

モバイル端末間協調ライブストリーミングのための 自律的なチャンク損失回避手法

森 研太^{†1} 畠山 翔^{†1} 重野 寛^{†1}

概要: Cellular-WiFi Hybrid Network における端末間協調ライブストリーミングではそれぞれのピアが携帯電話回線でチャンクを取得し、WiFi を用いて周囲のピアへ拡散することで、帯域変動による受信ビデオ品質の低下を防いでいる。しかし携帯電話回線から取得するチャンクを選択は、端末の帯域変動によるチャンクの損失を減らすため帯域の限りランダムに取得しており、チャンクを事前に分担して帯域抑制を行うことは難しい。そこで本論文では、前もって取得するチャンクを分担した上で、適宜帯域不足のピアを検知し別のピアがチャンクを取得するチャンクスケジューリング手法を提案する。本提案手法ではそれぞれのピアがすべてのチャンクに対して優先度付けを行うことで、携帯電話回線によってチャンクをサーバから取得するピアを一意に決定する。また一定時間チャンクの拡散が行われない場合は、先の優先順位を用いて、順次別のピアが該当のチャンクの取得を試みる。シミュレーションの結果、本提案手法は協調ピア数の帯域総和がビデオレートに対して不足がない場合は個別のピアの帯域変動状況に関わらずビデオの受信率 95%以上を実現すると共に、冗長なチャンクのダウンロードは各ピア 5%程度に抑えることを確認した。

キーワード: モバイルアドホックネットワーク, ライブストリーミング, Cellular-Wifi ハイブリッドネットワーク

An Autonomous Error Recovery Algorithm for Cooperative Mobile Live Streaming

Abstract: It is hard to watch live streaming video over an unstable cellular network. A cooperative live streaming using Cellular-WiFi Hybrid Network is bandwidth fluctuation tolerant. It is that each peer gets video chunks via a self cellular network and broadcasts the received chunks to other cooperative peers. The chunk selection algorithm is random and all peers try to download chunks as far as the cellular network permits. However, this random algorithm makes a lot redundant chunks. In this paper, we propose a new cooperative chunk download algorithm that determines who downloads which chunks beforehand, and finds the peer which use poor cellular connection and other peers undertake a task to download chunks adaptively. The point of this proposed algorithm is that each peer gives a random number for all chunks and makes the priority of the peer for a chunk. A peer having a maximum priority for a certain chunk tries to download the chunk on a cellular connection. Besides, if no one gets a chunk after the elapse of a certain period of time, a different peer tries to download the chunk by using the priority. Simulation results show that the proposed scheduling algorithm can reduce cellular traffic and achieve a stable video reception rate.

Keywords: mobile ad hoc network, live streaming, Cellular-Wifi hybrid network

1. はじめに

ビデオをリアルタイムにインターネット上で配信できるサービスとして、ライブストリーミング・サービスがある。

近年のスマートフォンの普及に伴い、モバイル端末を用いてビデオを視聴するユーザの数は増加の一途を辿っており、ライブストリーミングの需要もまた増加している。モバイル端末は一般的に携帯電話回線を用いてビデオコンテンツの視聴を行うが、携帯電話回線の帯域は変動が激しいために高画質な動画コンテンツを連続的に視聴することは困難である [1]。また、個々の端末が独立してビデオコン

^{†1} 現在、慶應義塾大学大学院理工学研究科
Presently with Graduate School of Science and Technology,
Keio University

テンツを要求することで生じる余剰なトラフィックのため、更に携帯電話回線の帯域を圧迫し、ビデオの再生品質に影響を与えるという問題も生じる [2].

そこで携帯電話回線を用いた通信と MANET での WiFi アドホック通信が並行して行われるネットワークである Cellular-WiFi Hybrid Network(CWHN) [3] がある. この CWHN においては全てのピアが自身の携帯電話回線を用いて通信を行うことができると同時に、隣人関係にあるピアからも WiFi を用いてデータの授受が可能である. よって CWHN の特性を用いれば、携帯電話回線からのビデオデータの取得に加え、MANET での隣人ピア間でビデオデータの授受もできるため、1つの端末で利用できる携帯電話回線の帯域以上に高品質なビデオの視聴が可能である.

