

## スループット監視に基づく複製サーバ選択方法のスケラビリティ向上

廣森聡仁<sup>1</sup> 山口弘純<sup>1</sup> 安本慶一<sup>2</sup> 東野輝夫<sup>1</sup> 谷口健一<sup>1</sup>

<sup>1</sup> 大阪大学大学院基礎工学研究科情報数理系専攻 <sup>2</sup> 滋賀大学経済学部情報管理学科

{hiromori,h-yamagu,higashino,taniguchi}@ics.es.osaka-u.ac.jp  
yasumoto@biwako.shiga-u.ac.jp

我々は、IPマルチキャストネットワークにおいて、複数のサーバがそれぞれ同一動画を複数の異なる品質で符号化し、個別のマルチキャストストリームとして配送する状況で、各受信者が自身で収集可能な情報だけで自律的にサーバ及び品質レベル（ストリーム）を決定する選択方法を提案してきている。この方式においては、受信者は全ストリームについて、その配送経路に沿って監視パケットを送信し情報収集を行うため、多数の受信者が存在する環境においてはこのトラフィックの増大が問題となる。本稿では、個々のストリームについてある一定数の受信者のみが監視パケットを送信することで、受信者数が増えた場合でも監視パケットによるトラフィックを低く抑える方法を提案する。監視パケットの数を抑えるとそれだけ得られる情報が少なくなるが、提案方式では各受信者が取得した情報を全ての受信者で共有することで、ネットワークシミュレータを用いた実験結果から、ネットワーク上に流れる監視パケットの数を抑えながら、共有した監視情報を用いて適切にストリーム選択が行えることがわかった。

## Improving Scalability in Monitoring-based Multicast Server Selection

Akihito Hiromori<sup>1</sup>, Hirozumi Yamaguchi<sup>1</sup>, Keiichi Yasumoto<sup>2</sup>,  
Teruo Higashino<sup>1</sup> and Kenichi Taniguchi<sup>1</sup>

<sup>1</sup> Graduate School of Engineering Science, Osaka University <sup>2</sup> Faculty of Economics, Shiga University

In our previous work, we have proposed a selection technique for replicated multicast video streams from different servers. In the technique, we have assumed that there exist replicated video servers, each of which encodes the same video into the ones of different quality levels and continuously transmits them as different multicast streams. Using an IGMP facility like *mtrace*, each receiver monitors the packet count information of all those streams on routers and periodically selects the one which is expected to provide low loss rate and to be suitable in the current available bandwidth of the receiver. However, under the existense of a large number of receivers, the amount of tracing traffic may cause a significant problem. In this paper, we improve the monitoring technique used in the selection technique to keep the amount of tracing traffic low even if the number of receiver is large. In order to reduce the amount of tracing traffic, only some part of receivers send *mtrace* queries for each stream. Also, each receiver selects a stream using the shared infoaion which is obtained by the other receivers. Our experimental results using a network simulator have shown that the amount of tracing traffic is reasonable and receivers could select appropriate streams.

### 1 概要

近年、ネットワーク基盤の整備発展に伴い、近い将来、ライブ中継やビデオオンデマンドなど、多くの動画コンテンツがネットワーク上で配送されることが予想される。しかし、一般に動画コンテンツは比較的大容量であるため、動画配信サーバに多数のアクセスが集中するような場合にはネットワーク帯域を圧迫する。放送型メディア配送では帯域を浪費しないマルチキャスト通信 [1] が有用であるが、単一のサーバで動画の配送では、安定して動画を受信できないことも考えられる。これに対し、最近では複製マルチキャストサーバを分散配置したもとの受信者が適切なサーバを選択する方式について、いくつか研究が始められている [2, 3]。

我々はこれまでに、IP ネットワークにおける複製マルチキャスト動画サーバの選択方法を提案してきている [3]。この方式では、複数のマルチキャストサーバが同じ動画コンテンツを異なる複数の品質レベルに符号化し、独立した別々のマルチキャストストリームとして配送する。そのもとの、各受信者はマルチキャストルータ上でのストリームのパケット数情報（転送されたパケットの数）を *mtrace* というツールを用いて定期的に監視し、その情報をもとに適切な品質が得られるストリームを動的に選択する。

[3] の方式では、各受信者が自律的にストリームを選択できるよう、各受信者は全てのストリームに対し *mtrace* を実行する。このため、受信者数が増加するとサーバ側のマルチキャストルータに大量の *mtrace* クエリーメッセージ (*mtrace* による監視メッセージ) が集中する恐れがある。ストリーム選択にはパケット数情報が有効であることは [3] で示されており、*mtrace* の実行を単純に抑制すると適切なストリーム選択が行えない可能性があるが、大規模なネットワーク環境の適応性 (スケラビリティ) を向上させるため、*mtrace* に伴う監視トラフィックを軽減させる必要がある。

