ライブ配信における
複数地点同期再生方式の提案と評価

筑波大学
図書館情報メディア研究科
2017 年 3 月
土屋　俊貴

目次

第1章　序論

IPTV や Video On Demand、ライブ配信などインターネットを利用した動画配信サービスが普及し、様々な動画を視聴できる環境が整ってきている。Cisco による年次予測では、インターネットトラヒックに占める動画の割合は 2020 年には 82%に達すると予測されている[1]。動画トラヒックの増加は、上で述べた動画配信サービスの普及による動画視聴者数や視聴回数の増加や、動画自体の高解像度化といった品質向上が要因として挙げられる。

また動画配信に利用される技術も変化してきており、従来は Real-time Transport Protocol と Real-time Transport Control Protocol を利用したストリーミングが主に使われていたが、最近では HTTP を利用した Adaptive Bitrate streaming (ABR streaming)が多く使われるようになってきている。ABR streaming とは再生する動画を予め複数の品質にエンコードしておき、視聴ユーザの再生デバイスや通信状況に応じて配信する品質を動的に変化させることで、ユーザの視聴品質を向上する技術である。また動画は数秒単位のセグメントと呼ばれる動画ファイルに分割され、各セグメントを Contents Delivery Network (CDN)のキャッシュサーバへコピーし、ユーザは最寄りのキャッシュサーバから動画を受信することによって配信サーバの負荷分散や動画再生の遅延を削減することができる。HTTP を利用した ABR streaming の規格としては、Apple が開発した HTTP Live Streaming (HLS)と国際標準規格である MPEG-DASH (Dynamic Adaptive Streaming over HTTP)が主に利用されている。

インターネットを利用した動画視聴の普及や動画配信技術の進歩に加え、Social Networking Service (SNS)の普及により、離れた場所にいる友達と同じ動画を見ながら SNS 等でコミュニケーションを取ることで同じ体験をしたように感じる体験共有型の利用形態やサービスが注目されている。しかし、インターネットによる配信では、通常の放送とは異なり各ユーザが同時にデータを受信できる保証はない。そのため、同じ動画を視聴していても、再生している位置がずれていくことがある。動画を一緒に見ているグループのうち、あるユーザの再生している位置がずれている状態である場面について発言した場合、見ている場面によっては話が合わなくなるといったことが生じてしまう。つまり、こうした体験共有型のサービスでは、同一の動画を視聴するユーザ同士で動画の再生位置が同期している必要がある。このような離れた場所にいるユーザ間の再生位置を同期させることを Inter-Destination Media Synchronization (IDMS)と呼ぶ[2]。

IDMS を実現させるための取り組みも多く行われている。文献[3]では標準化組織である IETF において、RTCP を用いた IDMS の実現方法の標準化を提唱している。これはユーザの視聴位置等の情報を同期サーバに伝える RTCP XR IDMS Report Block と同期サーバからユーザへ、同期の基準となるタイムスタンプを通知する IDMS Settings Packet の規格を示し、これらを利用したユーザ間の同期手順を提案するものである。この他にも RTP/RTCP を対象とした研究は多くある[4][5][6]。しかし現在の動画配信サービスでは HTTP を利用した ABR streaming が主流であり、RTP/RTCP の特徴や制御方法を利用した IDMS を活用することは難しい。文献[7]ではネットワークが混雑した状況における同期方式を示しており、サーバから同期制御パケットを送信し、その指示に従ってクライアントが再生の停止や再生位置のシークを行うことによって同期を取るものである。また文献[8]では MPEG-DASH を利用した環境における IDMS の実現方法を提案しており、MPEG-DASH の Manifest File を同期のセッションマネージャとして利用する方式の提案やユーザ同士で P2P ネットワークを構築し、相互に情報交換することによって集中管理ノードを必要としない同期方式を提案して

いる。文献[9]ではスマートフォンを利用している状況で、近くにいる他のスマートフォン利用者と同じ動画を視聴したい場合において、あるユーザが動画の一部分を配信サーバから受信し、それを近隣のユーザへ渡すという方式を取ることによってネットワーク負荷を考慮し、かつスムーズな動画再生を実現している。

またユーザの再生同期に関して、再生位置のずれによる影響を調査した研究もある。文献[10]ではクライアントが同時刻に画像を見ながら対話を行う、体験共有型アプリケーションを利用する際、どの程度の遅延であれば体験を共有していると感じるかを調査したものであり、共有手段として電話を利用した場合には、許容できる遅延は 2 秒以内であることを示している。また文献[11]も音声通話をしながら動画を見ている場合において、お互いの見ている映像に何秒程度ずれがあれば気付くのか、また何秒程度のずれでユーザが苛立つのかを調査したものであり、結果としては 2 秒ずれがあると 8 割程度にユーザが気付き、また 4 秒ずれがでると 9 割近いユーザがそのずれに苛立ちを示すことを明らかにしている。また ETSI による音声コミュニケーションに付随するサービスに関する規格[12]においては、SNS の遅延の許容時間は数秒程度とされている。本研究では SNS やチャット等によるテキストメッセージを利用してコミュニケーションを取っている環境を想定する。そのためこれらの調査結果をもとにずれの基準値を 4 秒として設定する。

本論文では、HTTP を利用した ABR streaming を利用したライブ配信における、複数地点同期再生方式の提案及び評価を行う。同期方式の検討にあたり、まずユーザ間の再生位置のずれや発生原因を明らかにすることを目的として行った計測実験[13][14]について解説する。計測実験では既存のライブ配信サービスである YouTube Live[15]を利用し、複数地点での同一のライブ配信を再生している状況における情報を収集した。そして計測実験での結果をもとにユーザ間の再生同期方式を提案し、ライブ配信を模擬したシミュレーションによって提案方式の評価を行う。

本論文の構成は以下の通りである。第 2 章では、現在のライブ配信環境の概要を示す。そして第 3 章では、計測実験の方法及び使用した計測システムの解説をし、その後計測実験の結果を示す。第 4 章では再生同期を行う提案方式を示し、第 5 章でシミュレーションによる評価実験の方法及び結果を示す。そして第 6 章で結論を述べる。

第2章　ライブ配信環境の概要

2.1　Adaptive Bitrate streaming (ABR streaming)

インターネットを利用して動画を視聴している場合、ユーザの利用可能帯域は時々刻々と変化しているため、ある時点で最適な品質であったとしても違う時点では不適切な品質となってしまうことがある。例えば動画を高画質で視聴していた場合に、利用可能帯域が減少したことで動画再生が停止してしまうということや、逆に利用可能帯域が少ない時点に視聴を開始したことで、十分な帯域が利用できるようになっても低画質のまま視聴をし続けてしまうといったことが起こりうる。ABR streaming ではこのようなユーザの視聴環境の変化に動的に対応することで、視聴品質の悪化を防ぐ技術である。

ABR streaming では、まずオリジナルの動画を予め複数のビットレートに変換した動画ファイルを生成し、ビットレートの情報を Manifest File に記録しておく。動画の再生クライアントでははじめに Manifest File をダウンロードし、その中から適切なビットレートを選択して動画をリクエストし、動画再生を開始する。その後クライアントは動画リクエストの Round Trip Time (RTT)や受信データのバッファ容量等を見ることで、ユーザの視聴環境の変化を監視し、適切なビットレートの選択を繰り返していく。

HTTP を利用した ABR streaming の規格である HLS や MPEG-DASH では、複数のビットレートに変換された動画ファイルを、さらにセグメントと呼ばれる数秒単位の動画ファイルに分割する。HLS や MPEG-DASH での Manifest File はビットレートの種類や各セグメントにアクセスするための URL、1 セグメントあたりの再生時間等が記録されている。再生クライントは Manifest File を参照して 1 セグメントごとに HTTP リクエストを送信し、動画ファイルを繰り返しダウンロードしていくような形で動画再生を行っていく。また HLS や MPEG-DASH は Web ブラウザの標準としてサポートされているため、専用のアプリケーションやプラグイン等を利用せず、HTML 5 の video 要素を利用して動画を再生することが可能である。

2.1.1　ライブ配信における ABR streaming

VOD のように既に動画コンテンツが存在している場合であれば、動画の変換やセグメント分割は一度行えばよい。したがって Manifest File にも変更はなく、再生クライアントは最初にダウンロードした Manifest File を参照して動画の最後まで再生し続けることが可能である。しかしライブ配信の場合、常に新しいセグメントが生成されていき、また古いセグメントは必要無くなっていくため、Manifest File も新しいセグメントの情報を追加し、古いセグメントの情報を削除していく。したがって再生クライアントも最初に Manifest File をダウンロードするだけではなく、その後も定期的に Manifest File をダウンロードし、新しいセグメントが追加されているかを確認する必要がある。

2.2　Contents Delivery Network (CDN)

CDN は地理的に分散するようにキャッシュサーバを複数配置し、オリジナルサーバのコンテンツをキャッシュしておくことで、そのコンテンツへのリクエストをキャッシュサーバに分散させ、オリジナルサーバへの負荷集中することができ、またユーザはオリジナルサーバよりもネットワーク的に近いキャッシュサーバからコンテンツが受信できるようになるため、

コンテンツの取得時間も短くなる。

CDN を利用したサイトへのアクセス手順は以下のように行われる。

1. ユーザがオリジナルサーバへリクエストを送信する
2. リクエストを受けたオリジナルサーバはユーザに近いキャッシュサーバへリクエストをリダイレクトする
3. リクエストを受け取ったキャッシュサーバがユーザへレスポンスを返す
4. 以降、ユーザはレスポンスが返ってきたキャッシュサーバへリクエストを送信する

ユーザをキャッシュサーバへ割り振る方法としては、上で述べたリダイレクトによる方法の他に、DNS での名前解決で直接キャッシュサーバの IP アドレスを返答することによって行う方式もある。

またキャッシュサーバがオリジナルサーバからコンテンツをキャッシュする方法としてプッシュ型とプル型がある。プッシュ型はあらかじめオリジナルサーバからキャッシュサーバへコンテンツを送信することでキャッシュする方式である。またプル型はキャッシュサーバがキャッシュしていないコンテンツへのリクエストを受信したとき、オリジナルサーバからコンテンツをダウンロードし、キャッシュする方式である。

動画配信において、HLS や MPEG-DASH を利用している場合、動画ファイルはセグメントごとに分割されており、ユーザは各セグメントに対して HTTP リクエストを送信すれば良い。そのため、セグメントをキャッシュサーバにキャッシュしておくことによって、ユーザはキャッシュサーバから動画の配信を受けることができるようになる。ライブ配信においても同様に新しく生成されたセグメント及び Manifest File をキャッシュサーバへプッシュすることでユーザはキャッシュサーバから動画配信を受けることができる。ただしライブ配信の場合、ある程度古くなったセグメントは必要なくなるため、キャッシュから削除する必要がある。

第3章　　計測実験

本章ではライブ配信サービスを利用しているユーザの再生状況を調査するために行った計測実験について述べる。計測実験は 2 回行い、実験 1 ではユーザ間の再生位置にずれが生じるかを調査することを目的とした。そして実験 2 では、より詳細な再生状況の取得を行い、ユーザ間のずれの発生原因を分析することを目的とした。

以下では、まず計測実験の実験方法を示し、次に計測に使用した計測システムの構成を解説する。その後実験 1、実験 2 の条件及び結果を述べる。

3.1　　実験方法

実験環境の概要を図 1 に示す。計測は動画配信サービスである YouTube Live を利用した。YouTube Live は YouTube が運営しているライブ配信サービスであり、MPEG-DASH を用いて動画配信をしている。そして通信環境や地理的位置がそれぞれ異なる複数ユーザが、再生クライアントを利用して同一のライブストリーミングを視聴した際の再生状況を計測する。また実験にあたってユーザから YouTube Live への通信経路やキャッシュサーバ等の利用状況は不明であるものとして計測を行った。

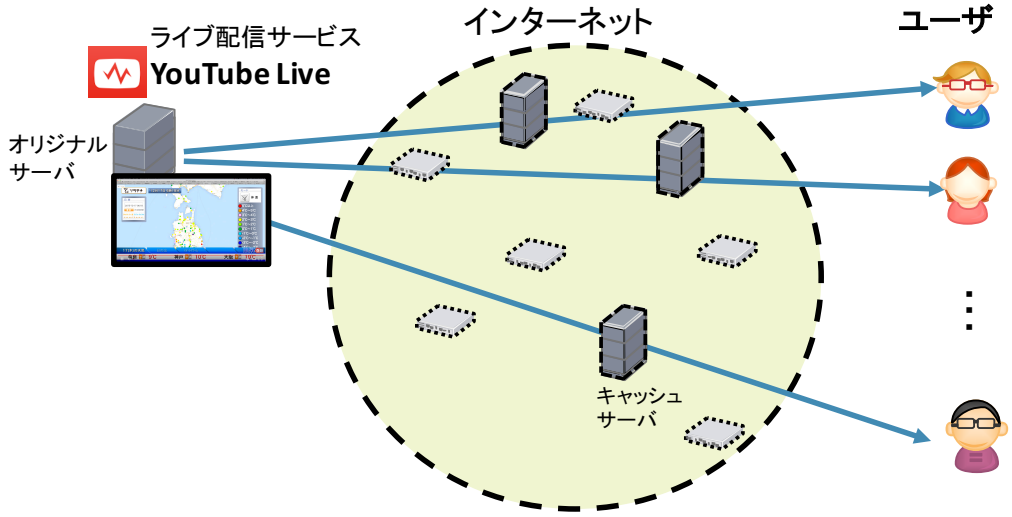

図 1. 計測実験環境

3.2　　計測システム

図 2 に計測実験に使用した計測システム及び、計測の流れを示す。ユーザは Web ブラウザを動画再生のクライアントとして利用している状況であり、計測システムはユーザブラウザにインストールする計測ツールと、計測ツールが取得した再生状況を収集・保存する収集サーバによって構成される。計測ツールは Google Chrome の拡張機能として実装したため、ユーザブラウザも Google Chrome を利用している必要がある。