この CWHN を用いたライブストリーミング・サービスでは高品質なビデオを視聴を実現するためそれぞれの端末ができる限り多くのビデオデータを自身の携帯電話回線を使って取得し、それをすべての隣人ピアに対して拡散する.

このビデオデータはチャンクとという単位で配信されており、もし誰がどのチャンクをダウンロードするというを事前にスケジューリングして各ピアに割り当ててしまうと、割り当てられたピアの回線状況が悪化した際やサービスから離脱した場合に再スケジューリングという再割り当てのためのオーバーヘッドが発生する. 一方で、スケジューリングを行わない場合は、取得するチャンクに重複が頻発するために余剰なトラフィックが発生してしまう.

そこで本稿では、すべてのピアに事前に取得するチャンクを公平に分担しながらも、ピアの帯域不足の際には各ピアが適宜判断して損失しそうなチャンクを取得するというチャンクスケジューリング手法である Autonomous Error Recovery Algorithm for Mobile Live Streaming (AERA) を提案する.

本提案手法では、MANET 上のピアは自身の携帯電話回線で取得するチャンクを事前に分担する. 各ピアはそれぞれのチャンクに対して乱数を生成し、それを MANET に参加するピアすべてにブロードキャストする. 各ピアが生成したこの乱数の大小を比較することでそれぞれのチャンクにピアの優先順位を設ける. この優先順位付けをすべてのチャンクに対して行った上で、最も大きな数値を提示したピアが携帯電話回線から該当チャンクをダウンロードする.

また、携帯電話回線の帯域が低下した際や、ピアが不意に離脱した際には先ほどの優先順位を用いてチャンク損失の回避を行う. このリカバリでは事前に設けたチャンクに対するピアの優先順位の 2 番目以降が、損失しそうな該当チャンクについて携帯電話回線からのダウンロードを順次試みる. このリカバリ機構によってピアの帯域が低下した状況でも、他のピアの分担チャンクのダウンロードを阻害することなくチャンクの損失を防ぐことができる.

以下、2 章では関連研究とそれらの問題点を説明し、3 章

では提案手法である AERA について説明し、4 章でシミュレーションによって受信率、回線負荷の評価を行う. 最後に 5 章で結論を述べる.

2. 関連研究

2.1 モバイル・ライブストリーミング

近年のスマートフォンの普及に伴い、モバイル端末向けのライブストリーミング・システムの需要が高まっている [4]. モバイル環境において各端末は自身の携帯電話回線を用いてインターネットへアクセスし、映像コンテンツを視聴する. しかし携帯電話回線の帯域は電波状況やユーザのモビリティにより変動が激しいため、高画質なビデオコンテンツを連続的に視聴することは困難である. そのため、Client-Server 方式を用いている多くのビデオ・プロバイダは [5] のように、モバイル端末向けにビデオを低画質にコーディングして配信している. だが近年ではモバイル端末の高機能化、高性能化によって画面が大きだけでなく、またピクセル密度の高いディスプレイを搭載する端末が一般的となってきた. そのためユーザの要求はより高画質なビデオの視聴へとシフトしており、それに伴いビデオ配信サービス側でも、非常に高画質なビデオを提供するものも多くなってきている. しかし携帯電話回線の品質がこれらユーザの要求と配信されるビデオ品質に追いついていないのが現状である. モバイル端末の特性を考慮した P2P ライブストリーミング・システムの研究も盛んに行われている [6] [7] [8]. しかし先にも述べた通り、モバイル環境においては電波状況によって端末の帯域が大きく変動する [1][2]. そのため頻繁な P2P ネットワークからの離脱及び復帰が発生してしまい、隣人関係の維持及び隣人ピアへの確実なチャンク転送を行なうことができないという問題が存在し、モバイル端末に適した P2P ライブストリーミング方式の考案は未だに困難な課題である [9][10].

