

## P2P ライブストリーミングサービスにおける ピアの性能に基づく隣人選択の検討

高山 和幸<sup>†1</sup> 藤本 貴也<sup>†1</sup>  
遠藤 伶<sup>†1</sup> 重野 寛<sup>†2</sup>

P2P ライブストリーミングサービスの中に、各ピアの送信帯域に応じて受信する動画の品質を変化させることで、送信を行う動機付けを行うものがある。このときピアは互いに接続しあい隣人と呼ばれる関係として接続を行う。しかし、既存の隣人選択手法では、高い送信量のピアの隣人として低い送信量のピアのみとなり孤立し、そのピアが本来受けるべき動画の画質よりも低下してしまう問題がある。本稿では、ピアの送信帯域を考慮することで、孤立するピアを無くし、高送信量のピアの受信動画品質差を減らす手法 NSBTB を提案する。NSBTB では、ピアの送信帯域を隣人に通知し、自身よりも送信帯域が多い隣人を優先的に選択する。また、シミュレーション評価により、サービスの参加ピアの合計送信帯域量が合計受信帯域量を下回る状態で、この手法が有用であることを示す。

### Examination of Neighbor Selection based on Peer Transmission bandwidth in P2P Live Streaming Service

KAZUYUKI TAKAYAMA,<sup>†1</sup> TAKAYA FUJIMOTO,<sup>†1</sup>  
REI ENDO<sup>†1</sup> and HIROSHI SHIGENO<sup>†2</sup>

Some of P2P Live Streaming Service encourage peers contribution by changing the quality of video according to peer transmission bandwidth. In this case peer connects each other and call each other as neighbor. However, the existing neighbor selection method has a problem. If high contribution peer have only low contribution peer as neighbor, peer receive video worse than the peer must have. In this paper, We propose NSBTB, the neighbor selection method consider the transmission bandwidth of the peer for eliminating isolation to reduce the difference between the quality of high contribution peer. NSBTB transmission bandwidth will inform neighbors of the peer-selected preference by large transmission bandwidth than the neighbor peer peer neighbor to prioritize high-volume transmission. The simulation results show that this method is

useful when less than the total amount of incoming bandwidth of participating peers total amount of transmission bandwidth of the service.

#### 1. はじめに

様々なアプリケーションにおいて P2P ネットワークを利用した手法が研究されている。<sup>1)2)</sup> P2P モデルのライブストリーミングサービスが現在インターネットのサービスとして広く運用されている。<sup>3)4)</sup> P2P ネットワークでは、接続されたコンピュータをピアと呼んで対等な存在として扱い、動画をお互いに直接やりとりを行う。これにより、配信サーバに集中していた配信コストをピア間に分散でき、サーバの配信負荷を低減し多くのユーザへ配信が可能となる。そのためこのサービスは参加しているピアが送信を行い、貢献することが必要となる。よって、送信量が少ないピアや行わないピアが多くなると、ピアに動画が行き渡らず動画の受信品質が下がってしまう問題がある。

この問題を解決するため、レイヤビデオを導入し各ピアの送信帯域に応じて動画の品質を変化させることで、送信を行う動機付けを行う P2P ライブストリーミングサービスが研究されている。<sup>5)</sup> この手法では、接続された各ピアは互いに接続し隣人とみなす。動画を、チャンクと呼ばれる単位に分割し、それを隣人同士で互いに要求と供給を行うことでチャンクの収集を行い、チャンクを揃えると動画が視聴可能となる。供給の際には、自分に対してチャンクを多く渡した人に優先的にチャンクを供給する。

また動画をレイヤと呼ばれる層に分けることで、チャンクを多く受信すればするほど動画の品質が上昇するようにした。そして、各ピアは自身の送信帯域に合わせてチャンクの要求を行う。送信帯域が低いピアは、チャンクが貰えないことを見込んで優先度の高いレイヤから要求を行う。送信帯域が高いピアは、チャンクが貰えることを見込んで、動画の画質を高める優先度の低いレイヤも一緒に要求を行う。これによって、チャンク送信量が少ない低性能ピアの受信品質を低め、チャンク送信量の多い高性能ピアの受信品質を高めることで、チャンク送信に対する動機付けを実現した。

