

高可用性マルチキャスト DV トランスレータの設計と実装

峯木 徹[†] 久松 剛[†] 朝枝 仁[†] 村井 純[‡]
 慶應義塾大学大学院 政策・メディア研究科[†] 慶應義塾大学 環境情報学部[‡]

概要: 近年、通信インフラのブロードバンド化により、高品質・大容量なストリーミング配信が一般化してきている。また、今後は TV 会議システムや IPTV のような時間的制約が厳格なリアルタイム性の高い映像・音声配信の需要が高まっていくと考えられる。インターネット上においてリアルタイム性の高いコンテンツ配信を多数の受信者に対して実現する場合の手法として、IP マルチキャストの利用が挙げられる。しかし IP マルチキャストでは、各受信者の環境や通信状況に応じた通信品質で対応することは困難である。本研究では、一対多型のリアルタイムストリーミング配信について、複数の「トランスレータ」と単一の「管理サーバ」の連携により、IP マルチキャストストリームをユニキャストストリームに変換し、各受信者の受信状況に応じてトランスレータや配信コンテンツの品質を変化させることで、送信者に複数の受信者が集中することに対する負荷分散を考慮した配信システムを提案する。本システムにより、一対多型のコンテンツ配信環境において、各受信者の通信状況やトランスレータの負荷状況に応じて配信を変更する高可用性なリアルタイムストリーミングが実現される。

Design and Implementation of Highly-available Multicast DV Translator

Gen Mineki[†] Tsuyoshi Hisamatsu[†] Hitoshi Asaeda[†] Jun Murai[‡]

1 はじめに

近年、通信インフラのブロードバンド化により、高品質・大容量なストリーミング配信が一般化してきている。また、今後はデータ送信から受信、そして再生までの時間的制約が厳格なリアルタイム性の高い映像・音声・動画配信の需要及び、コンテンツ配信の高品質化に対する要求が高まっていくと考えられる。具体的なアプリケーションとしては、イベントの生中継や IPTV、セミナーなどの一対多型のコンテンツ配信が想定される。しかし、一対多型コンテンツ配信の普及に対する課題として、通信品質を維持するための帯域確保、それに寄与する通信の効率化がある。

特に、一対多型通信を効率的に行う IP マルチキャストを利用する場合、光ファイバー、DSL、CATV、無線のように受信者の使用する通信インフラの多様化に対して、個別に通信品質の制御を行うことは困難である。

本研究では、一対多型のリアルタイムストリーミング配信について、複数の「トランスレータ」と単一の「管理サーバ」の連携により、IP マルチキャストストリームをユニキャストストリームに変換し、各受信者の受信状況に応じてトランスレータや配信コンテンツの品質を変化させることで、送信者に複数の受信者が集中することに対する負荷分散を考慮した配信システムを提案する。

この配信システムは受信者に対して特別な操作を要求しない。本研究の実現により、多様な環境に存在する受信者に対する一対多型のコンテンツ配信において、各受信者の通信状況に対応した可用性の高いリアルタイムストリーミング配信が可能とする。

2 既存の配信技術の課題

2.1 要素技術

本節では、一対多型のストリーミング配信で使用される要素技術に関して述べる。

2.1.1 ユニキャスト

ユニキャストを用いて送信者が多数の受信者に高品質なストリーミング配信を行う場合、受信者数倍のネットワーク帯域資源を使用する。そのため、計算機資源的にも、ネットワーク資源的にも非効率的である。

2.1.2 IP マルチキャスト

IP マルチキャストを用いて一対多型のストリーミング配信を行う場合、通信経路上にあるルータが自動的にデータを複製して送信する。

IP マルチキャスト通信を実現するためにはルータの IP マルチキャストへの対応、マルチキャストアドレスの管理、ドメイン間経路制御などの制約に加え、品質制御の観点からは、全ての受信者をグループメンバとして管理するため、個々の受信状況を把握することが困難であり、個々のコンテンツの品質制御を行うことができないという問題がある。

2.1.3 アプリケーションレイヤーマルチキャスト

アプリケーションレイヤーマルチキャスト (ALM)[1] では、マルチキャストグループに属するエンドノード同士で論理的なリンクを構成し、それぞれのノードがユニキャスト通信を行う。