2.2 Cellular-WiFi Hybrid Network とモバイル・ライブストリーミング

モバイル端末は、複数の無線通信インタフェースを並列的に使用することが可能である. この特性を利用し、携帯電話回線でのデータ取得と MANET を用いたモバイル端末間のデータ交換を同時に行う Cellular-WiFi Hybrid Network とよばれるネットワーク構成を用いた配信方式が注目されている. 同一の動画コンテンツを視聴しているユーザが近距離にいる可能性が高いと考えられるコンテンツでは Cellular-WiFi Hybrid Network を用いた配信方式は最も効果的であると言える. Cellular-WiFi Hybrid Network を用いた動画コンテンツ配信に関する研究がいくつかある [11], [3], [12].

3G-MOVi[11] では、ピア間で協調を行うために必要な情報は全て配信者が管理している. そのためピアは周期的に

通信を行って隣人情報の交換を行い、よりリアルタイムな通信状況を常に配信者へ通知し続けなければならない。システムを維持するための情報を携帯電話回線を経由して扱うため遅延は大きく、ライブストリーミングのような時間的制約が厳しいサービスにおいては隣人間の情報をどれだけ正確に取得できたとしても、コンテンツ配信者とピア間での遅延のため実際のピア間通信状況との同期が取れなくなってしまうことが考えられる。

また既存研究の多くでは協調するピア間で別々のチャンクを携帯電話回線で取得する際、BitTorrent [13] を参考にしたビデオチャンクの確率的な選択が用いられている [3]。しかしライブストリーミングでは、ある時刻に交換可能なチャンク数は少ないためそれぞれのピアが確率的にチャンクを選択して携帯電話回線で取得する場合、協調するピア間で取得するチャンクに重複が多く発生してしまう。

MicroCast[12] では 1 ピアをスケジューラの管理者にして、各ピアが取得するチャンクを一定数ずつ割り当てている。管理者を設けた中央集権的なチャンクスケジューリングは参加ピア間での同期が取りやすく、エラーに強い一方で、スケジューラ管理者と他のピア間でのオーバーヘッドと遅延が予測できないという点や、管理者となっているピアが離脱した際にはチャンクスケジューリングそのものが機能しなくなり、ピアのモビリティの大きな環境でのビデオの受信品質が大きく低下するという問題がある。

3. 提案

本節では、提案手法である自律的なチャンク損失回避スケジューリング手法、Autonomous Error Recovery Algorithm for Mobile Live Streaming(AERA) の概要及び動作を説明する。

3.1 動作概要

AERA では隣人との定期的な通信によって、それぞれのピアに携帯電話回線を用いてダウンロードするチャンクを公平に分担する。分担の段階では MANET 参加者によって公平にダウンロードされることになるため重複は発生し得ない。

しかしダウンロードするチャンクに冗長性がないため、ピアの帯域が悪化すれば分担分のチャンクを自身で取得できない場合が発生し、チャンクが隣人から拡散されないという場面が発生し得る。チャンクの再生が迫っているにも関わらずチャンクが隣人から拡散されないという状況となったピアは、メッセージ交換の際の情報を基に得られるピアのチャンクに対する重み付けに従って損失しそうなチャンクの取得を交互に試みる。これによって、MANET の参加ピア数分だけチャンク取得に対するリトライが発生し、かつ、時間的にリトライのタイミングをずらすことでチャンクの損失及び重複ダウンロードを回避する。

3.2 取得チャンクに分担

提案手法では隣人との定期的なメッセージ交換によって自身が取得するチャンクを一意的に決定する。

表 1 に今後使用する記号一覧を示す。

表 1 記号一覧

表記	意味
R	ビデオレート
T	ビデオ時間
N	MANET 参加ピア数
C	チャンクサイズ
Δt_{msg}	乱数交換間隔
c	チャンク番号
D	再生遅延時間
r	リカバリ開始のタイミグ ($0 \leq r \leq 1$)
Δt_{rec}	リカバリ試行間隔

MANET に参加するピアは定められた間隔 Δt_{msg} 毎に、チャンク番号に対して乱数値を付与する。この定期的な発生する乱数交換イベントによって隣人関係の状態を更新すると共に、各チャンクに対するピアの優先順位付けを行う。 Δt_{msg} によって区切られるチャンクの個数は必ず正数値であり、 Δt_{msg} 及び配信ビデオパラメータによって、チャンクセット 1 つあたりに含まれるチャンクの数式 1 で表される。

$$\left\lceil \frac{R}{C} \Delta t_{msg} \right\rceil \quad (1)$$

またビデオストリームの配信が始まってから終了するまでに乱数値が交換される回数は $T/\Delta t_{msg}$ である。

チャンクセットに対して乱数を付与し、隣人に対してその乱数列をブロードキャストし、隣人の乱数列が手元に集まったら、これらの乱数列の大きさを比較し自身が携帯電話回線で取得するチャンクを決定する。具体的なアルゴリズムを Algorithm 1 に示す。

