修士論文

離脱を考慮したスーパノード型アプリケーション層マルチキャストに関する研究

平成 21 年度修了

三重大学大学院 工学研究科
博士前期課程 情報工学専攻

谷元 勇太

目次

はじめに

ADSL や FTTH などのブロードバンドの普及に伴い，動画や音声などのマルチメディアコンテンツへの需要が増加してきた．特に動画のように大容量サイズのものが多く，配信サーバやネットワークにかかる負荷が懸念される．例えば，ライブストリーミングなどは受信者の数が多くなることが予想されるので，コンテンツ配信方法は重要な課題である．これに対して，アプリケーション層マルチキャスト (ALM) が注目を集めている．ALM には，マルチキャスト機能をエンドノードに持たせることにより，既存のインフラを変更することなく運用することができるという利点がある．

しかしその一方で，いくつかの問題点もある．最も大きなものは，中継ノードがユーザ端末であるために接続が不安定になることである．例えば，あるノードがマルチキャストグループから離脱すると，そのノードが中継している全てのノードがデータを受信できなくなるという問題が生じる．このとき，データを受信できなくなったノードは配信ツリーの再構築を試みるのだが，再構築後も配信ツリー長を短く保つことで，安定したストリーム配信が実現できる．これは，配信ツリー長を短く保つことで，不安定の原因となる中継ノードの数が減少するためである．また，中継するノードが少ないことから，遅延時間の短縮も可能であると考えられる．

本論文では，安定したストリーム配信を実現するためにスーパノード型 ALM の提案を行う．提案する離脱手法では，ソースノードから離れたノードを離脱ノードの代わりとして再構築を行う．この手法と既存手法の違いは影響を受けないノードも再構築に利用することである．これにより，参加離脱の繰り返し後も配信ツリー長を短く保つことが可能であり，結果として影響を受けるノード数を減らすことができる．また，ソースノードが離脱処理を全て行うことによって，ノード数におけるスケーラビリティに問題が生じる．そのため，ソースノードが管理している全体のノード情報をスーパノード群で分散して管理を行い，特定のノードに負荷が集中しないようにするスーパノード型の提案を行う．スーパノードの生成は，配信ツリーへの参加が繰り返されると参加ノードから生成される．

本論文の構成について述べる．第 1 章では，オーバレイネットワークとそのネットワーク上で利用される技術や性質，配信ツリーの一般的な構築方法や分類，特徴について述べる．第 2 章では，本論文で用いる用語の定義や関連研究における研究の方向性について述べる．第 3 章では，本論文で提案するスーパノード型 ALM の参加離脱手法やそれらを行うために必要となる制御情報の定義，利用方法について述べる．第 4 章では，シミュレーション実験の条件や評価方法，評価項目について述べる．第 5 章では，提案手法と既存手法の比較実験の結果と考察について述べる．

第 1 章
オーバレイ (P2P) ネットワーク

オーバレイネットワークとは仮想的に個々のユーザ同士が直接形成するネットワークのことであり，ユーザはお互い対等な立場として動作を行う．図 1 にオーバレイネットワークの例を示す．オーバレイネットワークの用途としては，テレビ会議や動画配信，ファイル共有，電話，グループウェアなどが挙げられる．オーバレイネットワークを構成する上で，クライアントサーバと P2P の 2 つのサービス形態が主に利用される．まずはクライアントサーバについて次節で述べる．

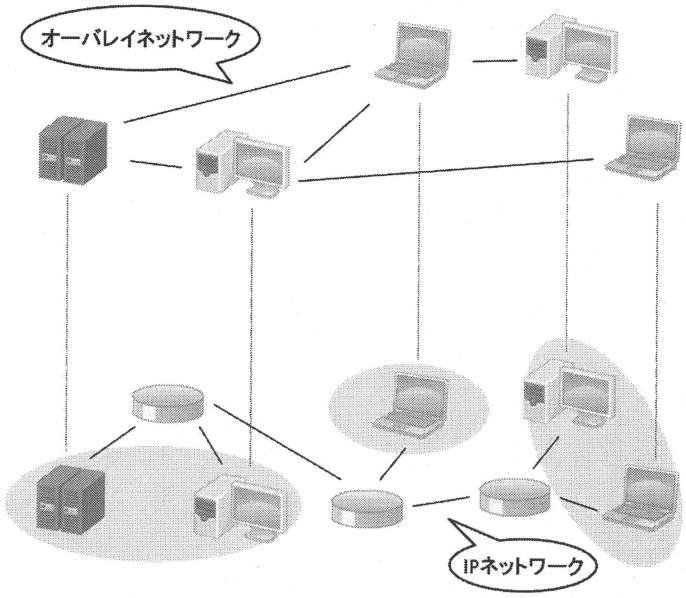

図 1　オーバレイネットワーク

1.1　クライアント/サーバ (C/S)

現在，web 検索サイト [1] などのインターネット上のサービスモデルの主流となっているクライアント/サーバモデルは，図 2 のようにサービスを提供するサーバと，サービスを要求するクライアントから構成されている．サーバはクライアントからの要求に応えるために常に待機しており，クライアントからサービスの要求を受けると，そのクライアントに対してサービスを提供する．このように，クライアントとサーバの機能は明確に区別されており，ほとんどの場合，サーバの数は

クライアントの数に対して非常に少ない．そのため，サーバは多数のクライアントにサービスを提供しなければならず，クライアントの数が増加しやすいサービスを提供している場合は膨大な管理作業が要求され，サーバの管理に必要となるコストが増加する．

また，近年ではADSL (Asymmetric Digital Subscriber Line) やFTTH (Fiber-to-the-Home) などの高速回線が急速に普及したことや，インターネットの利用者が増加したことにより，クライアントからサーバへの処理要求が大幅に増加し，サーバの負荷やサーバ周辺のトラフィックが増加している．そのため，サーバを構成する計算資源，及びネットワーク資源が過負荷となり，各クライアントに提供できるサービス品質の低下を招き，利用者に多大な影響を与える可能性がある．ここで，これまでに述べたクライアント/サーバモデルの問題点を以下にまとめる．

- サーバへの負荷とトラフィックの集中に伴うサービス品質の低下

次節では，これらの問題点を解決する新しいサービス形態であるP2Pについて述べる．

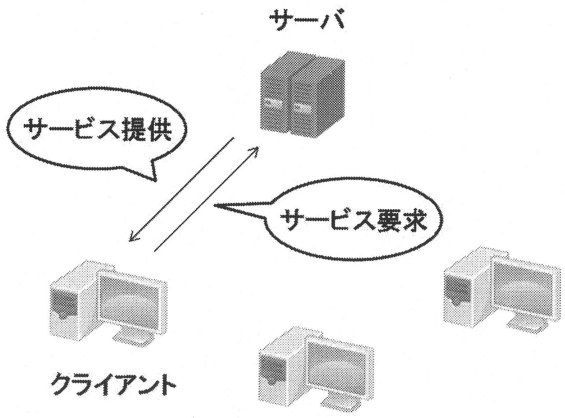

図2　クライアント/サーバモデル

1.2　P2P

P2Pモデルは前節で述べたクライアント/サーバモデルの問題点を解決するサービス形態として注目されている．図3のように，各ノードはサーバとしてもクライアントとしても機能する．そして，ノードが互いに協力してサービスを提供し合うことにより，クライアント/サーバモデルでは達成できなかった高品質なサービスの提供が可能となる．P2Pのメリットとしてスケーラビリティと耐障害性の2点が挙げられる．それぞれについて，以下に述べる．

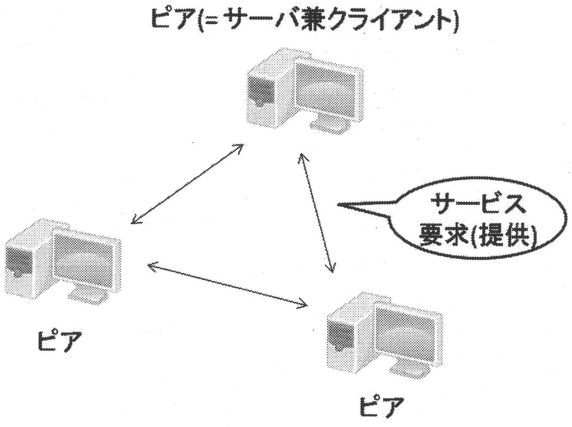

図 3　P2P モデル

1.2.1　スケーラビリティ

クライアント/サーバモデルでは，サービスを提供するサーバ周辺にトラフィックが集中する問題があったが，オーバレイネットワークでは，全てのノードがサービスの提供を行うため，サービスの要求・提供による負荷をネットワーク全体に分散させることができる．つまり，クライアントサーバモデルのように，クライアントの増加に対応するためにサーバの補強・増設をする必要が無いため，ノード数の増加に対するスケーラビリティが高い．

1.2.2　耐障害性

クライアント/サーバモデルでは，サーバもしくはサーバ周辺のネットワークに障害が発生した場合，そのサーバが提供するサービス全体が停止し，全クライアントはサービスを受けることができなくなる．しかし，オーバレイネットワークでは特定のノードやネットワーク障害の場合でも，別のノードを介してサービスを継続して提供できる．そのため，クライアント/サーバモデルと比較してサービスの耐障害性が向上する．

オーバレイネットワークでも IP ネットワークと同様にマルチキャストを用いることができる．次節では，マルチキャストについて述べる．

1.3 マルチキャスト

マルチキャストとは，ネットワーク内で複数の相手を指定して同じデータを送信することであり，オーバレイネットワークを構築する上で重要である．マルチキャストには，IP マルチキャストと ALM の 2 種類がある．まずは，IP マルチキャストについて述べる．

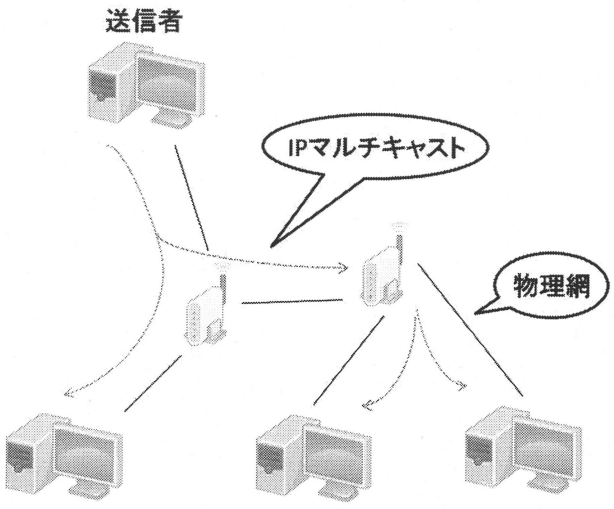

図 4　IP マルチキャスト

1.3.1 IP マルチキャスト

図 4 のように IP マルチキャストはデータの複製をルータが行う方式である．そのため，ユーザの動作に関係なく安定したマルチキャストが行うことができる．しかし，IP マルチキャストを利用するためにはルータのマルチキャストサポートが必要である．そのため，オーバレイネットワーク上の全てのルータが IP マルチキャストに対応している必要があり，実際には困難である．以下に IP マルチキャストの特徴をまとめる．

