

特別研究報告

題目

実時間動画像マルチキャストにおける
動的なマルチキャストグループ再構成手法

指導教官

村田 正幸 教授

報告者

吉田 剛

平成 15 年 2 月 19 日

大阪大学 基礎工学部 情報科学科

実時間動画像マルチキャストにおける動的なマルチキャストグループ再構成手法

吉田 剛

内容梗概

テレビ会議やライブ放送など，複数のクライアントに同一内容の動画像データを実時間で提供する際には，IP マルチキャスト技術などを用いて動画像データをマルチキャスト配信することにより，サーバへの負荷を減らし，動画像配信に必要な帯域を節約することができる．しかしながら，利用可能な帯域やサーバまでの経路上の負荷状態などのネットワーク環境，画像処理能力や通信能力などのシステム処理能力，あるいはユーザの好みといった要因によりさまざまに異なるクライアントの要求品質や動画像データ受信状態に応じたサービスを提供するためには，クライアントの要求品質に合わせて複数のマルチキャストグループを生成し，それぞれに適した品質の動画像データをマルチキャスト配信しなければならない．したがって，クライアントの要求品質に大きなばらつきが生じた場合には，サーバは様々な異なる品質の同一内容の動画像データを実時間で多数生成，配信しなければならないため，サーバ負荷が増大するとともに，特にサーバ近傍ではネットワーク負荷の増大による輻輳が発生する．

我々の研究グループでは，複数のクライアントのさまざまに異なる要求品質を同時に満足することのできる，アクティブネットワーク技術を利用した，効率的な実時間動画像マルチキャスト手法を提案している．提案システムでは，階層的なマルチキャストグループを構成し，ローカルサーバと呼ばれる高機能ノードがローカルなネットワーク内のマルチキャストグループを管理し，動画像データを配信するローカルサーバとなる．これまでの研究では，ローカルサーバが，自身の管理するマルチキャストグループのクライアントの要求品質や動画像データ受信状態のばらつきにもとづいてマルチキャストグループを動的に分割 (split)，統合 (merge) することにより，適切にマルチキャストグループを構成する手法を提案し，シ

シミュレーションによりクライアントの動画像データ受信状態にあった動画像マルチキャスト配信が実現されることを示している。例えば、あるリンクが輻輳状態に陥った場合には、輻輳の影響を被った一部のクライアントにあわせて、マルチキャストグループ全体に配信される動画像の品質が低く変更されてしまう。マルチキャストグループの分割により、輻輳の影響を受けていないクライアントを別のマルチキャストグループに分けて収容すれば、これらのクライアントに対しては高品質な動画像を提供し続けることができる。しかしながら、直接輻輳の影響を受け、提供される動画像の品質が低下しているクライアントについては、マルチキャストツリーの構造が変化するわけではないため、分割によって動画像データ受信状態が改善されるわけではない。

本報告では、一時的な輻輳の影響を受けているマルチキャストグループの管理を他のローカルサーバへ移し、ボトルネックリンクを回避するような新たなマルチキャストツリーの構築を促すことにより、クライアントの動画像データ受信状態を改善し、マルチキャスト配信される動画像の品質を高める手法 (move) を提案している。動画像データ配信レートの低いマルチキャストグループについて、移動先となるローカルサーバを決定し、マルチキャストグループを管理しているローカルサーバからの動画像マルチキャスト配信とあわせて移動先候補のローカルサーバからも動画像データを配信し、これらの配信状況にもとづいて適切なアルゴリズムにより、移動の成否を決定する。シミュレーションによる評価により、ネットワークの負荷変動に応じて適切にマルチキャストグループの管理が引き渡され、分割/統合だけを行う場合と比較して、分割/統合/移動により、クライアントに提供される動画像の品質が向上することを示した。

主な用語

実時間動画像マルチキャスト, アクティブネットワーク, 動画像品質調整, QoS, TCP-friendly

目次

1	はじめに	7
2	アクティブネットワーク技術を利用した実時間動画像マルチキャスト	11
2.1	実時間動画像マルチキャストシステムの概要	11
2.2	アクティブネットワーク	16
2.3	TCP-friendly Multicast Congestion Control (TFMCC)	17
3	単一ローカルサーバにおけるマルチキャストグループの動的再構成	20
3.1	フィードバック情報の収集	20
3.2	TFMCCにおける配信レートの安定	23
3.3	マルチキャストグループの分割 (split)	23
3.4	マルチキャストグループの統合 (merge)	25
4	ローカルサーバ間のマルチキャストグループの移動 (move)	26
4.1	シグナリングメカニズム	30
4.2	移動候補となるマルチキャストグループの選択アルゴリズム	33
4.3	マルチキャストグループ移動の成否判定アルゴリズム	34
5	シミュレーションによる評価	47
5.1	評価方法	47
5.2	マルチキャストグループの再構成を行わない場合の評価結果	48
5.3	マルチキャストグループの分割 / 統合を行う場合の評価結果	54
5.4	マルチキャストグループの分割 / 統合 / 移動を行う場合の評価結果	60
6	おわりに	66
	謝辞	67
	参考文献	68

目次

1	サービスモデル	12
2	手順 2, 3 の実行例	13
3	手順 4 の実行例	14
4	ローカルサーバにおける TFMCC を用いた動画像マルチキャストとグループ再構成の処理モデル	22
5	動的再構成と TFMCC の制御タイミング	22
6	マルチキャストグループの分割と移動の例 (初期状態)	27
7	マルチキャストグループの分割と移動の例 (輻輳の発生)	28
8	マルチキャストグループの分割と移動の例 (分割)	29
9	マルチキャストグループの分割と移動の例 (移動)	29
10	制御メッセージのやりとり	31
11	動画像レートと品質の関係	34
12	移動テストによるトポロジ例	40
13	移動テストによるボトルネックの変化 (パターン 1)	40
14	移動テストによるボトルネックの変化 (パターン 2)	41
15	移動テストによるボトルネックの変化 (パターン 3)	41
16	移動テストによるボトルネックの変化 (パターン 4)	41
17	移動テストによるボトルネックの変化 (パターン 5)	42
18	移動テストによるボトルネックの変化 (パターン 6)	42
19	移動テストによるボトルネックの変化 (パターン 7)	42
20	移動テストによるボトルネックの変化 (パターン 8)	43
21	移動テストによるボトルネックの変化 (パターン 9)	43
22	移動テストによるボトルネックの変化 (パターン 10)	43
23	移動テストによるボトルネックの変化 (パターン 11)	44
24	移動テストによるボトルネックの変化 (パターン 12)	44
25	移動テストによるボトルネックの変化 (パターン 13)	44

26	移動テストによるボトルネックの変化 (パターン 14)	45
27	移動テストによるボトルネックの変化 (パターン 15)	45
28	移動テストによるボトルネックの変化 (パターン 16)	45
29	シミュレーションにおけるトポロジ (初期状態)	48
30	配信レートの変化 (再構成なし)	51
31	クライアントの受信レートの変化 (再構成なし)	51
32	クライアントのフィードバックした TCP-friendly レート (再構成なし)	52
33	クライアントの計測した RTT の変化 (再構成なし)	52
34	クライアントの packets 棄却率の変化 (再構成なし)	53
35	要求完了度の変化 (再構成なし)	53
36	シミュレーションにおけるトポロジ (分割時)	55
37	配信レートの変化 (split / merge)	57
38	クライアントの受信レートの変化 (split / merge)	57
39	クライアントのフィードバックした TCP-friendly レート (split / merge)	58
40	クライアントの計測した RTT の変化 (split / merge)	58
41	クライアントの packets 棄却率の変化 (split / merge)	59
42	要求完了度の変化 (split / merge)	59
43	シミュレーションにおけるトポロジ (移動テスト中)	61
44	シミュレーションにおけるトポロジ (移動後)	62
45	配信レートの変化 (split / merge / move)	62
46	クライアントの受信レートの変化 (split / merge / move)	63
47	クライアントのフィードバックした TCP-friendly レート (split / merge / move)	63
48	クライアントの計測した RTT の変化 (split / merge / move)	64
49	クライアントの packets 棄却率の変化 (split / merge / move)	64
50	要求完了度の変化 (split / merge / move)	65

表目次

- 1 ボトルネックリンクの位置と動画像データ配信レートの変化パターン 35
- 2 ボトルネックリンクの位置と動画像データ配信レートの変化の関係 46

1 はじめに

近年，インターネットアクセス回線の広帯域化やコンピュータの高速，高機能化を背景に，動画像配信などの分散型マルチメディアアプリケーションの利用者が増加している．そこで，マルチメディアアプリケーションの生成する多量のトラフィックをネットワークに過大な負荷を与えず効率よく転送するための研究が活発に行われている．例えば，テレビ会議やライブ放送など，複数のクライアントに同一内容の動画像データを同時に実時間で配信することが求められるアプリケーションに対しては，IP マルチキャスト技術などを用いたマルチキャスト通信が効果的である．IP マルチキャストによる動画像配信では，サーバは同じ動画像データを要求するクライアントを1つのマルチキャストグループにまとめ，サーバを根としたマルチキャストツリーを構成し，マルチキャストツリーに動画像データを送信する．動画像データは，マルチキャストツリーの節に位置するマルチキャストルータによって必要に応じて複製され，マルチキャストグループに参加するクライアントに配信される．

IP マルチキャストを用いて動画像データをマルチキャスト配信することにより，それぞれのクライアントにユニキャストで動画像データを配信する場合と比較して，サーバの負荷や動画像配信に必要な帯域を大きく減らすことができる．しかしながら，利用可能な帯域やサーバまでの経路上の負荷状態などのネットワーク環境，画像処理能力や通信能力などのシステム処理能力，あるいはユーザの好みといったさまざまな要因によりクライアントの要求，あるいは受信，再生可能な動画像の品質は異なる．そのため，全てのクライアントを一つのマルチキャストグループに収容し，同一品質の動画像データを配信する従来の動画像IP マルチキャストでは，クライアントは望む品質の動画像を得ることができない．例えば，輻輳の影響を受けている，またはシステム処理能力の低いクライアントに合わせて低品質な動画像データを配信すると，他のクライアントは利用可能な帯域に余裕があり，高いデータ受信，画像復号処理能力を有していたとしても低品質な動画像を受信，再生しなくてはならないため，ユーザを満足させることができない．逆に，要求品質の高いクライアントに合わせて高品質な動画像データを配信すると，要求品質の低いクライアントは画像処理能力の不足により動画像を再生できない，あるいは，利用可能な帯域より高いレートで動画像データが配信されるため輻輳が生じ動画像品質が劣化する，といった問題が生じる．

このような問題を解決するために、さまざま異なるクライアントの要求品質に合わせて複数のマルチキャストグループを生成し、それぞれに適した品質の動画像データをマルチキャスト配信することが考えられる。例えば、サイマルキャストと呼ばれる手法では、動画像サーバが複数の品質の異なる動画像データを同時に生成、それぞれ異なるマルチキャストアドレスを用いて配信し、クライアントは適切なマルチキャストグループに参加することにより、望む品質の動画像データを受信、再生できる。また、階層構造を持つ動画像データを生成し、階層ごとに異なるマルチキャストアドレスを用いて配信、クライアントが適切な組み合わせのマルチキャストグループに参加し、受信した動画像データを組み合わせて復号化することにより望む品質の動画像を得られるといった手法もある [1]。しかしながら、サイマルキャストでは、クライアントの要求する動画像の品質が大きくばらつく場合には、動画像符号化、送出処理によりサーバの負荷が増加し、特にサーバ近傍で多くの帯域が必要となる。また、階層符号化を用いた手法では、配信する動画像データの階層数に制限があり、さらに、階層の組み合わせによって画質やレートを選択するため多様な品質要求に対応できない [2]。

そこで、我々の研究グループでは、置かれる環境のさまざま異なる複数のクライアントに対し、それぞれの望む品質の動画像を効率よく配信するため、アクティブネットワーク技術を用いた動画像マルチキャスト配信手法を提案している [3,4]。アクティブネットワークでは、ネットワーク管理者や、サービス提供者、サービス利用者が、アクティブノードと呼ばれる高機能なノードにプログラムコードを導入し、高度なパケット処理能力を持たせることによって、ネットワークの状態やサービスに応じて柔軟かつ動的にネットワークのふるまいを変更することができる [5]。文献 [3,4] では、パケット単位の動画像品質調整機能およびマルチキャストサーバ機能を有するアクティブノードを用いて階層的にマルチキャストグループを構成し、アクティブノードがクライアントの要求品質や動画像データ受信状態に応じて自身の管理するマルチキャストグループを動的に再構成する手法を提案している。提案手法では、スタブ・ネットワークと呼ばれるローカルなネットワークとスタブ・ネットワークをつなぐコア・ネットワークからなる階層構造のネットワークを対象としている。スタブ・ネットワーク内のアクティブノードはローカルサーバと呼ばれる。動画像サーバからローカ

ルサーバにコア・ネットワークを経由して動画像データが配信され、クライアントはローカルサーバを介して動画像データを受信する。ローカルサーバは、自身の管理するマルチキャストグループのクライアントの要求品質や動画像データ受信状態に大きなばらつきがある場合、マルチキャストグループを2つに分割 (split) し、また、同程度の品質の動画像データを配信している2つのマルチキャストグループがあれば、これらを1つのマルチキャストグループに統合 (merge) する。

このようなマルチキャストグループの動的な分割/統合によって、ローカルサーバは、要求品質や動画像データ受信状態に近いクライアントごとにマルチキャストグループを動的に構成し、動画像サーバから受信した動画像に品質調整を施し、それぞれのマルチキャストグループに適切な動画像データを配信する。しかしながら、分割、統合は、あるローカルサーバがサービスを提供しているクライアントを適切なマルチキャストグループに収容するための制御であり、構築されるマルチキャストツリーはいずれも同じローカルサーバを根とするため、マルチキャストツリー上のボトルネックリンクが下流のクライアントやマルチキャストグループに与える影響を回避できない。例えば、あるマルチキャストツリーの一部のリンクで輻輳が発生すると、その下流のクライアントの動画像データ受信状態が他のクライアントと比較して大きく低下する。マルチキャストグループに配信される動画像は、グループ内で動画像データ受信状態の最も悪いクライアントに合わせて品質調整されるため、グループ内の全てのクライアントの提供される動画像品質が著しく低下する。そこで、ローカルサーバは動画像データ受信状態のばらつきにもとづいてマルチキャストグループを分割するが、これは動画像データ受信状態のよいクライアントを動画像データ受信状態の悪いクライアントから分離するためであり、輻輳からの回復を目的としたものではない。

そこで本報告では、マルチキャストグループの管理を同一スタブ・ネットワーク内の他のローカルサーバへ移すことにより、マルチキャストツリーを再構築し、動画像データのマルチキャスト配信を妨げるボトルネックリンクを避け、クライアントの動画像データ受信状態の向上を図る手法 (move) を提案する。ローカルサーバは、動画像データ配信レートの低いマルチキャストグループについて、移動先となるローカルサーバを決定し、移動先候補のローカルサーバからも同時に動画像データを配信し、これらの配信状況にもとづいて適切な

アルゴリズムにより，移動の成否を判断する．また，シミュレーションにより，ネットワークの負荷変動に応じて適切にマルチキャストグループの管理が引き渡され，分割／統合だけを行う場合と比較して，分割／統合／移動により，クライアントに提供される動画像の品質が向上するか評価，検討する．

本報告の構成は以下の通りである．まず，第 2 章で本報告で前提とする実時間動画像マルチキャスト配信システムについて述べる．続いて第 3 章で，我々の研究グループがこれまでに提案した動的なマルチキャストグループ再構成手法である分割 (split) と統合 (merge) の概要を述べる．第 4 章で，本報告の提案手法であるマルチキャストグループの管理を移すことにより動画像データ受信状態の向上を図る手法 (move) についてメカニズムおよびアルゴリズムを述べる．第 5 章では，第 4 章で提案した手法をシミュレーションを用いて評価する．最後に第 6 章で本報告のまとめと今後の課題を示す．

2 アクティブネットワーク技術を利用した実時間動画像マルチキャスト

2.1 実時間動画像マルチキャストシステムの概要

本研究で提案する動画像マルチキャスト配信サービスは、図1のような複数のスタブ・ネットワークとそれらをつなぐコア・ネットワークによって階層構造を形成するネットワーク上における実時間動画像マルチキャストを対象とする。システムは、1台の動画像サーバ、置かれる環境の異なる複数のクライアント、複数のマルチキャストルータ、複数のアクティブノード、およびそれらをつなぐ複数のリンクからなる。アクティブノードは動画像サーバからのプログラムコードの導入によって、動画像品質調整機能およびマルチキャストサーバ機能を提供する。