Algorithm 1 ダウンロードチャンク決定手順

```

1: if collected random numbers lists from all peers then
2:   MyDlChunks = ∅
3:   for  $c_j \in$  対象のチャンクセット  $S$  do
4:     if  $c_j == \max(S)$  then
5:       MyDlChunks.add( $c_j$ )
6:     end if
7:   end for
8: end if

```

MANET に参加しているすべてのピアは、自分の乱数値リストも含めてすべてのピアの乱数値リストをローカルに保有するしているため、各ピアが Algorithm 1 を実行し、各々の取得するチャンクを決定する。

Algorithm 1 では、乱数値を比較し、該当のチャンクの取

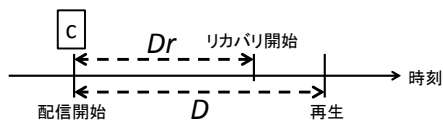


図 1 再生遅延時間 D とリカバリ開始位置

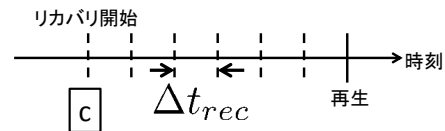


図 2 リカバリの試行タイミング

得を行うピアを決定している。具体的には、ピア i のチャンク j に対する乱数値 s_{ij} を、その他すべてのピア $i(i \in N)$ と比較し、最も大きい乱数値 s_{ij} を付与しているピアがチャンク j を自身の携帯電話回線でダウンロードする。

3.3 リカバリ

チャンクがメディアサーバに生成されてから、ユーザがそのチャンクを再生するまでには時間差がある。この再生遅延時間をここでは D としており、チャンクの生成から D 時間が経過するまでにそのチャンクを取得していないと、そのチャンクに該当する部分のビデオが抜け落ちてしまう。

図 1 はパラメータ r によってチャンク c のリカバリ開始のタイミングを決定している図である。リカバリプロセスが発生するタイミングは、再生遅延時間 D のうち、 Dr だけ経過した時点から開始する。チャンクの配信開始初期は、分担しているピアが自身の携帯電話回線でチャンクのダウンロードを試みており、ある時間 Dr 経過後からは、ダウンロードを試みているピアの帯域状態が良くないという仮定のもと、他のピアがリカバリプロセスが開始する。

リカバリを行うピアとそのタイミングは、チャンクによって異なる。リカバリが必要なチャンクを c' とする。このチャンク c' のリカバリは、現在チャンク c' のダウンロードを試行しているであろうピア以外が、チャンク c' のダウンロードを Δt_{rec} 間隔で順番に試みる。

まず Δt_{rec} を決定する。この間隔は、

$$\Delta t_{rec} = \frac{R/N}{C} \quad (2)$$

で決定する。

これはピアの携帯電話回線帯域が視聴中のビデオの分担チャンクを不足なくダウンロードできるだけの帯域があるときの 1 チャンクのダウンロード所要時間である。あるピアがリカバリを実行し、その実行開始から Δt_{rec} だけ待ってもチャンクが拡散されない場合は、リカバリを実行したピアには十分な帯域がなく、リカバリを実行できなかったとみなす。実際は自分がリカバリを行うタイミングで帯域が空いている場合のみチャンクのダウンロードを行い、チャンクのダウンロードを行っている最中ならばリカバリを実行しない。

続いて、リカバリを実行するピアの決定法を示す。リカバリが必要なチャンクのリカバリは Δt_{rec} 間隔で実行され、この間隔おきに別々のピアがリカバリを実行する。 $N = 4$

