

## ダイジェスト視聴可能な P2P ライブストリーミングの提案

鈴木 駿介<sup>†a)</sup> (学生員)      末田 欣子<sup>†,††</sup> (正員)

多田 好克<sup>†</sup> (正員)

Proposal and Evaluation of P2P live streaming system for digest viewing

Shunsuke SUZUKI<sup>†a)</sup>, Student Member, Yoshiko SUEDA<sup>†,††</sup>, and Yoshikatsu TADA<sup>†</sup>, Members

<sup>†</sup> 電気通信大学 大学院情報システム学研究科 〒182-8585 東京都調布市調布ヶ丘 1-5-1

Graduate School of Information Systems, The University of Electro-Communications 1-5-1, Chofugaoka, Chofu-shi, Tokyo, 182-8585 Japan

<sup>††</sup> 日本電信電話株式会社 NTT ネットワーク基盤技術研究所 〒180-0012 東京都武蔵野市緑町 3-9-11

NTT Network Technology Laboratories, Nippon Telegraph and Telephone Corporation 3-9-11, Midori-Cho, Chofu-shi, Tokyo, 180-8585 Japan

a) E-mail: shunsuke@sipa.is.uec.ac.jp

あらまし P2P (peer to peer) におけるライブストリーミングは配信者への負荷の集中を軽減できたり、サーバクライアント方式よりも低コストで実現できるなど、近年その需要は高まっている。しかし、既存の P2P ライブストリーミングシステムでは、途中参加したユーザはそれまでの配信内容を把握することができないといった問題がある。そのため、P2P ライブストリーミングにおいて、途中参加したユーザがダイジェストを視聴可能な P2P ライブストリーミングシステムが必要となる。本研究では、そのような P2P ライブストリーミングシステムにおいて、P2P ネットワーク内でダイジェストを作成し、保持し、広めるために各ピアが役割を持ったトポロジを提案する。シミュレーションの結果、ノード数が増加してもネットワーク全体のスループットは低下しなかった。さらに、役割を持たせなかった場合と比べ役割を持たせた場合は全体的なスループットが大きく、役割を与えることの有用性を確認することが出来た。

キーワード P2P, ダイジェスト, トポロジ, 役割

### 1. 背景

ネットワーク技術の発達に伴い、ネットワークを用いた動画配信サービスが急速に普及している。特にニコニコ生放送 [1] や USTREAM [2] といったリアルタイム動画配信サービスが人気である。動画配信の形態としてはサーバクライアント方式が一般的であるが、配信サーバのコスト削減や配信者の負荷を軽減するため P2P を利用したライブストリーミング配信が期待

されている。

P2P ライブストリーミングは主にアプリケーションレベルマルチキャスト (ALM: Application Level Multicast) によって行われる。ALM はアプリケーションによって実現されるため開発が容易で多くの研究がなされている。

ALM は主にツリー型とメッシュ型に分類される。ツリー型は遅延が少なく構築が容易であるというメリットがあるが、耐故障性やノードの離脱に弱いといったデメリットがある。メッシュ型は耐故障性やノードの離脱に強いというメリットがあるが、複数の経路を用いるため遅延が大きくなるといったデメリットがある。一方でこれらツリー型やメッシュ型の欠点を補うために複数クラスタ型 [3], [4] が提案されている。複数クラスタ型は複数のクラスタを構築し各サブストリームをそれぞれのクラスタで配信するといった方法である。複数クラスタ型ではツリー構造のように中継ノードにストリームを渡すため、配信の負荷が軽減される。またクラスタ内部では各ノードが複数の経路を持つため耐故障性に優れる。

また、動画配信サービスにおいてダイジェストを見ることは、動画全体の雰囲気を知るために有用である。リアルタイム動画配信であるライブストリーミングにおいても、途中から配信に参加した場合にそれまでの配信の内容を把握出来るという点においてダイジェストを見ることは有用である。しかし、P2P のライブストリーミングにおいては、ダイジェストを保存しておくサーバを用意することが出来ないなどの問題がある。また、野球など特定のスポーツ映像にのみ対応可能なダイジェスト生成方式 [5], [6] などもあるが、様々な内容の動画配信に対応させることは難しい。