しかし、既存手法では低性能ピアの数が多くなった際に、隣人をチャンク供給量で選択す

<sup>†1</sup> 慶應義塾大学大学院理工学研究科

Graduate School of Science and Technology, Keio University

<sup>†2</sup> 慶應義塾大学理工学部

Faculty of Science and Technology, Keio University

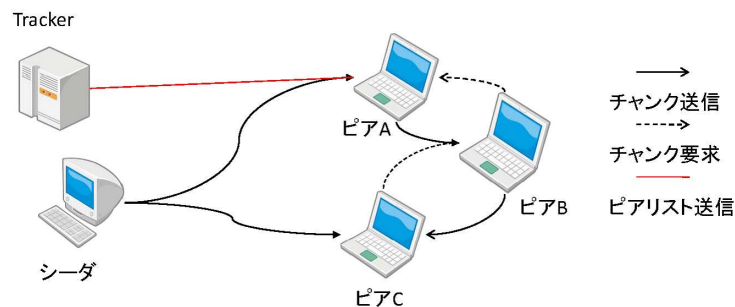


図1 P2P ライブストリーミングサービス

るために、高性能ピアの隣人が低性能ピアのみとなり孤立してしまう場合がある。孤立した高性能ピアは、周りに優先度の高いレイヤしか持たない低性能ピアしかいないために、動画の画質を高める優先度の低いレイヤが手に入らない。そのために、孤立した高性能ピアは画質が低下する。

つまり、高性能ピアは本来受けるべき受信品質を受けることが出来なくなり、同じ送信量の高性能ピアでも動画の品質に差が発生する問題があると言える。そのため、既存手法では、十分な動機付けが出来ているとは言えない。よって、隣人選択により高性能ピアの孤立を無くすことで動機付けをより厳密に行うべきである。

本稿では、ピアの送信帯域を考慮してピアが隣人選択を行うことで、高性能ピアの孤立を無くし、高性能ピア間の動画品質差を減らす手法を提案する。本提案手法では、各ピアが送信帯域を隣人に申告する。そして、ピアが決められた時間ごとに一人の隣人を、その隣人の送信帯域に基づき削除し、新しい隣人の探索を行う。同じ送信帯域を持つ高性能ピアは、性能の低いピアを削除することで高性能ピアの隣人と接続を行うことが出来る。そのため、高性能ピアの孤立を無くし、高性能ピアの品質差を減らすことが出来る。

この手法の有用性を示すために、同時視聴ピア数 5000 ピアで動画の種類が 1 種類のライブストリーミングサービスを想定したシミュレーション評価を行った。

## 2. 関連研究

図1にP2Pライブストリーミングサービスのトポロジを示す<sup>6)</sup>このサービスはシーダ、trackerサーバ、ピアから成り立つ。

- シーダ

動画の配信を行う。動画をエンコードし、チャンクへ分割しランダムにピアへとデータを送信するプッシュを行う。

- trackerサーバ  
参加している全ピアのリストを保持し、各ピアが隣人を検索する際に、そのピアへと全ての参加ピアのリストを渡す。
- ピア  
ピアはユーザとして動画を視聴する。システム参加時に隣人とよばれるデータ交換相手を一定数 Tracker から手に入れたリストを元にランダムに接続する。決められた時間ごとに最もチャンク供給量の低い隣人を入れ替える。ピアは自分の欲しいチャンクを隣人へと要求し、チャンクを手に入れる。また視聴後もチャンクを保持し、隣人の要求に応じてチャンクの供給を行う。

これに文献7),8),9)では、それぞれのピアの送信量にあわせて画質を変化させることにより、ピアに送信を行う動機を持たせた。文献5)においては、レイヤビデオを導入することによりこれを実現している。

図2に、レイヤ数  $n$ 、1レイヤあたりのビットレート  $B$  kbps レイヤビデオについて示す。レイヤビデオとは、動画の中にレイヤと呼ばれる層を設け優先順位を付けたものである。従来では、動画に含まれる全てのチャンクを収集する必要があった。しかし、レイヤビデオの導入によりチャンクを全て集めることが可能な場合は高いビットレートにより高い品質の動画を受信できるようになった。もし、チャンクをなんらかの理由により集める事が出来なかった場合でも、低いビットレートで低い画質の動画を視聴する事が出来る。これにより、サービスの差別化を図ることが可能となった。

レイヤビデオを導入したライブストリーミングサービスでは、これをピアに送信を行わせる動機としてレイヤビデオを導入した。このサービスでは、ピアが隣人の中で最も多くのチャンクを渡してくれた隣人に対して多くのチャンクの送信を行う。よって、より多く送信を行うピアはより多くチャンクを収集出来る為に、受信出来る動画の画質が向上する。逆に、少なく送信を行うピアはチャンクを収集することが出来ずに、受信出来る動画の画質が低下する。