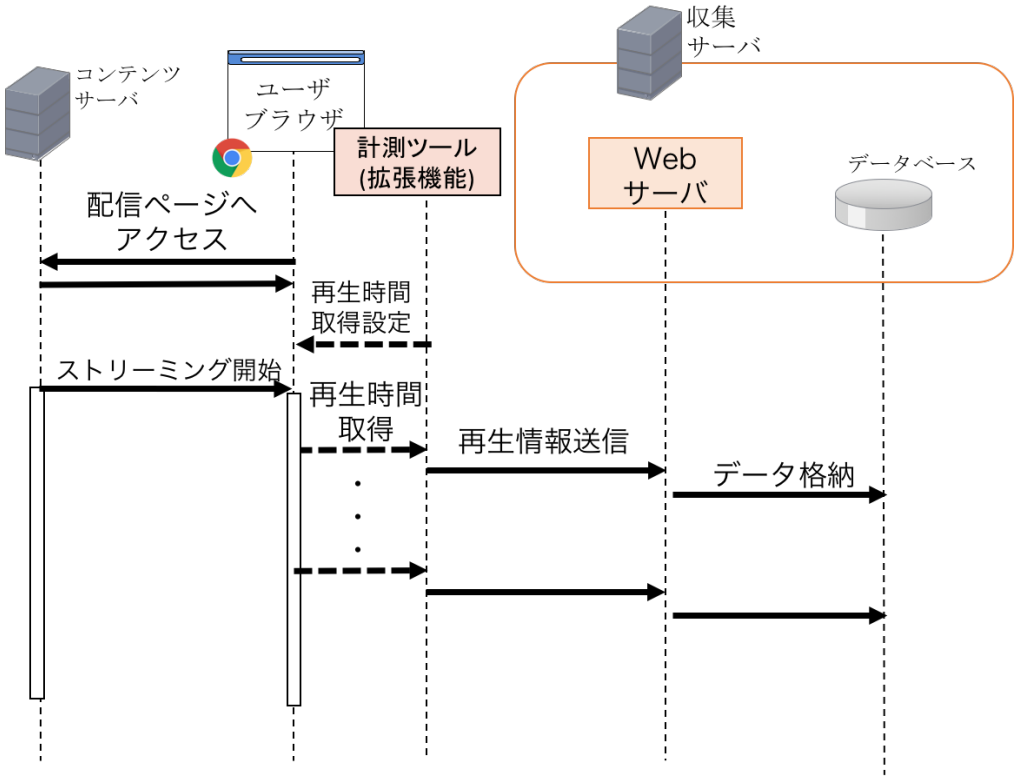

図 2 計測システムの構成と計測の流れ

3.2.1 計測ツール

計測実験では実際のライブ配信サービスとして YouTube Live を利用するため、動画配信サーバの情報やネットワーク情報は取得できない。また動画の再生ページに計測のための機能を追加することもできない。そのため配信側のページを変更することなくユーザの再生状況を取得する必要があり、今回は Google Chrome の拡張機能として計測ツールを実装した。

計測ツールは再生ページにアクセスしたときに有効になり、再生ページから video 要素を探索する。video 要素が見つかった場合、その video 要素に対して timeupdate イベントが発生したときに再生情報を取得する設定を行う(図 2: 再生時間取得設定)。timeupdate イベントは動画の再生位置が更新されたときに起動するイベントであり、Google Chrome の場合再生が順調に進んでいるときは 0.25 秒間隔で発生する。そのため再生情報の取得及び送信も 0.25 秒間隔で行われる。

再生情報として取得する情報を表 1 に示す。動画の再生位置やバッファリング範囲等の動画に関する情報は video 要素から取得し、リクエストの RTT や発着 IP アドレス等のネットワークに関する情報は Chrome 拡張機能の API を利用して取得した。また計測実験 1 では動画の再生位置と取得時刻のみを取得し、実験 2 でより詳細な情報を取得した。

表 1. 計測ツールで取得する再生情報

情報	取得元（情報名）	備考
動画の再生位置	video 要素（currentTime）	
再生情報の取得時刻	JavaScript (Date.now())	
バッファリング範囲(先頭・末尾)	video 要素（buffered start/end）	実験 2 のみ取得
動画解像度	video 要素(videoHeight/Width)	実験 2 のみ取得
セグメントのリクエスト RTT	API (chrome.devtools.network.onRequestFinished)	実験 2 のみ取得
リクエストの発着 IP アドレス	API (chrome.webRequest.onCompeleted)	実験 2 のみ取得

3.3 実験 1

3.3.1 実験条件

通信環境や地理的位置が異なる 5 種類のクライアントを利用して YouTube Live の同一のライブストリーミングを視聴した際の再生情報を 2 日間(2015/11/9 ~ 2015/11/11)計測した。クライアントの詳細を表 2 に示す。通信環境としては PC に有線接続をした場合と無線 LAN で接続をした場合、モバイルルータを利用して携帯回線で通信した場合の 3 種類とした。また計測はつくば二箇所(筑波大学と筆者の自宅に、それぞれ 1 クライアント)と、東京二箇所(3、4 は同一箇所)で行った。

また本実験ではクライアントは計測期間の間に自由に参加・離脱をするものとした。

表 2. 実験 1 での計測クライアント

No.	通信環境	計測場所
1	有線（学内ネットワーク）	筑波大学
2	有線（家庭用ネットワーク）	つくば
3	有線（家庭用ネットワーク）	東京
4	無線（家庭用ネットワーク）	東京
5	モバイルルータ	東京

3.3.2 実験結果

図 3 に 2 日間の再生位置のずれの変化を示す。ここでの再生位置のずれとは、同時に視聴しているユーザのうち最も進んでいる(最新に近い)ユーザの再生位置と最も遅れているユーザの再生位置の差を示したものである。破線が再生同期の基準値である 4 秒を示しており、2 日間で 4 秒以上のずれが発生した回数は 24 回であり、10 秒以上ずれがある状態で数時間再生し続けている状況も確認できる。また、図 3 において再生位置のずれがなくなる理由はユーザが離脱したことによって、ユーザ間のずれがなくなったことによるものである。次に

ユーザ間にずれが生じた時点での様子を図 4、図 5、図 6 に示す。これらの図はユーザ間のずれが発生した時点の前後 1 分半(計 3 分間)のユーザの再生位置の推移であり、1 本の実線が 1 人のユーザを示している。図 4 はユーザ 1 とユーザ 5 が同時に視聴している状況であり、最初に時点では、ほぼずれが無い状態である。しかし 29 分 20 秒あたりでユーザ 5 の再生が停止し、その状態が 1 分程度続いた後再生が再開された。そのため停止せずに動画が再生されていたユーザ 1 との間に約 1 分の差ができてしまうという状況が確認できた。また図 5 も再生が停止したことによって再生位置のずれが発生する状況であるが、どのユーザも再生の停止と再開を繰り返しており、頻繁に差の拡大・縮小が起きていることがわかる。図 6 ではユーザ 1 とユーザ 4 が再生している状況で、新たにユーザ 2 が視聴を開始するという状況である。このとき、ユーザ 1、ユーザ 4 ともに既に再生が停止したことによって最新よりも遅れており、最新のデータから再生を開始するユーザ 2 との間に再生位置のずれが発生するという状況が確認できた。

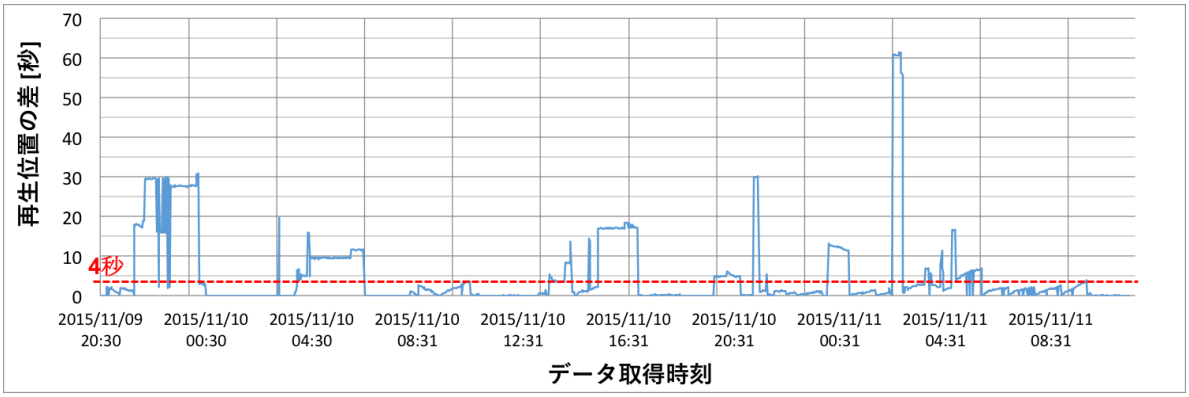

図 3. 計測実験 1 における再生位置のずれ

3.3.3 まとめ

計測実験 1 では、ライブ配信サービスである YouTube Live を利用し、複数地点でライブストリーミングを再生することによって、ユーザ間の再生位置にずれが生じることが確認できた。また、1 度ずれてしまった場合、長時間ずれがある状態で再生が続いていくことが確認できた。再生位置のずれが生じる原因としては、ユーザの再生が停止したことによって他の順調に再生していたユーザとの間にずれが生じる場合と、すでに遅れているユーザがいる状態で新たなユーザが参加することによって遅れているユーザと新規ユーザとの間にずれが生じる場合に 2 種類が確認できた。

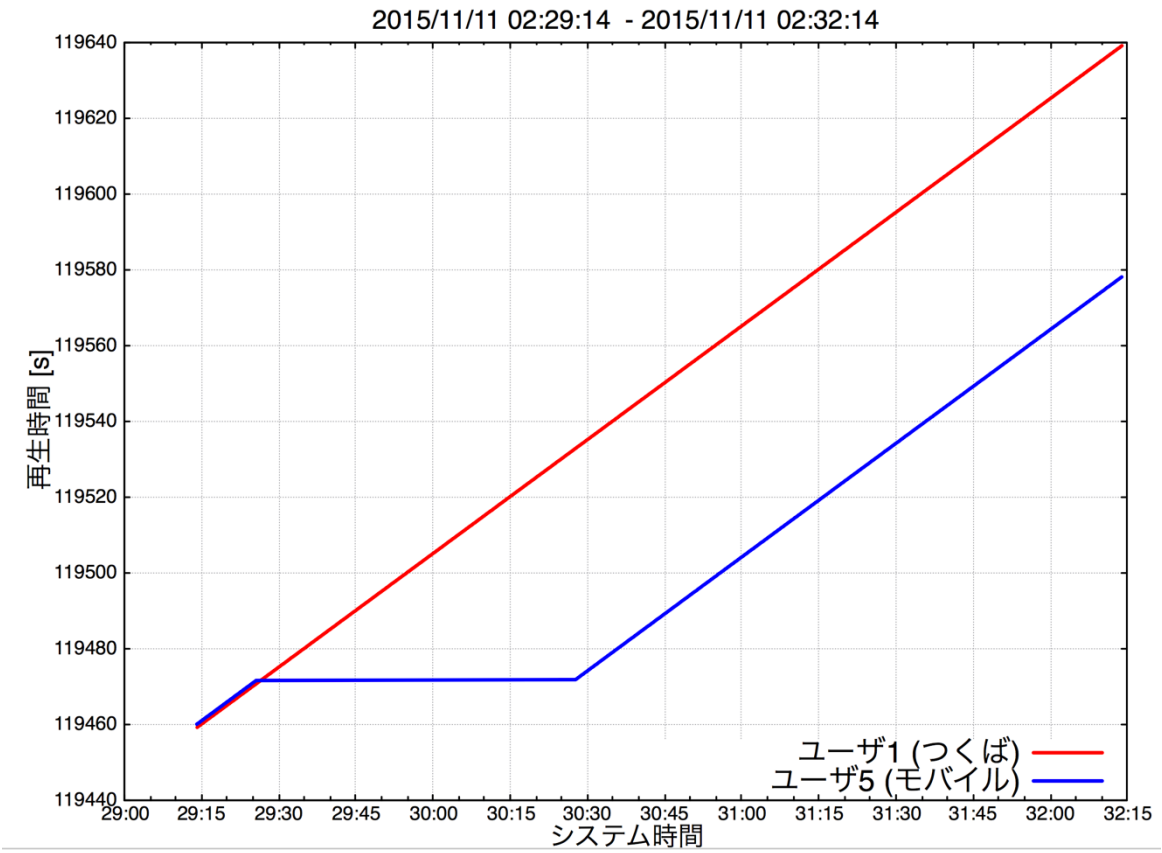

図 4. 実験 1: 再生位置のずれ 発生時の様子(1)

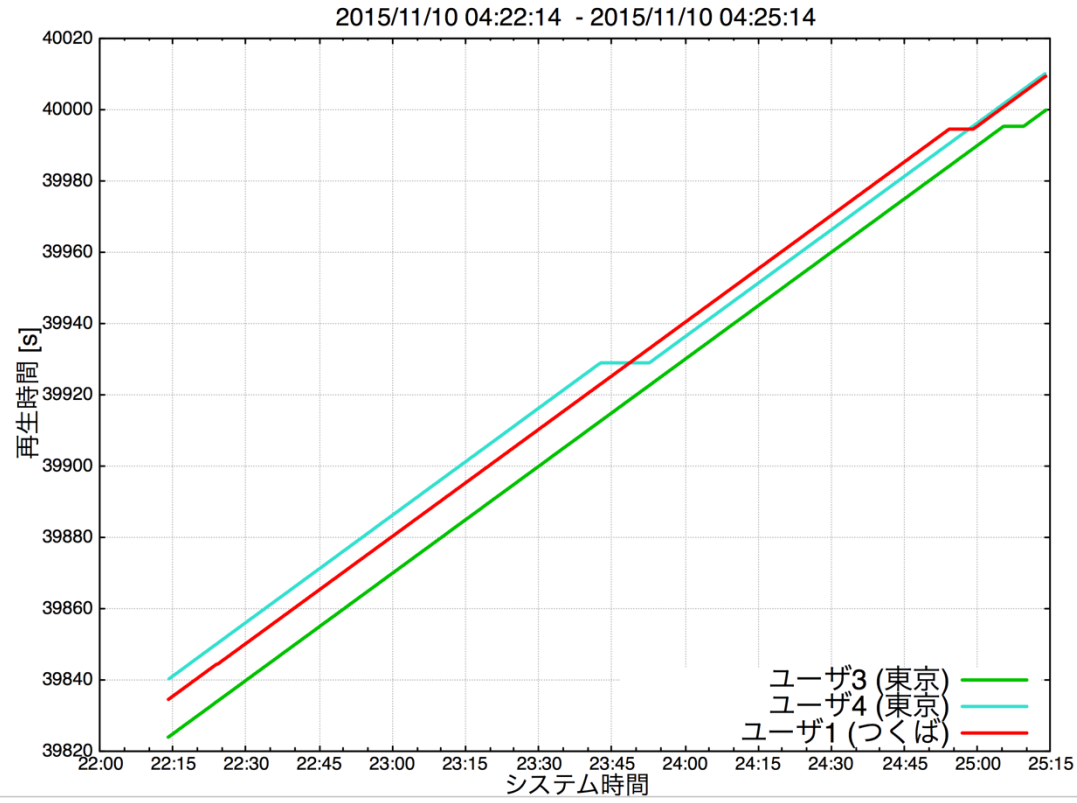

図 5. 実験 1: 再生位置のずれ 発生時の様子(2)

