情報が送られる。PCN におけるコンテンツ配信の中心となる技術が PointCast である。

### 3.3.1 PointCast の特徴

PointCast ではコンテンツが”チャンネル”単位で提供される。ユーザはその中から必要なチャンネルだけ登録することができる。さらに、各チャンネルには複数のコンテンツ項目が用意されており、ユーザは必要なものを選択する。これらは図 3.2 のような専用のクライアントアプリケーションで閲覧することができる。また、情報を提供する配信元もユーザからのリクエストを容易に把握することができる。

### 3.3.2 プッシュ型配信技術

PointCast のクライアントアプリケーションは、バックグラウンドで登録した必要なコンテンツを配信元のサーバから自動的に任意の間隔で取得する。ユーザの操作に因ることなくコンテンツを取得することが可能なためこれをプッシュ型配信と呼ぶ。プッシュ型配信はリアルタイム性を要求するコンテンツの配信に適している。また、ユーザは必要なコンテンツのみを登録し、サーバは登録されたコンテンツのみを配信するため、ユーザの要求とコンテンツのマッチングを実現していると言える。

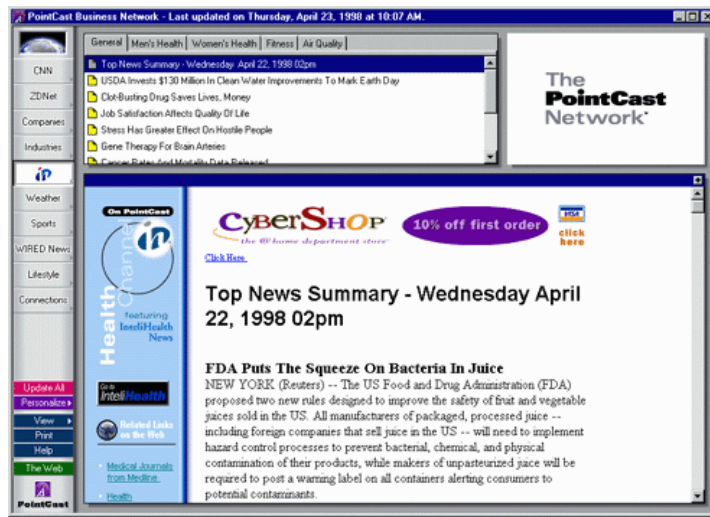


図 3.2: PointCast 専用クライアント

### 3.3.3 PointCast の問題点

この配信モデルではクライアントアプリケーションは情報更新の有無に関わらず、定期的にサーバに対してアップデートチェックを行う。このため、ユーザが広域に分布した際、IP ネットワークにおけるトラフィックが増加する。トラフィックの増加は、配信元のサーバとユーザが閲覧を行うエンドノード間の経路となる ISP の有するネットワークや経路上の機器への負荷となる。

PointCast はこの問題に対し、1998 年に IP マルチキャストに対応することによる解決を図った [10] が、IP マルチキャストは前述の通りネットワーク機器の対応を必要とする。また IP マルチキャストを通過させないポリシーの ISP も存在する。このため、IP マルチキャストを用いた手法は負荷に対する根本的な解決とはならなかった。

## 3.4 既存の配信モデルにおける問題点

前述した既存の技術はコンテンツ配信を行う際、ネットワーク環境への依存やコンテンツ発信元となるサーバへの負荷の増大、エンドノード環境の格差による配信モデルのボトルネック発生が起こる可能性がある。本研究ではこれらの問題点を解決する機構を組み込み、コンテンツ配信モデルの最適化を実現する。

## 第4章 コンテンツ配信モデルの提案

本章ではまず目的とするコンテンツ配信システムの要求事項についてまとめる。次に要求事項を基にコンテンツ配信モデルの提案を行う。

### 4.1 本研究の要求事項

本研究は、コンテンツ配信におけるユーザのコンテンツ取得機会の拡大と、コンテンツの品質とユーザ要求のマッチングの実現を目的とする。

本研究の目的を達成するためには以下の項目が必要となる。

- オーバーレイネットワーク上での配信網の構成の実現

IP マルチキャストのようなネットワーク層における配信の効率化技術は、現状のIP ネットワークではインターネットサービスプロバイダ (ISP) などのサービスや運用ポリシーに依存する。このため、広域への配信を行う際にはユーザの利用している接続サービスを考慮しなくてはならない。この依存問題を解決するためには、接続サービスや運用ポリシーといった実ネットワーク構成に依存しない上位レイヤにおける配信手法が必要である。

- ユーザ要求に基づく配信元への誘導の実現

本研究で提案するコンテンツ配信システムはユーザが要求し、閲覧できる最大限の品質のコンテンツを各ユーザに提供しなければならない。このためには各ユーザの所有する計算機資源量を適切に把握する必要がある。本システムではコンテンツの品質とユーザ要求のマッチングを実現するために、エンドノードの計算機資源量の情報を統一的に管理する機能が必要である。

### 4.2 本研究の対象とするモデル

本研究が対象とする配信形態、ノード環境、計算機資源について定義する。

### 4.2.1 配信形態

インターネットにおいて広域へのコンテンツのリアルタイム配信に即時性が要求される場合、既存の配信技術では広域へ分散した多数のユーザに対して到達させることは困難である。

広域へ分散した多数のユーザに対してコンテンツ配信を行う方法としては、配信の拠点や経路上の要所となる場所に中継用のサーバを設置する方法などがある。しかし、これは常設でない場合、突発的なコンテンツ配信の要求に対応できない。

そこで本研究では定常的に広域への配信に対応する機構が存在しない環境において、即時性を要求される単一のコンテンツデータを広域へ到達させるという配信形態を対象とする。

### 4.2.2 ノード環境

本研究においてコンテンツ配信網を構成する配信元のサーバ及び配信網に参加するエンドユーザのコンピュータをノードと定義する。本研究において提案する計算機資源の再配分モデルを図 4.1 に示す。ユーザの要求ベースにコンテンツを様々な方法で配信することは技術的に可能であるが、配信元がそれらを全て用意することは現実的ではない。

しかし、ユーザ環境によってはデータ処理に必要なネットワーク資源や計算機資源よりもエンドノード性能が勝っており余剰資源が存在する場合がある。インターネット上で扱われるコンテンツの場合、配信を受けたノードで余剰しているネットワーク資源と計算資源を用いて再配信を行うことが可能である。

エンドノードの資源量とコンテンツの閲覧可否には表 4.1 のような関連性がある。本研究はこれらの関連性をふまえて、コンテンツの取得機会の拡大の手法を決定する。

表 4.1: 資源量と閲覧可否の関連性

エンドノード資源量	閲覧可否	余剰資源
消費資源 < 利用可能資源		
消費資源 = 利用可能資源		×
消費資源 > 利用可能資源	×	×

### 4.2.3 計算機資源

コンテンツ配信網を構成するには、コンテンツデータの送信、受信、参加ノードの管理の 3 つを行う必要がある。各処理に伴い必要となる計算機資源を図 4.2 に示す。



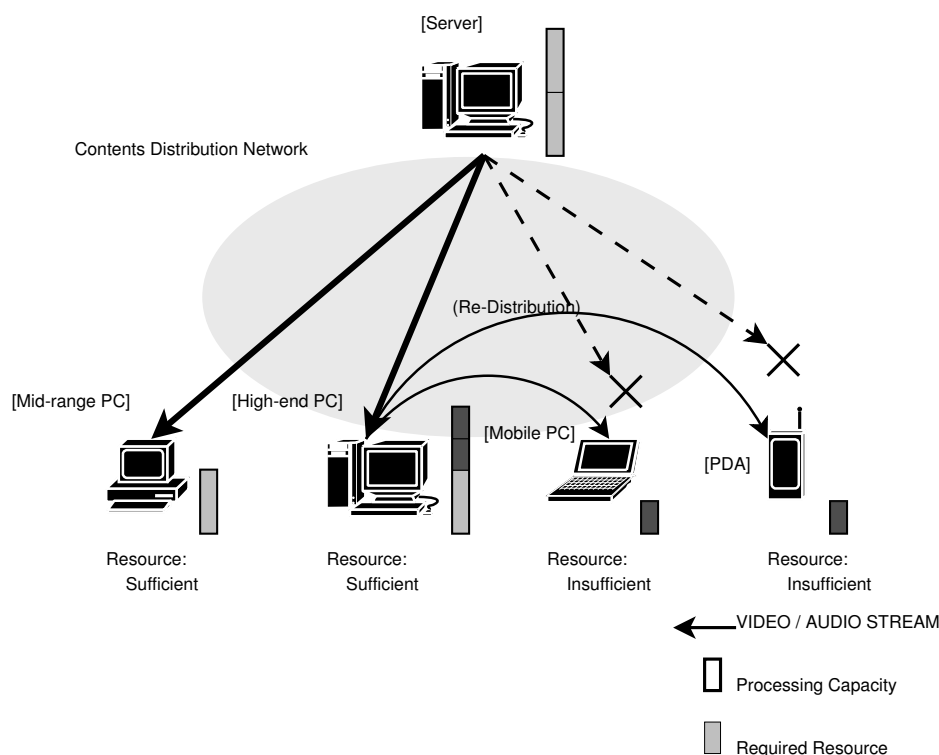


図 4.1: 資源の再配分モデル

### コンテンツデータの送信処理

- データの取り込み・生成

コンテンツ配信網において配信元となるノードは、コンテンツデータの取り込みまたは生成処理を行う。この際、データはCPUによって一時的にメモリに格納される。データの取り込み・生成処理のパフォーマンスはノードのCPU性能とメモリに依存する。本処理で考慮すべき計算機資源は、演算能力とCPU・メモリ使用率である。

- データの送信

配信網において配信元ノードや二次配信を行うノードはコンテンツデータ送信処理を行う。この際、データはCPUによってメモリからの読み出された後、ノードのNIC(Network Interface Card)を通してIPネットワークに送信される。データの送信処理のパフォーマンスはCPU性能とノードの実効帯域に依存する。本処理で考慮すべき計算機資源は演算能力と実効帯域である。