まず、動画像サーバは、動画像のタイトル、配信開始時刻や必要なシステム環境といった動画像マルチキャスト配信サービスに必要な情報をクライアントに伝える。クライアントは、スタブ・ネットワーク内の最寄りのアクティブノードにサービス参加メッセージを送信するとともに受信できる最大の動画像品質等の情報を伝えることによって、動画像配信サービスに参加することができる。また、サービスから離脱する場合は離脱メッセージを送信する。クライアントからの参加メッセージを受信したアクティブノードは、それらのクライアントに対して動画像データを提供するローカルサーバとなる。ローカルサーバは、動画像サーバや他のローカルサーバから動画像データを受信し、複数のマルチキャストグループを管理、それぞれのグループに合わせて品質調整した動画像データを配信する。ローカルサーバは、自身の管理する複数のマルチキャストグループのクライアントの要求品質や動画像データ受信状態のばらつきに応じて、動的にマルチキャストグループを分割、統合することにより、同程度の品質の動画像を要求するクライアントを同じマルチキャストグループに収容する。また、同一スタブ・ネットワーク内に存在する他のローカルサーバと協調し、マルチキャストツリーの再構成やマルチキャストグループの分担管理を行う。クライアントは、ローカルサーバによって指定されたマルチキャストグループに参加し、動画像データを受信、再生する。

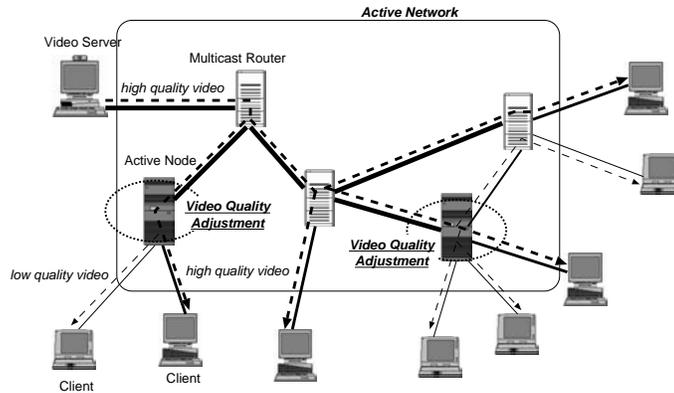


図 1: サービスモデル

提案システムの具体的な動作手順は以下のとおりである。

(1) サービス情報の通知

動画サーバは、動画のタイトルや時間、配信開始時刻、提供される品質種別、受信、再生に必要なシステム要件など動画マルチキャスト配信サービスに関する情報を、SAP (Session Announcement Protocol) [6]や SDP (Session Description Protocol) [7]を用いて、ネットワーク利用者に通知する。

(2) サービスへの参加

クライアントは、近傍のアクティブノードにサービス参加メッセージを送信することにより、サービスへの参加を表明するとともに、利用者の好みやクライアントシステムの制限などによって決定される動画品質の最大値を通知する。クライアントは、サービス参加メッセージを送ったアクティブノードによって管理される。ただし、マルチキャストグループ再構成によって他のアクティブノードに管理が委譲される場合がある。サービス参加メッセージを含め、メッセージやデータには必要に応じて ANEP (Active Network Encapsulation Protocol) ヘッダ [8] が付加されており、アクティブネットワーク技術にもとづくパケット処理が適用される。サービス参加メッセージを受信したアクティブノードは、必要なアクティブアプリケーションを取得、実行することにより、動画品質調整機能やマルチキャストサーバ機能などを獲得し、ローカルサーバとなる。

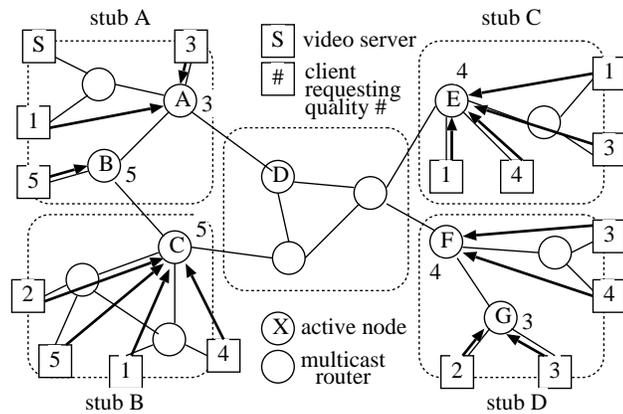


図 2: 手順 2, 3 の実行例

(3) 要求品質の集約

ローカルサーバは、クライアントからの要求品質のうち最も高いものについて動画像サーバに通知する。

図 2 に上記手順 2 および 3 の実行例を示す。図は A, B, C, および D の 4 つのスタブ・ネットワークとそれらを結ぶ一つのコア・ネットワークからなるネットワークを表している。図中, S と書かれた四角は動画像サーバ, アルファベットの書かれた白丸はアクティブノード, 白丸はマルチキャストルータをそれぞれ表す。また, 数字の書かれた四角はいずれもクライアントを表しており, それぞれの数字はクライアントが申告した要求品質を示している。なお, ここでは動画像の品質は 1 ~ 5 の 5 段階で表記され, 1 が最低品質の動画像を, 5 が最高品質の動画像を意味する。クライアントは, 矢印で指したアクティブノードに対して要求品質を申告する。アクティブノードは, 受信した要求品質を集約し, 最も高い要求品質をサーバに通知する。図では, それぞれのアクティブノードから動画像サーバに通知される要求品質を, アクティブノードの隣に示す。

(4) 動画像配信ツリーの構築

動画像サーバは, ネットワークのトポロジ, アクティブノードの負荷状態, および要求品質にもとづいて, ローカルサーバに動画像を配信するための階層化されたマルチキャストツリーを構築する。

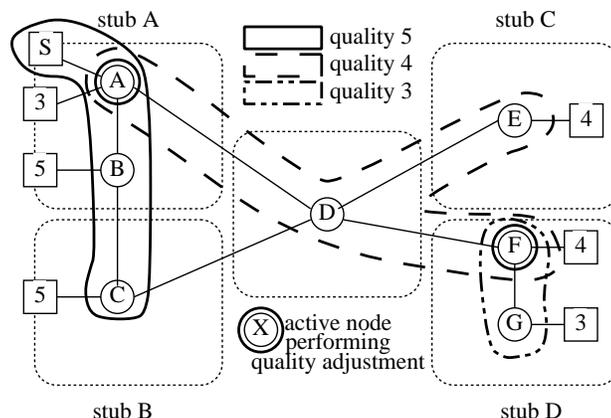


図 3: 手順 4 の実行例

図 3 に手順 4 の例を示す。手順 4 では、動画像サーバ、最高品質の動画像を要求する 1 台の仮想クライアントだけが接続されたアクティブノード、および、アクティブノード間を結ぶ最短経路を表す仮想リンクによって構成される論理ネットワークについて動画像配信ツリー構築アルゴリズム [9] を適用する。ツリー構築の結果、動画像サーバからはローカルサーバ A, B, C に対して最高品質の動画像がマルチキャスト配信されることとなる。ローカルサーバ A は、受信した最高品質の動画像に品質調整を適用することにより品質 4 の動画像データを生成し、マルチキャストサーバとなってローカルサーバ E および F に配信する。さらにローカルサーバ F は品質 4 の動画像データから品質 3 の動画像データを生成し、ローカルサーバ G に対してマルチキャスト配信する。このようにアクティブノードにおける動画像品質調整を利用して階層的にマルチキャストグループを構成することにより、さまざまな品質の動画像を効率的に配信することができる。なお、マルチキャストツリーはマルチキャストルーティングプロトコルによって構築される。また、ネットワークのトポロジ、アクティブノードの負荷状態、および要求品質の変化に対応するため、随時動画像配信ツリーを再構築する。

(5) ローカルサーバの設定

動画像サーバは、全てのローカルサーバに、動画像データの受信のためのマルチキャストグループのアドレスを通知する。

(6) クライアントの収容

ローカルサーバは、自身の管理する全てのクライアントを単一のマルチキャストグループに収容し、マルチキャストグループアドレスをクライアントに通知する。このマルチキャストグループでは、クライアントの要求品質のうち最も低いものにあわせて品質調整された動画データが配信されるが、本報告の提案手法により、動的にマルチキャストグループが再構成され、クライアントは置かれた環境に応じた適切な品質の動画像を受信することになる。

(7) 動画像配信ツリー上での動画像マルチキャスト配信

動画像サーバは、手順 (4) の結果にもとづいて、動画像データを生成、マルチキャスト配信する。ローカルサーバは、動画像サーバまたは他のローカルサーバから動画像データを受信する。また、必要に応じて品質調整を行った後、他のローカルサーバへマルチキャスト配信する。

(8) スタブ・ネットワークでの動画像マルチキャスト配信

ローカルサーバは、受信した動画像データをスタブ・ネットワーク内でマルチキャスト配信する。配信される動画像は、マルチキャストグループに属するクライアントの要求品質のうち最も低いものや下位層のレート制御機構の指定するデータ転送レートにあわせて品質調整される。

本報告では、帯域使用における TCP との公平性を考慮したマルチメディアマルチキャスト通信のためのレート制御機構である TFMCC (TCP-friendly multicast congestion control) [4, 10] を用いる。TFMCC では、それぞれのクライアントはパケット棄却率とマルチキャストサーバとの RTT から、TCP と公平なデータ転送レートを算出し、サーバにフィードバック情報として送信する。一方、サーバはクライアントの算出したレートのうち最も低いものをマルチキャスト通信におけるデータ転送レートとして選択する。したがって、TFMCC を用いた場合には、クライアントのうち最も受信レートの低いものに動画像データ転送レートがあろうよう、動画像品質調整を行うこととなる。TFMCC については 2.3 節で述べる。

(9) 動的なマルチキャストグループ再構成

ローカルサーバは、クライアントの要求品質や動画像データ受信状態の変化に応じてマルチキャストグループを分割、統合、移動する。例えば、マルチキャストグループ内のクライ

アントの動画像データ受信状態に大きなばらつきがある場合には、マルチキャストグループを分割し、それぞれに適した品質の動画像データを転送する。一方、同程度の動画像データを配信する2つのマルチキャストグループがある場合には、これらを一つのマルチキャストグループに統合する。また、マルチキャスト配信全体の品質向上を図るため、マルチキャストツリーのトポロジが変化することによりボトルネックリンクが回避され、動画像データ受信状態が向上することを期待し、マルチキャストグループの管理を同スタブ・ネットワーク内の他のアクティブノードに移す。マルチキャストグループ再構成の詳細なアルゴリズム、メカニズムは3章および、4章で述べる。

2.2 アクティブネットワーク

アクティブネットワークでは、ネットワーク管理者やサービス提供者、サービス利用者がアクティブノードと呼ばれる高機能なノードにプログラムコードを導入し、高度なパケット処理能力を持たせることによってネットワーク状況やサービスに応じた柔軟かつ動的にネットワークのふるまいを変更することができる [5]。アクティブノードの機能は、ノード OS、実行環境 (Execution Environment)、アクティブアプリケーション (Active Application) の階層構造をなしている。ノード OS は、実行環境がアクティブアプリケーションを実行するためのノードの資源、すなわちリンク帯域、CPU サイクル、ストレージやメモリの割り当てやスケジューリングを行う。実行環境は、ANEP ヘッダが付加された、あるいは送受信 IP アドレス、ポート番号、プロトコル番号などによって識別されたアクティブパケットを解釈し、アクティブアプリケーションを適用する仮想マシンであり、例えば JVM (Java Virtual Machine) などで実現される。アクティブアプリケーションは、アクティブパケットに処理を施すプログラムである。

アクティブノードへのプログラムの導入法には、プログラマブルスイッチ方式とカプセル方式の二つがある。プログラマブルスイッチ方式は、ノードで実行すべき処理プログラムをアクティブアプリケーションとして事前にアクティブノードに導入、アクティブノードによる処理を必要とするパケットには実行すべきプログラムの識別子を付加する。アクティブノードはこの識別子にもとづいてプログラムを適用する。カプセル方式では、プログラムそ

のものを処理を適用するパケット自身に内蔵して転送し，アクティブノードはパケットに付加されたプログラムに従って処理を実行する．

2.3 TCP-friendly Multicast Congestion Control (TFMCC)

帯域や転送遅延，パケット棄却に関する品質保証のないベストエフォート型のネットワークにおいては，過度のトラフィック流入による輻輳を防ぐため，ネットワークの負荷状態などに応じて適切にデータ配信レートを調整することが必要である．例えば，ユニキャスト用のトランスポート層プロトコルである TCP では，サーバはクライアントから返信される受信確認応答メッセージにもとづいてネットワークの混み具合を判断し，パケットの送出量を調整することにより輻輳制御を実現している．また，フィードバックメッセージから網内でのパケットの棄却を検出，再送することにより，信頼性のあるデータ通信を提供する．また，信頼性のあるマルチキャスト通信を行うためのマルチキャスト向け TCP なども提案されている [11]．一方，実時間動画像アプリケーションでは，動画像再生の途切れを防ぐため，動画像の符号化レート以上の一定のレートでのデータ転送が望ましく，またファイル転送などと異なり，多少のデータの欠落があってもアプリケーションの品質はそれほど大きく損なわれないため，遅延の要因となるパケット再送を必要としない．そのため，実時間動画像アプリケーションでは，トランスポート層で何の制御も行わない UDP を用いることが多い．しかしながら，UDP はネットワークの混み具合によらずアプリケーションの望むレートでデータを送出するため，多量のマルチメディアトラフィックの流入により帯域が UDP に占有され，TCP の通信性能が低下するだけでなく，ネットワークの輻輳を引き起こし，マルチメディア通信自身の品質も低下するという問題が指摘されている．そこで，UDP による実時間マルチメディア通信のためのレート制御手法としてユニキャスト用に TFRC (TCP-Friendly Rate Control) [12]，マルチキャスト用に TFRC の拡張として TFMCC (TCP-Friendly Multicast Congestion Control) [4, 10] が提案されている．

TFMCC では，クライアントはパケット棄却率と RTT (Round Trip Time) から，以下の式

を用いて，TCP-friendly レートと呼ばれるスループットの予測値を求める．

$$T_{TCP} = \frac{8s}{t_{RTT}(\sqrt{\frac{2p}{3}} + (12\sqrt{\frac{3p}{8}})p(1 + 32p^2))} \quad (1)$$

ここで， T_{TCP} は TCP-friendly レート [bps]， t_{RTT} は RTT [sec]， p はパケットロス率， s はパケットサイズ [byte] を表す．ただし，レートの変化を緩やかにするため RTT およびパケットロス率は平均化される．TCP-friendly レートは，同じ経路，パケット棄却率，RTT における TCP コネクションの平均スループットを近似的に導出したものであり [13]，マルチメディアアプリケーションがこのレートに合わせてマルチメディアデータ送出レートを調整すれば，TCP と公平に帯域を使用することができるとともに，輻輳制御を実現できる．

クライアントは，サーバとの間で設定された UDP コネクションを用いてクライアントにおける動画データ受信状態を表す T_{TCP} や伝搬遅延 t_{RTT} をフィードバック情報としてサーバに報告する．サーバは，クライアントの属するマルチキャストグループの現在の配信レートと T_{TCP} を比較し， T_{TCP} の方が高ければ配信レートを上げ， T_{TCP} の方が低ければ配信レートを下げる．ただし，TFMCC では，マルチキャストグループの中の全てのクライアントが良好に動画データを受信できるよう，サーバはクライアントから報告された TCP-friendly レート中最も低いものに合わせて配信レートを調節する．

TFMCC では，クライアント数の増加によるフィードバックトラヒックの爆発を防ぐための制御機構を用いている．マルチキャストへのデータ配信レートは全クライアント中の最も低い TCP-friendly レートにあわせるため，より高い TCP-friendly レートの報告は不要である．マルチキャストグループの中で最も低い TCP-friendly レートを報告したクライアントは CLR (Current Limiting Receiver) と呼ばれ，他のクライアントより優先してフィードバック情報を報告する．TFMCC では，マルチキャストグループに属するクライアントのうち，算出した TCP-friendly レートが現在の配信レートを下回るか，RTT がマルチキャストグループの最大 RTT を上回る場合に限り，フィードバックラウンドと呼ばれる最大 RTT の 6 倍で与えられる期間内に一度，サーバへフィードバック情報を報告することができる．ただし，CLR は自身の RTT に一度の頻度でサーバへフィードバックを報告することができる．CLR を指定する事で効果的に TCP-friendly レートを収集し，無駄なフィードバックを抑制するこ

とによって、フィードバックラヒックの爆発的な増大を防いでいる。

3 単一ローカルサーバにおけるマルチキャストグループの動的再構成

2.1 節で述べたように，ローカルサーバはマルチキャストグループ内の全てのクライアントが動画像データを受信できるように，マルチキャストグループ内で最も動画像データ受信状態の悪いクライアントに合わせて動画像品質調整を施した動画像データをマルチキャスト配信する．しかしながら，ネットワーク環境の変化などによりクライアントの動画像データ受信状態にばらつきが生じた場合，動画像データ受信状態の良いクライアントが要求品質より低品質な動画像データを受信しなければならなくなる．そこで，ローカルサーバは，ある定められた制御間隔ごとに，クライアントから収集した要求品質や動画像データ受信状態に関するフィードバック情報から得られるマルチキャストグループ内の動画像データ受信状態のばらつき具合にもとづきマルチキャストグループを動的に分割 (split)，または統合 (merge) することより，個々のクライアントにその要求品質に合った品質の動画像データを提供できるようにマルチキャストグループを再構成する [3]．例えば，マルチキャストグループ内のクライアントの TCP-friendly レートのばらつきが大きい場合，TCP-friendly レートの近いクライアント同士でマルチキャストグループが構成されるよう，ローカルサーバはマルチキャストグループを 2 つに分割する．また，配信レートの近いマルチキャストグループが複数存在した場合，これらのうち最も似通った 2 つのマルチキャストグループを統合し，マルチキャストグループの数を減らすことによりサーバへの負荷を軽減，帯域を節約する．本章では，このような単一ローカルサーバにおけるマルチキャストグループの再構成法を述べる．また，複数サーバを扱う場合については，次章で述べる．