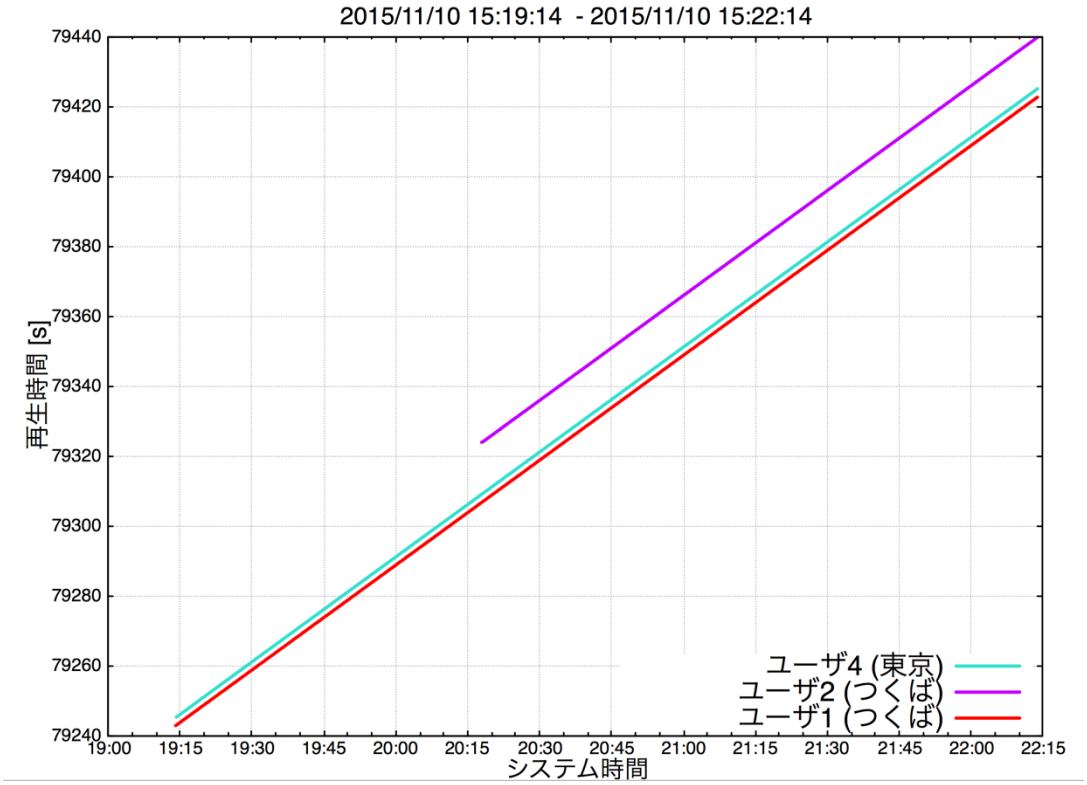

図 6. 実験 1: 再生位置のずれ 発生時の様子(3)

3.4 実験 2

3.4.1 実験条件

実験 2 では再生停止時のより詳細な状況を調査するため、表 1 に示した通り、取得する情報を増やして計測を行った。そして 6 地点 8 クライアントで、計 5 日間(2016/ 3/14 ～ 2016/3/18)の計測を行った。クライアントの計測位置は、つくば・筑波大学・東京(家庭用回線)・東京(モバイルルータ)・立命館大学・京都大学・広島大学・ストラスブール大学(フランス)である。また計測中、毎正時に全クライアントの動画再生ページを更新するものとし、1 時間ごとに再生位置をリセットするものとした。

3.4.2 実験結果

ネットワーク構成について

5 日間の計測において 193 個の相異なる発 IP アドレスを取得した。これは YouTube Live が CDN を利用して動画を配信しており、これらの CDN のキャッシュサーバのうち 193 個に接続した状況であると推定できる。また GeoIP[16]によってそれぞれの IP アドレスを検索した結果(表 3)、これらのキャッシュサーバのうち、そのほとんどが Google 管理下のアドレスであることがわかり、その中でも IP アドレスによって 4 グループに分けられることがわかった。またその他の IP アドレスとしては学術情報ネットワークである SINET[17]の IP アドレスのものが一つ、フランスの原子力・代替エネルギー庁の IP アドレスが 2 つ確認でき

た。このことから YouTube Live では Google が運用している CDN を利用していることが考えられる。また表 4 に示す各ユーザの接続サーバの比率から、筑波大学、立命館大学、京都大学、広島大学では SINET の IP アドレスへの接続比率が 99%以上であり、またそれ以外の場合、日本では Google のグループ 1 にあたる IP アドレスからの配信が 7 割以上となっており次いでグループ 2、グループ 4 の比率が高くなっている。フランスでは原子力・代替エネルギー庁管理下からの配信が 6 割、グループ 2 からの配信が 3 割であることがわかった。

表 3. 取得 IP アドレスの内訳

No.	管理者	IP アドレス取得数
1	Google(グループ 1: 173.194.0.0/16)	126
2	Google(グループ 2: 74.125.0.0/16)	98
3	Google(グループ 3: 208.117.0.0/16)	49
4	Google(グループ 4: 209.85.0.0/16)	10
5	情報・システム研究機構(SINET)	1
6	原子力・代替エネルギー庁(フランス)	2
7	不明	7

表 4. 接続サーバの比率

クライアント	1 位	2 位	3 位
つくば	グループ 1 (70.6%)	グループ 2 (24.8%)	グループ 4 (4.6%)
筑波大学	SINET (99.3%)	グループ 1 (0.7%)	グループ 2 (0.0%)
東京 (家庭)	グループ 1 (88.9%)	グループ 2 (9.4%)	グループ 4 (1.7%)
東京 (モバイル)	グループ 1 (76.7%)	グループ 4 (11.9%)	グループ 2 (11.4%)
立命館大学	SINET (99.0%)	グループ 2 (0.8%)	グループ 1 (0.2%)
京都大学	SINET (99.0%)	グループ 2 (0.9%)	グループ 1 (0.0%)
広島大学	SINET (99.1%)	グループ 2 (0.9%)	グループ 1 (0.0%)
フランス	原子力・代替エネルギー庁 (62.5%)	グループ 2 (31.0%)	グループ 1 (6.5%)

キャッシュサーバへの接続について

各ユーザの発 IP アドレスを分析したところ、動画を再生している間に IP アドレスが変化することは無く、毎正時の更新時にのみ変化することが確認できた。このことから、ユーザが接続するキャッシュサーバは動画の視聴開始時に選択され、以後はその選択されたサーバからデータを受信されていることがわかる。

再生停止の原因について

再生停止時の状況を調査した結果、バッファリングの終端位置と再生位置の差が縮まり、

再生位置がバッファリングの終端に追いつき、バッファに再生できるデータがなくなってしまうことによって再生停止が発生することがわかった。表 5 に各クライアントの再生停止回数とバッファリング不足による停止回数を示す。表 5 より、ほとんどのクライアントにおいて再生停止の原因はバッファ不足によるものであることがわかる。また正常時と再生停止時の様子を図 7、図 8 に示す。この図は 5 分間の様子であり、再生位置とバッファリングの終端位置(バッファ位置)、リクエストの RTT を示している。正常時の場合、再生位置とバッファ位置の間に常に余裕があり RTT も低くなっている。対して停止時の場合は、56 分程から バッファ位置が変わらなくなり、56 分 30 秒には再生位置がバッファ位置に追いついたことで再生停止が起きており、再生停止後もしばらくバッファ位置が変化せず 15 秒ほど再生が停止している様子が確認できる。また、このときの RTT を見ると、バッファが変化しなくなった時点と再生停止後の時点で RTT が高くなっており、リクエストの応答が返ってくるまでに時間がかかっていることがわかる。

表 5. バッファ不足による再生停止の比率

計測位置	再生停止回数	バッファ不足回数	比率[%]
つくば	25	24	96
筑波大学	16	16	100
東京 (家庭)	10	10	100
東京 (モバイル)	15	14	93.33
立命館大学	35	25	71.4
京都大学	24	23	95.83
広島大学	23	21	91.30
フランス	31	31	100

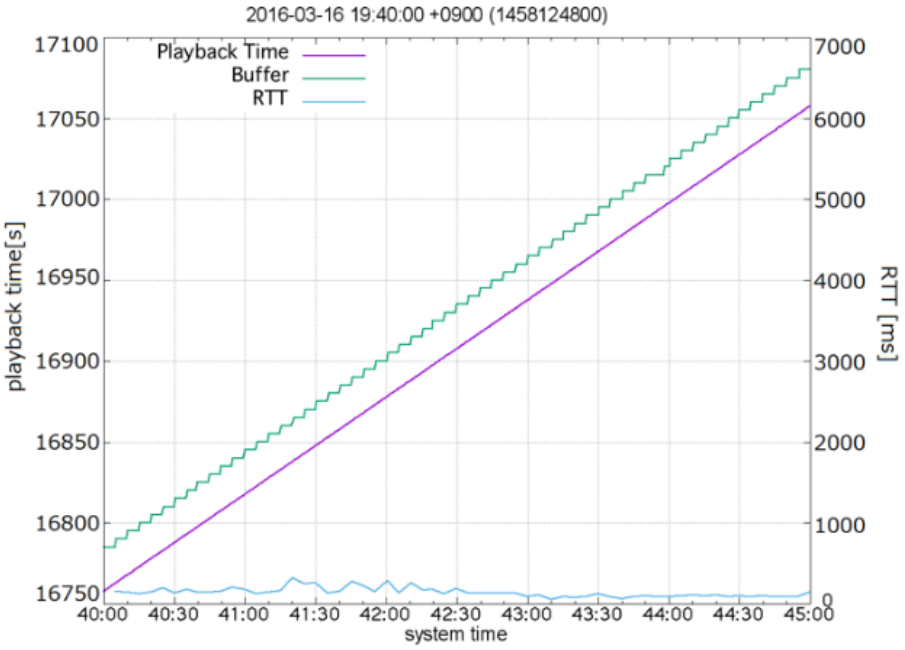

図 7. 再生時の様子(正常時)

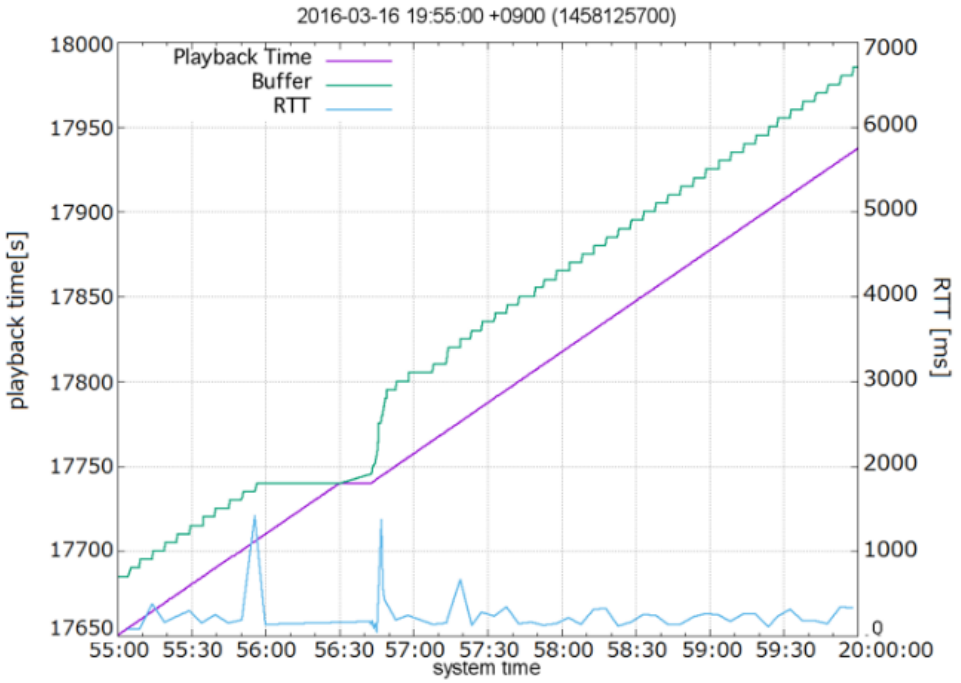

図 8. 再生時の様子(停止時)

再生停止の同時性について

図 9 にユーザごとの再生停止発生時刻とそのときの接続ミラーサーバを示す。再生停止の発生時刻を調査した結果、同時に複数のクライアントが再生停止する状況があり、またその状況が以下の 2 パターンであることを確認した。

1. 停止したクライアントが全て同一のキャッシュサーバに接続している場合
2. 停止したクライアントがそれぞれ異なるキャッシュサーバに接続している場合

図中①は SINET の同一サーバに接続している 3 クライアントが同時に再生停止している状況であり、②は、同時に停止したユーザが様々なサーバへ接続している状況である。
また、このような状況が発生したときこれらのパターンによって問題の発生箇所が推定できる。例えば図 10(a)の同一サーバの場合、そのサーバかサーバからユーザへのネットワークにおいて問題が起きている可能性が高い。また図 10(b)のように同時に停止したユーザが複数のサーバへ接続している場合は、配信者や配信者からオリジナルサーバへのネットワーク、オリジナルサーバからミラーサーバへのネットワークで問題が起きている可能性が高いと考えられる。