### コンテンツデータの受信処理

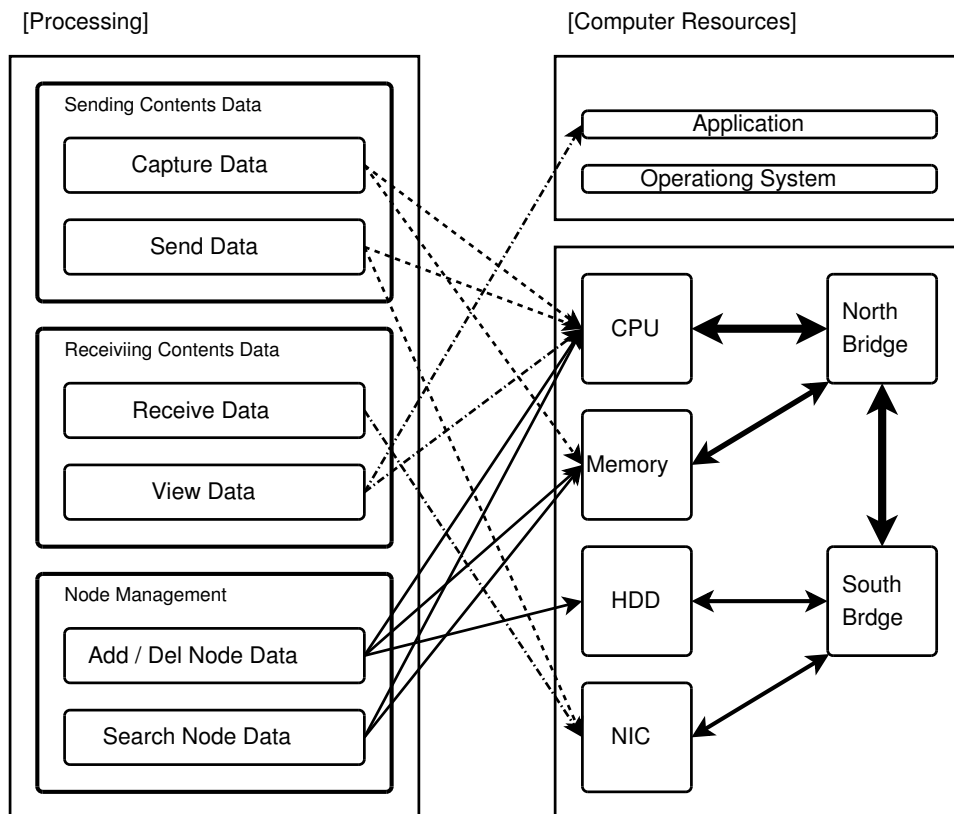


図 4.2: 処理と資源の関係

- データの受信

配信網においてコンテンツの取得を行うノードは、コンテンツデータの受信処理を行う。この際、データはIPネットワークからノードのNICを通して一時的にメモリに格納される。データの受信処理のパフォーマンスはノードの実効帯域とIPネットワークの伝送特性に依存する。本処理で考慮すべき計算機資源は、実効帯域と伝送特性である。

- データの再生・表示

コンテンツ配信網において閲覧を行うノードは、受信したコンテンツデータの再生・表示処理を行う。この際、データはCPUによってメモリから読み出された後、データ形式に対応したアプリケーションに伝達される。データの再生・表示処理のパフォーマンスはCPU性能と再生アプリケーションに依存する。本処理で考慮すべき計算機資源は演算能力とデータ形式である。

参加ノードの管理処理

- ノードの追加・削除

コンテンツ配信システムでは参加ノードの管理を行う必要がある。参加ノードの管理を行うノードは、ノード情報の追加・削除処理を行う。この際、ノードが IP ネットワークから NIC を通して受信したデータは、ノードがメモリもしくは HDD に作成するデータベースに格納される。ノードの追加・削除処理のパフォーマンスはノードの CPU 性能とメモリもしくは HDD に依存する。本処理で考慮すべき計算機資源は演算能力と CPU・メモリ使用率である。

- ノードの検索

参加ノードの管理を行うノードは、ノード情報を格納したデータベースを参照し配信網を保持するのに必要なデータを検索する必要がある。この際、データベースに格納したデータは CPU によってメモリから読み出され演算される。ノードの検索処理のパフォーマンスは CPU 性能とメモリに依存する。本処理で考慮すべき計算機資源は演算能力と CPU・メモリ使用率である。

コンテンツ配信網を構成するために、ノードに必要とされる計算機資源の詳細を以下に示す。

- 演算能力 (CPU 性能)

IP マルチキャストや XCAST といった配信の効率化技術を用いずにコンテンツ配信を行う場合、配信数の増加に伴い配信元のサーバに要求される演算能力が高くなる。また、エンドノードでも閲覧に必要な演算能力を考慮しなくてはならない。絶対的な演算能力は、サーバの CPU アーキテクチャや周辺ハードウェア構成に依存する。本研究では計算機資源の 1 つとして演算能力を考慮するため、CPU アーキテクチャや周辺ハードウェア構成にできるだけ依存しないおおよその指標となる値を定める必要がある。

本研究では、指標の算出法として Linux OS における古典的手法である *BogoMips*[11] を検討する。*BogoMips* とはプロセッサが一秒間に何百万回無益な処理をできるか計測した数値である。

*MIPS*(Millions of Instructions Per Second) とは、プログラムの計算速度を測る基準値である。しかし異なるアーキテクチャのコンピュータ間でこの値を公正に比較することは非常に困難である。

Linux OS ではカーネルまたはデバイスドライバ用にマシンのプロセッサ速度を計測するためのタイミングループが必要である。しかし *idle* 状態での空ループは精度が悪いのでカーネルは起動時にコンピュータ上で一種のビジーループを実行することによってその速度を測り、一種の指標としている。

指標としての *BogoMips* 値を確認するため、実際に数台の PC にて計測を行った。結果を以下の表 4.2 に示す。

表 4.2: BogoMips 値

アーキテクチャ	CPU 種別	クロック周波数	BogoMips 値
Intel x86	Pentium III	1.00GHz	1985.74
Intel x86	Pentium 4	2.40GHz	4797.27
Intel x86	Pentium M	1.60GHz	3191.60

BogoMips の計測ループはアセンブラ言語で記述されているので、Intel CPU 以外の場合、計測ループのコードは類似しているが同一ではない。また、CPU のパイプライン数やキャッシュといった CPU コアアーキテクチャによって 1 クロック当たりの処理能力は異なる。このため BogoMips 値を指標として使用するためには基準値と CPU アーキテクチャごとの係数を算出する必要がある。

BogoMips mini-Howto[12] で公開されている CPU 種別に応じた BogoMips 値などの資料と IPC(Instructions Per Clock cycle) の仕様を用いて P6 アーキテクチャ[13] の CPU である Pentium III を基準値として算出した CPU 係数の例を以下の表 4.4 に示す。

表 4.3: CPU 種別と CPU 係数

CPU 種別	コアアーキテクチャ	CPU 係数
Pentium II/III	P6	1.00
Pentium 4	NetBurst[14]	0.50
Pentium M	Pentium M[15]	1.25

- CPU・メモリ使用率

ノードにおける CPU やメモリの使用率の負荷状況は随時変化する。ノードが連続した高負荷状態になり、アプリケーションのプロセスに対して OS による資源配分が適切に行われなかった場合、処理落ちが発生しコンテンツの閲覧が不可能になる場合がある。

本研究では、処理落ちを防ぐためにアプリケーションが要求すると推定される演算能力の数値化が必要である。このためそれぞれ構成の異なる 3 台の PC と、ストリーミングアプリケーションの代表例として DVTS[16] を用い、DVTS における DV 送信コマンドである `dvsend` コマンドを用いて推定消費資源の数値化を行った。結果を以下の表 4.4 に示す。

上記の例では、`dvsend` コマンドの推定消費資源は、約 300 ~ 350 の間の値であると考えられる。このように消費資源を数値化し、管理することで配信の効率化を実

表 4.4: 演算能力と推定消費資源

CPU 種別	BogoMips 値	CPU 係数	CPU 使用率	推定消費資源
Pentium III	1985.74	1.00	16.39%	325.46
Pentium 4	4797.27	0.50	14.0%	335.81
Pentium M	3191.60	1.25	8.48%	324.78

現する。

- 実効帯域 (実効スループット)

配信元のサーバがパケットを送出する際、またエンドノードがパケットを受信する際、ノードが利用可能な実効帯域によってコンテンツの閲覧可否に影響を及ぼす。十分にバッファリングを行うアプリケーションの場合、バッファリングされたデータがアプリケーションに供給されている間は映像や音声の途切れは生じない。バッファ内のデータが枯渇するまでに必要な帯域が確保できなかった場合、コンテンツの途切れが生じる。またバッファリング量が少ない (リアルタイム性を重視した) アプリケーションの場合、実効帯域が必要帯域を下回るとはデータの不足となるため、コンテンツに途切れが生じる。

インターネットではノードが利用可能な実効帯域は随時変化する。NetPerf などのツールを用いて実効帯域を計測することはできるが、計測にかかる時間や途中経路に負荷をかけることになるため実用的ではない。ネットワークの実効速度はネットワークの伝送特性に影響を受けるため途中経路の伝送方式を考慮する必要がある。

ネットワークの実効速度はネットワークの伝送特性に影響を受けるため途中経路の伝送方式を考慮する必要がある。しかし、エンドユーザがネットワークの伝送特性を把握でき、途中経路の状態に依存する可能性が低い環境ではノードの NIC のリンク状態が一定の指標になる。

- 遅延 (RTT) ・ 揺らぎ (jitter) ・ パケット喪失

IP ネットワークでは、物理層での伝送メディアでの物理的な信号伝搬やその他のレイヤでの処理により、配信元からエンドノードへのパケット到達時間が変化する。特にインターネットでは、これらの時間の保証がなされないため遅延が生じる。

このため、エンドノードでのパケットの到着時間には揺らぎが生ずる可能性がある。配信側で一定間隔でパケットを送出した場合、理想的にはエンドノードでも同間隔でパケットを受信する事である。しかし、実際には IP ネットワークの特性上、様々なレイヤで生じる遅延が複合的に重なりパケットの到着時間には揺らぎが発生する。

また、途中経路上に存在するルータやスイッチといったネットワーク機器においてキューイング処理を行っている際、ネットワーク機器の性能不足やネットワークの輻輳によるパケット数の増加がパケット喪失を発生させる場合がある。

これらのインターネットの特性に起因する要因はエンドノードにおけるコンテンツの受信可否に影響を与える可能性がある。

- 配信データ形式

コンテンツの閲覧可否に影響を及ぼすものとして、エンドユーザが利用可能なコンテンツ配信アプリケーションの対応するデータ形式がある。配信元のサーバが提供するデータ形式とエンドユーザが利用するアプリケーションの対応するデータ形式が一致しない場合、エンドユーザはコンテンツを閲覧することができない。

また、ノードによっては CPU 性能がボトルネックとなり、閲覧の可否がデータ形式だけでなく、データ量によって決定する場合もある。このため、演算能力や実行帯域といった資源に応じて、CPU 使用率や必要帯域の増減といった品質制御が可能なコンテンツ配信アプリケーションが必要である。

### 4.3 配信の効率化手法

本研究では、既設のネットワークに一切変更を加えることなくコンテンツ配信網をオーバーレイネットワーク上に構築する。配信網に参加するノードが有する計算機資源をメトリックとするマルチキャスト配信機構を構築し、計算機資源の再配分を行うことによってコンテンツ配信の効率化を目指す。

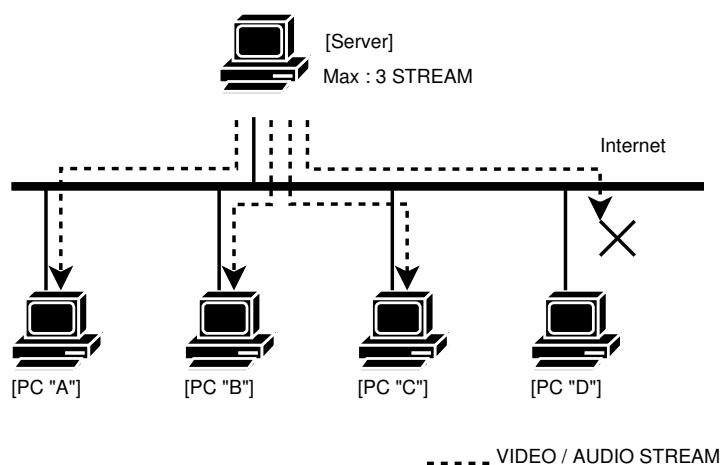


図 4.3: 既存の Unicast 配信システム

配信網に参加しているノードの内、余剰資源のあるノードは受信ノードとしての動作以外に自分自身が受信しているコンテンツを再配信することによって二次配布元とな

る事が可能になる.