本稿では、文献 [3] の方式において、監視トラフィックを軽減しながら、かつ適切にストリームが行われるよう各受信者の *mtrace* の実行対象、及び実行頻度についての検討を行う。提案方式では、各受信者の *mtrace* 実行対象は受信しているストリームのみとし、受信者は *mtrace* 実行結果を全受信者にマルチキャスト送信することによって、各受信者が全ストリームについてのパケット数情報を保持できるようにする。これにより、ストリーム選択に必要なパケット数情報を取得しつつ監視トラフィックを軽減することができる。にも関わらず、受信者数がさらなる増加により、サーバ側に近いマルチキャストルータにおいて、*mtrace* クエリーメッセージの数が増加することも考えられる。従って、個々

のストリームに対して、*mtrace* を実行できる受信者のうち、監視する時間間隔毎、その時間実際に *mtrace* を実行する受信者数をある一定数以下に制限されるようにする。このように各受信者の *mtrace* 実行頻度を低下させることで、多くの受信者が存在するような環境においても、監視トラフィックを低く抑えることができる。

我々はネットワークシミュレーター *ns*[4] を用いて、文献 [3] の方式と提案方式を比較するシミュレーションを行った。実験結果から、[3] の方式と比較して *mtrace* によるトラフィックが十分低いものとなっており、ノード数 100 の時にはサーバ側のマルチキャストルータ上では平均 2 Kbps と実用問題ない値となっていた。また、各受信者が実際に受信できたストリームの品質は以前の方式とほぼ同等の値となっており、ストリームの選択が適切に行われていることが確認できた。

## 2 複製サーバ選択アルゴリズム

提案方式では、各サーバは同一の動画コンテンツを  $L$  の異なる品質レベル（画像サイズ、解像度、フレーム数など）で符号する。以下ではこの品質レベルを自然数で表し、1 が最低品質、 $L$  は最高品質を表す。また、サーバ  $S_i$  から配送される品質レベル  $l$  の動画データストリームを単にストリームとよび  $st_{i,l}$  で表す。各ルータは文献 [5] で示されている監視機能を実装しているものとする。この機能を使った IP マルチキャストトラフィック監視用の代表的なツールに *mtrace*[5] がある。*mtrace* はマルチキャストグループアドレスと送信ホストアドレスを指定することで、送信ホストから *mtrace* を実行するユーザホストまでのマルチキャスト配送経路及びその経路上の各マルチキャストルータで転送された入出力パケット数を取得することができる。各受信者は一秒あたりにルータで転送されるパケット数を得るために、*mtrace* クエリーを定期的を送る。クエリーの結果は各ルータでのパケット数とクエリーの到着時刻となっており、最新のクエリー結果 2 つから各ルータで一秒あたりに転送されたパケット数を計算することができる。

### 2.1 受信者が選択アルゴリズムで利用する情報

受信者  $R_j$  は以下の情報を前もって知っている、または監視によって得られるものと仮定する。

- $bw_{i,l}$ ; ストリーム  $st_{i,l}$  の送信レート。各ストリームはあらかじめわかっている固定レートで配送されるものとする。
- $path_{i,j}$ ; サーバ  $S_i$  から受信者  $R_j$  までの配送経路。*mtrace* クエリーの結果から得ることができる。
- $gr_{(i,l),j}$ ; 受信者  $R_j$  に対するストリーム  $st_{i,l}$  のグラフテイングルータ。グラフテイングルータは、配送経路  $path_{i,j}$  上のルータのうちストリーム  $st_{i,l}$  を受信しており、受信者  $R_j$  に最も近いルータを表す。*mtrace* クエリーの結果から配送経路  $path_{i,j}$  上でストリーム  $st_{i,l}$  のパケット数が 0 でない最後のルータとして発見できる。
- $ratio_{(i,l)}@r$ ; ストリーム  $st_{i,l}$  のルータ  $r$  でのパケット到着率。以下  $count_{(i,l)}@r$  でルータ  $r$  で一秒あたりに転送されたパケット数を表す。これ