- ユーザの影響を受けずにマルチキャストを行うことができる
- ルータが IP マルチキャストに対応している必要がある

次に ALM について述べる．

1.3.2 ALM(Application Layer Multicast)

ALM はオーバレイマルチキャストとも呼ばれ，図 5 のようにデータの複製をユーザ端末が行う方式である．この方式ではアプリケーション間で独自にリンクを張ってマルチキャストを行うため，ルータの設定が不要である．しかし，マルチキャストを行う上でユーザ依存が大きいため，動作に問題がある．また，物理トポロジに依存しないオーバレイネットワークの構築を行うとネットワーク負荷が高くなることがある．以下に ALM の特徴をまとめる．

- ルータの設定が不要である
- ユーザ依存が大きい

近年ではルータの設定が不要であることから，本論文の研究分野ではマルチキャスト技術として ALM が多く用いられている．本論文でも，マルチキャスト技術として ALM を用いることにする．次節では，配信ツリーについて述べる．

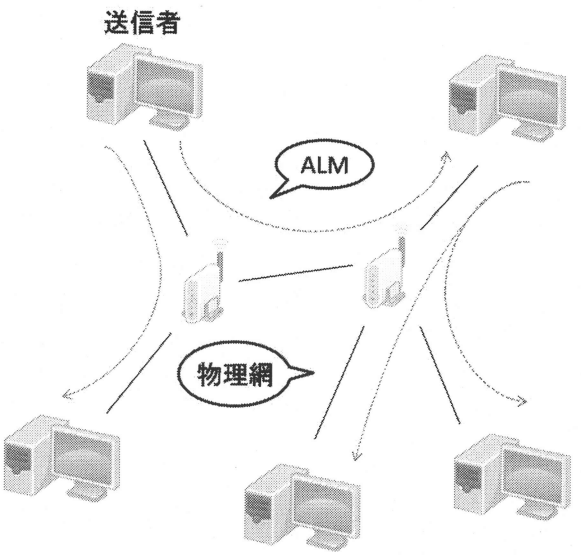

図 5　ALM(Application Layer Multicast)

1.4　配信ツリー (オーバレイマルチキャストツリー)

配信ツリーとはオーバレイネットワークの一種であり，ネットワークトポロジに木構造型を適用している．また，配信ツリーはノード間でコネクションをリレーさせることで，擬似的なマルチキャストを実現する．このとき，各端末ノードは IP 層を考慮する必要はないため，ネットワーク

構築にはアプリケーション層のみを考えれば良い．一般的な配信ツリーの例を図 6 に示す．各ノードは親ノードからデータを受信し，子ノードが接続されていれば，子ノードにデータを送信する．

配信ツリーは用途に応じていくつかのタイプがあるが，本論文では単一の配信ノードによるツリー型の ALM を対象とする．これは配信ツリーのルートに配信ノードが位置するもので，ライブストリーミングなどの放送型アプリケーションに適している．次に配信ツリーの分類について述べる．

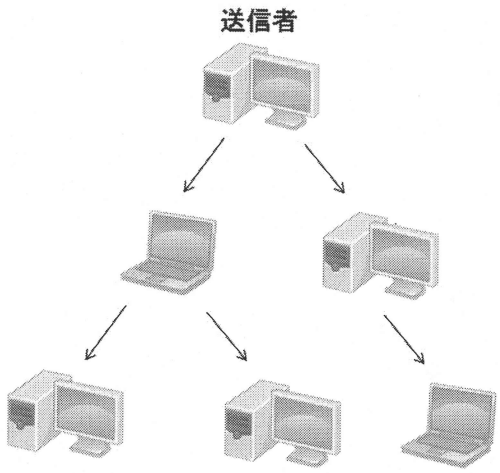

図 6　配信ツリー (オーバレイマルチキャストツリー)

1.5　配信ツリーの分類

ノード参加による接続先ノードの検索方法やノード離脱による再構築方法は配信ツリーの種類によって様々である．ここでは配信ツリーを，接続先ノードの検索方法を基準とした Hybrid-ALM, Pure-ALM，SN-ALM(スーパノード型 ALM) の 3 種類に分類し，それぞれの特徴について述べる．まずは，Hybrid-ALM について述べる．

1.5.1　Hybrid-ALM

Hybrid-ALM は，ストリーム配信を行っている全てのノードの情報をインデックスサーバ (IS) が管理する方式である．この方式では，データ配信に P2P 型をノード情報の管理に C/S 型を利用している．図 7 に一般的な Hybrid-ALM の例を示す．この方式では，ノードの参加時に IS が管理している全てのノードの情報を用いて接続先ノードの検索を行うため，目的に応じて最適なノードを接続先とすることができる．また，ノード離脱時には全体の情報を用いて目的に応じた最適な

配信ツリーを再構築することが可能である．しかし，ISに処理が集中するため，ISの負荷が大きくなる問題がある．以下にHybrid-ALMの特徴をまとめる．

- 目的に応じた最適な接続先の検索や再構築を行うことができる
- ノード数の増加によるスケーラビリティに問題がある

次にPure-ALMについて述べる．

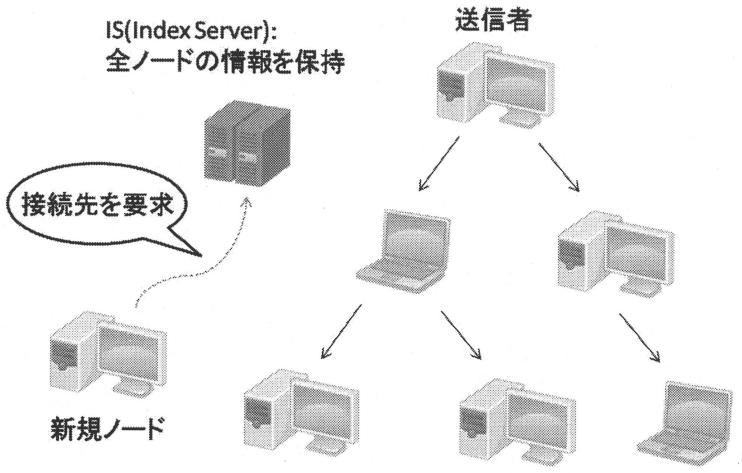

図7　Hybrid-ALMの例

1.5.2　Pure-ALM

Pure-ALMは，全ノードの情報を管理するノードが存在しない方式である．図8に一般的なPure-ALMの例を示す．接続先の検索はソースノードから葉ノードに向けて順に接続しているノードを用いて行う．この方式では，ISのような全体の情報を知っているノードが存在せず，近隣のノード同士で情報のやりとりを事前に行い，ノードの参加や離脱時に用いる．情報のやりとりは限定的であるため，ノード離脱時に目的に応じた最適な再構築が困難である．しかし，一元管理しているノードが存在しないため，ノード数増加によるスケーラビリティに優れている．以下にPure-ALMの特徴をまとめる．

- ノード数増加によるスケーラビリティに優れている
- 目的に応じた最適な再構築が困難である

最後にSN-ALMについて述べる．

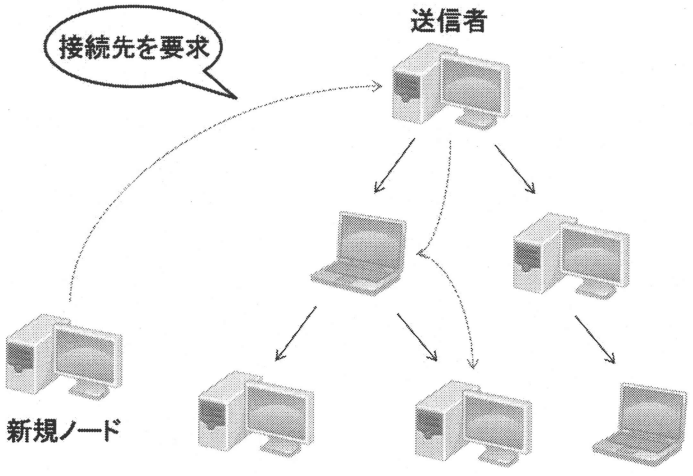

図 8　Pure-ALM の例

1.5.3　SN-ALM(スーパノード型 ALM)

SN-ALM(スーパノード型 ALM) はスーパノード (SN) と呼ばれるノード群が分散して全体の情報を管理し，その情報を用いて接続先の検索や再構築を行う方式である．スーパノードは一般に配信ツリーに参加しているノードの中から，滞在時間や端末性能を考慮して選択される．図 9 に一般的な SN-ALM の例を示す．図では配信ツリーをサブツリーごとに分割して管理を行い，サブツリーのルートがスーパノードとなる．

この方式では，全体の情報を分散して管理しているため，スーパノード群で必要な情報を共有することで最適な接続先の選択や目的に応じた最適な再構築を行うことが可能である．また，ノード数に応じてスーパノード数が調整されるため，ノード数増加によるスケーラビリティにも優れている．しかし，スーパノードの導入によって処理が複雑になる問題がある．以下に SN-ALM の特徴をまとめる．

- 最適な接続先の選択や目的に応じた最適な再構築が可能である
- ノード数増加によるスケーラビリティに優れている
- 処理が複雑になる

関連研究の多くは一般に，Pure-ALM 型を想定している．しかし，ここで述べたように SN-ALM のメリットは非常に大きいと考えられる．そのため，本論文では SN-ALM を想定した配信ツリーの提案を行う．次節では，一般に想定されている Pure-ALM 型の一般的な配信ツリーの構築方法