レイヤビデオを導入したP2Pライブストリーミングでは、動画をレイヤと呼ばれる層に分けることによってチャンクを多く受信すればするほど動画の品質が向上する。供給の際に、自分に対してチャンクを多く渡した人にチャンクを供給することで、チャンク送信実績が少ない低性能ピアは受信品質が低くなる。対して、チャンク送信実績が多い高性能ピアは

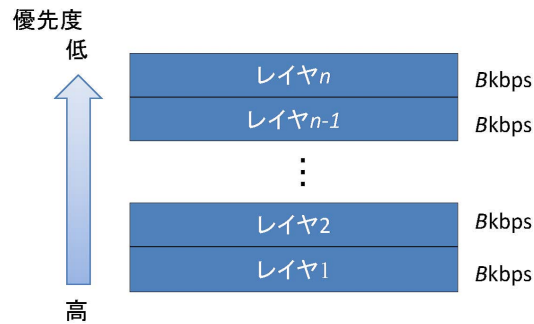


図2 レイヤビデオの層の分け方

受信品質が高くなり動機付けを行った。

しかし、既存手法には、低性能ピアが多くなった際に、隣人選択によって高性能ピアの隣人が低性能ピアのみになってしまう場合がある。この状態を「孤立」と呼ぶこととする。低性能ピアは送信量が少ないために、もらえるチャンク量も少ない。そのため、高性能ピアの欲しい優先順位の低いチャンクを高性能ピアが手に入らない。そのため孤立する高性能ピアが受信する品質は、本来高性能ピアが受信すべき画質よりも低くなってしまふという問題が発生する。よって、この孤立する高性能ピアをなくすことにより、より動機付けを厳密に行うことが出来ると言える。

### 3. 提 案

本稿では、P2P ライブストリーミングサービスにおいて、各ピアの送信帯域に基づいて隣人選択を行うことで、高性能ピアの孤立に対処し、高性能ピアの品質差を減らす手法 NSBTB(Neighbor selection based on transmission bandwidth) を提案する。各ピアは隣人に対して送信帯域を通知し、ピアはその情報に基づいて隣人の選択を行う。その際に、なるべく自分よりも送信帯域の多いピアを選択していくことにより、高性能ピアの孤立を無くし動機付けを厳密に行うことが可能となる。

#### 3.1 NSBTB

NSBTB では、全てのピアは自分よりも送信帯域が少ないピアを隣人から削除することで、送信帯域が多いピアを消去法で隣人選択する。これにより、高性能ピアが低性能ピアに囲まれる状態を無くす。既存手法では、ピアは隣人が自分に対して割り当てているチャンク

量に基づき、チャンクを割り当てている量が少ない隣人を一定時間毎に一人削除する。提案手法では、隣人が互いに自分の送信帯域を通知し、その送信帯域に基づき、自分よりも送信帯域が小さいか一番送信帯域が低い隣人を一定時間毎に一人削除する。どちらの場合においても削除された場合に、新しい隣人をランダムに探索する。しかし、全てのピアが性能の低いピアのみを削除し隣人を選択した場合、高性能ピアの帯域が余剰となった場合、その帯域を活用することが出来ない。そのために、サービスで再生出来る動画の最高ビットレート以上の送信帯域を持つピアは、送信帯域が高い隣人を選択すると同時に送信帯域の低いピアも一定数、隣人に持つ。これによって、高性能ピアの帯域が余剰になった場合、低性能ピアへ帯域を割り当てることが可能になる。より送信帯域の高い隣人を選択する隣人接続をメインリンク、より送信帯域の低い隣人を選択する隣人接続をサブリンクと呼ぶこととする。メインリンクは自身のチャンクを確保するために送信帯域の高いピアを隣人として選択し、サブリンクは高性能ピアの余剰帯域を活用するためにより送信帯域の低いピアを隣人として選択する。

#### 3.2 サービス全体のピア同士の接続状態

図3に、ある一定の時間が経て定常状態になった時の既存におけるピアの接続状態と提案におけるピアの接続状態を示す。既存においては、持っている送信帯域に関わらず持っているリソースの量で隣人選択を行うために、孤立する高性能ピアが発生する。提案において、参加している全てのピアはメインリンクの隣人接続において、自分よりも性能が高いピアを積極的に残そうとする。そのために、ピアは必ず自身よりも高い、あるいは自身と同じ送信帯域のピアを隣人として接続することが可能になる。しかし、自身よりも送信帯域が高いピアは逆に相手側からリンクの切断が行われるために、結果的に自身と同じ送信帯域のピアが集まることになる。