アプリケーションレイヤーマルチキャストには以下の問題点がある。

- 遅延の増加

ALM ではノードをリレーすることによって配信を行う。各ノードの受信している環境は異なるため、配信経路によってはネットワーク帯域資源や計算機資源

の低いインフラの使用により遅延が増加する可能性がある。

- 不安定なノードの存在
 配信経路上のノードは常に安定してネットワークに存在しているとは限らない。そのため、上流のノードが離脱した際、配信網を再構築する必要が生じ、配信網自体が不安定になる可能性がある。

2.2 関連研究

本節では、既存のアプリケーションや過去に提案された枠組みから本研究に関連の深いものについて述べる。

2.2.1 Dvrelay

既存の各受信者のネットワーク帯域資源に応じたストリーミングを実現するシステムとして Dvrelay [2] がある。Dvrelay の概要図を図 1 に示す。

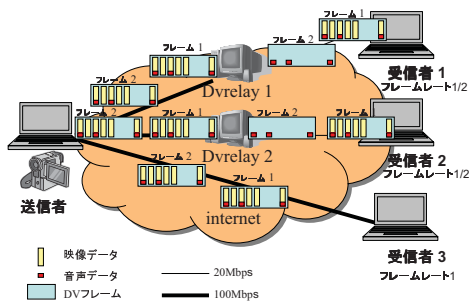


図 1: Dvrelay の概要

送信者と受信者との間に DV データの転送を行う中間ノードを設置し、下流の受信者の使用可能なネットワーク帯域に応じてフレームレートの変更を行う。しかし、この手法では複数の受信者への配信を行うことを想定していないため、受信者数と同数の Dvrelay を使用する必要がある。よって、複数受信者への配信を行うにはスケラビリティが不十分である。また、Dvrelay ではフレームレートの変更を手動で行うため、動的に変化するネットワーク帯域に応じて品質を変更する可用性の高い配信を行うことは困難である。

2.2.2 DVTS スプリッタ

DVTS スプリッタ [3] の概要図を図 2 に示す。

DVTS スプリッタは、送信者と受信者の間にトランスレータを設置する。トランスレータは下流の受信者数に応じてストリームを複製し、各受信者の使用するネットワーク帯域に応じて静的なフレームレートの変更を行う。しかし、DVTS スプリッタではトランスレータは手動で送信先アドレスを入力する必要があるため、配信先の動的な追加と削除を行うことができない。また、フレームレートの変更を手動で行うため、動的に変化するネットワーク帯域に応じて品質を変更する可用性の高い配信を行うことはできない。

2.2.3 階層化マルチキャスト

階層化マルチキャスト [4] は、IP マルチキャストにおける送信者によるストリーミングの送信ビットレート制御ができない問題に対し、階層符号化を用いることで受信者によるビットレート制御を可能にするための技術である。

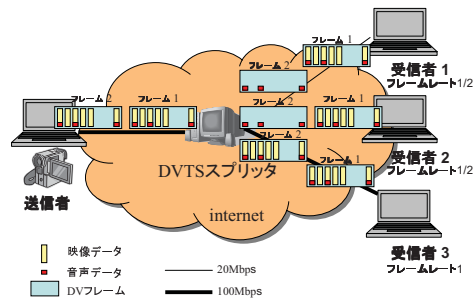


図 2: DVTS スプリットの概要

階層符号化 [5] とは、解像度等を変化させた画像を符号化することで、複数にレイヤリングする。下位レイヤーを補完する形で上位レイヤーが存在する。画像は最低限の品質を提供する基本レイヤーと、それに追加することでより空間的解像度や時間的解像度の高い高品質な画像を提供する拡張レイヤーに分類される。拡張レイヤー数の増加に従って、映像品質が向上する。

階層化マルチキャストでは、送信者は階層における最大品質でコンテンツを送信する。送信の際、レイヤー毎にマルチキャストチャンネルを割り当て、マルチキャスト送信を行う。受信者は基本レイヤーを配信するチャンネルに参加し、計算機、ネットワーク環境に応じて拡張レイヤーを配信するチャンネルに参加し、受信を行う。これにより受信者の通信品質に適応したストリーミング配信環境が構築される。

階層化マルチキャストでは、受信者環境に合わせた柔軟なサービス展開ができる一方で、オリジナルコンテンツを階層符号化しなければならないという問題がある。そのためコンテンツ送信者の負荷が大きい。また、途中経路で配信データの欠損が生じた場合、正常なストリーミングが届かない受信者が発生する可能性がある。