について述べる．

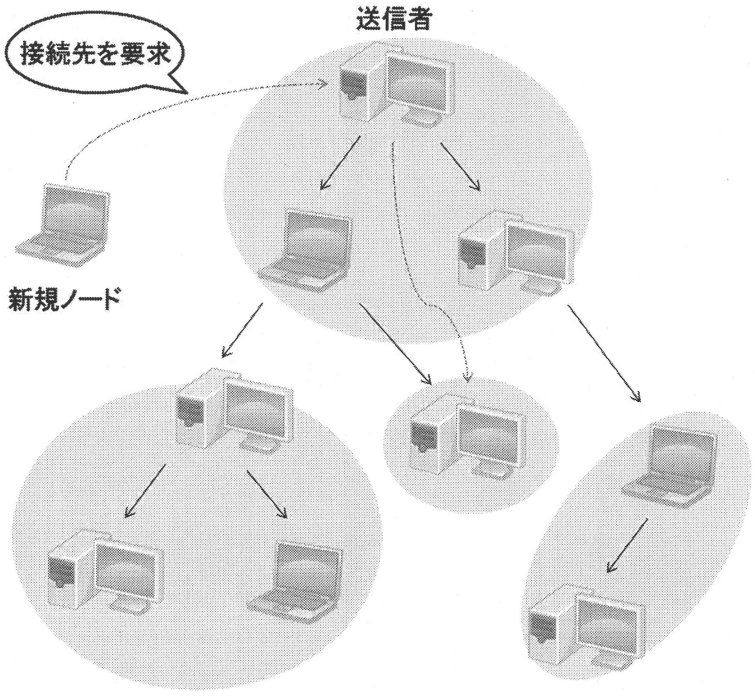

図 9　SN-ALM の例

1.6　配信ツリーの一般的な構築方法

配信ツリーの構築アルゴリズムは join，leave，failure 時に用いられる 3 つのサブアルゴリズムにより構成される．本論文では特にツリー型の ALM である PeerCast[2] を例に挙げ，各イベント発生時の配信ツリー構築アルゴリズムを説明する．

1.6.1　join アルゴリズム

join アルゴリズムはノードが配信ツリーに参加する際に用いられる．図 10 は新規ノードが配信ツリーに参加する例である．新規ノードが配信ツリーに参加するとき，新規ノードはまずソースノードへ接続要求を送信する (Tree-first 型)．ここでソースノードは帯域に余裕があるのであれば新規ノードを受け入れる．そうでなければ現在送信している子ノード群の中から 1 つ選択し，そのノードへリダイレクトさせる．この手順を繰り返すことにより新規ノードはいずれかのノードと接続し，データを受信する．

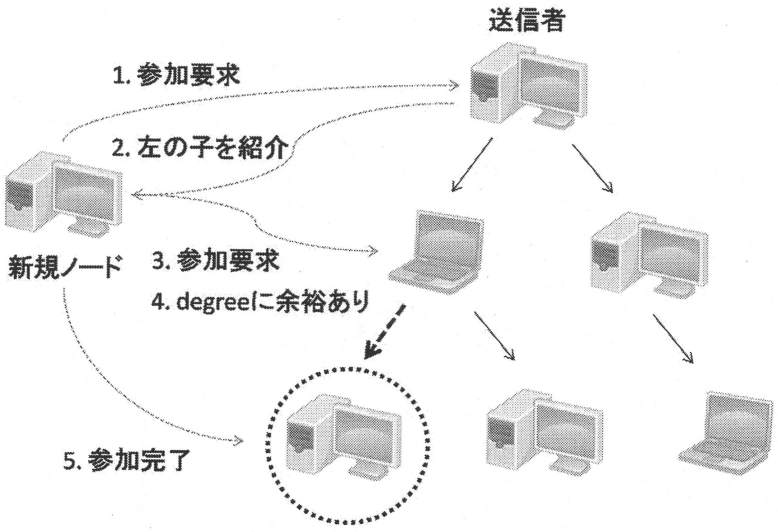

図 10　一般的な join アルゴリズム

1.6.2　leave アルゴリズム

leave アルゴリズムは配信ツリーへ参加中のノードが離脱するときに用いられる．図 11 に一般的な離脱の例を示す．ノードが離脱を行う際に，離脱ノードはまず親と全ての子に離脱メッセージを送信する．メッセージを受信した親はそのノードとのセッションを開放する．子は親ノードの消失によりそのままではデータを受信できなくなるため，新しい親を見つける必要がある．ここでは離脱ノードの親，すなわち祖父にあたるノードへ再接続要求を送信する．祖父の情報は離脱メッセージにより与えられる．その後は join アルゴリズムと同様の処理を行うことで，leave アルゴリズムが完了する．

1.6.3　failure アルゴリズム

ALM は中継ノードが信頼性の低い端末ノードであるため，一度張ったオーバレイリンクが確立されているかどうかを定期的に確認する必要がある．各ノードは親と子に対して一定間隔でメッセージを送信する．これをハートビートメッセージと言い，一定時間経過してもメッセージが届かない場合は障害が発生したと判断してこのリンクを切断する．その際の対応は leave アルゴリズムとほぼ同様であるが，故障ノードの子は祖父にあたるノードを知らないのでソースノードへ再接続要求を送信する．

本章では，オーバレイネットワークとそのネットワーク上で利用される技術や性質について述べ

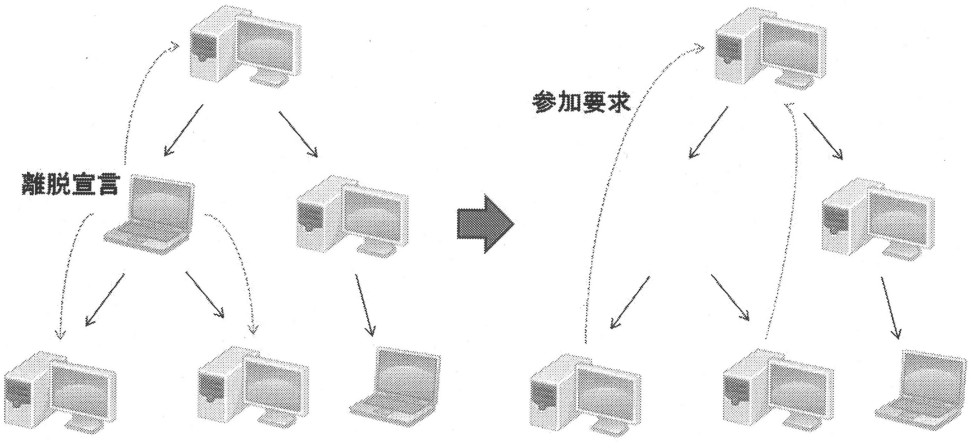

図 11　一般的な leave アルゴリズム

た．また，本論文の対象としている配信ツリーの一般的な構築方法や分類，特徴について述べた．次章では，本論文の研究で用いられている用語の定義や関連研究における研究の方向性について述べる．

第 2 章
関連研究

オーバレイネットワークは用途によって，重要な指標が異なってくる．例えば，テレビ会議のようなインタラクティブなアプリケーションの場合にはノード間の最大遅延を最小にすることが大事であり，ライブストリーミングのような放送型アプリケーションの場合には安定したストリーム配信を行うために配信ツリー長を短く保つことが重要である．

また，関連研究の多くは実環境を想定している．例えば，実際のノード間の距離を考慮してリンク遅延を設けたり，ユーザ端末のリソースに限りがあることからネットワーク帯域に制限を加えている．さらに，オーバレイネットワークへの滞留時間や端末の性能を考慮している研究もある．本論文では一般に考慮されているネットワーク帯域のみを考慮する．

次節では本論文の研究分野で用いる用語の定義について述べる．

2.1 用語定義

2.1.1 degree

実環境ではユーザ端末のネットワーク帯域には限りがある．そのため，ALM でデータを送信できる子ノード数を制限する必要がある．本論文では，ある時点で接続されている子ノードの数を degree と呼ぶことにする．

2.1.2 最大 degree

本論文では degree の上限値を最大 degree と呼ぶことにする．この値は配信ツリーに初めて参加する際に決定する．一般的には参加ノードからの自己申告や RTT(Round Trip Time，往復遅延時間) によって計測を行い，参加ノードの帯域を求める．そして，ストリーミングを行うのに十分な帯域を事前に求めておき，他のアプリケーションを考慮して参加ノードの最大 degree を決定する．これにより，中継ノードの負荷を抑えることができる．この時，ISDN などのナローバンドを利用しているノードの場合は，ストリーミングに必要な上り帯域が確保できない場合がある．しかし，現在は ADSL や FTTH が十分に普及してきており，将来的には全て，これらのブローバンドに移行するという前提で話を進める．

本論文では違和感なくストリーミングができ，かつ，他アプリケーションへの影響を考慮して，最大 degree を 2~5 と設定する．そのため，最大 degree が 0 の場合に起こり得る，子の接続がで

きない問題などは考慮しない.

2.1.3 空き degree

ある時点での子ノードとして接続受け入れ可能な degree 数とする．つまり，最大 degree と degree の差である.

2.1.4 ソースノード

データの配信を最初に行うノードのことである．単一ソースノードの場合は，通常は配信ツリーのルートがこれにあたる．複数ソースノードの場合は，例えば，テレビ会議では発言者がこれにあたる.

2.1.5 配信ツリー長

配信ツリーを構成するソースノードからの hop のことである．ただし，ソースノードは 0 とする.

2.1.6 hop

ルータやハブなどの中継機器を除いた仮想的なノード間の距離のことである.

2.2 関連研究について

この節では，関連研究の概要について述べる.

2.2.1 予備経路によるツリー修復

[3], [4], [5] では各ノードが離脱に備え，予め予備経路を確保することで，配信ツリーの再構築後でも配信ツリー長を短く保ち，データ転送遅延を抑えることが可能な手法が提案されている．[3] では予備経路の作成に当たり，コントロールパケットを用いており，新規ノードがツリーに参入する度に予備経路を計算する．新規ノードの親ノードが離脱する仮定の下，その親子間で全域木を作ることを目的としている．しかし，以下で述べる (1) 式を満たしていない場合，着目ノードの親子間だけでは全域木を作ることができない．この時，着目ノードより下位に位置するノードを用いるため，配信ツリー長が長くなる．また，ノードの参加・離脱の度に，その親と祖父にあたるノードは (1) 式を計算する必要があり，それによってコントロールオーバヘッドも増加する.

$$\sum_{j=0}^{n-1} d(c_j) \geq n-1 \tag{1}$$

(1) 式において，$d(c_j)$ は子 c_j の空き degree，n は子供の数を表す.