#### 3.3 メインリンクとサブリンクの数の決め方

メインリンクは全てのピアが必ず保持している隣人接続であるが、サブリンクは動画のビットレート以上の送信帯域を持つピアのみが保持する。サブリンクの数は、動画の最高ビットレートに対してピアの送信帯域がどれくらい余っているかの割合で決められる。また、メインリンクとサブリンクの数の和は常に一定である。メインリンクの数  $l_m$  は、式1の計算式で計算される。

$$l_m = \begin{cases} N & (B \geq u) \\ N \frac{B}{u} & (B < u) \end{cases} \quad (1)$$

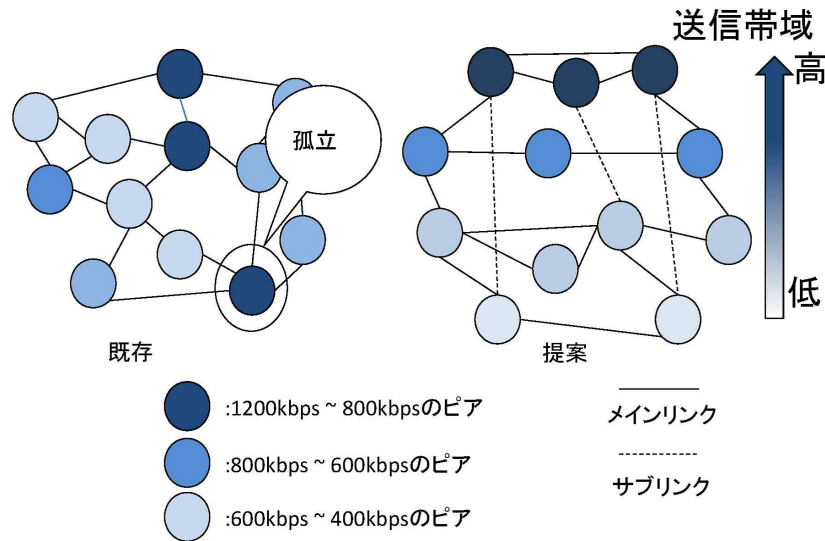


図3 ピアの接続状態

ここで、 $N$  はサービスで決められた隣人接続数で、 $B$  は動画の最高ビットレート、 $u$  は各ピアの送信帯域である。次に、サブリンクの数  $l_s$  は、式2の計算式で計算される。

$$l_s = \begin{cases} 0 & (B \geq u) \\ N(1 - \frac{B}{u}) & (B < u) \end{cases} \quad (2)$$

使われている変数の定義は、式2と全く同じである。以上のように、メインリンクとサブリンクの数を設定することで、動画のビットレートより高い送信帯域を持つピアの帯域が余剰となった時、その帯域を送信帯域が低いピアに割り当てることが可能になる。

### 3.4 メインリンクの隣人選択アルゴリズム

図4に実際にピアがメインリンクの隣人を削除するアルゴリズムを示す。各ピアはメインリンクとして接続している隣人をそれぞれ自身の持つ送信帯域と比較し、自身よりも送信帯域が高く、かつ最も送信帯域が低いピアを隣人から削除する。もし、隣人がすべてピアよりも送信帯域が大きい場合は削除を行わない。

図5に、ピアaにおける既存手法であるチャンク供給量を用いた隣人選択手法と、NSBTBでの隣人選択手法の具体例を示す。既存手法では、一定の間の隣人からピアへのチャンク供

```
//n_u[l]:隣人の送信帯域の配列, l_num:現在のメインリンクの隣人数
//l_m:最大のメインリンクの隣人数, upCapacity:自ピアの送信帯域,
//list[]:隣人リスト,
max = INT_MIN, max_node = -1;
for(i = 0; i < l_num; i++)
{
    up = n_u[i] - upCapacity
    if(up < 0){
        up* = -1
        if(max < up){
            max = up;
            max_node = i;
        }
    }
}
if(max_node != -1)
    delete(list[max_node]);
```

