平成15年度　　修士論文

大規模コンテンツ配信実現のための コンテンツ配信アーキテクチャに関する検討

指導教授　渡辺 裕 教授

2003年3月

早稲田大学大学院 国際情報通信研究科
国際情報通信学専攻

4202A128-9
清　水　　直　人
Naoto SHIMIZU

目 次

第1章

序論

1.1 研究の背景

近年，FDDI・xDSL・CATV に代表される高速ディジタル通信環境が爆発的な普及を見せ，今やインターネットは全ての産業の社会的基盤となるまでに至った．現在では，インターネット接続型携帯電話・IEEE802.11a・b・g などに代表される無線 LAN 規格に準拠した機器の登場を背景に，様々な機器からのインターネットへの接続が試みられている．インターネットは，あらゆる機器をつなぐユビキタス・ネットワークとして新たな局面を迎えようとしている (図 1–1).

半導体分野においては未だにムーアの法則の有効性は維持され，計算機の性能は飛躍的に向上している．このように通信・半導体の両分野において新たな革新的な技術が登場し，急速な進歩を遂げている中，マルチメディアコンテンツに代表される大容量コンテンツを含めたあらゆるコンテンツのインターネット上での効率的な配信・流通への要求が今まで以上に高まってきている．

本論文における効率的な・円滑なコンテンツ流通とは，1) 同時に多数のユーザ端末に対してコンテンツを配信できる，2) 実時間型 (ストリーミング型) コンテンツ配信において，良好な画像品質で配信できる，3) 蓄積型 (ダウンロード型) コンテンツを短時間で配信できる，ということを実現することである．

円滑なコンテンツ流通実現に向け，以下に大別される様々な研究がなされている．

1. 柔軟性，信頼性，スケーラビリティに優れた伝送ネットワークの実現

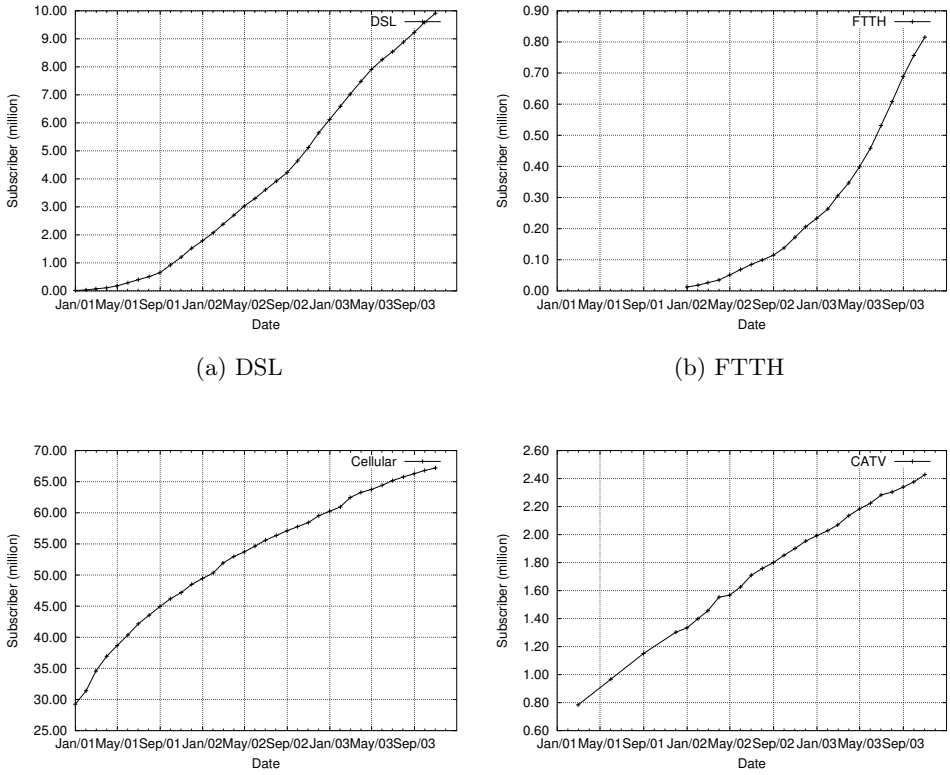
   1 対 1 (送信者 1 対受信者 1)，1 対多 (送信者 1 対受信者多，同報型通信)，実時間型，蓄積型など様々なコンテンツ配信への要求に対応し，ネットワークプロバイダおよびユーザ両者の視点からみて総合的に優れたネットワークの提供を目指す．

   これらを構成する技術として，

   a) コンテンツ配信アーキテクチャ

      CDN (Contents Delivery Network)，P2P (Peer-to-Peer) など．

   b) コンテンツ配信プロトコル



(a) DSL　(b) FTTH

(c) 携帯電話　(d) CATV

図 1–1: 各インターネット接続サービス加入者数の推移 (1)

TCP(Transfer Control Protocol)，UDP(User Datagram Protocol), RTP (Real-time Transfer Protocol) など.

c) コンテンツ配信技術

IP マルチキャストなど.

d) 経路制御技術

OSPF (Open Shortest Path First), RIP (Routing Information Protocol) など.

e) Application layer QoS control

各端末における輻輳制御，レート制御，FEC (Forward Error Correction) に代表されるパケットロスが存在する状況下でのエラーコントロールなど.

f) Network layer QoS control

Diffserv, Intserv, トランスコーダ付ルータにおけるレート制御技術など.

などの技術が挙げられる.

2. 効率の良い情報源符号化の実現

データ中の冗長性などを検出し，より高圧縮率の情報源符号化方式の実現を目指す.

映像，音声などのマルチメディアコンテンツにおいては，ISO/IEC JTC1 において規格化された国際標準符号化である MPEG-1, MPEG-2, MPEG-4 などが挙げられる．

3. 情報源符号化，ネットワークを統合した情報流通フレームワークの実現

通信プロトコル，符号化，メタデータ定義，著作権管理などコンテンツを円滑で流通する上で必要不可欠な技術の統合化を目指す．

例として，MPEG-21 が挙げれられる．これらの技術が標準化されることで，各端末間での互換性が維持され，円滑なコンテンツ流通が実現される．

従来より，1. 柔軟性，信頼性，スケーラビリティに優れた伝送ネットワークの実現および 2. 効率の良い情報源符号化の実現の研究は，それぞれのフィールドでは，優れた方式が多数提案されてきていた．しかしその一方で，それらがコンテンツ配信システム・アーキテクチャとして結合し，優れたコンテンツ流通サービスの提供が成されるまでには到っていなかった．そこで，現在，情報流通フレームワークが提案され，これら技術の真の結合を実現しようとしている．これにより，インターネットを通じ，多種多様なコンテンツが，様々な端末間で流通する優れたコンテンツ配信サービスが可能となる．例えば，放送メディアを通じて配信されていたマルチメディアコンテンツを，インターネットを介して配信しようとする放送と通信の融合もこのようなコンテンツ配信サービスが実現することで可能となるであろう．

しかし，現時点において，大多数の受信者に対して，インターネットを通じ，映像コンテンツのような大容量コンテンツを配信するための伝送手法の確立が成されていない．したがって，例え情報フレームワークが実現されても，大容量コンテンツの円滑な流通は困難なものとなる．そこで，本論文では，大多数の受信者に対応したコンテンツ配信ネットワークアーキテクチャに関する検討を行い，大容量コンテンツの円滑な流通を実現するための提案を行う．

1.2　本論文の目的

本論文では，前節で挙げた柔軟性，信頼性，スケーラビリティに優れた伝送ネットワークの実現を目指し，より多くの受信者に対する実時間型および蓄積型コンテンツの配信実現手法について提案を行う．特に，受信者千人程度の中規模コンテンツ配信，また現時点でまだサービスが提供されていない受信者一万，十万，百万規模の大規模コンテンツ配信の実現を目指す．

中・大規模コンテンツ配信においては，受信者側の帯域はもとよりコンテンツ送信側の処理能力，帯域が大きな問題となる．この問題を解決するための方法としては，IP マルチキャストの利用，CDN に代表されるエッジサーバ (コンテンツ複製配置サーバ) の配置，P2P 型プロトコルの利用などが挙げられる．

そこで本論文では，まずコンテンツ複製配置を利用したコンテンツ配信に着目する．そして，コンテンツ複製配置を利用したコンテンツ配信アーキテクチャ上では，以下の 2 点が重

要であることを明らかにする．

1. コンテンツ発見手法

   サービス利用者の要求コンテンツを保持している端末を発見する手法．

2. 最適ホスト選択手法

   要求されたコンテンツを複数のホストが保持していた際に最適な端末を発見する手法．

その上で，最適ホスト選択手法に着目し，各ホストのインターネット上での位置を考慮した最適ホスト選択手法の提案を行う．

次に，各端末でパケットの複製，マルチキャストルーティングなどのマルチキャストに関する機能を実現するアプリケーションレベルマルチキャスト (ALM) を用いたコンテンツ配信アーキテクチャを提案する．ALM では，IP マルチキャストルータではなく，各ホストがマルチキャストツリー構築の役割を担う．これにより，IP マルチキャストのようなネットワークインフラストラクチャの変更を要求することなく，多受信者に対して効率の良いコンテンツ配信が実現できる．しかし，その一方で，IP マルチキャストルータと比べ，セッションへの参加，離脱などが頻繁に起こる安定性のないホストがマルチキャストツリーの構築を行うことになる．そこで本論文では，ALM において，ホストの参加，離脱などのネットワーク状況の変化に追随できるマルチキャストツリー構築手法についても提案を行う．

本論文は，1) 最適ホスト選択手法の提案，2) ALM ツリー構築手法の提案と一見趣の異なる二つの目的より構成されるが，これらは密接に関連している．提案する最適ホスト選択手法は，複数の選択ホスト候補群の中から自ホストに遅延，帯域，ホップ数などのネットワーク的な距離が近いホストを選択する手法である．したがって，ALM ツリー構築過程における近接ホスト発見手法としても適用可能だからである．

最後に，能力 (CPU，メモリ，ネットワークの接続状態など) が異なる様々な端末から構成されるネットワークにおける映像コンテンツの大規模配信実現に向けての考察を行う．このようなネットワーク上では，Motion JPEG2000，MPEG-2, MPEG-4 における空間スケーラビリティ，SNR スケーラビリティのような多階層を有するデータの配信が有効である．そして，より大規模コンテンツ配信を実現するための今後の課題，将来動向を予測する．

1.3　用語の定義

本論文で用いる用語の意味を定義する．**端末**は，PC，携帯端末などのネットワーク上に接続された演算能力を有する機器のことを示す．**エンドホスト**，あるいは**ホスト**も同義である．**サーバ**は，コンテンツ配信ネットワークにおいて，停止・離脱することなく，コンテンツを提供し続ける端末を指す．停止・離脱する一般の端末との区別を，特に行いたい場合に用いる．したがって，サーバ $\subseteq$ 端末 (ホスト) である．**ノード**は，IP マルチキャストや ALM における配信ツリーを構成する要素を示す．例えば，IP マルチキャストの場合，マルチキャストルータが**ノード**であり，また，ALM の場合は，各端末が**ノード**である．**オーバーレイ**

ネットワークは，ある特定のサービスやアプリケーションを IP 層より上位層で柔軟に展開するために，論理的に構築されるネットワークのことであり，**物理ネットワーク**上に構築される．

1.4　本論文の構成

以下に本論文の構成を示す．

第 1 章は，本章であり，本論文の背景，目的について述べた．

第 2 章では，コンテンツ配信ネットワークと題し，基本的なコンテンツ流通アーキテクチャ(サーバクライアント型アーキテクチャ・P2P 型アーキテクチャ) およびそれを構成する技術 (IP マルチキャスト，ALM など) について概観する．また，円滑なコンテンツ流通を実現する上での問題点を整理する．

第 3 章では，最適ホスト選択手法と題し，まず従来より提案されている最適ホスト選択手法を整理する．最適ホスト選択手法のメトリックとして，RTT が優れていることを指摘した上で，RTT 予測手法である Global Network Positioning(GNP) を応用した最適ホスト選択手法について提案を行う．シミュレーションにより，提案手法の評価を行う．

第 4 章では，ALM によるコンテンツ配信と題し，近年盛んに研究されている ALM 手法に関して，従来手法を整理する．まず，ALM ツリーの基本的な構築方法について述べる．次に ALM の性能を測る上で用いられる性能評価尺度について述べる．そして，各 ALM 従来手法に関して，それらのアーキテクチャの詳細，それら手法の問題点を指摘する．ALM で用いられるツリー構築アルゴリズムについても言及する．

第 5 章では，クラスタリングを利用した ALM ツリー構築手法の提案と題し，各隣接端末のクラスタリングを用いた ALM ツリー構築手法について提案する．クラスタリングを用いることで，各端末のセッションからへの参加，離脱などの状況変化に追従できるツリーをコントロールオーバーヘッドを一定に保ちつつ，構築することが可能となるなど様々な利点がある．シミュレーションを通じて，提案手法性能評価実験を行い，提案手法の有効性を検証する．

第 6 章では，結論と題し，本論文の総括，および今後の課題について述べる．今後の課題では，本論文で提案した手法に関する課題のみではなく，より大規模なコンテンツ配信を実現するためには，どのようなコンテンツアーキテクチャが必要なのか，また様々な能力を持った端末より構成されるヘテロジーニアス環境に適したコンテンツ配信アーキテクチャとはどのようなものなのかということを中心としたコンテンツ配信アーキテクチャの将来動向を予測する．

付録 A では，ネットワークシミュレーションで用いられるネットワークトポロジについて整理する．本論文のシミュレーションにおいても，これらのネットワークトポロジを用いる．

第2章

コンテンツ配信ネットワーク

2.1 はじめに

本章では，まず基本的なコンテンツ配信アーキテクチャについて概観し，大規模コンテンツ配信を実現する上での課題を明らかにする．そのうえで，それら課題の解決に用いることが可能な技術として IP マルチキャスト，CDN，P2P(特に，ALM) の特徴についてまとめる．また，それら技術の問題点についても言及する．

2.2 円滑なコンテンツ流通を実現する上での課題

現在のコンテンツ配信アーキテクチャは次の 2 つに大別することが出来る [(2)]．

1. サーバクライアント型アーキテクチャ

   コンテンツを提供する端末 (サーバ) と，コンテンツを提供される端末 (クライアント) が明確に分離されたコンテンツ配信アーキテクチャ(図 2.1)．クライアントは，常にサーバより送信されたコンテンツを受信する．

2. Peer to Peer (P2P) 型アーキテクチャ

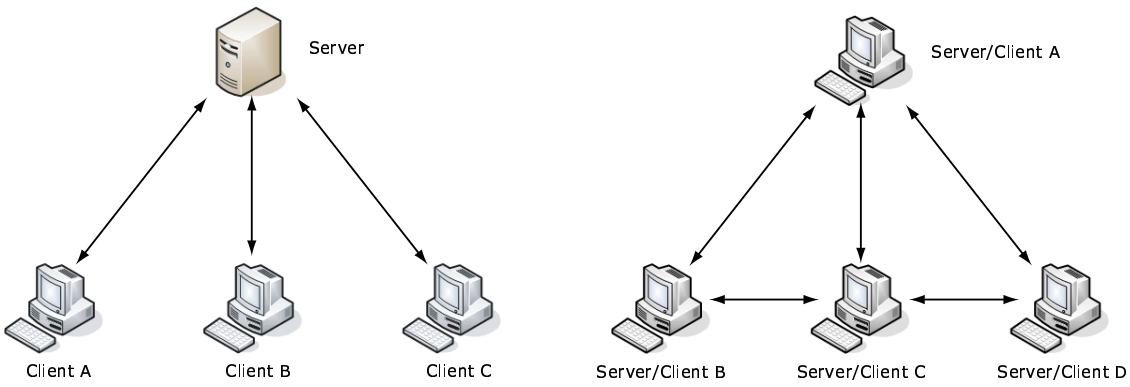
   各端末がコンテンツを提供し，また提供される関係にあるコンテンツ配信アーキテクチャ(図 2.2)．すなわち，各端末はサーバ・クライアント両者の役割を担うことができる．

また，それを構成するエレメントとして以下の 3 つが挙げられる．

1. 端末装置

   PC，携帯電話など各種端末を指す．近年の半導体技術，集積化技術の発展によりこれらの性能は飛躍的に向上し，様々な端末がコンテンツを発信したり，利用することが出来るようになってきている．しかし，各端末の性質および利用の仕方はそれぞれ大きく異なるため，要求されるコンテンツの特性も異なる．

2. サーバ



(a) サーバクライアント型　(b) P2P 型

図 2–1: コンテンツ配信アーキテクチャの基本形

P2P アーキテクチャにおいては，各端末がサーバにもなる．サーバクライアントアーキテクチャにおいては，サーバの容量，性能，コストがコンテンツ配信サービスにおいて大きな影響を与える．特に，大容量コンテンツを配信する場合には，同時アクセス数を十分に考慮入れなければならない．

3. ネットワーク

   コンテンツを配信するネットワークは，バックボーンネットワーク (WAN)，メトロポリタンネットワーク (MAN)，アクセスネットワーク，Local Area Network (LAN)，Home Area Network (HAN) に分類できる．配信されるコンテンツの容量，必要帯域，トラヒック量に合わせてこれらのネットワーク全てが必要なブロードバンド帯域を持たない場合，ボトルネックリンクを生じ，配信性能の低下及び配信不能が引き起こされる．特にユニキャストによって多くのユーザに同時配信を行うと，送信元のサーバから受信アドレスおよびポートだけが異なり，配信されるデータ自体は同一であるパケットが多数送出される．したがって，帯域が無駄に消費され，輻輳が生じやすくなる．Dense Wavelength Division Multiplexing (DWDM) などの技術によりバックボーンネットワーク高速化が計られ，アクセスネットワークにおいては，ADSL，光ファイバが普及し，LAN や HAN においては，ギガビットイーサネットなどの技術が確立されつつある．このように，近年の物理的なネットワーク技術は急速に進展している．したがって，今後，円滑なコンテンツ流通，特に大容量コンテンツを多数の端末へ配信するような場合に，IP 層より上位レイヤでどのようなアプローチを取っていくのかが重要な課題となる．

したがって，円滑なコンテンツ流通実現のための課題として，1) 多様な端末装置への対応手法，2) サーバへの同時アクセス可能数増加手法，3) ボトルネックリンク回避手法，の確立が必要不可欠である．

2.3　各課題に対する解決法

ここでは，上記に挙げた課題を解決するための手法を整理する．

1. 多様な端末装置への対応手法

   どのような端末装置をターゲットにコンテンツ配信を行うかを認識して，システム構築を行う．あるいは各種端末装置に合わせてコンテンツを変換する技術が必要である．

2. サーバ (コンテンツ送信端末) への同時アクセス可能数増加手法

   2 つの手法が考えられる．

   1 つは，ネットワークレイヤより提供される IP マルチキャスト技術を用いる手法である．IP マルチキャストでは，ネットワーク上の中継ノードにおいてデータの複製が行われるため，受信する端末数に限らず，送信されるデータは 1 端末分のデータで済む．したがって，受信端末数が増加しても，サーバへ要求される処理能力は一定であり，サーバへの投資コストを抑えることが可能になる．

   もう 1 つの方法は，コンテンツを 1 カ所のサーバに蓄積する集中型の形態を取るのではなく，多数のサーバにキャッシュする分散型の形態を取ることである．サーバクライアント型アーキテクチャにおいては，CDN が分散型にあたる．IP マルチキャストと CDN を組み合わせて利用することも可能である．

   P2P 型アーキテクチャは，本質的に分散型である．各端末間でコンテンツ配信ツリーを構築し，構築されたツリーに基づくパケットの複製，フォワーディングをルータではなく，端末上で行う ALM も P2P 型の一つの形態と言え，分散型にあたる．

3. ボトルネックリンク回避手法

   2. で挙げた IP マルチキャスト及び CDN，ALM の技術が同様に適用できる．

   IP マルチキャストを適応することで，前述したとおり，ネットワーク上の中継ノードにおいてデータの複製が行われるため，送信元から受信先アドレス，ポートが異なるだけのパケットが多数送出されることがなくなる．これにより，帯域の効率的利用が可能となる．

   CDN を適応する場合は，ユーザを近傍のキャッシュサーバへと誘導することで，ボトルネックリンクとなる可能性が高い国際間リンク，県内網における帯域負荷を大幅に軽減することが可能になる [(3)]．また，ALM においても近接端末間を接続するような，ネットワーク上の端末間距離に基づく配信ツリーを構築することで，これらのリンクを回避することができる．

したがって，円滑なコンテンツ流通を実現するためには，IP マルチキャストの活用あるいは分散キャッシュ型のコンテンツ配信アーキテクチャである CDN，P2P 型アーキテクチャを採用することが必要不可欠である．以下ではこれら 3 つの技術の概要および問題点について整理する．

2.4　IP マルチキャスト

2.4.1　概要

IP マルチキャストでは，ネットワーク中の中継ノードにおいてパケットの複製を行うことで，受信端末数に限らず，送信されるデータは 1 端末分のデータで済む．したがって，送信元は，ユニキャストでコンテンツを配信する際に必要な受信者数に比例した帯域およびデータ送信能力を保持する必要がなく，ネットワークリソースをより効率よく用いることができる．特に，動画像コンテンツなどの大容量コンテンツを多地点に対して配信する場合に非常に有効なソリューションとなる．

IP マルチキャストを構成する代表的な要素技術として，以下の技術が挙げられる．

1. マルチキャストアドレスの割り当て

   マルチキャストアドレスは各マルチキャストセッションを識別するために用いられる．IP Version 4 では，マルチキャストアドレスとして，クラス D (1110 より開始される IP アドレス) が用いられる．マルチキャストアドレスの割り当ては，非常に重要な問題であり，以下のことを実現する手段を提供しなければならない．

   a) アドレス取得法
   b) 割り当てが一意に行われたことを保証する方法
   c) 割り当てられたアドレスがどのような方針 (アドレスの利用期間，アドレスの開始時間，データの配信範囲など) で利用されるのかを確認する方法

   アドレス取得プロトコルとして，Session Announcement Protocol (SAP)[4] が，セッション情報記述プロトコルとして，Session Description Protocol (SDP)[5] などがある．

2. インターネットグループマネージメントプロトコル

   インターネットグループマネージメントプロトコル (IGMP) は，端末群が関心のあるマルチキャストグループに参加する際に用いるシグナリングプロトコルである．IP マルチキャストルータは，端末より IGMP パケットを受信することで，ルータ自身が接続されている LAN 内に，あるマルチキャストグループに関心のある端末が存在することを把握する．そして，そのマルチキャストグループ宛に配信されるデータを，リクエスト元の端末に配信できるようネットワーク上に構築されている当マルチキャストグループに対応した IP マルチキャストツリーに参加する．

   IGMP は，ルータが接続された LAN 内にだけ意味を持つ．IGMP パケットは，ルータによって転送されることはない．

3. マルチキャストツリーの構築手法

   マルチキャストルーティングプロトコルに構築されたツリー上を，データが転送されていく．マルチキャストツリーの種類として，以下の 3 種類に大別できる．

a) Shortest Path Tree

Shortest Path Tree (SPT) とは，送信元からマルチキャストグループの各受信者までを結ぶ各パス上のリンクコストが最小になるように構築されたツリーのことである．リンクコストとして，遅延を用いた場合，SPT は，最小遅延ツリーとなる．SPT は，遅延制約などのツリー制約を課せられたマルチキャストツリーを構築する場合に用いられる．SPT を構築するアルゴリズムとしては，Bellman-Ford 法 [6]，Dijkstra 法 [7] が最も良く知られている．

b) Minimum Spanning Tree

Minimum Spanning Tree (MST) とは，ネットワーク上全てのノードを結ぶ全域木であり，ツリーのコストの総和が最小になるものをいう．MST は，ツリー最適化問題などによく用いられる．MST を構築するアルゴリズムとして，Prim 法 [8]，Kruskal 法 [9] が有名である．また，Prim 法は，ノード間全てのパスコストを取得し，集中的に算出するアルゴリズムであるが，分散型 Prim 法も提案されている [10]．

c) Steiner Tree Steiner Tree とは，あるノード群を結ぶ最小コストのツリーをいう．対象ノード群には含まれないが，これらを最小コストで結ぶために必要なノード群を Steiner point と呼ぶ．ネットワーク上の全てのノードよりなるマルチキャストグループを結ぶ Steiner Tree は，MST と等価である．Steiner Tree を構築する問題は，NP 完全であることが知られている [11]．したがって，いくつかのヒューリスティックな解法が提案されている．代表的なアルゴリズムとして，KMB 法 [12] がある．

図 2–2 を物理トポロジとすると，各ホスト (太線部) をホスト H1 より構築した SPT を図 2–3 に，全ノード (ホスト群 + ルータ群) をつなぐ MST の例を図 2–4 に，各ホストを結ぶ Steiner Tree の例を図 2–5 にそれぞれ示す．太線で描かれたリンクがツリーに対応する部分である．また，Steiner Tree において，点線で囲まれたルータは，Steiner point である．

これらのツリー構築アルゴリズムは，最適化対象になる要素，すなわち各パスのコスト (遅延，帯域など) からなる最適化関数を定義し，課せられた制約を充足しつつ，定義した最適化関数を最大化させるということを行っている．したがって，アプリケーションの要求条件 (遅延優先，帯域優先，QoS 制約など) に応じてツリー構築アルゴリズムが選択される [13]．

実際に，Internet Engineering Task Force (IETF) で提案されているマルチキャストルーティングプロトコルとしては，Distance Vector Multicast Protocol (DVMRP)[14]，Multicast Open Shortest Path First protocol (MOSPF)[15]，Core Based Tree routing protocol (CBT)[16]，Protocol Independent Multicasting-Dense Mode (PIM-DM)[17]，Protocol Independent Multicasting-Sparse Mode (PIM-SM)[18] がある．これらのプロトコルは，配信規模は，小規模か大規模か，SPT かあるいは共有木 (一つのツリー

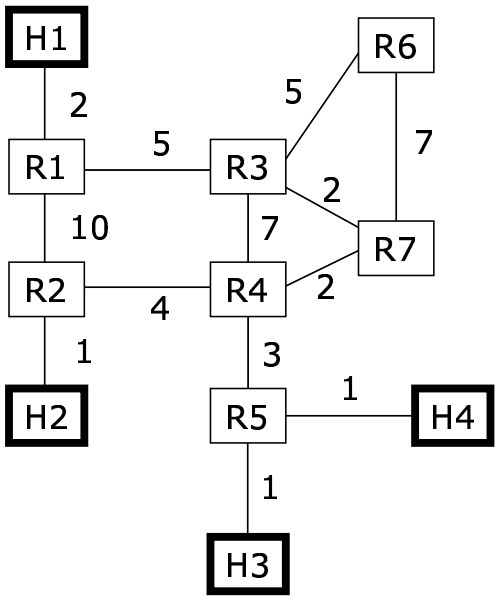


図 2–2: 物理ネットワーク

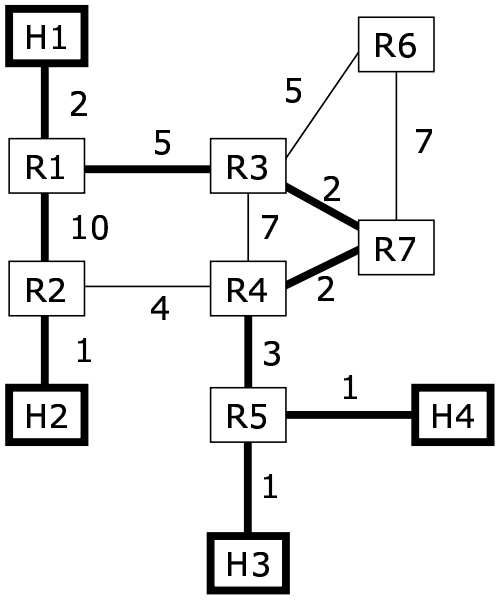


図 2–3: Shortest path tree

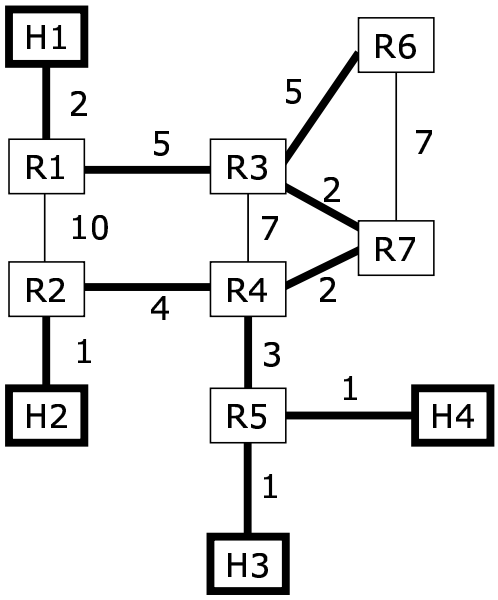


図 2–4: Minimum spanning tree

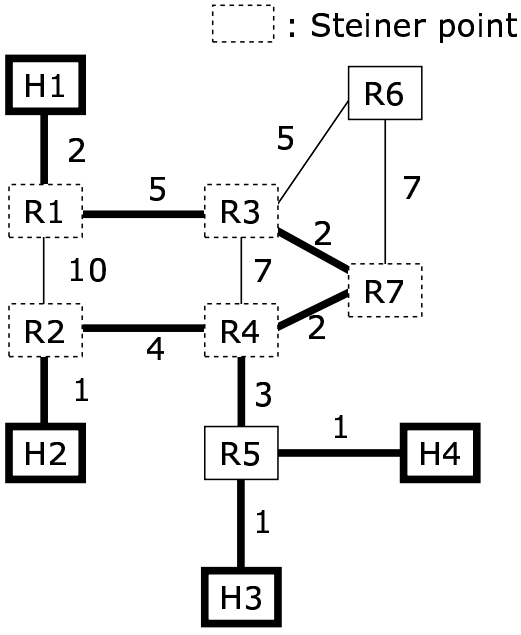


図 2–5: Shortest path tree

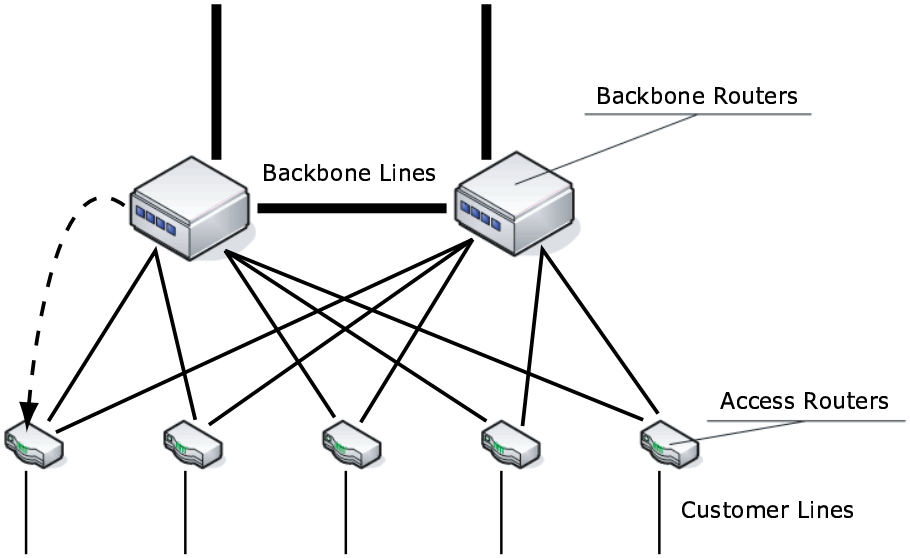


図 2–6: Router Migration Model

を複数のマルチキャストグループで共有する) か，サービスを受ける端末はインターネット上に広く分散しているか否かなど，それぞれの用途に合わせ，選択され用いられる．これらのプロトコルの詳細については，文献 [19] が詳しい．また，QoS 制約付のマルチキャストルーティングプロトコルについては，文献 [13] で詳しく説明されている．

2.4.2　問題点

IP マルチキャストに関して，様々な実験が実験ネットワークである Mbone で行われはじめ十数年経過したにもかかわらず，未だ普及の兆しがあまり見られていない．その理由として，現在の IP マルチキャストのサービスモデルが，商用サービスで使われることを意識していないことが指摘されている [20]．具体的には，1) マルチキャストグループ作成にかかわる認証，受信者認証，送信者認証などのグループ管理問題，2) マルチキャストアドレスの割り当て及び配布問題，3) マルチキャストルータへの攻撃，データの一貫性の保証などのセキュリティ問題，4) 伝送中に生じたマルチキャストツリーへの問題解決，モニタリングなどのネットワーク管理問題，が挙げられる．特にこれらの問題は，インタードメインで IP マルチキャストを実現しようとした場合により困難な問題となる．

また，上記課題以外にも，各 ISP のルータと LAN スイッチを IP マルチキャスト対応にしなければならないという問題がある．この問題を解決するためには，“router migration model” と呼ばれる ISPs が従っているモデルを打破する必要があることが指摘されている．router migration model とは，新しく購入されたルータはまずバックボーンネットワークに配置され，徐々にカスタマーアクセスポイントの方へと置き換えられていくモデルのことを指す (図 2–6).