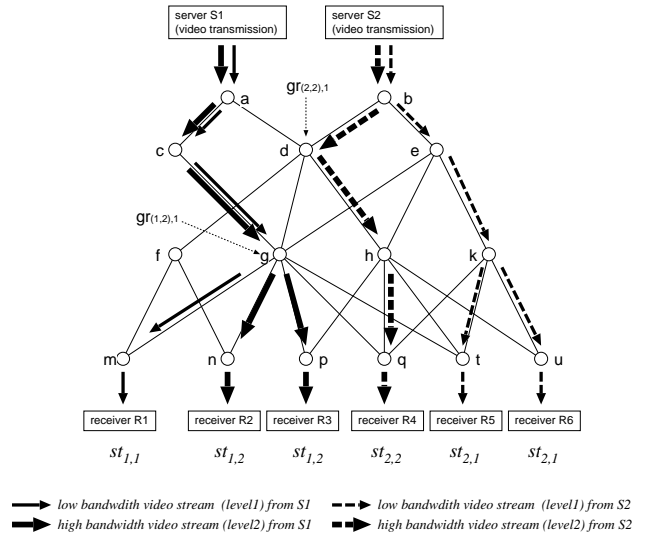


図 1: ストリーム選択の例

は *mtrace* クエリーを定期的に送ることで取得できる。これを用いて、 $ratio_{(i,l)}@r$  は  $\frac{count_{(i,l)}@r}{count_{(i,l)}@S_i}$  として計算できる。なお、 $count_{(i,l)}@S_i$  はサーバ  $S_i$  でストリーム  $st_{i,l}$  のパケットが一秒あたりどれだけ送信されたか示すもので、一般的にはこのパケット数を計ることはできない。しかし、 $S_i$  に属する同じ LAN のルータで破棄されるパケットはほとんどないとすれば、この値はこのルータで転送された数とほぼ同じだと考えられる。一般的に、輻輳がそのようなルータで起こることはまれであり、この仮定は適当なものだと考えられる。

- $ratio_{(i,l)}@R_j$ ; 受信者  $R_j$  でのストリーム  $st_{i,l}$  のパケット到着率。  $\frac{count_{(i,l)}@R_j}{count_{(i,l)}@S_i}$  として計算できる。

### 2.2 ストリーム選択アルゴリズムの基本方針

図 1 では 2 つのサーバ  $S_1, S_2$  がそれぞれ同一の動画コンテンツをレベル 1, レベル 2, レベル 3 の 3 つの品質レベル（この例ではそれぞれ低品質, 中品質, 高品質）でマルチキャスト送信しており、受信者  $R_j$  ( $j = 1..6$ ) はそれぞれあるサーバからのある品質レベルのストリームを受信している。例えば  $R_1$  はサーバ  $S_1$  からのレベル 1 のストリーム  $st_{1,1}$  を、 $R_5$  は  $st_{2,3}$  を受信している。

実際に得られるストリームの品質は受信しなければわからないが、提案方式では、グラフテイングルータでのパケット到着率、グラフテイングルータから受信者までのホップ数を受信後に得られるストリームの品質の指標とする。このもとの、各受信者は基本的に、受信者ホストでの現在受信中のストリームのパケット受信率が十分安定していると判断できる場合はより品質レベルの高いストリームを、安定していない場合はより低いストリームを動的に選択する。その際、実際の受信後のパケット到着率なるべく高いストリームを選択するため、ルータの監視結果に基づき、グラフテイングルータでのパケット到着率が基準値以上で、また受信者からグラフテイングルータまでのホップ数が最も小さいストリームを選択する。この基準値は動画の再生にほとんど問題がないと判断できるほど十分高いパケット到着率を表すものとする。このようにする

理由は、グラフテイングルータから受信者までのトラフィックは観測できないため、グラフテイングルータでのパケット到着率がそのまま受信者でのパケット到着率とならない可能性があるためである。提案方式では、グラフテイングルータから受信者までの距離が最も小さいストリームを選択することで、そのストリームを実際に受信したときに期待する品質が得られない可能性を減少させている。

例えば図1において、受信者  $R_1$  での  $st_{1,1}$  のパケット到着率が十分安定していると判断できた場合には、 $R_1$  はより高い品質で動画像を受信するために、レベル2のストリームを受信しようと試みる。まず、 $R_1$  はレベル2の全てのストリームについてグラフテイングルータ  $gr_{(1,2),1}(path_{1,1} = a-c-g-m \text{ 上の } g)$ ,  $gr_{(2,2),1}(path_{2,1} = b-d-f-m \text{ 上の } d)$  を定める。 $ratio_{(1,2)}@gr_{(1,2),1}$  と  $ratio_{(2,2)}@gr_{(2,2),1}$  (グラフテイングルータでのパケット到着率) のどちらも基準値以上となっているものとする。このとき、 $R_1$  は  $st_{1,2}$  と  $st_{2,2}$  のうち、グラフテイングルータへのホップ数が小さいストリームを選択する。この例では、 $R_1$  から  $g$  までの距離が1ホップ、 $d$  までの距離が2ホップとなっているので、 $R_1$  は  $st_{1,2}$  を選択する。

同様に、受信者でのパケット到着率が基準値を下回る場合は、品質レベルが同じかより低いストリームのうち安定して受信できると予想されるものを選択する。例えば、 $ratio_{(1,2)}@R_3$  が基準値を下回ったものとする。と、 $R_3$  は  $st_{2,2}$ ,  $st_{1,1}$ ,  $st_{2,1}$  のうちの一つを受信しようとする。どのストリームについてグラフテイングルータでのパケット到着率が基準値を超えていれば、 $R_3$  は現在の品質レベルと維持するために  $st_{2,2}$  を選択する。