そこで本研究では、配信に途中参加したユーザがダイジェスト視聴可能な P2P ライブストリーミングシステムにおいて、P2P ネットワーク内でダイジェストを生成し、保持し、広めるために各ピアが役割を持ったトポロジを提案する。ダイジェストを作成する専用のサーバは設置せず、ネットワーク内に存在するピアが作成し、それをネットワーク内に広める。システムはダイジェストが P2P ネットワーク内で枯渇しないような設計にする必要がある。

### 2. 関連研究

#### 2.1 階層型クラスタ構造

Yang Guo らは階層的クラスタ HCPS [7] を提案している。HCPS はクラスタ間で帯域のバランスを取る

ことにより、より品質の高い映像を流すことを可能にしている。クラスタ内部は完全結合となっており、各クラスタのアップロード容量が均一になるように構成されている。ヘッドノードのアップロード容量は大きくなっている。

HCPS はクラスタの中身が完全結合のため各ノードへのホップ数や遅延が多くなってしまうという課題がある。

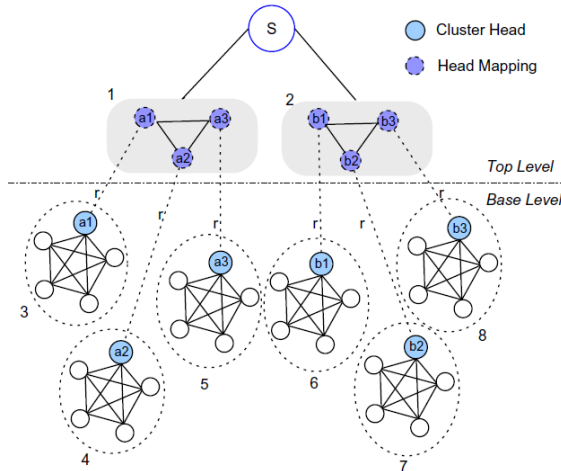


図 1 階層型クラスタの P2P ライブストリーミング  
Fig. 1 A hierarchical cluster in P2P live streaming

## 2.2 重畳クラスタ木型動画配信システム

元橋らは重畳クラスタ木方式の動画配信システム [8] を提案している。階層的なクラスタ構造になっており、クラスタ間の配信木の中継ノードであるゲートノードが存在する。ゲートノード選出方式として、滞在時間の長さや RTT を考慮した方法を提案している。離脱耐性向上型、配信遅延抑制型、ハイブリッド型の実験をしたところ、ハイブリッド型が最も性能が良いという結果が出ている。

重畳クラスタ木方式は全体として木構造のため下位のクラスタほど配信者からのホップ数が大きくなってしまうという課題がある。

## 2.3 ハイブリッドアーキテクチャ構造

Huey-Ing Liu らは局所性と貢献度を考慮したハイブリッドアーキテクチャである MeTree [9] を提案している。これはツリー構造とメッシュ構造のハイブリッドである。貢献度の高いピアには質の高い映像を配信し、貢献度の低いピアには質の低い映像を配信する。異なる貢献度を持つピアに異なる QoE (Quality of