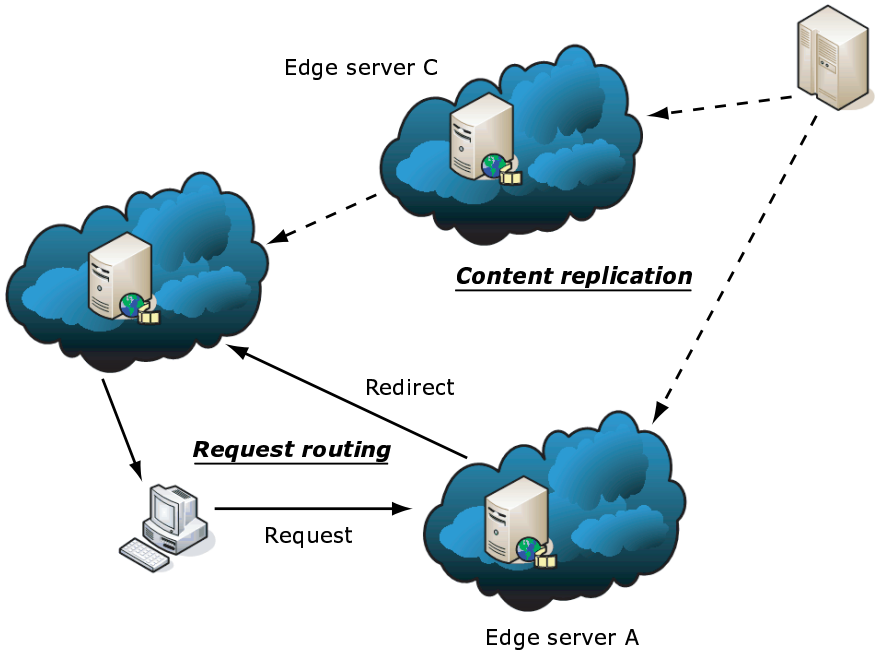


図 2–7: Request routing と Content replication

バックボーンネットワークに要求されるようなルータは，より広帯域を柔軟に扱えることを期待されるため，アンインテリジェントでシンプルなルータが好まれる．したがって，IP マルチキャスト対応ルータの配置が敬遠される．このように IP マルチキャストの普及には技術的課題以外の課題も残されている．

2.5　分散キャッシュ型コンテンツ配信アーキテクチャ

分散キャッシュ型コンテンツ配信アーキテクチャとして，CDN 及び P2P 型アーキテクチャについて言及する．これら 2 つの技術は組み合わせる研究もなされている [21]．

2.5.1　CDN

CDN に要求される機能は IETF の Content Distribution Interworking (CDI) グループにおいて議論されている．CDN のコアとなる機能としては，

1. Request routing

   最適なサーバへとリクエストを導く機能．

2. Content replication

   コンテンツをエッジサーバ (サロゲートサーバとも呼ばれる) に戦略的に配置する機能．

が挙げられる (図 2–7)．

その他の機能として，3) CDN サービスを行う業者間での相互運用に関する CDN Peering, 4) コンテンツの利用を測定，記録，管理する Accounting，5) 利用者の好み，端末の性能に合わせてコンテンツを調整する Contents Adaptation の機能が挙げられる．Contents Adaptation が，円滑なコンテンツ流通に必要不可欠であることは前述した通りである．

ここでは，Request routing 及び Content replication に必要な課題を整理する．

1. Request routing

   Request routing に必要な機能は，1) 最適なサーバを選択する機能，2) 最適なサーバへと誘導する機能，である．

   最適なサーバを選択する機能については，本論文の 1 つのテーマであり，従来研究についても提案手法と共に 3 章で述べる．

   最適なサーバへと誘導する機能としては，単に Domain Name System (DNS) 毎に誘導するサーバを変更するだけの方法や DNS を基にした上記のサーバ選択機能を組み込んだよりインテリジェントなシステムなども開発，提案されている [22]．

2. Content replication

   [23] では，各ノードを有限の記憶容量を有した Autonomous System (AS) と仮定し，クライアントが要求したコンテンツを取得するまでに通過する AS の数が最小となる場合を最適解とすることに基づく方法を 4 つ提案し，比較，検討を行っている．

   このような Content replication のアプローチは，以下に示すようなパラメータを組み合わせたコスト関数を構築し，その値が最大値あるいは最小値になるような組み合わせを見つける，いわゆる組み合わせ最適化問題に帰結することが出来る [24]．使用されるパラメータとして，

   a) リード率 ($reads_{ik} \geq 0$)
      あるクライアント i からコンテンツ k に対して 1 タイムユニットあたり行われたリードアクセス回数．

   b) 距離 ($dist_{ij} \geq 0$)
      クライアント i とサーバ j との距離．距離のメトリックとしては，遅延，ルータホップ数，トータルリンクコストが用いられる．

   c) オブジェクトサイズ ($size_k > 0$)
      コンテンツ k のサイズ．

   d) アクセスタイム ($acctime_{jk} \geq 0$)
      コンテンツ k がノード j によって最後にアクセスされた時間．

   e) 配置マトリックス ($x_{jk} \in \{0, 1\}$)
      ノード j がコンテンツ k を保持しているかどうかを表す行列．この行列は，始めは未定であり，配置の結果が保存される．

f) ルーティングマトリックス ($y_{ijk} \in \{0,1\}$)

クライアント i がコンテンツ k を取得するためノード j にリクエストを送るかどうかを表す行列．この行列は，始めは未定であり，最適化の結果が保存される．

などが用いられる．多くの方法では，次の 2 つの制約がコスト関数に適応される．

a) コピーされるコンテンツの数

$\sum_{j \in N} x_{jk} \leq P, \forall k$ を満たす，N が全ノード，P がコピーされるコンテンツ数の上限を表す．

b) 記憶容量

$\sum_{k \in K} size_k \cdot x_{jk} \leq S_j, \forall j$ を満たす，K が記憶するコンテンツ，S_j がノード j の記憶容量を表す．

上記パラメータとして距離が挙げられているが，これは Request routing の最適なサーバを選択する機能とも密接な関係がある．したがって，本論文で提案する手法を利用した Content replication アルゴズムを構築することも可能である．

[24] ではまた，今後の Content replication アルゴリズムの課題として，1) パフォーマンス，信頼性，可用性などを表す項もコスト関数に導入できるのではないかということ，2) サーバがコンテンツをコピーし始めて何秒後以降ならそのデータを他のノードから参照できるかを考慮すること，が挙げられている．

また，[25] では，3) 現在の CDN 用のほとんどのアルゴリズムは 10^4 以上のノードを持つシステムには対応できないためより分散的なアルゴリズムが求められること，4) Update 時間のデッドラインやクライアント側における最大遅延の QoS プロパティを保証できるアルゴリズムが必要であること，が指摘されている．

CDN の構築の仕方としては，

1. オーバレイアプローチ

   CDN 用にネットワークインフラストラクチャを構築しないアプローチ.

2. ネットワークアプローチ

   ネットワークインフラストラクチャを CDN 用に特化させるアプローチ.

がある [26].

ネットワークを CDN 用に構成するネットワークアプローチの方が，オーバレイアプローチに比べ安定したサービスが期待できるが，初期コストがかかる．一方で，インターネットのような既存のネットワークを用いるオーバレイアプローチは，ある程度コストを抑えることが出来る．[26] で指摘されているとおり，CDN を構築するための初期投資コストに関する問題は非常に困難な問題である．したがって，どちらのアプローチで，どのようなビジネスモデルを打ち出していくかは，CDN 事業者にとって非常に難しい問題である．

2.5.2　P2P

P2P 型アーキテクチャの例として，flash crowd と呼ばれる短時間の間にリクエストが集中する現象に耐えうるシステムとして提案された CoopNet[27]，PROOFS[28] が挙げられる．この例から分かるように P2P 型アーキテクチャの最大の利点は，コストパフォーマンス及び耐障害性にある．特に，近年，ALM と呼ばれる各端末をマルチキャストノードとし，マルチキャスト配信を行う手法が盛んに研究されているが，これらは P2P 型アーキテクチャの一種である．本論文でも，多数の端末が参加した場合においても安定したサービスが提供できる ALM ツリー構築法について検討，提案を行っている．ALM については，後述する．

P2P 型の欠点としては，優れたコンテンツ供給端末が必要である点が挙げられる．一般に P2P を形成している端末のコンテンツ保有数，提供帯域はサーバクライアント型アーキテクチャにおけるサーバと比較し，非常に劣る．そのため，多数の端末が集まりサービスを形成する．しかし，配信されるべきコンテンツを持っている端末数が少なければ，サービス形成に多くの時間を要する．したがって，ある程度のコンテンツ保有数，広い提供帯域を持った端末が必要となる．

そこで，CDN のコストの問題，P2P 型アーキテクチャの優れたコンテンツ供給端末が必要である点を解決するため，これら 2 つの技術を組み合わせたハイブリッド方式も提案されている [21]．この方式では，始めコンテンツは CDN を用いて，配信される．したがって，サーバクライアント型アーキテクチャであり，クライアント数が増えるにつれて，サーバの強化が要求される．しかし，ある点を過ぎた時点で，CDN によって既に配信されているクライアント間で P2P ネットワークを構成し，新しく要求を出した端末に対しては，P2P 側でサービスを提供する．最終的には，CDN でのコンテンツ配信は停止され，P2P ネットワークだけで配信される．

P2P 型アーキテクチャにおいて必要不可欠な技術として，

1. 最適ホスト選択

   複数の端末群が目的のコンテンツを保持している場合，それら端末群のなかから，端末の送信能力，自端末からのネットワーク距離などに応じて，最適な端末を選択する機能．CDN の項における最適サーバ選択にあたる．

2. コンテンツ発見

   クライアント端末の要求コンテンツを保持している端末群を発見する機能．P2P 型ネットワークは，十分なコンテンツ送信能力，帯域を保持したサーバが存在するわけではなく，それらと比べ能力的に劣る各端末により構成される．また，これらの各端末は，セッションに常時参加しているわけではなく，頻繁に参加，離脱処理を繰り返す．したがって，リクエスト元の端末が，安定，効率的に現在セッションに参加している端末群のなかから要求先端末を選択する必要が生じる．CDN のように常時セッションに参加している端末群が存在する場合においても重要な機能であるが，上記の理由から P2P 型アーキテクチャではより重要な機能であるといえる．

最適ホスト選択に付いては，3 章で述べる．ここでは，コンテンツ発見手法について言及する．

コンテンツ発見

1. コンテンツ特定フェーズ

   あらゆるコンテンツは，それ自身を一意に特定する Contents ID を持つ．このフェーズでは，ユーザが希望するコンテンツの Contents ID を特定する．Contents ID の特定には，コンテンツ情報を表すメタデータを操作することにより行われる．マルチメディアデータの属性や構造を記述するメタデータとして，MPEG-7 が標準化されている．

2. ホスト特定フェーズ

   Contents ID からそのコンテンツを所有するホストを特定する．Uniform Resource Locator (URL) により一意に識別される現在の World Wide Web (WWW) 上のコンテンツは，DNS によりこのフェーズが実現されている．しかし，このような DNS 型のコンテンツ管理の方法が現在より遥かに膨大なコンテンツ量の流通を実現する場合においても有効であるかどうかは疑問の余地がある．また，このホスト特定フェーズの技術と，前章で検討した最適ホスト選択手法をどのように結合させるかということも今後の課題である．

ここでは，ホスト特定フェーズの技術について近年注目を浴びている分散ハッシュテーブル (Distributed Hash Table: DHT) を用いる方法を挙げる．

DHT では，すべてのコンテンツ及びノードにそれぞれユニークな Contents ID と Node ID が付与されていることを仮定している．DHT では，Node ID に基づいてオーバレイネットワークが構築され，コンテンツ自身または位置情報などのそのコンテンツに関する情報は，Contents ID と Node ID が一致するノードにより管理される．コンテンツの発見に必要なクエリーのオーバレイネットワーク上でのホップ数は，オーバレイネットワークに参加しているノード数を N とした場合，各ノードが $O(logN)$ の隣接ノードを管理しているとき，最大 $O(logN)$ であることが証明されている．

Tapestry[29] を例に DHT の動作例を示す．

Node ID 7734 のノードは，Node ID が下桁から順に i 桁 ($0 \leq i \leq 3$) だけ一致するノードを隣接ノードとして論理リンクを張る．例えば，Node ID 7734 では，51E5，9C34，2A34，A734 などが隣接ノードとなり得る．

Node ID 39AA のノードが Contents ID 8734 のコンテンツを登録するために，Contents ID = 8734, Node ID = 39AA であるという情報を含んだクエリーを，Contents ID 8734 の管理ノードである Node ID 8734 のノードに向けて，隣接ノードである D6A4 に向けて送信したと仮定する．このクエリーは，Contents ID 8734 と一致する下桁が 1 つずつ増加するような Node ID を持つノードに転送されていく．例えば，D6A4，1634，A734，7734 とい

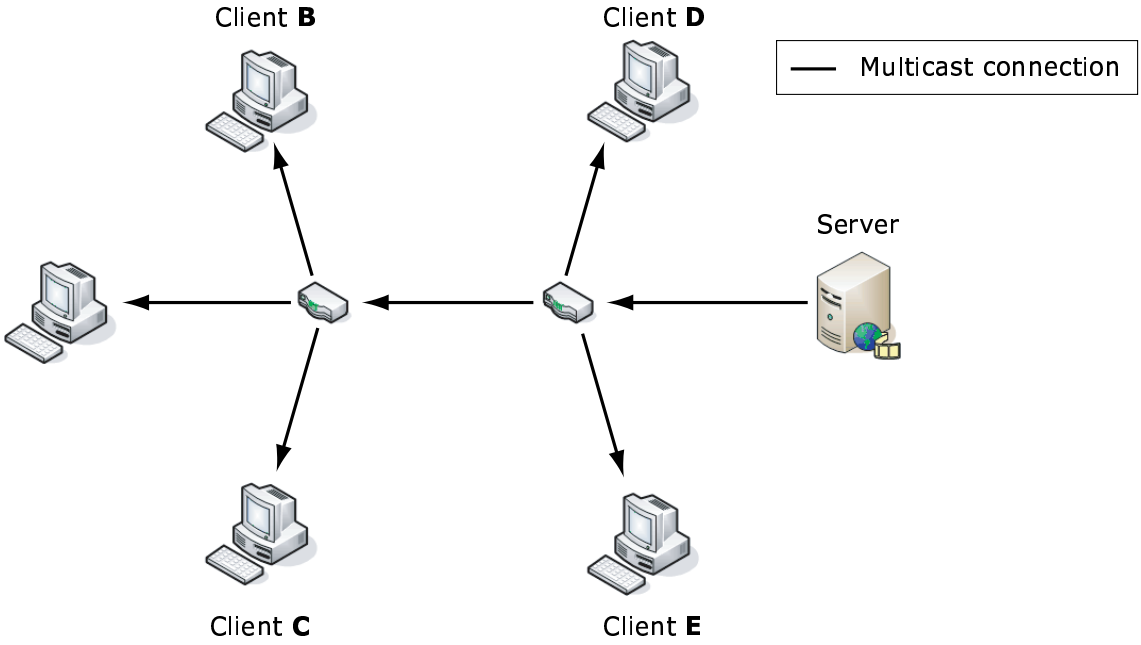


図 2–8: IP マルチキャスト

う様に転送されていく．そして最終的に Node ID 8734 を持つノードにこのクエリーが転送される．

同じようにして，コンテンツのリクエストクエリーなども送信される．

Pastry[30], Chord[31] などの DHT も，オーバレイネットワークの構築手法が Tapestry とは異なるが，基本的な考え方は同じである．

ALM

ここでは，本論文のテーマでもある ALM について簡単に説明する．より詳しい ALM 各種プロトコルについての説明は，4 章で述べる．

ALM は，マルチキャストサポートをネットワーク内の中継ノード，すなわちルータから各エンドホスト端末へと移行したものといえる (図 2–8，2–9)．

各端末がマルチキャスト機能を提供するため，IP マルチキャストで要求されるネットワーク内全ルータのマルチキャスト対応化という制約がない．したがって，現在のネットワークストラクチャを用いて，直ぐにマルチキャスト機能を提供することが可能である．また，ALM をミドルウェアとして提供することで，各アプリケーションを即時にマルチキャスト対応にすることが可能となる [32]．

IP マルチキャストと ALM の違いをまとめると，以下のとおりになる．

1. パケットの複製，フォワーディング機能を IP マルチキャストでは，ネットワーク内に配置されたルータが担当したが，ALM では，各端末が担う．

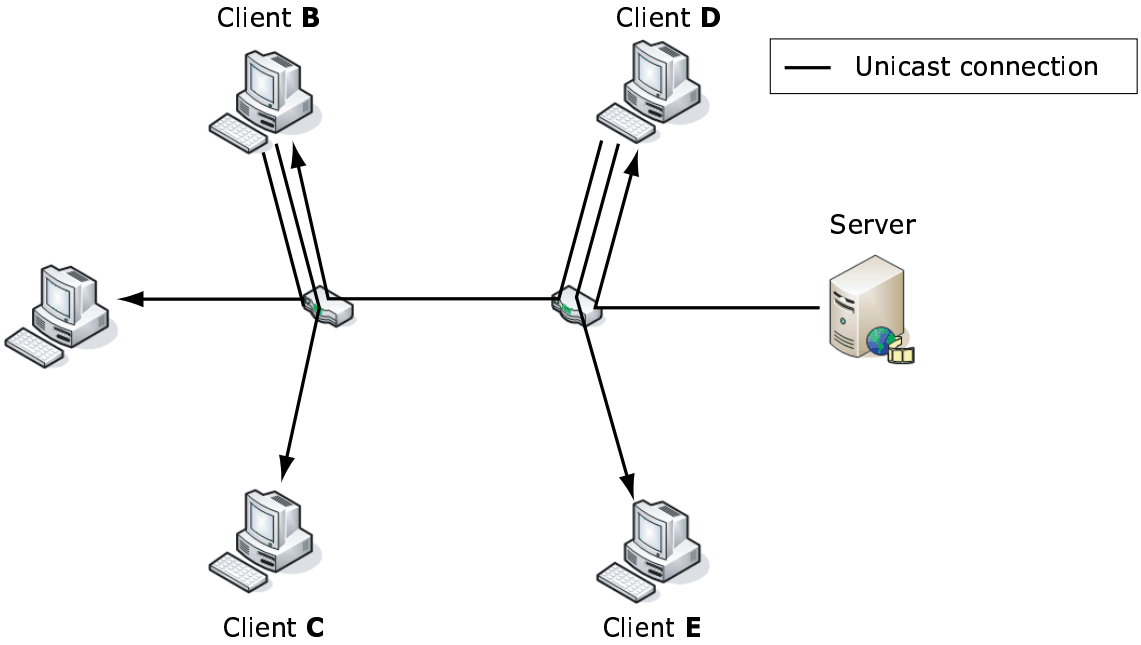


図 2–9: ALM

2. ALM では，物理的なネットワークトポロジは完全に隠される．全端末間で仮想的な有向グラフが構築可能である．これらのグラフは，端末間で測定したネットワークメトリックに基づいて構築され，最適化される．

3. IP マルチキャストでは，メンバーシップ情報は各マルチキャストルータ間で配布された．ALM では，例えば，ランデブーポイント (Rendezvous Point: RP) と呼ばれる全端末に既知となる端末を用意し，RP に対するリクエストのリプライとして提供する方法や，コンテンツ送信元に直接リクエストを送信する方法など様々な方法がある．ただし，これらの情報も ALM では，もちろん各端末が管理する．

4. ALM で構築されるツリーは，基本的にコントロール可能である．例えば，RP ですべての端末，リンク情報を取得し，中央集権的にツリーを構築することも可能である．

IP マルチキャスト，ALM は互いに相反する技術ではなく，IP マルチキャストと ALM を同時に用いることでよりスケーラビリティの高いコンテンツ配信アーキテクチャを構築する例もある [33]．なぜなら，IP マルチキャストは，特にインタードメイン間でサービスが提供することが困難であることを前述したが，インタードメイン間を結ぶトンネリングプロトコルとして，ALM を用いることが可能だからである (図 2–10)．

ALM の欠点として，以下のことが挙げられる．

1. ALM では，エンドホスト端末同士がデータをやりとりするため，あるネットワークリンクを同一データパケットが何度も通過する可能性がある．したがって，ネットワークリソースの有効利用という観点では，IP マルチキャストに劣る．

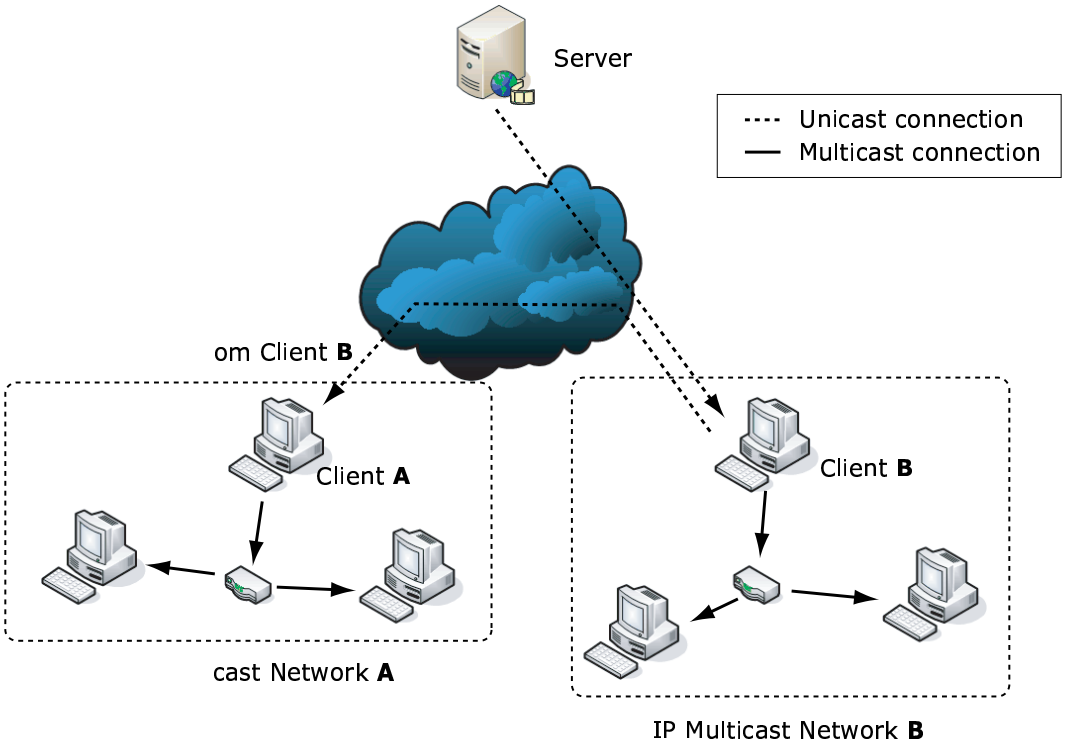


図 2–10: トンネリングとしての ALM プロトコル

2. ALM では，各端末がマルチキャスト機能の中核を担うことになるが，一般的に端末は，ルータと比較した場合，サービス提供継続性などの観点で不安定である．なぜなら，ルータは，長期間停止せずにサービスを提供できるように設計された機器であるが，端末ではユーザの恣意的なセッション離脱や導入されている OS の状態によりマルチキャストサービスの停止が頻繁に引き起こされる可能性があるからである．

したがって，ALM では，ツリーを構成するノードの頻繁な状態変化 (参加，離脱) に耐えうるツリー構築アルゴリズムおよびプロトコルが必要不可欠である．また，IP マルチキャストと比較し，どの程度の性能劣化があるのかを認識するための評価尺度を導入し，ALM プロトコルの性能をきちんと測定することが重要となる．

2.6　本章のまとめ

本章では，基本的なコンテンツ配信アーキテクチャについて論じ，大規模コンテンツ配信の実現には，1) 多様な端末装置への対応手法，2) サーバ (コンテンツ送信端末) への同時アクセス可能数増加手法，3) ボトルネックリンク回避手法，が重要であることを指摘した．そして，それら手法を実現可能な技術として，IP マルチキャスト，CDN，P2P(特に，ALM) を挙げ，それらの特徴および問題点を明らかにした．

第3章

最適ホスト選択手法

3.1 はじめに

前章において，分散キャッシュ型コンテンツ配信アーキテクチャでは，1) サービス利用者の要求コンテンツを保持しているホスト (CDN では，エッジサーバにあたる) の発見手法 (コンテンツ発見), 2) 要求されたコンテンツを複数のホストが保持していた際の最適ホスト選択手法 (最適ホスト選択) が重要であるを述べた．これらの手法を基に，CDN における Request routing 及び Contents Replication を実現できる．また，これらの技術は，P2P 型アーキテクチャにおいても重要であることを述べた．そこで，本章では，最適ホスト選択に着目し，従来研究に言及した上で，インターネット上での各ホストの位置を幾何学空間に射影する方法を基にした最適ホスト選択手法を提案する．また，提案手法の性能評価をシミュレーションにより行う．

3.2 最適ホスト選択に用いられるメトリック

最適ホスト選択において最も重要なメトリックとなるのが，平均スループットである．ただし，平均スループットを推定するのは，大量のネットワークリソースを消費するのど，現実的ではない．そこで，

1. IP path length

   パケットが送信元から受信元に届くまでに通過するホップ数.

2. AS path length

   パケットが送信元から受信元に届くまでに通過する AS の数.

3. 位置情報

   ホスト間の地理的距離など.

4. Round Trip Time (RTT)

などが代替メトリックとして使用される.