### 2.3 ストリーム選択アルゴリズム

各受信者  $R_j$  は受信しているストリーム  $st_{i,l}$  のパケット到着率に応じて別のストリームを選択するか、現在受信しているストリームを継続して受信するかの判断を一定期間ごとに行う。提案方式ではパケット到着率について二つの基準値  $P_1, P_2$  を用いて、 $R_j$  の状態を (1) “品質レベル増加状態”, (2) “継続受信状態”, (3) “品質レベル維持/減少状態” の3状態に分類する。ここで、 $P_1$  はストリームの受信がほぼパケット損失を伴わずに行われており、より高い品質レベルのストリームを受信するのに十分だと判断できる値とし、 $P_2$  は多少パケットロスが発生しているものの、動画像の再生にはほとんど影響がないと判断できる値とする。

1. 受信ストリーム  $st_{i,l}$  のパケット到着率が  $P_1$  ( $0 < P_1 \leq 1$ ) 以上である受信者  $R_j$  は “レベル増加状態” に属する。この場合、 $R_j$  は、品質レベル  $l+1$  の各ストリーム  $st_{i',l+1}$  のグラフテイングルータ  $gr_{(i',l+1),j}$  におけるパケット到着率が後述する  $P_2$  以上であるストリームのうち、グラフテイングルータから  $R_j$  までのホップ数が最小であるストリームを選択する。グラフテイングルータにおいて一定以上の品質(すなわちパケット到着率)が実現されており、かつそのグラフテイングルータからの距離が短いストリームを選択する。そのようなストリームが存在しない場合には  $R_j$  は  $st_{i,l}$  を受信し続ける。
2. 受信ストリーム  $st_{i,l}$  のパケット到着率が  $P_2$  ( $0 < P_2 < P_1$ ) 以上  $P_1$  未満である受信者  $R_j$  は “継

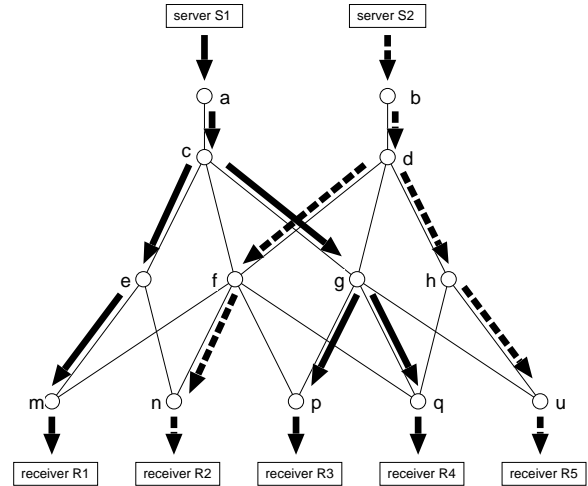


図2: 二つのマルチキャスト配送木

続受信状態”に属する。この場合、 $R_j$  は  $st_{i,l}$  を継続して受信する。

3. 受信ストリーム  $st_{i,l}$  のパケット到着率が  $P_2$  未満である受信者  $R_j$  は “品質レベル維持あるいは減少状態” に属する。この場合、 $R_j$  は品質レベル  $l$  もしくは  $l-1$  のストリームから、そのグラフテイングルータでのパケット到着率が  $P_2$  以上であるストリームのうち最も品質レベルの高いストリームを選択する。そのようなストリームが一意に決まらない場合は、 $R_j$  はグラフテイングルータから  $R_j$  までの距離が最も短いものを選択する。

## 3 スケーラビリティを考慮した監視方法

2章で示したように、提案方式は各受信者は *mtrace* を用いてパケット数情報を取得し、その情報をもとにストリームを選択する。ACK/NACKによるフィードバックを元にした高信頼性マルチキャストにおいて、ACK/NACK 爆発する問題 [6] が指摘されているが、提案方式においてもサーバに近いマルチキャストルータで *mtrace* クエリーのパケット数が増大する可能性がある。ここで *mtrace* クエリーのパケットの量を簡単に見積ってみる。 $N$  人の受信者が  $T_M$  秒ごと定期的に *mtrace* クエリーのパケット (たかだか 0.2kb) を送信する。それらのパケットは配送経路上をサーバのほうへと転送されサーバに集中する。 $L$  を品質レベルの数だとすると、サーバ近くのマルチキャストルータにおける *mtrace* のトラフィックは以下ようになる。

$$\frac{0.2(KB) * 8(bit/B) * 2N(packet) * L}{T_M(second)} \quad (bps)$$

例えば  $N = 10,000$ ,  $T_M = 2$  and  $L = 3$  とすると、*mtrace* のトラフィックは  $\frac{0.2 * 8 * 10,000 * 3}{2} = 24Mbps$  となり、配送木の根にあたるルータでは一秒あたりに 5000 のクエリーを処理することになる。この値は実用的なものではなく、多人数の受信者に対応するためにはこのクエリーの数を抑える必要がある。