3.1 フィードバック情報の収集

本報告では，実時間動画像マルチキャストのためのレート制御手法として TFMCC を用いている．マルチキャストグループの再構成のためには，ローカルサーバは，マルチキャストグループの全クライアントの TCP-friendly レートを知る必要があるが，2.3 節で述べたとおり TFMCC ではフィードバックの抑制を行っているため，TFMCC から得られる情報をそのまま用いることはできない．また，TFMCC では，CLR は RTT ごと，その他のクライアント

はフィードバックラウンド内の任意のタイミングでフィードバック情報を送信し，サーバはフィードバック情報を受信するたびに送信レートを制御する．大規模なネットワークにおいても RTT は高々数百ミリ秒～数秒であるため，TFMCC のレート変更に追従してマルチキャストグループを再構成するのはネットワークに大きな変動をもたらすため，適当でないと考えられる．そこで，マルチキャストグループの動的再構成には TFMCC と独立したフィードバック機構を用いる．

図 4 は，ローカルサーバにおける動的再構成メカニズムと TFMCC メカニズムの関係を示す概念図である．TFMCC は，フィードバック情報をクライアントから受信し，配信レートを決定する．TFMCC の指定する配信レートに応じて動画データが品質調整された後，TFMCC によりレート制御され，マルチキャスト配信される．一方，動的再構成メカニズムは，TFMCC とは別のコネクションを用いてクライアントからフィードバック情報を収集し，TFMCC から得られる配信レート，他のローカルサーバとの通信によって得られる情報とあわせて，マルチキャストグループの再構成を行う．

図 5 に，CLR ではないあるクライアントにおける，TFMCC および動的再構成のためのフィードバック情報の送出手の様子と，あるローカルサーバにおける，TFMCC および動的再構成のためのフィードバック情報の収集の様子と再構成制御のタイミングとの関係を示す．クライアントは，2.3 節で述べたとおり，フィードバックラウンドに 1 回だけ必要に応じてフィードバックを送信する．ローカルサーバの TFMCC は，マルチキャストグループのクライアントからフィードバック情報を受取るごとに，必要であれば送信レートを制御する．一方，動的再構成においては，クライアントは，定められた制御間隔毎にローカルサーバにフィードバック情報としてフィードバック送信時の TCP-friendly レートを報告する．したがって，ローカルサーバではフィードバック間隔内にクライアント数に等しいフィードバック情報を受信することになる．ローカルサーバは，フィードバック間隔より長く設定された制御間隔ごとに，制御間隔内に収集されたフィードバック情報にもとづいて，マルチキャストグループの動的再構成を行う．ただし，動画サービス参加時に申告した最大要求品質を上回るレートのフィードバックは行わない．

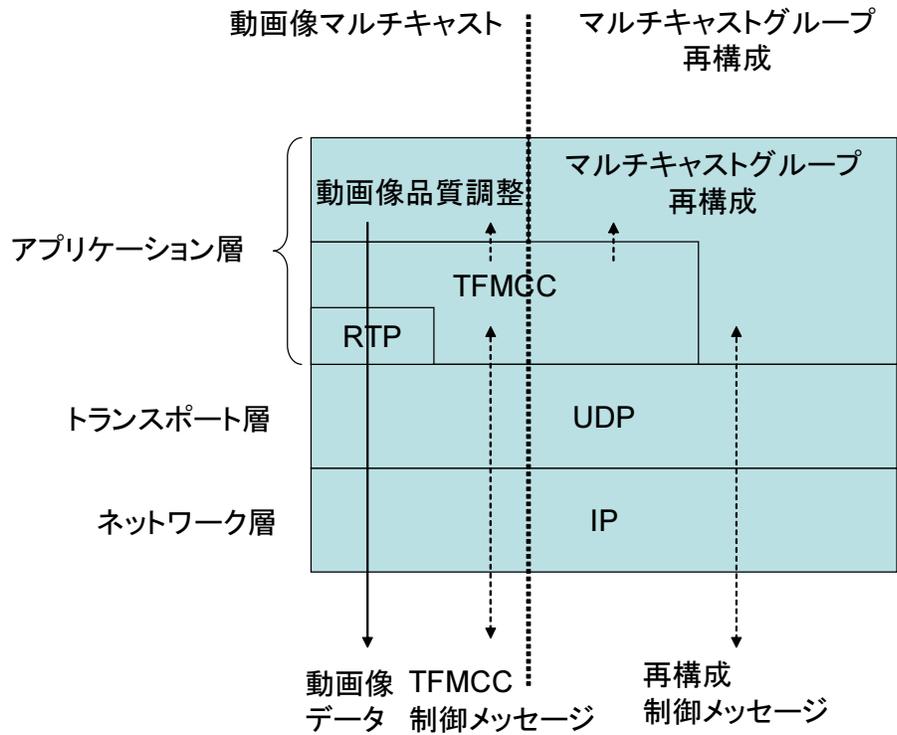


図 4: ローカルサーバにおける TFMCC を用いた動画マルチキャストとグループ再構成の処理モデル

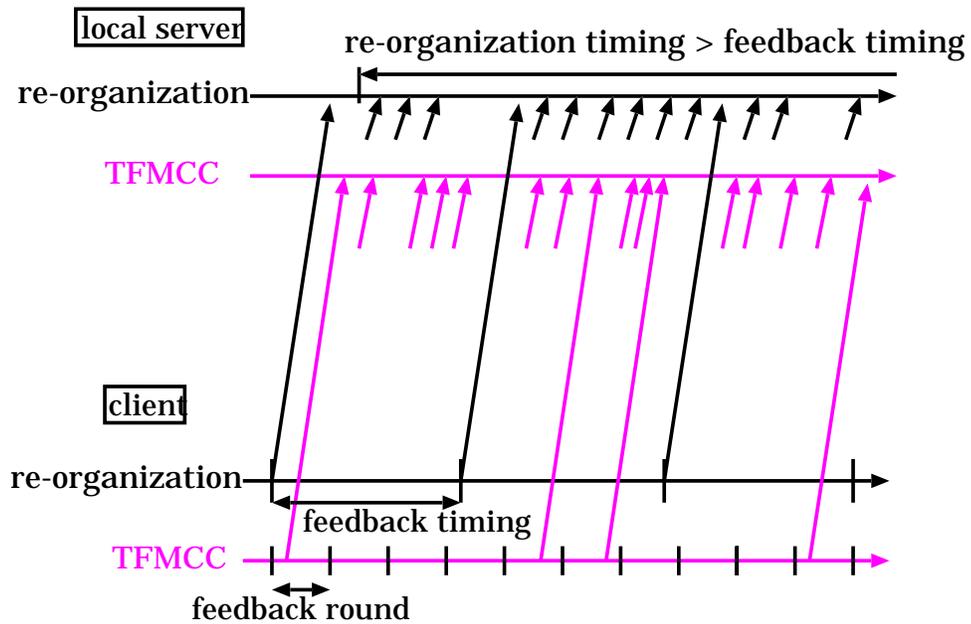


図 5: 動的再構成と TFMCC の制御タイミング

3.2 TFMCC における配信レートの安定

本稿では，マルチキャストグループの再構成の対象として，ローカルサーバからの配信レートが安定しているものを選択する．なぜなら，配信レートが不安定なマルチキャストグループを再構成することにより，不安定であることから，さらに分割などの再構成が起こる可能性が高いと考えられる．そのため，再構成が次々と生じることによるローカルサーバの負荷の増大や他のマルチキャストグループへの悪影響などが考えられるからである．また，マルチキャストグループの再構成はローカルサーバにできる限り負荷がかからないよう考慮すべきである．そこで，ローカルサーバはマルチキャストグループ再構成タイミング間隔ごとに，変動するマルチキャストグループの配信レートのばらつき C_v を算出し，その値が閾値 T_v より高い場合，そのマルチキャストグループは不安定とみなし，再構成の対象としない．

3.3 マルチキャストグループの分割 (split)

ローカルサーバは，自身の管理するマルチキャストグループについて，マルチキャストグループ内の全クライアントから報告された TCP-friendly レートから動画像データ受信状態のばらつきが最も大きいと判断されるマルチキャストグループの分割を試みる．

N_i 台のクライアントを収容しているマルチキャストグループ G_i ($1 \leq i \leq M$) における TCP-friendly レートのばらつきを変動係数 C_i で表す．また， M はローカルサーバの管理するマルチキャストグループの数を指す．変動係数 C_i は TCP-friendly レートの標準偏差 σ_i と，平均 μ_i から次式で表される．

$$C_i = \frac{\sigma_i}{\mu_i} \quad (2)$$

この変動係数 C_i が大きければマルチキャストグループ内のクライアントの動画像データ受信状態に大きなばらつきがあり，値が小さければ全てのクライアントが望み通りの動画像データを受信できていることを表す．ここで，マルチキャストグループ G_i のメンバーであるクライアント $c_{i,j}$ ($1 \leq j \leq N_i$) の TCP-friendly レートを $r_{i,j}$ [bps] とし，マルチキャストグループは配信レート昇順に，クライアントは TCP-friendly レート昇順に並んでいるもの

とする．したがって，マルチキャストグループ内のクライアント 1 がそのマルチキャストグループの CLR (current limiting receiver) となり，マルチキャストグループ G_i への配信レート R_i [bps] は $r_{i,1}$ になるようレート制御される．

まず，ローカルサーバはクライアントの TCP-friendly レートから各マルチキャストグループの変動係数 C_i ($1 \leq i \leq M$) を算出する．その中で最も大きい変動係数 $C_m = \max_i C_i$ が閾値 T_s より大きければマルチキャストグループ G_m を二つに分割する．分割後の各マルチキャストグループ内で動画データ受信状態のばらつきが小さくなるように，レート順に番号の連続したクライアントが収容される．例えば，分割点を k とすると，クライアント $c_{m,1} \sim c_{m,k-1}$ から成るマルチキャストグループとクライアント $c_{m,k} \sim c_{m,N_m}$ から成るマルチキャストグループの 2 つが生成される．マルチキャストグループ G_m のクライアント数は N_m であるため， $N_m - 1$ 通りの分割点が考えられる．ローカルサーバは，全ての分割点のうち，分割後の 2 つのマルチキャストグループの変動係数の平均値が最小となるものを選択する．

クライアント $c_{m,k} \sim c_{m,N_m}$ から成るマルチキャストグループ $G_{m'}$ には新しいマルチキャストアドレスが割り当てられ，新たに動画データが配信される．これまでの提案手法 [3] では，ローカルサーバの管理する全てのマルチキャストグループが分割の対象となっていた．しかしながら，ネットワークの負荷変動の影響を受けて動画データの配信レートの安定していないマルチキャストグループについては，クライアントの動画データ受信状態のばらつきが適切でないマルチキャストグループ構成のためか，負荷変動によるものか判断できないため，分割の対象とはしない．したがって M はローカルサーバの管理するマルチキャストグループのうち動画データ配信レートが安定しているものの数となる．また，従来手法では分割により新たに構成したマルチキャストグループへの配信レートは TFMCC の初期値にもとづいて設定されているが，本稿では，分割によるマルチキャストグループの増加によりネットワークへの負荷を増大させないため，2 つのマルチキャストグループの配信レートの合計が元のマルチキャストグループ G_m の配信レート R_m を越えないよう，元のマルチキャストグループの配信レート R_m とそれぞれのマルチキャストグループの CLR であるクライアント 1 およびクライアント k の TCP-friendly レート $r_{m,1}$ および $r_{m,k}$ にしたがって

それぞれ次式によって算出したレートで配信を開始，継続する．

$$R_m = R_m \times \frac{r_{m,1}}{r_{m,1} + r_{m,k}} \quad (3)$$

$$R_{m'} = R_m \times \frac{r_{m,k}}{r_{m,1} + r_{m,k}} \quad (4)$$

3.4 マルチキャストグループの統合 (merge)

マルチキャストグループの分割の後，ローカルサーバは同程度の品質の動画像データを配信する2つのマルチキャストグループの統合を試みる．ただし，同一の制御タイミングで分割したマルチキャストグループは統合の対象としない．

まず，番号の連続する，すなわち配信レートの近いと考えられる2つのマルチキャストグループの組み合わせ G_i と G_{i+1} ($1 \leq i \leq M-1$) について，動画像データ転送レート $R_i = r_{i,1}$ ， $R_{i+1} = r_{i+1,1}$ の変動係数 V_i を計算する．全ての組み合わせにおいて最も小さい変動係数 $V_m = \min_i V_i$ が閾値 T_m 以下であれば，マルチキャストグループ G_m と G_{m+1} を統合して新しいマルチキャストグループとする．統合後のマルチキャストグループの動画像転送レートは $r_{m,1}$ であり、マルチキャストグループ G_m の配信レートに等しいため、マルチキャストグループ G_{m+1} のクライアントにグループ G_m のマルチキャストアドレスを通知し，マルチキャストグループ G_{m+1} を解放する．

4 ローカルサーバ間のマルチキャストグループの移動 (move)

3章で述べたように、ローカルサーバは、マルチキャストグループの動的な分割、統合を行うことで、マルチキャストグループ内のクライアントの要求品質や動画データ受信状態のばらつきを抑え、クライアントにあった品質の動画を提供する。例えば、マルチキャストグループの分割では、動画データ受信状態の悪いクライアントと同一のマルチキャストグループに属する動画データ受信状態の良好なクライアントが新たに生成されたマルチキャストグループに收容されることにより、提供される動画の品質が向上する。しかしながら、動画データ受信状態の悪いクライアントの動画品質が、改善されるわけではない。

例えば、図6のようなスタブ・ネットワーク内について考える。2台のローカルサーバ S1 および S2, 4台のクライアント C1 から C4, マルチキャストルータ R1 から R4 からなる。4台のクライアントは全てローカルサーバ S1 の管理する一つのマルチキャストグループに收容され緑色の線のようにマルチキャストツリーが構築されている。また、スタブ・ネットワーク内には動画配信サービスに参加しないホスト T1 および D1 が存在し、ホスト T1 からホスト D1 へは TCP コネクションが設定されているものとする。初期状態では、全てのクライアントは高品質な動画データを受信することができるが、図7のようにホスト T1-D1 間で TCP セッションが開始されるとマルチキャストルータ R1-R2 間のリンク L1 で輻輳が発生し、下流に位置するクライアント C1 および C2 の算出する TCP-friendly レートが低くなる。その結果、マルチキャストグループ全体の配信レートが低下するため、ボトルネックリンクを通らないクライアント C3 および C4 にもクライアント C1 および C2 にあわせて品質調整された低品質な動画データが配信される。クライアント C3 および C4 の TCP-friendly レートは高いため、マルチキャストグループ内で TCP-friendly レートにばらつきが生じ、ローカルサーバ S1 はマルチキャストグループを二つに分割する。分割により、クライアント C3 および C4 にはボトルネックリンク L1 を通らない新たなマルチキャストツリーが構築され、これらクライアントの TCP-friendly レートに応じた高品質な動画データが配信されるようになる。一方、クライアント C1 および C2 へのマルチキャストツリーは依然としてリンク L1 を経由するため、これらのクライアントの受信する動画の品質は低いままである。

このように単一ローカルサーバでのマルチキャストグループの分割や統合では、ツリー

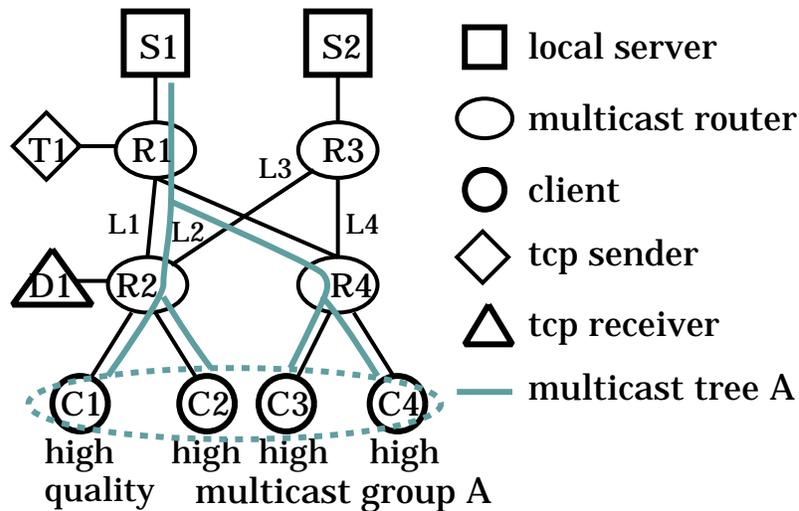


図 6: マルチキャストグループの分割と移動の例 (初期状態)