3.3　従来研究

ここでは，最適ホスト選択に用いるメトリックに関する研究および最適ホスト選択に関する従来手法について整理する．また，それら従来手法の問題点について指摘する．

3.3.1　最適ホスト選択に用いるメトリックに関する研究

文献 [37] では，IP path length, AS path length, RTT がどの程度相関性があり，どのメトリックが，最適ホスト選択メトリックとして最も有効であるのかを考察している．IP path length と AS path length との比較では，この 2 つのメトリックの相関性は最大で 70%程度であり，最適ホスト選択メトリックとしては，IP path length の方が優れていると指摘している．その理由として，AS path length は，あくまで Inter-AS topology を反映しているに過ぎず，各 AS を通過するのに要する時間を隠している点，ルータから BGP テーブルが取得できない場合，測定が困難である点が挙げられている．また，IP path length と RTT との比較では，相関性が 50%程度であり，1995 年当時 0%であった状況とは違うことを指摘している．この理由として，インターネットインフラストラクチャの改善が挙げられている．最後に，最適ホスト選択メトリックとしては，RTT が最も優れていることが実験により示されている．

また，文献 [38] では，前述した 4 つのメトリックについて比較，検討している．結果として，RTT が最も優れていること，過去の RTT の情報をどのように用いればより予測が向上するかが示されている．

したがって，本論文では最適ホスト選択のメトリックとして，RTT を採用する．しかし，RTT を測定するためには，各ホスト間でプローブパケットを送信する必要がある．これにより，大量のプローブパケットが発生し，トラヒックを消費してしまう．また，測定にもかなりの時間を要する．そこで，少量のプローブパケットでホスト間の RTT を予測する手法が必要となる．

3.3.2　最適ホスト選択に関する研究

文献 [34] では，最適ホスト選択として，次の 2 つの方法を提案している．第 1 の方法が，traceroute と呼ばれるコマンドを使って，ユーザからサーバ選択装置，サーバ選択装置から各サーバへの経路情報を取得し，ユーザとサーバ間の距離を見積もる方法である．すなわち，この方法は，メトリックとして IP path length を使っている．第 2 の方法が，ホストの IP アドレスを用いる方法である．あるホストの IP アドレスを $x.\,y.\,z.\,w(\ 0 < x,\ y,\ z,\ w < 256\ )$，ネットマスクを $m1.\ m2.\ m3.\ m4(\ 0 \leq m1,\ m2,\ m3,\ m4 < 256\ )$ とすると，次式 (3·1) よりホストの位置を相対化した数値 α を求める．

$$\alpha = (x\ \&\ m1) \cdot 256^3 + (y\ \&\ m2) \cdot 256^2 + (z\ \&\ m3) \cdot 256 + (w\ \&\ m4) \quad (3{\cdot}1)$$

& は，論理和を表す．

ホスト選択の際には，次式 (3·2) の値 β が最も小さくなるホストを最適ホストとして選択する．

$$\beta = |\text{ユーザの位置を相対化した数値} - \text{各サーバの位置を相対化した数値}| \quad (3\cdot2)$$

この手法は，各ホストへの IP アドレスの割り当てが，経路情報の数を減らすための経路の集約手法に基づいて割り振られていること，すなわち，β の値が小さい場合は，同じネットワークに属している可能性が高いということに基づいている．この手法は，上記 4 つの内の手法の中では，位置情報メトリックに近いものである．

また，文献 [35] では，network-aware clustering と呼ばれる BGP テーブルより取得した AS 情報でホストをクラスタリングする手法 [36] を用いた P2P システムアーキテクチャを提案している．これは，最適ホスト選択のメトリックとして AS path length を用いている手法と言える．

3.3.3　最適ホスト選択に関する従来研究の問題点

文献 [34] で提案された方法のうち，traceroute を用い，ユーザ－サーバ間の距離を見積もる方法についてだが，セキュリティの関係上，traceroute を許可していないネットワークも存在する．したがって，もしそのようなネットワークが，ユーザ－サーバ間に存在する場合，traceroute が動作せず，距離推定が困難になる．また，traceroute で取得できる情報は，IP path length であり，前述したとおり，RTT に基づく手法がより最適ホスト選択には向いていると考えられる．

また，ホストの IP アドレスを用いる方法に関しては，IP アドレスがきちんと経路集約を考慮し，割り当てられている場合は，有効な解決法である．しかし，実際，現在の IP Version 4 のネットワークにおいては，そのような経路集約が完全になされていないのが現状であり，例え両者の IP アドレスが似ていても，近接ネットワークに所属しているとは限らない．

文献 [35] の方法は，BGP テーブルをきちんと取得できる場合に関しては良いが，一般的にそれらは非公開であり，取得不能の場合が多い．

3.4　提案手法

ここでは，従来研究に基づき，最適ホスト選択に用いるメトリックとして，RTT を採用する．RTT の測定の仕方としては，ICMP パケットを用いた ping コマンドで測定する方法や各端末のアプリケーション層間で測定する方法がある．セキュリティの問題で，ICMP パケットが許可されない場合もあるが，そのような場合でも端末間のアプリケーション層間で RTT を測定することができる．つまり，端末間の RTT は，いかなる場合でも測定することが可能である．

しかし，各ホスト間の RTT を測定するためには，RTT 測定用の大量のプローブパケットをやり取りする必要がある．これにより，特にセッションが大規模である場合は，これらのプローブパケットによる帯域の圧迫が生じる．そこで，ここでは，少量のプローブパケットのみを用いることで，ホスト間の RTT を予測する RTT 予測手法を応用した最適ホスト選択手法を提案する．

3.4.1　RTT 予測手法

RTT 予測手法として，Global Network Positioning (GNP) と呼ばれる手法が提案されている [39]．この手法では，インターネットを幾何学空間と仮定し，各ホストの位置をその空間上の座標としてマッピングする．この座標を用いて，各ホスト間の RTT を予測する．具体的な手順を次に示す．

1. Landmark Operations

   インターネットをある幾何学空間 S にマッピングするために，Landmark と呼ばれる幾つかの基点となるホストを用意する．Landmark をどのように選択するかは今後の課題とされている．ここでは，N 台の Landmark，$L_1, L_2, \cdots, L_N$ を仮定する．また，ホスト H の幾何学空間 S での座標を c_H^S とし，ホスト H_1, H_2 間の S 空間上での距離を $\hat{d}^S_{H_1H_2} = f^S(c^S_{H_1}, c^S_{H_2})$ とする．

   各 Landmark 間で RTT を測定し，$N \times N$ 距離行列の下半分を生成し，この距離行列を対称行列とする．そして，以下の式 (3·3) で示される $f_{obj1}(\cdot)$ を最小にするように $c^S_{L_1}, \cdots, c^S_{L_N}$ を決定する．

$$f_{obj1}(c^S_{L_1}, \cdots, c^S_{L_N}) = \sum_{L_i, L_j \in \{L_1, \cdots, L_N\} \mid i>j} \varepsilon(d_{L_iL_j}, \hat{d}^S_{L_iL_j}) \tag{3·3}$$

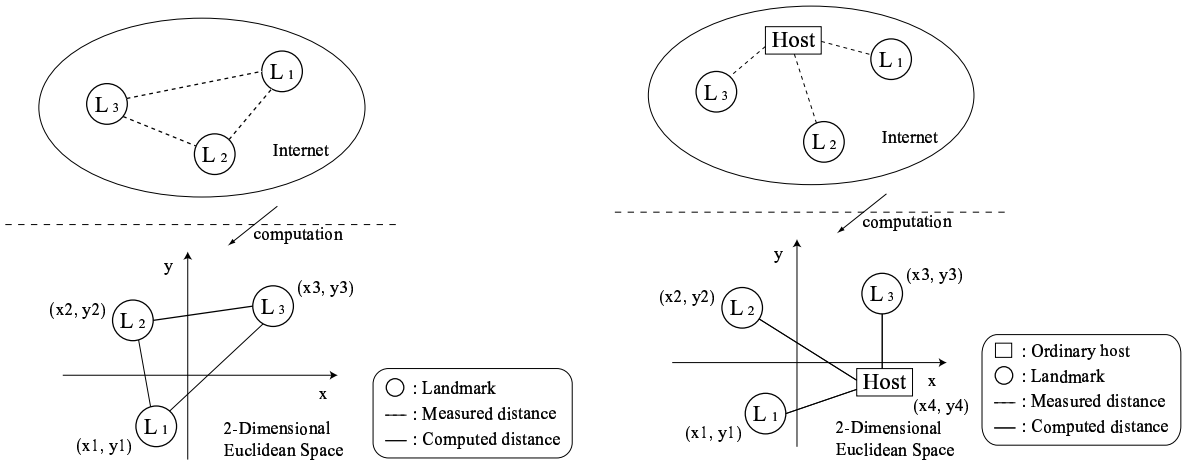
   ε は，エラー関数であり，この論文では，式 (3·4) としている．

$$\varepsilon(d_{H_1H_2}, \hat{d}^S_{H_1H_2}) = (\frac{d_{H_1H_2} - \hat{d}^S_{H_1H_2}}{d_{H_1H_2}})^2 \tag{3·4}$$

   $c^S_{L_1}, \cdots, c^S_{L_N}$ が決定したら，その情報を幾何学空間 S の識別情報と距離関数 $f^S(\cdot)$ と共に GNP に参加する全てのホストに配布する．この論文では，その情報の配布方法，プロトコルについては言及していない．Landmark が 3 台ある場合の操作を図 3–1(a) に示す．

2. Ordinary Host Opertaions

   ホスト H は幾何学空間 S 上での座標 c_H^S を得るために，N 台の Landmark に対して，プローブパケットを送信し，RTT を測定する．そして，式 (3·5) を最小にするように c_H^S を決定する．



(a) Landmark operations　(b) Ordinary host operations

図 3–1: Landmark operations と Ordinary host operations

$$f_{obj2}(c_H^S) \;=\; \sum_{L_i \in \{L_1, \cdots, L_N\}} \varepsilon(d_{L_i H}, \hat{d}^S_{L_i H}) \tag{3·5}$$

ε は，式 (3·4) で決定したエラー関数である．

このように，幾何学空間 S における各 Landmark の座標を決定すること，各ホストの座標を決定することは多次元データ最小化問題 (multi-dimensional global minimization problem) に帰結できる．ホストにおける座標決定プロセスを図 3–1(b) に示す．

各ホスト間における RTT は，幾何学空間 S 上での距離により表される．したがって，各ホストに対して，幾何学空間 S における自ホストおよび測定対象ホストの座標，距離関数 $f^S(\cdot)$ を用いることで，これらホスト間の RTT を推測することが可能となる．

手法の有効性は，シミュレーションにより示されている．しかし，次の 2 点の理由よりこの手法をそのまま最適ホスト選択に用いた場合，スケーラビリティに欠ける点が存在する．それは，

1. 幾何学空間上における各ホストの座標を計算するため，中央サーバが必要となる．
2. 座標決定のための多大な計算コストが必要となる．
3. Landmark がシステムダウンした場合，サービス継続ができない．

である．これらの問題の原因は，正確な RTT 予測の実現に起因する．すなわち，ホスト間の正確な RTT 予測を実現するためには，座標空間における距離関数を厳密に決定する必要があり，そのためには上記要因が密接に関連している．

しかし，最適ホスト選択で重要なのは，正確な RTT 予測のような絶対的な予測ではなく，どのホストが RTT の面で最適であるかといった相対的な予測手法である．

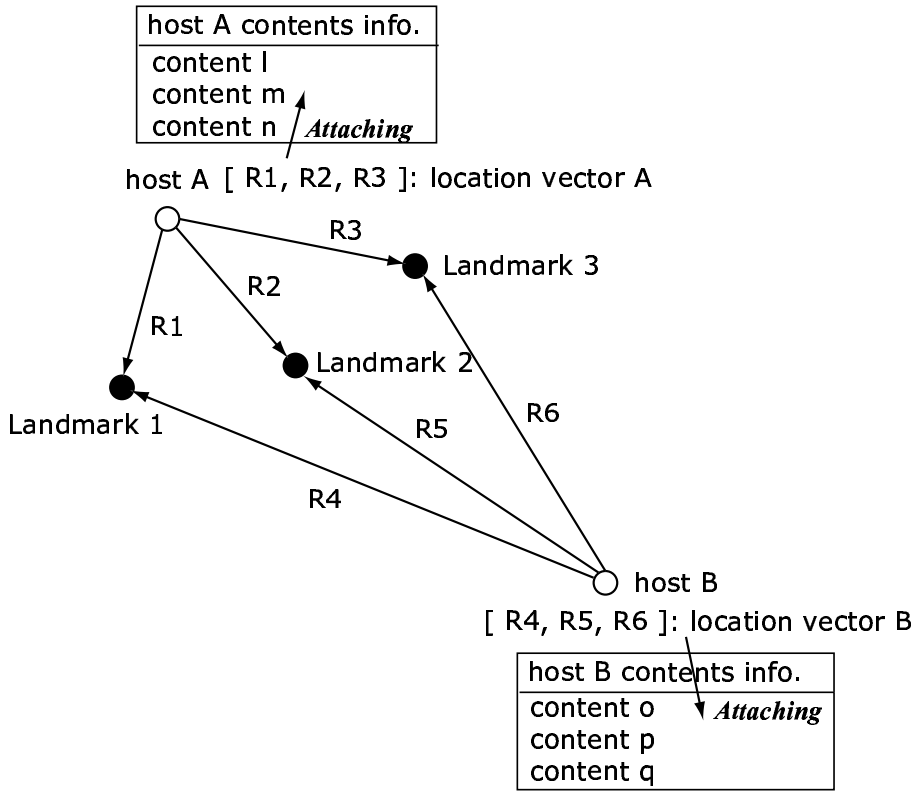


図 3–2: 位置ベクトルの取得とコンテンツ情報への付加

そこで，GNP を応用した相対的な RTT 予測に基づく最適ホスト選択手法について提案を行う．

3.5　提案手法概要

提案手法の手順を以下に示す．

1. 各ホストは，数台の Landmark と呼ばれる特定のホストに対して，プローブパケットを送信する．
2. 各ホストは，取得した RTT を要素とした位置ベクトルを決定する．
3. 各ホストは，自分の管理するコンテンツ情報の中に 2. で得た位置ベクトルを付加する (図 3–2)．
4. 目的のコンテンツを複数のホストが保持していることが分かった場合，要求ホストは自分の位置ベクトルとコンテンツを保持しているホストの位置ベクトルを比較し，最適ホストを決定する．

4. での位置ベクトル比較においては式 (3·6) より導出される α が 0 に最も近いホストを最適ホストとする．A, B は，それぞれ比較する各ホストの位置ベクトルである．

$$\alpha = \arccos\left(\frac{AB}{|A||B|}\right) \tag{3·6}$$

式 (3·6) は，位置ベクトル A および B 間の相関係数を表す．したがって，α が 0 に近いほど，両者の位置ベクトルは似たベクトルを持つこととなる．すなわち，両者のホストは，近い位置に存在しているということを表している．したがって，α が 0 に最も近いホストが最近傍ホスト，すなわち最適ホストであると考えられる．

この手法では，要求したコンテンツを保持している目的ホスト群との間でそれらの位置ベクトルを取得し，相関係数を算出するため，GNP で挙げた問題点である 1) 幾何学空間上における各ホストの座標を計算するために中央サーバが必要となる，2) 座標決定のための多大な計算コストが必要となる，点が解決される．

また，Landmark がシステムダウンし，完全な位置ベクトルが取得できない場合でも，位置ベクトルにおける欠落部を 2 つのホスト間で共通して除外することで，相関係数の算出が可能である．次に例を挙げる．あるホスト A，B を仮定し．ホスト A の位置ベクトルを $[Ra_1, Ra_2, Ra_3, Ra_4, Ra_5, Ra_6]$，ホスト B の位置ベクトルを $[Rb_1, Rb_2, Rb_3, Rb_4, Rb_5, Rb_6]$ とする．この時，Landmark 4 がシステムダウンし，ホスト B の位置ベクトルの 1 要素である Rb_4 が取得できなかったとする．しかし，この場合においても，ホスト A － B 間の相関係数算出の際に，ホスト A の位置ベクトルから Ra_4 を除外することで，相関係数の算出は可能である．したがって，GNP で挙げた問題点 3) Landmark がシステムダウンした場合，サービス継続ができない点についても解決される．ただし，多数の Landmark がシステムダウンした場合は，多数の位置ベクトルを削除することとなり，最適ホスト選択の精度に影響を与えるものと考えられる．

3.6　提案手法性能評価実験

提案手法に関して，以下の実験を行い，提案手法の性能評価を行った．

3.6.1　評価項目

提案手法性能評価実験として，以下の 2 点について，1) ランダムにホストを最適ホストと選択した場合，2) すべてのホストに対して，プロービングパケットを送信し，最適ホストを選択した場合，3) 提案手法を用いて最適ホストを選択した場合，についてネットワークシミュレータである ns-2[(40)] を用いて，シミュレーションを行った．

Download time あるホストが決められたコンテンツ群をすべてダウンロードするのに要する時間．

Traffic ネットワーク上に流れる全トラヒック量．

Download Time は利用者側，Traffic はサービスプロバイダ側の観点からの評価である．

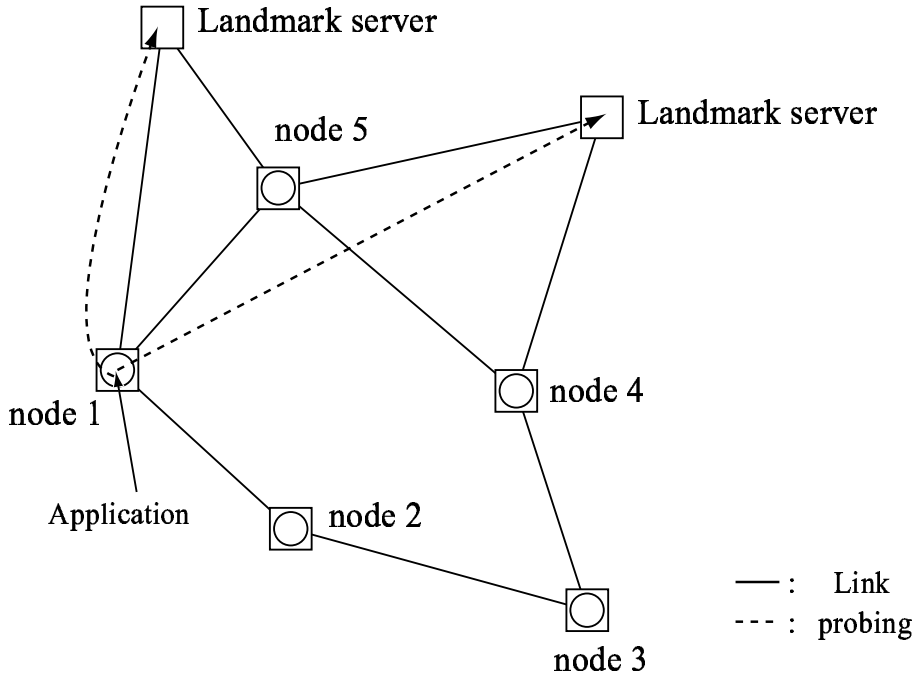


図 3–3: Simulation model

3.6.2　シミュレーションモデル

まず，ネットワークトポロジジェネレータである BRITE[41] を用いて，Waxman 型のフラットなランダムトポロジ* を構築した．

次に，トポロジ上の N ノードからランダムに選択した n ノードをホストに，残りの $N-n$ ノードにルータを配置した．各ホストは，いくつかのコンテンツを保有し，それぞれのコンテンツをどのホストが保持しているかについては，全てのホストの間で既知であるものと仮定した．図 3.6.2 に，シミュレーションモデルを示す．

各パラメータについては，表 3–1 のように設定した．

Landmark サーバの位置，全てのコンテンツの配置を変化させて，15 回実験を行った．

表 3–1 より分かるように，ホスト間の伝播遅延は，全てのホスト間で 1ms である．この理由について，次に説明する．ns-2 では，ネットワーク上の各リンクについて，1) 帯域，2) 伝播遅延，が設定できる．したがって，パケットの遅延は次式で表せる．

$$\begin{aligned} \text{遅延} &= \text{伝播遅延} \\ &\quad + \frac{\text{パケットサイズ}}{\text{帯域}} \\ &\quad + \text{キュー待ち時間} \end{aligned} \tag{3·7}$$

伝播遅延 >> (パケットサイズ/帯域) の場合では，各ホストが最適ホストを伝播遅延のみ

*ネットワークトポロジについては，**付録 A** を参照のこと．

表 3–1: Simulation parameters

パラメータ	値
ホスト数	1000
Landmark サーバ数	10
コンテンツ数	100
各コンテンツのサイズ	10Mbyte
各ホストが 保持しているコンテンツ数	5
各ホスト間の帯域	1-11Mbps の一様分布
各ホスト間の伝播遅延	1ms
ネットワークトポロジ	Waxman

表 3–2: The average value of “Download time”

Random	Best	Proposed
3396	2533	3196

[sec]

を考慮して選択する．しかし，最適ホストは，より多くの帯域を提供できるホストであり，帯域をより考慮する必要がある．

文献 [38] では最適ホスト選択メトリックとして，RTT が最適であることを示しており，これはより小さい遅延をもつリンクがより広帯域を保持する傾向があるということを表している．したがって，本シミュレーションにおいて，より広帯域をもつリンクがより小さな遅延をもつようにするため，各リンクの伝播遅延を 1ms 固定に設定した．しかしながら，実ネットワークにおいて，この仮定はすべてのリンクにおいて成立するわけではない．シミュレーショントポロジにおいて，どのように伝播遅延，帯域を割り当てていくかは，今後も十分検討する必要がある．

3.6.3　実験結果

図 3–4 および図 3–5 に，それぞれ “Download Time” と “Traffic” の実験結果を示す．

表 3–2 に “Download Time” の平均値を，表 3–3 に “Traffic” の平均値を示す．

表 3–2 および表 3–3 より，ランダムに最適ホストを選択した場合と比較し，提案手法ではより短い “Download Time” とより少ない “Traffic” が実現できていることが分かる．しかしながら，図 3–4 および図 3–5 を見れば分かるとおり，何回かの実験では，ランダムより悪い結果を出力している．

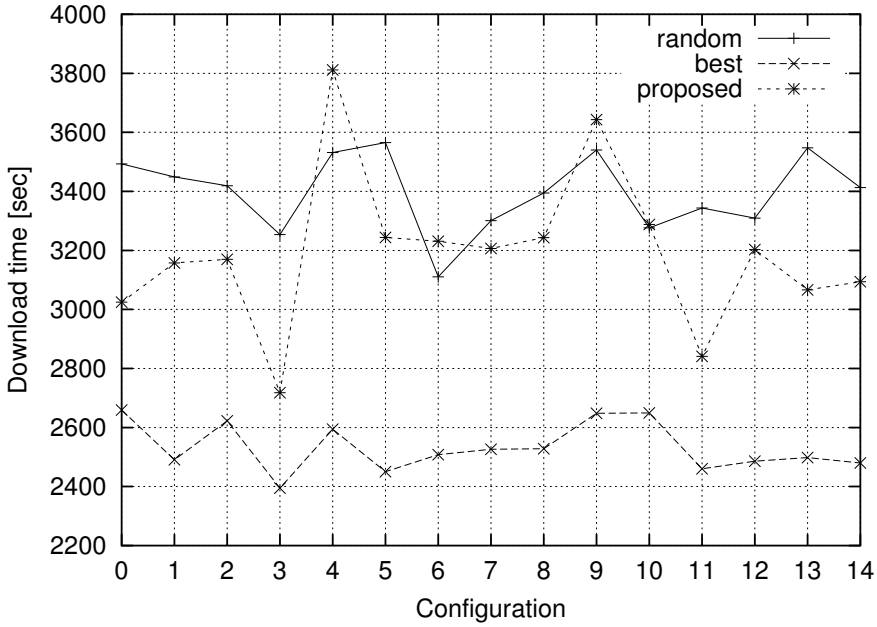


図 3–4: The result of “Download time”

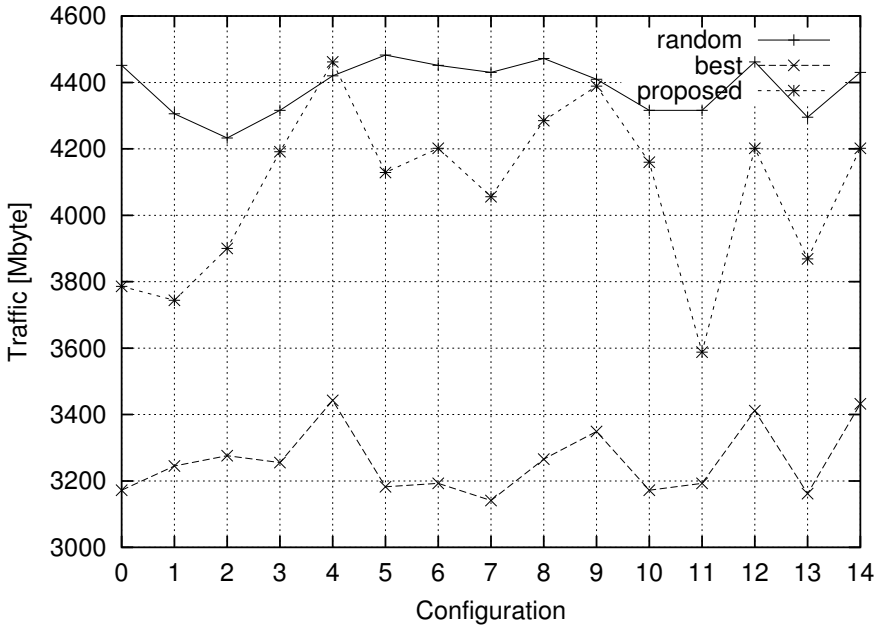


図 3–5: The result of “Traffic”

表 3–3: The average value of “Traffic”

Random	Best	Proposed
4386	3259	4077

[Mbyte]

3.6.4　考察

以上の結果より提案手法の多少の有効性は実証できた．しかしながら，ランダムに最適ホストを選択した場合より悪い結果を出力している場合もあった．この理由として，シミュレーションに用いたネットワークトポロジが考えられる．Waxman 型のようなフラットなトポロジでは，すべてのホストが類似した位置ベクトルを保持しており，その結果，最適ホストを選択し間違える可能性が高くなったものであると考えられる．一方，Transit-Stub 型のような階層型を有するトポロジであり，このトポロジを用いた場合，Waxman 型を用いた場合と比較して，各ホスト間で異なる位置ベクトルを保持すると思われる．インターネットは，Transit-Stub 型のような階層型を有していると考えられている．したがって，Transit-Stub 型を用いての実験および実ネットワークの実験においてはよりよい結果が得られるものと思われる．

3.7　本章のまとめ

本章では，分散キャッシュ型コンテンツ配信アーキテクチャにおいて，1) コンテンツ発見手法, 2) 最適ホスト選択手法，が重要であるという前章の指摘を踏まえ．インターネット上での各ホストの位置を幾何学空間に射影する RTT 予測手法を応用した最適ホスト選択手法の提案を行った．具体的には，1) Landmark と呼ばれるホストをインターネット上に複数配置する，2) 各ホストが Landmark 間との RTT 測定を行い，その値を基に位置ベクトルを構成する，3) 各ホストの位置ベクトルの相関係数を算出し，その値がより 0 に近いホストを最適ホストとして選択する，という手法である．相関係数の比較を行うことで，従来の RTT 予測手法をそのまま最適ホスト選択に適応する場合と比較して，様々な利点が得られることを述べた．また，提案手法の性能評価として，各端末より構成される P2P ネットワークを想定したシミュレーション実験を行った．各コンテンツを全端末間でランダムに配置した上で，1) 各端末が目的のコンテンツ群をダウンロードする際に要する時間，および 2) その際のネットワーク上に流れる全トラヒック量，に関してそれぞれ，a) ランダムに最適ホストを選択した場合，b) すべてのホストに対してプロービングを行い，最適ホストを選択した場合，c) 提案手法を用いて最適ホストを選択した場合，で比較を行った．ダウンロードする際に要する時間および全トラヒック量共に提案手法により，ランダムに最適ホストを選択する場合と比べ，削減可能なことを確認した．今後の課題については，6.2 で述べる．

第4章

ALMによるコンテンツ配信

4.1 はじめに

大規模コンテンツ配信を実現する上で，マルチキャスト機能は，必要不可欠な技術である．しかし，2章で述べたとおり，IPマルチキャストはそのアーキテクチャに本質的に起因する様々な問題点を内包しており，普及の兆しがあまりみられていない．そこで，我々は，各端末でマルチキャスト機能を提供するALMに注目し，ALMを用いた大規模配信アーキテクチャについて検討を行う．本章では，まずALMツリーの基本的な構築方法を述べる．次に，ALMツリーの性能を測る上で用いられる性能評価尺度について述べる．最後に，各ALMプロトコルに関して，アーキテクチャの詳細，および問題点について述べる．また，ALMツリー構築アルゴリズムについても整理する．

4.2 ALM構築方法

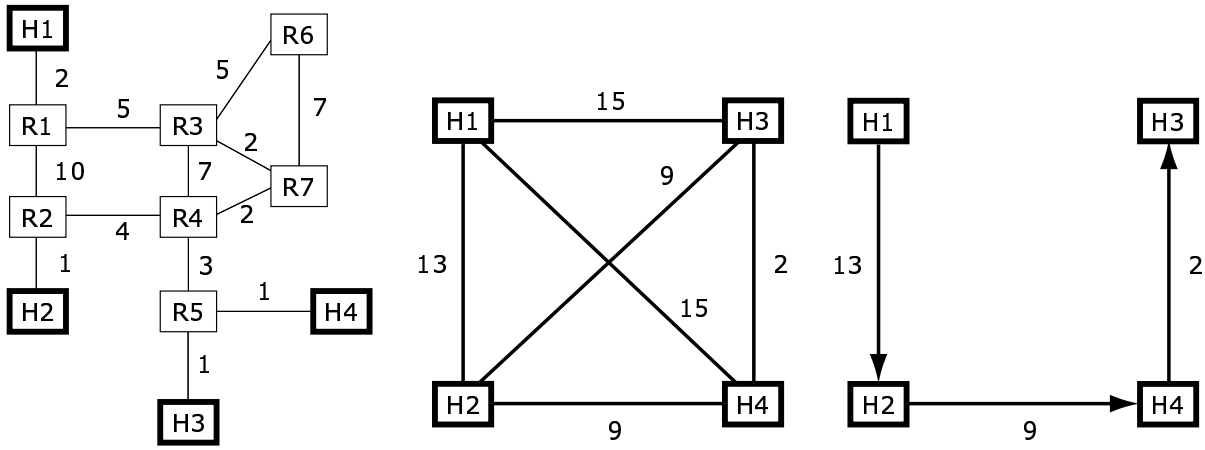
ALMでは，セッション参加端末を以下の2つのトポロジに組織化する．

1. コントロールトポロジ

   コントロールトポロジ上で隣接する端末同士が定期的に，リフレッシュメッセージの交換を行い，ある端末の予期せぬ離脱の発見，およびトポロジの修正などを行う．また，より良いデータトポロジを構築するため，端末間のRTT情報などを互いにコントロールトポロジを通じて交換し合う．