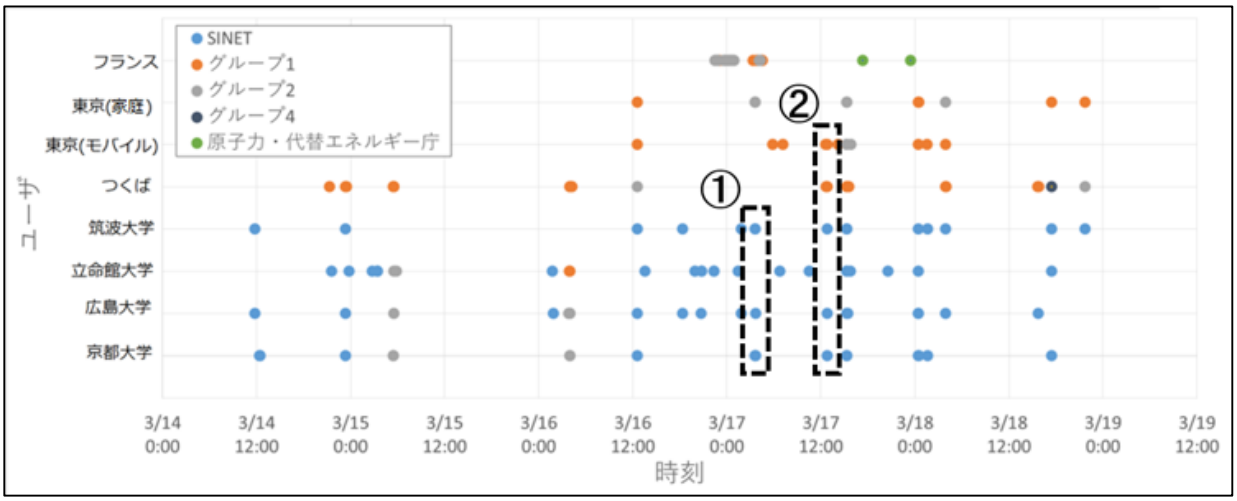

図 9. 再生停止の発生時刻と接続ミラーサーバ

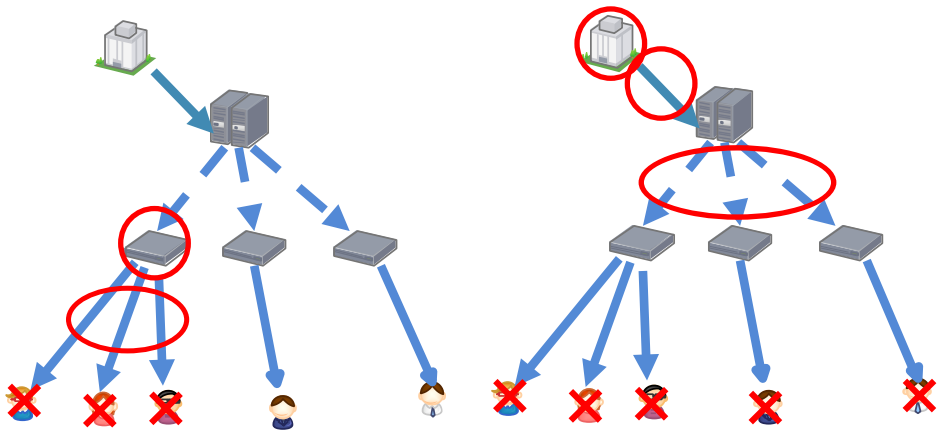

図 10. 複数ユーザの同時再生停止

3.4.3 まとめ

計測実験 2 ではより詳細な情報を取得し、ネットワーク状況や再生状況の分析をすることを目的として行った。その結果ネットワーク環境に関しては、YouTube Live において CDN を利用している状況や、接続するキャッシュサーバの偏りが確認できた。また再生中にキャッシュサーバの切り替わりは起きず、ページへの接続時にのみ切り替わりが起きるという動作も確認できた。再生停止時の状況に関して、再生停止の原因はデータが受信できなかったことによるバッファ不足であることが明らかになった。また複数クライアントが同時に停止するという状況が確認でき、そのときに停止したクライアントが接続しているキャッシュサーバによって、停止の原因となる箇所がある程度予測できるということを示した。

第4章　提案方式

計測実験により、ユーザの再生はデータが正常に受信できず、バッファに余裕が無くなったことで再生が停止するといった状況が多く確認できた。また一度ユーザ間の再生位置にずれが生じると、長時間ずれのある状態のまま再生が続いていく状況も確認した。

そこでユーザのバッファ不足による再生停止の解消と、ユーザ間のずれの解消を目的とした同期再生方式を提案する。提案する同期再生方式はネットワーク状況を改善し再生停止を解消するネットワーク制御と、クライアントの動画再生を制御しユーザ間のずれを解消するクライアント制御から成る。

4.1　構成要素と想定環境

提案方式の構成要素及び想定している環境を図 11 に示す。

コアネットワークはコアルータ同士が接続することで構築され、コアルータ間のリンクには帯域と送信キューの最大キュー長が設定されている。パケットは送信キューに入ってから順に送信されていき、このキュー長を超えるパケットは破棄される。またユーザは各コアルータ配下にいるものとし、コアルータ–ユーザ間の通信ではパケットドロップは起きないものとする。

キャッシュサーバは各コアサーバに設置されており、ユーザからのリクエストに応じてデータを送信する。キャッシュサーバには起動状態があり、停止しているキャッシュサーバへのリクエストは破棄される。またキャッシュサーバはリクエスト処理能力及びリクエストキューのキュー長が設定されており、リクエスト処理能力に応じてリクエストが処理され、リクエストキューのキュー長を超えるリクエストは破棄される。

動画再生を行うユーザは再生バッファを持っており、現在の再生位置と再生バッファ内のセグメントとの間にある程度の余裕ができるよう、キャッシュサーバにセグメントをリクエストする。セグメントのリクエストにはタイムアウト時間が設定されており、タイムアウト時間を過ぎた場合はもう一度リクエストを再送する。また動画の再生速度を持ち、現在の再生位置は再生速度に応じて更新される。現在の再生位置が、再生バッファ内のセグメントに追いつき、再生できるセグメントがなくなった場合は再生が停止し、また再生できるセグメントを受信したときに再生が再開する。またユーザは一緒に動画を視聴するグループに所属している場合があり、同期再生ではこのグループ単位で同期処理が行われる。

コントローラはネットワークやサーバの状況を監視し、状況に応じて制御を行うための制御ノードである。ネットワーク情報としてリンク使用率と送信キューのキュー長をコアルータから 1 秒間隔で収集し、サーバ情報としてはリクエスト受信数及びリクエストキュー長を各キャッシュサーバから収集する。そして、これらの負荷状況や制御要求に応じてネットワーク制御を行い、サーバやユーザへ指示を出す。

コーディネータはユーザの再生状況の監視し、同期制御を行う制御ノードである。同期グループに所属しているユーザは定期的にコーディネータへ現在の再生位置を通知する。通知を受けたコーディネータは、そのグループの再生位置同期の基準となる同期ターゲットを計算し、ユーザへ通知する。

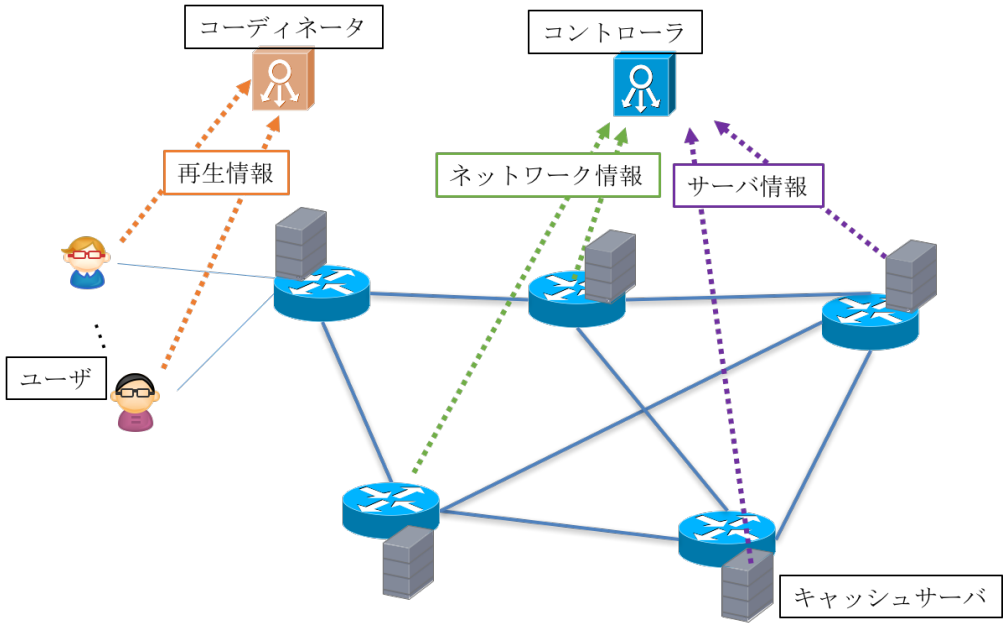

図 11. 構成要素及び想定環境

4.2 ネットワーク制御

4.2.1 サーバの動的切り替え

計測実験において、クライアントは、再生開始時に選択されたキャッシュサーバは再生が終了するまで変更されないということを確認した。しかしネットワーク状況は時々刻々と変化するため、接続時に最適であったとしても、ある程度時間が経つと動画再生に十分な帯域を確保できなくなるといった場合がある。そこでネットワーク制御として動画再生中に接続するサーバを動的に切り替え、正常にデータが受信できるようにする制御方式を提案する。

キャッシュサーバの切り替えはユーザからの制御要求によって行われ、その要求が発生するタイミングはクライアントがリクエストを送信してからデータ受信が完了せずタイムアウトした時点とした。ユーザはネットワークやサーバの負荷に関して詳細な情報を持っていないが、リクエストがタイムアウトするという状況ではネットワークまたはキャッシュサーバにおいて何らかの問題が発生していると考えられる。そのためユーザが負荷状況を考慮して制御要求を出すタイミングとして、リクエストのタイムアウト時が適切であると考えた。

また切り替え先のキャッシュサーバの選択はクライアントからサーバへのネットワーク状況とサーバの負荷を考慮したコスト式を用いて選択する。コスト式を以下に示す。

$$\mathrm{cost} = \alpha Umax(C, S_i) + \beta Request(S_i)$$

ここで C は切り替えを行うクライアントを示し、Si は切り替え候補となるキャッシュサーバを示している。そして Umax(C, Si)はサーバ Si からクライアント C への最短経路上の最大リンク使用率を示している。また Request(Si)はサーバ Si の負荷率を示しており、

負荷率 $= \frac{\text{受信リクエスト数}}{\text{処理可能リクエスト数}}$によって求められる。そして α、β は重みであり、ネットワーク状況とサーバ負荷のどちらを重視するかによって変化させる。また、ここでのリンク使用率及び受信リクエスト数はコントローラが 1 秒間隔で取得しており、コスト式の計算では直近 5

秒の平均値を使用している。

サーバ切り替えの流れを以下に示す。

1. クライアントのリクエストがタイムアウトし、制御要求をコントローラへ送信
2. コントローラが現在起動している全サーバに対してコスト式を計算
3. 計算したコストが最小となったサーバを切り替え先として選択
4. コントローラがクライアントへ切り替え先サーバを通知
5. クライアントが切り替え先サーバへ接続し、リクエストを再送

4.2.2 サーバ起動制御

全キャッシュサーバを起動することで、コアネットワークへトラヒックが流れなくなり、ネットワーク負荷が解消する。しかし全てのコアルータにキャッシュサーバを配置し、起動し続けることは非常に高いコストがかかり現実的ではない。そのためいくつかのコアルータを選択してキャッシュサーバを設置することになるが、このときどのコアルータへキャッシュサーバを設置するべきかを決定することは困難である。こうした問題を解決するために、提案方式では予め全てのコアルータにキャッシュサーバを設置しておきネットワーク状況に応じて起動・停止することで、ネットワーク状況にあったキャッシュサーバの配置を行う。

サーバ起動を行う条件は以下の通りである。1 秒間隔でこの条件を確認し、条件を満たす場合にサーバ起動を行う。

1. 5 秒間の平均リンク使用率が 80%超えるリンクが存在
   または
   5 秒間の平均受信リクエスト数をもとに計算した負荷率が 80%を超えるサーバが存在
2. 前回サーバを起動してから 5 秒以上経過している

起動するキャッシュサーバは以下のようにして決定する

1. 全てのコアルータ間リンクを調査し、以下の値が最も大きくなるリンクを探索する
   - 上り側のルータのサーバが起動している場合:
     $0.5 \times$リンク使用率 $+ 0.5 \times$サーバ負荷率
   - 起動していない場合: リンク使用率
2. 選択されたリンクの上り側ルータに設置されているサーバの起動状態で分岐
   - 起動している場合:
     選択されたリンクの下り側ルータに接続されているリンクのうち最もリンク使用率が高いリンクの下り側ルータのサーバを起動する
   - 停止している場合:
     そのリンクの下り側ルータのサーバを起動する
3. 最も負荷率が低いサーバを停止

2.においてサーバ起動が起動している場合に、下り側ルータのサーバを起動するのではなく、さらに 1 ホップ先のルータのサーバを起動する理由は、サーバが起動している隣接ルータにさらにサーバを起動してしまうと、サーバが 2 つ並ぶことになり、その 2 つのルータ間のリンクにトラヒックが流れづらくなってしまい、それによってネットワーク全体の利用効率が下がってしまうと考えたためである。

4.3 クライアント制御

ネットワーク制御によってクライアントが正常にデータを受信できるようになったとして

も、すでにクライアント間にずれが生じている場合、そのずれは解消されない。クライアント制御ではネットワーク制御によってデータの通信が正常になった状態でクライアントの再生を制御することで、クライアント間のずれを解消する。

クライアント制御ではクライアントの動画再生速度を変更することによって行う。再生速度変更の流れを以下に示す。

1. 同期グループに所属しているクライアントが現在の再生位置をコーディネータへ送信する
2. コーディネータが、このクライアントの同期グループ内の平均再生位置を計算し、クライアントへ返す
3. クライアントの再生位置と同期グループの平均再生位置との差を取り、その差の大きさによって以下のように動画再生速度を変更する
   - クライアントの再生位置 – 同期グループの平均再生位置 > 2 秒:
     同期グループ内において、再生位置が進んでいるため、再生速度を 0.9 倍にする
   - クライアントの再生位置 – 同期グループの平均再生位置 < −2 秒:
     同期グループ内において、再生位置が遅れているため、再生速度を 1.1 倍にする
   - 既に再生速度が変更されている場合:
     平均再生位置との差の絶対値が 1 以下になった時点で、再生速度を 1.0 倍(通常)に戻す

このようにして再生速度を変更することで、グループ内のユーザの再生位置が平均に近付いていき、ずれを解消することができる。

第5章　評価実験

5.1　実験環境

提案方式の評価を行うため、ネットワークシミュレータである ns-3[18]を利用してシミュレーションを行った。シミュレーションに使用したトポロジである Palmetto を図 12 に示す。このトポロジは米国の ISP である Spirit Communications によって実際に運用されているバックボーンネットワーク[19]であり、ノード数 45、平均次数 3.11 のトポロジとなっている。