のトポロジがほとんど変化しないため、経路上で発生した輻輳が下流のクライアントやマルチキャストグループに与える悪影響を回避できない。そこで、本報告では新たに、ローカルサーバが、マルチキャストグループの移動、すなわち同一スタブ・ネットワーク内の他のローカルサーバへのマルチキャストグループの管理換えを試みる手法 (move) を提案する。例えば、図 8 からさらにクライアント C1 および C2 のマルチキャストグループをローカルサーバ S2 に移せば、図 9 のようにボトルネックリンクを回避することができ、輻輳によるパケット棄却がなくなり、クライアント C1 および C2 の TCP-friendly レートは高くなるため、配信される動画像の品質も向上する。

しかしながら、どのマルチキャストグループを移動の対象とするかについては、マルチキャストグループの移動によって動画像マルチキャスト配信サービス全体の品質がどれだけ向上するかについて考慮しなければならない。そこで、提案手法ではマルチキャストグループの移動による動画像データ受信状態の向上の可能性を測るため、移動の候補となったマルチキャストグループは、現在そのマルチキャストグループを管理しているローカルサーバ (以下、現サーバと呼ぶ) と、移動先の候補となるローカルサーバ (以下、候補サーバと呼ぶ) から同時に動画像データを受信する。移動テスト中の両サーバからの動画像データ受信状態から、マルチキャストグループの移動による動画像データ受信状態の向上が期待できる場

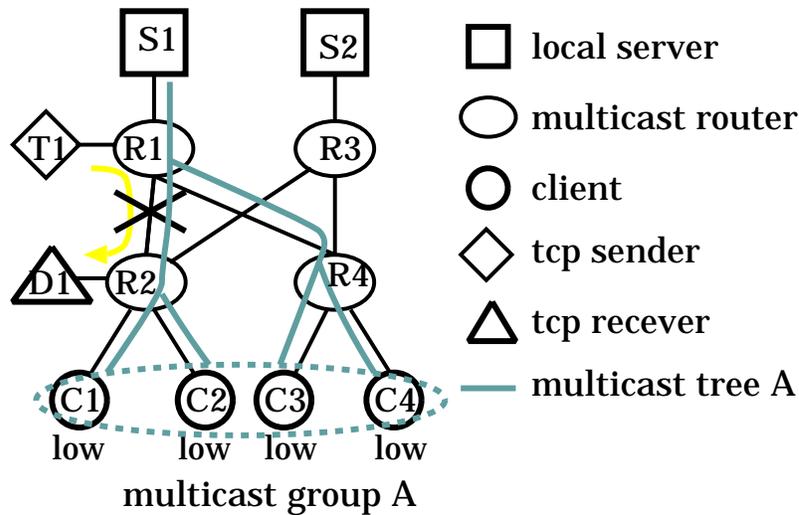


図 7: マルチキャストグループの分割と移動の例 (輻輳の発生)

合, マルチキャストグループの管理を候補サーバへ移す.

なお, 配信レートが不安定なマルチキャストグループや, 同一タイミングにおいて分割または統合の対象となったマルチキャストグループは移動の対象としない. また, ローカルサーバは, 自身の管理するマルチキャストグループの動画データ配信レート, クライアントからのフィードバックから得られる TCP-friendly レート, および他のローカルサーバとの通信により得られる情報のみ利用可能であり, マルチキャストツリーのトポロジやリンクの空き容量などを直接知ることはいできない. なお, スタブ・ネットワーク内のアクティブノードは, マルチキャストグループを管理中かどうかに関わらず全てローカルサーバとみなし, 移動先の候補となりうる.

本章では, マルチキャストグループの移動について, シグナリングメカニズムや移動の候補となるマルチキャストグループの決定方法, さらに, マルチキャストグループの移動による動画データ受信状態の向上の予測法について検討する.

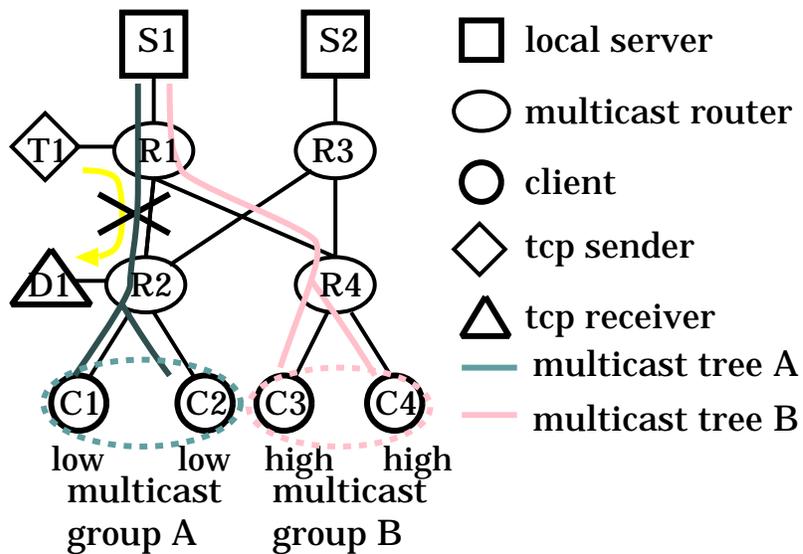


図 8: マルチキャストグループの分割と移動の例 (分割)

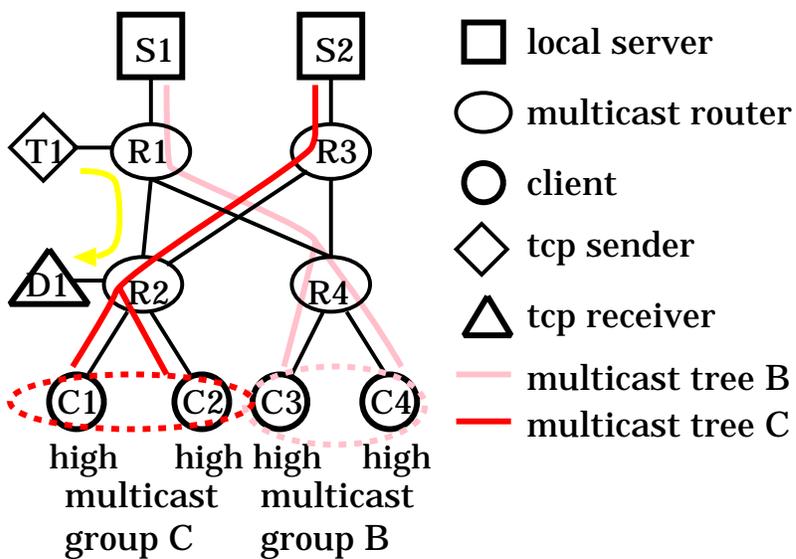


図 9: マルチキャストグループの分割と移動の例 (移動)

4.1 シグナリングメカニズム

本節では，図 10 にもとづき，マルチキャストグループの移動のメカニズムと制御メッセージのやりとりについて述べる．

1. 現サーバ (図中，current server) は，移動テストの要求，実中でなければ，管理している全てのマルチキャストグループ中，最も配信レートの低いものの移動を試みる．現サーバは，新たにマルチキャストグループを管理する候補となるサーバ (図中，candidate server) を決定し，マルチキャストグループを移動すべきか判断するための移動テストを準備するようテスト要求メッセージ (“probe_req”) を候補サーバに送信する．テスト要求メッセージには，メッセージを送信した時間を表すタイムスタンプが含まれている．
2. テスト要求メッセージを受信した候補サーバは，移動テストを実行していなければ現在管理中のマルチキャストグループの数，アクセスリンクの空き帯域，サーバの負荷などから移動テストを受け付けるかどうか判断する．また，すでに自身の管理するマルチキャストグループの移動のためのテスト要求をしている場合は，テスト要求した時刻とテスト要求メッセージに含まれているタイムスタンプを比較し，早いほうを優先する．移動テストを受け付ける場合には，候補サーバはテストセッションを開始するための準備を行う．ローカルサーバの変更時にも途切れなく動画像データの受信，再生を行えるよう，候補サーバはクライアントが現サーバから受信中のものと同じの動画像データをマルチキャスト配信する．そのため，動画像マルチキャスト配信に参加しておらず，動画像データを受信または中継していない場合，候補サーバは，動画像サーバもしくは他のローカルサーバに動画像データ配信を要求する．移動テストを行う準備が完了したら，テスト受諾メッセージ (“probe_ok”) を現サーバに送信する．テスト受諾メッセージには，移動テストを行うためのマルチキャストグループ (以降，テストグループと呼ぶ) のアドレスが含まれている．また，移動テストを受け付けない場合は，テスト拒否メッセージ (“probe_no”) を現サーバに送信する．テスト拒否メッセージには特別なパラメータは含まれない．

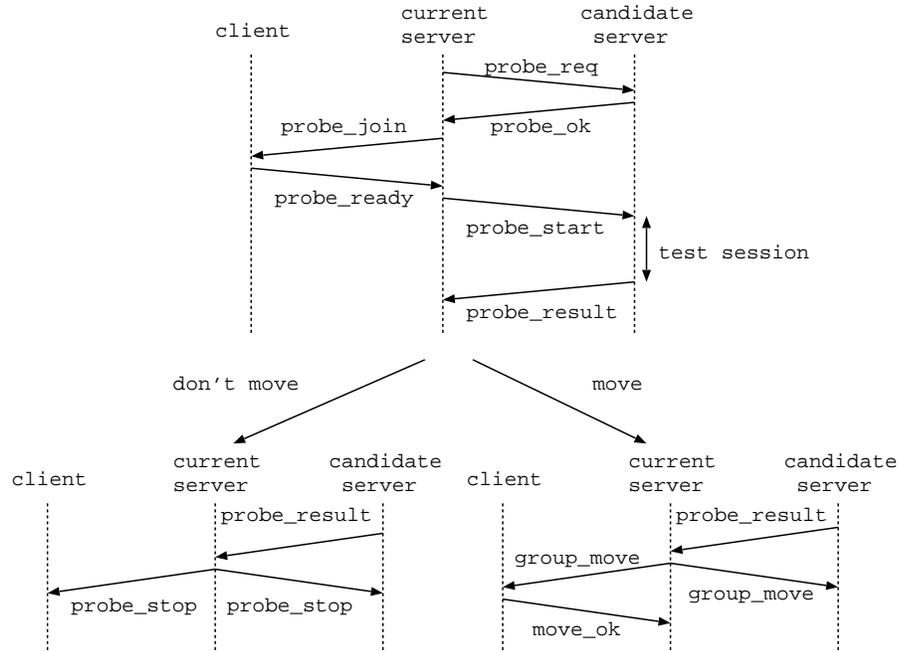


図 10: 制御メッセージのやりとり

3. 現サーバは、候補サーバからテスト受諾メッセージを受信すると、移動の候補であるマルチキャストグループの全クライアントにテスト参加要求メッセージ (“probe_join”) を送信する。テスト参加要求メッセージには、移動テストを行うためのテストグループのアドレスが含まれる。また、候補サーバからテスト拒否メッセージを受信した場合は、移動テストを終了する。
4. テスト参加要求メッセージを受信したクライアントは、候補サーバが準備したテストグループに参加し、現サーバにテスト準備完了メッセージ (“probe_ready”) を送信する。テスト準備完了メッセージには、特別なパラメータは含まれない。
5. 現サーバは、移動候補のマルチキャストグループの全クライアントからテスト準備完了メッセージを受信すると、候補サーバにテスト開始メッセージ (“probe_start”) を送信する。移動開始メッセージには、特別なパラメータは含まれない。また、全クライアントからテスト準備完了メッセージが届かない場合には、候補サーバと移動候補のマルチキャストグループの全クライアントにテスト停止メッセージ (“probe_stop”) を

送信しテストを終了する。テスト停止メッセージには特別なパラメータは含まれない。
なお、メッセージ待ちのタイムアウト時間は、移動候補となるマルチキャストグループの最大 RTT の 3 倍とする。

6. 候補サーバは、テスト開始メッセージを受信すると、テストグループへ動画データを送信する。移動テスト中も現サーバは動画データの送信を継続する。

なお、候補サーバはテストグループに対して再構成を行わない。また、フィードバックによるネットワークのトラヒックの増加を考慮して、クライアントは候補サーバには TFMCC の制御用のフィードバックのみ行う。クライアントは、移動が決定されるまで両方のマルチキャストグループから動画データを受信するが、候補サーバから受信した動画データはただちに破棄する。なお、移動テストは配信レートが決定するかタイムアウトとなるまで複数の制御期間にわたって継続されるが、移動テスト中のマルチキャストグループは分割、統合の対象とはしない。また、移動テストに参加している現サーバと候補サーバのいずれも自身の管理する他のマルチキャストグループに対する移動は試みず、他のローカルサーバからのテスト要求も受け付けない。

7. 候補サーバは、テストグループの配信レートが安定するか、ある定められた複数の再構成タイミングが経過すると、現サーバにテスト結果メッセージ (“probe_result”) を送信する。テスト結果メッセージにはテストグループへの配信レートが含まれる。

8. 現サーバは、テスト結果メッセージを受信すると、テストグループの配信レートと現在動画データを提供している移動候補のマルチキャストグループの配信レートを比較し、より高品質な動画データを配信できると判断したローカルサーバに当該マルチキャストグループの管理を移す。

現サーバが引き続きマルチキャストグループを管理する場合には、現サーバはテスト停止メッセージ (“probe_stop”) を候補サーバと移動候補であったマルチキャストグループの全クライアントに送信する。テスト停止メッセージには停止対象であるテストグループのアドレスが含まれる。テスト停止メッセージを受け取った候補サーバは、移動テスト用の動画データ配信を停止し、マルチキャストグループを解放する。また、

クライアントはテストグループから離脱し、現サーバから引続き動画像データを受信する。

一方、候補サーバがマルチキャストグループを新しく管理する場合、現サーバは移動メッセージ (“group_move”) を候補サーバと移動対象であるマルチキャストグループの全クライアントに送信する。移動メッセージには特別なパラメータは含まれない。クライアントは、移動メッセージを受信すると、候補サーバから受信している動画像データを復号、表示するようにし、現サーバからの配信のためのマルチキャストグループから離脱するとともに、現サーバに移動完了メッセージ (“move_ok”) を送信する。移動完了メッセージには特別なパラメータは含まれない。候補サーバは、動画像データの配信を継続する。現サーバは、移動対象のマルチキャストグループの全てのクライアントから移動完了メッセージを受信すると、当該マルチキャストグループへの動画像データ配信を終了する。

4.2 移動候補となるマルチキャストグループの選択アルゴリズム

マルチキャストグループの移動では、マルチキャストグループを管理するローカルサーバを変更することによって、下位のマルチキャストルーティングプロトコルにより決定されるマルチキャストツリーがより空き帯域の大きいリンクを通るトポロジに変化し、動画像データ受信状態が向上することを期待する。ここで、 i 番目のマルチキャストグループ G_i の j 番目のクライアントの RTT を $t_{i,j}$ [sec]、TCP-friendly レートを $r_{i,j}$ [bps]、マルチキャストグループ G_i の配信レートを R_i [bps] と表記する。また、マルチキャストグループは配信レート昇順に番号が割り当てられ、クライアントは属するマルチキャストグループ内の TCP-friendly レート昇順に番号が割り当てられている。

ローカルサーバの管理する全てのマルチキャストグループについて移動による品質向上の可能性はある。したがって、移動の対象とするマルチキャストグループを決定するためには、移動によってローカルサーバの提供するマルチキャスト配信サービス全体の品質がどれだけ向上するかを考慮する必要がある。しかしながら、マルチキャストグループの移動による品



図 11: 動画レートと品質の関係

質向上の程度はあらかじめ予測不能である．一般に，動画における符号化レートと品質の関係は図 11 のように上に凸な関数となると考えられ，移動による配信レート向上による品質向上の度合いは配信レートが低いマルチキャストグループの方が大きいと期待できる．そこで，ローカルサーバは自身の管理するマルチキャストグループのうち，最も配信レートが低いマルチキャストグループについて移動を試行する．

4.3 マルチキャストグループ移動の成否判定アルゴリズム

マルチキャストグループの移動では，候補となったマルチキャストグループに対して現サーバと候補サーバの両方から同時に動画データを送信し，配信状態にもとづいて，ローカルサーバの切り替えによりクライアントの動画データ受信状態が向上し，より高い品質の動画を得ることができるかどうかを判定する．現サーバは，自身の管理するマルチキャストグループの配信レートとクライアントの動画データ受信状態，および配信レートの安定時に候補サーバから通知されるテストグループからの配信レートのみから移動の成否を判定しなければならず，本節では，そのためのアルゴリズムを提案する．