### 3.1 提案する監視方式

提案方式ではこの問題に対処するために、一つのストリームに対しいくつかの受信者のみが *mtrace* クエリー

を送信している。また、*mtrace* クエリーを送信しないストリームについての情報を受信者間で共有できるようにしている。以下で提案方式の詳細を述べる。*mtrace* クエリーの結果を共有するために、全ての受信者が属しているマルチキャストグループ  $C$  をコントロールグループとして用いる。各受信者は、ストリーム選択の際に用いる、パケット数情報を保持している。この情報は全てのストリームのサーバから各受信者への配送経路についてのもので、自身が実行した *mtrace* の結果、 $C$  から得られた結果で更新される。

- 各受信者  $R_j$  はストリーム受信開始前に、全てのストリームについて配送経路を調べるために、全てのストリームについて *mtrace* クエリーを送信する。
- 各受信者  $R_j$  は現在受信しているストリーム  $st_{i,l}$  についてのみ *mtrace* クエリーを送信する権利をもつ。
- 監視期間  $T_M$  秒ごとに、コントロールグループ  $C$  に同期のためのメッセージが送信される (サーバのうちの一つによる)。
- $R_j$  は  $\lambda$  の指数分布に基づいて値が定まるランダムなタイマーをもっているものとする。受信者はそれぞれ同期メッセージを受信するとタイマーをスタートさせる。
- タイマーが切れるまでに  $R_j$  は受信しているストリーム  $st_{i,l}$  について *mtrace* クエリーの結果を得られないときには、直ちに *mtrace* クエリーメッセージを送信する。
- $R_j$  が *mtrace* クエリーの結果を得たあと、直ちにコントロールグループ  $C$  に対してそれを送信し、自身が保持するパケット数情報を更新する。
- $R_j$  は  $C$  によって得られたクエリーの結果で、ストリーム選択の際に用いる、自身が保持するパケット数情報を更新する。
- 受信者が送る *mtrace* クエリーの総数を制限するために、 $R_j$  は動的に指数分布のパラメータ  $\lambda$  を変化させる。この目的のために  $R_j$  は、 $T_M$  ごと  $C$  によって得られた *mtrace* の結果をカウントする。これを  $N_\lambda$  とする。 $N_\lambda$  は一定期間内に *mtrace* クエリーを送った受信者の数と同じである。 $R_j$  は次の式で *mtrace* による最大のトラフィック (サーバ側と受信者側) を見積もる。

サーバ側:

$$\frac{0.2(KB) * 8(bit/B) * \frac{2N_\lambda}{M} (packet)}{T_M(second)} (bps)$$

受信者側:

$$\frac{0.2(KB) * 8(bit/B) * N_\lambda(packet)}{T_M(second)} (bps)$$

$M$  をサーバの数とすると、各受信者は上記の予想トラフィックによって次の期間の  $\lambda$  の値を決定する。<sup>1</sup> 予想トラフィックのうちいずれかがコントロールトラフィックとして許される値を越えると、受信者は  $\lambda = 2\lambda$  とする。そうでなければ  $\lambda = \alpha\lambda$  ( $\alpha < 1$ ) とする。

<sup>1</sup> *session\_delay* はマルチキャストセッションにおける端末間の予想遅延とすると、 $\lambda$  は始め  $1/\lambda = 3 * session\_delay$  とする。はじめに受信者が *mtrace* による結果を得るために *mtrace* クエリーを送信するが、結果をえるまでに  $2 * session\_delay$  かかる。そして、 $C$  で配送されるが、その配送に *session\_delay* かかる。よって、 $\lambda$  をこのような値とした。この初期値をつかうと、約 63% の受信者が  $C$  で *mtrace* の結果を送信することになる。

図 2 を用いてこのアイデアを説明する。二つのサーバ  $S_1, S_2$  があり、それらはそれぞれ  $st_{1,1}, st_{2,1}$  を送信している。受信者  $R_1, R_3, R_4$  は  $st_{1,1}$  を受信しており、 $st_{1,1}$  にのみ *mtrace* クエリーを送信できる (一方、 $R_2, R_5$  は  $st_{2,1}$  のみに *mtrace* クエリーを送信できる)。  $T_M$  ごとに  $R_1, R_3, R_4$  のいくつかは *mtrace* クエリーを送信できる。ここでは  $R_3, R_4$  がこの期間にクエリーを送るものとする。そのとき彼らは配送経路上の各ルータの最新のパケット数を知ることができ ( $a-c-g-p$  と  $a-c-g-q$ )、それぞれその結果を受信した後に  $C$  に送信する。一方、 $R_1$  は  $C$  から得られた情報からその配送経路上 ( $a-c-e-m$ ) の  $a, c$  の情報は得ることができるが、 $e, m$  の情報を得ることができない。この場合には、 $R_1$  は  $e, m$  について自身が保持していた以前の情報を用いる。提案方式では指数分布に基づいて値が定まるランダムなタイマーにに従い各受信者が *mtrace* クエリーを送信しており、そのうち各受信者の配送経路上のパケット数情報が更新されることになる。ただ、サーバに近いルータの情報は受信者側のものと比べて頻繁に更新されることになる。