3 設計

本章では、目的とするコンテンツ配信モデルに対する要件を述べ、次に要件に基づいてコンテンツ配信モデルの提案を行う。

3.1 要件

- コンテンツ品質状態の検知と処理
 コンテンツ配信システムは、配信データの欠損を検知し、速やかに処理しなくてはならない。このため、受信者のコンテンツ品質を統一的に監視し、対処する機能が必要である。
- IP マルチキャストとユニキャストを併用した品質制御
 一対多型の配信技術として、IP マルチキャストを使用が効果的である。一方、各受信者に対する品質制御を行う機能が必要となる。
- 負荷分散
 不特定多数の受信者に対して、ユニキャストを行う場合、計算機資源的にもネットワーク帯域資源的にも資源を圧迫する可能性がある。そのため、送信サーバにかかる負荷を考分散する機能が必要となる。

3.2 システム概要

本節では、管理サーバ、及びトランスレータを用いたコンテンツ配信システムモデルを提案する。本提案システムのモデルを図3に示す。

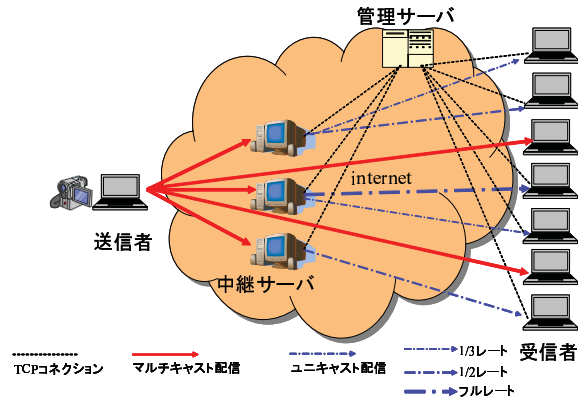


図 3: システム概要図

本システムでは、送信者から配信された IP マルチキャストストリームをユニキャストストリームに変換するためにトランスレータを送受信者間に設置する。変換したユニキャストストリームの受信者への配信は管理サーバの指示に従って行う。また、送信者に複数の受信者が集中することに対する負荷を分散させるため、複数のトランスレータを設置する。コンテンツ品質状態を統一的に監視、対処するために、単一の管理サーバを設置する。管理サーバはトランスレータ、及び受信者の双方向通信を行うことで、受信者に配信するトランスレータの選択や必要に応じて品質制御を行う。

3.3 動作概要

本節では、本システムの動作概要を述べる。図4, 5, 6に動作の流れを示す。

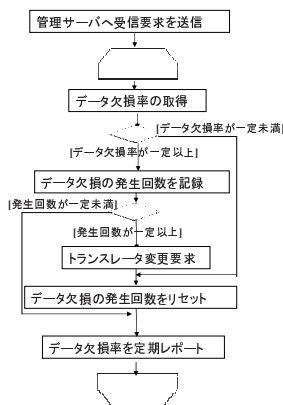


図 4: 受信者の動作概要

3.3.1 トランスレータの選択と変更

管理サーバがトランスレータを選択するために、管理サーバはトランスレータのリストを作成する必要がある。そ

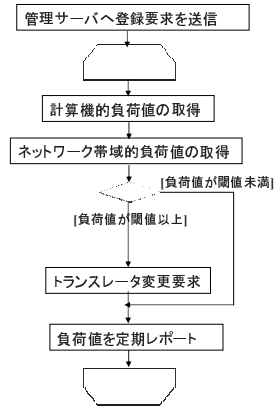


図 5: トランスレータの動作概要

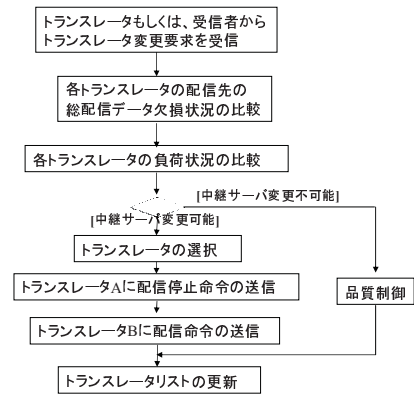


図 6: トランスレータ変更要求を受信した管理サーバの動作概要

こで、トランスレータは始めに管理サーバへ登録要求を行う。管理サーバはトランスレータを登録し、登録されたトランスレータのリストを作成する。管理サーバは受信者から受信要求があった場合、リストから配信を行うトランスレータを選択し、選択したトランスレータへ配信命令を送信する。トランスレータは配信命令に従って、受信者への配信を開始する。