[4] では，[3] の問題点であるコントロールオーバヘッドの増加と確実に着目ノードの親子間で全域木を作ることを目的としている．問題点を解決するため，各ノードは少なくとも 1 つ degree を余らせている．これにより，(1) 式を常に満たし，親子間で全域木を作ることができる．以下に，予備経路作成の手順について説明する (図 12 の第 1 予備経路)．

1. 新規ノードの親ノードは自分の親 (新規ノードの祖父) に対して子の情報を送信する．
2. 受信した祖父ノードは RTT を用いて，反応の早い孫ノードを祖父ノードからの予備経路とする．
3. その次に応答した孫ノードを 1 つ前に応答した孫ノードの予備経路とし，全ての孫ノードが予備経路を作成するまで行う．

この手法では，子ノード間で直線状に予備経路を作成するため，再構築後に配信ツリー長が長くなりやすい問題がある．また，離脱や障害により予備経路を用いて切り替えが行われた場合に degree が最大に達してしまうことがある．その場合は，最大 degree に達したノードをルートとするサブツリーの中から，2 ノード以上の受け入れが可能なノードを検索する．そして，最後に受け付けたノードをそのノードに接続させることで，各ノードは常に degree を 1 つ余らせることができる．

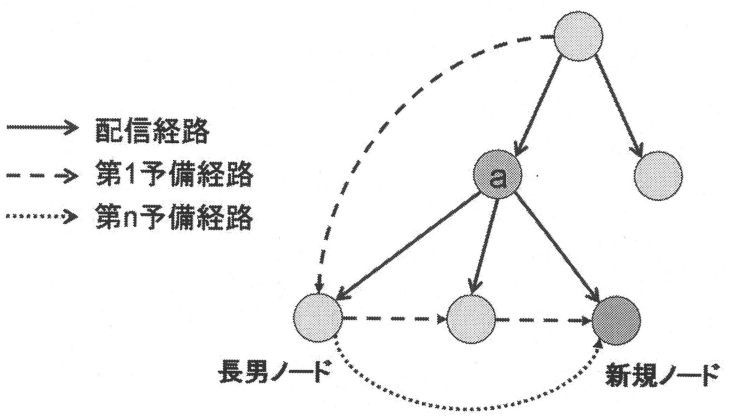

図 12　a を中心とした予備経路作成の例

[5] では，[4] に複数の予備経路を適用することで，配信ツリー長が長くなることを防いでいる．この手法では [4] と同様に 1 本の予備経路 (第 1 予備経路) の作成を行った後，提案する方法で複数の予備経路 (第 n 予備経路) を作成する．第 1 予備経路はノード離脱時に次の接続先として保証される経路であるが，第 n 予備経路は degree に余裕がある場合に限り接続できる経路である．

第 1 予備経路の作成方法は，[4] と同様である．ただし，指標値として，RTT ではなく空き

degree としている．第 n 予備経路の作成方法は以下のとおりである (図 12 の第 n 予備経路)．

1. 長男ノード (祖父ノードからの予備経路先ノード) の空き degree が 3 以上であれば，長男ノードから新規ノードへ予備経路の作成を行い，これを第 n 予備経路 (n : 2 以上の整数) とする．

ここで，空き degree を 3 以上としている理由は，親子間で確実に予備経路を作成するための余裕の 1 本と第 1 予備経路に用いる 1 本以外の空き degree を使用するためである．ノードの離脱が行われた際の予備経路の適用方法を図 13 に示す．まず，離脱ノードの親ノードは長男ノードへの第 1 予備経路を適用する．次に長男ノードは第 1 予備経路に切り替え，その後に空き degree に余裕があれば，第 n 予備経路を適用する (図 13:右)．そうでなければ，第 1 予備経路の切り替え先のノードが同様の処理を行う．

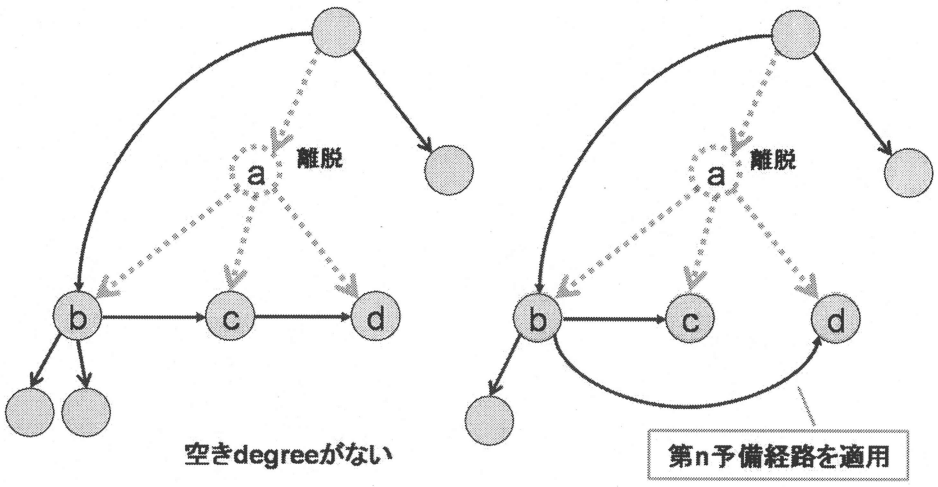

図 13　予備経路を適用した再構築の例

2.2.2　滞留時間を考慮した配信ツリー構築

[6], [7] では，上流ノードの離脱というイベントの回数そのものを減らして，安定したストリーム配信を実現しようとしている．そのための研究アプローチとして，ある時点での過去の視聴時間が長いノードほど将来の視聴時間も長い確率が高いという前提の元に，視聴者の視聴時間 (滞留時間) の統計的分布 ([8], [9]) とツリー構築アルゴリズムを用いて，長時間離脱していないノードを配信ツリーの上流に配置している．以下に，[6]，[7] が提案する join, leave アルゴリズムを示す．

join アルゴリズム

基本は 1.6.1 項に示した join アルゴリズムと同様である．異なる点は帯域に余裕がない場合の指標値として，滞留時間を用いていることである [6]．しかし，この手法では滞留時間さえ長ければ，ツリーにおける深い位置にあるノードを親として選択する．そのため，離脱なしで参加が続いた場合に深さの不均衡が解消されないことから滞留時間が長いノードの下流に多くのノードが集まりやすいという問題点がある．また，滞留時間が長いノードほど将来も長く接続する性質はあくまで確率的なものであり，滞留時間が長くても直後に離脱するノードも存在する．そのようなノードが離脱した場合に下流に集まったノードに影響が及ぶ．[7] では深さを指標値として取り入れることでこの問題を解決している．

leave アルゴリズム

提案されている離脱手法は離脱ノードの直近の子ノードから滞留時間の長いノードを代わりとする．そして，代わりのノードを離脱ノードとみなして，葉ノードまで再帰的に行う手法である (図 14)．この手法の問題点として最大 degree を固定とした場合では配信ツリー長を短く保つことが可能であるが，そうでない場合に配信ツリー長が長くなってしまう問題が残っている．

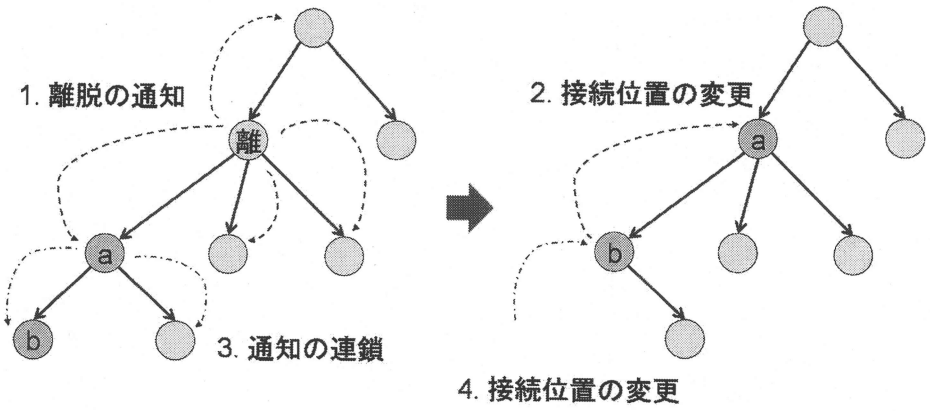

図 14　leave アルゴリズムの例

2.2.3　複数ツリーによる構築

ツリー型 ALM の影響問題について，映像符号化の面からカバーする提案を [10] で行っている．しかし，中央サーバによる集中制御で管理されているため，スケーラビリティの問題がある．その問題点を解決するために，[11] では分散ストリーミングシステム (DALSIM: Distributed ALM Streaming System with Independent Multi-tree) を提案している．この手法では [12] で提案さ

れている高位階層間独立スケーラブル映像符号化方式を用いる．この符号化方式の特徴を以下に示す．

- 一つの送信ストリームを複数ストリームに分割する．
- 全ての分割したストリームが相互に独立で対等である．
- 受信した分だけ画質を向上することが可能である．

この符号化方式によって分割されたストリームごとに異なる配信ツリーを構築し，それらのツリーを合成してストリーミングを行う (図 15)．これによって，システム全体としてはストリーミングが停止することなく継続することが可能であるようにする．

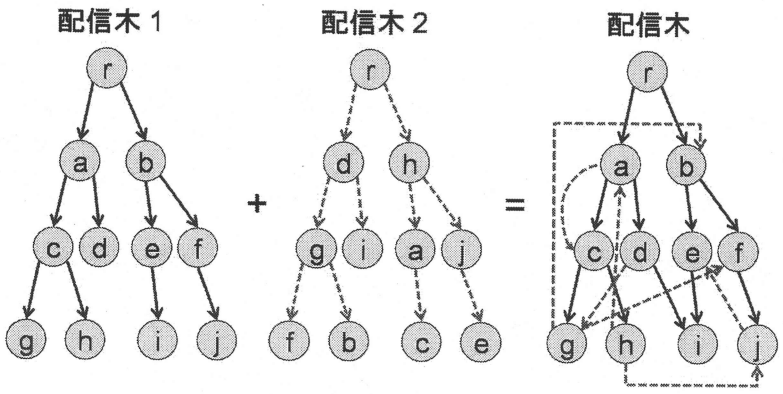

図 15　独立した複数ツリーとツリー合成

2.2.4　既存の SN-ALM

[13] で提案されている SN-ALM では，初期状態ではソースノードをスーパノードとして全体の情報を管理する．そして，スーパノードが管理できるノード数の上限を越えた接続要求を受け付けた場合，管理ノードの中から 1 ノードをスーパノードに変更する．スーパノードの選出はスーパノード追加時での最適な接続先ノードとする．また，ノード i においてソースノードからのホップ数を H_i, degree を C_i とし，H_i の最大値を H_{max}, 最大 degree を C_{max} とした時の接続適正値を式 (2) で求める．

$$x_i = (1-w)\frac{H_{max}-H_i}{H_{max}+1} + w\frac{C_{max}-C_i}{C_{max}} \tag{2}$$