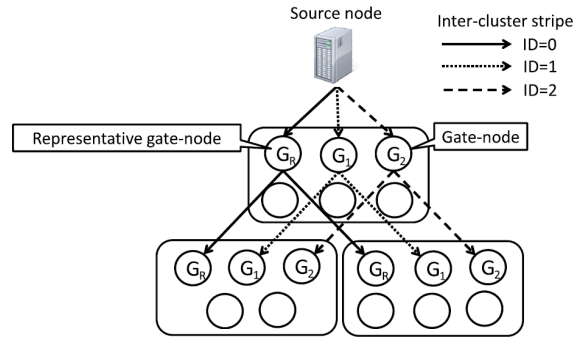


図 2 階層型ストライプ配信  
Fig. 2 A striped broadcast

Experience) を与えているが、それ故ネットワーク全体の QoE は低下してしまう。

MeTree では貢献度を意識した設計のため貢献度の低いノードは良い映像が見られずネットワーク全体の QoE は低下してしまうという課題がある。

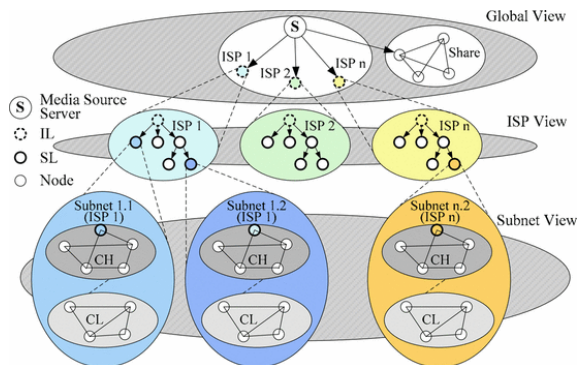


図 3 MeTree  
Fig. 3 MeTree

### 3. 提案システム

本研究の目的は、配信に途中参加したユーザがダイジェスト視聴可能な P2P ライブストリーミングシステムにおいて、P2P ネットワーク内でダイジェストを生成し、保持し、広めるために各ピアが役割を持ったトポロジを設計することである。本研究の要求条件の 1 つは遅延や離脱耐性を考慮した階層型クラスタにすることである。もう 1 つの要求条件はネットワーク内のダイジェストを枯渇させないことである。要求条件を達成するためにクラスタ内部の論理ホップ数は少なくなるように配置する。またダイジェストはクラスタ内ですべての種類が揃っているようにする。こうすることでダイジェストを取得するまでの論理ホップ数を軽減させることが出来る。

#### 3.1 配信内容に対するコメント

本研究で提案する P2P ライブストリーミングシステムでは、参加ユーザが配信内容に対して自由にコメント投稿が出来ることを想定する。コメントは 1 ユーザにつき何回でも投稿出来るものとする。各ユーザが投稿したコメント数はシステム内で管理し、各ノードの役割を決定するために指標として利用する。

#### 3.2 ノードの役割

3 つの役割を各ノードに与える。

**ゲートノード:** 配信者ノードから受け取ったストリームの情報をクラスタ内で一番最初に受取り、それを広める役割を担う。これは配信者ノードの負担を軽減し、素早くクラスタ内に広めるために必要である。高帯域でコメント数の多いピアが選ばれる。

**セミゲートノード:** ゲートノードから受け取ったデータをクラスタ内に広める役割を担う。さらにゲートノードが離脱した場合には即座にこのピアがゲートノードの代わりになることを考えている。ゲートノードの次に高帯域でコメント数の多いピアが選ばれる。

**ダイジェストノード:** ダイジェストを保有し、新規参加ノードに渡す役割を担っている。コメント数が多いピアが選ばれる。これはコメント数の多いピアは離脱耐性が低いと考えられ、ダイジェストの枯渇を防ぐ狙いがある。

また、トラッカーサーバを用いることとする。各ノードのコメント数や帯域幅は常にトラッカーサーバが計算し、管理する。

#### 3.3 ノードの役割決定方法

ノードの役割はダイジェストノード、ゲートノード、セミゲートノード、ノーマルノードの順番で決定する。

まずダイジェストノードはコメント数の多い全ノードの上位 20% を選択する。その後ゲートノードは、全ノードからダイジェストノードを除いたうち帯域幅上位 10% を選択する。そしてセミゲートノードは、全ノードからダイジェストノードとゲートノードを除いたうち帯域幅上位 10% ~ 20% を選択する。最後にノーマルノードは、全ノードからダイジェストノードとゲートノードとセミゲートノードを除いたノードとする。

#### 3.4 ノード間の接続方法

まず、クラスタ外部での接続方法を示す。配信者ノードは全てのゲートノードと接続する。1 つのゲートノードは他のクラスタのゲートノード 1 つずつと接続することとする。

次にクラスタ内部での接続方法を示す。1 つのゲートノードは他の全てのゲートノードと接続する。また対応するセミゲートノード 1 つずつと接続する。1 つのセミゲートノードはダイジェストノードの 3 分の 1 と接続する。またノーマルノード 2 つずつと接続する。1 つのダイジェストノードはダイジェスト未取得ノーマルノード全てと接続する。ノーマルノードはノーマルノードの 4 分の 1 と接続することとする。