3.3.2 受信者からのコンテンツ受信状態の定期通知と変更要求

管理サーバが受信者は受信しているコンテンツ品質の状態を監視するために、受信者は管理サーバへ受信コンテンツの状態を定期的に送信する。本論文ではこれを以降レシーバレポートとする。配信データの欠損が生じた場合は、一時的なものか、継続的なものかを判断するため、数回のレシーバレポートの後に配信データの欠損が解決しない場合、受信者はトランスレータ A の変更要求を管理サーバへ送信する。管理サーバはリストの中からより負荷状況に余裕があり、配信データの欠損が少ない他のトランスレータ B を選択する。トランスレータ A, B は管理サーバからの指示に従い、配信元トランスレータの変更を行う。

また、変更可能なトランスレータが存在しない場合は、

品質制御で対応する。

3.3.3 トランスレータからの負荷状態の定期通知と変更要求

負荷分散を考慮するために、トランスレータにかかる負荷を調整する必要がある。トランスレータは自身にかかる計算機負荷とネットワーク帯域負荷の負荷状況を管理サーバに対し定期的に送信する。本論文では、これを以降定期レポートとする。負荷が増大した場合は、速やかに配信している受信者を他のトランスレータに移行するために、トランスレータの変更要求を管理サーバへ送信する。管理サーバは受信者からの変更要求と同様の動作を行うことで対応する。

4 実装

本章では第3章で述べた設計に基づいて提案手法の実装を行う。

4.1 実装環境

本システムの実装を行ったソフトウェア環境は表1の通りである。

OS	Debian Linux 2.6.18
プログラム言語	C 言語
コンパイラ	gcc-4.1.2
ライブラリ	libc-2.3.6
アプリケーション	DVTS1.0f

表 1: ソフトウェア環境

4.2 DVTS (Digital Video Transport System)

本提案方式では、Digital Video Transport System (DVTS)[6] を映像配信システムとして使用する。DVTS は DV 機器から IEEE1394 インタフェースを用いて DV フォーマットのデータを読みだし、そのデータに対し IP パケット化を行い、リアルタイム伝送を行うシステムである。DVTS は NTSC 映像方式の場合で片方向約 30Mbps の帯域を利用して 29.97fps のリアルタイム映像の送受信を行う。通信方式としては UDP [7] を用いる。

DV フォーマットはフレーム内圧縮方式を用いるストリームで、フレーム間圧縮方式を用いるストリームより、品質制御を行う際の計算機資源に対する負荷も少ないことから、レートコントロールが容易であるという特徴をもつ。

パケット到達順序を把握する方法として RTP/RTCP [8] を用いている。DVTS では、RTP のシーケンス番号からパケットの順番を把握し、RTCP によりフレームレートの制御を行っている。このとき、フレームレートの制御は、フレーム単位で映像データのみを間引くことで実現している。

4.3 実装概要

4.3.1 トランスレータ変更のトリガ

- 受信者のパケットロス値

定期的な受信者から送信されるレシーバレポートのパケットロス率は経験則から 10% 以上の場合を対象とする。このとき、パケットロスを一時的なものか継続的なものかの判断するため、一定回数以上検知したとき、受信者より変更要求メッセージを受信し、変更の対象とする。
- トランスレータの負荷

本システムの実装は 100Mbit/s の NIC を対象とし

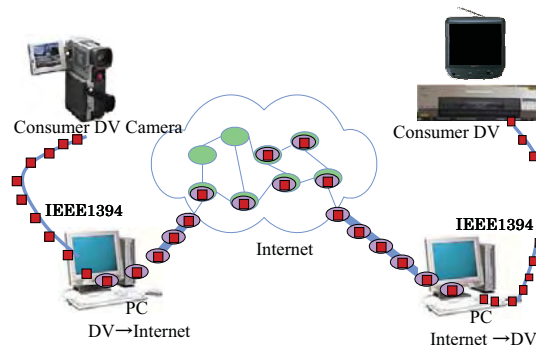


図 7: DVTS 概要

ている。第 4.2 節のように DVTS では 1fps の場合、使用ネットワーク帯域は約 1Mbps に相当する。そのため、配信総 fps が 100fps を上回る場合、ネットワーク使用帯域の限界を上回ると考え、トランスレータの変更の対象とする。