ただし，w は $0 < w < 1$ とする．新規ノードの参加は x_i を最大にするノードに接続を行う．しかし，[13] では，離脱を考慮した提案がされていないため，本論文では離脱を考慮した提案を行う．

2.2.5 代替ノードを用いた再構築手法

[14] は離脱時に離脱ノードの代わりとなるノード (代替ノード) を配信ツリー上から検索し，そのノードを用いて再構築する手法である．代替ノードは条件にあったソースノードから離れたノードを選択する．これによって，再構築後も配信ツリー長を短く保つことができる．具体的な配信ツリーの構築手法については次章で述べる．この手法と既存手法の違いは影響を受けないノードも再構築に利用することである．[14] では配信ツリーの構築に必要となる情報をソースノードが全て管理する方式を用いているため，ソースノードに負荷がかかるという問題がある．そのため，本論文では代替ノードを用いた離脱手法にスーパノード型を適用した提案を行う．

本章では，本論文の研究で用いられている用語の定義や関連研究における研究の方向性について述べた．次章では，本論文で提案する SN-ALM の参加離脱手法やそれらを行うために必要となる制御情報の定義，利用方法について述べる．

第 3 章
提案するスーパノード型 ALM

本論文の提案手法は [13] と同様に，初期状態でソースノードをスーパノード (SN) として全体のノード情報を管理する．そして，join アルゴリズム (3.2 節) を用いて配信ツリーへの参加が繰り返されると参加ノードからスーパノードを生成する．このようにして生成されたスーパノード群で全体のノード情報を分散して管理を行い，あるノードに負荷が集中しないようにする．

参加ノードが配信ツリーから離脱する場合は leave アルゴリズム (3.3 節) を用いる．このアルゴリズムでは，ソースノードから離れたノードを離脱ノードの代わりとするため，再構築後も配信ツリー長を短く保つことが可能である．

3.1 制御情報

スーパノードはグループに関するノード情報を管理する．その情報を用いて，参加時に必要となる候補ノード情報と離脱時に必要となる代替ノード情報を求める．これらの情報はグループ内で参加離脱が行われる度に更新される．

3.1.1 候補ノード情報

候補ノードとは，そのグループに参加する時の接続先となるノードのことである．この情報は図 16 のフォーマットに従って，各スーパノードからソースノードに集められて参加時に利用される．図 17 に候補ノードの求め方の例を示す．ここで，図 17 の x/y,z は，degree / 最大 degree, hop を示す．hop はソースノードからの距離とする．

SN_{ID}	候補ノードの hop

図 16　候補ノード情報

スーパノードは管理しているノード情報を用いて，候補ノードを最大で 1 つ求める．候補ノードとなる条件は子の接続が可能で，かつ，hop が最小であるノードである．図 17 の場合，子の接続が可能であるノードは degree が最大 degree に達していない C, D, E, F であり，これらのノードから hop が最小である C が候補ノードとなる．

また，グループ内の全てのノードの degree が上限に達している場合はノードの参加受け入れが

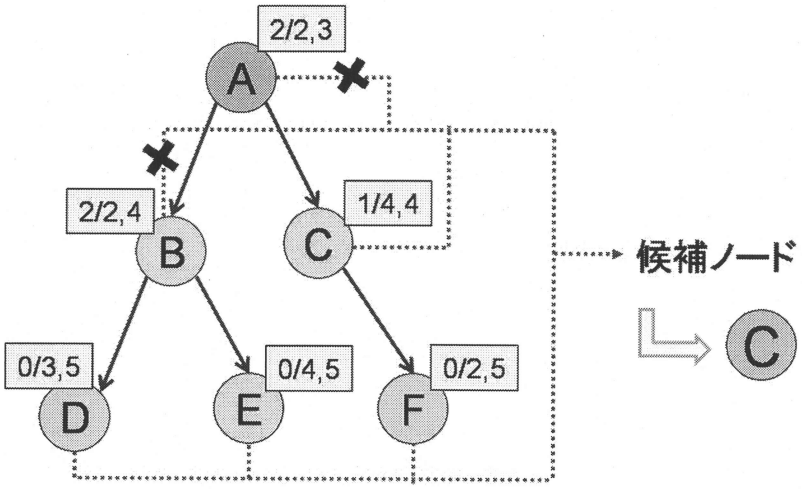

図 17　候補ノードを求める例

不可能である．その場合は図 16 の 候補ノードの hop の情報は ”-” とする．

3.1.2　代替ノード情報

代替ノードとは離脱ノードの代わりとなるノードのことであり，代替葉ノードと代替節ノードから選択される．この情報は図 18 のフォーマットに従ってスーパノード群で共有されて，離脱時に利用される．ここで，図 18 の type はスーパノードかそうでないノーマルノードか (SN か NN) を示す．スーパノードは管理しているノード情報を用いて，代替葉ノードと代替節ノードをそれぞれ最大で 1 つ求める．図 19 に代替ノードの求め方の例を示す．

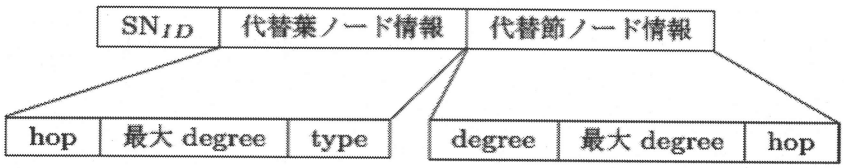

図 18　代替ノード情報

代替節ノードはスーパノードを除く，子の接続が可能な節ノードが代替節ノードの候補となる．スーパノードを除いた理由はスーパノードの離脱では代替ノードがグループ情報を求める必要があるため，処理が複雑になるからである．また，子の接続が可能なノードに限定した理由は再構築時間を短くするためである．この理由については，leave アルゴリズム (3.3 節) で述べる．これらの候補から，最大 degree, degree, hop の優先順位で求める．それぞれ候補から最大 degree は最大

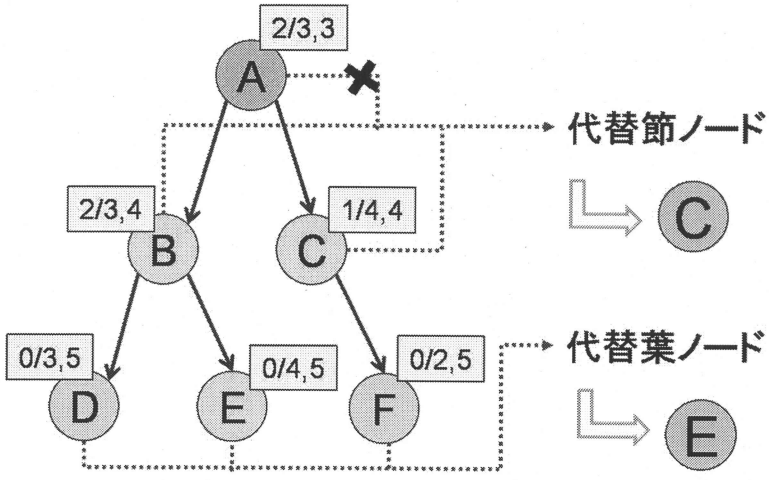

図 19　代替葉 (節) ノードを求める例

を，degree は最小を，hop は最大を優先する．図 19 の例では，代替節ノードの候補は B, C となり，最大 degree において C が最大であるため，代替節ノードは C となる．

代替葉ノードは葉ノードが代替葉ノードの候補となる．これらの候補から，最大 degree, hop, type の優先順位で求める．それぞれ候補から最大 degree は最大を，hop は最大を，type はスーパノードを優先する．スーパノードを優先する理由は無駄なスーパノードを減らすためである．図 19 の例では，代替葉ノードの候補は D, E, F となり，最大 degree において E が最大であるため，代替葉ノードは E となる．

また，条件を満たす代替ノードが求められない場合がある．その場合は図 18 の代替葉 (節) ノード情報は ”-” とする．

3.2　join アルゴリズム

join アルゴリズムはノードが配信ツリーへ参加時に適用される．図 20, 21 にノード参加の例を示す．ここで，図の (x,y) は (管理数，管理上限数) とする．管理数はスーパノードが現在管理しているノード数，管理上限数はその上限値とする．また，表 1 は図 20 の候補ノード情報リストを示している．

ソースノードは新規ノードから参加要求を受け付ける (図 20:1)．そして，ソースノードは候補ノード情報リスト (表 1) から，参加受け入れが可能でかつ，候補ノードの hop が最小である SN_p を選択し，そのスーパノードを新規ノードに通知する (図 20:2)．図 20 の場合は表 1 を参照すると，SN_B が最小の hop となるため，SN_B が新規ノードに紹介される．その情報を用いて新規ノー

ドは，SN_p に参加要求を行う (図 20:3).

表 1　候補ノード情報リスト

SN	A	B	C	D	E
hop	-	2	3	3	3

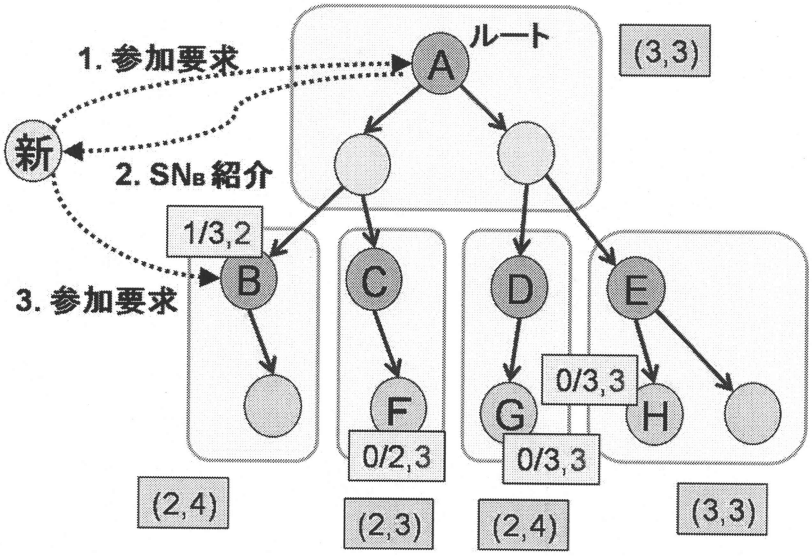

図 20　ノード参加 (1)

参加要求を受け付けた SN_p は以下に場合分けして処理する (図 21).