#### 3.5 提案するトポロジの全体イメージ図

提案するトポロジの全体イメージを図 4 に示す。配信者ノードから各クラスタのゲートノードにストリームの情報を送り、各ゲートノードがその情報をクラスタ内に広める。

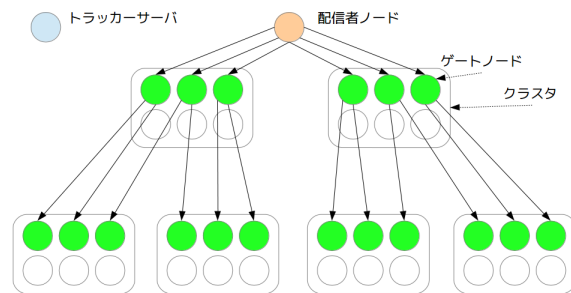


図 4 提案するトポロジ全体イメージ図

Fig. 4 The whole image of our proposal topology

クラスタの内部構造を図 5 に示す。図 5 は図 4 における枠で囲ったクラスタ部分の詳細イメージである。

#### 3.6 新規参加ピア

新規参加ピアの行動について述べる。新規参加ピアはネットワーク接続時に配信者ノードにつなぐ。その後、トラッカーサーバが新規参加ピアの帯域や各クラ

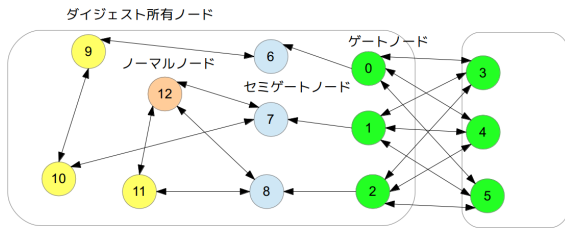


図 5 提案するクラスタ構造

Fig. 5 The clustering structure of our proposal system

スタへの論理ホップ数を計算し、最も論理ホップ数の小さいクラスタに配置される（もしくは RTT の小さいクラスタに配置される）。クラスタに配置された後、ダイジェスト保有ピアからダイジェストを受け取り、ダイジェストの映像を見ることが出来る。

### 3.7 再構築のタイミング

再構築のタイミングは 3 つある。1 つ目は新規参加ピアがダイジェストを取得し終わった時である。ダイジェスト保有ノードと接続を切断したあと、帯域に応じて接続数を決定する。帯域が十分に高い場合にはセミゲートノードになる。2 つ目はコメント数が一定数を越えた時である。この場合、対象のノードはダイジェスト保有ノードになる。3 つ目はノードが離脱した時である。離脱ノードに接続していたピアは帯域に応じた接続数を維持するように他のピアと接続する。特にゲートノードが離脱した場合はセミゲートノードが即座にゲートノードになる。また、特にセミゲートノードが離脱した場合はクラスタの中で帯域が高いピアがセミゲートノードになる。

### 3.8 トポロジの設計の比較

既存のトポロジ設計との比較を表 1 に示す。

表 1 既存システムと提案システムの比較

Table 1 Comparison with existing systems and our proposal system

	HCPS	重畳	MeTree	提案
ダイジェスト	なし	なし	なし	あり
クラスタ	完全結合	一部メッシュ	一部メッシュ	一部メッシュ
ゲートノード	高帯域ピア	滞在時間の長いピア	高帯域ピア	高帯域+コメント量の多いピア

## 4. シミュレーション

### 4.1 前提条件

シミュレーションを行うにあたっての前提条件について示す。全ノードには固有の帯域幅を割り当てる。表 2 にそれを示す。

表 2 帯域幅分布

Table 2 Bandwidth distribution

帯域幅 (kbps)	256	320	384	448	512	640	768	1024	1500	3000
割合 (%)	10.0	14.3	8.6	12.5	2.2	1.4	6.6	28.1	1.4	14.9

また全ノードには固有のコメント数を割り当てる。サンプルとして 1 つの動画に対して分析を行った結果を利用する [10]。平均コメント数は 4 で標準偏差は 6.8 である。分析の結果を図 6 に示す。