2. データトポロジデータトポロジは，コントロールトポロジ上のサブセットであり，データの配信経路を表すために用いられる．

一般的に，コントロールトポロジはメッシュ型，データトポロジはツリー型で構築される．なぜなら，コントロールトポロジで交換されるメッセージのサイズは小さく，メッシュ型でトポロジを構築しても大して帯域を圧迫しないからである．また，メッシュ型で構築することで，多数の端末間で情報交換を頻繁に行うこととなり，端末離脱の早期発見，よりよいデータトポロジを構築することが可能となる．一方，データトポロジを用いて配信されるデータは，一般的に非常にサイズが大きいため，ツリー型で構築される．それゆえ，多数の



(a) 物理ネットワーク　(b) コントロールトポロジ　(c) データトポロジ

図 4–1: コントロールトポロジとデータトポロジ

ALM プロトコルにおいて，コントロールトポロジのことをメッシュ，データトポロジのことをツリー (ALM ツリー) と呼んでいる．

コントロールトポロジおよびデータトポロジの例を図 4–1 に示す．

4.2.1　ALM プロトコルの分類

コントロールトポロジおよびデータトポロジの構築の仕方により ALM プロトコルは以下に分類できる [42]．

1. 集中型アプローチ

   端末間の情報を中央サーバに集め，それらの情報を基にトポロジを算出する．そして，トポロジ情報を配信する．

2. 分散型アプローチ

   分散型アプローチは，コントロールトポロジおよびデータトポロジをどのような順番で構築するかにより，以下に分類できる．

   a) Mesh-first アプローチ

      始めに参加端末をオーバレイ上にメッシュ型で組織化する．すなわち，コントロールトポロジがまず構築される．そして，そのトポロジ上で，ルーティングアルゴリズムを実行することで，ツリーを構築していく．

   b) Tree-first アプローチ

      始めに配信ツリーを構築した上で，各端末がセッションに参加している複数の端末を発見し，その端末との間でいくつかの追加パスを張り，コントロールトポロジを構築していく．

4.4 において，従来より提案されている各 ALM プロトコルについて，実際に分類を行う．

4.3　ALM 性能評価尺度

ALM では，各端末がマルチキャストツリー上のノードとしての機能を果たすことで，IP マルチキャストのようにネットワークストラクチャの変更を要求せずにマルチキャストツリーを構築できる．しかし，その一方で，ALM は，端末間のユニキャストコネクションにより成り立つため，あるリンク上を同一パケットが何度も通過してしまうなど，IP マルチキャスト配信ツリーと比較し，ネットワークリソースを消費してしまう．また，ALM では，各端末をチェーン上に結んでいくこととなるため，コンテンツを持っている端末自身と直接コネクションを確立した場合と比較し，遅延が大きくなる．したがって，各 ALM プロトコルの提案にあたっては，提案手法の性能評価が重要となる．以下に性能評価に用いられる評価尺度について挙げる．

1. データパスの品質

   a) ストレス

   各リンクまたは各端末を同一のパケットが何度通過したかを表す．IP マルチキャストでは，各リンクのストレスは 1 となる．

   b) ストレッチ

   各端末を経由することによる遅延の度合いを表す．次式により表される．

$$\frac{\text{送信元から受信先までのオーバーレイネットワーク上での遅延}}{\text{送信元と受信先間で直接コネクションを確立した場合の遅延}} \tag{4·1}$$

   したがって，各端末が直接送信元と結ばれるスター型の配信では，各端末のストレッチは 1 となる．

   c) 次数 (Node degree)

   各端末から何個のコネクションが確立されているかを表す．次数が高くなるほど，多くのコネクションが確立されている状態であり，帯域の消費も大きくなる．

   d) リソース使用率 (Resource usage)

   ネットワーク全体でのネットワークリソース使用率を表す．次式で表される．

$$\sum_{i \in Link} Delay_i \times Stress_i \tag{4·2}$$

図 4–2 に，図 4–2(a) を物理ネットワークとして，オーバーレイネットワークを構築した場合のストレッチおよびストレスの例を示す．

図 4–2(b) は，すべての端末が，コンテンツを保持している端末に直接接続している例であり，ストレッチはすべて 1 である．しかし，コンテンツ保持端末付近でのストレスが増加数する．

図 4–2(c) および図 4–2(d) は，オーバーレイネットワークとして ALM を構築した例である．図 4–2(c) では，ストレス，ストレッチ共に大きい値を示しているのに比べ，図 4–2(d) では，ストレス，ストレッチ共に低い値に抑えつつ，バランスの取れたトポロジが構築されているといえる．したがって，図 4–2(d) のようなオーバーレイネットワークを構築することが望まれる．

2. エンド端末のパフォーマンス

   a) 端末停止後のパケットロス率 (Losses after failures)
   ある端末が予期せず動作を停止させた場合にどの程度のパケットロスが引き起こされるかを表す．

   b) リカバリータイム
   ある端末がセッションから離脱した際，再び通常のサービスを受けられるようになるまでに要する時間を表す．

   c) データパケット受信開始遅延 (Time to first packet)
   セッション参加開始後，実際にデータパケットを受信し始めるまでに要する時間を表す．

3. コントロールオーバーヘッド

   オーバーレイネットワークを維持していくために必要なオーバーヘッドを表す．

4.4　ALM に関する従来研究および問題点

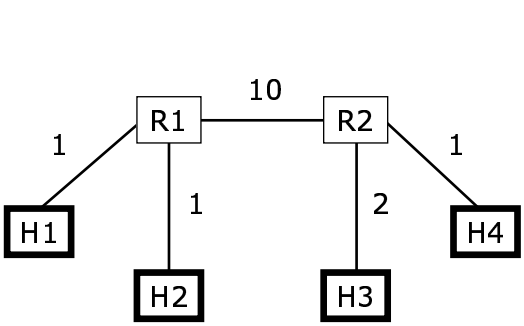
以下では，従来より提案されている各種 ALM プロトコルについて，1) 集中型および 2) 分散型 - a) Mesh-first アプローチ，b) Tree-first アプローチ，に分類しつつ，整理する．また，各種プロトコルの問題点についても言及する．

4.4.1　集中型 ALM プロトコル

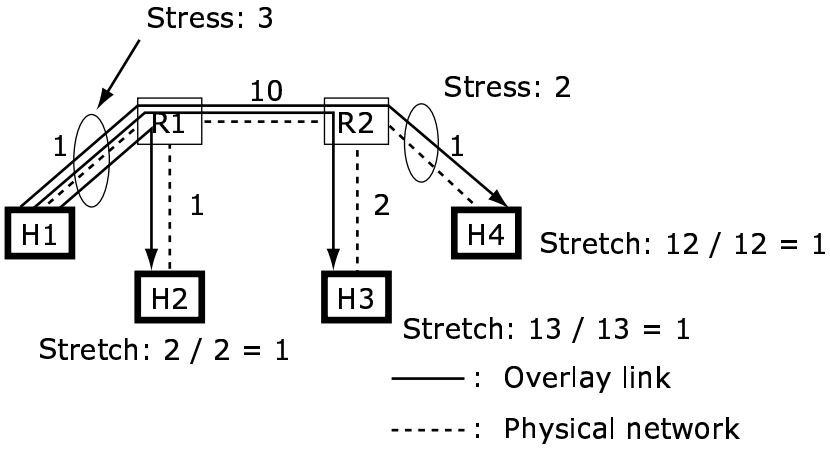
Application Level Multicast Infrastructure(ALMI)[43]

ALMI は，ビデオ会議システムなど，1) 参加人数が少ないがセッション数が多い，2) ネットワーク上に広く参加者が分散している，場合に適した ALM プロトコルである．

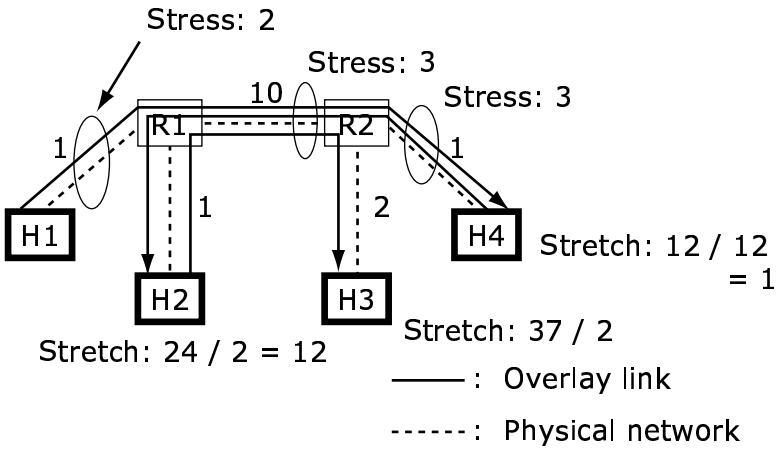
アーキテクチャ　ALMI のセッションは，セッションコントローラと複数のセッションメンバより構成され，セッションメンバ間で，双方向の共有木が作成される．木の作成は，セッ



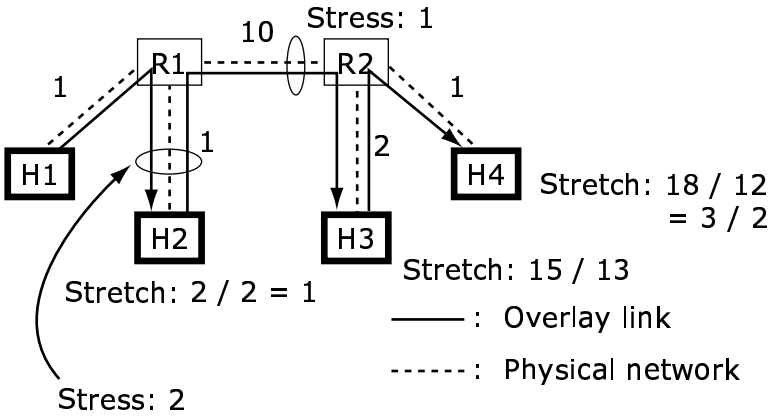
(a) 物理トポロジ



(b) Fully connected



(c) 悪いオーバーレイネットワークの例



(d) 良いオーバーレイネットワークの例

図 4–2: データパスの品質例

ションメンバから集めた情報を基にセッションコントローラで行われる．すなわち，セッションを中央制御で管理する．中央制御を行うことで，セッション作成，セッションの管理などが容易となる．

問題点　セッションコントローラによる中央管理方式であるため，セッションコントローラの信頼性，耐障害性が問題になる．しかしながら，文献 [43] では，1) バックアップコントローラを用意しておく，2) コントロールオペレーション (エンド端末の参加，ALM ツリーの改善など) を提供できなくても，既に構築されている ALM ツリーを用いてサービスの継続が可能である，ことを挙げ，中央制御の耐障害性を一概に低いと見なすことはできないと主張している．また，中央制御の方が，セッション管理やトポロジの構築などが容易になるという利点があり，耐障害性に対する欠点を補うだけ利点を提供しているともいえる．しかし，大規模なコンテンツ配信においては，中央コントローラが制御可能なセッション参加端末数を大幅に超えてしまうため，ALMI のような集中型の ALM プロトコルは，採用できないと考えられる．実際，ALMI は，参加人数が少ないアプリケーションをターゲットにしている．

4.4.2　分散型 (Mesh-first アプローチ)

Narada[44]

分散型 Mesh-first アプローチである Narada プロトコルについて述べる．

アーキテクチャ　Narada プロトコルは，1) 自組織化 (Self-organizing)，2) オーバーレイの効率化 (Overlay efficiency)，3) インクリメンタルな自トポロジの改良 (Self-improving in an incremental fashion)，を方針として設計されている．

実際には，全域木を直接構築するのではなく，Mesh, Tree という順番に 2 段階アプローチを取っている．その理由として，

1. 同一セッションにおいて複数の送信端末が存在する場合，グループマネージメント機能を送信端末ごとに用意し，多数の木の間で複製するのではなく，メッシュとして抽象化し，扱うべきである．

2. メッシュ分割修正，メッシュ最適化のための分散ヒューリスティックな手法は，Tree-first アプローチで要求されるループ回避アルゴリズムより容易に実装できる．

3. データトポロジ構築にあたり DVMRP[14] などの標準ルーティングアルゴリズムを用いることができる．

4. Mesh を最初に構築することで，参加端末の離脱に強いため，高負荷なトポロジ修正アルゴリズムの作動回数を削減できる．

が挙げられている．

Narada におけるセッション確立の手順は以下のとおりになる．

1. セッションへの参加 4–3(a)

   Rendezvous point(RP) *よりセッションに参加している端末群に関する情報を取得する．その情報を基に，接続する端末をランダムに選択し，いくつかの端末とコネクションを確立する．これにより，メッシュが構築される．コネクション確立後は，互いにリフレッシュメッセージを交換しあう．

2. メッシュの修正 4–3(b)

   ある端末より T_m 時間リフレッシュメッセージが受信できなかった場合，その端末は停止したものとみなす．端末の停止により，構築したメッシュが分断された場合は，メッシュ修正アルゴリズムを起動させる．

3. メッシュの改良 4–3(c)

   セッション参加，メッシュの修正により構築されたメッシュトポロジは，各端末がランダムに接続端末を選択していることなどがあり，ネットワーク距離などの面で準最適となっている．したがって，メッシュ改良アルゴリズムが必要となる．Narada では，各端末間同士で，ランダムにプローブを行い，その情報を基にユーティリティ値と呼ばれるものを算出している．この値を基に，新しいリンクの確立，廃棄を決定している．

   また，文献 (45) では，各端末間の帯域も考慮に入れたメッシュの改良アルゴリズムについて検討を行っている．

4. データトポロジの作成 4–3(d)

   構築されたコントロールトポロジ上で Reverse Path Forwarding(RPF) アルゴリズムを動作させることでデータトポロジを構築する．

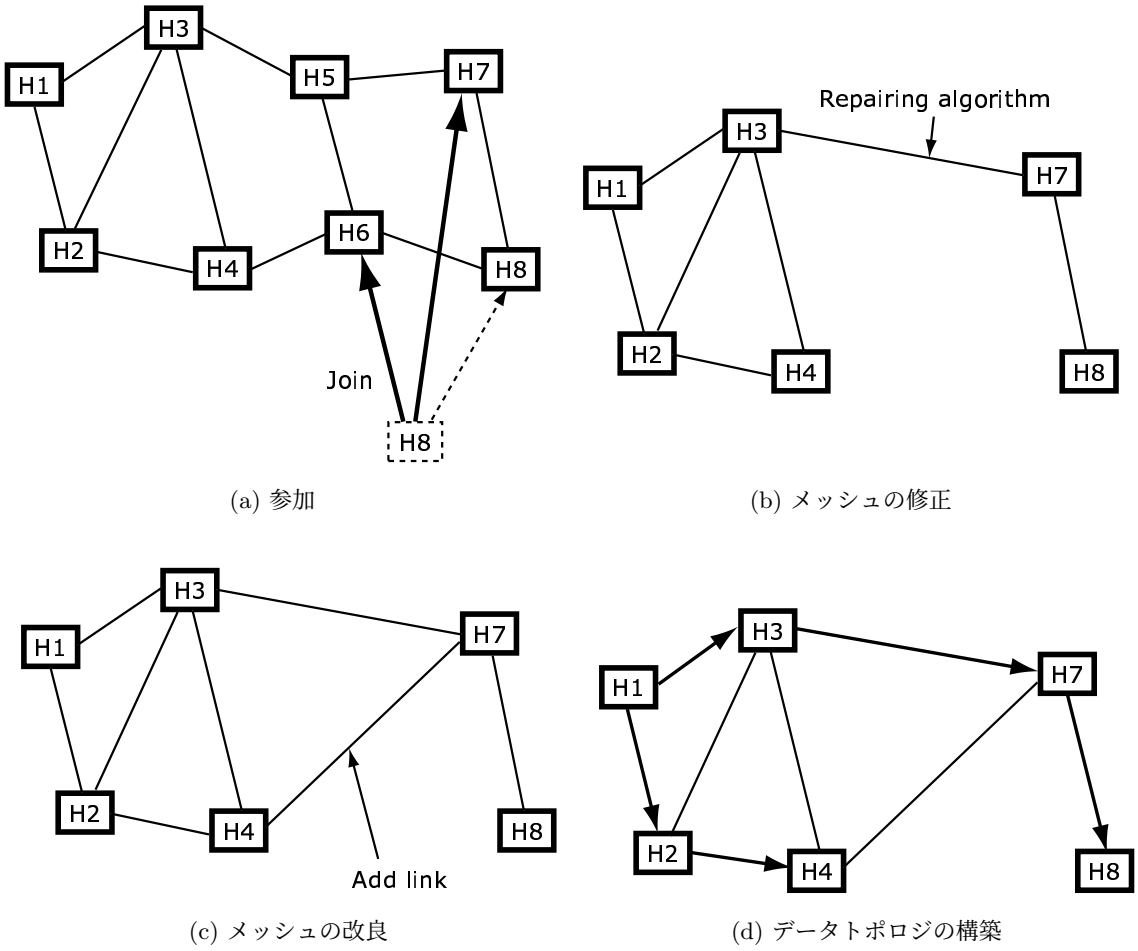
問題点　Narada では，各端末が全端末への距離を含んだルーティングテーブルを管理するなど，大多数の端末が参加するアプリケーションではスケールしない．実際に，文献 (44) の中で，Narada プロトコルは，オーディオ・ビデオ会議，仮想遠隔授業，ネットワークゲームなどの参加人数が少なく，ネットワーク上に疎に分散されているマルチキャストグループに適しており，インターネット TV などの用途には向かないことを述べている．

4.4.3　分散型 (Tree-first アプローチ)

Host Multicast Tree Protocol(33)

分散型 Tree-first アプローチである HMTP について述べる．

*セッションに参加している端末を管理しているサーバは，一般に Rendezvous point(RP) と呼ばれる．多くの ALM プロトコルでこのようなサーバがセッションへの参加プロセス時に用いられる．



(a) 参加　(b) メッシュの修正

(c) メッシュの改良　(d) データトポロジの構築

図 4–3: Narada プロトコル

アーキテクチャ　HMTP は，Tree-first アプローチに属するプロトコルである．すなわち，HMTP では，まずデータトポロジである配信ツリーを構築する．ただし，その後，Narada のように明示的なメッシュは構築しない．各メンバ同士が定期的にお互いの端末を認識しあい，その情報をキャッシュしておくことで，データトポロジの切断などに対応する．

以下に，HMTP におけるセッション確立および維持の手順を述べる．

1. セッションへの参加

   セッション参加手順を以下に示す (図 4–4).

   a) 全てのメンバを valid potential parents とマークする．また，過去の potential parents を保存するためのスタック S を確保する．

   b) HMRP に対して問い合わせを行い，root ノードを発見する．また，root ノードを potential parent とみなす.

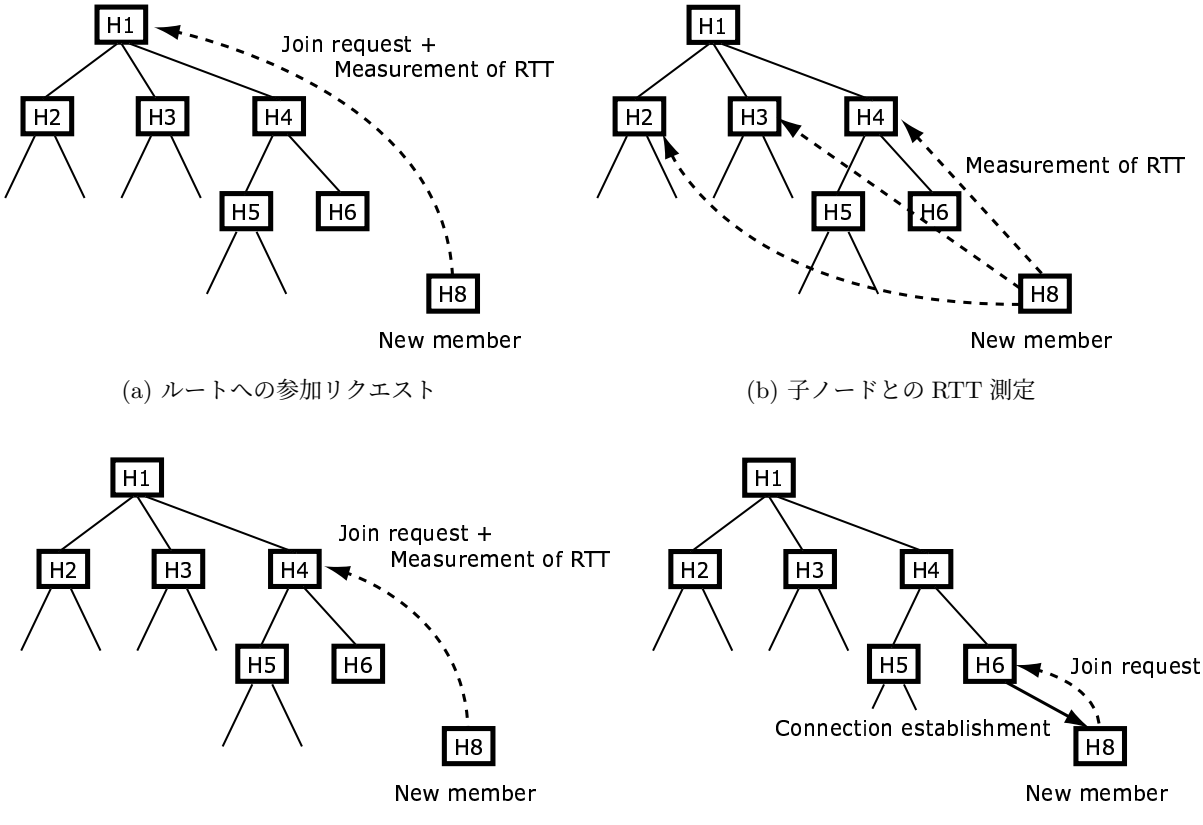
   c) potential parent に対して，そのノードの子ノードを発見するために，問い合わ

せを行う．また，自ノードー potential parent およびそれら子ノード間の RTT を測定する．

d) それらの中で invalid とマークされたものを除外した上で，自ノードと最近傍のノードを選択する．この時，すべてのノードが invalid とマークされていた場合，スタック S のトップに登録されているノードをポップする．そのノードを新しい potential parent に設定し，手順 3 に戻る．

e) 最近傍ノードが現在の potential parent ではない場合は，現在の potential parent をスタック S に push し，この最近傍ノードを新しい potential parent に設定し，手順 3 に戻る．

f) 最近傍ノードが現在の potential parent の場合は，そのノードに対して，join リクエストを送信する．拒否された場合は，その potential parent に対して invalid とマークした上で，手順 4 に戻る．許可された場合は，そのノードが親ノードとなる．したがって，そのノードとの間でユニキャストコネクションを確立する．



(a) ルートへの参加リクエスト　(b) 子ノードとの RTT 測定

(c) 選択ノードへの参加リクエスト　(d) 親ノードの決定

図 4–4: HMTP におけるセッションへの参加プロセス

このようなツリーの上位にある端末より順に，参加リクエストを送信するアルゴリズムは，HMTP 以外の ALM プロトコルにおいても用いられる一般的なアルゴリズムで

ある．

2. ツリーの維持，参加端末停止への対応

   すべての子ノードは，自身の親ノードとの間でリフレッシュメッセージを定期的に交換している．それに対して，親ノードは，子ノードに対して，パスメッセージを返信する．このようなメッセージを交換し合うことで，予期せぬ端末の離脱を発見できる．また，パスメッセージには，ルート (送信者) から親ノードまでパスを記述したルートパス情報が含まれている．すなわち，各ノードは，親ノードからのルートパス情報を受信した際，その情報に自ノードの情報を追加し，それを自分の子ノードに対するパスメッセージとして送信するわけである．これにより，後述するツリーの改善およびツリー分断時のツリー構造変更に伴うデータパスのループ生成抑止が可能となる．

3. ツリーの改善

   ネットワークの状態およびセッション参加端末は，時間的に変化するため，ツリーの品質が時間の経過と共に劣化する．そのため，ツリーの品質を改善するアルゴリズムが必要となる．そこで，ルートパス上に存在するノードからランダムに 1 ノード選択し，そのノードをルートとしたセッション参加アルゴリズムを試みる．それにより選択された親ノード候補が現在の親ノードよりネットワーク距離的に近い場合は，そのノードを新しい親ノードとして選択する．ただし，頻繁なパスの変更を防ぐため，改善値がある閾値を超えた場合にのみ，親ノードの変更を行う．その後，親ノードの変更を自分の子孫ノード† に対して，パスメッセージを伝播させることで伝える．

4. ツリー分断への対応

   あるノードが予期せぬ停止を起こした場合，そのノードの親ノードおよび子ノードはセッション継続のための適切な処理を行う必要がある．親ノードは，停止したノードを自身の子ノードリストから削除する．子ノードは，以下の手順を行うことで新たな親ノードを発見する．

   a) ルートパスをスタック S に格納する．自ノードと停止した親ノードをスタックからポップする．

   b) スタックから，グランドペアレントノードをポップし，potential parent とする．ランダム秒待機後，セッション参加アルゴリズムを手順 3 から開始し，新しい親が見つかるまで行う．ランダム秒待機することで，ツリー分断アルゴリズムの対象ノードが一斉にグランドペアレントノードに対してリクエストを送信することを抑止できる．

5. ループ生成抑止

---

†子ノードを含め，自ノード以下に接続された全てのノードのこと．

新しい親ノードを選択する際に，ルートパスにそのノードが含まれていた場合，ツリー上にループが形成されることになる．したがって，ルートパスを常に監視することでループ生成を抑止することができる．しかし，あるノードが親ノード変更後，その情報を含んだパスメッセージを受信していない子孫ノードが，親ノードの変更を行った場合，ループが生成される可能性がある．その場合は，ループ発見ノードがループの解消を行う．

問題点　HMTP では，Narada のように全端末間情報ではなく，数端末間の情報を保持することでサービスを提供できる．すなわち，インターネット TV などの大規模コンテンツ配信の用途には向くと思われる．しかし，メッシュが構築されないため，時としてツリー分断からの回復が遅れ，多くのパケットロスが引き起こされる．

また，HMTP におけるツリーの改善法では，各ノードがそれぞれ最適化を実行するため，ツリーの最適解に収束するまでには時間がかかると思われる．また，データツリーの品質は，ツリーの最適解と比較した場合，かなり劣化したものとなる可能性がある．

NICE[46]

分散型 Tree-first アプローチである NICE について述べる．

アーキテクチャ　NICE は，ストリームミングやコンテンツ配信など多数の受信端末へ伝送する場合を想定しているプロトコルである．一般に，多数の受信端末間でコントロールトポロジを構築すると，コントロールオーバーヘッドが増加する．したがって，コントロールパケットによる帯域の圧迫にもつながる．そこで，NICE では，隣接ノード間でクラスタリングを構築し，それを単位にコントロールトポロジを構築することで，コントロールオーバーヘッドを抑えつつ，データ配信ツリーを構築している．

以下のルールを全端末間で適用することで上記のことを実現している．

1. 全端末は，必ず最下層レイヤである L_0 に属する．
2. クラスタリングプロトコルが，L_0 に属す端末をクラスタセットへと組織化する．
3. レイヤ L_i に属する全てのクラスタのヘッドは，レイヤ L_{i+1} に参加する．したがって，レイヤ L_i に参加している端末は，レイヤ $L_0, \cdots, L_{i-1}$ に属するクラスタのヘッドとなる．
4. レイヤ L_i に属することがなければ，レイヤ $L_j (j > i)$ に属することはない．
5. 各クラスタは，k から $3k-1$ のサイズを持っている．k は定数である．また，クラスタヘッドは，クラスタの中心に位置する端末がつとめる．
6. 全レイヤ数は，$log_k N$ レイヤであり，最上位レイヤには，1 端末 (ルート) のみ属する．

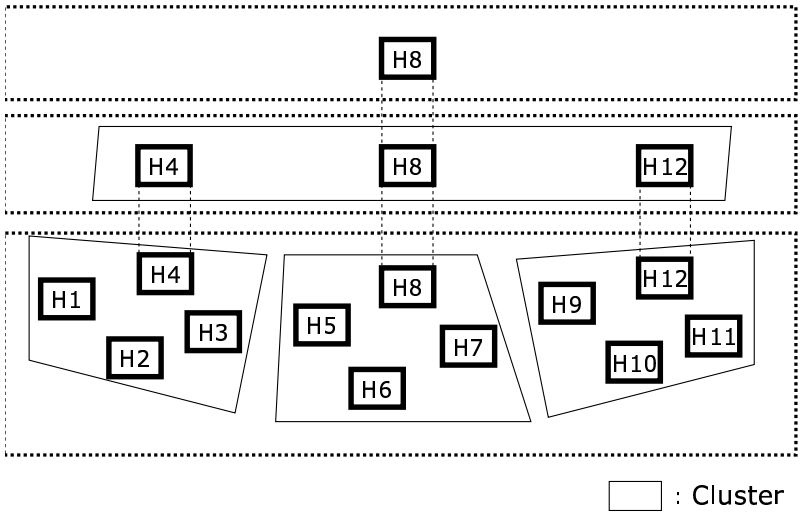


図 4–5: NICE におけるレイヤ構造

各クラスタがコントロールトポロジに相当する．各クラスタのサイズが，k を下回った場合は，クラスタのマージが，逆に $3k-1$ を上回った場合は，クラスタのスプリットを行う．