- SN_p の管理数に空きがある場合，候補ノードを親として新規ノードを接続する (図 21:(a)).
- SN_p の管理数に空きがなく，候補ノードが
  - 葉の場合：候補ノードを新規 SN として管理外とし，新規ノードを新規 SN に接続させる (図 21:(b)).
  - 節の場合：新規ノードは新規 SN として，候補ノードに接続する (図 21:(c)).

参加処理が行われた後，変更があったグループのスーパノード (または，新規 SN) は，候補ノード情報と代替ノード情報を更新し，変更があれば通知を行う.

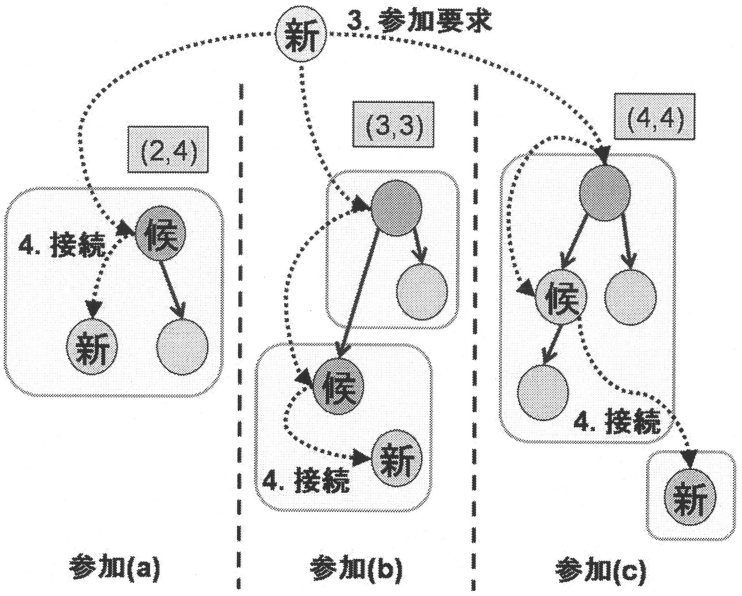

図 21　ノード参加 (2)

3.3　leave アルゴリズム

leave アルゴリズムはノードが配信ツリーから離脱する際に適用される．提案手法では代替ノード情報を用いて，代替葉ノード，代替節ノード，再参加の順で離脱処理を行う．条件に合う代替ノードを発見できるとその時点で再構築を行い離脱処理を完了するが，そうでなければ，再参加処理を行う．図 22, 23 にノード離脱の例を示す．また，表 2 は図 22 の代替ノード情報リストを示している．

ノード離脱では，まず離脱ノードが自分の所属しているグループのスーパノード (SN_l) に対して離脱要求を行う (図 22:1)．図 22 の場合は，SN_A が SN_l に該当する．離脱要求を受けた SN_l は代替ノード情報リスト (表 2) を用いて，代替ノードの検索を行う (図 22:2)．代替ノードの検索条件は以下とする．

代替葉：離脱ノードの degree 以上の最大 degree をもつ葉ノード．
代替節：離脱ノードの degree 以上の最大 degree であり，かつ，離脱ノードの degree より少ない degree である節ノード．

また，代替ノード情報の代替節を求める場合 (3.1.2 項) に子の接続可能なノードに限定した理由

は，代替節ノードの検索条件から利用されることがないためである．

表 2　代替ノード情報リスト

SN	A	B	C
代替葉	0/3,6,NN	0/4,6,NN	0/2,6,NN
代替節	1/3,5	-	-

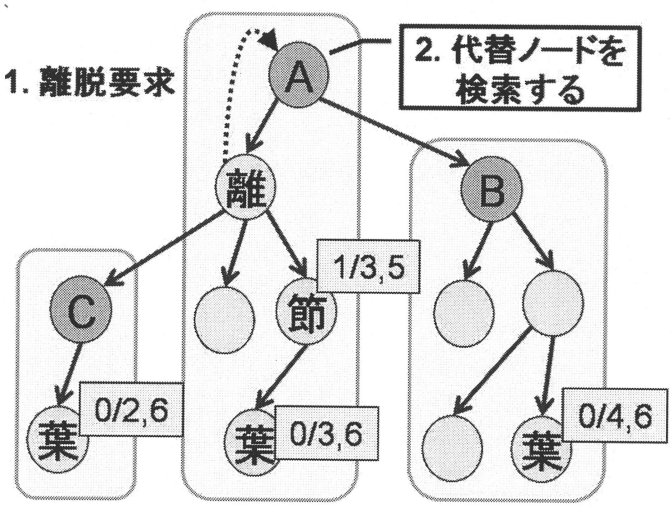

図 22　ノード離脱 (1)

図 22 の場合は，離脱ノードの degree は 3 である．代替ノードの検索順序は代替葉ノード，代替節ノードであるので，まずは代替葉ノードから検索を行う．表 2 を参照すると，条件を満たす代替葉ノードは A, B の代替葉ノードである．条件を満たす代替葉ノードが複数存在する場合は，最大 degree, (degree), hop, (type) の優先順位で 1 つ選択する．それぞれ，最大 degree は最大を，degree は最小を，hop は最大を，type はスーパノードを優先する．そして，選択した代替ノードを用いて再構築を行う (図 23:(a)-3)．もし条件に合う代替葉ノードが発見できなければ，同様にして代替節ノードの検索を行い，再構築を行う (図 23:(b)-3)．また，代替節ノードを用いて再構築を行う場合は，代替節ノードの元の位置で再帰的に離脱を行う．再帰的に離脱を行う際に離脱処理がループしないようにするため，代替節ノードの条件に離脱ノードの degree より少ない degree であるという条件を加えた．

条件に合う代替ノードが発見できなければ，再参加を行う (図 23:(c))．再参加では，まず離脱ノードの子ノードを全てスーパノードに変更する (図 23:(c)-3)．スーパノードに変更したノード (SN_i) の管理数が上限を超えた場合は SN_i 内のノードを 1 つ選択し，SN_k として管理から外す．そ

して，スーパノードに変更されたノード (と SN_k) はソースノードに参加要求を行う (図 23:(c)-4). 参加要求を受けたソースノードは，3.2 節の図 21:(c) と同様にして処理を行い，再構築を行う．

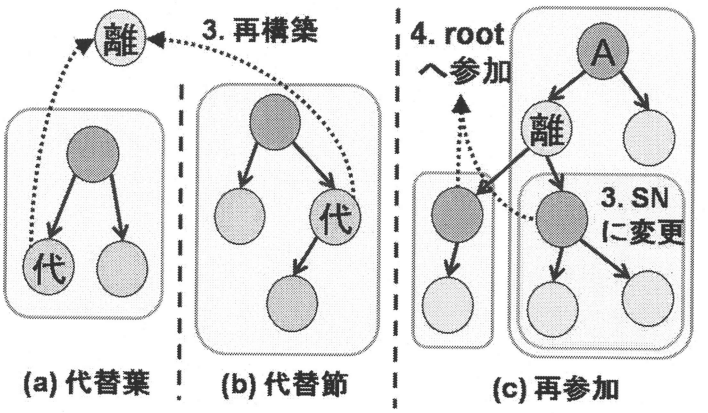

図 23　ノード離脱 (2)

離脱処理が行われた後，変更があったグループのスーパノード (または，新規 SN) は，候補ノード情報と代替ノード情報を更新し，変更があれば通知を行う．

3.4　failure アルゴリズム

各ノードは隣接するノードと定期的に接続が維持されているかの確認のためのメッセージ (ハートビートメッセージ) を送信している．そのため，離脱宣言を行わず離脱を行ったとしても別の隣接ノードが離脱に気付くため，離脱ノードの代わりにスーパノードに対して通知を行うことで，その後の処理は leave アルゴリズムを用いることができる．しかし，離脱の宣言を行った方が再構築時間が短くなると考えられるため，通常は離脱宣言を行う．

本章では，本論文で提案する離脱を考慮した SN-ALM について述べた．提案する離脱手法では，代替葉ノード，代替節ノードの優先順位で代替ノードとし，離脱ノードの条件に合う代替ノードが発見できれば，そのノードを用いて再構築を行う．条件に合う代替ノードが発見できなければ再参加を行うことで再構築が完了する．次章では，シミュレーション実験の条件や評価方法，評価項目について述べる．

第 4 章
シミュレーション実験

4.1 シミュレーションモデル

4.1.1 シミュレーションパラメータ

実験条件を表 3 に示す．ここで，ノード数は配信ツリーを構成するユーザ数，イベント数は参加離脱の回数 (間隔は連続して行う参加または離脱の回数) とする．また，表 3 を 1 セットとし，計 50 セットの平均を実験結果として求めた．

表 3　実験条件

ノード数	イベント総数 (間隔)	管理上限数	最大 degree
5000	30000(2500)	100	2,∼,5

4.1.2 実験方法

以下に実験の流れを示す．

1. 全ノードをツリーに参加させて，性能評価を行う．
2. ツリーからイベント間隔数のノードを離脱させる．
3. ツリーにイベント間隔数のノードを参加させる．
4. 性能評価を行う．
5. 2,3 のイベント間隔数の合計がイベント総数に達するまで，2∼4 の処理を繰り返す．

4.2 評価項目

提案手法の性能を評価するため，シミュレーションを用いて [5] と比較を行った．性能の評価値は以下の項目とする．

4.2.1 配信ツリー平均長

配信ツリー平均長は以下の式 (3) で求める．

$$(\text{配信平均ツリー長}) = \frac{\sum_{j=0}^{n-1} H(j)}{n} \tag{3}$$

n はノード数，H(j) は j ノードの hop とする．

4.2.2 節ノード数

節ノード数は配信ツリーに参加している節ノードの数である．ノード離脱では一般に葉ノードの離脱より節ノードの離脱の方がオーバーヘッドが大きいと考えられる．また，配信ツリー上からは任意のノードの離脱が繰り返し行われると想定される．そのため，節ノード数の少ないツリーの方が手法としては有効である．

4.2.3 スーパノード数

スーパノード数はツリー上に存在しているスーパノードの数であり，初期状態ではソースノード (スーパノード) のみである．実験では，全スーパーノード数と管理率 5% 以上のスーパノード数を求める．管理率とはスーパノードが管理しているノード数の割合である．スーパノード数は少ないスーパノード数で均等に全体のノードを管理する状態が理想である．