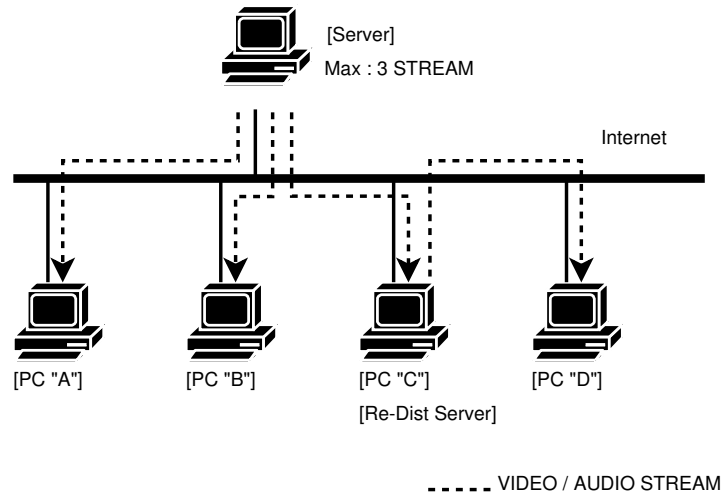


図 4.4: 再配信による効率化

#### 4.4 取得機会の拡大化手法

既存の単一品質によるコンテンツ配信システムを図 4.5 に示す. 単一品質による配信はコンテンツの取得機会を制限する.

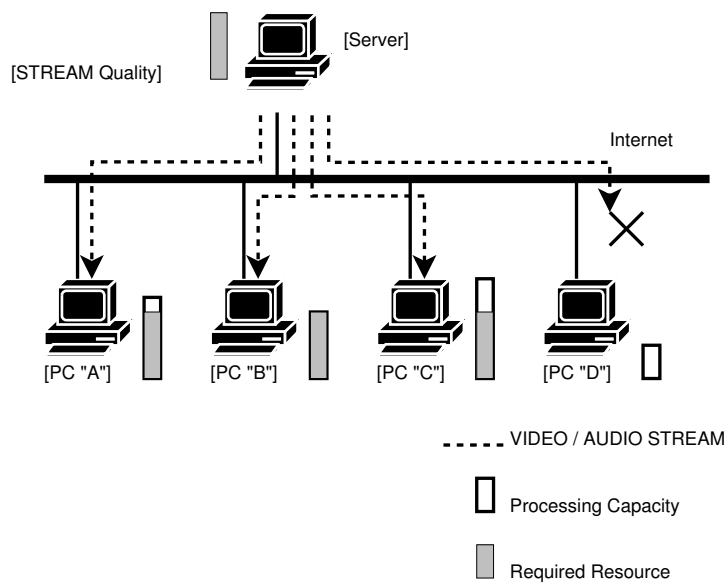


図 4.5: 単一品質による配信システム

また、配信元のサーバが IP マルチキャストなどの広域への配信技術を用いてコンテンツ配信を行うことが不可能な場合、エンドユーザがコンテンツを取得可能な機会はサーバが配信可能なノード数が上限となる。しかし、本来受信しか行わないエンドノードがサーバから受信したコンテンツの再配信を行うことによって、サーバが配信可能なノード数の上限を超えてユーザのコンテンツ取得機会を拡大することが可能になる。本研究におけるコンテンツの取得機会の拡大に対するアプローチを図 4.6 に示す。

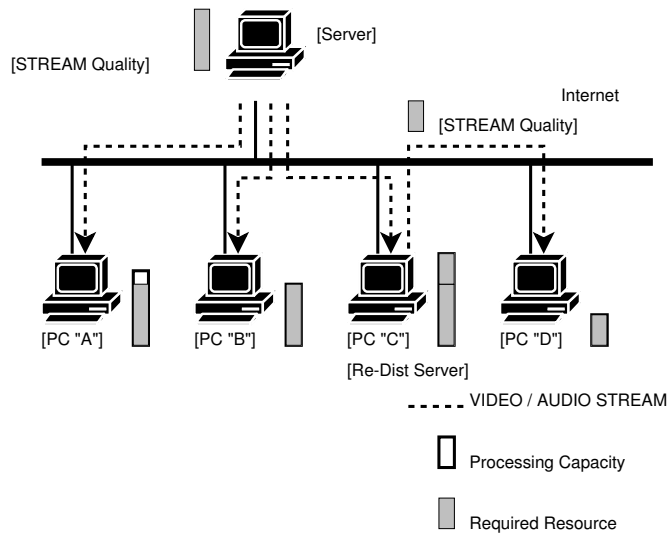


図 4.6: 品質制御による取得機会の拡大

エンドノードに要求されるネットワーク環境や計算機資源、配信データ形式、データ量によってコンテンツの取得機会は制限される。本研究では、余剰資源を有するノードが配信データの品質制御を行い再配信を行う。これにより利用可能資源量が要求に満たないために配信元が直接提供する配信データでのコンテンツ閲覧が不可能なノードでも閲覧が可能になる。

## 4.5 システムモデル

既存の技術はコンテンツ配信を行う際、ネットワーク環境への依存やコンテンツ発信元となるサーバへの負荷の増大、エンドノード環境の格差による配信モデルのボトルネック発生が起こる可能性がある。本研究ではこれらの問題点を解決する機構を組み込み、コンテンツ配信モデルの最適化を実現する。前述したように、コンテンツ配信の効率化を実現するには、計算機資源を過不足なく適切に再配分するための指標となるメトリックを設定する。

本研究の提案するコンテンツ配信システムのモデルは以下の図のようになる。

コンテンツ配信を行うノードは、ツリー構造の頂点であるルートノードとなる。コンテンツ閲覧者のノードで構成されるルート以外のノード内で、計算機資源が豊富なノード



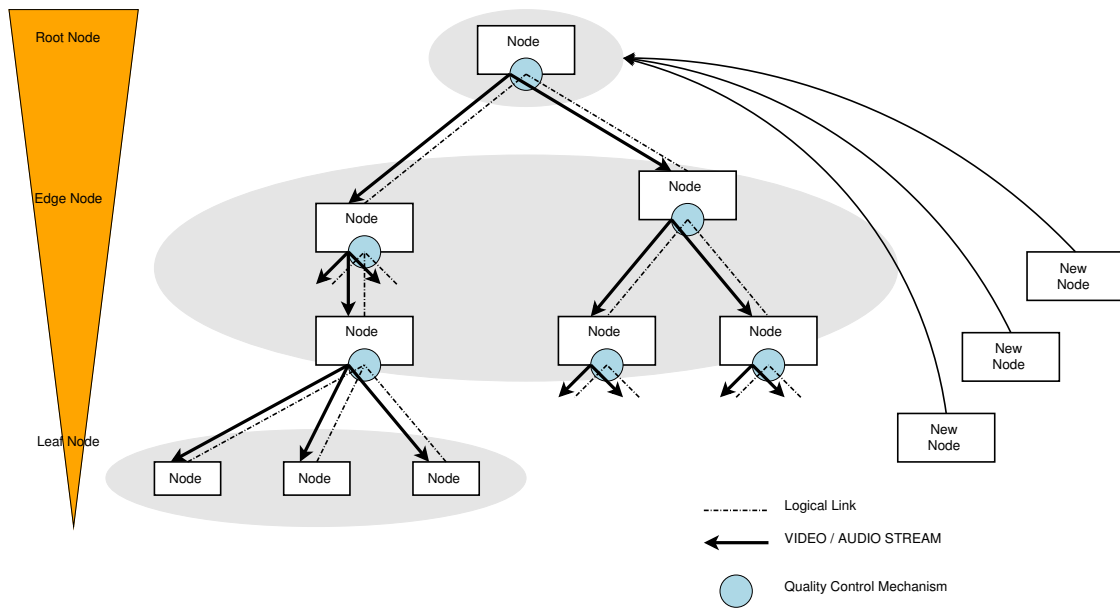


図 4.7: システムモデル

ドは論理リンクの上位に配置する。下位には上位ほど資源を持たないノードを配置し更に下位ノードへのコンテンツ配信を行うエッジノードとする。資源が少なく他のノードにコンテンツを提供することができない受信専用ノードは最下位のリーフノードとする。

ツリー構造の論理リンクの上位に配置されたノードはルートノードから品質の高いコンテンツを閲覧することが可能とする。その対価として下位ノードの要求に応じたコンテンツの再配信を行うことが要求される。一方、下位ノードは自身に適したコンテンツ配信を上位ノードに要求する、ただし上位ノードよりも品質の高いコンテンツを閲覧することはできない。

# 第5章 計算機資源に基づく ALM映像配信機構の設計

本章では、計算機資源に応じたコンテンツ配信モデルの実証例として既存の映像配信機構を対象とした際の設計について述べる。本機構の設計概要と全体構成について述べる。また、アプリケーションレイヤマルチキャスト機構と映像配信機構の各機能における詳細な設計について述べる。

## 5.1 設計概要

3章において従来のユニキャストやIPマルチキャストに対してコンテンツ配信の効率化を考慮した既存の研究について取り上げ、これらの配信モデルの技術的問題点を挙げた。また4章においてリアルタイム性の高いコンテンツ配信におけるユーザのコンテンツ取得機会の拡大と、コンテンツの品質とユーザ要求のマッチングを実現する計算機資源に応じたコンテンツ配信モデルを検討した。これらの要件を考慮した結果、本機構の必要機能は以下の通りとする。