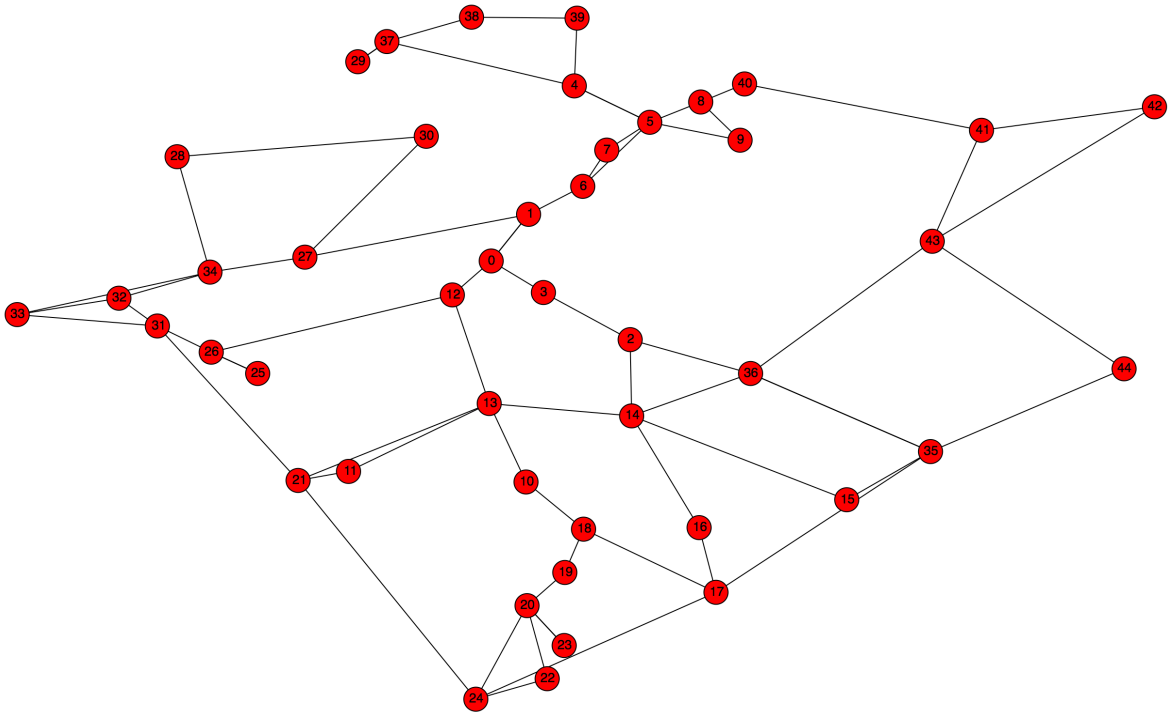

図 12. Palmetto

その他のシミュレーション条件を表 6 に示す。ユーザのバッファサイズやリクエストのタイムアウト時間、1 セグメントあたりの再生時間やファイル容量は YouTube Live での計測で取得した値をもとにシミュレーション規模に合わせてスケーリングしたものである。また、サーバのリクエスト処理能力及びリクエストキュー長は実際にラップトップ上で Web サーバとして Apache httpd を起動し、ベンチマークを行った結果をもとに設定している。

表 6. シミュレーション条件

条件名	値
シミュレーション時間	100 秒
コアルータ間リンクのリンク帯域	100Mbps
コアルータの送信キュー長	1000 パケット
ユーザ数	2250 (1 コアルータあたり 50 ユーザ)

ユーザの再生開始時間	シミュレーション開始後 10-20 秒(ランダム)
1 グループあたりのユーザ数	10 ユーザ
グループ数	67 (全ユーザの 3 割がグループに所属)
ユーザがバッファするセグメント数	最大 3 セグメント
リクエストのタイムアウト時間	4 秒
1 セグメントの再生時間	4 秒
1 セグメントのファイル容量	100k Byte
1 セグメントのパケット数	68 パケット(1 パケット 1500Byte)
サーバのリクエスト処理能力	200 リクエスト / 秒
サーバのリクエストキュー長	100 リクエスト
コントローラの情報取得間隔	1 秒
コーディネータの情報取得間隔	1 秒

また、シミュレーションでは動画配信以外のネットワーク負荷を模擬するために背景負荷を設定した。背景負荷は正弦波によるモデルとして、以下のように求めた。

$$Traffic = B + V \times \sin(t + \theta_i)$$

ここで B は基準となるトラヒック容量であり、V が変動するトラヒック容量を示す。そしてθ_iがリンク i の初期位相である。これによってリンクごとに$B \pm V$の範囲で背景負荷をかけることができる。

またユーザが再生開始時に接続するサーバは最短ホップ数のサーバであるものとした。最短ホップ数のサーバが複数ある場合は、その中からランダムで選択する。

評価実験はネットワーク制御として、サーバの動的切り替えのみを行う場合と、動的切り替えとサーバ起動の両方を行う場合の 2 種類を実験した。クライアント制御に関してはどちらの方式でも実行されているものとした。

5.2 実験 1: サーバの動的切り替え

実験 1 ではネットワーク制御としてサーバの動的切り替えのみを行う場合の評価をする。実験はサーバ切り替えを行なわない場合と行った場合との結果を比較する。サーバの起動数は 5 台とし、ランダムで選択した。図 13 にサーバの配置を示す。ランダム選択の結果、9, 16, 19, 24, 26 のサーバが起動されるノードとして選択された。また各リンクの太さはユーザが最短ホップ数のサーバに接続した場合に通ると推測されるトラヒック量をおおまかに示したものであり、線が太くなるほど、多くのトラヒックが通過することを示す。背景負荷としては基準トラフィック容量を 20Mbps、変動トラフィック容量を 20Mbps とし、各リンクの初期位相をランダムで設定した。また実験結果は、定常状態になってからの情報を見るために、シミュレーション開始後 30 秒から 100 秒までのデータのみを対象とする。

シミュレーション終了時のクライアントごとの総再送回数を、切り替え無しの場合を図 14 に、有りの場合を図 15 に示す。結果より、全体の総再送回数は切り替え無しで 7223 回なのに対し、切り替え有りでは 1357 回と大幅に再送回数を削減することができた。また、再送が発生したクライアントの平均再送回数は切り替え無しで 8.3 回、切り替え有りで 2.1 回となった。

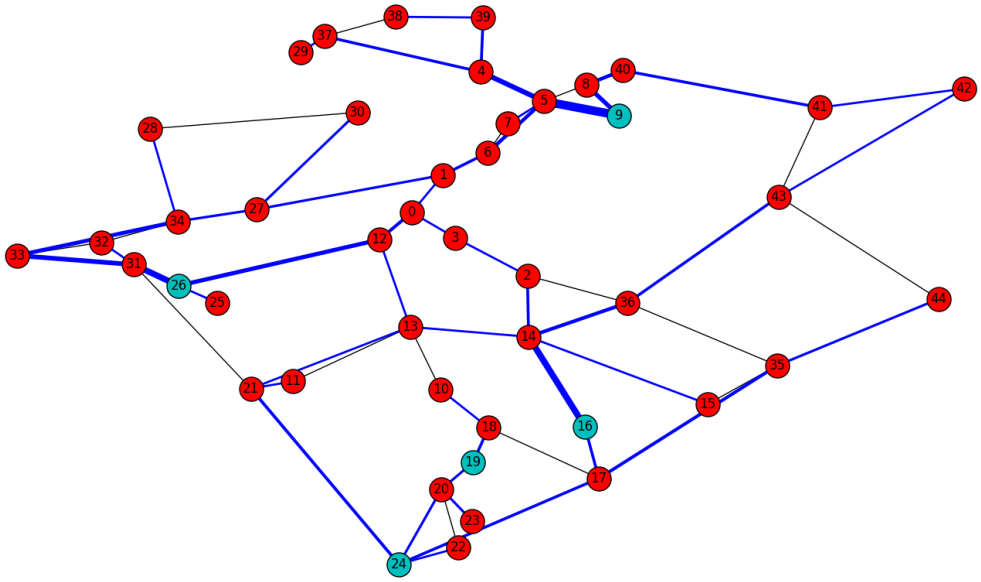

図 13. サーバ配置(実験 1)

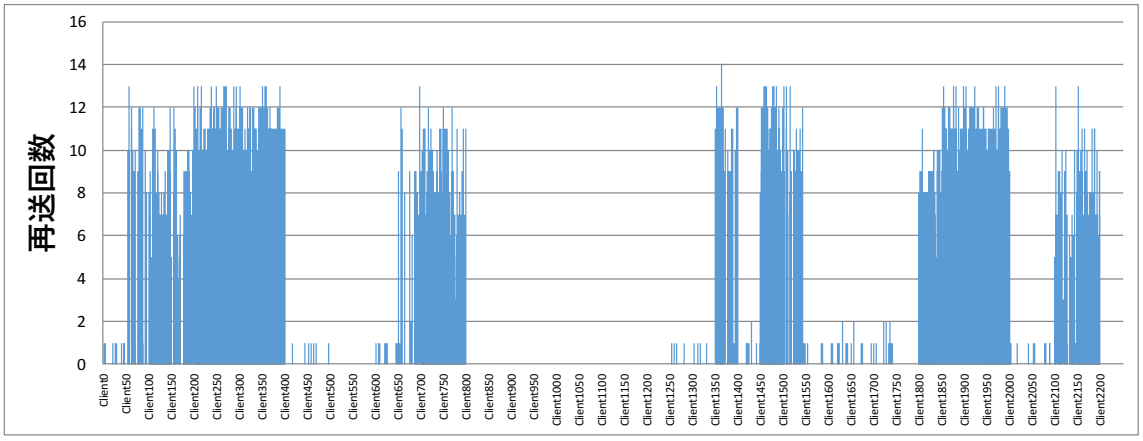

図 14. クライアント別 総再送回数(サーバ切り替え無し)

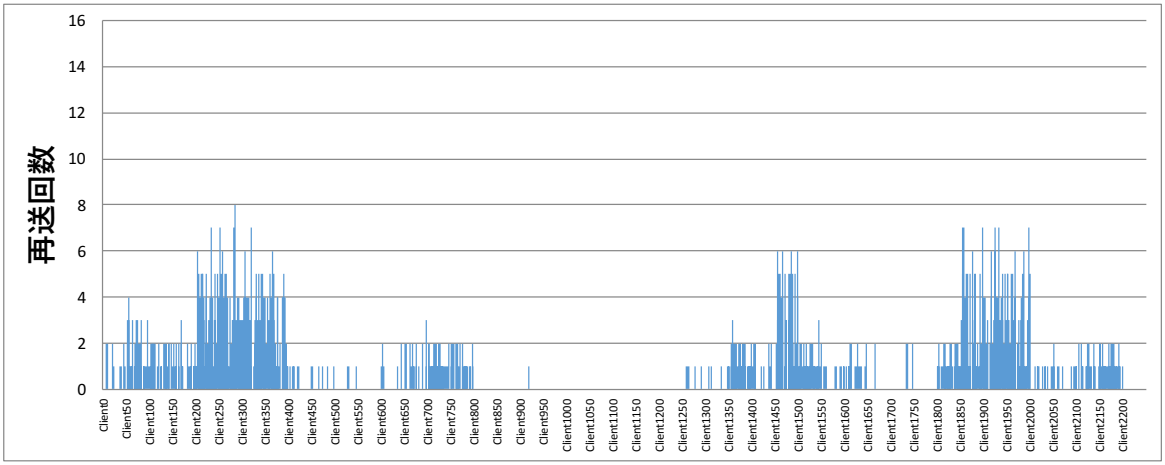

図 15. クライアント別 総再送回数(サーバ切り替え有り)

次に、接続サーバごとの再送が行われた回数の累積値を、切り替え無しの場合を図 16 に、切り替え有りの場合を図 17 に示す。切り替え無しの場合では再送が行われたのはコアルータ 16, 26, 9 に配置されたサーバのみであるが、コアルータ 26 のサーバに関してはほぼ再送は発生していない。このように再送回数に大きく差が開く理由は、ユーザが最も近いサーバに接続することにより、リクエストが集中しやすいサーバと集中しにくいサーバに偏りが生まれたためであると考えられる。特にコアルータ 9 に配置されたサーバの再送回数が多く、シミュレーション時間が 70 秒あたりになるまで常に再送が発生している。対して切り替え有りの場合、全サーバで再送が発生しているが、再送回数は全体的に少なくなっており、特定のサーバにリクエストが集中せず、分散できていることがわかる。また、50 秒程度で再送が収まっており、サーバ切り替えによって通信状況が改善できていることがわかる。