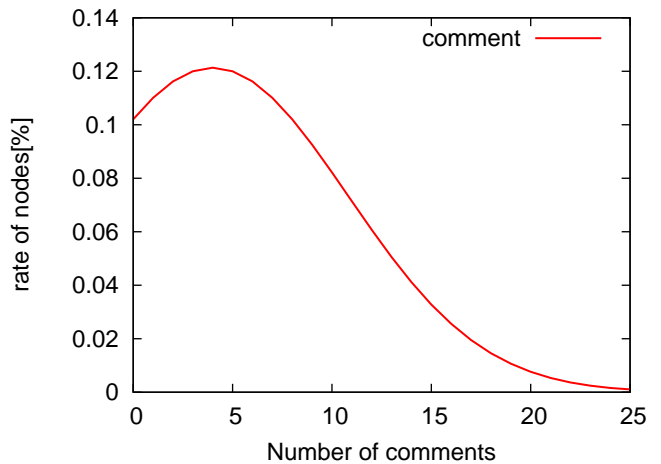


図 6 コメント数分析の結果

Fig. 6 Result of number of comments analysis

固有のコメント数を割り当てた結果を表 3 に示す。

表 3 コメント数分布

Table 3 Number of comments distribution

コメント数	2	5	7	10	12	15	17	20	22	25
割合 (%)	16	17	17	16	15	13	10	2	1	1

全ノード数に応じたクラスタの数を予め設定する。全ノード数とクラスタ数対応表を表 4 に示す。

その他の前提条件として、ノード間遅延を全てのノード間で 100ms に設定している。またパケットロス率は

表 4 ノード数とクラスタ数対応表  
Table 4 Number of nodes and clusters distribution

ノード数	200	400	600	800
クラスタ数	7	10	14	18

0%として考慮しないこととする。

#### 4.2 結 果

シミュレーションで用いたトポロジの例を図 7 に示す。この例ではノード数を 200 とし、クラスタを 7 としている。配信者ノードは 1, ゲートノードは 21, セミゲートノードは 21, ダイジェストノードは 42, ダイジェスト取得済ノーマルノードは 84, ダイジェスト未取得ノーマルノードは 28 でそれぞれ分布している。

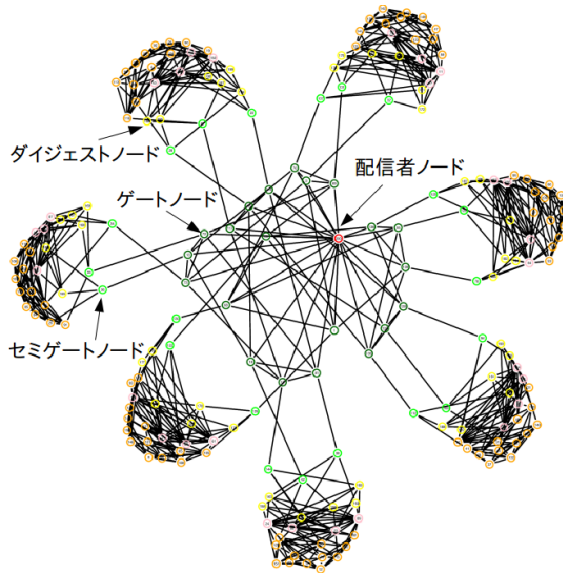


図 7 提案するトポロジ (200 ノード)  
Fig. 7 Proposal topology(200 nodes)

図 8 と図 9 は各ノード数におけるスループット推移の様子である。また図 10 はスループットの推移の平均を示している。他のノード数と比較してノード数が 200 の時にスループットが低下しているが、ノード数が大きくなってもスループットの低下は見られないことがわかる。

さらに役割を与えた場合と与えない場合との比較を行う。ノード数とクラスタ数の対応は表 4 に従うものとする。また役割に関しては、ゲートノード、セミゲートノード、ダイジェストノードを廃止して全てのノードがノーマルノードになるようにする。配信者ノード