- 計算機資源の数値化及び対応するメトリックの設定
- メトリックに基づく ALM 配信網の構成
- ノード情報の管理及び配信網の再構成
- リアルタイム配信に対応する品質制御可能な映像配信機構
- 映像配信機構の外部からの制御

### 5.1.1 全体構成

本機構の全体概要を図 5.1 に示す。本機構において ALM 機構を構成するモジュールは、以下の通りである。本機構ではノードの動作モードに応じて構成モジュールが表 5.1 のように変化する。また、本機構は ALM 機構と映像配信機構によって構成される。

- ノード情報取得モジュール  
ノードの計算機資源の取得及び数値化、メトリックの設定

5.1. 設計概要

- ノード間リンクモジュール  
ノード情報の送受信, ノード間リンクの保持 (生存確認)
- ノード情報管理モジュール  
ノード情報の追加・削除・参照・更新
- 映像配信機構制御モジュール  
映像配信の開始・停止, 宛先の指定, 品質の指定

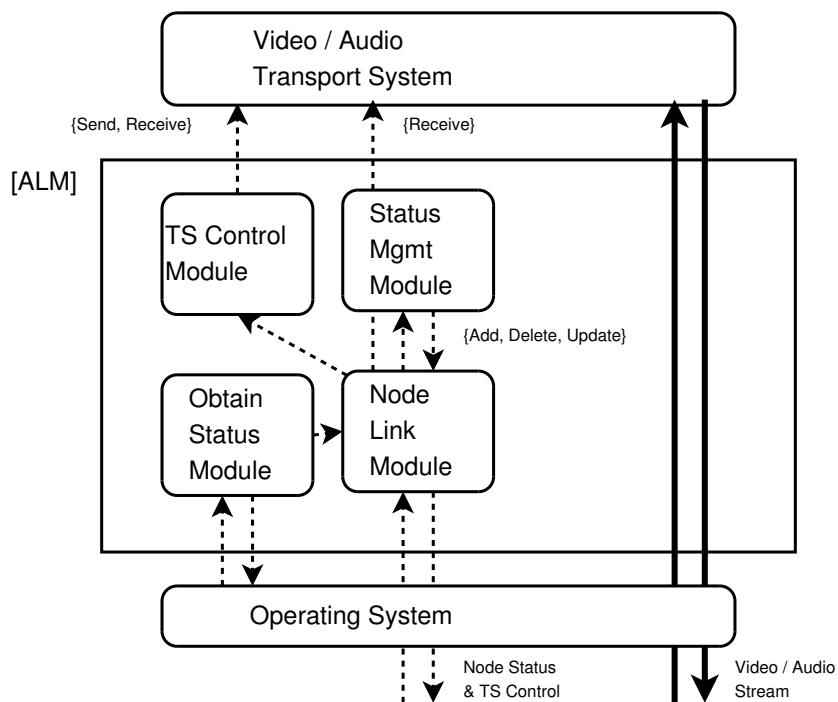


図 5.1: 全体概要図

表 5.1: 構成モジュール

	Root Mode	Edge Mode	Leaf Mode
ノード情報取得モジュール			
ノード間リンクモジュール			
ノード情報管理モジュール		-	-
映像配信機構制御モジュール			
映像配信機構 (外部)			

5.1. 設計概要

5.1.2 動作概要:Root Mode

本機構を Root Mode として動作させた場合の各モジュールの関係及び動作概要を図 5.2 に示す。

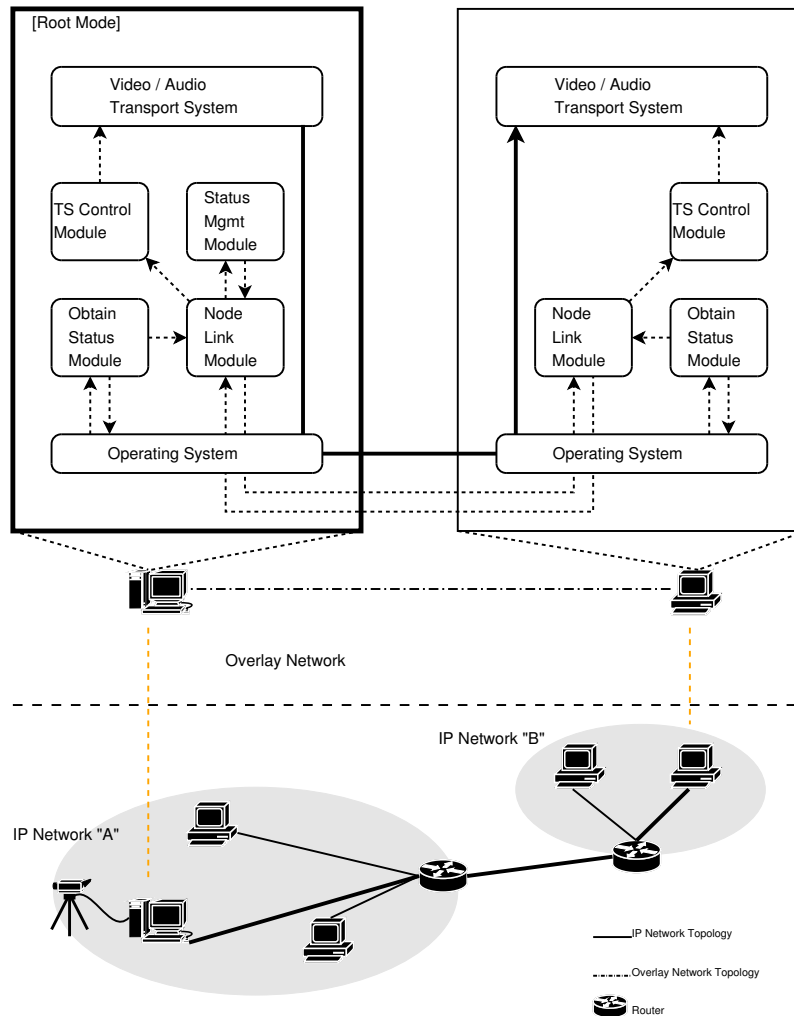


図 5.2: 動作概要:Root Mode

1. ノード情報取得モジュールによる OS からのノード情報の取得
2. ノード情報管理モジュールへのノード情報の追加・他ノードからの接続要求
3. ノード情報管理モジュールによる接続要求ノードと送信ノードのマッチング
4. 映像配信機構制御モジュールによる送信先と送信品質の設定
5. 接続要求ノードへの映像配信機構による映像データの送信

5.1. 設計概要

5.1.3 動作概要:Edge Mode

本機構を Edge Mode として動作させた場合の各モジュールの関係及び動作概要を図 5.3に示す。

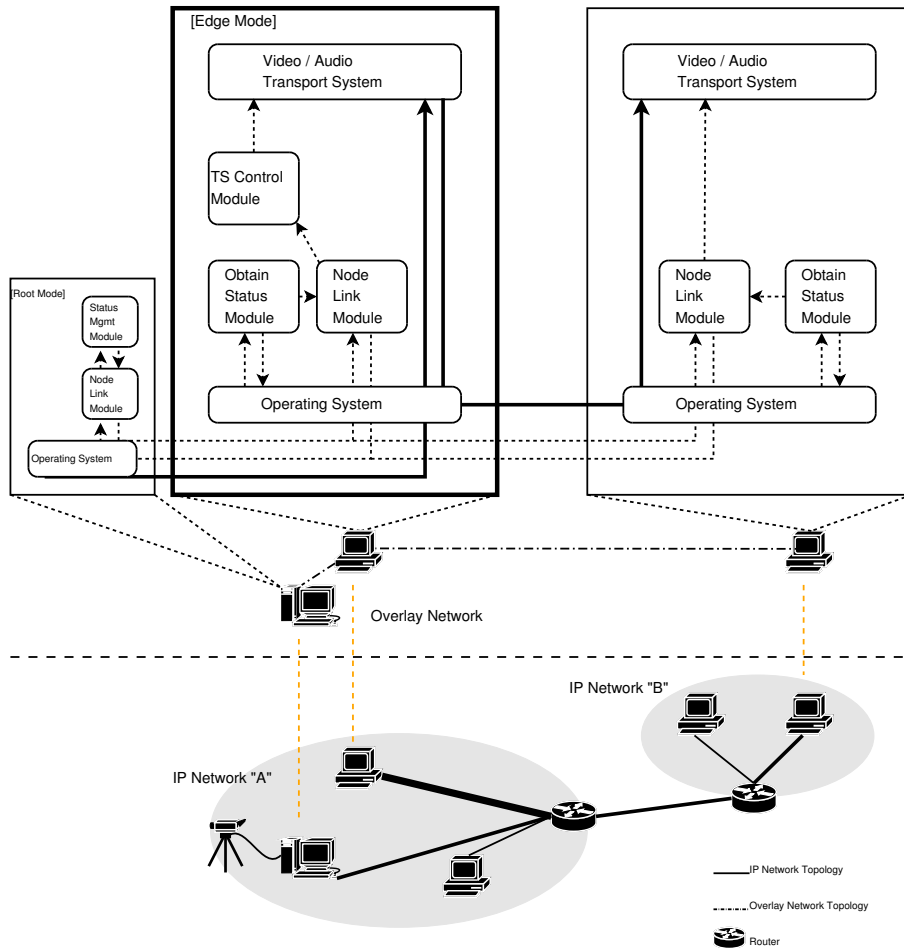


図 5.3: 動作概要:Edge Mode

1. ノード情報取得モジュールによる OS からのノード情報の取得
2. ルートノードのノード情報管理モジュールへのノード情報の追加
3. ルートノードによる上流ノードの決定・映像データの受信
4. ルートノードによる再配信命令の受信・再送信先と送信品質の設定
5. 指定ノードへの映像配信機構による映像データの再送信