次に  $R_j$  が受信していないストリームの情報について考える。 $R_j$  はそのようなストリームに対し *mtrace* クエリーを送信しないので、コントロールグループ  $C$  以外からはこのような情報を得ることができず、また、配送木に含まれていないルータの情報を知ることができない。例えば、 $R_2, R_5$  は *mtrace* クエリーの結果をコントロールグループ  $C$  に送っているが、これには  $R_1$  についての配送経路  $path_{2,1}$  上の  $m$  の情報は含まれていない。しかしながら、 $R_1$  は  $b, d, f$  の情報を知ることができ、 $R_1$  はストリーム選択に必要な情報は得ることができていることになる。(コントロールグループ  $C$  から送られた *mtrace* クエリーの結果に含まれているもののうち、受信者に最も近いルータがグラフティングルータとなる。)

### 3.2 考察

最後に、さらなるスケーラビリティ向上として以下のようなことが考えられる。

- それぞれのストリームについてコントロールグループを割り当てることも考えられる。このようにすると、各受信者は受信者にとって関係のあるストリームのコントロールグループのみに入るだけでよいことになり、*mtrace* のトラフィック総量はさらに減ると考えられる。4章のシミュレーション実験では簡略化のためコントロールグループは一つとしている。
- 許容できるコントロールグループのトラフィック総量としては、RTCP [7] のレート制御にあるように、一番品質の低いストリームの送信レートの 5% から 10% が妥当であると考えられる。提案方式の実験結果ではこの制限以下であることが示された。詳しくは 4章で述べる。
- *mtrace* の結果を送る際、TTL を制御することで結果を送る受信者とは遠く離れた受信者にその結果を送ることを防ぐことができる。一般的に、地理的に離れているそのような受信者間では配送経路を共有しないと考えられ、このような制御もまた効果的である。

## 4 シミュレーション実験による性能評価

提案方式のシミュレーション実験による評価を行うため、まず受信者がマルチキャストストリームの配送経路上の到着パケット数を取得するためのネットワーク層レベルのプロトコルを IP マルチキャストのツール *mtrace* の動作原理に基づいて設計し、それをネットワークシミュレータ ns-2 [4] 上に実現した。各受信者は自身が受信しているストリームのパケット到着率に基づいてストリームの選択を行うが、選択アルゴリズム (2.3 節) で利用する閾値  $P_1, P_2$  はそれぞれ 95%, 85% とした。

各受信者が、直前の選択後、新しいストリームを選択可能となる時間は、二つのパラメータ  $T_{up}, T_{down}$  を用いて以下のように調整する。品質レベル  $l$  のストリームを受信しており、品質レベル  $l+1$  のストリームを受信しようとする場合には、前の選択から  $T_{up}$  (初期値 2 秒) だけ待たなければならない。この品質レベルアップが失敗して  $l$  のストリームに戻った場合には、その直後の再度の品質レベルアップは失敗する可能性が高い。そこで、品質レベルアップの試みをしばらくを抑制するために、 $T_{up}$  を 2 倍にする。品質レベルアップが成功した場合には  $T_{up}$  を 2 秒に初期化する。一方、現在の品質レベルかより低いレベルのストリームに移行するときは、 $T_{down}$  待つものとする。シミュレーション実験では  $T_{down}$  を 0.5 秒に固定している。このような受信レート制御方法は、階層化マルチキャストにおける階層適応方法として文献 [8] などで紹介されている。

### 4.1 ユーザ満足度評価

提案方式において受信者全体として実際にどの程度の品質を実現できるかを調べるためのシミュレーション実験を行った。実験では、ストリーム  $st_{i,l}$  を受信した時間 ( $t_a, t_b$  をそれぞれ受信開始時間と終了時間とする) とそのときのパケット到着率  $p_t$  からそのストリームを受信したときに得られた全体の品質を  $l$  と  $p_t$  の積分 (すなわち、 $\int_{t_a}^{t_b} l * p_t dt$ ) と定義し、満足度と呼ぶ。実験では、受信者全体に対し満足度総和を計測し、文献 [3] の方式と比較した。[3] の方式では、受信者は全てのストリームに対して *mtrace* を実行しており、頻繁にパケット数情報を更新している。一方、提案方式では、受信者は受信しているストリームにのみ *mtrace* を実行しており、その他のストリームについてはコントロールグループ  $C$  からパケット情報を得ている。従って、提案方式では [3] の方式に比べ、各受信者が保持するパケット数情報は頻繁に更新されておらず、そのためにストリーム選択に影響がでることが考えられるため、提案方式において適切にストリーム選択が行われているかを調べた。