テストグループにおける動画配信と現サーバからの動画配信は，互いに影響を与えあう．マルチキャストツリーにおける移動テスト前，移動テスト中，マルチキャストグループの移動後のボトルネックリンクの位置の変化に着目すると，現サーバからのマルチキャストツリーにおける移動テスト前，移動テスト中のボトルネックリンクの位置の変化と，候補

表 1: ボトルネックリンクの位置と動画データ配信レートの変化パターン

			候補ツリー			
			テスト中のボトルネック			
			独立リンク		共有リンク	
			移動後のボトルネック		移動後のボトルネック	
			独立リンク	共有リンク	独立リンク	共有リンク
現ツリー	テスト前のボトルネック	独立リンク	パターン 13	パターン 11	パターン 12	パターン 14
		共有リンク	パターン 6	パターン 5	パターン 1	パターン 2
	共有リンク	独立リンク	パターン 9	パターン 10	パターン 7	パターン 8
		共有リンク	パターン 16	パターン 15	パターン 3	パターン 4

サーバからのマルチキャストツリーにおける移動テスト中，候補サーバへのマルチキャストグループの移動後のボトルネックリンクの位置の変化について，それぞれボトルネックリンクが二つのマルチキャストツリーが共有するリンクに位置するのか独立するリンクに位置するのかによって，表 1 に示す 16 通りの状況が考えられる．以降ではそれぞれの状況における，移動テスト前，移動テスト中，移動後の現サーバおよび候補サーバからの動画データ配信レートの関係を整理し，どのような関係が観測されればマルチキャストグループ移動による配信レートの向上が期待できる状況であると判断できるかについて考察する．

移動テスト前の現サーバからの配信レートを R_p [bps]，移動テスト中の現サーバからの配信レートを R [bps]，移動テスト中の候補サーバからの配信レートを r [bps]，移動後の候補サーバからの配信レートを r_a [bps] と表記する．なお，簡単のため，現サーバからの動画配信のためのマルチキャストツリーを現ツリー，候補サーバからの動画配信のためのマルチキャストツリーを候補ツリーと呼ぶ．また，考察を容易にするため，ツリーのトポロジを模式的に図 12 のように示す．ここで，それぞれのトポロジにおいて左の四角を現サーバ，

右の四角を候補サーバとし，楕円を中継するマルチキャストルータ，丸をサービスに参加しているクライアントとする．また，左から移動テスト前，移動テスト中，移動後のトポロジを表し，緑色の線が現ツリー，青色の線が候補ツリーを表している．また，図 12 では描いていないが，赤色の×が現ツリーのボトルネック，紫色の×が候補ツリーのボトルネックを指す．

現ツリーと候補ツリーのテスト中のボトルネックリンクが同一共有リンクである場合には，移動テスト前，移動テスト中，および移動後の現ツリーおよび候補ツリーのボトルネックリンクの変化によってパターン 1 から 4 に分けられる．まず，移動テスト前後で現ツリーのボトルネックリンクが変化し，かつ移動テスト中と移動後の候補ツリーのボトルネックリンクが変化する場合には図 13 のようになる．よって，移動テスト前は現サーバからの配信レートはボトルネックリンクとなるリンク A で利用可能な帯域により決定し，移動テスト中はリンク C で利用可能な帯域により決定される．移動テスト前にはボトルネックでなかったリンク C が移動テストによって新たに現ツリーにとってのボトルネックリンクとなるため $R_p > R$ となる．また，リンク C は候補ツリーにとっても移動テスト中のボトルネックリンクであり，TFMCC によりレート制御を行う両ツリーの配信レートは等しくなるため， $r = R$ となる．また，マルチキャストグループの移動後にリンク B が新たなボトルネックリンクとなることから， $r_a > r$ である．したがって，以下の関係が成立する．

$$R_p > R = r < r_a$$

移動テスト前の動画データ配信レート R_p とマルチキャストグループ移動後の配信レート r_a に注目すると， $R_p > r_a$ または $R_p < r_a$ であることから，マルチキャストグループの移動による配信レート向上の可能性がある．

移動テスト前後で現ツリーのボトルネックリンクは変化するが移動テスト中と移動後の候補ツリーのボトルネックリンクが変化しない場合 (パターン 2)，図 14 のようになる．移動テスト前は現サーバからの配信レートはボトルネックリンクであるリンク A で利用可能な帯域により定められ，移動テスト中はリンク C で利用可能な帯域により決まる．移動テストによりリンク C が新たなボトルネックリンクとなるため $R_p > R$ である．また，リンク C は候補ツリーにとっても移動テスト中のボトルネックリンクであるため $r = R$ である．この

時，リンク C を競合するセッション数を N とすると，リンク容量は NR と推定できる．ただし，セッション数には現ツリー，および候補ツリーにおける動画像配信のためのセッションも含まれる．マルチキャストグループの移動によって現ツリーの配信が停止した後もリンク C がボトルネックリンクであれば，リンク C の容量が全てのセッションにより公平に分割されているので， $r_a = \frac{NR}{N-1} = \frac{Nr}{N-1} > r$ となる．したがって，

$$R_p > R = r < r_a$$

が成立し， $R_p > r_a$ または $R_p < r_a$ であるからマルチキャストグループの移動による配信レート向上の可能性はある．

次に，移動テスト前と移動テスト中で現ツリーのボトルネックリンクが同じ場合について検討する．移動テスト中と移動後の候補ツリーのボトルネックリンクが変化するパターン 3 では，図 15 のように，現ツリーのボトルネックはリンク C であるので移動テストによって候補ツリーのトラフィック流入により $R_p > R$ となる．また，リンク C が候補ツリーにとっても移動テスト中のボトルネックリンクであるため TFMCC により $r = R$ となる．さらに，マルチキャストグループの移動によってリンク B が新たなボトルネックリンクとなるので $r_a > r$ であるが，リンク C はボトルネックリンクでないので $R_p > r_a$ である．したがって，

$$R_p > r_a > r = R$$

となり， $R_p > r_a$ であるからマルチキャストグループの移動による配信レートの向上は期待できない．

一方，移動テスト中と移動後の候補ツリーのボトルネックリンクが同じ場合 (パターン 4) には，図 16 のようになる．現ツリー，候補ツリー共にリンク C がボトルネックリンクとなるので，移動テストにより $R_p > R$ となり，移動テスト中は $r = R$ となる．また，移動後もリンク C がボトルネックであるから $r_a = R_p$ となる．したがって，

$$r_a = R_p > R = r$$

となり， $R_p = r_a$ であるからマルチキャストグループの移動による配信レートの向上は期待できない．

続いて、現ツリーと候補ツリーが移動テスト中に別のリンクをボトルネックとする場合について考察する。移動テスト中に同じリンクをボトルネックとする場合 (パターン 1-4) と同様、まず移動テスト前と移動テスト中の現ツリーのボトルネックリンクが変化する場合について考える。現ツリーのボトルネックが移動テスト前はリンク A で移動テスト中はリンク C となる場合には、移動テスト中と移動後の候補ツリーのボトルネックリンクが変化する場合 (パターン 5) では、図 17 のように移動テスト中の候補ツリーのボトルネックリンクがリンク B で移動後がリンク C となることを意味する。しかしながらマルチキャストグループの移動によって現ツリーは動画配信を停止するためリンク C に空き帯域ができるため、移動後の候補ツリーにとってリンク C はボトルネックリンクになることはなく、従ってこのような状態は発生しない。

一方、移動テスト中と移動後の候補ツリーのボトルネックリンクが同じパターン 6 では、図 18 のようになり、候補ツリーのボトルネックリンクはリンク B である。ボトルネックリンクの変化から $R_p > R$ であり、また $R > r = r_a$ となる。したがって、

$$R_p > R > r = r_a$$

となり、 $R_p > r_a$ であるからマルチキャストグループの移動による配信レートの向上は期待できない。

現ツリーのボトルネックリンクが移動テスト前はリンク C で移動テスト中はリンク A の場合 (パターン 7-10) には、移動テストによってリンク C には新しくテスト用のトラフィックが混入する一方で共有されていないリンク A の負荷は変わらないため、移動テスト前にボトルネックリンクでなかったリンク A が移動テスト中にボトルネックリンクになることはなく、これらの状況は発生しない。

次に、現ツリーと候補ツリーが移動テスト中に別のリンクをボトルネックとするが、移動テスト前と移動テスト中で現ツリーのボトルネックリンクが変化しない場合について考える。まず、現ツリーのボトルネックリンクがリンク A で、移動テスト中から移動後で候補ツリーのボトルネックリンクがリンク B からリンク C に変化することが考えられるが (パターン 11)、マルチキャストグループの移動によって現ツリーは動画配信を停止し、リンク C に空き帯域ができるため、移動後の候補ツリーにとってリンク C がボトルネックリン

クになることはなく、このような状況は発生しない。

逆に、候補ツリーのボトルネックリンクがリンク C からリンク B に変化する場合 (パターン 12) には、現ツリーのボトルネックリンクがリンク A で変化しないことから $R_p = R$ であり、移動テスト中の候補ツリーのボトルネックリンクがリンク C であるため $r > R$ となり、さらに $r_a > r$ である。したがって、

$$r_a > r > R = R_p$$

が成立し、 $r_a > R_p$ であるからマルチキャストグループの移動により配信レートが向上する。

また、候補ツリーのボトルネックリンクが常にリンク B である場合 (パターン 13)、現ツリーのボトルネックリンクがリンク A で変化しないので $R_p = R$ であり、また候補ツリーもボトルネックリンクがリンク B であるので $r = r_a$ となる。したがって、

$$r_a > R_p, \text{ if } r > R$$

$$r_a \leq R_p, \text{ if } r \leq R$$

となり、移動テスト中の配信レート r および R の大小関係によってマルチキャストグループの移動による配信レート向上の成否が決定する。

一方、候補ツリーのボトルネックリンクが変化せずに常にリンク C の場合 (パターン 14) には、現ツリーにおいては $R_p = R$ であり、候補ツリーのボトルネックリンクはリンク C であるため $r > R$ となる。さらにマルチキャストグループの移動によって現ツリーでのマルチキャスト配信が終了しリンク C に空きが生じるため $r_a > r$ である。したがって、

$$r_a > r > R = R_p$$

が成立し、 $r_a > R_p$ であるからマルチキャストグループの移動により配信レートが向上する。

最後に、移動テストにより現ツリーのボトルネックリンクが変化せずに常にリンク C の場合について検討する。移動テスト中と移動後の候補ツリーのボトルネックリンクがリンク B からリンク C に変化する場合には、マルチキャストグループの移動によって現ツリーが動画像配信を停止し、リンク C に空き帯域ができるため、移動後の候補ツリーにとって新たにリンク C がボトルネックになることはない。

一方，候補ツリーのボトルネックリンクが常にリンク B の場合 (パターン 16) には，移動テストによって $R_p > R$ となる．また，候補ツリーのボトルネックリンクがリンク B であるため $R > r$ ， $r_a = r$ となる．したがって，

$$r_a = r < R < R_p$$

となり， $R_p > r_a$ となり，マルチキャストグループの移動による配信レートの向上は期待できない．

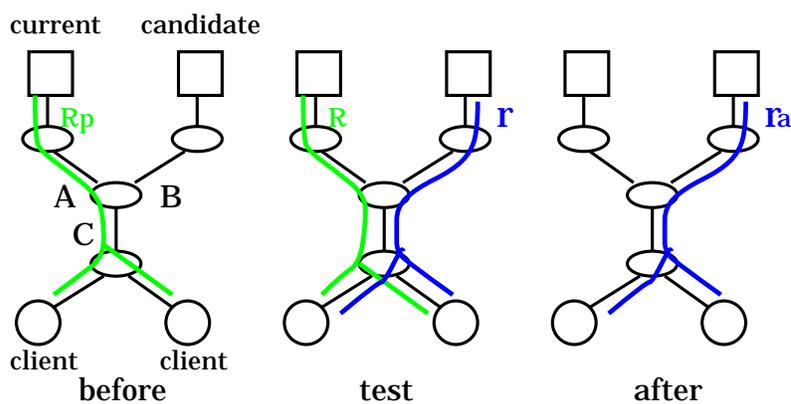


図 12: 移動テストによるトポロジ例

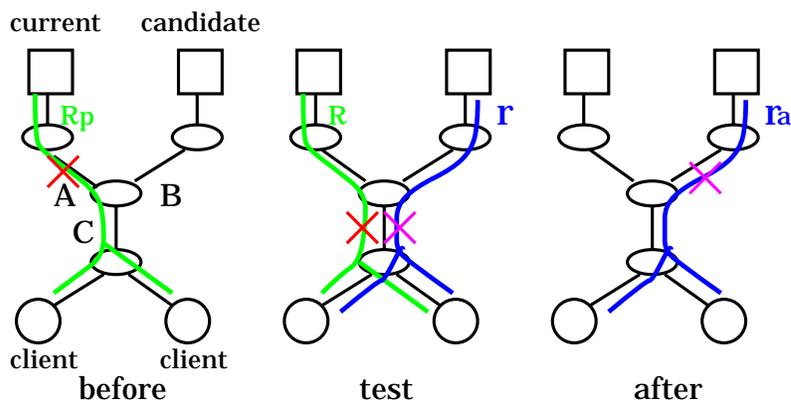


図 13: 移動テストによるボトルネックの変化 (パターン 1)

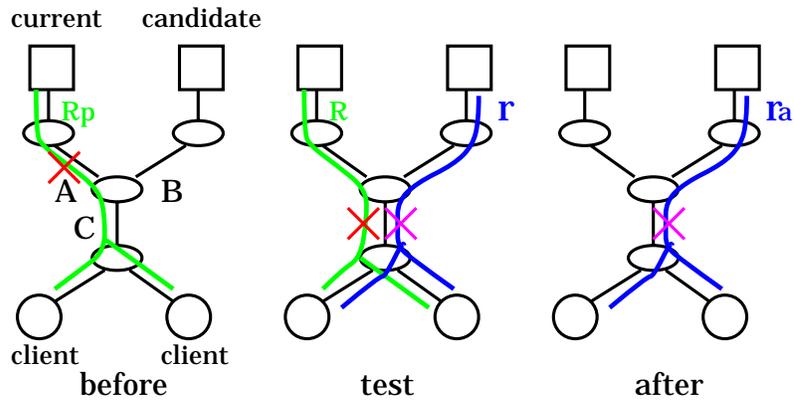


図 14: 移動テストによるボトルネックの変化 (パターン 2)

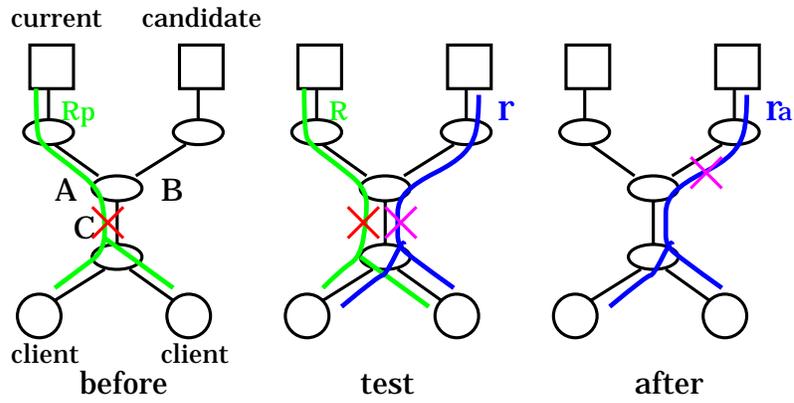


図 15: 移動テストによるボトルネックの変化 (パターン 3)

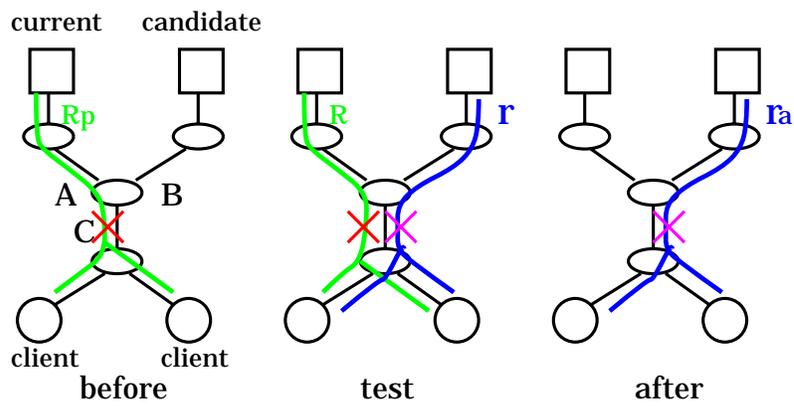


図 16: 移動テストによるボトルネックの変化 (パターン 4)