図4 メインリンクの隣人選択アルゴリズム

給量、すなわち一定の間、隣人が自分に割り当ててくれた帯域に基づいて隣人を削除する。この場合だと、ピアaに割り当てている供給量が70KBと最も少ないピアeを隣人から削除を行う。しかし、このピアeの送信帯域は1000kbpsであり、ピアaよりも送信帯域の高いピアにあたる。つまり、既存手法においては、本来、自分よりも高い送信帯域のピアの削除を行っている。この問題はライブストリーミングが実際のリソースに基づきランダムにチャンク要求が行われるために発生する。よって、既存手法では、自ら高性能ピアを隣人から削除し、孤立してしまう問題がある。提案手法においては、送信帯域に基づいて隣人の削除を行う。この場合だと、送信帯域が280kbpsで、ピアaの送信帯域580kbpsよりも低いかつ最も送信帯域の低いピアeの削除を行う。提案手法の場合は、送信帯域に基づいて隣人選択が行われるので、高性能ピアの孤立を無くすことが可能になる。

### 3.5 サブリンクの隣人選択アルゴリズム

図6に実際にピアがサブリンクの隣人を削除するアルゴリズムを示す。基本的には、ほと

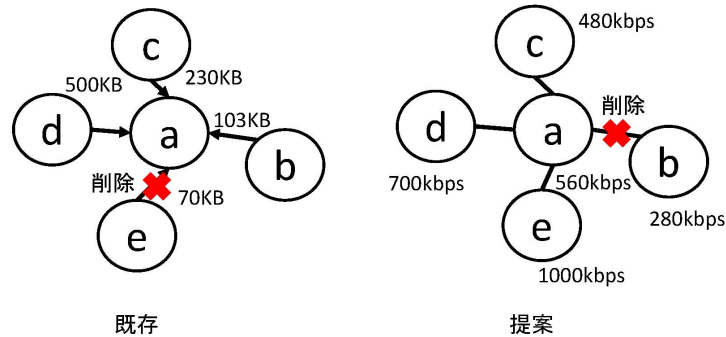


図 5 隣人選択手法の例

```

//n_u[l]:隣人の送信帯域の配列, l_num:現在のメインリンクの隣人数
//l_m:最大のメインリンクの隣人数, upCapacity:自ピアの送信帯域,
//list[]:隣人リスト,
min = INT_MAX, min_node = -1;
for(i = 0; i < l_num; i++)
{
    up = n_u[i] - upCapacity
    if(up > 0){
        if(min < up){
            min = up;
            min_node = i;
        }
    }
}
if(min_node != -1)
    delete(list[min_node]);
    
```

図 6 サブリンクの隣人選択アルゴリズム

んどコードは同じであり不等号の向きを変えただけである。各ピアはサブリンクとして接続している隣人をそれぞれ自身の持つ送信帯域と比較し、自身よりも送信帯域が高く、かつ最も送信帯域が低いピアを隣人から削除する。もし、隣人がすべてピアよりも送信帯域が大きい場合は削除を行わない。

### 3.6 ピアの動作詳細

#### a ピアの参加時の動作

まず、ピアは送信帯域と動画の最高ビットレートを元に式 1、式 2 からメインリンクとサブリンクの数を決定する。ピアは参加する際に、まず tracker に自分の参加を知らせると同時に、tracker からサービスに参加している全ピアリストを取得する。tracker から取得したピアリストの中からランダムにピアを一つ選択し、隣人としての接続を要求するメッセージを送信する。隣人接続要求するメッセージを受信したピアは、隣人の数が一定の数に達していなければ、隣人として接続を行う。隣人接続の際にピアは互いに送信帯域の情報を送信し、送信帯域を隣人に通知する。ピアは、一定の数になるまで、この隣人探索要求を繰り返す。もし、一定の数にならず全てのピアに隣人接続要求を行った場合は、隣人探索を終了する。この時点においては、全ての隣人をランダムで選択し接続している。そのため、送信帯域による隣人選択は全く行われていないので隣人として様々な送信帯域のピアが存在している。

また、すべて隣人選択をランダムで行っているため、隣人がメインリンクとサブリンクであるかどうかランダムに振り分ける。

#### c ピアの隣人削除時の動作

ピアはサービスに参加してから毎一定時間後に隣人削除の動作を行う。メインリンクのみのピアは、節 3.4 にて説明したアルゴリズムで隣人削除を行う。メインリンクとサブリンクを持つピアは、まず節 3.4 にて説明したアルゴリズムでメインリンクの隣人を削除した後に、節 3.5 にて説明したアルゴリズムでサブリンクの隣人の削除を行う。削除を行う場合、ピアは隣人に対して削除を行うメッセージを隣人へと通知し、隣人はそれを受けて該当のピアの削除を行う。