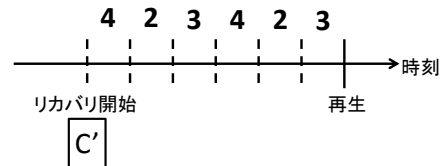


図 3 リカバリの試行例

で、ピア 1 がチャンク c' のリカバリ開始時刻までにチャンク c' の拡散を行えなかった際には、他のピアが順にチャンク c' の取得を試みる。このリカバリの繰り返しはチャンクの取得に成功し、他ピアへチャンクが届くか、または取得に失敗し続けチャンクの再生時刻になるまで繰り返される。

リカバリの試行順はリカバリが必要なチャンク c' へ付与された乱数値の大きい順に行われる。チャンク c' に対する乱数値が表 2 の状態ならば、ピア 1 を除いて、図 3 のようにリカバリが試行される。

表 2 チャンク c' に対する各ピアの乱数値

peer i	1	2	3	4
$r_{ic'}$	88	12	5	33
リカバリ実行順序	×	2	3	1

4. シミュレーション評価

提案手法を評価するために、シミュレーションを行った。以下その評価結果を述べる。

4.1 シミュレーション条件

表 3 シミュレーション条件

パラメータ	値
同時視聴ピア数	1 - 6
対象動画数	1
チャンクサイズ	100[kbit]
再生遅延	3.0[sec]
ビデオレート	1.0[Mbps]
動画の長さ	1800[sec]
チャンク数	18000
乱数値リスト交換間隔	3.0[sec]
携帯電話回線平均下り帯域	0.4[Mbps]
リカバリ試行間隔 Δt_{rec}	0.03 [sec]
1 シナリオ当たりのシミュレーション回数	10 回

表 3 にシミュレーションで使用した条件を示す。またす

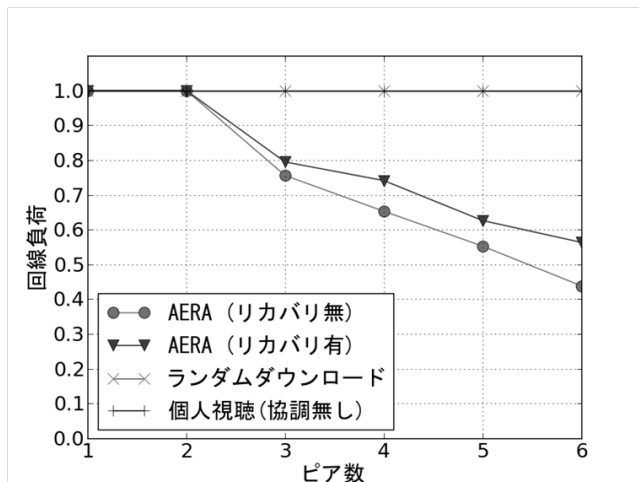


図 4 ピア数変化に対する携帯電話回線負荷の変化

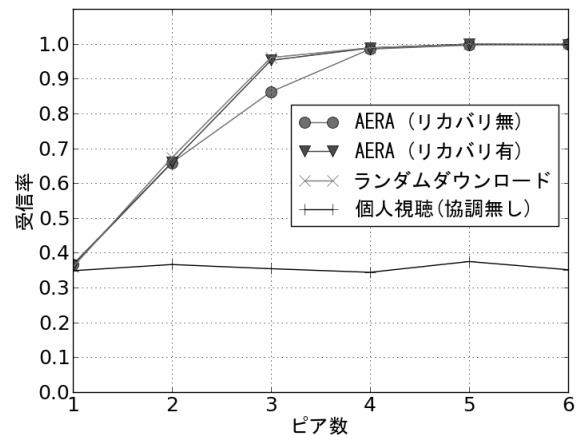


図 5 ピア数変化に対するビデオ受信率の変化

すべてのピアは互いに直接通信可能な距離に存在しているものとする。ピア間で構築している MANET で使用する WiFi の帯域は、ピアがビデオチャンクを隣人に送信するのに使用する帯域に比べて十分な帯域が使用できるものとする。また携帯電話回線におけるパケットロスが発生せず、十分な時間があれば必ずチャンクを取得できるものとする。すなわち、ピアは自身の携帯電話回線で要求したチャンクは 100% 受信できる。シミュレーションの流れは、まず全てのピアはシミュレーション開始時点で一斉に隣人関係を構築する。続いて実験開始後すぐにメディアサーバはビデオの配信を開始し、メディアサーバの最後のチャンクの配信期限が切れた時点でシミュレーションを終了する。