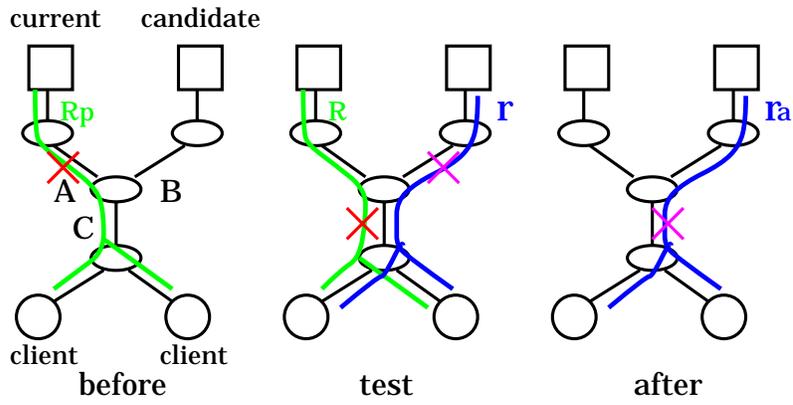


図 17: 移動テストによるボトルネックの変化 (パターン 5)

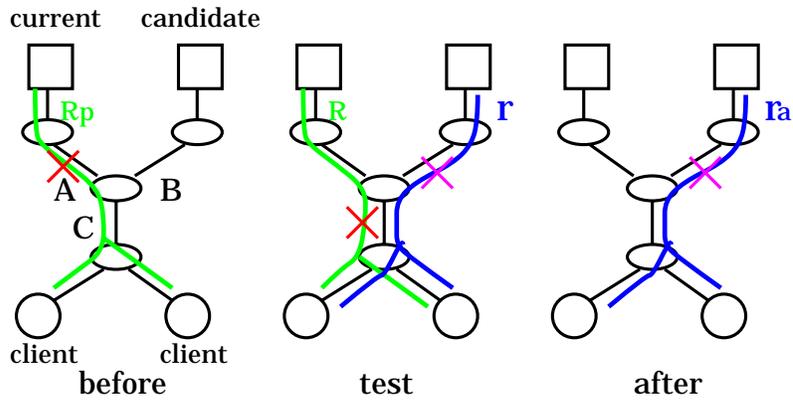


図 18: 移動テストによるボトルネックの変化 (パターン 6)

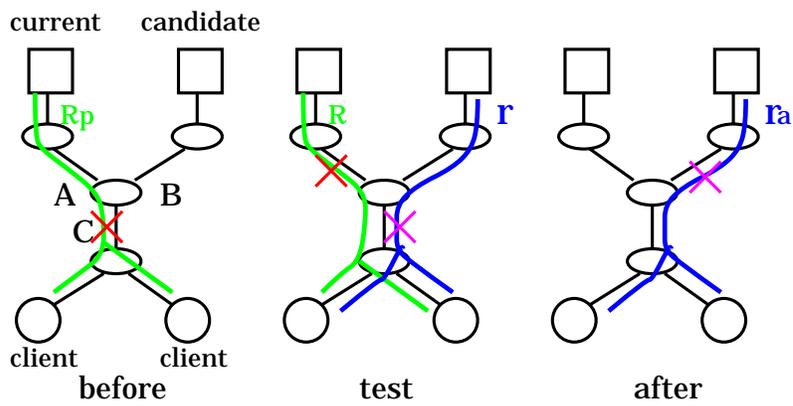


図 19: 移動テストによるボトルネックの変化 (パターン 7)

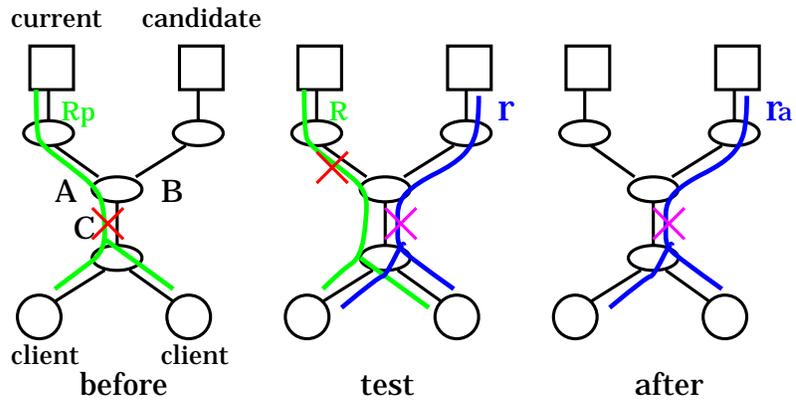


図 20: 移動テストによるボトルネックの変化 (パターン 8)

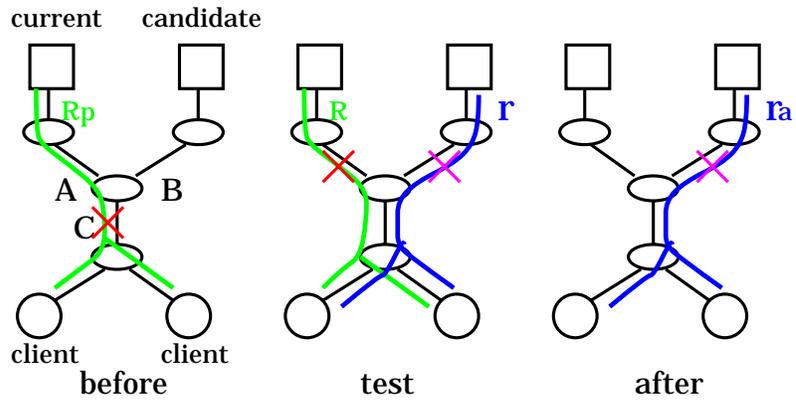


図 21: 移動テストによるボトルネックの変化 (パターン 9)

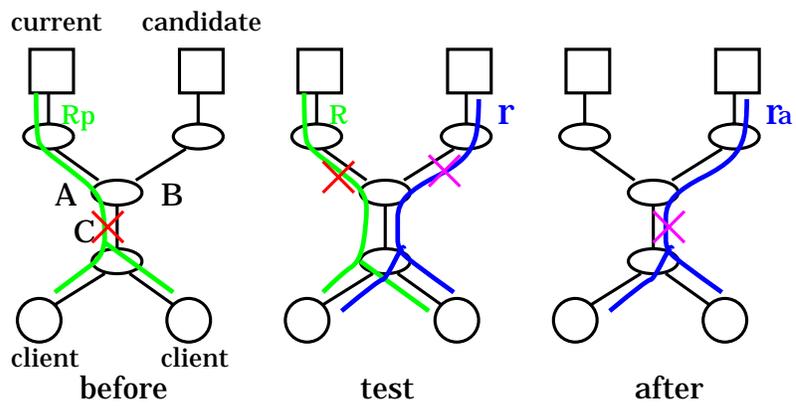


図 22: 移動テストによるボトルネックの変化 (パターン 10)

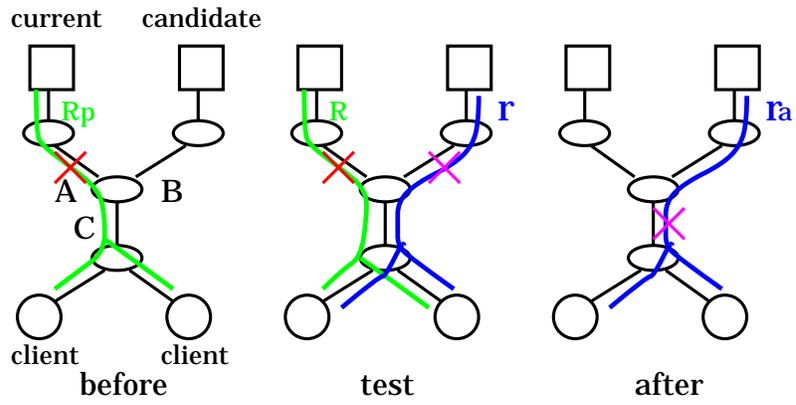


図 23: 移動テストによるボトルネックの変化 (パターン 11)

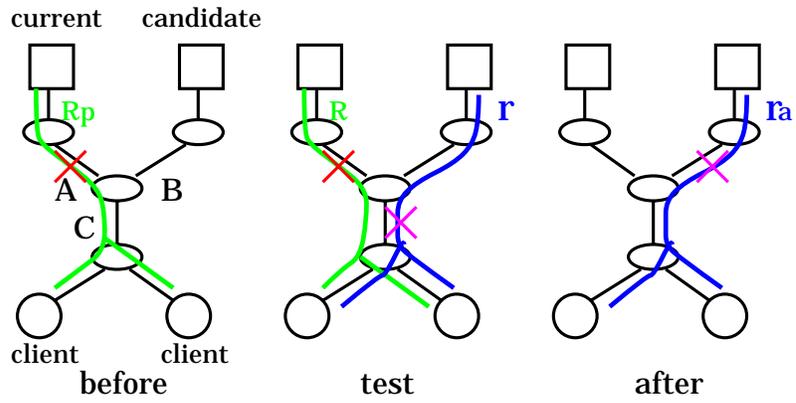


図 24: 移動テストによるボトルネックの変化 (パターン 12)

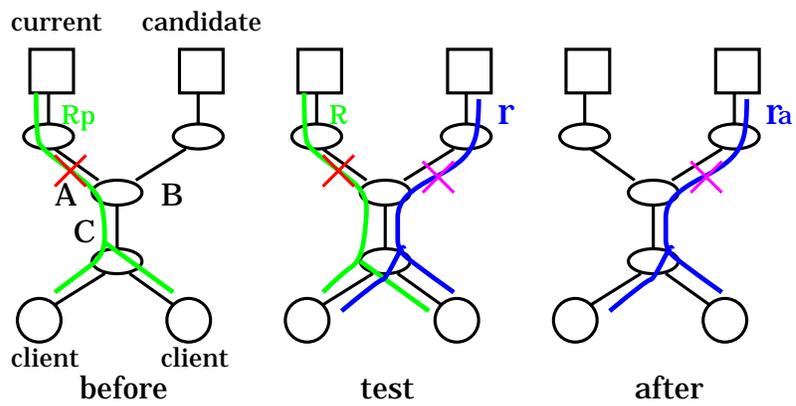


図 25: 移動テストによるボトルネックの変化 (パターン 13)

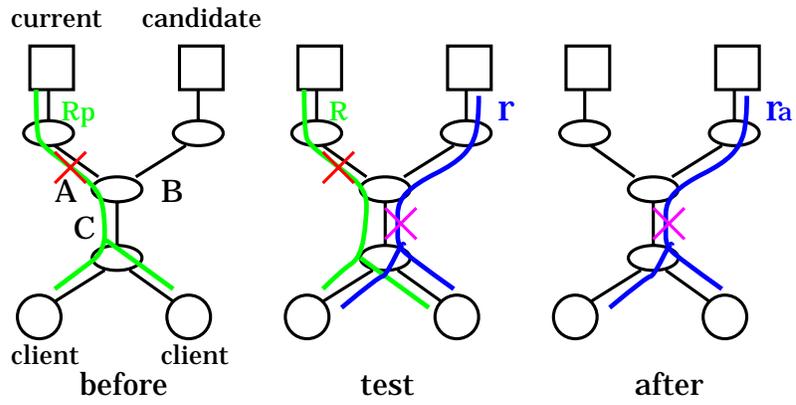


図 26: 移動テストによるボトルネックの変化 (パターン 14)

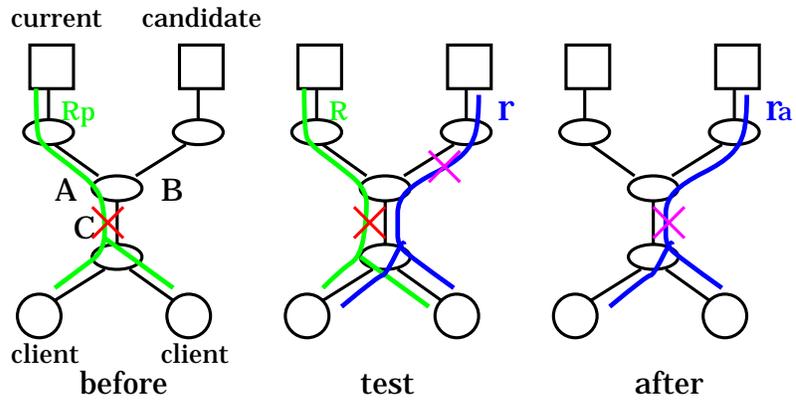


図 27: 移動テストによるボトルネックの変化 (パターン 15)

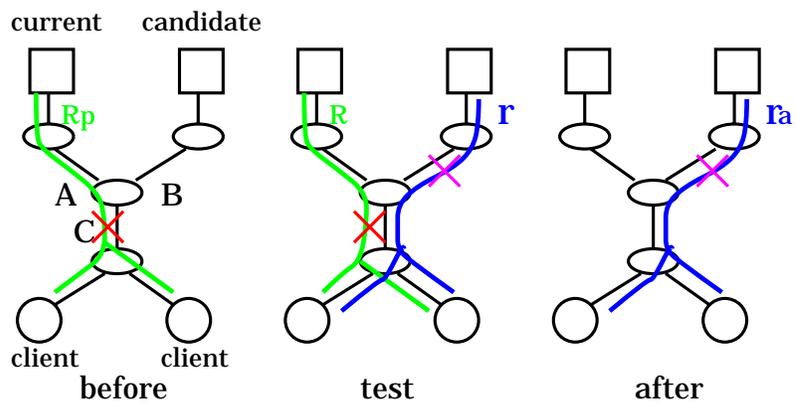


図 28: 移動テストによるボトルネックの変化 (パターン 16)

表 2: ボトルネックリンクの位置と動画データ配信レートの変化の関係

			候補ツリー			
			テスト中のボトルネック			
			独立リンク		共有リンク	
			移動後のボトルネック		移動後のボトルネック	
現ツリー	テスト前のボトルネック	独立リンク	$\begin{cases} r_a > R_p, \text{ if } r > R \\ r_a \leq R_p, \text{ if } r \leq R \end{cases}$	N/A	$\underline{r_a > r > R = R_p}$	$\underline{r_a > r > R = R_p}$
		共有リンク	$R_p > R > r = r_a$	N/A	$R_p > R = r < r_a$	$R_p > R = r < r_a$
		独立リンク	N/A	N/A	N/A	N/A
		共有リンク	$R_p > R > r = r_a$	N/A	$R_p > r_a > r = R$	$r_a = R_p > R = r$
	共有リンク	独立リンク	N/A	N/A	N/A	N/A
		共有リンク	$R_p > R > r = r_a$	N/A	$R_p > r_a > r = R$	$r_a = R_p > R = r$

以上から，移動テスト前，移動テスト中，マルチキャストグループ移動後における動画データ配信レート R_p ， R ， r および r_a の関係は表 2 のようになる，このうち，マルチキャストグループの移動によって確実に配信レートの向上が期待できるのは $r_a > R_p$ の成立する下線を引いた組み合わせである

現サーバがマルチキャストグループの移動の成否を判定する際に利用可能な情報は，移動テスト前の現ツリーにおける配信レート R_p ，移動テスト中の現ツリーおよび候補ツリーにおけるそれぞれの配信レート R および r である．したがって，予想される状況によらず $r > R$ ，すなわち移動テスト中の候補ツリーでの配信レートが現ツリーのものを上回れば，配信レートの向上が期待できるため，マルチキャストグループを移動する．

5 シミュレーションによる評価

5.1 評価方法

本章では，マルチキャストグループの管理を他のローカルサーバに移し，動画配信サービスの品質向上をはかる，提案手法の有効性をシミュレーションによって評価する．ネットワークの負荷が増大し，クライアントの動画データ受信状態が低下した際に，何の制御も行わずマルチキャスト配信を継続した場合，マルチキャストグループの分割，統合を行った場合，および分割，統合，移動の全てを行った場合を比較する．

図 29 に示すネットワークを対象に，ネットワークシミュレーター ns-2 [14] を用いてシミュレーションを行った．図 29 はあるスタブ・ネットワークを示し，2 台のローカルサーバ S1，S2 が，マルチキャストグループ再構成機能と動画品質調整機能を有し，スタブ・ネットワーク内の動画マルチキャストを行っている．スタブ・ネットワークには 4 台のクライアント C1 ~ C4 が存在し，マルチキャストルータ R1 ~ R4 を介してローカルサーバに接続されている．シミュレーション開始時には，これら 4 台のクライアントは全てローカルサーバ S1 の管理する一つのマルチキャストグループ A に收容され，対応するマルチキャストツリー A が構築されている．クライアントはいずれも最大要求品質を 5 [Mbps] とし，時刻 0 から 200 秒間動画マルチキャスト配信サービスを受ける．スタブ・ネットワーク内には動画配信サービスに参加しないホストが 2 台存在し，ホスト T1 からホスト D1 へ 4 本の TCP コネクションが設定されている．これらコネクションには時刻 30 秒から 100 秒までトラヒックが発生する．したがって，この期間はマルチキャストルータ R1 と R2 を結ぶリンク L1 がボトルネックとなる．なお，マルチキャストルータ R1-R2 間のリンクを L1，R1-R4 間のリンクを L2，R3-R2 間のリンクを L3，R3-R4 間のリンクを L4 と呼び，全てのリンクの容量は 5 [Mbps]，伝搬遅延は 10 [msec] とする．また，ルータの出力ポートには 100 パケットの Drop-Tail バッファが備えられている．