4.2.4 影響ノード数

影響ノード数は 1 回の離脱によってストリーミングの影響を受けるノード数のことである．影響ノード数は以下の式 (4) で求める．

$$(\text{影響ノード数}) = \frac{\sum_{j=0}^{e-1} Inf}{e} \tag{4}$$

$$Inf = |Tr_{leave}| - 1 + Inf_{leave} \tag{5}$$

$$Inf_{leave} = \begin{cases} Inf_{alt} & N_{alt} \in R \\ 0 & N_{alt} \notin R \end{cases} \tag{6}$$

$$Inf_{alt} = \begin{cases} 0 & Tr_{alt} \in Tr_{leave} \\ |Tr_{alt}| & Tr_{alt} \notin Tr_{leave} \end{cases} \tag{7}$$

e は節離脱イベント数，N_{id} は id のノード，R は離脱ノードの代わりとなる条件を満たす代替ノードの集合，Tr_{id} は N_{id} をルートとする部分木，$|Tr_{id}|$ は Tr_{id} を構成するノード数を示す．また，alt, leave はそれぞれ代替ノード, 離脱ノードの id を表す．

本節では，シミュレーション実験の条件や評価方法，評価項目について述べた．次節では，提案手法と既存手法の比較実験の結果と考察について述べる．

4.3 実験結果

4.3.1 配信ツリー平均長

図 24 に配信ツリー平均長の実験結果を示す．[5] と比較して，配信ツリー平均長は約 34% 短く保つことが分かった．

表 4 に提案手法の離脱の割合を示す．葉 (節) ノード離脱は全離脱のうち葉 (節) ノードが離脱された割合，代替葉，代替節，再参加はそれぞれが節離脱で適用された割合を示す．提案手法の離脱は再参加が行われた場合，配信ツリー長が極端に長くなる場合がある．しかし，表 4 の実験結果から再参加が行われることはほとんどないことが分かった．再参加が行われる場合は，例えば，以下の条件を満たす場合である．

- 代替ノードリストから最大 degree が 5 の代替ノードが存在しない．
- 最大 degree が 5 で接続可能な節ノードが存在しない．

この状況で，配信ツリー上から degree が 5 の節ノードが離脱した場合である．しかし，葉ノード数は全ノードの約 70% 存在する (図 25) ことや離脱によって接続可能なノードが増加することから，十分にノード数が配信ツリーに存在している場合にはそのような状況になる可能性が低い．このことから再参加がほとんど行われなかったと考えられる．また，再参加が行われた場合でも，代替ノードを用いた離脱手法や参加手法によって，早い段階で短く保たれると考えられる．

表 4　離脱の割合 (%)

葉ノード離脱	節ノード離脱		
	代替葉	代替節	再参加
71.4	28.0	0.6	0

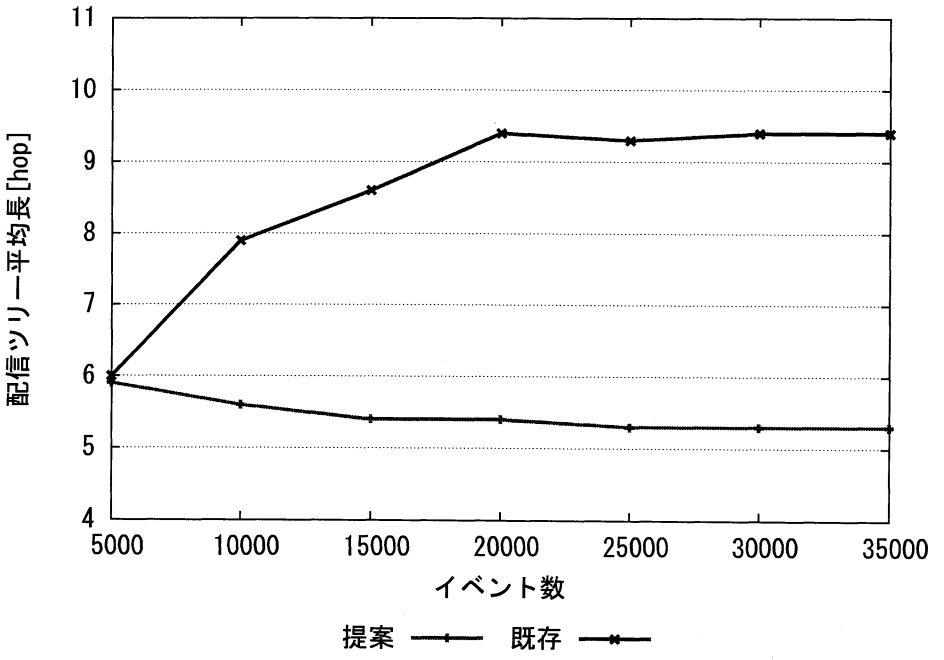

図 24　配信ツリー平均長

4.3.2　節ノード数

図 25, 26 に節ノード数，最大 degree 別の配信ツリー平均長の実験結果を示す．[5] と比較して，節ノード数は約 9.6% 低く保たれることが分かった．

提案手法の方が低く保たれた要因としては，3.3 節の leave アルゴリズムが関係している．提案手法では参加離脱が繰り返し行われると上流に最大 degree の大きいノードが集まる傾向にあることが図 26 から分かっている．そのため，節ノード数が減少傾向にあると考えられる．

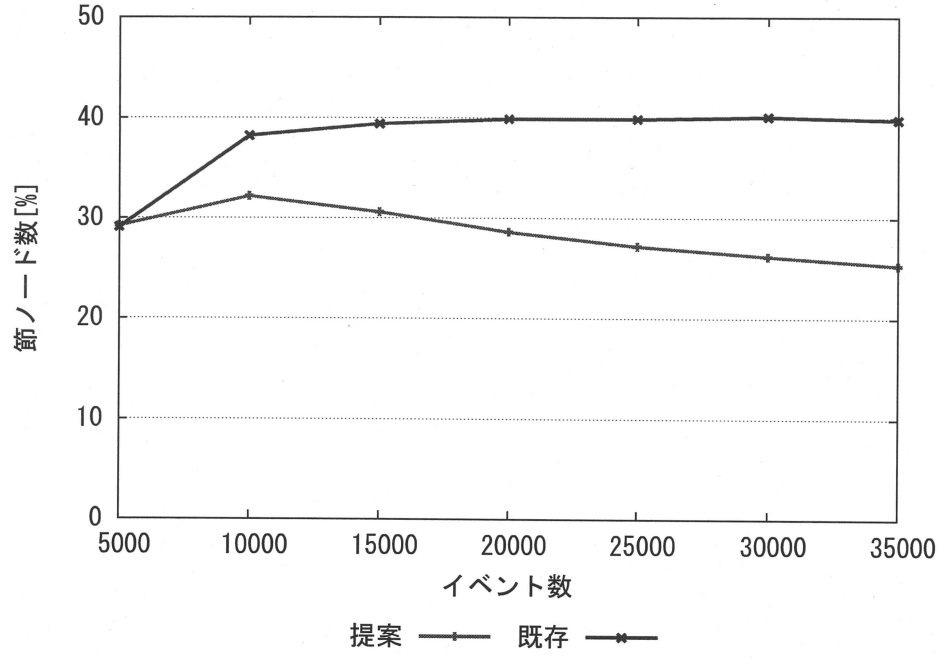

図 25　節ノード数

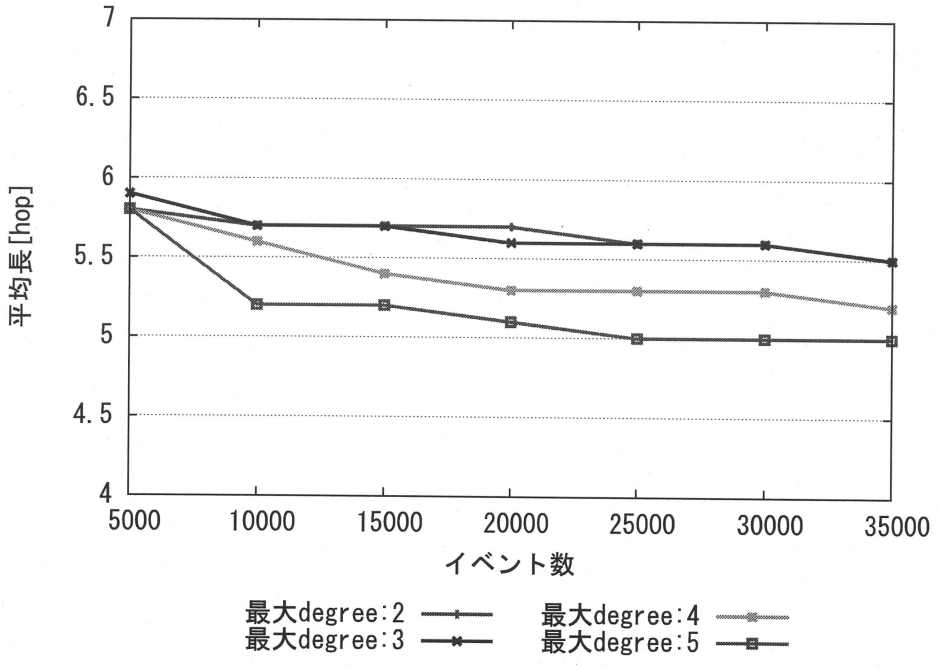

図 26　最大 degree 別の配信ツリー平均長

4.3.3 スーパノード数

図 27 にスーパノード数の実験結果を示す．図 27 から約 3.6% のノードに負荷が分散されることが分かった．これは管理率 5% 以上のスーパノード群が全体の約 90% のノードを管理しているという実験結果 (図 28) から．管理率 5% 以上のスーパノード群が全体のノードを管理していると言えるからである．

また，今回の実験では 5000 ノードで管理数が 100 であるので，理論上の理想的なスーパノード数は全体の 1% となる．しかし，現状は全体のスーパノード数は約 10% あることから，無駄なスーパノードができていることがわかる．そのため，無駄なスーパノードを減らす必要があることが分かった．