5.1. 設計概要

5.1.4 動作概要:Leaf Mode

本機構を Leaf Mode として動作させた場合の各モジュールの関係及び動作概要を図 5.4 に示す。

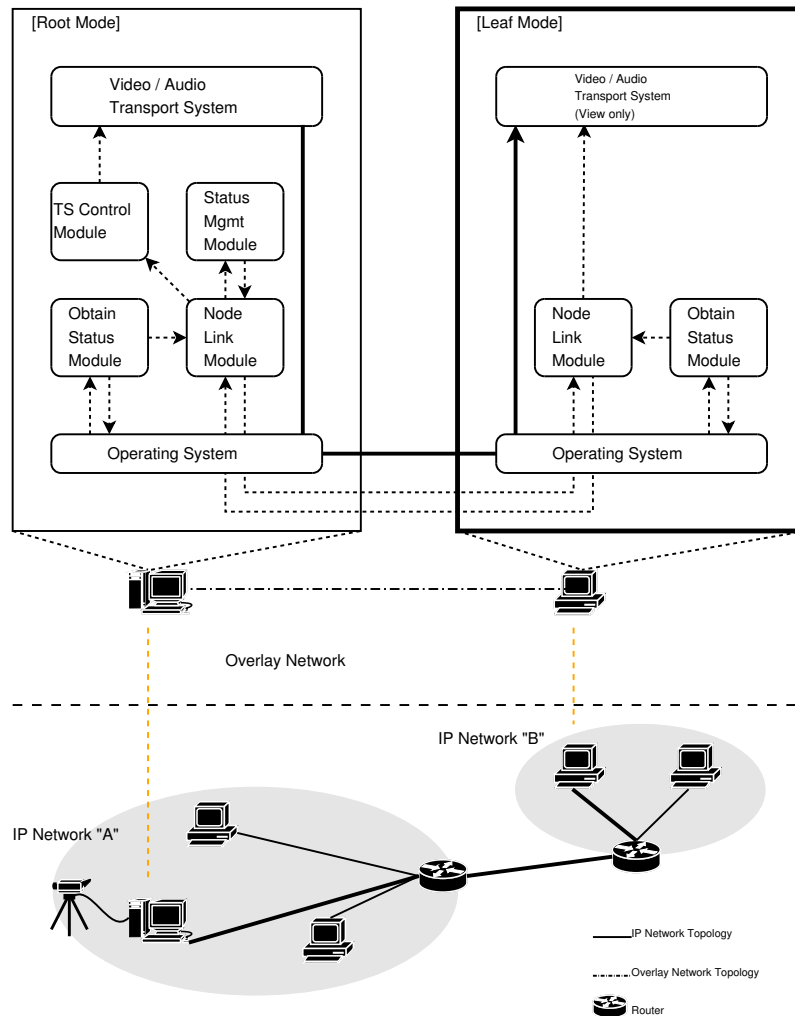


図 5.4: 動作概要:Leaf Mode

1. ノード情報取得モジュールによる OS からのノード情報の取得
2. ルートノードのノード情報管理モジュールへのノード情報の追加
3. ルートノードによる上流ノードの決定・映像データの受信

## 5.2 ALM 機構の機能

ALM 機構では計算機資源の数値化及びノードのメトリックの設定, メトリックに基づくオーバーレイネットワーク上での論理リンクの構成, ノード情報の管理・ノード間リンクの保持, 映像配信機構の制御を行う。

### 5.2.1 ノード情報取得モジュール

ノード情報取得モジュールでは, 配信網に参加する全てのノードについて一定の間隔で計算機資源の取得を行う。得られたデータはノード間リンクモジュールを経由してノード情報管理モジュールで使用される。

ノード情報取得モジュールの機能

ノード情報取得モジュールの機能を以下に挙げる。

- 演算能力測定  
OS が提供するデータに基づいて, 配信網に参加する全てのノードの演算能力を測定する。測定値は CPU アーキテクチャによる差が生じないように CPU 係数を用いて修正する。
- CPU・メモリ使用率測定  
OS が提供するデータに基づいて CPU・メモリの使用率を測定する。測定した使用率は演算能力の測定値と組み合わせ推定消費資源の算出に使用する。
- 実効帯域測定  
外部のツールもしくは OS が提供するデータに基づいて, 実効帯域の測定を行う。測定値は映像配信機構の使用帯域と比較し, 最大接続可能ノード数を決定するために使用する。

### 5.2.2 ノード間リンクモジュール

ノード間リンクモジュールでは, ノード情報取得モジュールを用いて計測されたノード情報をルートノードのノード情報管理モジュールへ送信する。また, ノード同士で生存確認を行う。さらに, ノード情報管理モジュールが送信した映像配信機構の制御命令を映像配信機構モジュールへ伝達する。

ノード間リンクモジュールの機能

ノード間リンクモジュールの機能を以下に挙げる。

## 5.2. ALM 機構の機能

- ノード情報の送受信

ノード情報の送受信は全て通信を行う各ノードのノード間リンクモジュールを経由して行われる。ノード情報取得モジュールが提供するデータをルートノードのノード情報管理モジュールに送信する。

- ノード同士の生存確認

配信網に参加している全てのノードは一定の間隔で生存確認を行う。ルートノードはノード情報管理モジュールによって保持しているノードに対して生存確認を行う。エッジノードは下流のノードに対して行う。ルートノード以外のノードが上流のノードとの生存確認が不可能になった場合、ルートノードに問い合わせを行う。

- 制御命令の伝達

ルートノードのノード情報管理モジュールが送信した映像配信機構の制御命令を映像配信機構制御モジュールに伝達する。

### 5.2.3 ノード情報管理モジュール

ノード情報管理モジュールでは、配信網に参加する全てのノード情報の管理を行う。ノード情報管理モジュールは基本的にルートノードにおいて使用する。また、ノードのルートノードに対する接続ノードの問い合わせに対して最適な接続ノードの検索と映像配信機構の制御命令の送信を行う。

#### ノード情報管理モジュールの機能

ノード情報管理モジュールの機能を以下に挙げる。

- ノード情報の追加・削除・更新

ノード間リンクモジュールを経由して配信網に参加する全てのノードからノード情報を取得し、データをモジュール内のデータベースに追加する。また、ノード同士の生存確認が不可能になったノードのデータの削除、ノード情報が変更されたノードのデータの更新を行う。

- 最適な接続ノードの検索

配信網に参加する新規ノードや、上流ノードとの生存確認が不可能になったノードはルートノードに対して問い合わせを行う。ノード情報管理モジュールはノード情報の参照を行い、問い合わせ行ったノードと接続可能なノードの条件を比較する。また、計算機資源に余剰のあるノードを優先して接続ノードを決定する。



### 5.3. 映像配信機構の機能

- 映像配信機構の制御命令の送信

問い合わせノードに対応する最適な接続ノードが決定された後、接続ノードの映像配信機構制御モジュールに新たな送信先データを送信する。送信先データにはノードの IP アドレス・映像品質を含む。

#### 5.2.4 映像配信機構制御モジュール

映像配信機構制御モジュールでは、ノード情報管理モジュールが送信した制御命令をノード間リンクモジュール経由で受信し、映像配信機構の制御を行う。本モジュールが制御を行うことによって映像・音声データの流れが決定する。

映像配信機構制御モジュールの機能

映像配信機構制御モジュールの機能を以下に挙げる。

- 映像配信機構の制御

ルートノードのノード情報管理モジュールが送信した制御命令を、ノード間リンクモジュールを経由して受信する。これに基づいて外部の映像配信機構に対して送受信などの制御を行う。これは ALM 機構・映像配信機構双方が処理可能である汎用的なデータ形式で制御データの伝達を行う。

### 5.3 映像配信機構の機能

本研究の対象とする配信形態は、定常的に広域への配信に対応する機構が存在しない環境において、即時性を要求される単一のコンテンツを広域へ到達させる形である。本機構における映像配信機構には以下の要求事項がある。

- 配信データの複製による再配信
- 配信データの品質制御による資源の再配分

#### 5.3.1 配信データの複製

本機構では ALM 機構で構成された配信網内でノード同士が Unicast による映像配信を行う。このため、リーフノード以外のノードでは配信データの複製を行い、再配信を行う。再配信を行う際、ノードは図 5.5 のように最低 1 本の複製したストリームを出力する。余剰資源を有するノードでは図 5.6 のように多数のストリームを出力する。

5.3. 映像配信機構の機能

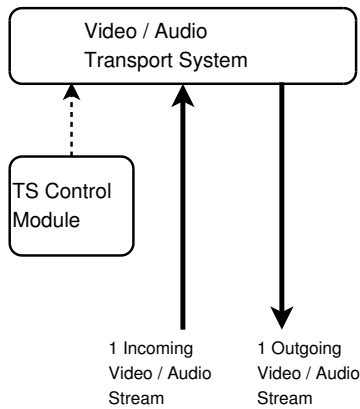


図 5.5: 1 : 1 の複製

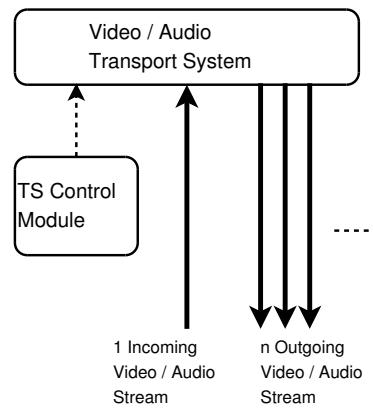


図 5.6: 1 : n の複製

5.3.2 配信データの品質制御

本機構では利用可能資源がデータを配信するノードに接続するための要求資源に満たないノードに対しての取得機会の拡大を実現するために、配信データの品質制御を行う。図 5.7は品質制御を行わない配信, 図 5.8は品質を 1/3 に落とした場合である。

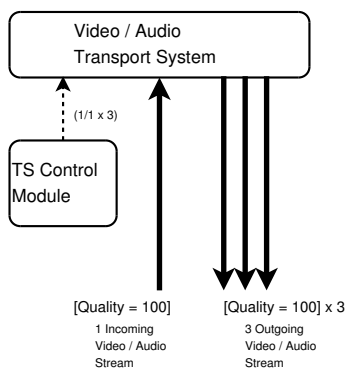


図 5.7: 品質制御:無

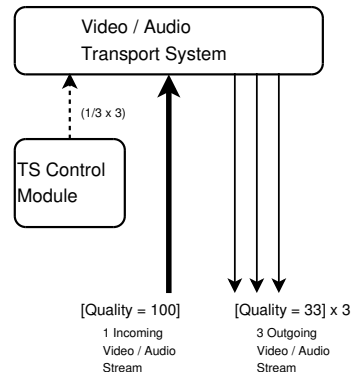


図 5.8: 品質制御:有

品質制御を行うために、本機構は品質を変化させることが可能な何らかのストリーミング機能を有する。代表的な方法を表 5.2に示す。

表 5.2: 品質制御の方法

	フレーム間引き	フレーム間圧縮	フレーム内圧縮
代表例	DV	MPEG2	MotionJPEG
圧縮率	-		
品質劣化			
要求資源			×

# 第6章 計算機資源に基づく ALM映像配信機構の実装

本章では, 5章の設計に基づいてプロトタイプ実装を行った ALM 映像配信機構 RBMcast (Resource-Based Multicast) について述べる.

## 6.1 実装概要

映像配信機構としてインターネット上で DV データの送受信を行う DVTS(Digital Video Transport System) を対象に, 計算機資源に基づく ALM 映像配信機構である RBMcast を設計し, その設計に基づいて実装を行った.