評価に関しては，それぞれのマルチキャストグループへの動画データ配信レートおよびクライアントの受信レート，再構成制御のフィードバックとしてクライアントが算出した TCP-friendly レート，TCP-friendly レート導出に用いる RTT とパケット棄却率．および要求

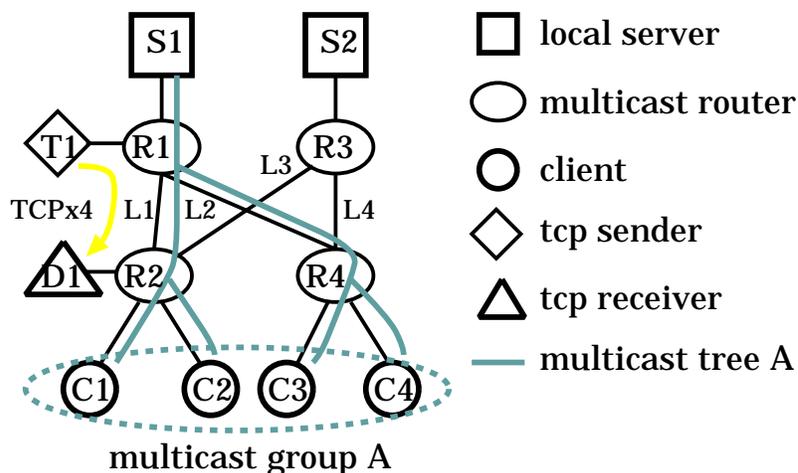


図 29: シミュレーションにおけるトポロジ (初期状態)

の充足度について考慮する．要求の充足度は，どの程度クライアントの要求に対応した動画
 像が配信されたかを示す指標である．クライアント i の充足度 E_i は，TCP-friendly レート
 r_i [bps]，フィードバック間隔 T_i [sec]，および T_i 中に受信した動画像データ量 [bit] から次
 式で与えられる．

$$E_i = \frac{D_i}{r_i \times T_i} \quad (5)$$

E_i が 1 に近づく程，クライアントの要求品質に応じた動画像データが配信されたことを示す．

5.2 マルチキャストグループの再構成を行わない場合の評価結果

通常のマルチキャスト通信と同様に，マルチキャストグループの再構成を行わない場合
 の，ローカルサーバ S1 からの動画像データ配信レートの変化を図 30，それぞれのクライア
 ントにおける動画像データ受信レートの変化を図 31，ローカルサーバ S1 がクライアント
 からのフィードバック情報として得た TCP-friendly レートの変化を図 32，クライアントで
 TCP-friendly レートの算出に用いた平滑化された RTT とパケット棄却率の変化をそれぞれ
 図 33 および図 34 に示す．なお，本評価では，クライアントの最大要求品質を 5 [Mbps] と
 しているため，図 32 に示されるフィードバックされた TCP-friendly レートは 5 [Mbps] で制
 限されている．また，配信された動画像データによるクライアントの要求の充足度の変化を

図 35 に示す。

図 30 より、動画像データ配信開始直後は、TFMCC のメカニズムにより動画像データ配信レートが急激に増大するが、20 秒を経過すると平均レートがリンク容量である 5Mbps に近くなることが分かる。ただし、図中、赤色の実線は初期状態で設定されたマルチキャストグループ A における動画像データ送信レートを、4 本の灰色の破線は 4 つの TCP セッションのスループットをそれぞれ表している。20 秒経過後の送信レートが安定している状態でも、瞬間的に配信レートがリンク容量を超える場合があるが、ルータにおいて超過トラヒックが蓄積されるため、図 33 に示されるとおりバッファリングにより遅延が増大する一方で、図 34 に示されているとおりパケット棄却は発生していない。また、図 31 に示されるとおり、クライアントにおける動画像データ受信レートはいずれもほぼ 5Mbps となっている。

時刻 30 秒で 4 本の TCP セッションが同時にデータ転送を開始すると、リンク L1 がボトルネックとなるため、下流に位置するクライアント C1 および C2 においては、バッファリングによる RTT の増大(図 33)、輻輳によるパケット棄却の発生(図 34)により、図 32 に示されるとおり TCP-friendly レートが急激に低下する。一方、輻輳したリンク L1 を経由しないクライアント C3 および C4 においては TCP トラヒック流入の影響を受けないため、TCP-friendly レートが変化せず、高いレートでの動画像データ配信を受けることができる。しかしながら、本評価においては、ローカルサーバはマルチキャストグループの再構成を行わないため、動画像データ受信状態のよいクライアント C3 および C4 と動画像データ受信状態が悪化したクライアント C1 および C2 は同じマルチキャストグループ A に収容され続ける。TFMCC では、マルチキャストグループでのデータ配信レートは、そのグループに属するクライアントのうちで最も低い TCP-friendly レートを申告したもの、すなわち CLR によって決定されるため、輻輳の影響を受けないクライアント C3 および C4 にも低いレートの動画像データが配信されることとなる。したがって、要求品質に対する配信データ品質によって与えられる要求充足度は、クライアント C3 および C4 において大きく低下することが図 35 から明らかである。一方、クライアント C1 および C2 においては、動画像データの配信レートは低いものの、利用可能な帯域に応じた品質の動画像データが提供されるため、要求充足度は 1 に近い。なお、時刻 35 秒における全クライアントの要求充足度の低下

は輻輳によって引き起こされたパケット棄却により受信データ量が減少したことによる。

時刻 100 秒で TCP セッションが終了すると、リンク L1 における輻輳状態が改善されるため、クライアント C1 および C2 の動画データ受信状態が向上し、マルチキャストグループ A の配信レートも増加する。その結果、クライアント C3 および C4 の要求充足度も高くなる。なお、時刻 105 秒におけるクライアント C1 および C2 の要求充足率の低下は TCP セッションが終了したことにより、ルータにおけるバッファあふれが解消し、クライアント C1、C2 の動画データ受信状態が良くなり、TCP-friendly レートが急激に上昇したのに対し、TFMCC の配信レートは急激なトラヒック変動を避けるためゆるやかに増加していることによる。

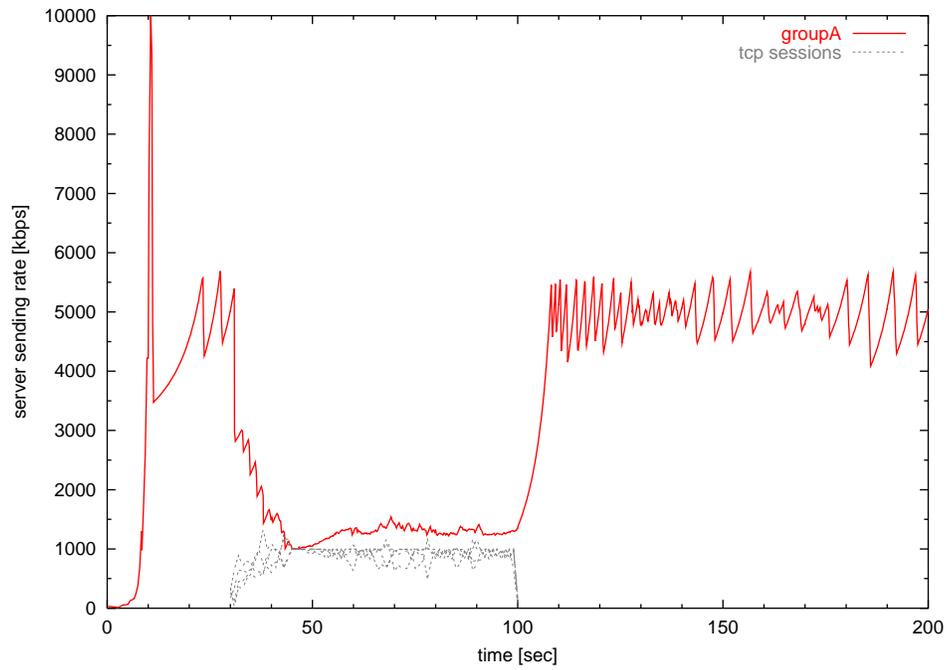


図 30: 配信レートの変化 (再構成なし)

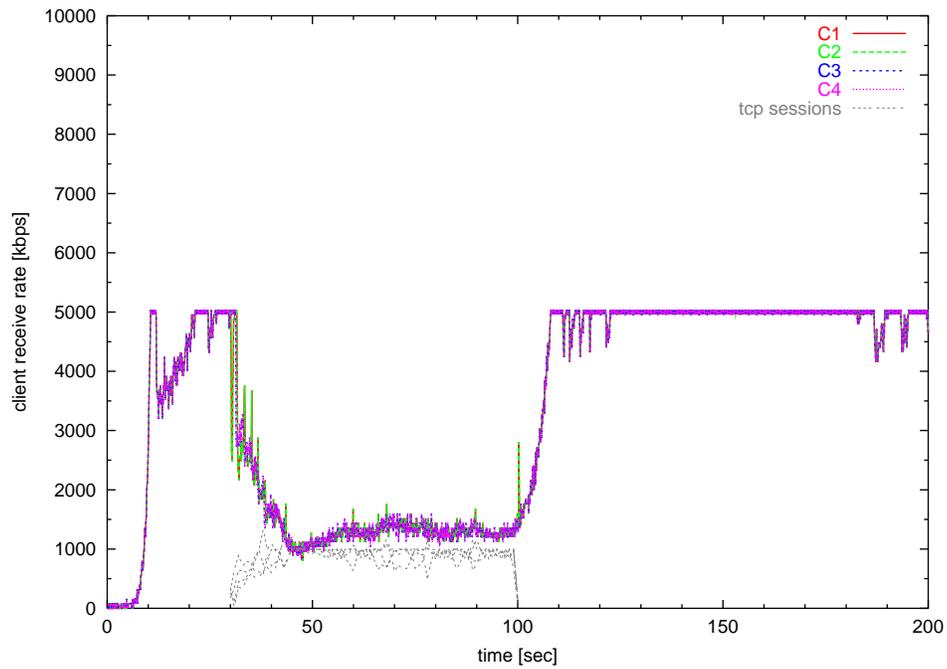


図 31: クライアントの受信レートの変化 (再構成なし)

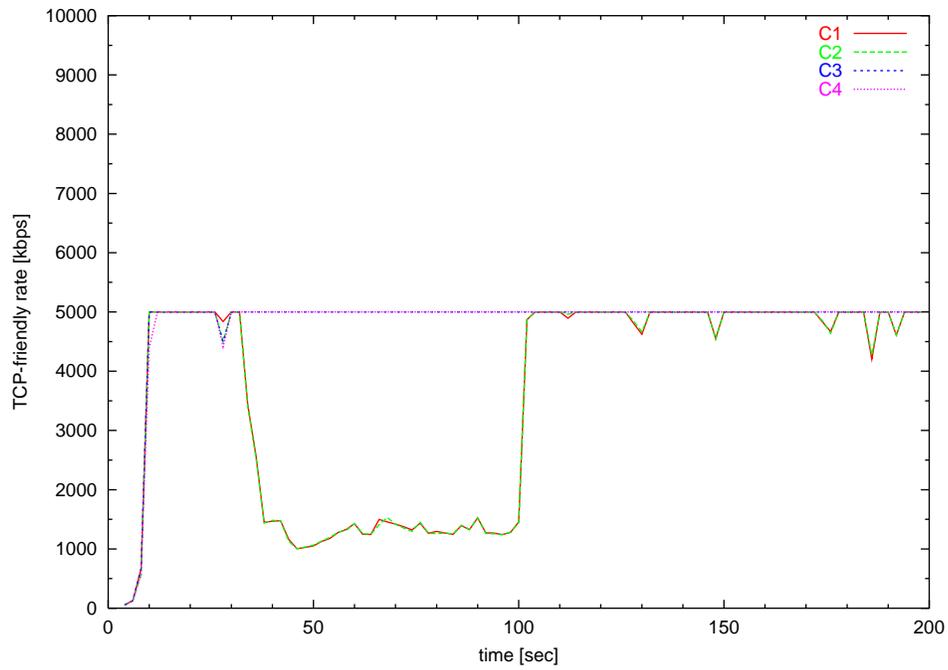


図 32: クライアントのフィードバックした TCP-friendly レート (再構成なし)

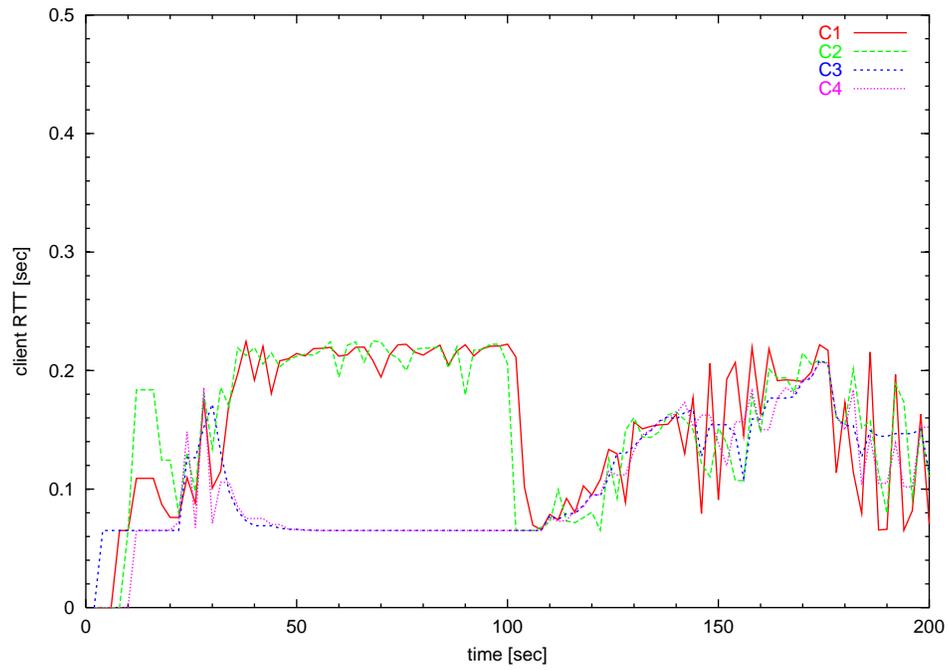


図 33: クライアントの計測した RTT の変化 (再構成なし)

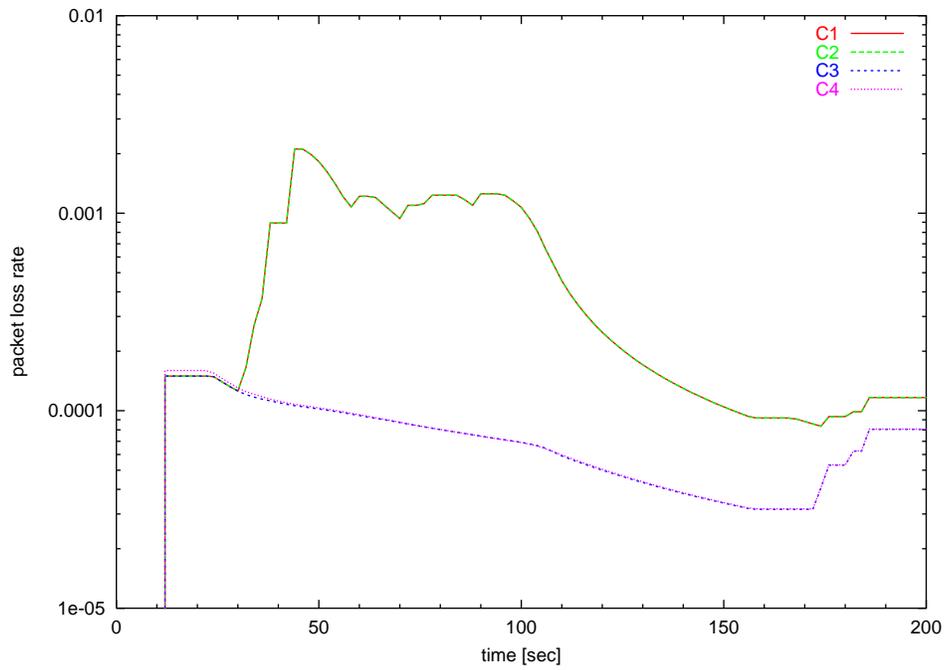


図 34: クライアントの packets 棄却率の変化 (再構成なし)