以上の条件でシミュレーションを 10 回繰り返す、結果としてその算術平均を用いた。

4.2 比較対象

提案手法の比較対象の 1 つであるランダムダウンロードでは個々のピアが帯域の許す限りアクティブチャンクを無作為に選択し、携帯電話回線で取得し、取得が完了後、アドホック通信を用いて隣人ピアへ取得したチャンクをすべて拡散している。また、他のピアから受信したチャンクを重複して自身でダウンロードすることはない。

アクティブチャンクを選択がそれぞれのピアで完全にランダムなので、ピア数が少なければチャンクの重複ダウンロードは少なく、同時にピア間で事前にダウンロードするチャンクの取り決めを行っていないため帯域がスケジューリングによって制限されることもない。よってこのランダムダウンロードで得られるビデオ受信品質が本シミュレーションでの携帯電話回線帯域下での品質の上限であると言える。

4.3 ピア数の変化に対する受信ビデオ品質並びに回線帯域負荷

図 4,5 にそれぞれピア数変化に対する携帯電話回線負荷とビデオの受信率を表したものを示す。ピア数が 3 の場合は、ピアの総帯域がビデオレートとほぼ同じというパラメータ設定になっている。このとき提案手法ではランダムダウンロードの受信率と同等の 95% になっていることが確認できる。これは携帯電話回線の帯域変動によるチャンク損失が発生しそうな場面で帯域に余裕があるピアのリカバリが各ピアで適切に行われているためである。また回線負荷は 80% と抑制されていることがわかる。

提案手法のリカバリを行わない場合の受信率は約 85% であり、その一方で回線負荷は 80% 弱とまだ帯域に余裕がある。これは事前にダウンロードするチャンクを取り決めているため、ピアの帯域低下によるチャンク損失が発生しやすい状況にもかかわらず、チャンクの損失に対処できていないためである。

協調ピアの総帯域がビデオレートに対して大きく不足する場合 (ピア数 1-2 のとき) はピアの総帯域がビデオレートに対して大きく不足しているためいずれのスケジューリング手法を用いても、帯域負荷は 100% で、また受信率に差は出ない。

一方で、協調ピアの総帯域がビデオレートに対して十分足りている場合 (ピア数 4-6 のとき) はピアの総帯域がビデオレートに対して十分大きいいためいずれのスケジューリング手法を用いても、1 ピア当たりが取得する必要のあるチャンク数が少なく、チャンクを損失する可能性が極めて小さいので 100% の受信率が達成できる。

4.4 チャンクの重複率

本稿でのチャンクの重複とは、あるピアがチャンクの取得を実行している際に、他のピアが同一のチャンクの取得を開始してしまう状況を指す。よって、チャンク重複

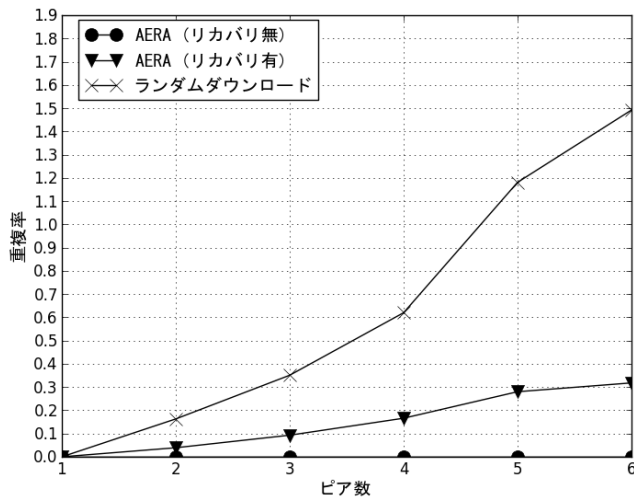


図 6 重複チャンクダウンロード量

率 100% とは、MANET 上にビデオストリームの倍のトラフィックが発生している状態を意味する。

図 6 に協調するピア数を変化させた場合の受信チャンク重複率を示す。

図よりピア数の変化に対してピア当たりの重複率が 5% 程度に収まっていることが確認できる。冗長性によるチャンク損失の可能性の低下を考えれば許容できる重複率であるといえる。