## 6.2 実装環境

本機構の実装を行ったソフトウェア環境を表 6.1に示す.

表 6.1: 実装ソフトウェア環境

	環境 1	環境 2
OS	VineLinux 3.0	RedHat Linux 9
プログラミング言語	C 言語	
コンパイラ	gcc 3.3.2	gcc 3.2.2
ライブラリ	glibc 2.3.3	glibc 2.3.2

本機構の実装を行ったハードウェア環境を表 6.2に示す.

## 6.3 DVTS

本機構では映像配信機構として DVTS(Digital Video Transport system) を用いる. DVTS は民生用デジタル AV 機器を用いた映像・音声配信システムである. DVTS は IEEE1394 インタフェースから DV フォーマットのストリームを取得し, そのストリー

表 6.2: 実装ハードウェア環境

	環境 1	環境 2
CPU	Celeron 400MHz	Pentium 4 2.40GHz
Memory	256MB	1024MB
HDD	6.4GB	120GB
NIC	100Base-TX	100Base-TX
DV 機器 (1)	SONY DV VCR DCR-PC10	

ムに対し IP データグラム化を行う。そして、インターネットを介して映像・音声の転送を行う。DVTS はリアルタイム配信に対応する品質制御可能な映像転送機構という設計要件を満たすシステムである。これらの特徴より、本機構の映像配信機構は DVTS を拡張する形で実装を行う。

## 6.4 ALM 機構

本機構において、ALM 機構には以下の機能の実装を行った。

### 6.4.1 ノード情報取得

ノード情報として演算能力とデータリンク帯域の取得を行う。本実装では OS として Linux を用いているため、演算能力情報は `proc file system`、データリンク帯域情報は MII(Media Independent Interface) によって提供されている。本実装ではこれらの値の取得を関数 (`get_bogomips`, `get_ifspeed`) 化した。

#### 演算能力

演算能力の指標として Linux Kernel が `proc file system` を用いて提供している `Bogomips` 値を用いる。`Bogomips` 値は `/proc/cpuinfo` ファイルから取得が可能であるため、図 6.1 の `get_bogomips` 関数を作成した。

#### データリンク帯域

ノードのリンク速度の取得方法として Linux の MII が提供する図 6.2 の `show_basic_mii` 関数を利用する。MII はメディアに依存しないインタフェースの状態を閲覧・操作するためのツールである。データリンク帯域の取得には MII を用いて `get_ifspeed` 関数を作成した。

```
double get_bogomips(void)
{
    (省略)
    char *p;
    double bogomips;

    /* Open /proc/cpuinfo Proc File System */
    FILE *fp = fopen("/proc/cpuinfo", "r");
    if(fp == NULL){
        perror("fopen");
    }
    /* Compare Strings */
    while ((i = strncmp(buf, str_bogomips, 8)) != 0){
        fgets(buf, sizeof(buf), fp);
    }
    fclose(fp);
    (省略)
    /* Return Bogomips Value */
    return (bogomips);
}
```

図 6.1: get\_bogomips 関数

### 6.4.2 ノード間リンク

#### ノード情報の送受信

ノード情報の管理には図 6.3 の `rbm_node_t` 構造体を `nodedb` 構造体配列に格納し利用する。ノード情報の初期設定値はノードのシリアル番号である `nodeid`, ノードの計算機資源メトリックである `metric`, ノードの要求する DV データの受信フレームレートである `recv_rate`, DV データの送信時 (ルートノード)・再送信時 (エッジノード) のフレームレートである `send_rate`, 送信可能な DV ストリームの本数である `send_stream`, 及びノードの IP アドレスである。

#### ノードの生存確認

ノード間リンクを保持するため, ルートノード・エッジノード間, DVTS の送信ノード

```
int show_basic_mii(long iaddr, int phy_id)
{
    /* Scan for the highest negotiated capability */
    for (mii_reg = 0; mii_reg < 8; mii_reg++)
        mii_val[mii_reg] = mdio_read(iaddr, phy_id, mii_reg);
    (省略)
}

int mdio_read(int skfd, int phy_id, int location)
{
    /* read MII PHY register */
    (省略)
}
```

図 6.2: show\_basic\_mii 関数

```
struct rbm_node_t{
    /* Node ID & Node METRIC */
    int nodeid;
    int metric;

    /* Request Receive DV Stream Frame Rate */
    int recv_rate;

    /* Send DV Stream Fram Rate & Number of Stream */
    int send_rate;
    int send_stream;

    /* Node IP Address */
    struct sockaddr_in addr;
};
```

図 6.3: rbm\_node\_t 構造体

ド・受信ノード間で図 6.4 の TCP Socket を作成し, `connect()` と `accept()` を利用した生存確認を行う。

```
{
int listen_fd, fd, rc, len;
    (省略)
    /* create socket, bind, and listen */
    listen_fd = socket(AF_INET, SOCK_STREAM, 0);

    /* waiting for connection */
    for(;;){
        fd = accept(listen_fd, (struct sockaddr *)&inet_addr, &len);
    }
}

#define SLEEP_SEC 3

{
int fd, rc, len;
    (省略)
    /* connect with a node */
    rc = connect(fd, (struct sockaddr *)&inet_addr, len);

    /* transmits to the connected socket */
    for(;;){
        rc = send(fd, buf, sizeof(buf), 0);
        /* check every 3sec */
        sleep(SLEEP_SEC);
    }
}
```

図 6.4: ノードの生存確認



## 6.5. 映像配信機構

### 6.4.3 ノード情報管理

RBMcast 配信網に参加しているノードのノード情報は、ルートノードの `nodedb` 構造体配列内で管理を行う。ルートノードは配信網を構成する他の全てのノードから `rbm_node_t` 形式のデータを受信する。

#### 接続ノードの検索

ルートノードは新規に配信網に対して接続要求があった場合、または既に接続済みのノードが上流ノードとの切断によって配信網の再構成を要求する場合、ノードデータベースである `nodedb` 構造体配列を検索条件に応じてソートする。本実装ではソートアルゴリズムとしてバブルソートを用いた。

### 6.4.4 DVTS 制御

参加ノードの要求に応じてルートノードはエッジノードにおける映像配信機構である DVTS の開始・停止及び DV データの送信先・送信品質の指定を行う必要がある。本実装では、送信先としてルートノードから指定された接続ノードの IP アドレス、接続ノードが要求するフレームレートをファイルに書き出すことで DVTS 制御機構と DVTS との間のデータの受け渡しを行う。

## 6.5 映像配信機構

本機構において、映像配信機構には以下の機能の実装を行った。

### 6.5.1 DV データの複製

### 6.5.2 DV データの品質制御

## 第7章 評価

本章では, 第6章において実装を行った RBMcast に対する評価について述べる.

### 7.1 評価概要

本評価は, 本研究の提案するコンテンツ配信モデルの有効性の検証を目的とする. 定性評価及び定量評価は, 以下の項目について行う.

- 定性評価:設計に基づき実装を行った RBMcast の実現した機能
- 定量評価:RBMcast における ALM 機構の性能

### 7.2 本機構の実現した機能

本機構は, 映像配信機構に DVTS を用いて計算機資源に基づく ALM 映像配信機構を実現した. 本機構が実現した機能は以下の5項目である.

- 計算機資源の数値化及び対応するメトリックの設定
- メトリックに基づく ALM 配信網の構成
- ノード情報の管理及び配信網の再構成
- リアルタイム配信に対応する品質制御可能な映像配信機構
- 映像配信機構の外部からの制御

本機構のシステムモデルでは, コンテンツ配信を行うノードは, ツリー構造の頂点であるルートノードとなる. 配信網の参加しているノードの内, 余剰資源のあるノードは受信ノードとしての動作以外に自分自身が参加しているコンテンツの再配信を行う.

また, 利用可能資源が要求に満たないノードのために, 余剰資源を有するノードが配信データの品質制御を行う. 再配信や品質制御は計算機資源に基づく指標であるメトリックの値が小さいノードに優先的に割り振られる.

このシステムモデルが実現されたかを実機を用いて動作を検証する.

## 検証 1: エッジノードの参加

本検証では, RBMcast 配信網のエッジとなるノードが ALM 機構への参加を行う. 配信網におけるノードの初期状態を図 7.1 に示す. 図中の PC 'A' は配信元となるルートノード, PC 'B' は既に配信網に参加済みの他のノードである. ルートノードが保持しているノードデータベースが `nodeb[]` である. 新規の参加ノードである PC 'C' はルートノードに対して問い合わせを行う.

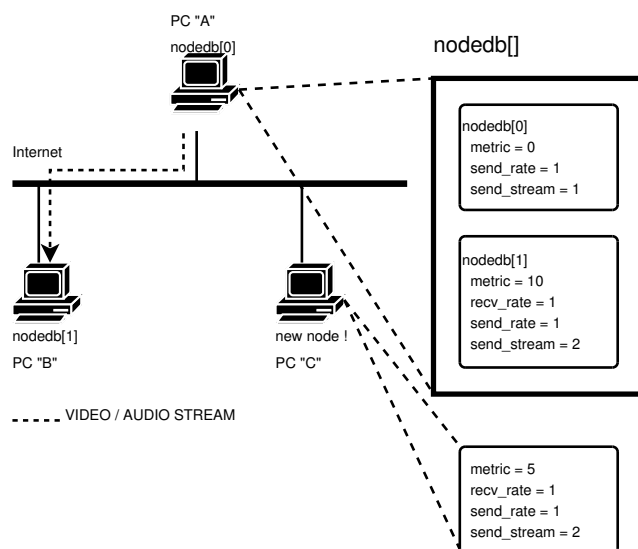


図 7.1: ノードの初期状態

PC 'C' からの接続要求を受信したルートノードは `nodeb[]` を参照して, 接続可能なノードの中からメトリックの値が小さい `nodeb[0]` を配信元として設定する. ALM 機構への参加が完了したノードの状態を図 7.2 に示す.

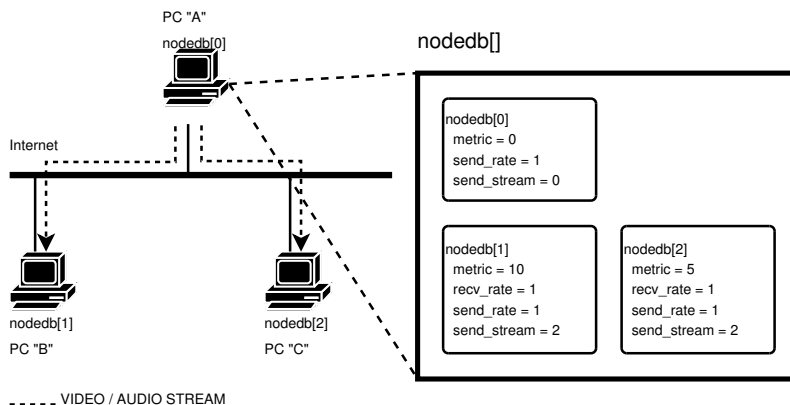


図 7.2: エッジノードの参加状態

これにより, PC 'C' はエッジノードとして配信網に参加できたことが分かる.