図 4–5 に，NICE におけるレイヤ構造の例を示す．

NICE におけるセッション参加は，上位レイヤのクラスタヘッドに順にセッション参加リクエストを送信することで行っていく．これは，HMTP のセッション参加アルゴリズムと基本的には同じである．

データトポロジは，以下に示すパケットフォワーディングルールを適応することで実現される．データ送信端末は，自分が属するコントロールトポロジ上のすべての端末に対して，データパケットを送信する．レイヤ $L_0, \cdots, L_j$ に属する端末 h がデータパケットを端末 p より受信したとする．いくつかのレイヤ中，p および h が同じクラスタに属するレイヤを L_i とすると，端末 h は，クラスタ $C_k(k \neq i | i, k \in layer)$ のクラスタヘッドの場合，そのクラスタに属する他の端末に対してデータパケットの送信を行う．

問題点　クラスタヘッドは，そのクラスタに属する端末にデータパケットを送信する機能が要求されるため，少なくとも $3k-2$ 端末に対して，データを送信できる出力帯域が必要とされる．また，より上位レイヤに属するクラスタのヘッドほど要求帯域が大きくなる．これは，クラスタサイズが参加端末の使用可能帯域により決定してしまうということを表している．したがって，クラスタサイズは基本的に小さくならざるをえない．これは，コントロールトポロジの範囲が小さくなることを意味し，全端末間情報を用いた ALM ツリーの最適解と比較し，大幅に性能が劣化するものと思われる．

4.4.4　その他の ALM プロトコル

上記に挙げた ALM プロトコルの他にも，以下のような ALM プロトコルが提案されている．

Delaunay Triangulation Overlays[47]　オーバーレイネットワークとして，ドローネ三角形を用いた構築法を提案している．まず，各端末に平面座標 (x, y) を割り当てる．次に，オーバーレイ上の各端末に関連付けれた全ての頂点より構成されるドローネ三角形を構築する．構築されたドローネ三角形上のエッジにより接続された各頂点に関連付けられた各端末間をオーバレイネットワーク上で接続する．ドローネ三角形上でコンパスルーティング [48] を行うことで，オーバレイネットワーク上でルーティング情報を構築するルーティングプロトコルを用いる必要がなくなるという利点がある．ただし，ドローネ三角形を用いたネットワークトポロジからオーバレイーネットワークトポロジへのマッピングは，準最適なマッピングであり，物理ネットワークを十分に考慮したオーバレイトポロジを構築することができない．

Bayeux[49]　Bayeux は，2.5.2 であげたコンテンツ検索技術の一つである Tapestry[29] 上に構築した ALM プロトコルである．これらのプロトコルの利点は，Delaunay Triangulation Overlays と同様にルーティングプロトコルを動作させる必要がない点にある．Bayeux では，Tapestry で構築した分散ハッシュテーブルを基にして，オーバーレイネットワーク上でのネクストホップを決定する．類似したプロトコルに，Pastry[30] を基にした ALM プロトコルである Scribe[50] がある．

4.5　ALM ツリー構築アルゴリズム

ALM ツリーの構築アルゴリズムを用いることで，端末間のネットワーク距離情報が与えれらた場合，あるメトリックに最適なツリーを構築することが可能となる．ツリーの種類については，2.4.1 で取り上げた．基本的に構築されるツリーは，IP マルチキャストと ALM で違いはない．ただ，ツリーを構成するノードが，ルータであるか，端末であるかが違う．以下に ALM を対象とした集中型ツリー構築アルゴリズムおよび分散型ツリー構築アルゴリズムについて整理する．集中型ツリー構築アルゴリズムは，集中型アプローチの ALM プロトコルで，分散型ツリー構築アルゴリズムは，分散型アプローチの ALM プロトコルで用いることができる．

4.5.1　S. Y. Shi et al.[51]

Shi らは，ALM での Min max-latency problem(送信者－受信者間の最大ネットワーク距離を最小化させたツリーを構築する問題) に対する集中型アルゴリズムの提案を行っている．

アルゴリズムの詳細

以下に示す 2 つのプロセスより構成される．

1. 次数割り当てプロセス (dimensioning process)

   セッショントラヒックモデルより各端末でのトラヒック負荷を定量化し，各端末に適切な帯域を設定する．すなわち，ルーティングアルゴリズムで用いる入力ネットワーク上での各ホストの次数を決定する．

2. ルーティングアルゴリズム

   上記の次数割り当てプロセスにより設定した次数制約を維持しながら，送信者－受信者間の最大ネットワーク距離を最小化させたツリーを構築する．

帯域の分散，すなわち各ホストの次数を均一化することは，より遅延を増加させる傾向にある．すなわち，帯域と遅延は相反する関係にある．そこで，一方を固定値として設定し，もう一方を最適化する方法をとっており，この考えに基づく以下の 2 つのルーティングアルゴリズムが提案されている．

1. 次数制約付き半径最小全域木 (Minimum diameter, degree-bounded spanning tree: MDDBST)

   次数制約付き半径最小全域木 (Minimum diameter, degree-bounded spanning tree: MDDBST) は，各ホストの帯域を固定化し，ツリーの遅延最適化を行うアルゴリズムであり，Prim 法 [8] に類似した greedy algorithm(貪欲アルゴリズム) である．

   MDDBST アルゴリズムは，低遅延の ALM ツリーを構築することは可能である一方，各ホストの次数が均一に与えられているため，トラヒックの負荷分散の面で問題がある．そこで，高負荷のホストから近くの低負荷のホストへとコネクションを張りなおす方法がトラヒックを分散させる場合に有効である．このような考え方を応用したのが，次に示す load-balancing routing algorithm である．

2. 半径制約付き次数バランスを有する全域木 (Bounded diameter, residual-balanced spanning tree: BDRBST)

   半径制約付き次数バランスを有する全域木 (Bounded diameter, residual-balanced spanning tree: BDRBST) は，負荷分散を考慮に入れたアルゴリズムである．

   余剰帯域 (次数) $d_{max}(v) - d_T(v)$ の最小値を最大にすることによって，ボトルネックとなっているホストの負荷分散を行っている．しかし，その一方で，ホスト間のエンドツーエンドの遅延は増加する．

   そこで，遅延限界の条件を満たしながら，負荷分散を実現するために，$balance\,factor\ M$ を導入している．すなわち，M は半径 (diameter) と負荷分散 (load balancing) の間のトレードオフを調整するパラメータである．アルゴリズムは，MDDBST アルゴリ

ズムを改良することで実現している．最小な $\delta(v)$ ($\delta(v)$ は，ツリー上における各頂点および v 間の最長パスを表す) をもつノードを 1 つ選択するかわりに，$\delta(v)$ が小さい M 個のノードを選択する．そして，そのノードのうち，最小余剰帯域が最大化するようなノードを選択する．もしも，M=1 なら，このアルゴリズムは MDDBST のアルゴリズムと同一になる．一方，M がマルチキャストセッション中のホスト数と同一なら，負荷分散のみを考慮したルーティングアルゴリズムとなる．シミュレーションを通じて，小さな M の値 (e.g. 5) が送信者－受信者間の最大遅延の制約を満たしながら，良好な負荷分散が実現できることが示されている．

また Shi らは，文献 [52] で，上記手法の改善を行っている．

4.5.2　S. Banerjee et al. [53]

Banerjee らは，品質の良い ALM ツリーを構築するため，以下のツリーを構築する分散型アルゴリズムの提案を行っている．

1. 次数制約付き平均遅延最小有向全域木 (Minimum average-latency degree-bounded directed spanning tree)

   ホスト r をルートとし，各ノードにおいて次元制約数を満足しつつ，平均遅延を最小にする有向全木域 T を発見する．

2. 次数制約付き最大遅延最小有効全域木 (Minimum maximum-latency degree-bounded directed spanning tree)

   ホスト r をルートとし，各ノードにおいて次元制約数を満足しつつ，最大遅延を最小にする有方向全木域 T を発見する．

アルゴリズムの詳細

アルゴリズムは初期ツリー構築フェーズと部分的ツリー改善フェーズ (Local transformations phase) から成る．

初期ツリー構築フェーズ　実際にデータ配信を始める前に図 4–6 に示したアルゴリズムを用いて，ツリーを構築する．

各ホストの次数制約が 2 の場合の初期ツリーを図 4–7 に示す．

Local Transformations

各期間毎 (τ:transformation period) に，以下に示す部分的ツリーの改善を行う．

1. Child-Promote(図 4–8(a))

   あるノードがグラントペアレントノードに接続し，親ノードからは離脱する．

Procedure : $CreateInitialTree(r, S)$
$SortedS \longleftarrow$ Sort S in increasing order of dist. from r
　{ $Assert$: SortedS[1] = r}
$i \leftarrow 1$
for ($j \leftarrow 2$ to N) do
　while
　　($SortedS[i].NumChildren \ = \ SortedS[i].DegBd$)
　　$i++$
　end while
　$SortedS[j].parent \leftarrow Sorted[i]$
　$SortedS[i].NewChildren++$
end for

図 4–6: 初期ツリー構築アルゴリズム

2. Parent-Child Swap(図 4–8(b))

   あるノードが親ノードと関係を逆転させる.

3. Iso-level-2 Swap(図 4–8(c))

   グラントペアレントノードが共通なノード同士で親ノードの交換を行う.

4. Iso-level-2 Transfer

   あるノードが，親の兄弟ノードへと移動する.

5. Aniso-level-1-2 Swap(図 4–8(d))

   あるノードの子ノードと孫ノードが位置を交換する.

4.5.3　ALM ツリー構築アルゴリズムのまとめ

集中型アルゴリズムおよび分散型アルゴリズムの特徴および利点，欠点を整理する.

1. 集中型アルゴリズム

   利点 全端末間情報をもちいることができるため，パス変更の頻度自体は少なく，品質の良い配信ツリーが構築できる．また，常に大局的なツリー変更を実行するのではなく，大幅なツリーの改善を望めるツリーの一部分のみを改善することができるなど，ツリー構築アルゴリズムに柔軟性を持たせることが可能である.

   欠点 アルゴリズムを実行する端末が停止・離脱した場合，ツリー構築も停止してしまう．また，大局的なツリーの変更を行う場合，多数のデータパケットロスを伴う可能性がある.

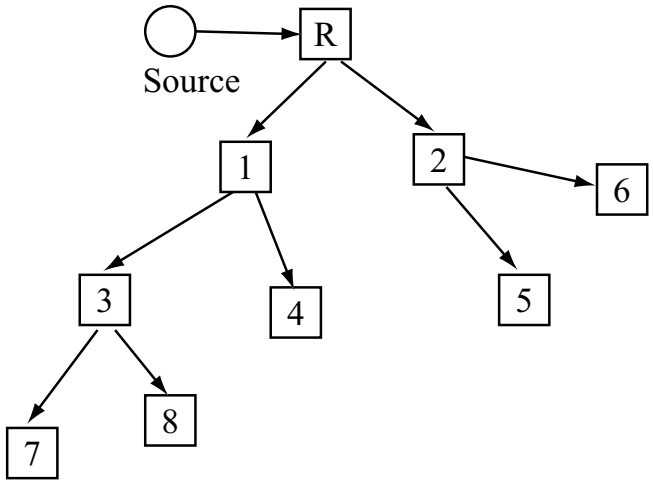
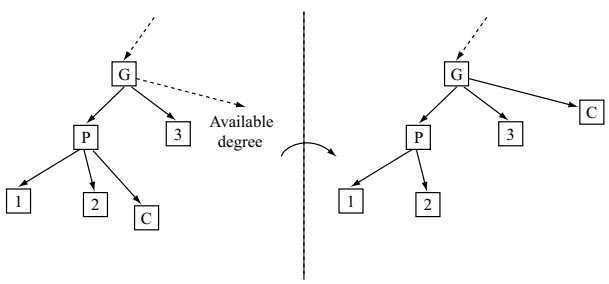
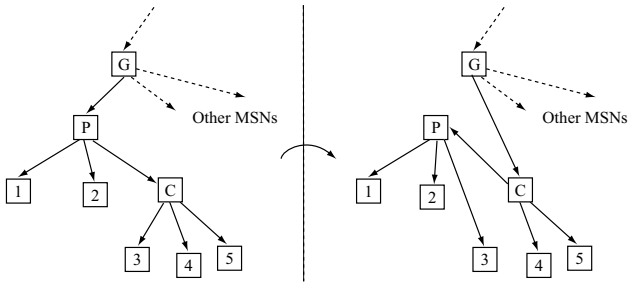


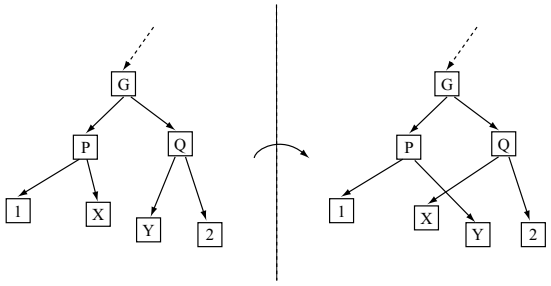
図 4–7: Initialization process



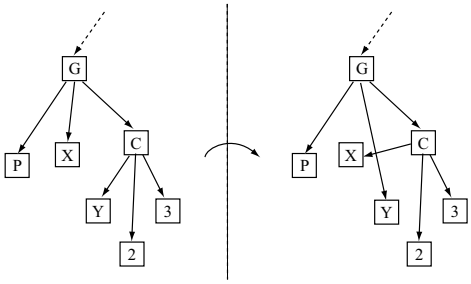
(a) Child promote operation



(b) Parent-child swap operation



(c) Iso-level-2 swap operation



(d) Aniso-level-1-2 swap operation

図 4–8: 部分的ツリー改善アルゴリズム

2. 分散型アルゴリズム

利点 ある端末が停止・離脱しても，ツリー構築を続行できる．また，局所的なパスしか変更されないため，集中型アルゴリズムのパス変更時と比較し，パス変更時のパケットロスが少ない．

欠点 局所的な情報のみを用いてツリーの改善を行うため，頻繁にパス変更が行われる可能性がある．すなわち，時としてツリーが安定しない場合がある．

このように，集中型アルゴリズム，分散型アルゴリズムはそれぞれ利点と欠点があり，これらは基本的にお互い相反する関係を持っている．

したがって，集中型アルゴリズム，分散型アルゴリズムの利点をうまく組み合わせたハイブリッド型が有効であると思われる．具体的には，集中型の制御を行う中央端末が存在する場合は，集中型アルゴリズムに基づくツリー構築を行う．そして，中央端末の停止・離脱により，集中型アルゴリズムを継続できない場合には，分散型アルゴリズムに切り替え，ツリーの改良を行う．

4.6　本章のまとめ

本章では，まず基本的な ALM ツリー構築手順，および ALM プロトコルの性能評価尺度について述べた．次に，従来より提案されている ALM プロトコルについて，1) 集中型アプローチ，2) 分散型アプローチ - a) Mesh-first アプローチ，b) Tree-first アプローチ，へ分類し，そのうえで各プロトコルのアーキテクチャの詳細および問題点を指摘した．最後に，ALM ツリー構築手法に関して，1) 集中型ツリー構築アルゴリズムおよび 2) 分散型ツリー構築アルゴリズム，に分類し，従来手法を分類，整理した．また，集中型ツリー構築アルゴリズム，分散型ツリー構築アルゴリズムの利点および欠点について論じ，これらを組み合わせたハイブリッド型が有効であるという点を指摘した．

第5章

クラスタリングを用いたALMツリー構築手法の提案

5.1 はじめに

本章では，まず，ALMを構成する端末内の遅延を測定する実験を行い，ALMツリー構築の際に，端末内の遅延を考慮する必要があるかどうかの有無を確認する．

次に，前章で挙げた各種ALMプロトコルの問題点およびALMツリー構築アルゴリズムの考察を踏まえ，クラスタリングを用いたALMツリー構築手法を提案する．

集中型ツリー構築アルゴリズムは，多数の端末が参加する大規模セッションにおいては，アルゴリズムを実行する中央端末への負荷が大きくなると共に，端末間のネットワーク距離を測るプローブパケットの量も増大する．逆に分散型ツリー構築アルゴリズムでは，これら状況変化に追従できるが，良好なツリーを構築するまで頻繁なツリー改善が行われるため，ツリーの安定に時間が要することが予測される．ALMでは，IPマルチキャストルータと比べ，セッションへの参加，離脱などが頻繁に起こる安定性のない端末が，マルチキャストツリーの構築を行うこととなる．したがって，状況変化に追従でき，しかもツリー改善を行う回数が少なく，良好な品質のツリーが構築できるALMツリー構築手法が望まれる．本提案手法は，集中型アルゴリズムと分散型アルゴリズムのハイブリッドアルゴリズムとして動作可能なため，上記要求を満たすことが可能であると考えられる．最後に，シミュレーション実験により本手法の有効性について確認する．

5.2 予備実験 (各端末内部での遅延測定)

5.2.1 概要

ALMでは，各端末がALMツリーを構成し，パケットの複製，フォワーディングを行う．したがって，マルチキャスト機能を提供することを前提に作成されたIPマルチキャストルータとは異なり，各端末自身がボトルネックとなる可能性がある．そこで，ALMツリー構築アルゴリズムの提案にあたり，まず各端末内部でのパケットフォワーディングに伴う遅延がどの程度あるかを測定する実験を行った．

あるパケットを受信したら，接続している他の端末に対してフォワーディングを行うアプ

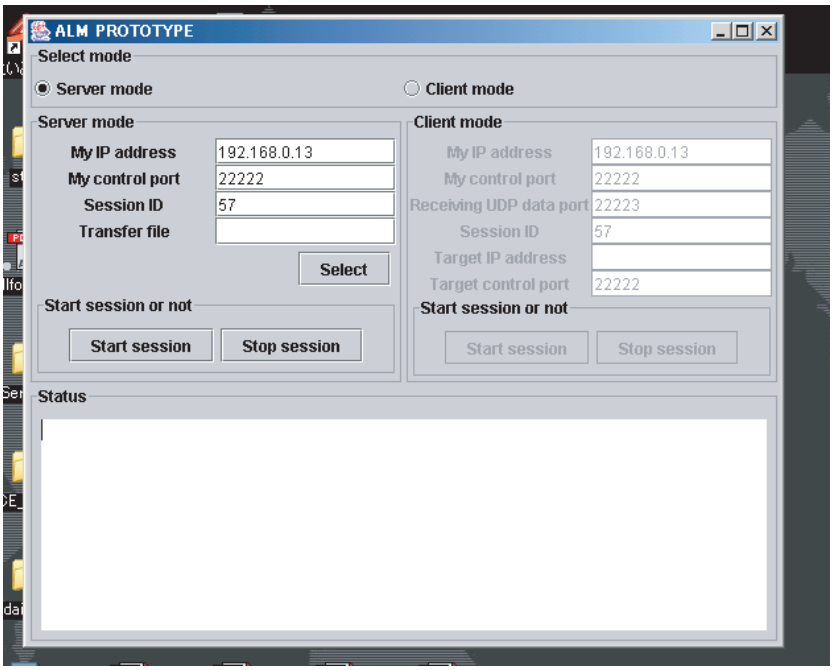


図 5–1: パケットフォワーディングアプリケーション

表 5–1: 端末内遅延測定結果

1 回目	2 回目	3 回目	4 回目	5 回目
10 msec	20 msec	10 msec	10 msec	10 msec

リケーション (図 5–1) を Java で実装し，3 台の PC を用いて，端末内遅延を測定した．

すなわち，図 5–2 の点線の四角形で囲まれた部分の遅延を測定した．

実験には，以下の PC を用いた．

ホスト 1 CPU: Pentium II 400MHz, RAM: 256MB, OS: Linux 2.4.18, Java 環境: JDK 1.3.0

ホスト 2 CPU: Pentium III 700MHz, RAM: 256MB, OS: Windows XP SP1, Java 環境: JDK 1.4.0

ホスト 3 CPU: Pentium III 2000MHz, RAM: 512MB, OS: Windows XP SP1, Java 環境: JDK 1.4.0

5.2.2　実験結果

5 回実験を試行した結果を表 5–1 に示す．

測定結果より，端末内での遅延は，非常にわずかである確認できる．本実験で端末内遅延の測定に用いた端末群は，現在では容易に手に入るスペックの PC である．したがって，一

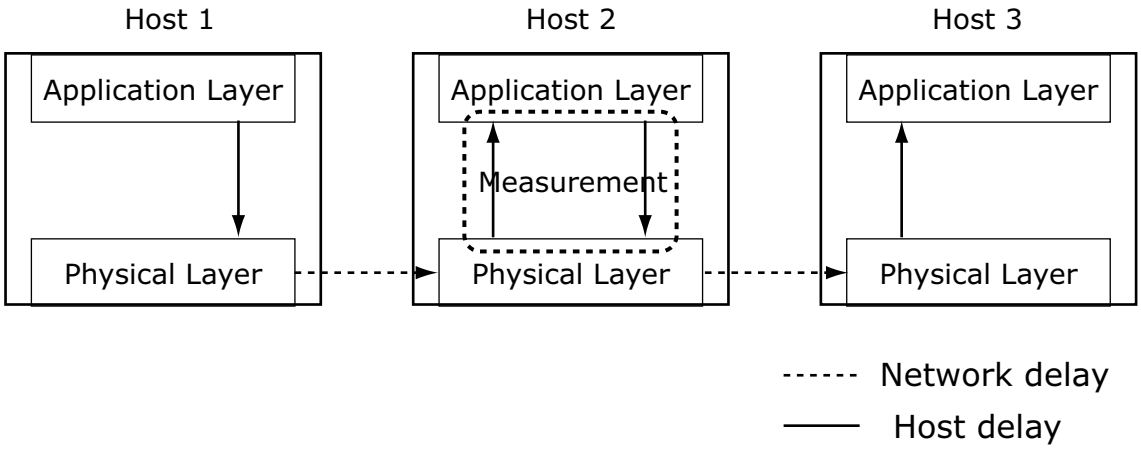


図 5–2: 端末内遅延測定実験

般的な PC より構成される ALM を用いてコンテンツ配信を実現するためには，端末内の遅延より，各端末間のネットワーク遅延を考慮する必要があることが分かる．以上の結果を踏まえ，本論文では，ALM ツリーを構築する際，各端末間のネットワーク遅延のみを考慮する．

5.3　クラスタリングを用いた ALM ツリー構築手法

5.3.1　提案手法の概要

提案手法では，隣接端末同士でクラスタリングを構成し，クラスタリングを単位に，集中型アルゴリズムを用いて，ALM ツリーを構築する (図 5–3).

これにより，参加端末の停止・離脱，物理ネットワーク状況の変化に追従でき，良好な品質を維持したマルチキャストツリーが構築できると考えられる．各クラスタのヘッドが，集中型アルゴリズムを実行する中央端末に相当する．クラスタを構成することで，集中型アルゴリズムによるツリー構築が可能となる一方，各端末が実行するプロービングの範囲を各クラスタ内に閉じ込めることができ，コントロールオーバーヘッドを一定に保つことも可能となる．

本提案方式のツリーの構築では，前章で挙げた集中型アルゴリズムと分散型アルゴリズムを融合させたハイブリッド型を用いる．すなわち，セッション初期状態では，各端末は分散型アルゴリズムを用いてツリーを構築し，端末数がある値を超えた時点で，クラスタを形成し，集中型に移行する．クラスタヘッドが離脱・停止し，集中型アルゴリズムを続行できない場合は，クラスタヘッドの再選択が終了するまで分散型で動作する．

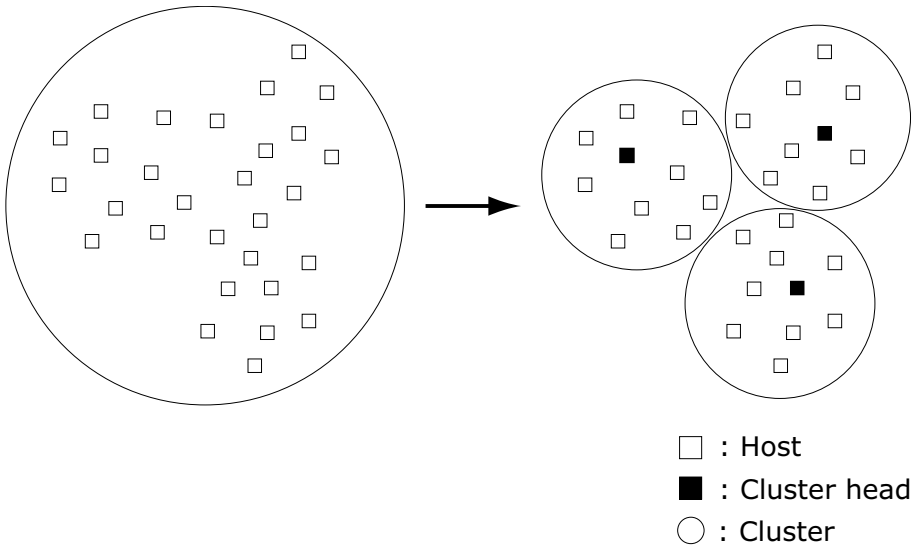


図 5–3: クラスタリングを用いた ALM 構築手法

5.3.2　予備実験 (クラスタを単位とするツリー構築アルゴリズムの有効性の調査)

各クラスタに属する端末間情報のみの局所的な情報を用いてツリーを構築するため，全ての端末間情報を用いてツリーを構築した場合と比較して，準最適なツリーとなるはずである．そこで，最適解であるツリーの品質とクラスタを用いて構築したツリーの品質との間にどの程度の性能差が存在するのか，ツリーの品質比較実験を行う．

評価方法

シミュレーションは，次のように行った．

構築する木は，Minimum Spanning Tree とした．まず Waxman 法 [60] を用い，ノード数 5000 のフラットなランダムトポロジーを構築した．トポロジジェネレータとしては，BRITE[41] を用いた．次に，5000 ノードの中から，ランダムに N ノード選択し，それを端末とした．したがって，残り $5000 - N$ ノードは，ルータとなる．そして，以下に示す 2 項目に関して，1) ホスト全体で MST を構築した場合，2) クラスタを単位として MST を構築した場合，3) 端末間をランダムに接続する全域木を構築した場合 (項目 1 のみ)，について測定を行った．

項目 1 ツリーコスト

項目 2 ツリー構築までに要する時間

項目 1 は，クラスタを単位として MST を構築した場合に，全体の端末間情報を考慮に入れて算出する最適な MST と比較しどの程度の性能劣化が引き起こされるのかを，項目 2 は，計算コストがどの程度削減されているか，についてそれぞれ表している．計算コストの削減は，変化により素早く追随できるツリーの構築につながる．

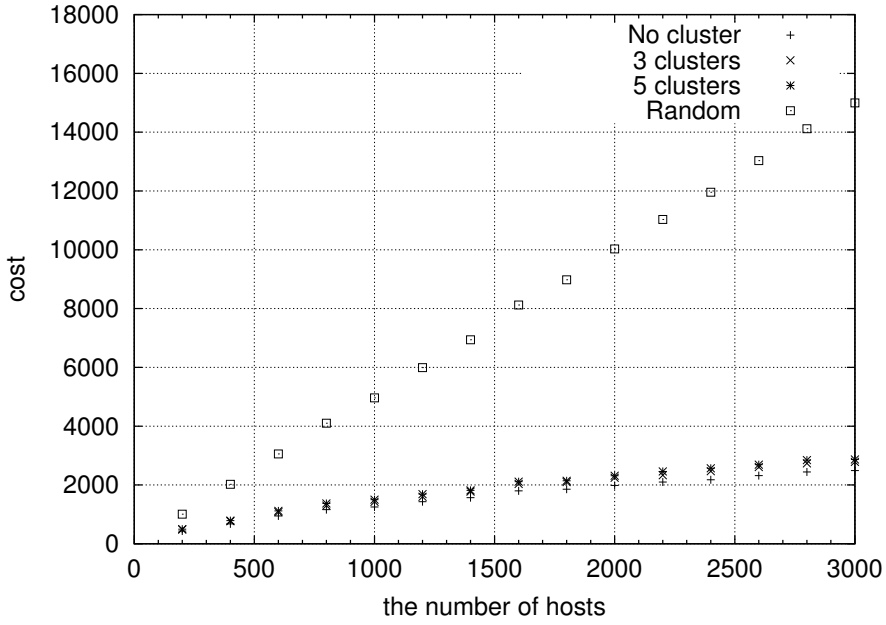


図 5–4: 端末数とツリーコストの関係

MST は，端末間の遅延をコストメトリックとして，Prim 法 [8] を用いて構築した．クラスタを単位としての MST の構築は，クラスタ内 MST とクラスタ中心同士の MST を組み合わせることで行った．クラスタを単位として MST を構築した場合において，クラスタ構築時間はツリー構築時間には含まない．(実行環境 - CPU: Pentium III 700MHz, RAM: 256MB, OS: Windows XP, Java 環境: JDK1.4.0)

評価結果

図 5–4 に端末数とツリーコストの関係を，図 5–5 に端末数とツリー構築時間の関係を，図 5–6 にクラスタ数とツリーコストの関係，図 5–7 にクラスタ数とツリー構築時間の関係をそれぞれ示す．