提案手法において重複率が 0% でないのは、リカバリを実行しているピアの帯域が小さく、次のピアがリカバリを実行しそのどちらもがチャンクの取得に成功してしまう場合があるからであると考えられる。また、各ピアの帯域の総和がビデオレートを上回るシナリオで、重複率が増加するのは上記と同様で、帯域の小さい状態のピアがリカバリを実行できる確率がピア数の増加と共に大きくなるからであると考えられる。

4.5 ピアの離脱による影響

ピア数が増えるシナリオでシミュレーションを行う。

ピア数は開始段階で 4 ピア、終了段階までに 2 ピアへと変化させる。これは、ピアの携帯電話回線帯域とビデオレートに対して、帯域が足りている状態から、不足する状態までの変化を観察するためである。シナリオの詳細について次に記す。

ピア離脱シミュレーションのシナリオ

- 0-100 秒 4 ピアが協調
- 100 秒 1 ピアが離脱
- 100-200 秒 3 ピアが協調
- 200 秒 1 ピアが離脱
- 200 秒以降 2 ピアが協調

図 7 に協調ピア数が増えた際のピアの帯域負荷状況を示す。この図より、区間ごとの帯域負荷とチャンクの受信

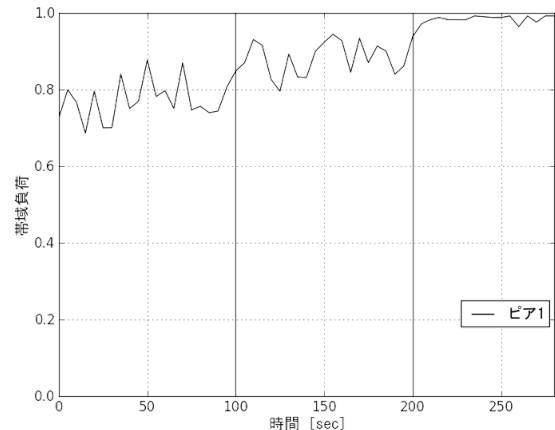


図 7 オーバーレイの変化と帯域負荷

率を算出したのが表 4 である。時間が進みピア数が減少するに従って帯域負荷が上昇しており、リカバリの回数が増加していることがわかる。

以上よりリカバリが機能していることと、ピアが状態の変化に適宜対応していることが確認できた。

表 4 シナリオ区間ごとの帯域負荷とチャンクの受信率

区間 [sec]	ピア数	受信率	帯域負荷
0 - 100	4	0.98	0.72
100 - 200	3	0.91	0.85
200 - 300	2	0.70	0.98

5. おわりに

ここ数年の高性能なスマートフォンの普及に伴い、モバイル端末向けのライブストリーミング・サービスの需要が増加している。しかしモバイル端末が自身の携帯電話回線を用いてコンテンツの受信を行う場合、携帯電話回線の帯域変動が激しいために高画質な動画コンテンツを連続的に視聴することは難しい。

携帯電話回線及び MANET でのアドホック通信の 2 つを並立して利用するネットワークに Cellular-WiFi Hybrid Network がある。本稿ではこの Cellular-WiFi Hybrid Network をライブストリーミングへ適用することを考えた。Cellular-WiFi Hybrid Network を用いた端末間協調ライブストリーミングにおいては、各ピアが自身の携帯電話回線でチャンクを取得し、MANET で隣人関係にあるピアへ取得したチャンクをブロードキャストする。この際、最も高品質なビデオを視聴するためにランダムダウンロードを行えば取得するチャンクに重複が頻発するために余剰なトラフィックが発生してしまう。一方で、誰がどのチャンクをダウンロードするということを事前にスケジュールして各ピアに割り当ててしまうと、割り当てられたピアの回線状況が悪化した際やピアが離脱した場合に再スケジュールのためのオーバーヘッドが発生し、リアルタイム性を損な