この実験では、ノード数 20 から 220 のネットワークをランダムに生成し、計 35 のネットワークにおいてそれぞれ 120 秒のシミュレーションを行った。各リンクの帯域は、280kbps, 560kbps, 1.1Mbps のいずれかとした。各シミュレーションではサーバの数は 3 とし、ユーザの数は総ノード数の半分とした。各サーバは品質レベル 1, 2, 3 の 3 つのストリームを送信しこれらの送信レートはそれぞれ 256kbps, 512kbps, 1Mbps とし

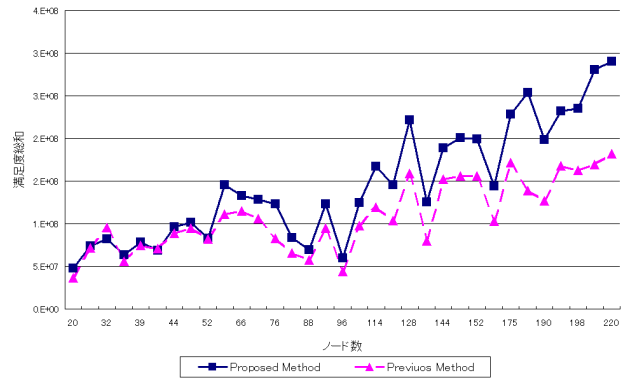


図 3: 満足度総和のノード数による変化

た。また、マルチキャストルーティングプロトコルは DVMRP とした。

図 3 では、各方法で得られた満足度の総和をノード数ごとに示しており、縦軸が満足度の総和、横軸がノード数を表す。図 3 より、いずれの方式もノード数 50 まではほぼ同じ値となっていることがわかる。ノード数が増えたと提案方式の値は [3] よりも大きい満足度総和を達成しているといえる。これは [3] の方式において、受信者の数が増加するとその分 *mtrace* のトラフィックが増加し、ストリームの配送に影響を及ぼしたためである。

また、満足度総和が時間経過とともにどのように変化するかを調べた。ノード数 50, 100, 200 での結果を図 4 に示す。ノード数 50 (図 4 (a)) の結果においては、提案方式では、実験開始から十数秒間は [3] の方式と比べ、満足度総和は劣ってはいるが、実験終了時にはより高い値となっている。ノード数 100 (図 4 (b)) の結果では、提案方式の満足度総和は実験開始時から [3] の値を上回っており、実験終了時には約 30% 上回っていた。ノード数 220 (図 4 (c)) の結果では、提案方式と [3] との差がより顕著に表れている。どのノード数においても、時間の経過に伴い受信者間で共有する情報が増加すると、ストリーム選択が適切に行われるようになることわかる。

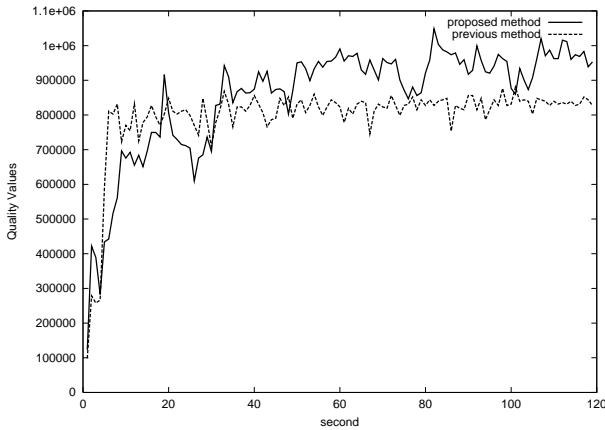
### 4.2 監視パケットによるトラフィックの測定

[3] と比較してどの程度 *mtrace* によるトラフィックが軽減できているか測定するシミュレーション実験を行った。なお、このシミュレーション実験では、監視方式で利用する  $\lambda(3.1)$  を固定とし、動的にトラフィックの量を制御していない。サーバ側での *mtrace* による監視トラフィックの各事例での平均流量を図 5 に示す。

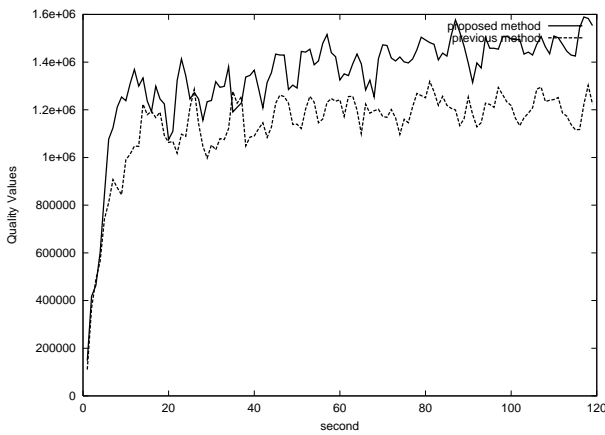
図 5 では横軸はノード数を、縦軸は *mtrace* に伴うトラフィックの平均を示している。[3] の方式と比較すると、トラフィックの量は 30 分の 1 以下となっており、かなりのトラフィックを軽減できていることがわかる。ノード数が増加すると、トラフィック量も増加しているが、その量はノード数 220 のネットワークにおいても 6 kbps と十分低く抑えられている。