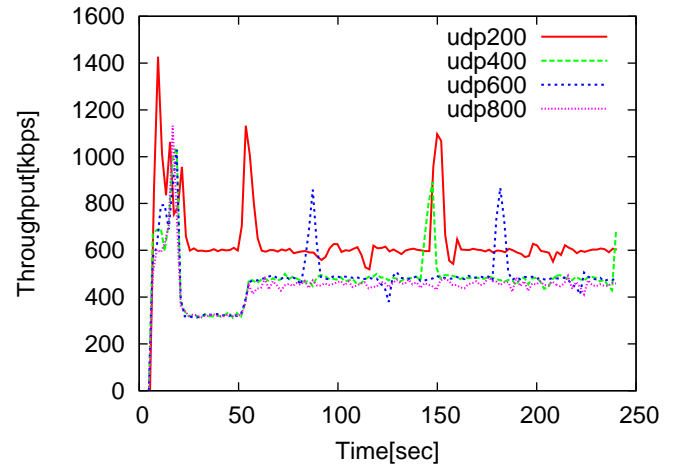


図 8 各ノード数におけるスループット推移の様子  
Fig. 8 Throughput versus node number

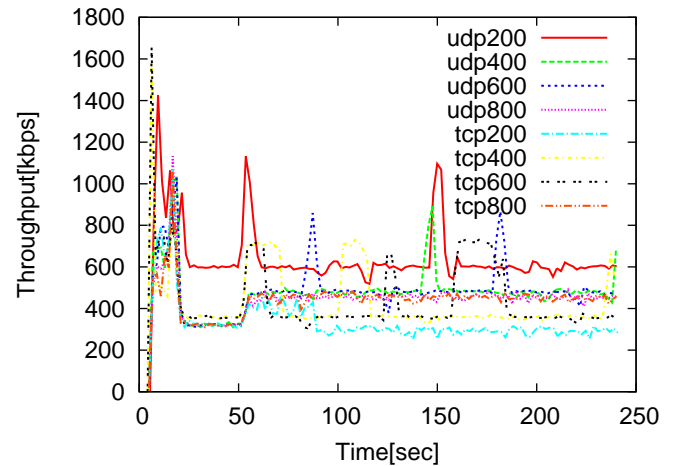


図 9 各ノード数における TCP 通信を含めたスループット推移の様子  
Fig. 9 Throughput versus node number including TCP

からは各クラスタに 1 つずつ繋がるようにする。図 11 は役割を与えなかった場合のトポロジの例を示している。この例ではノード数を 200 とし、クラスタを 7 としている。

図 12 と図 13 は役割を与えない場合の各ノード数におけるスループット推移の様子である。ノード数が大きい場合にスループットが低下している様子がわかる。

図 14 は役割を与えた場合と与えない場合のノード数におけるスループットの推移の様子の平均の比較を



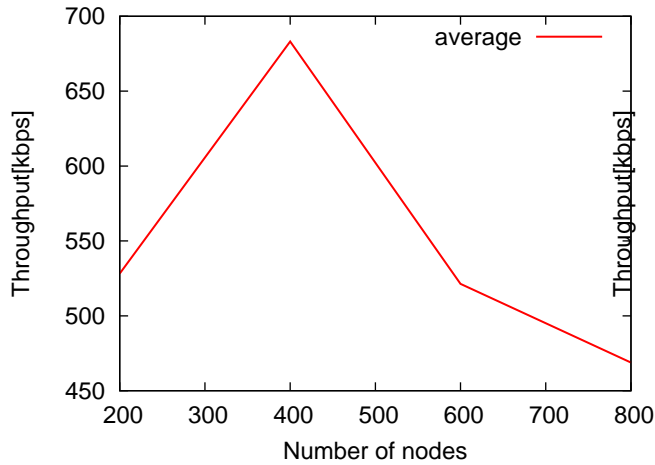


図 10 各ノード数におけるスループット推移の様子の平均  
Fig. 10 Average throughput versus node number

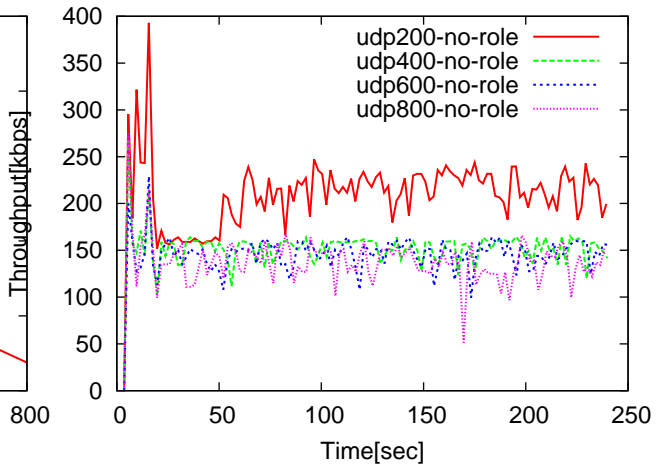


図 12 役割を与えない場合の各ノード数におけるスループット推移の様子  
Fig. 12 Throughput versus node number in case of no role