#### 検証 2:リーフノードの参加

本検証では, RBMcast 配信網のリーフとなるノードが ALM 機構への参加を行う. 配信網におけるノードの初期状態の図を 7.3 に示す. 図中の PC 'B', PC 'C' は既に配信網に参加済みのエッジノードである. 新規の参加ノードである PC 'D' はルートノードに対して問い合わせを行う.

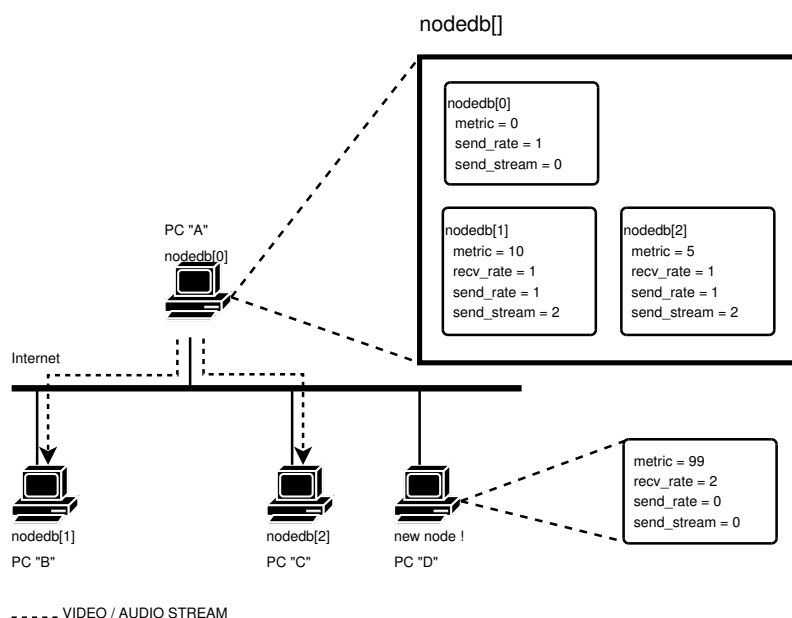


図 7.3: ノードの初期状態

PC 'D' から接続要求を受信したルートノードは nodedb を参照して,  $recv\_rate = 2$  である PC 'D' が接続可能なノードを検索する. その中からメトリックの値が小さい nodedb[2] を配信元として設定する. ALM 機構への参加が完了したノードの状態を図 7.4 に示す.

これにより, PC 'D' はリーフノードとして配信網に参加できたことが分かる.

## 7.3 ALM 機構の性能

RBMcast の ALM 機構において, 参加ノードの要求条件に応じてルートノードが有するノードデータベースから最適な条件の接続ノードを検索し, 参加ノードと接続ノードのマッチングを行うノード情報管理モジュールの性能を検証する. 評価はノード情報管理モジュールが, 検索時に参照するデータベースに蓄積されているノード数や検索条件の数を変化させることで行う.

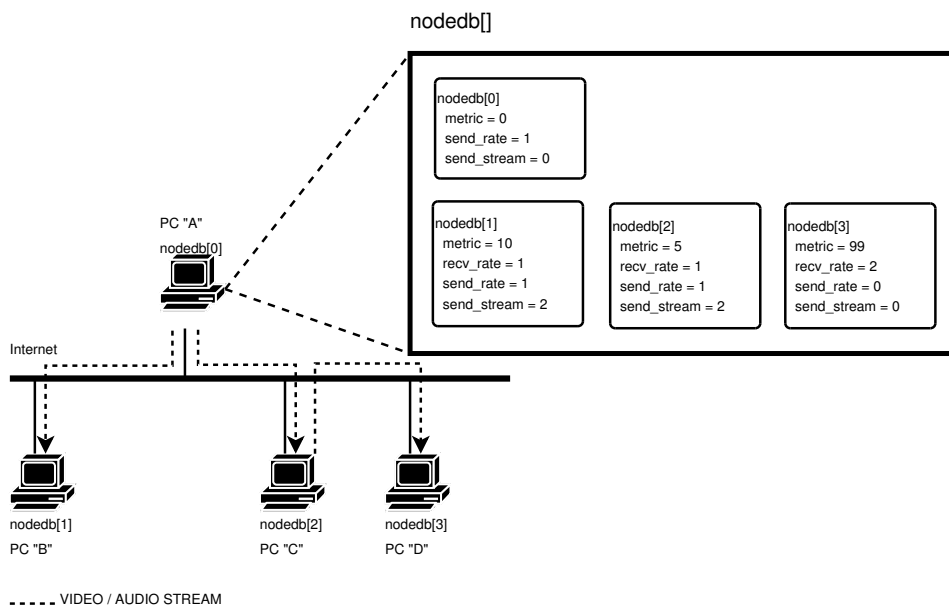


図 7.4: リーフノードの参加状態

また, RBMcast 配信網の参加ノードにおいて上流ノードとのノード間リンクが何らかの原因で切断され, 配信データの欠損が発生した場合, 配信網の再構成を行い再びデータを受信するまでのデータの欠損量の測定を行い RBMcast の実用性を検証する.

### 7.3.1 評価環境

ALM 機構のノード情報管理モジュールの性能検証及び評価を行った環境を表 7.1 に示す.

表 7.1: 評価環境

OS	VineLinux 3.0
Linux Kernel	2.4.26
CPU	Celeron 400MHz
Memory	256MB
HDD	6.4GB

### 7.3.2 ノード数によるマッチング時間

本節では RBMcast を用いた配信網の参加ノード数によるマッチング時間の変化の測定を行う。RBMcast のノード情報管理モジュールに対してランダムに生成したノードのダミーデータをデータベースに与え、ノードの参加要求に対してデータベースから最適な条件の接続ノードを検索するまでのマッチング時間を測定する。

検索条件は本機構の初期設定値である 3 項目、送信可能本数 (send\_stream)、計算機資源メトリック (metric)、送信可能レート (send\_rate) に設定した。

なお、マッチング時間の計測には gettimeofday 関数を用いた。

#### マッチング時間の安定性

100 ノード分のダミーデータを用いて同じ検索条件でのマッチング時間の測定を 100 回行い、検索条件が同条件におけるマッチング時間の安定性の検証を行った。測定結果を図 7.5 に示す。

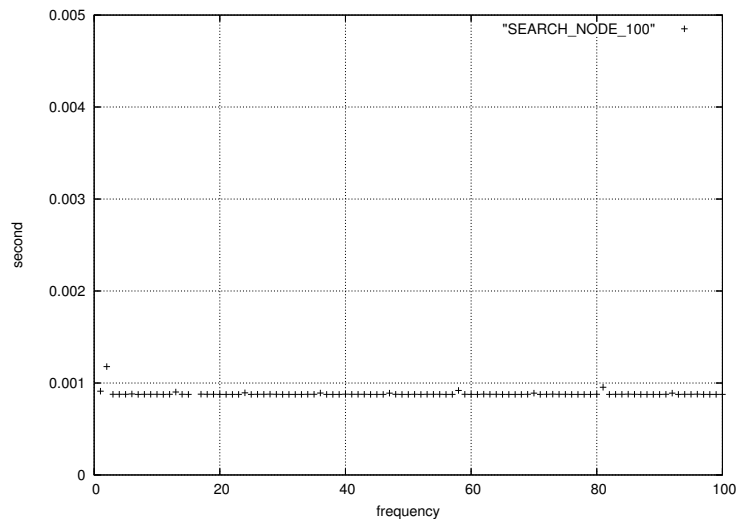


図 7.5: 検索回数とマッチング時間の変化

検索回数に関わらずマッチング時間はほとんど変化せず、安定しているのが分かる。

#### ノード数とマッチング時間

ノード管理モジュールに与えるダミーデータ数を 1 から 100 まで変化させ、参加ノード数の変化がマッチング時間に与える影響の検証を行った。測定結果を図 7.6 に示す。測定結果の信頼性を上げるため、検索に用いる参加ノードの要求値を変更して測定を 3 回行った。

参加ノード数の増加に伴ってマッチング時間が増加している。ノード数が 100 の場合

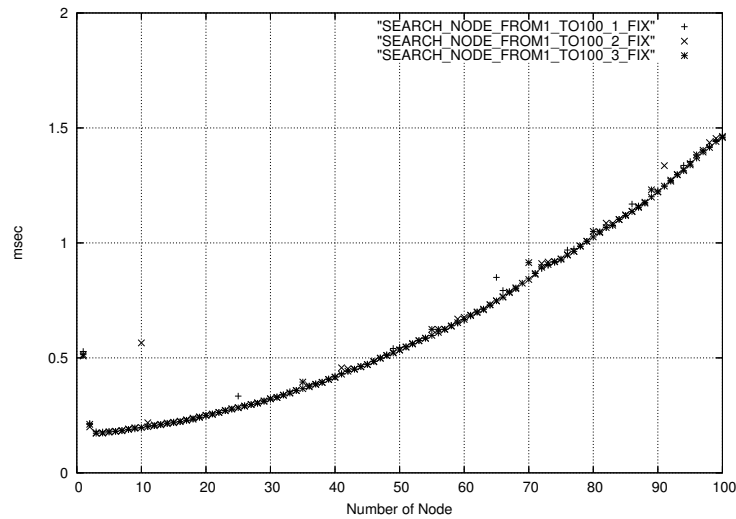


図 7.6: ノード数 (~100) とマッチング時間の変化

のマッチング時間は 1.5msec であり, 1 から 100 までのノード数の増加に伴う変化量は 1.6msec であることが分かる. 次に, ダミーデータを 1000 ノード分に増加した場合の測定結果を図 7.7 に示す.