## 5 あとがき

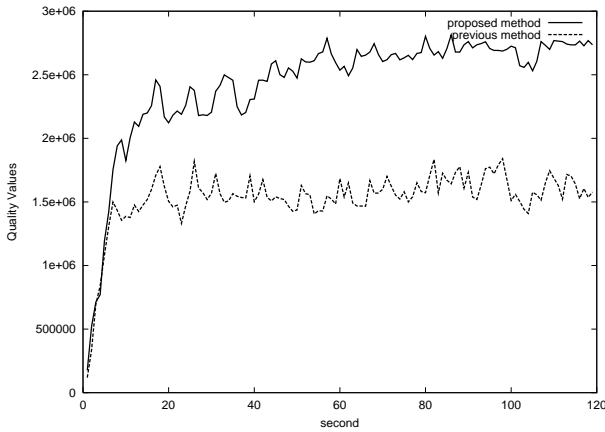
本稿では、大規模な環境におけるスケラビリティを考慮した、IP マルチキャストネットワーク上での複製サーバを用いた動画配信におけるストリーム選択方式におけるスケラビリティの向上について検討お



(a) 50 Nodes



(b) 100 Nodes



(c) 220 Nodes

図 4: 満足度総和の時間による変化

よび実証を行った。同方式においては、ある一定期間に監視メッセージを送信する受信者を一定数に抑え、監視メッセージの集中を抑えている。

一般にサーバ選択に用いる情報の最新性とスケーラビリティはトレードオフの関係にあると考えられる。本稿では、このトレードオフの追及を行っており、効率よくネットワークの情報をあつめ、かつ受信者の高い満足度を達成できているといえる。今後の研究としてはアプリケーションへの適応や、MBONE など実際のネットワークトポロジーでのシミュレーション実験を考えている。

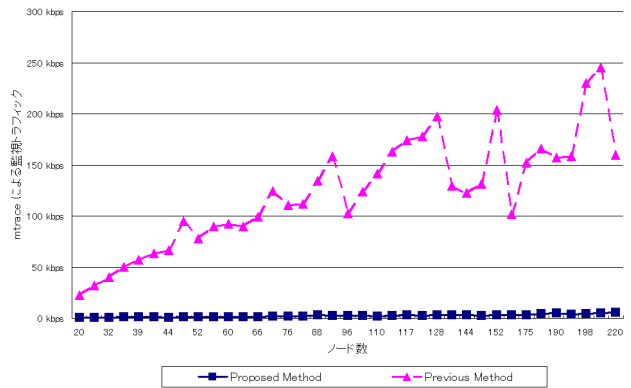


図 5: mtrace によるトラフィック

## 参考文献

- [1] C. Diot, J. Crowcroft, and W. Dabbous. Multi-point communication: A survey of protocols, functions, and mechanisms. *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 3, pp. 277–290, 1997.
- [2] Z. Fei, M. H. Ammar, and E. W. Zegura. Optimal allocation of clients to replicated multicast servers. In *Proc. of 1999 Int. Conf. on Network Protocols (ICNP'99)*, pp. 69–76, 1999.
- [3] 廣森聡仁 山口弘純 安本慶一 東野輝夫 谷口健一. 動画画像マルチキャスト配信における複製サーバ選択アルゴリズムの提案と性能評価. マルチメディア, 分散, 協調とモバイル (DICOMO 2001) シンポジウム論文集, pp. 331–336, 2001.
- [4] Berkeley MASH Research Group University of California. The network simulator ns-2, 2000. <http://www-mash.cs.berkeley.edu/ns/>.
- [5] W. Fenner and S. Casner. A “traceroute” facility for IP multicast. In *Internet Draft*, 2000.
- [6] Brian Neil Levine and J.J. Garcia-Luna-Aceves. A comparison of reliable multicast protocols. *Multimedia Systems*, Vol. 6, pp. 334–348, 1998.
- [7] Henning Schulzrinne, Stephen L. Casner, Ron Frederick, and Van Jacobson. RTP: A transport protocol for real-time application. In *Request For Comments 1889*, 1996.
- [8] V. Jacobson, S. McCanne, and M. Vetterli. Receiver-driven layered multicast. In *Proc. of ACM SIGCOMM'96*, pp. 117–130, 1996.