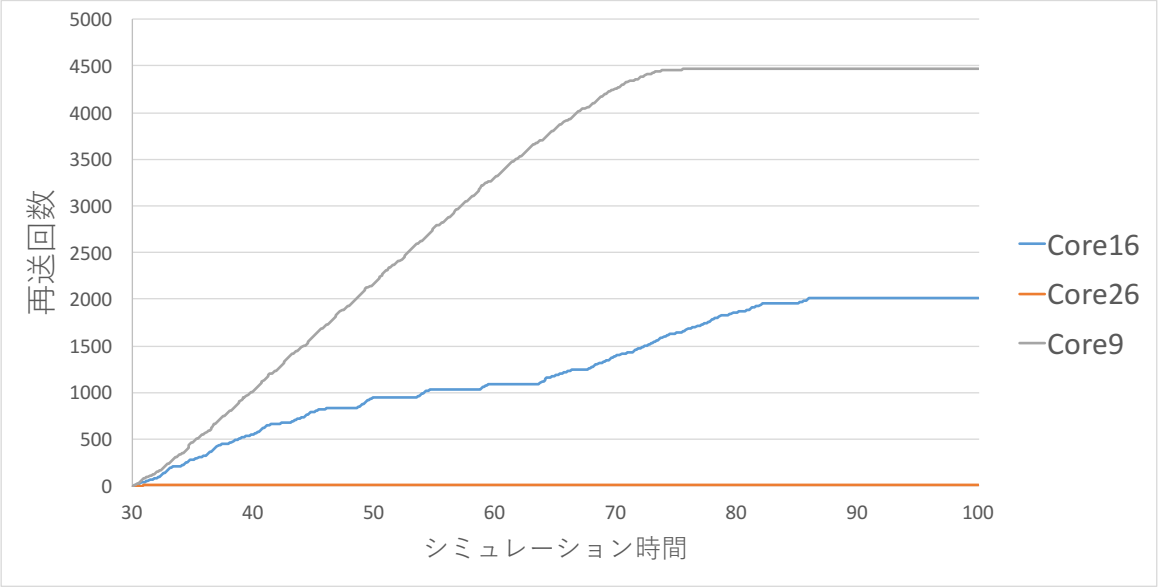

図 16. 接続サーバごとの累積再送回数（切り替え無し）

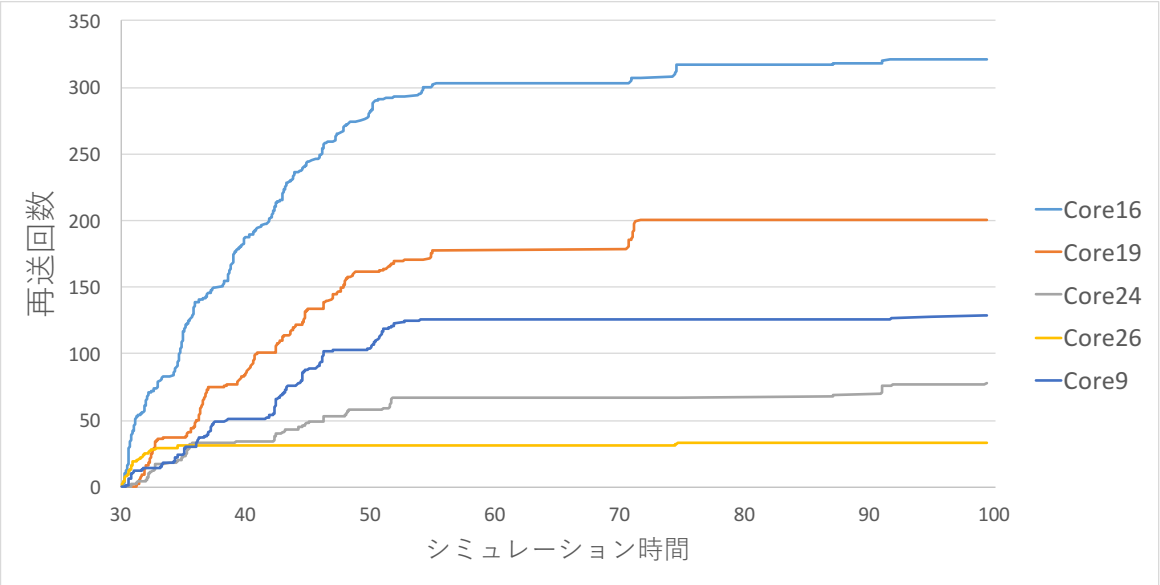

図 17. 接続サーバごとの累積再送回数（切り替え有り）

次にネットワーク負荷の状況として、輻輳リンクのリンク使用率を示す。切り替え無し(図18)の場合、9－5 リンクと 16－14 リンクにおいて、リンク使用率が長期間 100%近くになっており、この輻輳リンクでパケットドロップが発生するため、コアルータ 9 及びコアルータ 16 に設置されているサーバの再送回数が高くなったと考えられる。また、コアルータ 26 に設置されているサーバに関しては、26－12 リンクと 26－31 リンクの二方向から同程度のリクエストをしていたため、パケットドロップが発生するようなリンク輻輳が発生せず、再送回数が少なくなったと考えられる。次に切り替え有りの場合のリンク使用率の推移を図 19 に示す。切り替え無しの場合で長時間輻輳していた 9－5 リンク及び 16－14 リンクは 100%に達することはあるが、長時間の輻輳は見られなかった。また、切り替え無しの場合では輻輳していなかった 0 － 1 リンクの輻輳が確認できた。これは、コアルータ 9 の近くにいるユーザがコアルータ 9 のサーバから他のサーバへ切り替えたことによって、トラヒックが 0－1 リンクに集中したのだと考えられる。

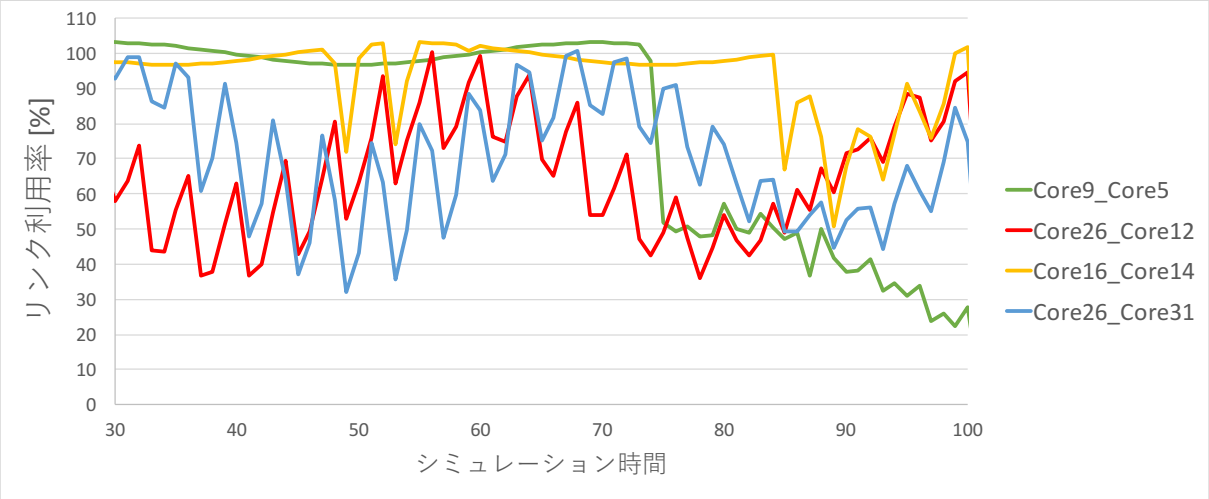

図 18. 輻輳リンクのリンク使用率（切り替え無し）

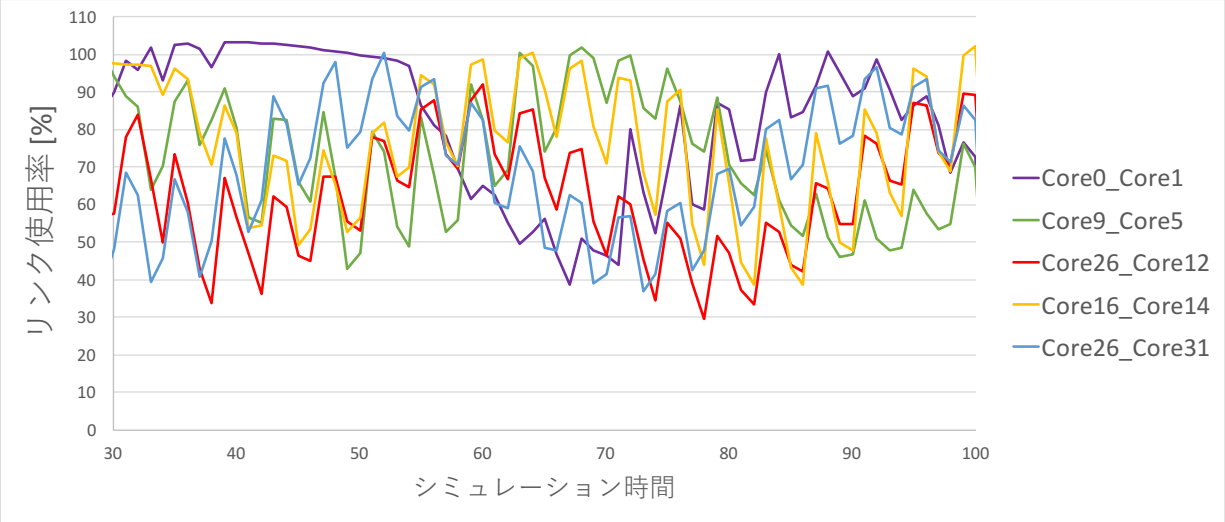

図 19. 輻輳リンクのリンク使用率（切り替え有り）

最後にユーザ間の再生同期について、グループごとの再生位置の差を、切り替え無しの場合を図 20 に、有りの場合を図 21 に示す。これはグループ内で最も再生位置が進んでいるユ

ーザと最も遅れているユーザとの間の再生位置の差を示したものであり、差の大きさが最大のグループから上位 10 グループを示している。切り替え無しの場合では、70 から 80 秒程度までグループ内の再生位置の差は広がり続け、最大では 50 秒を超える差が確認できる。その後、再生停止が抑えられクライアント制御によって再生位置の差が徐々に小さくなっていく様子が確認できる。切り替え有りでも同様の動作をしているが、最大の差が 30 秒程度と小さくまた、50 秒前後で差の広がりが抑えられている様子も確認できた。このことからサーバ切り替えを行うことによって再生位置のずれが解消し、クライアント制御での同期制御が有効に働く状況を、より早く整えられることがわかる。

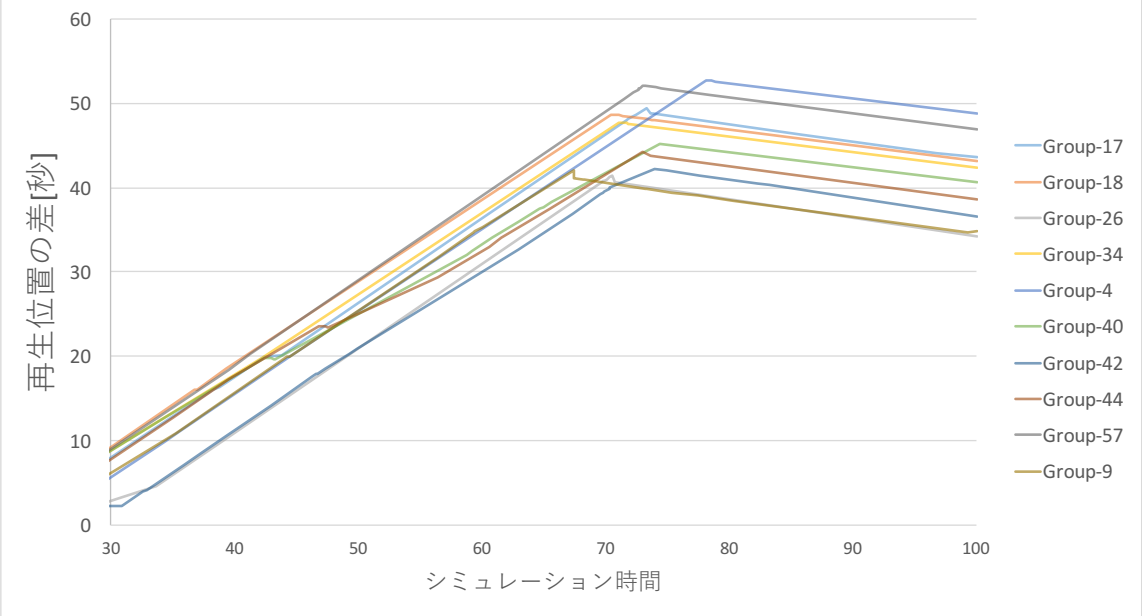

図 20. グループ内の再生位置の差（切り替え無し）

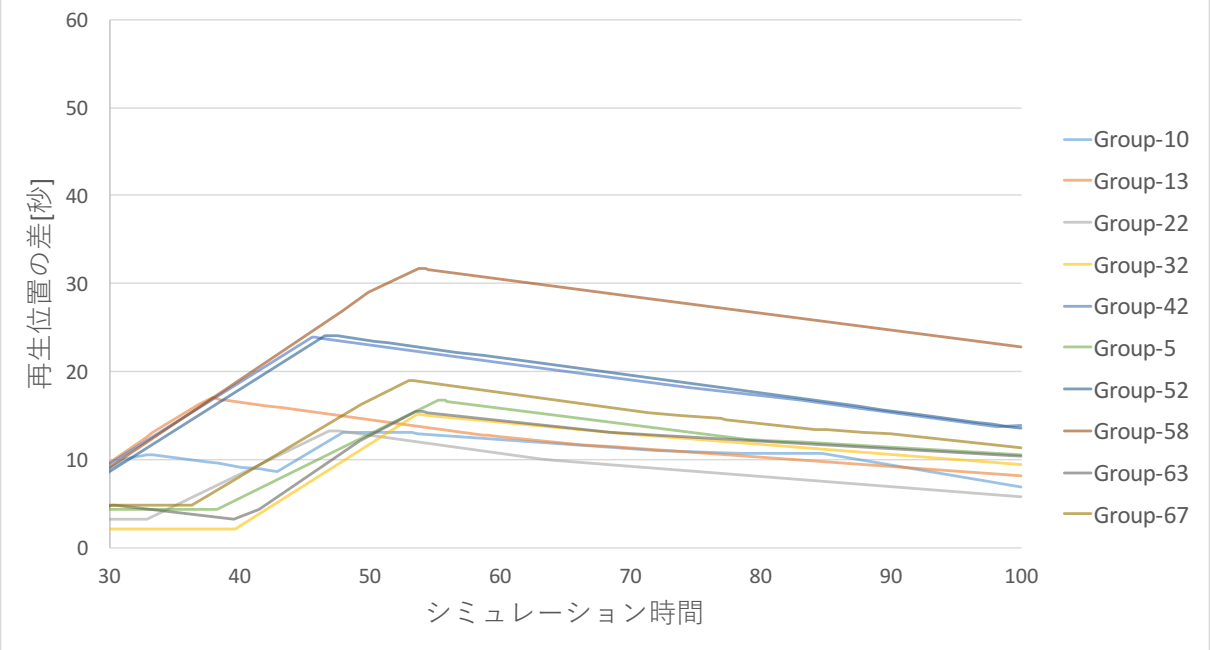

図 21. グループ内の再生位置の差（切り替え有り）

5.3 実験 2: サーバ起動制御

実験 1 ではサーバ切り替えによって、輻輳状態が解消され再生状況が改善されることを示した。しかしサーバ切り替えはすでに起動しているサーバを切り替えるため、そのサーバの配置に大きく影響を受けるため、その配置によってはサーバ切り替えが有効に働かない場合がある。実験 2 では、実験 1 とは異なるサーバ配置によってサーバ切り替えのみでは再生状況が改善できないことを示し、その後サーバ起動制御によって、再生状況が改善されることを示す。

実験 2 での初期サーバ配置を図 22 に示す。実験 1 でのサーバ配置(図 13)はサーバ同士が比較的分散して配置されていたのに対し、実験 2 でのサーバ配置はコアルータ 11, 23, 24, 25, 31 とサーバ同士が近くに集中している配置となっている。