図 5–4 より，クラスタを単位として構築した MST は，全端末間情報を用いて構築した MST と比較して，ほとんど性能差がないことが分かる．また，図 5–5 より端末全体で MST を構築する場合と比較して，より短時間で MST を構築していることが分かる．すなわち，クラスタリングを用いて，MST を構築することにより，ある程度の性能を持つ MST をより素早く構築することが可能になることが分かる．図 5–6 および図 5–7 の結果から，クラスタ数を大きくすることで，ツリーコストの増加，構築時間の減少が確認できる．クラスタ数の増加により，各クラスタに参加数する端末は減少するため，より局所的な情報のみを用いてのツリー構築が行われるからである．

以上の結果より，クラスタリングを用いてツリーを構築した場合でも，全端末間情報を用いて構築した最適なツリーの品質と比べ，ほぼ遜色がないということが明らかになった．

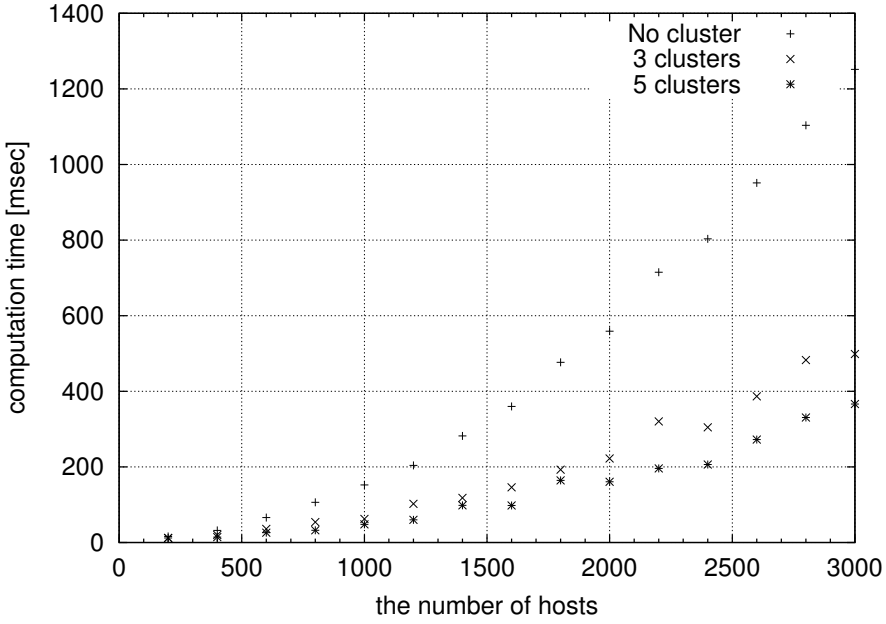


図 5–5: 端末数とツリー構築時間の関係

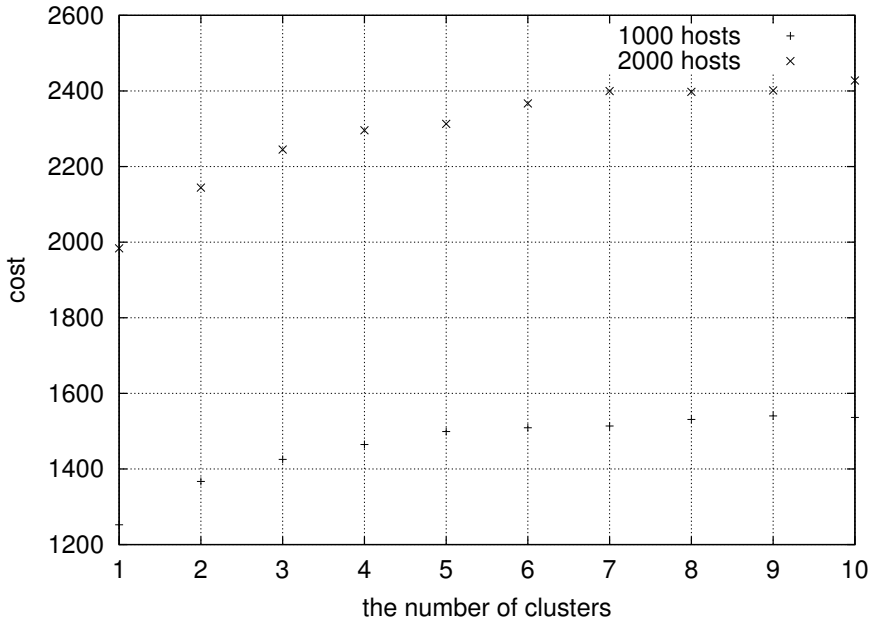


図 5–6: クラスタ数とツリーコストの関係

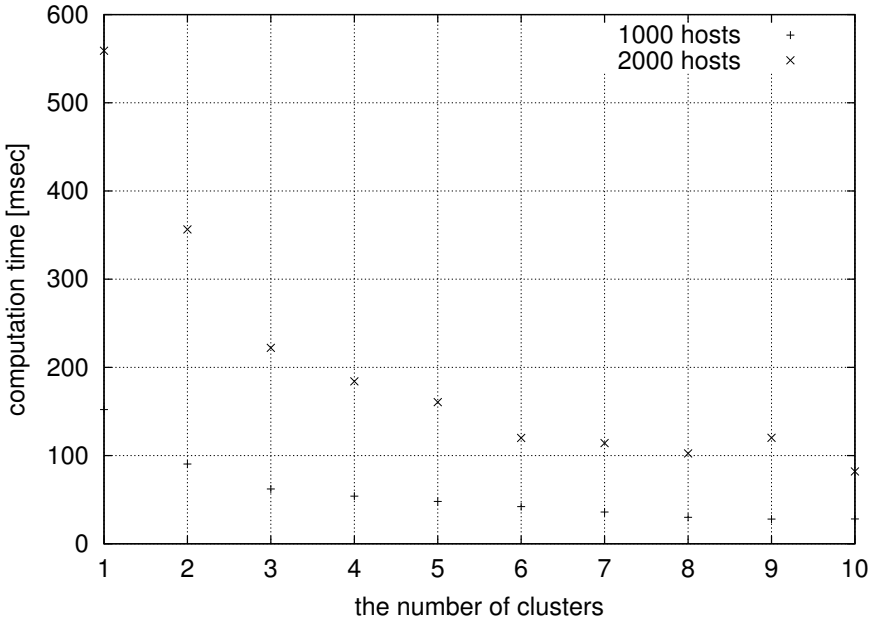


図 5–7: クラスタ数とツリー構築時間の関係

この結果を踏まえ，以下では，クラスタリングを用いた ALM ツリー構築アルゴリズムの詳細について述べる．

5.3.3　提案手法を用いた ALM アーキテクチャ

本論文では，構築するツリーとして，Source specific tree を想定する．Source specific tree は，ある 1 つのノード (ルートノード) を起点としたツリーである．Source specific tree としては，4.5 で挙げた次数制約付き半径最小全域木 (MDDBST)，半径制約付き次数バランスを有する全域木 (BDRBST)，次数制約付き平均遅延最小有向全域木，次数制約付き最大遅延最小有効全域木などが望ましい．ここでは，Source specific tree として，次数制約付き平均遅延最小全域木を対象とする．ただし，上記に挙げた木以外でも，1) ビデオ会議やネットワークゲームなどの複数のソースとなるノードが存在する場合に適した木，2) 共有木，を構築する際，提案手法内で用いるツリー構築アルゴリズムを変更するだけで，それらのツリー構築に応用可能である．

以下では，まず，クラスタ構成法について述べ，次にセッションへの参加，ツリーの構築，維持手順を示す．

クラスタ構成法

クラスタ構成は，以下のように行う．

1. ルートノードを基点としたツリーを構築する．ツリー構築アルゴリズムついては後述

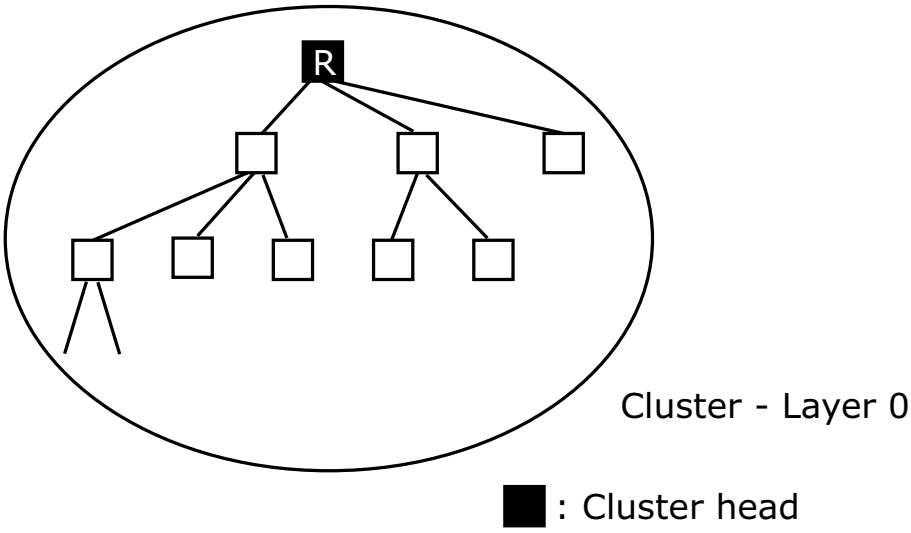


図 5–8: ルートノードをクラスタヘッドとするレイヤ 0 のクラスタ

する．これはルートノードをクラスタヘッドとしたクラスタが 1 つ生成されている状態である．このクラスタは，レイヤ 0 に属する (図 5–8).

2. 参加端末数が一定数を上回った時点で，クラスタのスプリットを行う．クラスタのスプリットでは，まず，ツリー上で現クラスタヘッドの子ノードが新しく生成されるクラスタのヘッド候補 (図 5–9，ノード 1，2，3) として選択される．

   子ノードがクラスタヘッド候補として選択される理由については，後述する．

   各クラスタヘッド候補のうち，下位レイヤに属するクラスタのヘッドではないノードであり，かつツリー上でそのノード以下に一定数の端末数が存在する場合は，そのノードは，新しいクラスタのヘッドと正式に決定される．

   クラスタヘッド候補が下位レイヤに属するクラスタのヘッドである場合 (図 5–10，ノード 1)，あるいはヘッド以下の端末数が一定数存在しない場合 (図 5–9，ノード 3) は，最もネットワーク距離的に近い他のクラスタヘッド候補の下に移動する (図 5–11，ノード 3 がノード 2 の下に移動した場合).

   下位レイヤに属するクラスタのヘッドであるノードを新しいクラスタのヘッドとして認めないことで，クラスタのスプリットの際に，各ノードが属するクラスタ数が 3 以上になることを防ぐ．これは，各ノードのコントロールオーバーヘッドを制限することにつながる．また，ツリー上でヘッド以下の端末数が一定数以下のものは，新しいクラスタのヘッドとなることを制限することで，生成されるクラスタ内の端末数をある値以上に保つことができる．なぜなら，生成されるクラスタ内の端末数があまりに少ない場合，所属端末数が少数のクラスタが多数生成されるため，品質の良いツリーを構築することができないからである．また，これにより各クラスタ内の端末数のバランスをとることもできる．

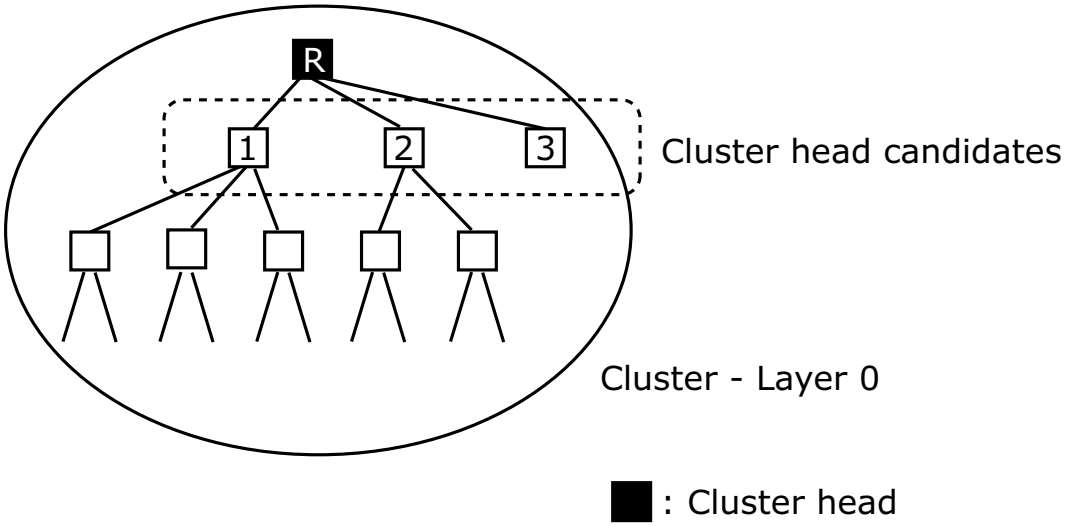


図 5–9: 新クラスタのヘッド候補選択

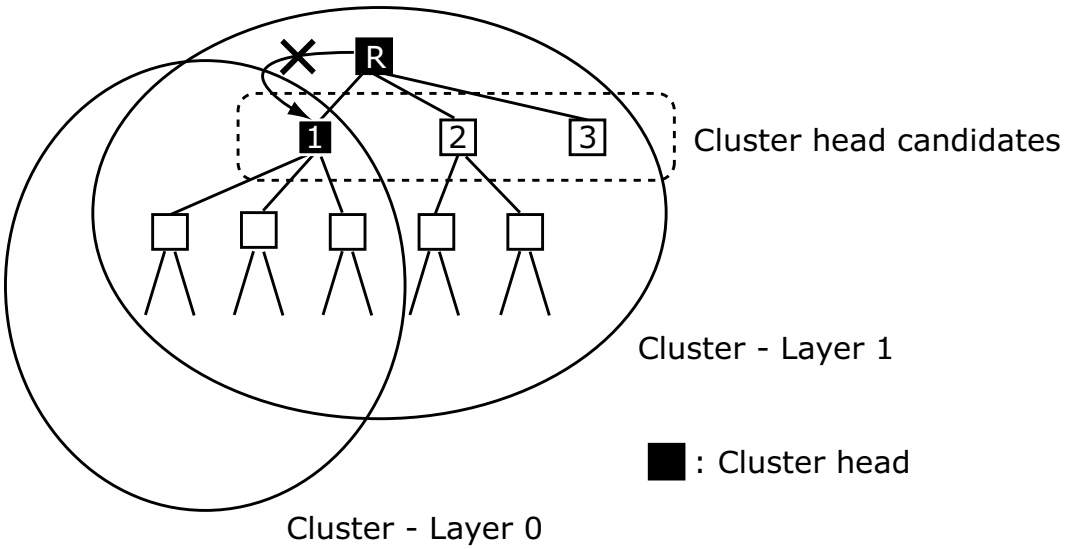


図 5–10: クラスタヘッド候補が下位レイヤに属するクラスタのヘッドである場合

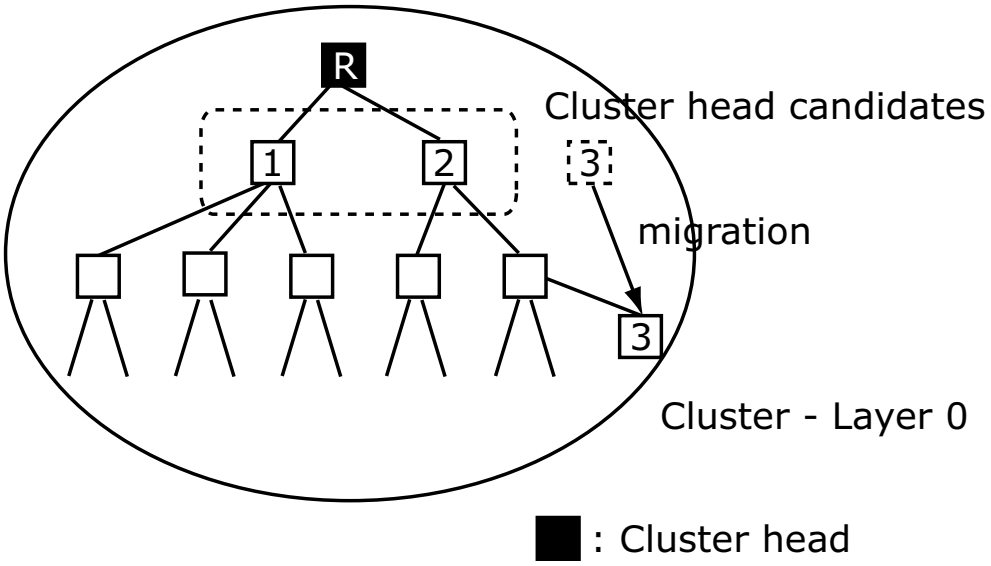


図 5–11: 他のクラスタヘッド候補下への移動

ルートノードは，レイヤ 0 のクラスタからは離脱し，新しく生成されたクラスタヘッドと共にレイヤ 1 に属するクラスタを新たに構成する (図 5–12)．したがって，レイヤ 1 が最上位レイヤとなる．また，最上位レイヤのクラスタヘッドは常にルートノードが担う．

3. 最上位レイヤ (レイヤ L_h) に属するクラスタをルートノードがスプリットする場合は，新しい生成されたクラスタヘッドと共に，ルートノードは最上位レイヤ数を L_{h+1} とし，レイヤ L_{h+1} に属するクラスタを構成する．ルートノードは，レイヤ L_h のクラスタからは離脱する (図 5–12 ルートノードは，レイヤ 0 のクラスタから離脱している)．新しく生成されたクラスタは，レイヤ L_h に属する．

   その他のレイヤ (レイヤ $L_i|i = 0, \cdots, (highest_layer - 1)$) に属するクラスタがスプリットを行う場合，新しく生成されたクラスタは，レイヤ L_i に属する．この際，スプリット対象のクラスタ (レイヤ L_i) のヘッドは，レイヤ L_{i+1} のクラスタにも属しているが，スプリットを行う際，レイヤ L_i からは離脱する．新しく生成されたクラスタのヘッドは，レイヤ L_{i+1} に参加する．

   したがって，レイヤ L_i のクラスタ内のノードは，L_{i-1} のクラスタを持つノードと L_i のみに属するノードが存在する．また，スプリットを行うクラスタヘッドは，下位レイヤからの離脱を行う．

4. クラスタ内の端末数が，端末の離脱などにより一定値を下回った場合は，クラスタ (レイヤ $L_i|i \in layer, i \neq highest\_layer$) のマージが行われる．この際，マージを行うクラスタのヘッドが属する上位レイヤ L_{i+1} のクラスタ内のノードのうち，レイヤ L_i のクラスタヘッドであるノードをマージ対象クラスタヘッドと呼ぶ (図 5–13)．

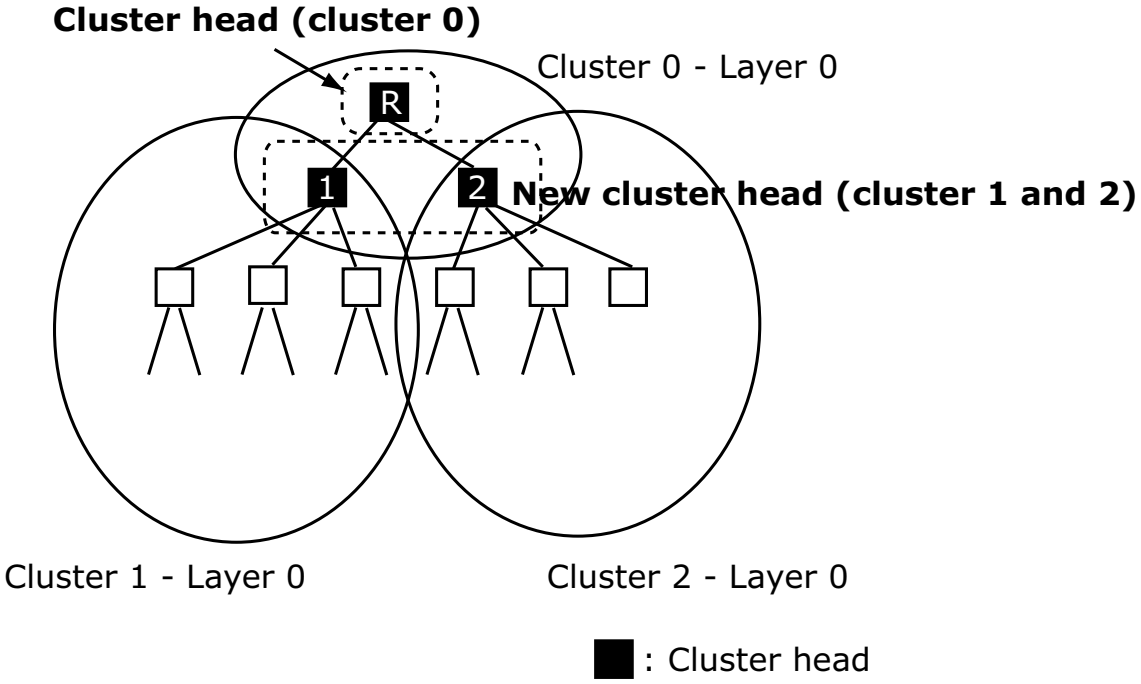


図 5–12: クラスタ分割

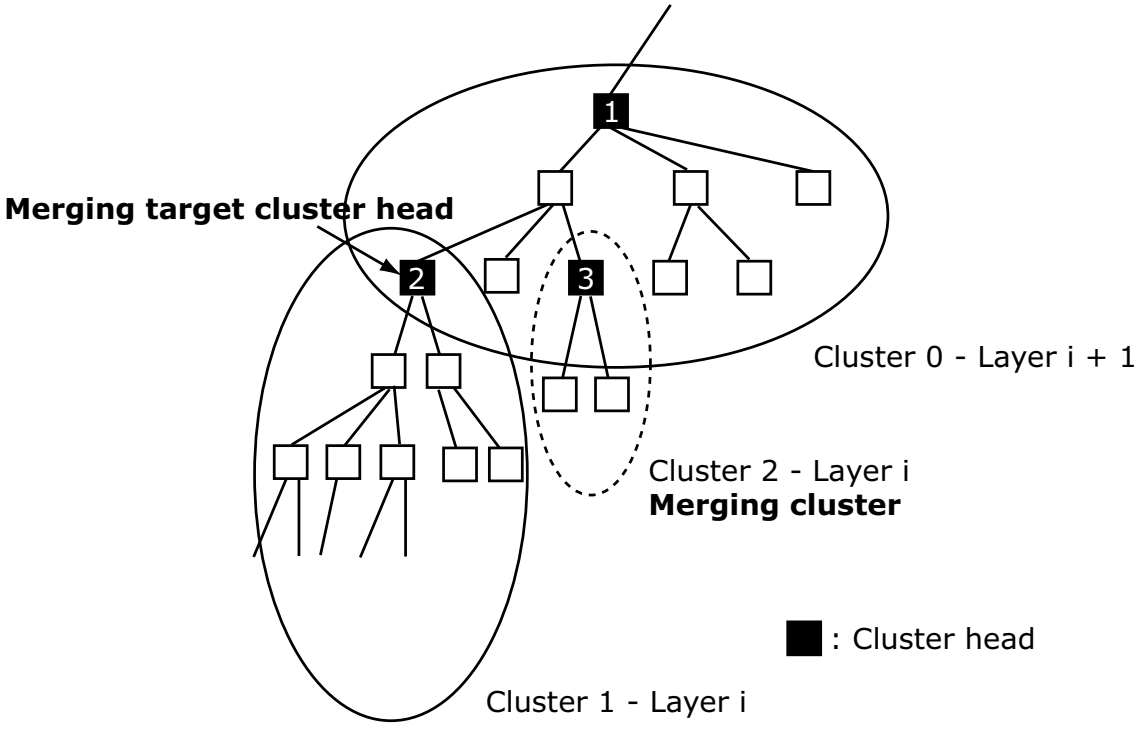


図 5–13: クラスタ合成

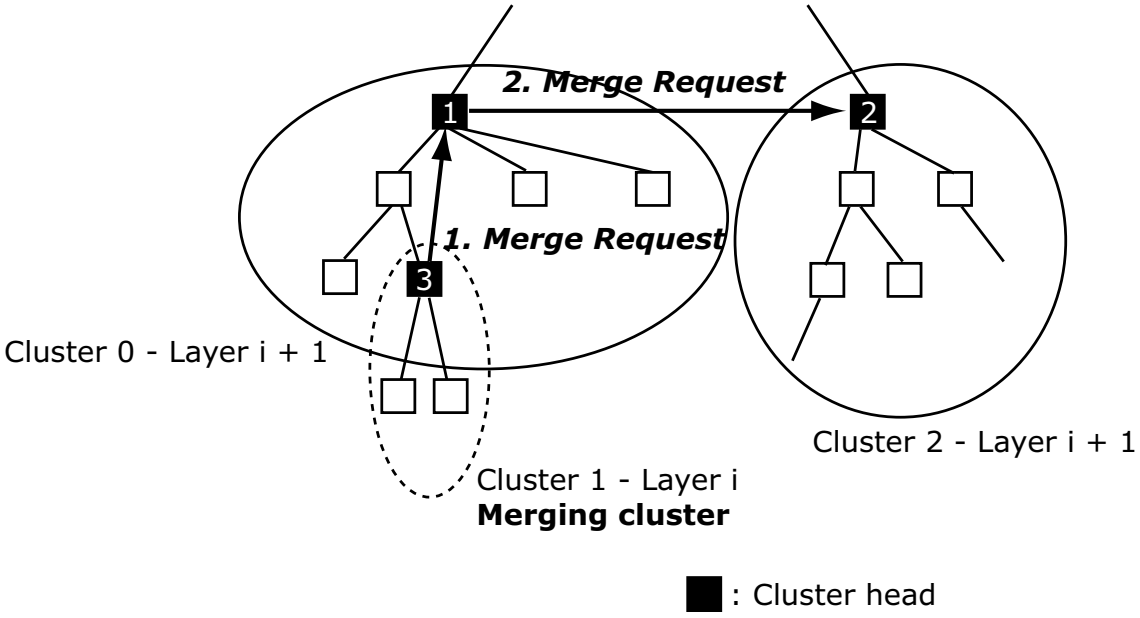


図 5–14: 上位レベルでのクラスタ合成リクエスト

最も近いマージ対象クラスタヘッドに対して，マージリクエストを行うことで，クラスタのマージを行う．レイヤ L_{i+1} のクラスタ内のノードは，L_i のクラスタを持つノードと L_{i+1} のみに属するノードが存在するため，レイヤ L_i のクラスタのマージあるいは，レイヤ L_i のクラスタを持つノードが離脱が頻繁に生じた場合，マージ対象クラスタが存在しない可能性がある．その場合は，レイヤ L_{i+1} のヘッドに対して，上位レベルでのマージを要求する (図 5–14).