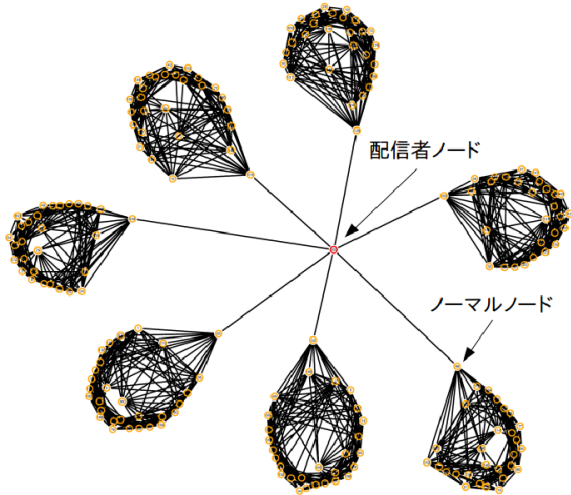


図 11 役割を与えない場合のトポロジ  
Fig. 11 Topology in case of no role

示している。ノード数が大きくなるに連れて、役割を与えない場合はスループットが低下しているのに対し役割を与えた場合はスループットが低下していないことがわかる。さらに役割を与えたほうが全体的なスループットの値が大きく、役割を与えることの有用性が高いことがわかる。

#### 4.3 要求条件の達成度

要求条件の1つは、遅延や離脱耐性を考慮した階層型クラスタにすることであった。シミュレーションで観測できたものはスループットの様子のみであった

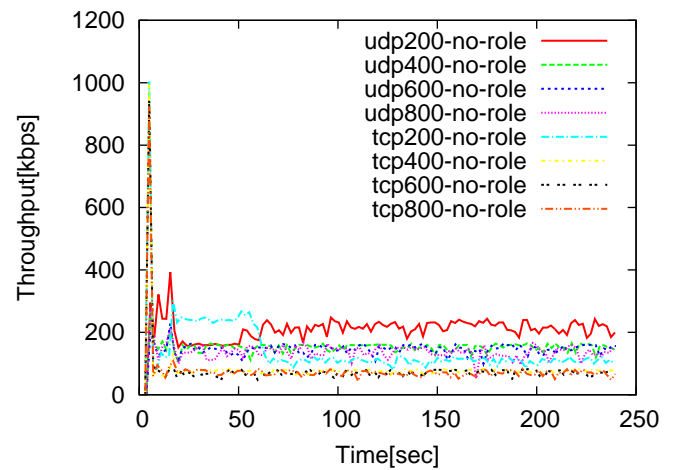


図 13 役割を与えない場合の各ノード数における TCP 通信を含めたスループット推移の様子  
Fig. 13 Throughput versus node number in case of no role including TCP

め、遅延や離脱耐性に対する評価が出来なかった。しかしクラスタの構築は階層型に配置することが出来た。要求条件のもう1つは、ネットワーク内のダイジェストを枯渇させないことであった。これについても、ピアが離脱した場合などを考慮したシミュレーションを行っていないため、評価が出来なかった。しかし、各クラスタにダイジェストを持つノードを配置し、クラスタ内ですべての種類のダイジェストを保持させるこ