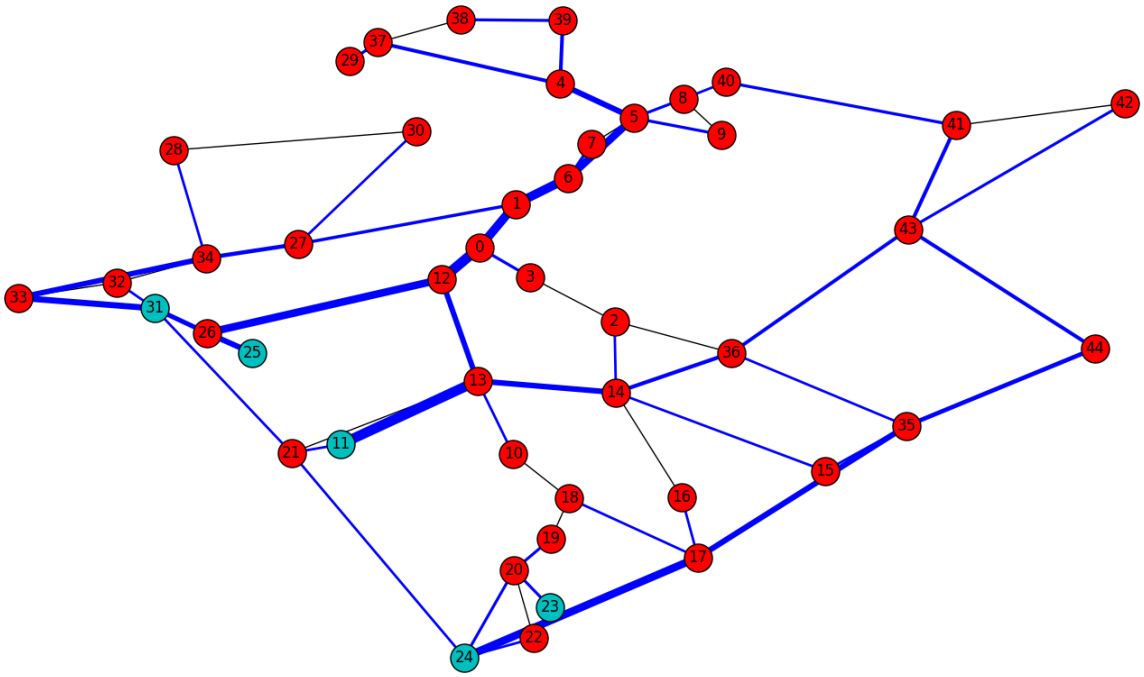

図 22. サーバ配置(実験 2)

また、実験 1 と同様に背景負荷としては基準トラフィック容量を 20Mbps、変動トラフィック容量を 20Mbps とし、各リンクの初期位相をランダムで設定した。またシミュレーション開始後 30 秒経過するまではサーバ起動は行われないものとした。

サーバ切り替えのみでのシミュレーション結果として、クライアントごとの総再送回数を図 23 に、接続サーバごとの再送回数の推移を図 24 に示す。全再送回数は 13637 回で、再送が発生したクライアントの平均再送回数は 8.7 回であった。図 24 より、シミュレーション終了時まで、常に再送が発生し、サーバ切り替えによって負荷が軽減できていないことがわかる。また輻輳リンクのリンク使用率(図 25)においても、常に 1 本以上のリンクがリンク使用率 100%に達しており、サーバ切り替えが効果的に働いていない。

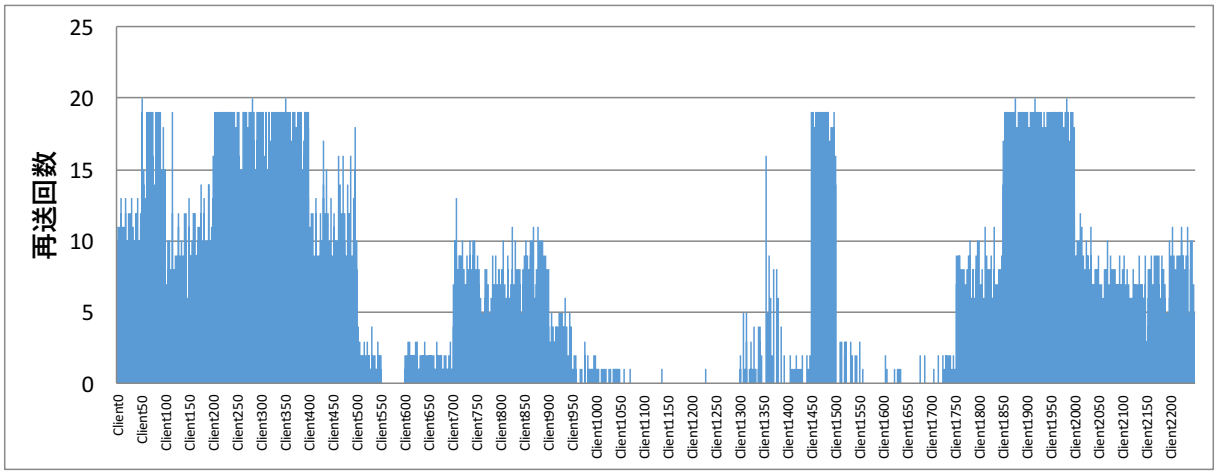

図 23. クライアントごとの総再送回数（サーバ起動無し）

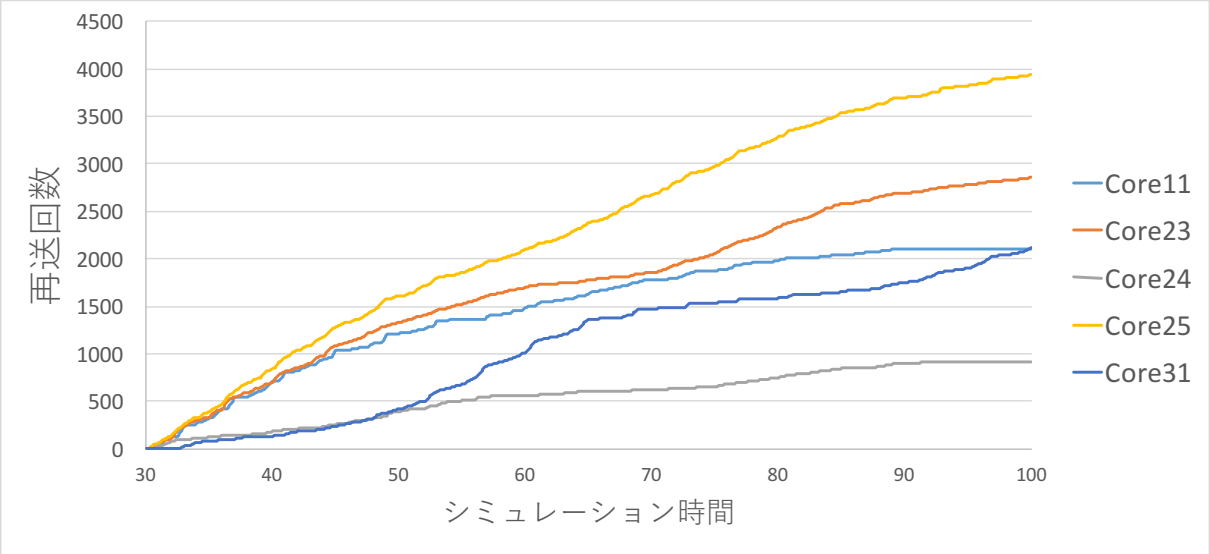

図 24. 接続サーバごとの累積再送回数（サーバ起動無し）

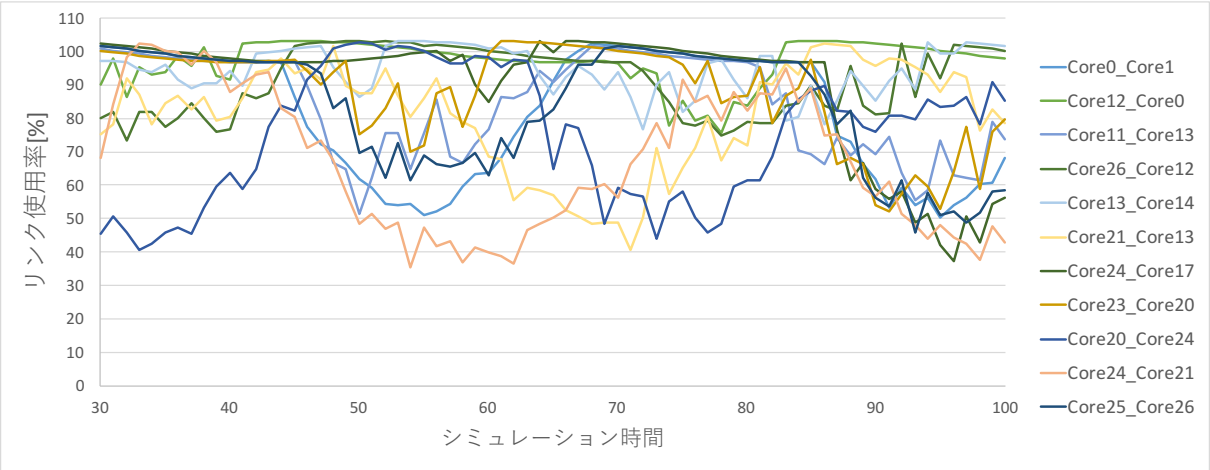

図 25. 輻輳リンクのリンク使用率(サーバ起動無し)

次にサーバ起動制御を行った場合の実験結果を示す。まずクライアントごとの総再送回数を図 26 に示す。全再送回数は 5321 回であり、サーバ起動無しの場合の約 4 割まで減少した。また再送が発生したクライアントの平均再送回数は 3.8 回であった。次にリンク使用率が 100%に達したリンクのリンク使用率推移を図 27 に示す。60 秒程度までは 100%に達しているリンクがあるが、その後、全リンクのリンク使用率が減少し、100%に達するリンクはなくなっている。このことから、サーバ起動によって輻輳が解消されていることがわかる。

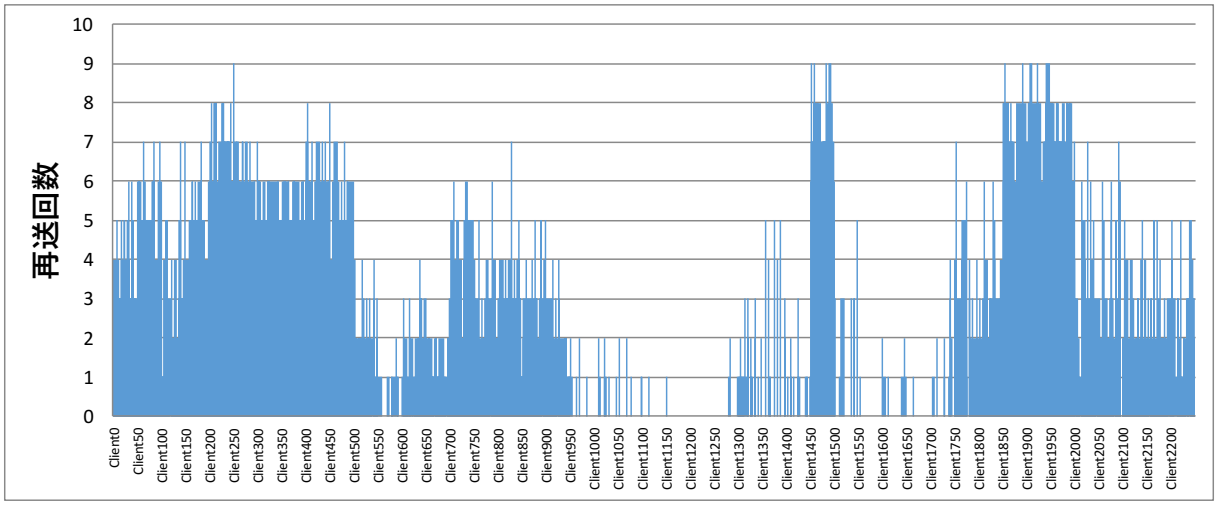

図 26. クライアントごとの総再送回数（サーバ起動有り）

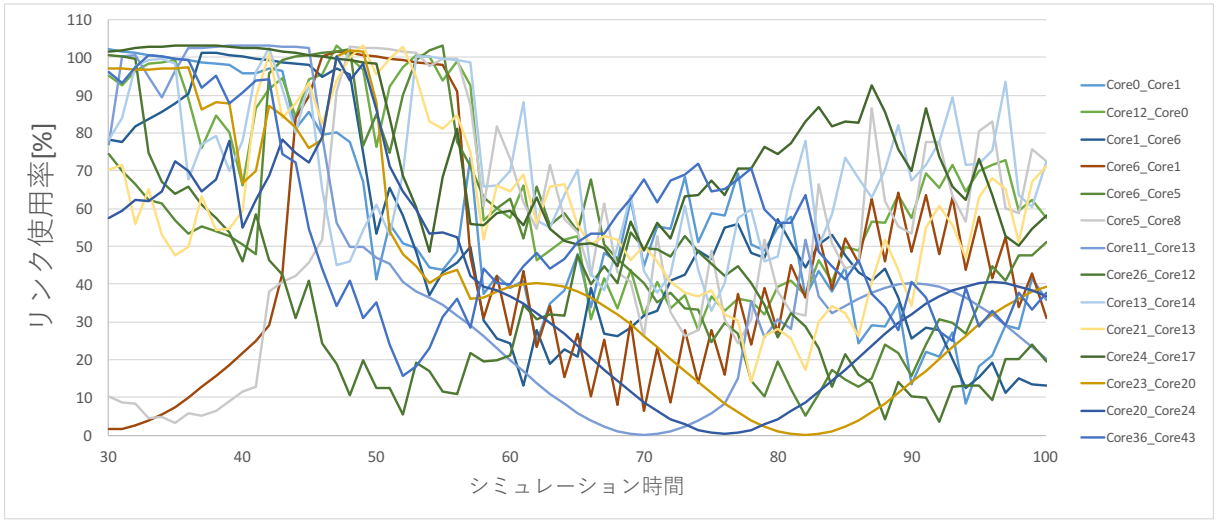

図 27. 輻輳リンクのリンク使用率（サーバ起動有り）

次に、サーバ起動制御によるサーバ配置の変化を表 7 に示し、シミュレーション終了時のサーバ配置を図 28 に示す。サーバ起動制御によって、近くに固まってサーバが配置されていた状態から、全体に分散して配置される状況となった。また、次数が高いルータがサーバの設置場所として選ばれていることから、ハブとなっているようなルータのサーバを起動することによって効率的に負荷が分散できているがわかる。

表 7. サーバ起動/停止の様子

シミュレーション時間	起動サーバ	停止サーバ
30	12	25
35	1	12
40	6	1
45	5	11
50	13	23
55	0	6
60	34	0
68	33	34
76	11	33
83	34	11
88	35	34
97	0	35