また、計算機資源の負荷の値として LoadAverage を用いる。Load Average の値として Linux Kernel が用いる proc file system を利用する。このとき、LoadAverage の値が経験則から 6 以上の場合、トランスレータから変更要求メッセージを受信し、変更の対象とする。

● エラー処理

トランスレータの異常停止を検知した場合、停止したトランスレータの配信していた受信者の IP アドレスを格納したデータベースを参照する。この時、映像・音声配信の途切れを最小限にすることが重要であるため、生存しているトランスレータの負荷状況に関係なく、受信者をランダムに各トランスレータに割り振る。その後、トランスレータの負荷が均一になるようデータベースを参照し配信先の変更を行う。

4.3.2 トランスレータ選択手法

トランスレータの変更要求の条件が満たされた場合、変更先のトランスレータの選択を行う。トランスレータ、あるいはネットワークの負荷が高い場合、配信コンテンツ品質に対して影響がでると考えられる。配信データの欠損を解消するために、トランスレータの選択の手法が必要である。本システムでは、以下の 3 つの全てを満たすことをトランスレータの選択のトリガとする。

● トランスレータ側での比較

管理サーバはトランスレータの配信先の総パケットロス値、配信総 fps、LoadAverage の 3 つを定期レポートの値を元に比較し、最も余裕のあるトランスレータを選択する。

● 受信者側の比較

変更先のトランスレータから過去に配信を受けていた場合、配信を受けていた時のパケットロス値を記録しておく。そして、記録した過去のパケットロス値と現状のパケットロス値を比較し、現状のパケットロス値が過去のパケットロス値よりも大きい場合はトランスレータの変更を行う。

● 双方の比較

受信者の記録した過去のパケットロス値と変更先のパケットロス値の受信者単位の平均値の比較を行い、過去のパケットロス値が変更先のパケットロス値の平均値よりも小さい場合はトランスレータの変更を行う。

以上の項目が満たされなかった場合、全てのトランスレータ、あるいはネットワークが高負荷であると判断し、配信ストリーミングの4.3.3品質制御によって対応を行う。

4.3.3 品質制御手法

トランスレータの変更条件を満たせず、受信者で頻繁にパケットロスが発生している場合、管理サーバはトランスレータにフレームレートコントロールで配信レートを制御させることでパケットロスの発生を抑える。また、一定回数以上のレシーバレポートでパケットロス値が10%未満であった場合、配信レートを上げ、パケットロスの変化を監視する。パケットロスが再び10%以上発生する場合、配信レートを戻す。このとき、配信レートを上げるタイミングとしては、配信レートを戻した回数に応じて2のべき数倍していく。

5 評価

本研究の有効性を確かめるため、4つのトポロジを構築して実験を行った。トランスレータの評価環境を表2に示す。本実験では送信者1ノード、トランスレータ3ノード、受信者7ノード、dummysnet [9]を用いた実機での動作検証を行った。

	トランスレータ A	トランスレータ B	トランスレータ C
CPU	AMD Athlon(tm) XP 2200+ 1.80GHz	Intel Pentium4 3.00GHz	Intel Pentium4 2.80GHz
メモリ	256KB	512KB	512KB
HDD	189.92GB	114.50GB	189.92GB
NIC	100BASE-TX	100BASE-TX	100BASE-TX
OS	Debian Linux 2.6.18	Debian Linux 2.6.18	Debian Linux 2.6.18

表 2: ハードウェア環境

5.1 実験 1: 狭帯域にいる受信者への配信

本実験のトポロジを図8に示す。受信者 A トランスレータ間に dummysnet を設置し、パケットを制限することで擬似的に狭帯域環境を再現する。結果を図9に示す。

本実験の目的はパケットロスが10%以上でトランスレータが変更されることを確認すること。また、変更後にパケットロス率の変化を見て品質制御が行われることを確認すること。以上の2点である。

dummysnet により受信者 A では30%強のパケットロスが発生している。5回のレシーバレポートの後トランスレータを変更するが、パケットロスに変化がない。そのため、他のトランスレータに変更せず、品質制御を行い、パケットロスを抑えることができた。