#### d ピアの隣人削除後の動作

ピアは隣人削除後に新しい隣人を探索する。この隣人探索時の動作は、ピアのサービスへの参加時とは違いメインリンクの隣人を探索するフェーズとサブリンクの隣人を探索するフェーズの 2 つに分けることが出来る。隣人を探索するピアはまず、tracker に新しいピアリストの取得を要求するメッセージを送信し、tracker からサービスに参加している全ピアのリストを取得する。tracker からリストを取得したピアは、まずメインリンクにピアの探

索を行う。ピアの探索は、tracker から取得したリストに基づいてランダムに選択したピアに隣人接続を要求するメッセージを送信することで行われる。節 3.6 で設定したメインリンクに数にまでメインリンクの隣人数が達するか、全てのピアに隣人接続要求を行った場合、隣人探索を終了する。メインリンクの隣人探索が終了すると、サブリンクを持つピアの場合は、サブリンクの隣人探索を行う。サブリンクの隣人探索はメインリンクと同様に tracker から取得したサービスに参加している全ピアのリストに基づいてランダムに隣人接続を要求するメッセージを送信することで行われる。節 3.6 で設定したメインリンクに数にまでサブリンクの隣人数が達するか、全てのピアに隣人接続要求を行った場合、隣人探索を終了する。サブリンクとメインリンクは異なる種類の隣人接続として扱っているが、メインリンクとサブリンクの隣人にそれぞれ同じピアが隣人として接続することは無いため、サブリンクの隣人探索の際に、メインリンクの隣人へ隣人接続要求は行わない。

#### e ピアが隣人から削除された場合の動作

ピアはサービスに参加してから、いつでも隣人の削除を受け付ける。もし、隣人から削除された場合は、そのピアが一定時間経て隣人削除の動作に移るまで新しい隣人の追加は行われない。

#### d ピアの離脱時の動作

ピアは、サービスから離脱する際は、メインリンク、サブリンク全ての隣人へと隣人削除メッセージを送信し、隣人をすべて削除する。また Tracker へサービス離脱メッセージを送信を行う。サービス離脱メッセージを受信した Tracker はピアリストからピアを削除する。

## 4. シミュレーション評価

提案手法 NSBTB の有用性を示すために、既存手法における隣人選択<sup>?)</sup> [layer] 比較評価を行う。

### 4.1 シミュレーション条件

本稿では、受信品質の指標としてビットレートをを用い、受信ビットレートを比較することで評価を行う。簡単のために、送信帯域が 1000kbps の高性能ピアと送信帯域が 400kbps の低性能ピア、2 つのピアの参加のみを仮定する。同時視聴ピア数は 5000 ピアの中規模のライブストリーミングサービスを想定している。表 1 にシミュレーションで使用した条件を示す。

シミュレーション時間を 300 秒として、最初の 30 秒でピアが参加を行い、30 秒後から実際の動画ビットレートの計測を 1 秒毎に行った。またシミュレーションは 5 回行い、その

表 1 シミュレーション条件

同時視聴ピア数	5000
隣人数	20 人
隣人更新間隔	10sec
高性能ピア送信帯域	1000kbps
高性能ピア受信帯域	1500kbps
低性能ピア送信帯域	400kbps
低性能ピア受信帯域	1500kbps
動画ビットレート	800kbps
レイヤ数	3
シミュレーション時間	300sec

平均値を算出した。

### 4.2 高性能ピアのビットレート度数分布

図 7 に、高性能ピアの参加割合が 30% 時の全ての高性能ピアのシミュレーション開始後 270 秒目におけるビットレートの度数分布を示す。横軸は受信ビットレート、縦軸はそのビットレートを受信しているピアの個数を表示している。

既存においては、最高で 750kbps を受信するピアがいる一方で最低 575kbps を受信し送信量が同じにも関わらず 175kbps の動画品質差が生じている。しかし、提案では最大で 800kbps から最低で 750kbps と 50kbps に品質差が少なくなっており、目的が達成されていることがわかる平均的なビットレートの向上が見られるのは、周りに低送信量のピアのみで大きく動画品質が低下していたピアだけでなく、良い動画品質を受信していたピアも低送信量ピアにチャンクを取られていたがメインリンクにおいて積極的に低送信量ピアの切断を行うからである。