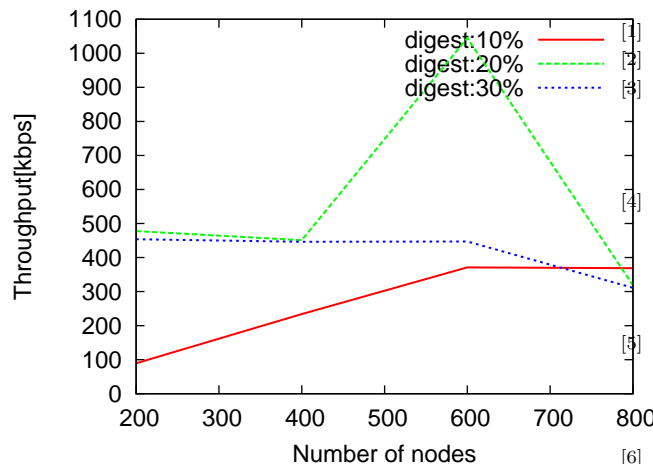


図 14 役割を与えた場合と与えない場合の各ノード数におけるスループット推移の様子との平均の比較

Fig. 14 Comparison with average of throughput in case of role and no role

とは出来た。

## 5. ま と め

本研究では、配信に途中参加したユーザがダイジェスト視聴可能な P2P ライブストリーミングシステムを提案した。トポロジ設計として階層型のクラスタ構造を提案した。さらに各ノードに役割を持たせ、ダイジェストがネットワーク内で枯渇しない設計を行った。さらにシステムは配信者から各ノード数へのホップ数を小さくし、ユーザの体感品質が高くなることを目指した。

シミュレーションの結果、提案システムはノード数が大きくなってもスループットの低下は見られなかった。さらに役割を与えた場合と与えない場合とで比較したところ、役割を与えた場合の方が全体としてスループットの値が大きく、役割を与えることの有用性を確認することが出来た。しかし、2つの要求条件を満たすシミュレーションは出来ず、達成具合を確認することが出来なかった。

今後は、要求条件の達成度を確認するために、配信者ノードからの各ノードヘストリームの情報が到達するまでの遅延情報や、ホップ数情報などの分析を行う予定である。さらに他の P2P ライブストリーミングシステムとの比較を行い、本研究の有用の検証を行う予定である。

謝辞

## 文 献

- [1] ニコニコ生放送, <http://live.nicovideo.jp/>.
- [2] USTREAM, <http://www.ustream.tv/>.
- [3] V. Padmanabhan, H. Wang, P. Chou, and K. Sripanidkulchai, "Distributing streaming media content using cooperative networking," *Proc. NOSDAV '02*, pp.177–186, May 2002.
- [4] G. Bianchi, N. Melazzi, L. Bracciale, F. Piccolo, and S. Salsano, "Streamline: An optimal distribution algorithm for peer-to-peer real-time streaming," *IEEE Trans. Parallel Distrib. Syst.*, vol.21, no.6, pp.857–871, June 2010.
- [5] 熊野雅仁ら, "野球中継のハイライトシーン実時間配信を目的とした特徴のマイニングによる PC シーン of 自動検出", *映像情報メディア学会誌*, vol.59, No.1, pp.77–84, 2005.
- [6] 橋本隆子ら, "スポーツ映像におけるシーン重要度算出アルゴリズムとその評価", 信学会 DEWS2003.
- [7] Yang Guo, Chao Liang, Yong Liu, "Hierarchically Clustered P2P Video Streaming: Design, implementation, and evaluation", *Computer Networks*, pp.3432–3445, 2012.
- [8] 元橋 智紀, 藤本 章宏, 廣田 悠介, 戸出 英樹, 村上 孝三, "多様な配信木により離脱耐性と遅延抑制を向上させる重畳クラスタ木型動画配信システム", *通信技術の革新を担う学生論文特集*, p.132–142, 2014 年.
- [9] Huey-Ing Liu, その他, "MeTree: A Contribution and Locality-Aware P2P Live Streaming Architecture", *AINA 24th IEEE International Conference on*, pp.1136–1143, 2010.
- [10] "ニコニコ動画のコメントを分析してみる@ 2013 年冬アニメ", <http://blog.livedoor.jp/mgpn/archives/51887779.html>, 2014 年閲覧.

## 付 録

### 1.

(平成 xx 年 xx 月 xx 日受付)

**Abstract** Live streaming in P2P(peer to peer) can solve a concentration of load on a distributor, and its demand has increased in recent years. But, users joined since live streaming starts unable to understand the content of a distribution in the existing method. Therefore, a live streaming system that users joined since live streaming starts can watch digests is required. In this study, we propose new topology that peers have roles of creating, keeping and spreading digests in P2P network for such P2P live streaming. As a result of simulation, when the number of peers so increased, the throughput of network did not decrease. We show the peers having role are higher throughput than the peers having not role.

**Key words** P2P, digest, topology, role