5.2 実験 2: トラフィックが競合している受信者への配信

本実験のトポロジを図10に示す。受信者全てが途中経路を共有することでトラフィックの競合を発生させ、その状況での動作を確認する。結果を図11に示す。

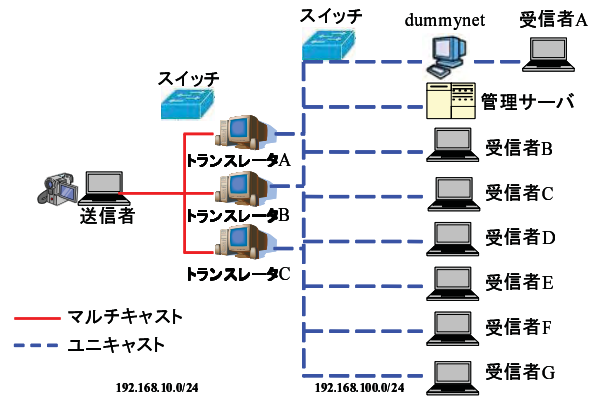


図 8: 実験 1 のトポロジ

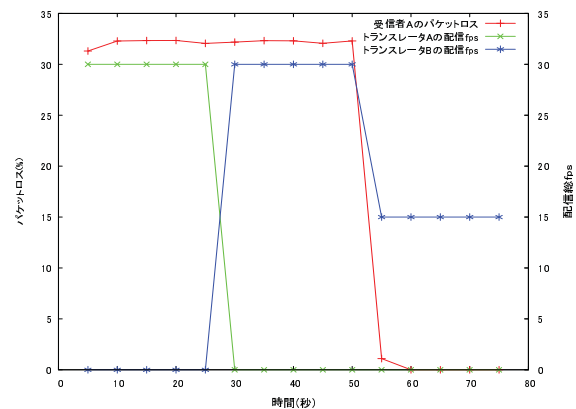


図 9: 受信者 A のパケットロス率とトランスレータの配信 fps の変化

本実験の目的は途中経路を共有することで、受信者の配信品質が他の受信者に影響を与える状況での品質制御を確認することである。

全ての受信者への途中経路を共有することでトラフィックの競合が発生し、全体的に50%弱のパケットロスが発生している。受信者の5回のレシーバレポートの後、全ての受信者がトランスレータの変更を行った。次の5回のレシーバレポートの後、レポートタイミングが一番遅い受信者 B を除き、全ての受信者の配信レートが1/2となり、最終的に受信者も配信レートを下げることでトラフィックの競合を解消する。

5.3 実験 3: 負荷が高いトランスレータからの配信

本実験のトポロジを図12に示す。ネットワーク帯域資源負荷と計算機資源負荷の2種類の負荷をかけて動作の検証を行う。この時、ネットワーク帯域資源負荷では受信者7台全てをトランスレータ A に接続する。計算機資源負荷では本システムとは別にプログラムを起動し、計算機資源に対し負荷をかけ、その動作を確認する。結果を図13, 14に示す。