クラスタヘッドの子ノードを新たなクラスタのヘッド候補とする理由を説明する．一般的なツリー構築アルゴリズムでは，よりツリーの上位に存在するノードは，下位に存在するノードと比較し，帯域が広く，また演算能力，安定性の面で優れた端末である．なぜなら，ツリー上位に帯域が広く，安定性のある端末を配置することで，末端ノードまでの遅延が短く，安定性のあるツリーを構築できるからである．

クラスタヘッドの条件としては，広帯域，演算能力，安定性を保持していることが挙げられるが，上記条件は，ツリー上位ノードに求められる条件と一致する．つまり，ツリー上位に存在するノードは，クラスタヘッドとなる条件を満たしているノードであるとも言える．したがって，クラスタヘッドの子ノードを新たなクラスタヘッド候補として選択した．また，このようにクラスタヘッド候補を選択することで，容易に新しいクラスタヘッドを絞り込むことができ，かつクラスタ生成に伴うデータツリーへの変更を極力抑えることが可能となるなどの利点もある．

提案手法の留意点

現在のクラスタヘッド (レイヤ L_i) の子ノードが新しいクラスタのヘッドに選択されるためには，常に現在のクラスタヘッドが上位レイヤ L_{i+1} のクラスタでは，末端ノードに存在しなければならない．すなわち，本提案手法では，クラスタ毎にツリーが改善されるが，その際，所属するクラスタよりレイヤが低いクラスタのヘッドであるノードは，常に末端ノードとなることを保証する必要がある．

クラスタ数決定法

クラスタサイズについて，以下のように定式化できると考えられる．

C_s ＝ f(Control_traffic, Computation_cost，クラスタリカバリに伴うコスト，ヘッドノードの演算能力および安定性)

Control_traffic は，端末間の帯域を占めるコントロールメッセージ，プロービングパケットの比率，Computation_cost は，集中的アルゴリズムによりツリーを構築する際の計算コストを示す．クラスタの機能は，クラスタヘッドの離脱により，停止する．そこで，クラスタヘッドの再選択，クラスタリカバリが必要となるが，その際，クラスタ内の端末数が多くなればなるほど，これらを実現するために要する時間が長くなる．したがって，許容できる時間で再構成可能なクラスタサイズに制限する必要がある．ヘッドの演算能力および安定性は，安定したサービスを提供するために考慮に入れる必要がある．ヘッドが離脱・停止を繰り返す場合，常にクラスタ再構成の必要が生じるからである．具体的な関数 f については，今後の課題である．

セッションへの参加およびツリー構築・維持手順

以下に，各端末のセッションへの参加およびツリーの構築・維持手順を示す．

1. セッションへの参加

   HMTP[33] および NICE[46] のアルゴリズムを採用する．参加端末は，最上位レイヤから順にリクエストを送信し，必ずレイヤ 0 に所属するクラスタに参加する．

2. ツリーの構築

   次数制約付き平均遅延最小全域木の構築は，NP 完全問題である [51]．したがって，ここでは，図 5–15 に示す greedy algorithm に基づくアルゴリズムを用いた．まず，ルートノードの子ノードを決定する．子ノードの決定は，現在親ノード候補であるノード (この場合は，ルートノード) との距離に各ノードの次数の逆数を乗じた値を，まだ親ノードが決定していないノード毎に算出し，より小さい値を持つノードから順に子ノードと決定する．そして，決定した子ノードは，順次親ノード候補となる．

ノード間距離にノードの次数の逆数を乗じることで，より短い遅延およびより大きな次数を持っている端末がルートより順次選ばれる．

Procedure : $CreateTree(r)$
$S \leftarrow (Nodes - \{r\})$
$parent \leftarrow r$
$parentCandidateList \leftarrow$ FIFO List
$i \leftarrow 1$

for $(j \leftarrow 2$ to $N)$ do
　$SortedS \leftarrow$ Sort S in increasing order of $((dist.$ from $parent) \times \frac{1}{degree})$
　$k \leftarrow 0$
　while $(k < parent.DegBd)$
　　$parent.NewChild = SortedS[k]$
　　$SortedS[k].parent = parent$
　　$potentialParentList.add(SortedS[k])$
　　$S \leftarrow (S - \{SortedS[k]\})$
　　$i++$
　　$k++$
　end while

　if $(i == N)$
　　break
　endif

　$parent = potentialParentList.pop()$
end for

図 5–15: ツリー構築アルゴリズム

しかし，上述した通り，本提案手法は，ツリー構築アルゴリズムには依存しないため，図 5–15 に挙げたアルゴリズム以外のツリー構築アルゴリズムでも適用可能である．

3. ツリーの維持

ツリーは，クラスタを単位に維持される．クラスタ生成法は，上述した通りである．クラスタを単位に，集中型アルゴリズムでツリーを構築する．クラスタヘッドに何らかの障害がある場合は，分散型アルゴリズムに切り替える．

5.3.4　最適ホスト選択手法との連携

上記参加手法において，各端末は，最上位レイヤからレイヤ 0 に所属するクラスタに接続されるまでリクエストを繰り返し送信し続ける．したがって，より参加端末数が多くなればなるほど，ツリー上位に位置する端末は多数の参加リクエストおよびリクエスト送信先決定の際の RTT 測定に用いられるプロービングパケットを受信する．また，ツリーの深さも長くなるため，参加処理が完了するまでかなりの時間を要するようになる．この問題を解決するために，3.4 で提案した最適ホスト選択手法を用いることが可能である．具体的には，セッションへの参加時，各端末は，最適ホスト選択手法で提案した手法を用いて，それぞれの位置ベクトルを取得する．そして，その位置ベクトルを用いて参加リクエスト先を決定するのである．これにより，多数のプロービングパケットの送信を抑制することができる

5.3.5　従来手法との比較および提案手法の優位点

ここでは，HMTP および NICE と提案手法の比較を行う．

1. HMTP

   4.4.3 において，HMTP の問題点として，各ノードがそれぞれ最適化を実行するため，ツリーの最適解に収束するまでには時間がかかり，また，構築されるデータツリーの品質も，ツリーの最適解と比較した場合，かなり劣化したものとなる可能性があることを指摘した．これは，ツリーの改善を全て分散型アルゴリズムで行うことによって生じる．一方，本提案手法では，クラスタを単位とした集中型アルゴリズムにより，ツリーの改善を行えるため，HMTP と比較し，ツリーの構築も短時間でまたより良い品質のツリーが構築できると思われる．

   ただし，本提案手法を HMTP プロトコルに対しても応用可能であると考えている．具体的には，HMTP ではランダムに端末を選択し，プロービングを行うが，この際，プロービング対象となる端末をコントロールすることで，クラスタを構成する．これは，あるノード p が，ツリー上で自ノード p 以下，N レベル (ツリーの深さ) に属するノード群に，それらノード群のリストを送信することで可能となる．これにより，p を起点としたレベル N 個までのノードを含む，クラスタが構成されると言える (図 5–16).

2. NICE

   NICE も本提案手法同様クラスタを構成する．しかし，4.4.3 で指摘した通り，NICE では，より上位レイヤに属するノードほど要求帯域が大きくなる．また，クラスタサイズが参加端末の使用可能帯域により決定してしまうため，クラスタサイズは基本的に，小さくなる．

   一方，本提案手法では，各ノードが所属するクラスタ数は，最大 2 であり，また各クラスタヘッドが，管理するクラスタの全てのノードにデータパケットを送信するのではなく，ヘッドを基点としたツリーを構築するため，クラスタ数をより大きくできる．

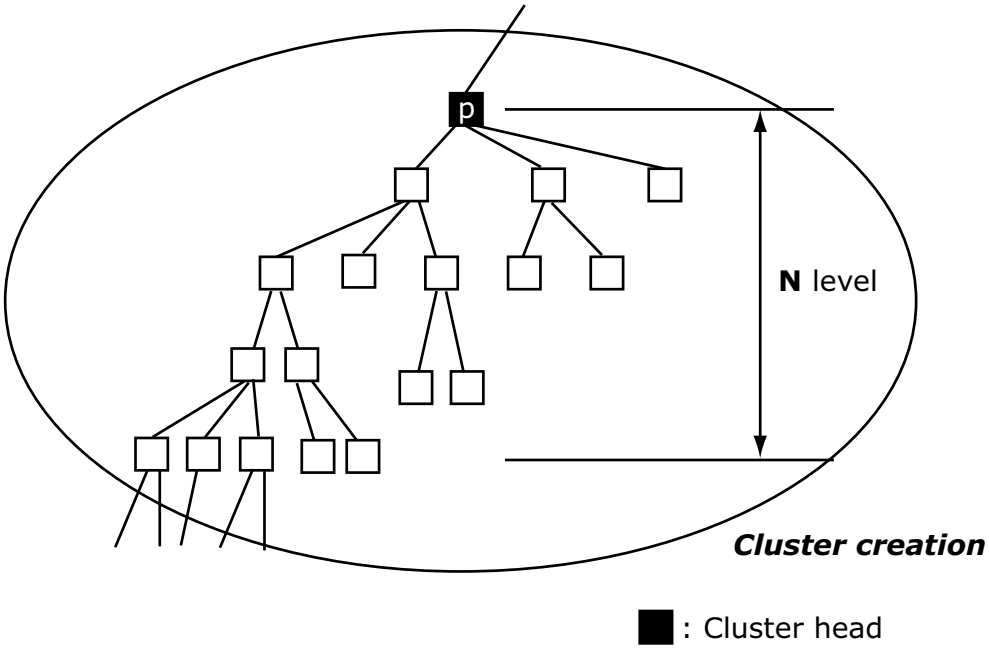


図 5–16: HMTP へのクラスタ適用

これにより，ツリー構築により多くの端末間情報を用いることができ，構築されるツリーの品質は向上する．

ただし，NICE は，共有木としての利用を想定しているため，ツリー品質の比較を行う際には，本提案手法を用いて，次数制約付き平均遅延最小全域木ではなく，次数制約付きの MST を構築する必要がある．

5.4　提案手法性能評価実験

5.3.2 では，端末群が全て参加している状態でクラスタリングを用いた場合および全端末間情報を用いた場合のツリーの品質に関して比較を行った．すなわち，端末の変化がない静的なシナリオでの実験であった．ここでは，5.3.3 で提案した ALM プロトコルを実際にシミュレーション上で実装し，各端末が時間ごとに参加するという動的なシナリオにおいて，本提案手法により構築したツリーと全端末間情報を用いて構築したツリーのツリーの間にどの程度の性能差が存在するのかを確認する実験を行った．また，クラスタを構成する端末数の変化がツリー構成にどのような影響を及ぼすかについても比較検討を行った..

5.4.1　評価手法

シミュレーションは，次のように行った．

Waxman 法を用い，ノード数 3000 のフラットなランダムトポロジーを構築した．トポロジの各リンクの平均遅延は 0.34 秒，各ノード間の平均ホップ数は 5.6，各ノード間の平均遅

表 5–2: シミュレーションにおける各端末のセッション参加時間

端末 ID	参加処理開始時間
0 - 499	10 秒 - 5000 秒 まで 10 秒毎
500 - 999	8000 秒 - 10495 秒 まで 5 秒毎
1000 - 1999	13000 秒 - 22900 秒 まで 10 秒毎

延は，1.46 秒である．次に，3000 ノードの中から，以下に示す表に従って，各端末のセッションへの参加処理を開始させた．

端末 499 - 500，999 - 1000 間の間の時間は，端末の参加処理を一時停止させ，ツリーの安定性を見るために設定した．そして，配信ツリーの品質を調べるため，送信端末 (端末 ID 0) から，10 秒から 100 秒毎に 230 回，23010 秒までデータパケットを送信した．

5.5　本章のまとめ

本章では，まず，ALM を構成する端末内におけるパケットフォワーディングの際の遅延を測定する実験を行い，各端末内での遅延は，ほとんど存在しないことを確認した，したがって，ツリー構築時は，端末内遅延ではなく，端末間遅延 (ネットワーク距離) を考慮する必要があることが分かった．

次に，前章で挙げた各種 ALM プロトコルの問題点および ALM ツリー構築アルゴリズムの考察を踏まえ，クラスタリングを用いた ALM ツリー構築手法を提案した．提案手法では，基本的にクラスタヘッドを中心としたクラスタ内集中型ツリー構築アルゴリズムを用いて，ツリーを構築する．一方，クラスタヘッドが停止するなど，集中型アルゴリズムが提供できない場合は，分散型に切り替える．これにより，集中型の利点および分散型の利点を兼ね備えた，セッション参加端末の変化に追従できるツリー構築が可能となる．ただし，提案手法では，クラスタ毎に集中型アルゴリズムでツリーを算出するため，全端末間情報を用いて構築した最適解であるツリーとの間に大幅な性能差が存在する可能性があった．そこで，全端末が参加済みという静的なシナリオで，1) 各端末をクラスタリングし，クラスタ毎にツリーを構築した場合，および 2) 全端末間情報を用いてツリーを構築した場合，とでツリー品質の比較実験を行った．その結果から，両者の間にはあまり品質差が存在しなく，提案手法を用いることでより少ない計算コストでツリーを構築できることが分かった．

次に，提案手法を，ALM プロトコル内で動作させるためのクラスタ構成法および，提案手法が動作しているセッションへの参加手順，ツリーの構築および維持手順を示した．また，参加時における端末間距離測定に伴うコントロールパケットの増大が特にツリーの上位に位置する端末では深刻になるということを指摘し，それを解決する手法として，3 章で提案した最適ホスト選択手法が適用できるということを述べた．

そして，提案手法を実装した ALM プロトコルを実際にシミュレーション上で動作させ，各端末が時間ごとに参加するという動的なシナリオにおいて，本提案手法により構築したツ

リーと全端末間情報を用いて構築したツリーのツリーの間にどの程度の性能差が存在するのかを確認する実験を行った.

シミュレーション実験により本手法の有効性が確認できた.

第6章

結論

6.1 総括

本研究では，未だ実現されていない規模のコンテンツ配信実現に向け，いくつかの問題点を明らかにした上で，それを解決するための手法として，1) 最適ホスト選択手法，2) クラスタリングを用いたALMツリー構築手法に関して提案を行った．また，シミュレーションを通して本手法の有効性を示した．以下，各章の内容を述べる．

第1章では本研究の背景および目的，本論文の構成について述べた．

第2章では第1章で述べたことを受け，コンテンツ配信アーキテクチャとして，1) サーバクライアント型アーキテクチャ，2) Peer to Peer (P2P) 型アーキテクチャについて述べた．それを受け，大規模コンテンツ配信を実現する上での課題として，1) 多様な端末装置への対応手法，2) サーバ (コンテンツ送信端末) への同時アクセス可能数増加手法，3) ボトルネックリンク回避手法，が必要不可欠であることを指摘した．そして，それら手法を構成可能な技術として，IPマルチキャスト，CDN, P2P (ALM) を取り上げ，それらの特徴および問題点について論じた．IPマルチキャストについては，アーキテクチャ，構築されるツリーの種類を説明した．また，IPマルチキャストの問題点として，1) マルチキャストグループ作成にかかわる認証，受信者認証，送信者認証などのグループ管理問題，2) マルチキャストルータへの攻撃，データの一貫性の保証などのセキュリティ問題，などを挙げ，IPマルチキャストの普及が進まない要因について論じた．CDNおよびP2Pのような分散キャッシュ型コンテンツ配信アーキテクチャについては，1) コンテンツ発見手法，2) 最適ホスト選択手法，が重要であることを指摘した．また，ALMでは，ツリーを構成するノードの参加・離脱などの状況変化に追従可能なツリー構築アルゴリズムおよびプロトコルが重要であるということを述べた．

第3章では，分散キャッシュ型コンテンツ配信アーキテクチャにおける最適ホスト選択手法の重要性を受け，インターネット上での各ホストの位置を幾何学空間に射影するRTT予測手法を応用した最適ホスト選択手法の提案を行った．提案手法では，まず，1) Landmarkと呼ばれるホストをインターネット上に複数用意し，2) 各ホストがLandmark間とのRTT測定を行い，その値を基に位置ベクトルを構成する，そして，3) 各ホストの位置ベクトルの相関係数を算出し，その値がより0に近いホストを最適ホストとして選択する，という

ことを行う．従来より提案されている RTT 予測手法をそのまま最適ホスト選択に適応する場合と比べ，相関係数の比較を行うことにより，従来手法のボトルネックを解決することが可能となった．提案手法の性能評価としては，各端末より構成される P2P ネットワークを基とした以下のようなシミュレーション実験を行った．各コンテンツを各端末に配置した上で，1) 各端末が目的のコンテンツ群をダウンロードする際に要する時間，2) ネットワーク上に流れる全トラヒック量，に関してそれぞれ，a) ランダムに最適ホストを選択した場合，b) すべてのホストに対してプロービングを行い，最適ホストを選択した場合，c) 提案手法を用いて最適ホストを選択した場合，で比較を行った．実験結果から提案手法の有効性を確認した．

第 4 章では，大規模コンテンツ配信を実現する上でのマルチキャスト機能の重要性を受け，ALM を用いたインターネット上でのマルチキャスト機能実現に着目した．まず ALM ツリー構築方法について，特にコントロールトポロジおよびデータトポロジについて説明した．次に，ALM 手法は，これらトポロジの構築方法の違いにより，1) 集中型アプローチ，2) 分散型アプローチ - a) Mesh-first アプローチ，b) Tree-first アプローチ，に分類できることを述べた．そして，ALM ツリーの性能を測る上で用いられる性能評価尺度として，1) ストレス，2) ストレッチ，などを紹介した．従来より提案されている各 ALM プロトコルに関して，上記分類をあてはめた上で，アーキテクチャの詳細および問題点について述べた．最後に，ALM ツリー構築手法に関して，1) 集中型ツリー構築アルゴリズムおよび 2) 分散型ツリー構築アルゴリズム，に分類を行い，従来手法を分類，整理した．集中型ツリー構築アルゴリズム，分散型ツリー構築アルゴリズムの利点および欠点について論じた．そして，集中型アルゴリズム・分散型アルゴリズムを組み合わせたハイブリッド型が有効であるということを論じた．

第 5 章では，4 章で挙げた各種 ALM プロトコルの問題点および ALM ツリー構築アルゴリズムの考察を踏まえ，新たな ALM ツリー構築手法を提案した．まず，ALM ツリー構築手法確立にあたり，各端末内の遅延を考慮する必要があるのかどうかを確かめる実証実験を行った．その結果，端末内遅延はほとんどなく，ツリーの構築は，ネットワーク遅延のみを考慮にいれれば良いことが分かった．次に，本提案手法としてクラスタリングを用いた ALM ツリー構築手法を提案した．ALM では，IP マルチキャストルータと比べ，セッションへの参加，離脱などが頻繁に起こる安定性のない端末が，マルチキャストツリーの構築を行うこととなるため，それら変化に追従できるツリー構築アルゴリズムが要求される．本提案手法では，参加端末群よりクラスタを形成し，クラスタを単位に集中的アルゴリズムを用いるため，より短時間で準最適なツリーへと収束できる．また，端末間で交換されるプロービングパケットの到達範囲もクラスタ内に閉じることとなり，コントロールパケット量を制限しつつ，有益な端末間のネットワーク距離情報を得ることが可能となる．したがって，状況変化に追従できるツリー構築が可能となる．クラスタヘッドが停止したなど，クラスタ内において，集中的アルゴリズムによるツリー構築が提供できない場合は，分散的アルゴリズムに切り替えることで，ツリー構築を停止することなく，動作しつづけることが可能である．

提案手法を，ALM プロトコル内で動作させるためのクラスタ構成法，および提案手法が

動作しているセッションへの参加手順，ツリーの構築，維持手順に関しても論じた．また，参加時における端末間距離測定に伴うコントロールパケットの増大が特にツリーの上位に位置する端末では深刻になるということを指摘し，それを解決する手法として，3 章で提案した最適ホスト選択手法が適用できるということを述べた．

本提案手法を実装した ALM プロトコルを実際にシミュレーション上で動作させ，本提案手法を用いて構築したツリーおよび全端末間情報を用いて構築したツリー (最適解となるツリー) の品質比較実験を行った．それより，提案手法で構築したツリーと最適解となるツリーとの間にはほとんど性能差が存在しないことが分かった．したがって，本提案手法を用いることで，ツリーの品質を良好に維持しながら，各端末間で交換されるコントロールパケットの削減および端末の変化に追従可能なツリーが構築できる．すなわち，本提案手法の有効性が確認できた．

第 6 章は結論であり，本論文の総括および今後の課題について述べている．

6.2　今後の課題

以下に今後検討すべき課題について述べる．

6.2.1　最適ホスト選択手法に関する課題

まず，第一に，3.6.4 で述べた，Transit-Stub 型を有するシミュレーショントポロジでの実験である．Waxman 型は，フラットなトポロジであるため，全ての端末が類似した位置ベクトルを有する可能性がある．一方，Transit-Stub 型は，階層構造を有するトポロジであるため，Waxman 型の場合と比較し，より異なる位置ベクトルを有するものと考えられる．したがって，最適ホスト選択の精度が上昇すると考えられる．

次に，各位置ベクトルの時間的な変化も考慮にいれることにより，精度が向上するかどうかの検証である．現在の提案手法では，ある時点での Landmark への RTT を，位置ベクトルとし，それ以降は固定値として用いていた．しかし，ネットワーク状況は，刻々と変化するため，位置ベクトルも適宜変更する必要があると考えられる．その際，過去の位置ベクトルを破棄するのではなく，現在の位置ベクトルとの変化具合も考慮にいれることが可能であると考えられる (図 6–1)．

なぜなら，近接に存在する端末同士では，位置ベクトルの変化も同様な傾向を示すと考えられるからである．

また，本手法のみで最適ホスト選択を行うのではなく，他のメトリック測定と組み合わせることにより，より精度が上昇するものと思われる．すなわち，本手法により最適ホストと識別される上位 N ホストに対して，RTT 測定，帯域測定などを行うのである．実際，いくつかのメトリック測定を組み合わせることで近接ホストをより精度よく発見可能であることが，文献 [55] で示されている．

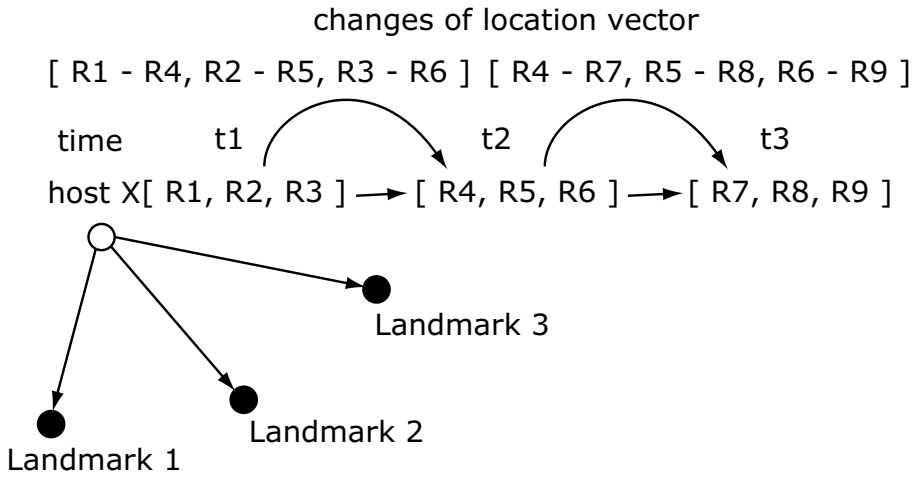


図 6–1: 位置ベクトル変化量の考慮

6.2.2　クラスタリングを用いた ALM ツリー構築手法に関する課題

最適ホスト選択手法に関する課題同様，Transit-Stub 型を有するシミュレーショントポロジでの実験がまず挙げられる．Transit-Stub 型を用いることで，ツリー品質にどのような変化が起こるのか調査する必要がある．

次に，各端末のクラスタ間移動に関する検討が挙げられる．現在の実装では，一度所属したクラスタを端末が離脱し，より最適だと思われるクラスタの下に移動することについてはまったく考慮に入れていない．クラスタ間移動を実現する方法として，以下のような方法が考えられる．

1. 各端末は，自分が属するクラスタ (レイヤ L_i) の上位レイヤ L_{i+1} にあるクラスタ内に存在し，かつレイヤ L_i のクラスタヘッドでもある端末群に対して，定期的にプロービングパケットを送信し，ネットワーク距離 $d_i(i \in targetnodes)$ を測定する (図 6–2).

2. その距離が，現在のクラスタヘッド間距離 d_{head} に比べ，一定値以上小さければ，クラスタ間移動を行う．

クラスタ間移動は，常に一定の閾値を取る方法や，$d_i - d_{head}$ に値を応じて確率的に移動を行う方法が考えられる．特に，後者の方法に関しては，例え，$d_i - d_{head} > 0$ であっても，クラスタ間移動を行うことで，クラスタを単位に構築したツリーが局所解に陥るのを防ぐことができる可能性がある．この考えは，Simulated Annealing (SA) [56] 法における考え方と類似している．ただし，ALM のようにツリーを構成するノードの状態が頻繁に変更される場合，このようなクラスタ間移動を行うことは，よりツリーの安定性を損なう恐れがあり，十分に検討する必要がある．

ALM プロトコルの従来手法との性能比較も行う必要がある．特に，ストレス，ストレッチなど ALM 独自の性能評価尺度での比較も重要である．

提案した ALM プロトコルを実装し，実ネットワークでの実験も重要な課題である．

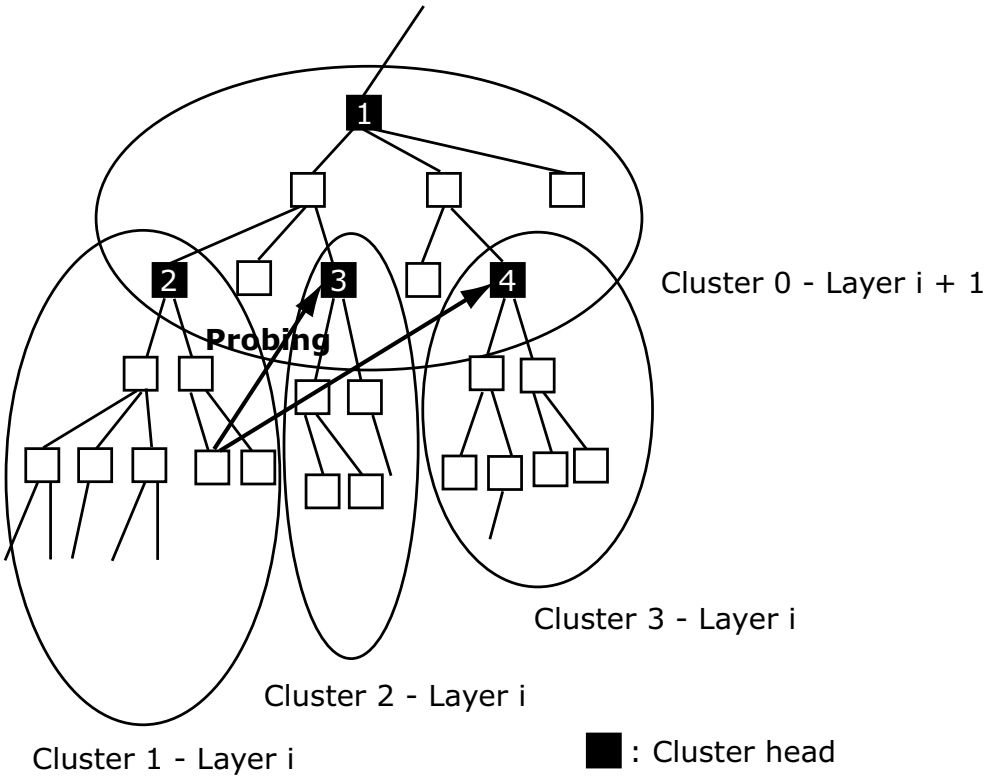


図 6–2: クラスタ間移動の決定

6.3　より大規模なコンテンツ配信実現に向けて

2 章で述べたが，大規模コンテンツ配信の実現には，1) 多様な端末装置への対応手法，2) サーバ (コンテンツ送信端末) への同時アクセス可能数増加手法，3) ボトルネックリンク回避手法，の実現が今後も重要であると考えられる．

2 および 3 を実現する手段として，本研究を通じて提案を行った 1) 最適ホスト選択手法，および 2) クラスタリングを用いた ALM ツリー構築手法が適応できる．しかし，より大規模なコンテンツ配信を実現するためには，これら技術の単体での利用ではなく，本提案手法を含め，IP マルチキャスト技術，CDN 技術 o および様々な端末装置への対応を実現するための技術などを統合した統合コンテンツ配信アーキテクチャ実現が必要不可欠である．そこで以下では，本研究の中では言及をしなかった多様な端末装置への対応手法を実現させるための技術および統合コンテンツ配信アーキテクチャについて論じる．

6.3.1　多様な端末装置への対応手法

ここでは，配信するコンテンツを動画像コンテンツと仮定する．動画像コンテンツを演算能力，ディスプレイのサイズ，ネットワーク帯域が異なる様々な端末装置に対して，配信する場合，MPEG-2, 4 などで想定されている以下のスケーラビリティ機能を用いることが有効である．

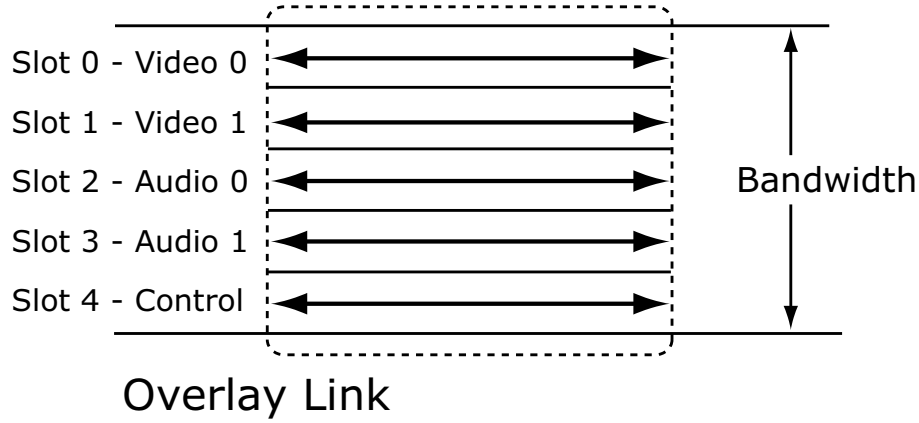


図 6–3: Emma プロトコル

1. SNR スケーラビリティ
2. 空間スケーラビリティ
3. 時間スケーラビリティ

上記スケーラビリティを有するコンテンツを配信することで，各端末が自端末の能力にマッチしたコンテンツを受信することが可能となる．これらスケーラビリティを有するコンテンツの提供手段としては，1) あらかじめスケーラビリティを有するコンテンツを送信端末に保持しておく方法，2) ネットワーク上にトランスコーダ機能付きのルータを配置する方法，3) セッションに参加している各端末がトランスコーダ機能を提供する方法など，が考えられる．スケーラビリティを有するコンテンツの配信手法，すなわち，多階層配信システムでは，拡張層が基本層に依存してしまうということを考慮に入れる必要がある．すなわち，拡張層のみを受信しても，対応する基本層が受信できなければ，復号ができないため，いかに基本層を保護するかということである．基本層保護については，IP マルチキャストおよび Diffserv の組み合わせを中心に様々な研究がなされている [57]．ALM での基本層保護の実現には，Emma[58] など配送するコンテンツ毎に優先度を行う ALM プロトコルが有効であると考えられる．Emma では，各端末が配信するデータの優先度に応じて，動的にどのデータを自ホップの端末に配信するべきかを決定する．具体的には，使用可能帯域を図 6–3 に示すようにスロットに分割し，それぞれに配信する各コンテンツを割り当てる．もし，スロットがフルになった場合は，各コンテンツの優先度に応じてスロットの再割り当てを行う．したがって，基本層の優先度を高めることで，基本層の保護が実現できる．

また，近年，上記，スケーラビリティを有する符号化の他に Multiple Description Coding (MDC) と呼ばれる符号化が注目を集めている [59]．MDC では，スケーラビリティを有する符号化同様の機能を実現できるが，各層に基本層，拡張層の区別がないため，各層毎の優先度に基づく配送の際の保護処理などが必要なくなる (図 6–4).