うという問題があった。

そこで本論文では、すべてのピアに事前に取得するチャンクを公平に分担しながらも、ピアの帯域不足の際には各ピアが適宜判断して損失しそうなチャンクを取得するチャンクスケジューリング手法である AERA の提案をした。本提案手法では、MANET 上のピアは自身の携帯電話回線で取得するチャンクについて事前に分担を行うため、それぞれのピアはそれぞれのチャンクに対して乱数を生成し、最も大きな数値を提示したピアが該当のチャンクを携帯電話回線から取得している。これによって MANET に参加するピアは携帯電話回線で取得するチャンクを公平に分担する。

また、携帯電話回線の帯域が低下した際や、ピアが不意に離脱した際には先ほどチャンクの事前分担に使用した乱数を用いてチャンク損失の回避(リカバリ)を行う。このリカバリでは先ほど交換した乱数を比較し、2 番目以降の大きな乱数を持つピアが順に損失しそうな該当のチャンクについて携帯電話回線からのダウンロードを試みている。

最後に本提案手法を、シミュレーションによって評価した。その結果、本提案手法によって 1 ピア当たりの冗長なチャンクのダウンロードをピア数の変化に関係なく約 5% に抑制することに成功した。また携帯電話回線の帯域変化に関わらずピア数に対する最大ビデオ受信品質と同程度のチャンク受信率を達成すると同時に、携帯電話回線のトラフィック削減という目的に対し、全ピアの受信品質を維持しながら携帯電話回線使用率の削減を実現できたことから、本提案手法の有用性を示した。

謝辞 本研究の一部は、JSPS 科研費 (B) 課題番号 25280032 (2013 年) の助成により行われた。

参考文献

- [1] Kim, S.: Cellular network bandwidth management scheme by using nash bargaining solution, *Communications, IET*, Vol. 5, pp. pp.371-380 (2011).
- [2] Yami, E. and Vakili, V.: Adaptive Bandwidth Allocation and Mobility Prediction in Mobile Cellular Networks, *Next Generation Mobile Applications, Services and Technologies, 2009. NGMAST '09. Third International Conference* (2009).
- [3] Hanano, H., Murata, Y., Shibata, N., Yasumoto, K. and Ito, M.: Video ads dissemination through WiFi-cellular hybrid networks, *IEEE Pervasive Computing and Communications* (2009).
- [4] Ickin, S., Fiedler, M. and Wac, K.: Live video streaming for mobile devices: An application on android platform, *Sustainable Internet and ICT for Sustainability (SustainIT)* (2012).
- [5] YouTube, M.: <http://m.youtube.com>.
- [6] Tan, E., Guo, L., Chen, S. and Zhang, X.: SCAP: Smart caching in wireless access points to improve P2P streaming, *27th IEEE Int. Conf. Distrib. Comput. Syst.* (2007).
- [7] Shioh-yang, W., Jungchu, H. and Chieh-Ming, C.: Head-light Prefetching and Dynamic Chaining for Cooperative Media Streaming in Mobile Environments, *Mobile Computing, IEEE Transactions*, Vol. 8, pp. pp.173-187 (2009).
- [8] Shioh-yang, W. and Cheng-en, H.: QoS-Aware Dynamic Adaptation for Cooperative Media Streaming in Mobile Environments, *IEEE Trans. Parallel and Distributed Systems*, Vol. 22, pp. pp.439-450 (2011).
- [9] Zhijie, S., Jun, L., Zimmermann, R. and Vasilakos, A.: Peer-to-Peer Media Streaming: Insights and New Developments, *Proceedings of the IEEE*, Vol. 99, pp. pp.2089-2109 (2011).
- [10] Mantoro, T., Ayu, M. and Jatikusumo, D.: Live video streaming for mobile devices: An application on android platform, *Uncertainty Reasoning and Knowledge Engineering (URKE), 2012 2nd International Conference* (2012).
- [11] Iyer, T., Hsieh, R., Rizvandi, N. B., Varghese, B. and Boreli, R.: Mobile P2P Trusted On-Demand Video Streaming, *CoRR*, Vol. abs/1204.0094 (2012).
- [12] Keller, L., Le, A., Cici, B., Seferoglu, H., Fragouli, C. and Markopoulou, A.: MicroCast: Cooperative Video Streaming on Smartphones, *MobiSys '12 Proceedings of the 10th international conference on Mobile systems, applications, and services* (2012).
- [13] BitTorrent: <http://www.bittorrent.com/> (2013).