ネットワーク帯域資源負荷

本実験の目的は配信 fps が100以上でトランスレータの変更を確認することである。

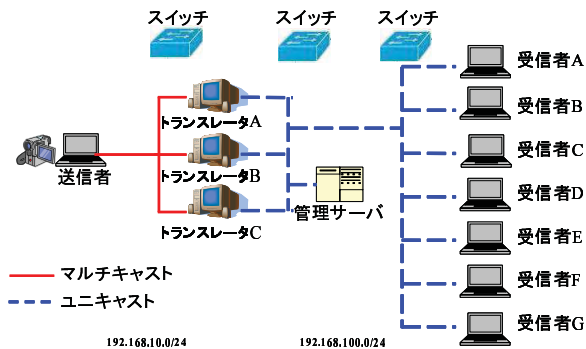


図 10: 実験 2 のトポロジ

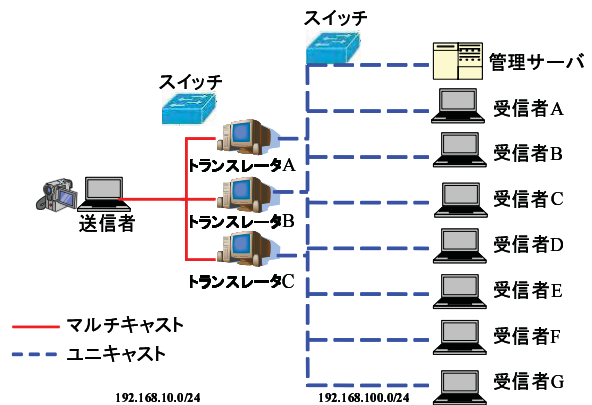


図 12: 実験 3 のトポロジ

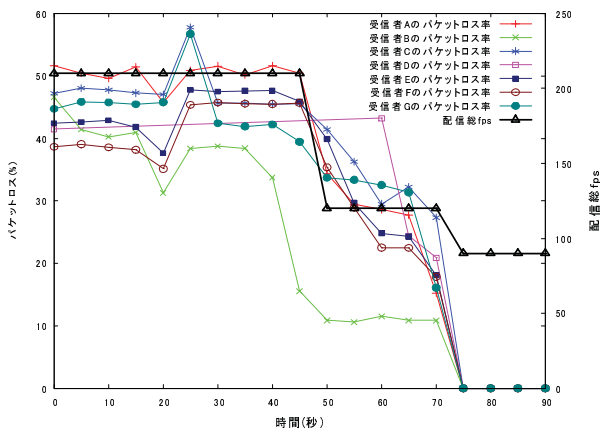


図 11: 受信者のパケットロス率と配信 fps の変化

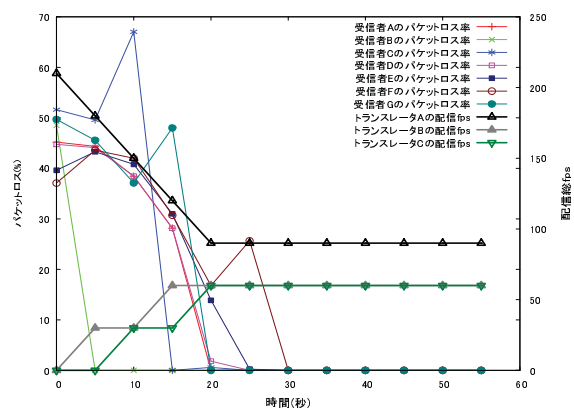


図 13: 各受信者のパケットロス率と各トランスレータの配信 fps の変化

トランスレータの配信可能なネットワーク帯域を上回るため、トランスレータの変更を行う。これにより、トランスレータの配信可能 fps である 100 を下回ったため、パケットロスを抑えることができた。この時、トランスレータを変更した瞬間にパケットロスが増加する傾向がある。これは、トランスレータを変更する際に配信を一時的に停止させるために RTP のシーケンス番号に差異が生じるためだと考えられる。

計算機資源負荷

本実験の目的は LoadAverage が 6 以上でトランスレータが切り替わるのを確認することである。

トランスレータに本システムとは別にプログラムを動作させることで計算機資源負荷を発生させる。LoadAverage が 6 に近づくにつれ、パケットロスが発生する。トランスレータ変更の閾値である LoadAverage が 6 の値を超えると受信者 B, C はトランスレータの変更を行い、パケットロスを抑えることができた。しかし、受信者 A は他のトランスレータに新たに配信を行う余剰資源がないため、トランスレータを変更をしない。その後も LoadAverage を上げていくが発生するパケットロスは全て閾値未満のものとなり、品質制御は行われない。これは、トランスレータと受信者が 1 対 1 で配信を行う場合、LoadAverage は無視できるためと考えられる。これにより、パケットロスを抑えた配信が可能であることが確認できた。

6 まとめ

本研究では、一対多型のリアルタイムストリーミング配信について、複数の「トランスレータ」と単一の「管理サーバ」の連携により、IP マルチキャストストリームをユニキャストストリームに変換し、各受信者の受信状況に応じてトランスレータや配信コンテンツの品質を変化させることで、送信者に複数の受信者が集中することに対する負荷分散を考慮した配信システムを提案した。また、本システムの評価環境として、DVTS を用いた実装を行い、その動作について評価を行った。実験より、各受信者の受信状況に応じた配信が可能であること、IP マルチキャストでは困難な受信者の環境に合わせた配信が可能であることを示した。

本研究の実現により、多様な環境に存在する受信者に対する一対多型のコンテンツ配信において、各受信者の通信状況に対応した可用性の高いリアルタイムストリーミング配信が可能となった。