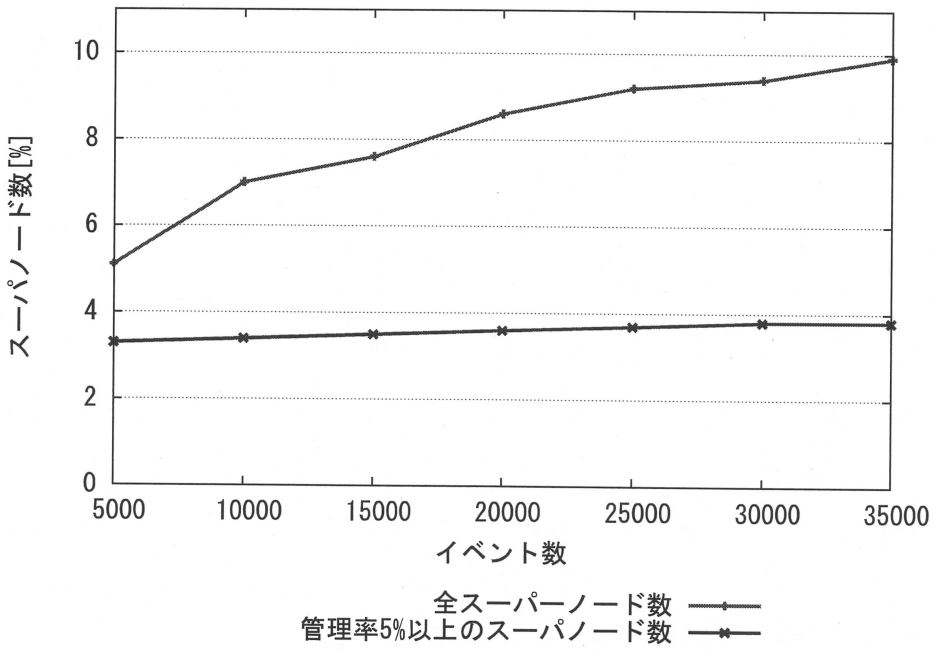

図 27　スーパノード数

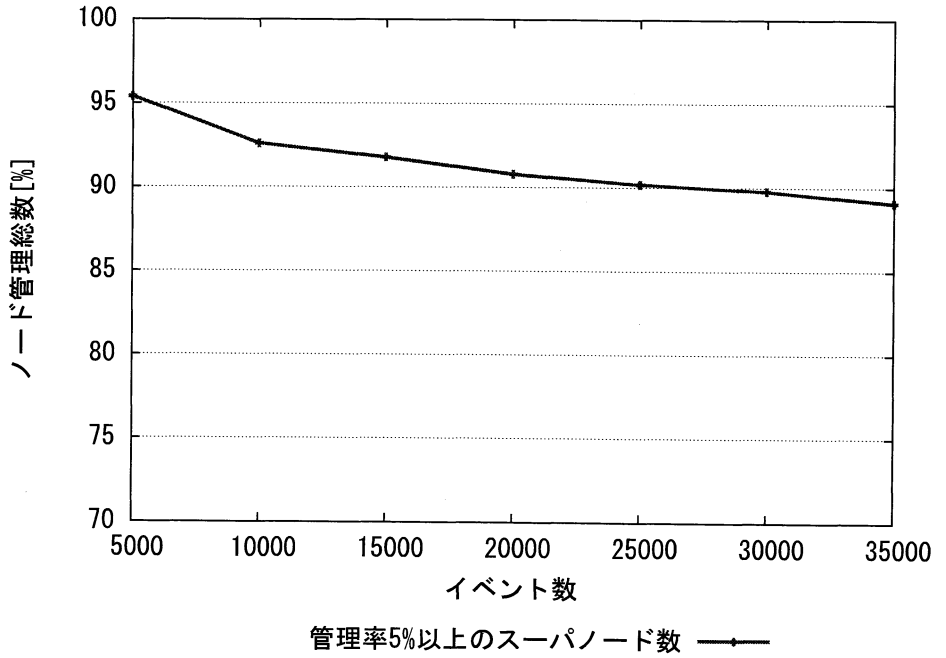

図 28　管理率 5% 以上のスーパノードのノード管理総数の割合

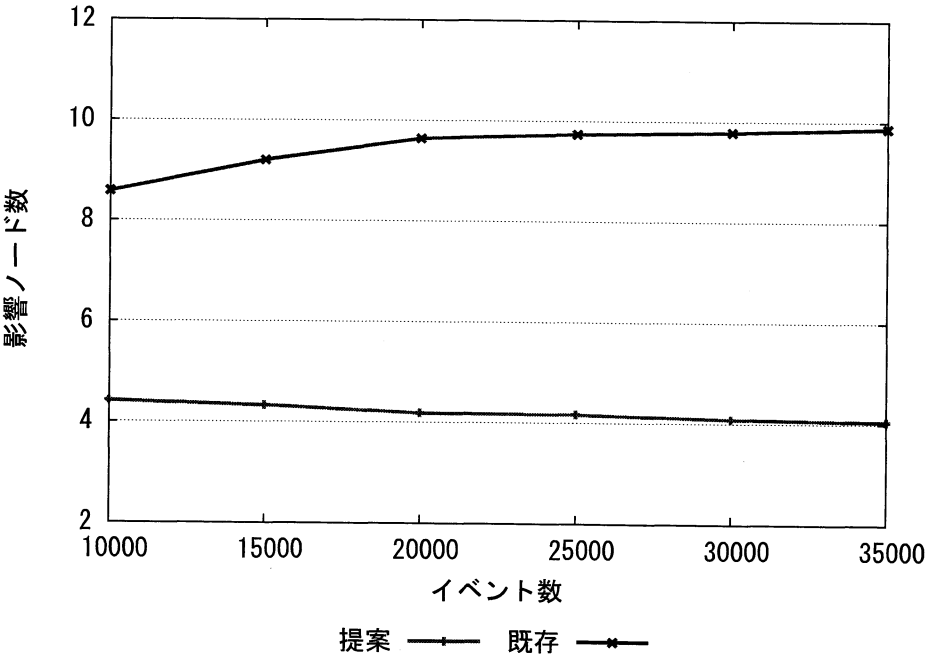

図 29　影響ノード数

4.3.4 影響ノード数

図 29 に影響ノード数の実験結果を示す．[5] と比較して，影響ノード数は約 55% 低くなることが分かった．提案手法は離脱時に本来は影響を受けないノードを代替ノードとして再構築することがある．しかし，参加離脱を繰り返すことにより，配信ツリー平均長が短くなり，節ノード数が少なくなることから，葉ノードの離脱が増える (図 25)．葉ノードの離脱は他のノードに影響を与えないため，影響ノード数が低くなったと考えられる．

本章では，提案手法と既存手法 [5] の比較実験の結果と考察について述べた．実験結果から，既存手法と比較して参加離脱が繰り返し行われても配信ツリー長が短く保たれ，影響ノード数が既存手法より約 55% 低くなることが分かった．また，特定の 1 ノードに情報が集中していたために起きていた負荷が約 3.6% のノードに分散されることも分かった．

おわりに

本論文では，代替ノードを用いた離脱手法にSN-ALMを適用した手法の提案を行った．提案手法では，スーパノードが再構築の際に離脱ノードの代わりとなるノード(代替ノード)を検索し，再構築を行った．また，シミュレーションによる既存手法との比較を行った結果，大規模なノード数で十分に参加離脱を繰り返し行っても，既存手法と比較して配信ツリー長を短く保ち，離脱時に影響を受けるノード数を減らせることが分かった．また，SN-ALMを適用することで，ソースノードの負荷がどの程度分散されるかが分かった．

今後の展望

本論文では，再構築時間について求めていないため，求める必要がある．しかし，シミュレーションでは信頼性のある値を求めることが難しいため，本論文では求めていない．そのため，今後の課題としては実環境で大規模なノード数による評価を行い，比較を行いたいと考えている．また，実験結果からスーパノード数が僅かずつではあるが，参加離脱を繰り返すことによって，上昇傾向にあることも分かっている．したがって，その上昇を抑えることも無駄なパケットを飛ばさないために必要となる．

謝辞

日ごろから多くの御指導を頂きました太田義勝教授，鈴木秀智准教授に深く感謝いたします．そして，日頃何かとお世話になりました落合美子事務員に感謝いたします．また，本論文作成にあたって特にお世話になりました太田義勝教授に深く感謝いたします．最後に，日頃から熱心に討論して頂いた研究室の諸氏に感謝いたします．

参考文献

[1] http://www.google.co.jp/

[2] H. Deshhpande, M. Bawa, H. Garcia-Molina, "Streaming Live Media over Peers.", Technical Report 2002-21, Stanford University, Mar. 2002.

[3] M. Yang. Z. Fei. "A Proactive Approach to Reconstructing Overlay Multicast Trees.", in Proceedings of IEEE INFOCOM 2004, Mar. 2004.

[4] T.Kusumoto, J.Katto, and S. Okubo. Proactive route maintenance for tree-based application layer multicast and its implementations. IEICE TRANS.INF.SYST., 2006.

[5] 高木 健士, 北 望, 重野 寛, "オーバーレイネットワークにおける複数の予備経路を利用した経路再構築手法の検討", 情報処理学会研究報告, Vol.2007, No.58(20070606) pp. 37-42, 2007.

[6] 吉川達, 山本幹, "接続安定性を考慮したALMツリー構築法", 電子情報通信学会技術研究報告, NS, IEICE technical report 105(470) pp.45-48, 2005.

[7] 吉川達, 山本幹, "接続安定性を考慮したALMツリー構築法の改良", 電子情報通信学会技術研究報告, NS, IEICE technical report 105(627) pp.89-92, 2006.

[8] K.Sripanidkulchai, B.Maggs, and H.zhang, "An Analysis of Live streaming Workloads on the Internet", IMC'04, oct 2004, Taormina, Sicily, Italy.

[9] E.Veloso, V.Almeida, W.Meira, A.Bestavros, and S.Jin, "A Hierarchical Characterization of a Live Streaming Media Workload", in Proceedings of Internet Measurement Workshop(IMW), Nov 2002.

[10] 小口敦司, 中里秀則, 富永英義, "複数ツリー型ALMシステムにおけるツリー構築手法に関する一検討, " 電子情報通信学会NS研究会, NS 2005-54, Jun. 2005.

[11] 小口敦司, 中里秀則, 富永英義, "独立複数ツリー型分散ALMストリーミングシステムの提案, "電子情報通信学会技術研究報告, NS, IEICE technical report 106(492), pp.35-40, 2007.

[12] 涌井道子, 永吉功, 花村剛, 富永英義, "スケーラブル動画像伝送における独立データ構造に関する研究, " 画像符号化シンポジウム PCSJ2003, p-2.01, pp.13-14, Nov, 2003.

[13] 岩崎 侑希, 肝付 兼次, 三好 匠, "スーパノードを利用したP2Pストリーム配信手法", 電子情報通信学会総合大会講演論文集, No.2(20060308)p.68, 2006.

[14] 谷元勇太, 太田義勝, 鈴木秀智, "代替ノード検索を用いたALMツリー再構築手法", 情報処理学会第71回全国大会, 3U-2, 2009.3.