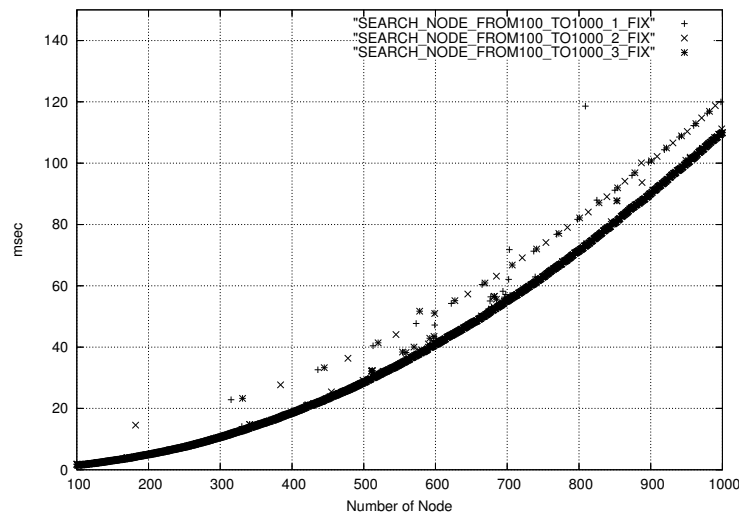


図 7.7: ノード数 (~1000) とマッチング時間の変化

100 ノードの場合と比較してさらに参加ノード数の増加に伴ってマッチング時間が増加している. ノード数が 1000 の場合のマッチング時間は 110msec 程度であり, 100 から 1000 までのノード数の増加に伴う変化量もほぼ同様であることが分かる.

### 7.3.3 検索条件数によるマッチング時間

本節では RBMcast を用いた配信網にノードが参加する際、検索条件によるマッチング時間の変化の測定を行う。検索条件を RBMcast のノード情報管理モジュールの初期設定値である 3 から変更し、詳細な検索条件におけるマッチングに要する時間を測定する。

#### 条件数とマッチング時間

100 ノード分のダミーデータを用いて検索条件を初期設定値の 3 項目から 10 項目まで変化させ、検索条件数の変化がマッチング時間に与える影響の検証を行った。測定結果を図 7.8 に示す。測定結果の信頼性を上げるため、検索に用いるダミーデータを変更して測定を 3 回行った。

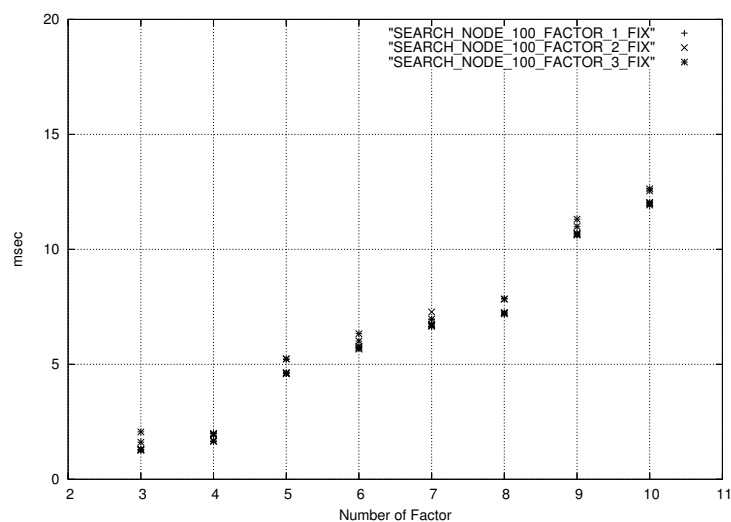


図 7.8: 条件数とマッチング時間の変化

検索条件数の増加に伴ってマッチング時間が増加している。参加ノード数が一定であるため、変化量が条件数に比例していることが分かる。

### 7.3.4 再構成時のデータ欠損量

本節では RBMcast を用いた配信網において、ある参加ノードとその上流ノードとのノード間リンクが切断され配信データの欠損が発生した時点から再びデータの受信が行われるまでに失われたデータ量の測定を行う。



## 第8章 結論

本章では、第6章における実装、第7章における評価に基づき本研究のまとめ及び今後の課題について述べる。

### 8.1 まとめ

本研究では計算機資源に基づく Application Layer Multicast(ALM) 映像配信機構の実装を行った。本機構を用いることにより、コンテンツ配信におけるユーザのコンテンツ取得機会の拡大と、コンテンツの品質とユーザ要求のマッチングを実現することができる。

サーバ・クライアント型やIPマルチキャストを用いた従来のコンテンツ配信モデルは、ユーザ環境の多様化や即時性の高いコンテンツなどに対応することができない。

既存の配信モデルには、ユーザのネットワーク環境への依存やコンテンツの配信元となるサーバの負荷の増大、エンドノード環境の格差による配信モデルのボトルネック発生が起こる可能性がある。これらの問題を解決するためには、オーバーレイネットワーク上での配信網の構成及びユーザ要求に基づく配信もとへの誘導の実現が必要である。

本研究では、計算機資源に応じたコンテンツ配信モデルを提案した。ユーザ環境によってはデータ処理に必要な計算機資源よりもエンドノード性能が勝っており余剰資源が存在する場合がある。インターネット上で扱われる映像コンテンツの場合、配信を受けたノードで余剰している計算機資源を用いて再配信を行うことが可能である。

そこで、計算機資源に基づく ALM 映像配信機構を設計し、以下の4つのモジュールの設計を行った。

- ノード情報取得モジュール  
ノードの計算機資源の取得及び数値化、メトリックの設定
- ノード間リンクモジュール  
ノード情報の送受信、ノード間リンクの保持(生存確認)
- ノード情報管理モジュール  
ノード情報の追加・削除・参照・更新
- 映像配信機構制御モジュール  
映像配信の開始・停止、宛先の指定、品質の指定

本機構では映像配信機構として DVTS(Digital Video Transport System) を拡張し、ネットワークから DV データを受信し再送信する機能と、再送信時に品質制御を行う機能を追加した実装を用いた。

本研究により、定常的に広域への配信に対応する機構が存在しない環境においてリアルタイム性の高いコンテンツの配信が可能になった。

## 8.2 今後の展望

本研究における今後の発展として以下の事項が挙げられる。

- 動的な配信網の再構成によるノード関係の最適化

本機構の実装ではノード管理に関する処理の単純化のため、既に RBMcast 配信網に参加しているノードに対して意図的に接続ノードの変更は行わない。本研究の ALM 機構に動的に配信網の構成を行う自律性を持たせる事で、ノード数が増加していった場合、計算機資源の再配分の効率が高くなるトポロジに自動的に修正することが可能になる。

しかし、現状の ALM 機構においてノードによる自律的な配信網の再構成はノードの接続と切断のくり返しの問題となる。このため次項の論理リンクの冗長化によって再構成時のコンテンツデータの欠損を防止する必要性が生じる。

- 論理リンクの冗長化によるデータ欠損の防止

本機構では、ノード間でコンテンツデータを送受信する論理リンクは 1 本である。配信網を構成しているノードが複数の同条件の上流ノードに対して、予め接続しておくことで論理リンクの冗長化を行うことができ、上流ノードの切断によって配信網を再構成する必要性が生じた場合、コンテンツデータの欠損を防止することが可能になる。

- メディア変換機構との連携

本機構の実装では映像配信機構として DVTS を用いたため、DV データ形式を受信可能なことが配信網への参加の前提条件となった。しかし、例えば MPEG2 における GOP(Group of Pictures) を間引くといったトランスコード技術や、映像データを一度デコードし別のデータ形式にエンコードすることによる品質制御を行う、再エンコード技術を利用したメディア変換機構と連携することでデータ形式への依存性が解消される。

# 謝辞

本研究を進めるにあたり、御指導を頂きました、慶應義塾大学環境情報学部教授の村井純博士、徳田英幸博士、同学部助教授の楠本博之博士、中村修博士、同大学環境情報学部専任講師の南政樹氏、重近範行氏に感謝致します。

絶えずご指導とご助言を頂きました慶應義塾大学政策・メディア研究科大学助手の杉浦一徳氏、慶應義塾大学大学院政策・メディア研究科修士課程の入野仁志氏、久松剛氏に感謝致します。

## 参考文献

- [1] RealNetworks, Inc. RealPlayer WWW page. <http://www.real.com/>.
- [2] RealNetworks, Inc. RealPlayer WWW page. <http://www.real.com/>.
- [3] S. Deering. RFC1112 Host Extensions for IP Multicasting. <http://www.ietf.org/rfc/rfc1112.txt>, August 1989.
- [4] Yang-hua Chu, Sanjay G. Rao, Srinivasan Seshan and Hui Zhang. A Case for End System Multicast. *ACM SIGMETRICS 2000*, June 2000.
- [5] Cisco Systems Inc. Overview - Multicast Routing. <http://www.cisco.com/warp/public/614/17.html>, March 1999.
- [6] R. Boivie, N. Feldman, IBM, Y. Imai, Fujitsu, W. Livens, Colt Telecom, D. Ooms, OneSparrow, O. Paridaens, Alcate, E.Muramoto and Panasonic. Explicit Multicast (Xcast) Basic Specification. <http://www.ietf.org/internet-drafts/draft-ooms-xcast-basic-spec-06.txt>, June 2004.
- [7] 三村 和, 中内 清秀, 森川 博之, 青山 友紀. RelayCast : ピアツーピア型ストリーム配信のためのミドルウェア. 電子情報通信学会技術研究報告 Vol.102 No.214, pages 7–12, July 2002.
- [8] Satish Ramakrishnan and Vibha Dayal. The PointCast Network. *the 1998 ACM SIGMOD international conference on Management of data*, page 520, June 1998.
- [9] EntryPoint Inc. EntryPoint. <http://www.entrypoint.com/>.
- [10] CMP Media LLC. PointCast Pushes Multicast Capabilities. <http://www.internetweek.com/news/news0305-7.htm>, March 1998.
- [11] Wim van Dorst. What are BogoMips. <http://www.tldp.org/HOWTO/BogoMips/x78.html>, September 2004.
- [12] Wim van Dorst. BogoMips mini-Howto. <http://www.tldp.org/HOWTO/mini/BogoMips.html>, September 2004.

- [13] Intel Corporation. Pentium(R) III Processor Implementation Tradeoffs.  
*[http://www.intel.com/technology/itj/q21999/articles/art\\_2.htm](http://www.intel.com/technology/itj/q21999/articles/art_2.htm)*, Q2 1999.
- [14] Intel Corporation. The Microarchitecture of the IntelR PentiumR 4 Processor on 90nm Technology.  
*[http://www.intel.com/technology/itj/2004/volume08issue01/art01\\_microarchitecture/p01\\_abstract.htm](http://www.intel.com/technology/itj/2004/volume08issue01/art01_microarchitecture/p01_abstract.htm)*, February 2004.
- [15] Intel Corporation. The IntelR PentiumR M Processor: Microarchitecture and Performance.  
*[http://www.intel.com/technology/itj/2003/volume07issue02/art03\\_pentiumm/p01\\_abstract.htm](http://www.intel.com/technology/itj/2003/volume07issue02/art03_pentiumm/p01_abstract.htm)*, May 2003.
- [16] Akimichi Ogawa. DVTS (Digital Video Transport Sysyem) WWW page.  
*<http://www.sfc.wide.ad.jp/DVTS/>*, November 2001.