7 今後の展望

本研究における今後の展望として以下の事項が挙げられる。

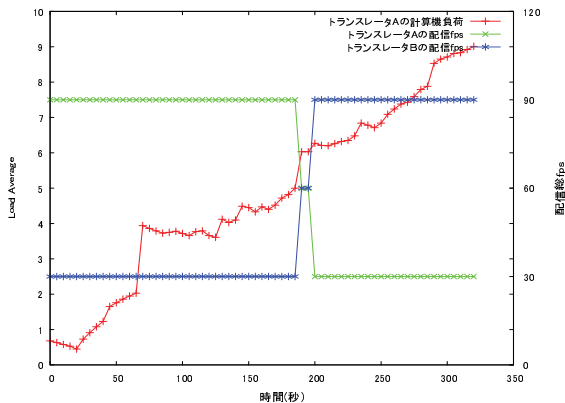


図 14: 各受信者のパケットロス率とトランスレータの Load Average の変化

- トランスレータ選択手法

本システムでは、各トランスレータと各受信者はそれぞれ同一ネットワーク上での実装を行った。しかし、実際は共に異なるネットワーク上に存在する可能性がある。そのため、トランスレータ選択手法において、RTTなどを考慮したトランスレータ選択手法が必要である。

- レポートタイミングの調整

本システムでは管理サーバに機能が集中している。そのため、管理サーバに対する負荷がボトルネックとなる可能性がある。管理サーバにかかる負荷の1つとしてレポートのタイミングが考えられる。

本システムにおいて、レポートのタイミングは全て一定である。しかし、トランスレータ、受信者が増加した場合、全てのレポートが同時に管理サーバに送信され、管理サーバへの負荷がバースティな物となり、管理サーバの処理能力を超える可能性がある。よって、受信者の状況に合わせた動的にレポートタイミングを調整し、管理サーバに対する負荷を均一にするシステムが必要である。

- 管理サーバ機能の分散

トランスレータ、受信者数がより大規模になった場合、管理サーバを複数台用い、分散配置する必要がある。このとき、管理サーバ同士を連携させ、対規模性を高める必要がある。

また、各受信者はストリーミングを受信するに当たって、管理サーバを発見、選択する手法を考える必要がある。

- 複数コンテンツへの対応

本システムではマルチキャストの配信に対して、一つのストリーミングにのみ対応している。しかし、実際は複数のストリーミング配信が存在する可能性がある。そのため、複数のコンテンツ配信に対応する機能が必要である。

8 謝辞

本研究は、情報通信研究機構 (NICT) の委託研究「ダイナミックネットワーク技術の研究開発」の助成を受けて実施したものである

参考文献

- [1] J.Liu, SG.Rao, B.Li, H.Zhang. Opportunities and challenges of peer-to-peer internet video broadcast. *Proceedings of the IEEE*, 96:11, 2008.
- [2] Y Tsuchimoto, M Abdul Awal, P Saengudomlert, T Sanguank-otchakorn, and K Kanchanasut. Bandwidth Adjustable DVTS on the Heterogeneous Internet Environments for Distance Learning. *Proc. IEEE/IPSJ SAINT workshop*, 2007.
- [3] 峯木 徹, 松園 和久, 朝枝 仁, 中村 修. レート変更を行う DVTS スプリッタの設計と実装, 2008. 情報処理学会 DPS 研究会.
- [4] S.McCanne, V.Jacobson and M.Vetterli. Receiver-driven Layered Multicast. *Proc. ACM SIGCOMM*, Aug 1996.
- [5] S.Sakazawa, Y.Takishima, M.Wada and K.Amano. Bandwidth-Adaptive VideoTransmission Method for Heterogeneous Network Environment. *Proc. International Workshop on New Video Media Technology*, pages 49–54, 1997.
- [6] A.Ogawa. *DVTS (Digital Video Transport System) WWW page*, Nov 2001. <http://www.sfc.wide.ad.jp/DVTS/>.
- [7] J. Postel. *User Datagram Protocol*, August 1980. RFC 768.
- [8] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. *A Transport Protocol for Real-Time Applications*, July 2003. RFC 3550.
- [9] L. Rizzo. *dummysnet WWW page*, Dec 2008. <http://info.iet.unipi.it/luigi/ipdummysnet/>.