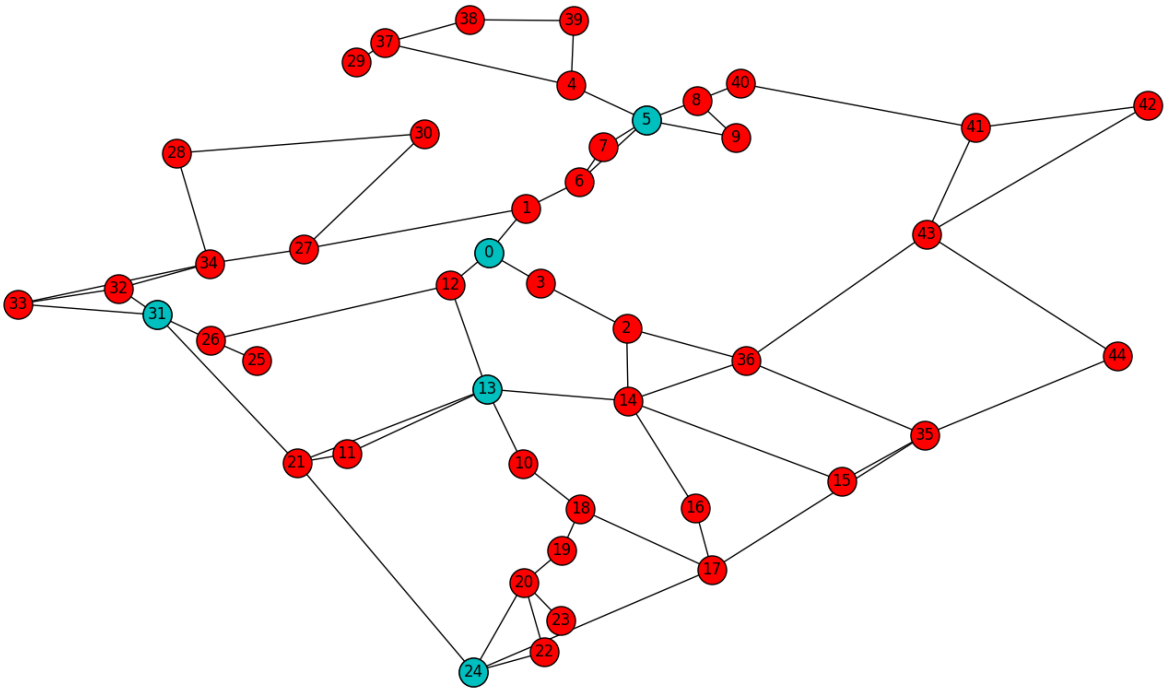

図 28. シミュレーション終了時のサーバ配置

最後にグループごとの再生位置の差を、サーバ起動無しの場合を図 29 に、サーバ起動有りの場合を図 30 に示す。サーバ起動無しの場合、再生停止によりシミュレーションが終了するまで再生位置の差が広がり続けていく様子が確認できた。対してサーバ起動有りの場合は 55 秒あたりで再生位置の差の広がりは抑えられ、それ以降はクライアント制御によって差が小さくなっていく様子が確認できた。このことからサーバ起動制御を行うことで、サーバ切り替えだけでは対処できない、サーバ配置の問題を解消し、再生状況を正常できることがわかる。

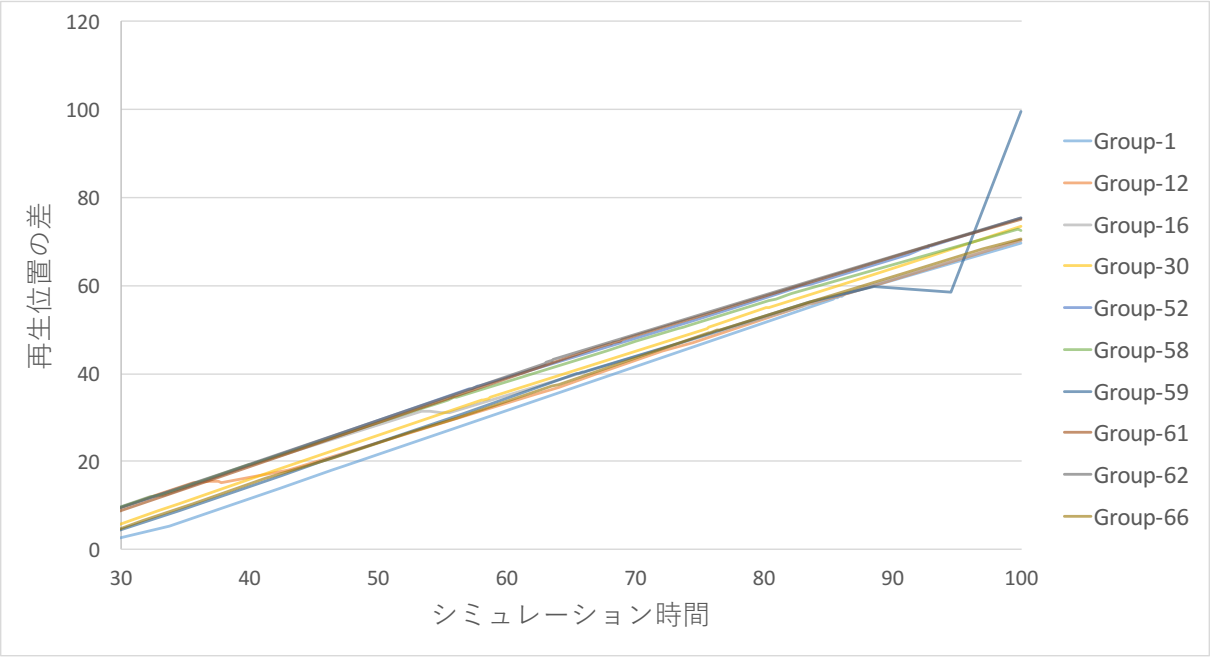

図 29. グループ内の再生位置の差（サーバ起動無し）

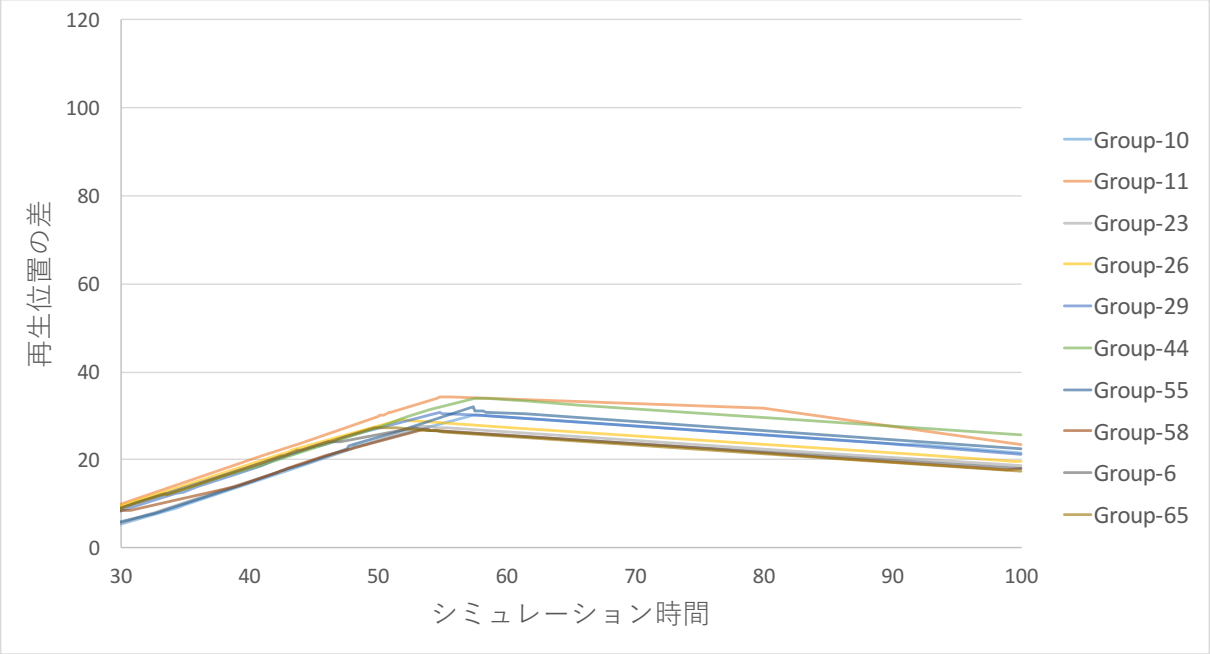

図 30. グループ内の再生位置の差（サーバ起動有り）

第6章　結論

本研究は、インターネットを利用したライブ配信サービスにおいて、離れた場所にいる人と一緒に動画を見る際に必要となる再生位置同期の方式を提案したものである。本論文ではまず、既存のライブ配信サービス利用時の再生状況を調査するために実装した計測システム及び計測ツールを紹介した。そしてこれらを用いてライブ配信サービスである YouTube Live を利用し、複数地点での計測実験を行った。その結果、再生停止が頻繁に発生しユーザ間に大きなずれが発生すること、再生停止の原因がデータ受信失敗によりバッファ不足であることを明らかにした。

この結果を受けて、ネットワーク状況の改善を目的としたネットワーク制御と、ユーザ間の再生位置のずれを解消することを目的としたクライアント制御の２つから成る同期再生方式を提案した。ネットワーク制御では、動画再生中にネットワーク状況を考慮して選択されたキャッシュサーバへ接続を切り替えるサーバ切り替えと、自動的に適切なサーバ配置へ近付けるためにサーバの起動・停止を行うサーバ起動制御を提案した。またクライアント制御では、同期グループ内での平均再生位置と自分の平均再生位置との差による再生速度変更により、グループ内の再生位置のずれを解消していく方式を提案した。

提案方式の評価はライブ配信を模擬したネットワークシミュレータを用いて行った。サーバ切り替えでは切り替えを行わない場合と比較し、リクエストの再送回数を 8 割以上削減することができた。また、ユーザ間の同期に関しても切り替え無しの場合よりも早い段階で差の広がりが抑えられることが確認できた。しかし、サーバ切り替えでは起動しているサーバの配置の影響を大きく受けることも確認できた。特にサーバ同士が比較的近い位置に集まっているような場合、サーバ切り替えを行っても十分な効果が得られずネットワーク状況が改善できなかった。こうした状況においてサーバ起動制御を行い、より適切な場所へサーバを起動していくことによってサーバ切り替えが有効に働くようになり、その結果、サーバ切り替えのみの場合と比較して、4 割以上リクエストの再送回数を削減した。

結論として、ネットワーク状況の改善を目的としたネットワーク制御とユーザ間のずれを解消することを目的としたクライアント制御から構成される提案方式が、ライブ配信サービスを利用したユーザ間の同期再生のための方式としての有効性を示した。

謝辞

本研究に際し、研究目標や研究手順、論文執筆に至るまで終始適切な助言を賜り、手厚く指導して下さった指導教員である川原崎雅敏教授に深く感謝致します。森嶋厚行教授にはお忙しい中副指導教官を受けてくださったことと共に、発表の質疑応答の際には助言や指摘を頂き、感謝致します。また IIJ イノベーションインスティテュート主幹研究員である新麗さんには本研究テーマのきっかけを頂くとともに、計測実験において適切な助言や実験環境の提供など様々な点でサポートをしていただき深く感謝致します。また、計測実験にご協力頂きました、広島大学情報メディア教育研究センター 近堂徹准教授、立命館大学情報理工学部山本寛准教授、京都大学大学院情報学研究科 津崎善晴氏、ストラスブール大学 Cristel Pelsser 教授に感謝致します。そして川原崎研究室の皆様には研究における議論などにおいてお世話になりました。ここに感謝の意を表します。

参考文献

[1] Cisco, “The Zettabyte Era—Trends and Analysis - Cisco:”, http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html, (参照 2017-01-09).

[2] M. Montagud, F. Boronat, H. Stokking, R. Van Brandenburg, “Inter-destination multimedia synchronization: Schemes, use cases and standardization”, Multimedia. Systems, vol. 18, No. 6, pp. 459-482, 2012.

[3] R. van Brandenburg, H. Stokking, O. van Deventer, “Request for comments 7272: Inter-Destination Media Synchronization Using the RTP Control Protocol”, IETF, https://tools.ietf.org/rfc/rfc7272.txt, 2014-06.

[4] F. B. Seguí, J. C. G. Cebollada, J. L. Mauri, “An RTP/RTCP based approach for multimedia group and inter-stream synchronization”, Multimedia Tools and Applications, vol. 40, No. 2, pp. 285-319, 2008.

[5] M. Montagud, “Design, development and evaluation of an adaptive and standardized RTP/RTCP-based IDMS solution”, Proceedings of the 21st ACM international conference on Multimedia (MM’13), pp. 1071-1074, 2013.

[6] H. M. Stokking, M. O. van Deventer, O. A. Niamut, F. A. Walraven と R. N. Mekuria, “IPTV inter-destination synchronization: A network-based approach”, 2010 14th International Conference on Intelligence in Next Generation Networks, pp. 1-6, 2010.

[7] K. Ok, D. Kwon, H. Ju, “A robust group synchronization approach to network congestion based on control events for media streaming”, 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS), pp. 380-383, 2015.

[8] B. Rainer, C. Timmerer, “Self-Organized Inter-Destination Multimedia Synchronization For Adaptive Media Streaming”, Proceedings of the ACM International Conference on Multimedia (MM '14), pp. 327–336, 2014.

[9] L. Keller, A. Le, B. Cici, H. Seferoglu, C. Fragouli, A. Markopoulou, “MicroCast: Cooperative Video Streaming on Smartphones”, Proceedings of the 10th international conference on Mobile systems, applications, and services (MobiSys’12), pp. 57 - 70, 2012.

[10] 大塚 雅博, 片岡 春乃, 末田 欣子, 下村 道夫, 淺谷 耕一, 水野 修, "共同体験型コミュニケーションサービスの受容性",電子情報通信学会論文誌. B, 通信 J95-B(2), pp. 366-370, 2012.

[11] R. Mekuria, P. Cesar, D. Bulterman, “Digital TV: The Effect of Delay when Watching Football”, Proceedings of the 10th European conference on Interactive tv and video - EuroiTV '12, pp. 71 - 74, 2012.

[12]ETSI EG 202 057-2, "Voice telephony (and voiceband related services like fax, data transmission and SMS)".

[13]土屋 俊貴, 新 麗, 川原崎 雅敏, “動画コンテンツの同期性計測システムの開発”, 信学技報, 115(371), pp.37-42, 2015-12-17.

[14]土屋 俊貴, 新 麗, 川原崎 雅敏, “複数地点でのライブ配信再生状況の計測と評価”, 信学技報 116(203), pp.25-30, 2016-08-29.

[15]YouTube Live, https://www.youtube.com/live, (参照 2016-08).

[16]MaxMind, Inc., GeoIP, https://www.maxmind.com, (参照 2016-08).

[17]SINET, https://www.sinet.ad.jp/, (参照 2016-08).

[18]ns-3, https://www.nsnam.org/, (参照 2017-01-09)

[19]Spirit Communications, “PalmettoNet”, https://spiritcom.com/partner/carrier?t=services, (参照 2017-01-09)