### 4.3 高性能ピア・低性能ピアの平均受信ビットレート

図 8 に、高性能ピア、低性能それぞれの高性能ピアの参加割合を変化させた時の 30 秒後から 300 秒までの平均受信ビットレートを示す。横軸は、高性能ピアの参加割合。縦軸は、ピアの平均受信ビットレートを表示している。

シミュレーション条件より、高性能ピアの参加割合が 50% において、全ピアの平均送信帯域は 700kbps で全ピアの受信帯域は動画の最高受信ビットレート 800kbps である。つまり、高性能参加割合が 0 ~ 50% 時は、全てのピアの送信帯域の総計が受信帯域の総計を下回っている。

そして、高性能ピアの参加割合が 0 ~ 50% で、送信帯域が不足しており、その時には、

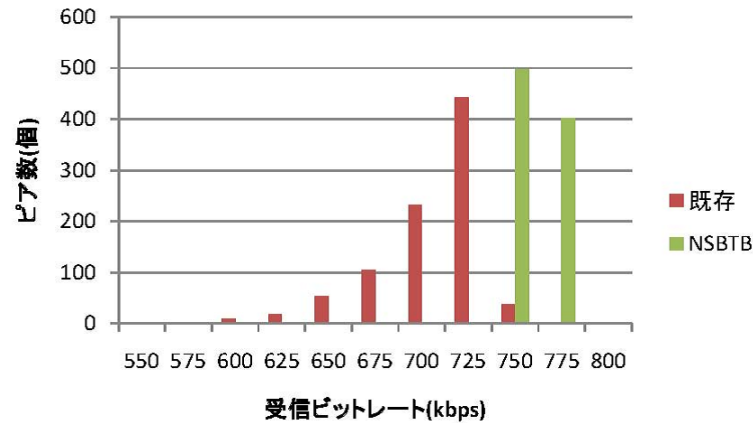


図 7 高性能ピアの参加割合 30% 時の高性能ピアのビットレート度数分布

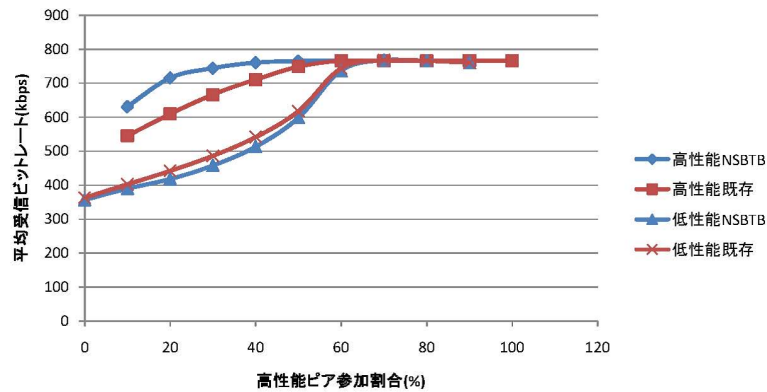


図 8 高性能・低性能ピアの平均受信ビットレート

高性能ピアのビットレートが向上していることがわかる。

#### 4.4 サプリングの有無による平均受信ビットレートの影響

サプリングの有無による影響を考察するために、サプリングがある場合とない場合とで、それぞれ平均ビットレートを計測し評価を行った。サプリングが無い場合は、高性能ピアも

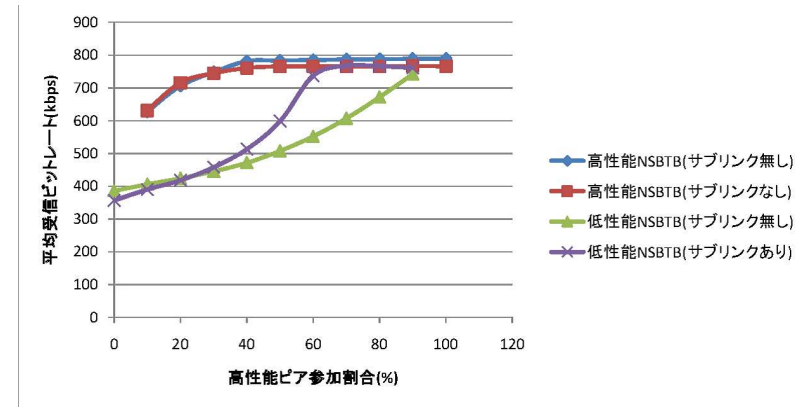


図 9 高性能・低性能ピアの平均受信ビットレート

低性能ピアも全てメインリンクのみで隣人接続を行っている。図 9 に、高性能ピア、低性能それぞれの、高性能ピアの参加割合を変化させた時の 30 秒後から 300 秒までの平均受信ビットレートを示す。横軸は、高性能ピアの参加割合。縦軸は、ピアの平均受信ビットレートを表示している。