MDC を用いたコンテンツストリーミングアーキテクチャとして，Cooperative Network (CoopNet) が提案されている [27]．CoopNet では，まずコンテンツを MDC を用いて符号

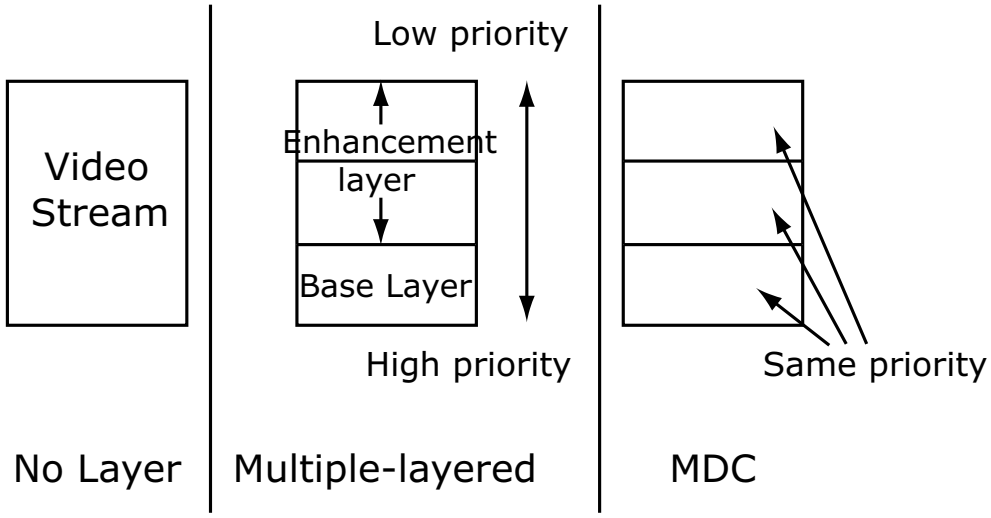


図 6–4: MDC

化し，N 階層に分ける．セッション参加端末間で複数の ALM 配信ツリーを構築し、各ツリーごとに 1 階層分のデータを配信している．これにより，ある配信ツリーが何らかの原因によって，機能を停止した場合でも，その他のツリーを通じて，残りの階層のデータは受信できるため，多少のコンテンツの画質などの品質劣化などは起こるものの，サービス自体は無停止のまま続けることが可能となる．

6.3.2　統合コンテンツ配信アーキテクチャ

より大規模なコンテンツ配信を実現するためには，本提案手法を含め，IP マルチキャスト技術，CDN 技術などを統合した統合コンテンツ配信アーキテクチャ実現が必要不可欠である．ここでは，統合アーキテクチャの IP マルチキャスト，CDN，ALM，および 6.3.1 で述べた多様な端末への対応技術の統合を考える．

IP マルチキャストおよび ALM の連携であるが，2.5.2 で述べたとおり，ALM は，IP マルチキャストのトンネリングプロトコルとして動作させることが可能であり，容易に両技術の連携は可能である．IP マルチキャストは，特にインタードメインでの展開が困難であるが，ALM との連携を用いることで，ドメインをまたいでのマルチキャスト機能提供が実現される．CDN におけるマルチキャスト機能の利用は，エッジサーバ間およびエッジサーバ — 受信者間に適用できる．多様な端末への対応は，エッジサーバおよび ALM を構成する各端末でトランスコーダ機能を実現することで成されると考えている．以上を踏まえ，予測されるコンテンツ配信アーキテクチャ例を図 6–5 に示す．

まず各エッジサーバより構成されるネットワークで，コンテンツが，エッジサーバに接続されたクライアントのリクエストを満たすように分配され，次に各受信者に IP マルチキャストあるいは ALM (図では，ALM 手法として本論文の提案手法を挙げている) を用いて配信される．各端末の能力に合わせて，配信ツリーが構築され，境界端末では，トランスコー

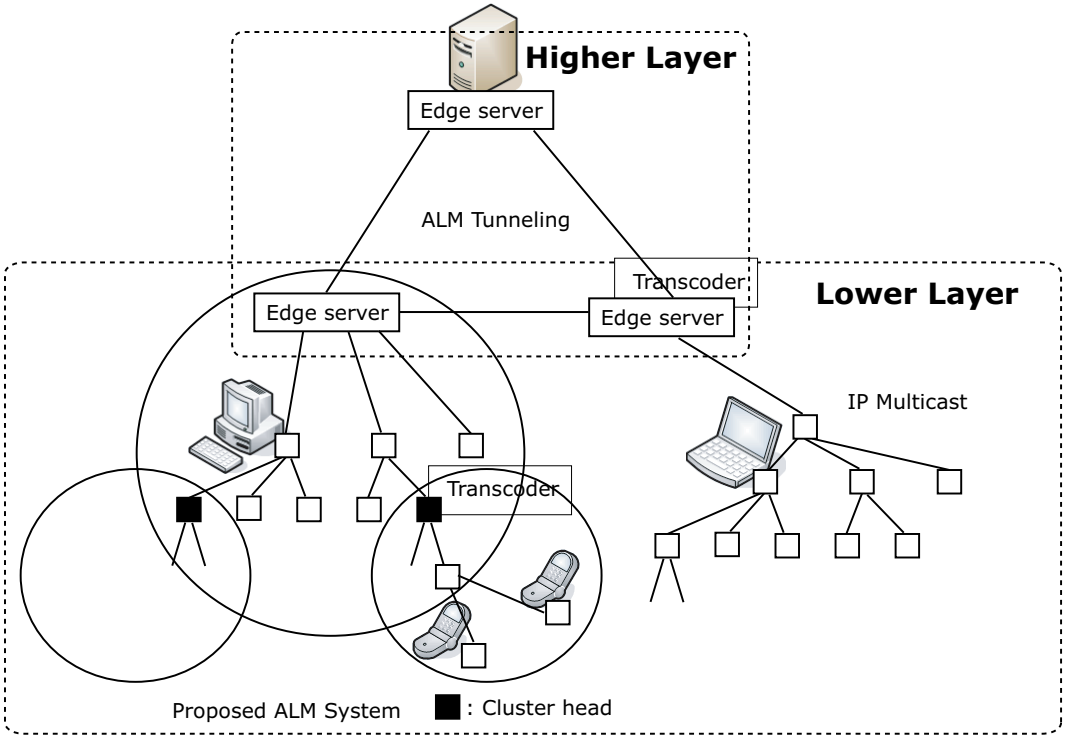


図 6–5: 想定される統合コンテンツ配信アーキテクチャ

ダが動作している．多階層を有するコンテンツの各層は，それぞれの特徴に合わせて，配信される．今後，このようなコンテンツ配信アーキテクチャをいかに低コストで実現するかが大きな課題となるであろう．

謝辞

本研究の機会及び素晴らしい実験環境を与えて下さり，日頃から惜しみなく御指導して頂きました渡辺 裕 教授に心から感謝致します。

貴重な時間を割いて，研究の方向性を御指導頂きました富永 英義 教授に深く感謝いたします.

また, 研究の方向性をはじめ，研究の細部に至るまで，数々の有意義な御意見，御助言を賜わりました亀山 渉 教授に深く感謝致します.

研究の進め方から文章の書き方まで丁寧に御指導下さった平成 13 年度 富永研究室卒 宮澤 敏記 氏にはこの場を借りて心から深く感謝致します.

様々な方向性の指針を与えてくださった渡辺研究室 金田 瑞規氏，中神 央二氏，Abhay Ghatpande 氏に心から感謝致します．また，貴重な御意見，様々な御提案を頂いた AVS ゼミの皆様および富永研究室の皆様に御礼申し上げます.

最後に，私をここまで育てて下さった家族に深く感謝します.

平成 16 年 3 月 15 日

参考文献

(1) 総務省, “インターネット接続サービスの利用者数等の推移【平成 15 年 11 月末現在】（速報）,” 2003.

(2) 青山 友紀, “コンテンツ配信の仕組みとコンテンツ管理の課題,” テレコミュニケーション 4 月号, pp116-121, 2002.

(3) 鍋島 公彰, “ストリーミング CDN ストリーミングシステム (　),” Internet Week 2001, 2001.

(4) M. Handley, C. Perkins, and E. Whelan, “Session Announcement Protocol,” IETF, RFC 2974, Oct. 2000.

(5) M. Handley, and V. Jacobson, “SDP: Session Description Protocol,” IETF, RFC 2327, Apr. 1998.

(6) T. H. Cormen, C. E. Leiserson, and R. L. Rivest, “Introduction to Algorithms,” The MIT Press, McGraw-Hill, 1995.

(7) E. Dijkstra, “A Note on Two Problems in Connection with Graphs, ” Numerical Mathmatics, 1959.

(8) Robert Prim, “Shortest connection networks and some generalizations,” Bell System Technical Journal, Vol.36, pp.1389-1401, Nov. 1957.

(9) J. B. Kruskal, “On the shortest spanning tree of graph and the traveling salesman problem,” Proceddings of the American Mathematical Society, 7, pp.48-50, 1956.

(10) R. Gallager, P. Humblet, and P. Spira, “A Distributed Algorithm for Minimum-Weight spanning trees,” ACM Transaction on Programming Languages and Systems, pp.66-77, Jan. 1983.

(11) P. Winter, “Steiner Problem In Networks: A Survey,” Networks, Vol.17, pp.129-167, 1987.

(12) L. Kou, G. Markowsky, and L. Berman, “A Fast Algorithm for Steiner Trees,” Acta Information, pp.141-145, 1981.

(13) B. Wang, and J. C. Hou, “Multicast Routing and Its QoS Extension: Problems, Algorithms, and Protocols,” IEEE NETWORK, Vol.14, No.1, pp.22-36, 2000.

(14) D. Waitzman, C. Partridge, and S. Deering, “Distance Vector Multicast Routing Protocol,” IETF, RFC 1075, Nov. 1988.

(15) J. Moy, “Multicast Extensions to OSPF,” IETF, RFC 1584, Mar. 1994.

(16) A. Ballardie, “Core Based Trees (CBT) Multicasting Routing Architecture,” IETF, RFC 2201, Sep. 1997.

(17) S. Deering, D. Estrin, D. Farinacci, V. Jacobson, A. Helmy, D. Meyer, and L. Wei, “Protocol Independent Multicast Version 2, Dense Mode Specification,” Work in Progress.

(18) D. Estrin, D. Faricacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, L. Wei, “Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification,” IETF, RFC 2362, Jun. 1998.

(19) Maria Ramalho, “Intra- and Inter-Domain Multicast Routing Protocols: A Survey and Taxonomy,” IEEE Communications Surveys & Tutorials, Vol.3, No.1, 1st Quarter 2000.

(20) C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen “Deployment Issues for the IP Multicast Service and Architecture,” IEEE Network, Vol.1, No. 14, Jan. 2000.

(21) D. Xu, H. Chai, and C. Rosenberg, “Analysis of a Hybrid Architecture for Cost-Effective Streaming Media Distribution,” SPIE/ACM Conference on Multimedia Computing and Networking (MMCN’03), Santa Clara, CA, Jan. 2003.

(22) A. Biliris, C. Cranor, F. Douglis, and M. Rabinovich, “CDN Brokering,” the 6th International Web Caching and Content Distribution Workshop (WCW’01), Jun. 2001.

(23) J. Kangasharju, J. W. Roberts, and K. W. Ross, “Object replication strategies in content distribution networks,” SPIE ITCom Conference on Scalability and Traffic Control in IP Networks , Jun. 2002.

(24) M. Karlsson, and M. Mahalingam, “Do We Need Replica Placement Algorithms in Content Delivery Networks,” the 7th International Web Caching and Content Distribution Workshop (WCW’02), Aug. 2002.

(25) M. Karlsson, C. Karamaolis, and M. Mahalingam, “A Framework for Evaluating Replica Placement Algorithms,” Technical Report HPL-2002, HP Laboratories, Jul. 2002.

(26) I.Lazar, and W. Terrill, “Exploring Content Delivery Networking,” IT Professional Vol.3 No.4, pp.47-49, 2002.

(27) V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai, “Distributing Streaming Media Content Using Cooperative Networking,” ACM Network and Operating System Support for Digital Audio and Video (NOSSDAV 2002), May 2002.

(28) A. Stavrou, D. Rubenstein, and S. Sahu, “A Lightweight, Robust P2P System to Handle Flash Crowds,” IEEE ICNP 2002, Nov. 2002.

(29) B. Y. Zhao, J. Kubiatowicz, and A. Joseph, “Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing,” UCB Tech. Report UCB/CSD-01-1141, Apr. 2000.

(30) A. Rowstron, and P. Druschel, “Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems,” IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Nov. 2001.

(31) I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, “Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications,” ACM SIGCOMM 2001, Aug. 2001.

(32) 三村, 中内, 森川, 青山, “ピアツーピアマルチキャストミドルウェアの実装,” 電子情報通信学会総合大会, B-6-85, Mar. 2002.

(33) B. Zhang, S. Jamin, and L. Zhang, “Host Multicast: A Framework for Delivering Multicast To End Users,” Proceedings of IEEE INFOCOM’02, Jun. 2002.

(34) 三浦，川島，高屋，“サーバ選択装置，サーバ選択方法，及びサーバ選択プログラムを記録した記録媒体,” 特開 2002-91843, May 2002.

(35) B. Krishnamurthy, J. Wang, and Y. Xie, “Early Measurements of a Cluster-based Architecture for P2P Systems,” ACM SIGCOMM Internet Measurement Workshop 2001, Nov. 2001.

(36) B. Krishnamurthy, and J. Wang, “On Network-Aware Clustering of Web Clients,” ACM SIGCOMM 2000, Aug. 2000.

(37) K. Obraczka, and F. Silva, “Network Latency Metrics for Server Proximity,” Proceedings of the IEEE Globecom 2000, Dec. 2000.

(38) B. Huffaker, M. Fomenkov, D. J. Plummer, D. Moore, and k claffy, “Distance Metrics in the Internet,” IEEE International Telecommunications Symposium (ITS) , Sep. 2002.

(39) T. S. E. Ng, and H. Zhang, “Predicting Internet Network Distance with Coordinates-Based Approaches,” Proceedings of IEEE INFOCOM’02, Jun. 2002.

(40) “The Network Simulator - ns-2,” http://www.isi.edu/nsnam/ns/

(41) “BRITE,” http://www.cs.bu.edu/brite/

(42) A. E. Sayed, and V. Roca, “A Survey of Proposals for an Alternative Group Communication Service,” IEEE Network, Vol.17, No. 1, Jan. 2003.

(43) D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel, “ALMI: An Application Level Multicast Infrastructure,” 3rd Usenix Symposium on Internet Technologies & Systems (USITS 2001), Mar. 2001.

(44) Y. H. Chu, S. G. Rao, and H. Zhang, “A Case for End System Multicast,” ACM SIGMETRICS, Jun. 2000.

(45) Y. H. Chu, S. G. Rao, S. Seshan, and H. Zhang, “Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture,” ACM SIGCOMM, Aug. 2001.

(46) S. Banerjee, B. Bhattacharjee, and C. Kommareddy, “Scalable application layer multicast,” ACM Sigcomm, Aug. 2002.

(47) J. Liebeherr, M. Nahas, and W. Si, “Application-Level Multicast With Delaunay Triangulation Overlays,” IEEE Journal on Selected Areas in Communications, Vol.20, No.8, Oct. 2002.

(48) E. Kranakis, H. Singh, and J. Urrutia, “Compass routing on geometric networks,” Proceedings of 11th Canadian Conference on Computational Geometry (CCCG’99), pp.51-54, Aug. 1999.

(49) S. Q. Zhuang, B. Y. Zhao, A. D. Joseph, R. Katz, and J. Kubiatowicz, “Bayeux: An architecture for scalable and fault-tolerant widearea data dissemination,” In Eleventh International Workshop on Network and Operating Systems Support for Digital Audio and Video(NOSSDAV 2001), Jun. 2001.

(50) A. Rowstron, and P. Druschel, “Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems,” IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, pp.329-350, Nov. 2001.

(51) S. Y. Shi, J. S. Turner, and M. Waldvogel, “Dimensioning Server Access Bandwidth and Multicast Routing in Overlay Networks,” Proceedings of NOSSDAV 2001, Jun. 2001.

(52) S. Y. Shi, J. S. Turner, “Routing in Overlay Multicast Networks,” Proceedings of IEEE INFOCOM’02, Jun. 2002.

(53) S. Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee, and S. Khuller, “Construction of an Efficient Overlay Multicast Infrastructure for Real-time Applications,” Proceedings of IEEE INFOCOM’03, Apr. 2003.

(54) A. Chakrabarti, and G. Manimaran, “A Case for Tree Migration and Integrated Tree Maintenance in QoS Multicasting,” pp.1007-1017, Computer Communications, vol.26, no.9, Jun. 2003.

(55) T. S. E. Ng, Y. Chu, S. G. Rao, K. Sripanidkulchai, and H. Zhang, “Measurement-Based Optimization Techniques for Bandwidth-Demanding Peer-to-Peer Systems,” Proceedings of IEEE INFOCOM’03, Apr. 2003.

(56) S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” Science, Number 4598, pp.671-680, May 1983.

(57) 小貝，市川，中里，横田，浦野，富永, “Diff-serv におけるマルチキャスト映像配信に関する検討,” 電子情報通信学会技術研究報告　情報ネットワーク（IN) 2000-51, 2000.

(58) 山口，中村，廣森，安本，東野，谷口, “ ユーザプリファレンスに基づく転送制御を行うアプリケーションレベルマルチキャストの一方式, ”情報処理学会研究報告（情処技報），Vol.2001， No.107 (2001-DPS-107)，pp.43-48, May 2002.

(59) Vivek Goyal, “Multiple Description Coding: Compression Meets the Network,” IEEE Signal Processing Magazine, vol.18, no.5, pp.74-93, Sep. 2001.

(60) Bernard M. Waxman, “Routing of multipoint connections,” IEEE Journal on Selected Areas in Communications, 1988.

(61) M. Doar, and I. Leslie, “How bad is naive multicast routing?,” IEEE INFOCOM ’93, 1993.

(62) E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, “How to Model an Internetwork,”, IEEE INFOCOM ’96, 1996.

(63) K. L. Calvert, M. B. Doar, and E. W. Zegura, “Modeling Internet Topology,”, IEEE Communications, Jun. 1997.

(64) Matthew. B. Doar, “A Better Model for Generating Test Networks,” IEEE Globecom ’96, Nov. 1996.

(65) M. Faloutsos，P. Faloutsos, and C. Faloutsos, “On Power-Law Relationships of the Internet Topology,” ACM SIGCOMM 1999, Sep. 1999.

(66) H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, “Network Topology Generators: Degree-Based vs. Structural,” ACM SIGCOMM 2002, Aug. 2002.

付 録 A

シミュレーショントポロジ

シミュレーション実験において，実ネットワークでの実験結果と同程度の信頼性を得るためには，実験に用いるネットワークトポロジが重要である．つまり，いかに実ネットワークが有する特徴を保持したシミュレーション用のネットワークトポロジを構築できるかということである．ここでは，本研究で用いたシミュレーションネットワークトポロジを含め，今まで提案されているものを整理する．最適ホスト選択手法，コンテンツ発見手法を提案し，その有効性を検証するためにはシミュレーションが重要である．ここでは，今まで提案されているシミュレーションネットワークトポロジについて整理する．

1. Random Graph Model

   Random Graph Model は，ある平面上のランダムな地点に点を配置し，ある確率 p でその点のペアを結ぶことによりネットワークトポロジを構築するモデルである．

   Random Graph Model にも幾つかの種類があり，それらは確率 p を決定するための手法が異なる．

   a) Pure random

      確率 p が一定である．

   b) Waxman[60]

      式 (A·1) により，ノード u から v へのリンク (エッジ) を構築するかの確率を決定する．

$$P(u,v) \;=\; \alpha e^{\frac{-d}{(\beta L)}} \quad (\; 0 \;<\; \alpha,\; \beta \;\leq\; 1 \;) \tag{A·1}$$

      α，β はモデリングパラメータ，d はノード u から v へのユークリッド距離，L はある 2 点ノード間の最大距離である (図 1b)．

      α が増加すれば，リンクの数が増加する．一方，β が増加すれば，距離が短いリンクより長いリンクの比率が増加する．

   c) Doar Leslie[61]

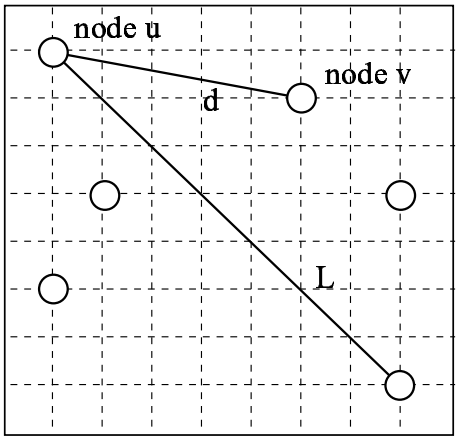


図 A–1: Waxman Model

Waxman の変形型であり，リンク確率関数が次式 (A·2) で与えられる．

$$P(u,v) \;=\; \alpha \frac{k\varepsilon}{n} e^{\frac{-d}{(\beta L)}} \quad (\;0 \;<\; \alpha,\; \beta \;\leq\; 1\;) \tag{A·2}$$

ε が平均 node degree(ある node から出ているリンクの数)，n が全ノード数, k が α 及び β に依存する定数である．Waxman に比べて，構築するリンク数を直接コントロール出来るが，Waxman における α パラメータを Doar Leslie モデルにあわせて選ぶことが可能なので，本質的に両者の違いはないと言える．

d) Exponential[62]

Waxman の変形型であり，リンクを構築する 2 点間の距離に注目したモデルである．リンク確率関数は次式 (A·3) で与えられる．

$$P(u,v) \;=\; \alpha e^{\frac{-d}{(L-d)}} \quad (\;0 \;<\; \alpha,\; \beta \;\leq\; 1\;) \tag{A·3}$$

2 点間の距離が長くなればなるほど，その 2 点間でリンクが構築される確率が減少するモデルである．

e) Locality[62]

リンク確率関数は次式 (A·4) で与えられる．

$$P(u,v) \;=\; \begin{cases} \alpha & \text{if } d < r \\ \beta & \text{if } d \geq r \end{cases} \tag{A·4}$$

2 点間の距離によって，異なる確率を持つカテゴリーを形成するモデルである．

[62] では，これら Random Graph Model の特徴を 1) node degree distribution: 平均 node degree，2) Hop-depth distribution: あるノード u をルートにし，他の全てのノードに対しての shortest path tree を構築した場合の tree の深さ，3) Length-depth distribution: 距離メトリックを用いた Hop-depth distribution, 4) number of bicomponent: biconnected component と呼ばれる循環リンクを構築しているリンクセットの数 (したがって，その内のいつかのリンクが落ちても，2 点間のリンクは維持されているということになる)， で比較している．

結果として，Random Graph Model は，現実のインターネットと比較して上記メトリックが異なっていることが示されている．そして，ノード数 n が増加すれば，平均 node degree もそれに合わせて増加してしまう点も指摘している．また，[63] では，Waxman 型について，1) バックボーンネットワーク，階層構造がなく現実的ではない点，2) connected network の保証がなされていない点，3) node が増えると，リンク数もそれに合わせて増え過ぎる点，が指摘されている．

2. 階層型ドメインモデル

[62] では，Transit-Stub Model と呼ばれる階層型トポロジモデルを提案している．これは，Georgia Tech Internetwork Topology Models (GT-ITM) と呼ばれる Internet のモデルを構築，解析するためのツールの一部として含まれている．[64] では，Tiers とよばれる階層型トポロジモデルを提案している．

現在のインターネットは，相互接続されたルーティングドメインの集まりとみなすことができる．ルーティングドメインでは，共通の管理化にある幾つかのノードがグループを構成し，そのグループ内でルーティング情報を共有している．そのようなドメインの特徴として，リンクを張る 2 つのノードは同じドメインに属する routing locality と呼ばれる性質が挙げられる．

各ノードが，LAN 又は HAN にあたり，これが Stub domain と呼ばれるドメインを構成する．そして，各 Stub domain が Transit domain とよばれる Stub domain 同士を相互接続するためのドメインを構成する (図 3)．したがって，Stub domain が MAN, Transit domain が WAN にあたると考えられる．

T を全 Transit domain 数，N_T を Transit domain 中の平均ノード数，S を Stub domain の平均数，N_S を Stub domain 中の平均ノード数，L を Stub domain 中の LAN の平均数，N_L を LAN 中の平均ホスト数とすると，全ルーティングノード N_R と全ホスト数 N_H はそれぞれ式 (A·5)，式 (A·6) のように表される．

$$N_R = TN_T(1 + SN_S) \tag{A·5}$$

$$N_H = TN_TSN_SLN_L \tag{A·6}$$

また，ドメイン間，ノード間の接続性に影響を与えるパラメータを次のように仮定する．

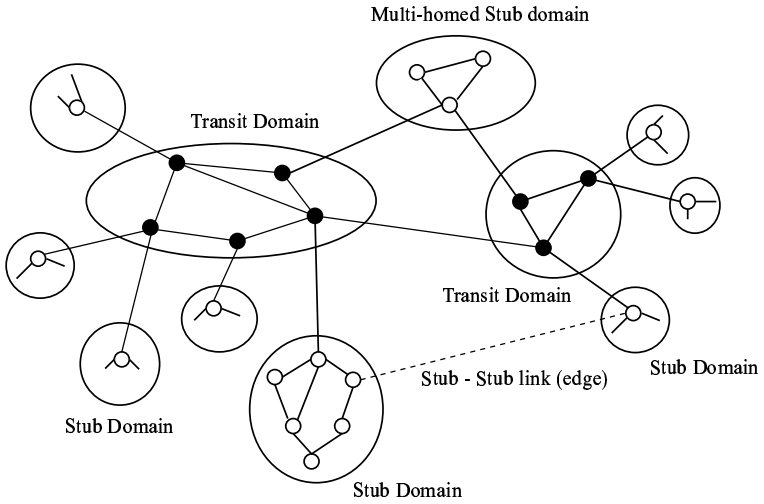


図 A–2: Transit-Stub Model

E_T: 同じ Transit domain に属するノード間での平均リンク数．各 Transit domain のグラフがきちんと結合されているように，E_T の値は十分大きく設定しなければならない ($E_T \geq 2$).

E_S: 同じ Stub domain に属するノード間での平均リンク数．各 Stub domain のグラフがきちんと結合されているように，E_S の値は十分大きく設定しなければならない ($E_S \geq 2$).

E_{TT}: Transit domain 間での平均リンク数．Transit domain がお互いに接続されるように，E_{TT} の値を十分大きく設定しなければならない ($E_{TT} \geq 2$).

E_{ST}: Stub domain と Transit domain 間での平均リンク数．すべての Stub domain が少なくとも 1 つの Transit domain に接続されるように設定する．すなわち，$E_{ST} \geq 1$ である．Multi-homed stub domain は，Transit domain に対して 1 つ以上のリンクがある．

E_{LS}: LAN と Stub domain 間での平均リンク数．すべての LAN が少なくとも 1 つの Stub domain に接続されるように設定する．すなわち，$E_{LS} \geq 1$ である．LAN は，1 つ以上の Stub domain に接続可能である．

Transit-Stub Model では，LAN，HAN は表さないので，$N_L = E_{LS} = 0$ である．Tiers では，複数の WAN をサポートしていないので，$T = 1$ となる．

(62) では，Trainsit-Stub Model の利点として，低い平均 node degree を保ったまま，大規模なグラフを作成できる点を挙げている．これは，Random Graph Model にはない特徴である．

3. Power-laws Model

[65] では，インターネットにおいて，1) ノード (ドメインあるいはルータ) から出るリンク数とランク，2) ノード数とノードから出るリンク数 (node degree)，などの関係がべき乗分布に従うということを指摘している．

Transit-Stub Model，Tiers により生成されるトポロジでは，この関係が成り立たないことが明らかになっている．そこで，このような関係が成立するようにトポロジを生成するモデルが Power-laws Model である．しかし，実際のインターネットは Transit-Stub Model のような階層モデルを有しているのに，それを無視することの疑問が残る．そこで [66] では，Power-laws Model で生成されるトポロジが階層モデルを Transit-Stub Model のほど厳密ではないが有している点を明らかにしている．そして，このような緩い階層モデルの方が，現実のインターネットを表現するのに適切であると指摘している．

ここでは，シミュレーショントポロジモデルについて述べたが，これらはあくまでノードとリンクの関係を表現するだけである．したがって，グラフ作成後，リンクの帯域，遅延を作成されたグラフに割り当てる必要がある．リンク帯域，遅延の割り当て法については，シミュレーショントポロジのようなモデルが存在しないのが現状であり，各シミュレーション実行者が各自の判断で設定している場合がほとんどである．今後，リンク帯域，遅延の割り当てモデルの確立，およびより実ネットワークに近いネットワークトポロジの構築法が期待される．

図一覧

表一覧

研究業績

	題　　名	発表年月	発表掲載誌	連名者
海外				
(1)	A Study on Content-Oriented Coding Scheme and Decoder Downloadable System	2002 年 7 月	World Multiconference on Systemics, Cybernetics and Informatics (SCI)	O. Nakagami N. Shimizu T. Miyazawa W. Kameyama H. Watanabe H. Tominaga
(2)	A Novel Decoder-downloadable System for Content-oriented Coding	2002 年 11 月	IEEE Global Telecommunications conference GLOBECOM02	N. Shimizu T. Miyazawa W. Kameyama H. Watanabe H. Tominaga
(3)	A Novel Content Distribution Architecture Utilizing Network Distance Prediction	2003 年 7 月	World Multiconference on Systemics, Cybernetics and Informatics (SCI)	N. Shimizu W. Kameyama H. Watanabe
国内				
(4)	コンテントオリエンティッド符号化実現のためシステム提案	2002 年 6 月	画像電子学会第 30 回年次大会	清水 直人 宮澤 敏記 亀山 渉 渡辺 裕 富永 英義
(5)	ホストの相対的位置情報を利用したコンテンツ配信に関する検討	2003 年 3 月	電子情報通信学会総合大会	清水 直人 亀山 渉 渡辺 裕
(6)	大規模コンテンツ配信のためのアプリケーションレベル・マルチキャストツリー構築に関する検討	2003 年 9 月	FIT (情報科学技術フォーラム) 2003	清水 直人 亀山 渉 渡辺 裕