高性能ピアの参加割合が 60% 以上、つまり高性能ピアの帯域が余っている時においてサプリングがある場合、低性能ピアの画質の低下が抑制出来ていることがわかる。.. 全ピアの合計送信帯域と受信帯域は変わらないので、高性能ピアの画質向上の分低性能ピアの画質低下が起こるのは避けられないが、サプリングによって低性能ピアに余剰な帯域を分け与えることが出来ている。

#### 5. おわりに

レイヤビデオを導入したライブストリーミングサービスでは、高性能ピアが低性能ピアに囲まれて孤立し本来受信すべき画質よりも低下し、高性能ピア間で受信画質に差が発生してしまう問題点があった。そこで、ピアの送信帯域を考慮することで孤立するピアに対処し、高性能ピアの品質差を減らす手法 NSBTB を提案した。NSBTB では、ピアの送信帯域を隣人に通知し、自身よりも送信帯域が多い隣人を優先的に選択する。自身よりも送信帯域が多いピアを隣人として接続することによって、高性能ピアは高性能ピア同士と隣人となり、高性能ピアの受信画質差を低減する。

また NSBTB を、シミュレーションにより既存手法との比較評価を行った。その結果、

NSBTB は高性能ピアが 50% 以下で、全ピアの送信帯域の合計が受信帯域の合計を下回った時、受信ビットレートの差が 175kbps から 50kbps へと低減した。送信量ピアが同じピアは同じ画質を受信する P2P ライブストリーミングサービスにおいて、高送信ピアの画質差を低減させたことから、本提案方式の有用性を示した。

## 6. 謝 辞

本研究の一部はグローバル COE プログラム「アクセス空間支援基盤技術の高度国際連携」により行われました。

## 参 考 文 献

- 1) 松本敬, 遠藤伶, 重野寛. ブロックのレアルティを考慮した効率的な p2p ファイル共有手法. 情報処理学会論文誌, Vol.51, No.6, pp. 1310–1319, 2010-06-15.
- 2) Takaya Fujimoto, Rei Endo, Kei Matsumoto, and Hiroshi Shigeno. Video-popularity-based caching scheme for p2p video-on-demand streaming. In *Advanced Information Networking and Applications. AINA 2011. The 25th IEEE International Conference on*, pp. pp.748 – 755, March. 2011.
- 3) Pplive. <http://www.pptv.com/>, 2011. PPLive Homepage(2011/04/25 現在).
- 4) Ppstream. <http://www.ppstream.com/>, 2011. PPLive Homepage(2011/04/25 現在).
- 5) Zhengye Liu, Yanming Shen, K.W. Ross, S.S. Panwar, and Yao Wang. Layerp2p: Using layered video chunks in p2p live streaming. *Multimedia, IEEE Transactions on*, Vol.11, No.7, pp. 1340 –1352, nov. 2009.
- 6) Xinyan Zhang, Jiangchuan Liu, BoLi, and Y.-S.P. Yum. Coolstreaming/donet: a data-driven overlay network for peer-to-peer live media streaming. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, Vol.3, pp. 2102 – 2111 vol. 3, march 2005.
- 7) F.Pianese, D.Perino, J.Keller, and E.W. Biersack. Pulse: An adaptive, incentive-based, unstructured p2p live streaming system. *Multimedia, IEEE Transactions on*, Vol.9, No.8, pp. 1645 –1660, dec. 2007.
- 8) Zhengye Liu, Yanming Shen, K.W. Ross, S.S. Panwar, and Yao Wang. Substream trading: Towards an open p2p live streaming system. In *Network Protocols, 2008. ICNP 2008. IEEE International Conference on*, pp. 94 –103, oct. 2008.
- 9) J.D. Mol, D.H.P. Epema, and H.J. Sips. The orchard algorithm: Building multicast trees for p2p video multicasting without free-riding. *Multimedia, IEEE Transactions on*, Vol.9, No.8, pp. 1593 –1604, dec. 2007.