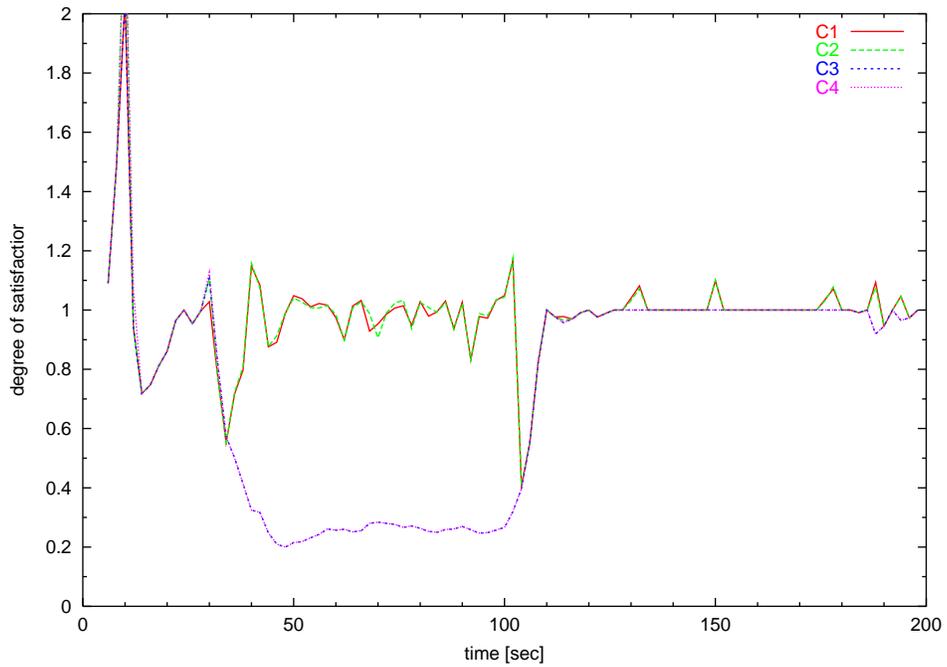


図 35: 要求完了度の変化 (再構成なし)

5.3 マルチキャストグループの分割 / 統合を行う場合の評価結果

次に，文献 [3] で提案され本報告において改良を行った，クライアントの動画データ受信状態のばらつきに応じてマルチキャストグループの分割および統合を行った場合の結果を，5.2 節と同様に図 37～42 に示す．シミュレーションに際しては，提案手法であるマルチキャストグループの分割 / 統合 / 移動の制御間隔が 10 秒，クライアントのフィードバック送信間隔を 2 秒，分割の閾値 T_s を 0.2，統合の閾値 T_m を 0.2 とした．なお，これらパラメータはシミュレーションの結果から定めたものであり，適切な設定法については今後の課題としたい．本シミュレーションにおいては，マルチキャストグループの分割が 2 度発生したため，マルチキャストグループ A を分割して新たに出来たマルチキャストグループをマルチキャストグループ B，マルチキャストグループ A をさらに分割して新たにできたマルチキャストグループをマルチキャストグループ C と呼ぶこととする．なお，分割時のそれぞれの配信レート R_A, R_B, R_C の関係は， $R_A < R_B, R_A < R_C$ となる．

図 39 に示されるとおり，再構成を行わない場合と同様に，時刻 30 秒の TCP トラフィック流入により，クライアント C1 および C2 とクライアント C3 および C4 の TCP-friendly レートに大きな差が生じている．再構成の制御タイミングである時刻 40 秒にフィードバックとして得られているクライアントの TCP-friendly レートは，クライアント C4, C3, C1, C2 の順に 5000kbps, 5000kbps, 2534kbps, 2500kbps であり，変動係数 $C_A = 0.381$ は閾値 $T_s = 0.2$ を上回るため，図 37 に示されるとおり，マルチキャストグループ A の分割が行われている．クライアントの TCP-friendly レートから，最適な分割点としてクライアント C1 と C3 の間が選択され，クライアント C3 と C4 からなるマルチキャストグループ B がマルチキャストグループ A から分割，新たに構成された．したがって，マルチキャストグループ A の CLR はクライアント C2，マルチキャストグループ B の CLR はクライアント C4 となる．この制御タイミングにおけるマルチキャストグループ A の配信レートが 1668kbps であったことから，それぞれのマルチキャストグループにおける初期配信レートは式 (1) から，1112kbps および 556kbps となる (図 37)．分割後のマルチキャストツリーは図 36 のようになる．図のとおり，分割により新たに構築されたマルチキャストツリーはボトルネックリンク L1 を経由しないため，収容されたクライアント C3 および C4 は TCP セッションによる輻輳の影響を

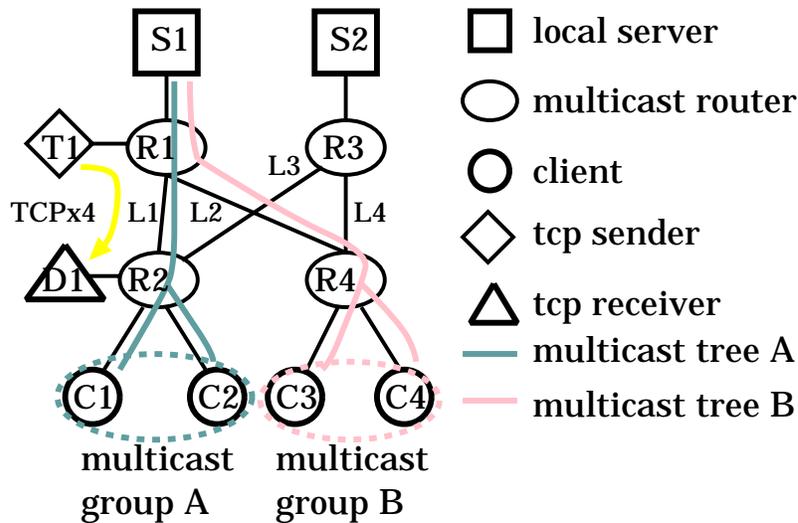


図 36: シミュレーションにおけるトポロジ (分割時)

受けず、高いレートで動画配信を受けていることが図 37 に示されている。その結果、図 42 に示されるとおり、再構成を行わない場合の結果である図 35 と比較して、クライアント C3 および C4 の要求充足度が向上し、ほぼ要求どおりの品質の動画配信が提供されていることが分かる。

時刻 60 秒の制御タイミングにおいては、マルチキャストグループ A がさらに 2 つに分割されていることが図 37 に示されている。これは、クライアント C1 において、前回の制御タイミングである時刻 50 秒から時刻 60 秒の間に、図 40 に示されている瞬間的な RTT の増加が発生したため、RTT とパケット棄却率から算出される TCP-friendly レートがクライアント C2 と比較して低くなったためである。この時、マルチキャストグループ A の変動係数 C_A は 0.252 であった。その結果、クライアント C1 だけからなるマルチキャストグループ A とクライアント C2 だけからなるマルチキャストグループ C が構成された。なお、この状態は一時的なものであったため、次の制御タイミングである時刻 70 秒には、これらのマルチキャストグループ間の変動係数 V_A は 0.069 となり、閾値 $T_m = 0.2$ を下回るため、統合され、クライアント C1 とクライアント C2 の双方がマルチキャストグループ A に收容されることとなる。本報告で改良された分割手法では、式 (1) を用いて分割前の配信レートを 2 つのマルチキャストグループで案分するため、マルチキャストグループ分割によって急激に

ネットワーク負荷が増大することはないが，サーバ負荷や制御パケット送受信のオーバーヘッドを考慮すると，このような瞬間的なネットワーク変動によるグループの分割，再統合は避けるのが望ましい．そのためには，変動係数導出に用いる TCP-friendly レートの平滑化，制御パラメータである閾値 T_s , T_m の適切な設定法や，よりよいばらつきの定量化手法などについて検討しなければならないが，これらについては今後の課題としたい．

TCP セッションにおけるデータ転送が終了した時刻 100 秒からは，リンク L1 の輻輳状態が改善されるため，図 37 に示されるとおり，マルチキャストグループ A の配信レートが向上していく．再構成を行わない場合の図 30 と比較すると，配信レートの回復が緩やかであるが，これは TFMCC メカニズムによりローカルサーバ S1-マルチキャストルータ R1 間の帯域をマルチキャストグループ間で公平に共有しようとするためであり，図 37 から時間も経過するにしたがってマルチキャストグループ間の公平性が高まっていくのが分かる．時刻 150 秒の制御タイミングでは，マルチキャストグループ A, B 間の変動係数 $V_1 = 0.180$ は閾値 $T_m = 0.2$ を下回るため，これらのマルチキャストグループは統合される．マルチキャストグループ統合後は，動画像データ配信レートが増加する．

図 42 に示されているとおり，シミュレーションを通じて，マルチキャストグループの分割，統合やパケット棄却による受信データ量の減少による瞬間的な変動が発生しているが，クライアント C3 および C4 の要求充足率には，図 35 に示されるような長期的な低下が発生していない．したがって，マルチキャストグループの分割により，それぞれのクライアントの動画像データ受信状態に応じた動画像マルチキャスト配信が実現できていると言える．ただし，クライアント C3 および C4 はマルチキャストグループ分割により，輻輳により動画像データ受信状態の悪化したクライアント C1 および C2 の影響を回避でき，動画像データ受信状態が向上しているが，輻輳の影響を直接受けているクライアント C1 および C2 については低品質な動画像しか提供されない．4 章で述べたとおり，本報告で提案しているマルチキャストグループの移動手法は，このような問題を解決するものである．

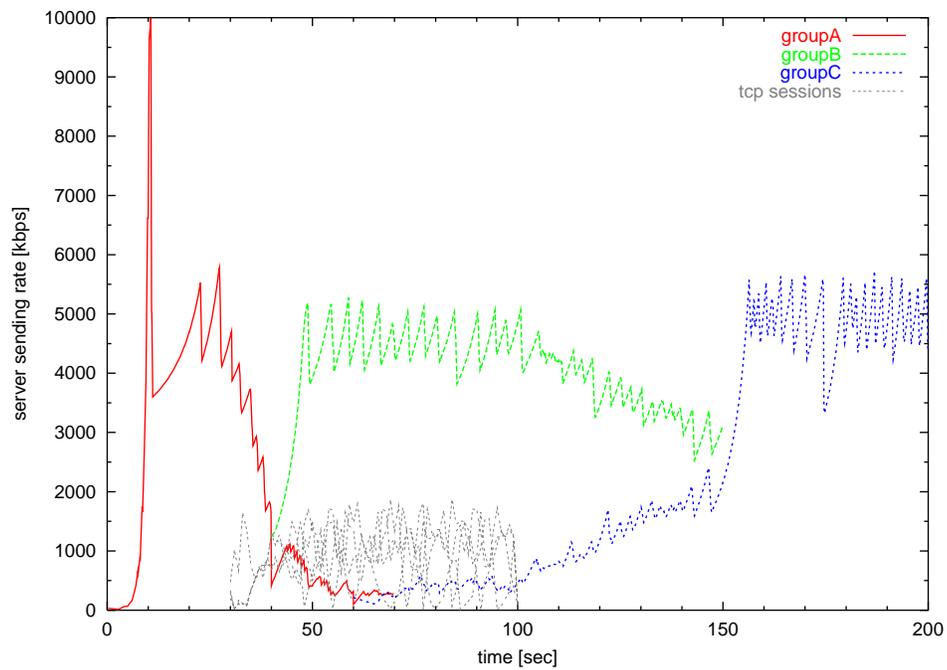


図 37: 配信レートの変化 (split / merge)

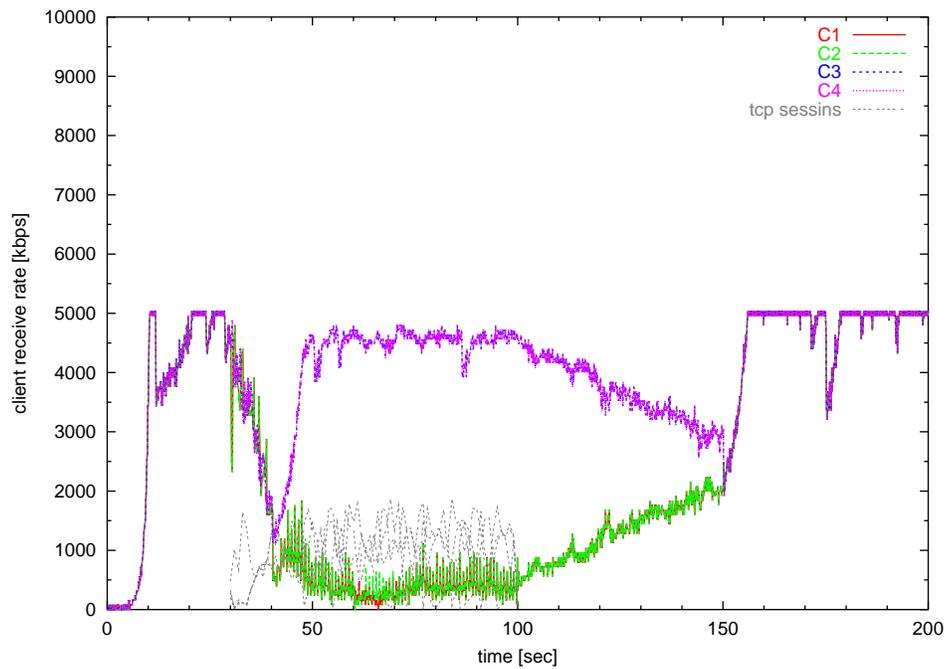


図 38: クライアントの受信レートの変化 (split / merge)

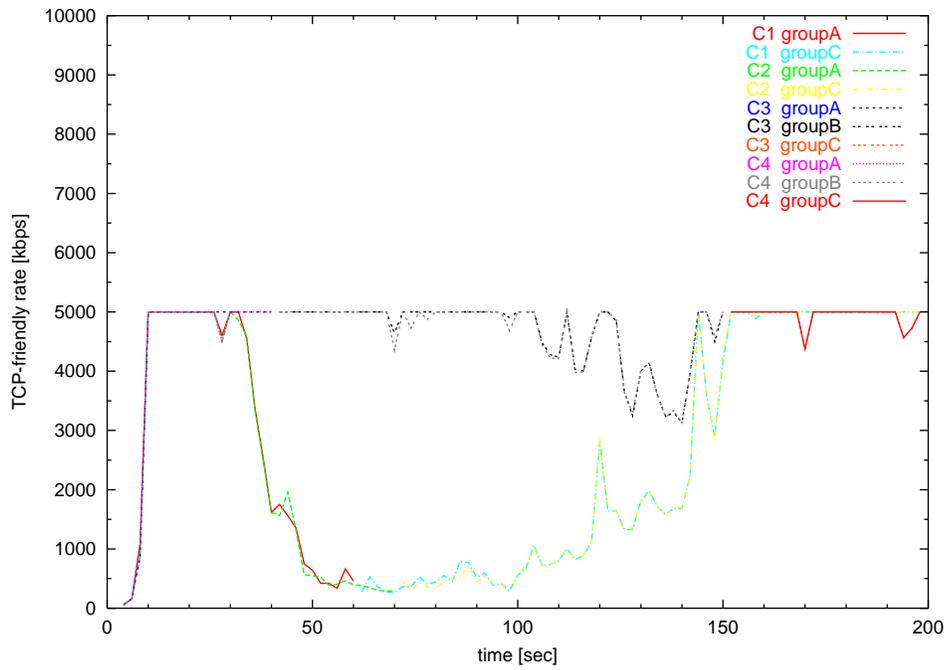


図 39: クライアントのフィードバックした TCP-friendly レート (split / merge)

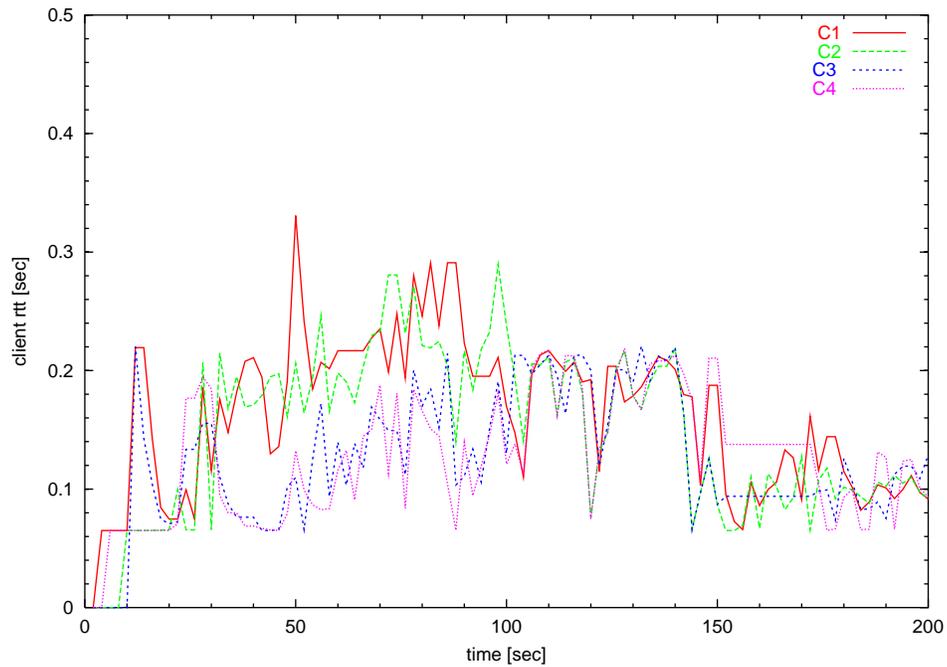


図 40: クライアントの計測した RTT の変化 (split / merge)

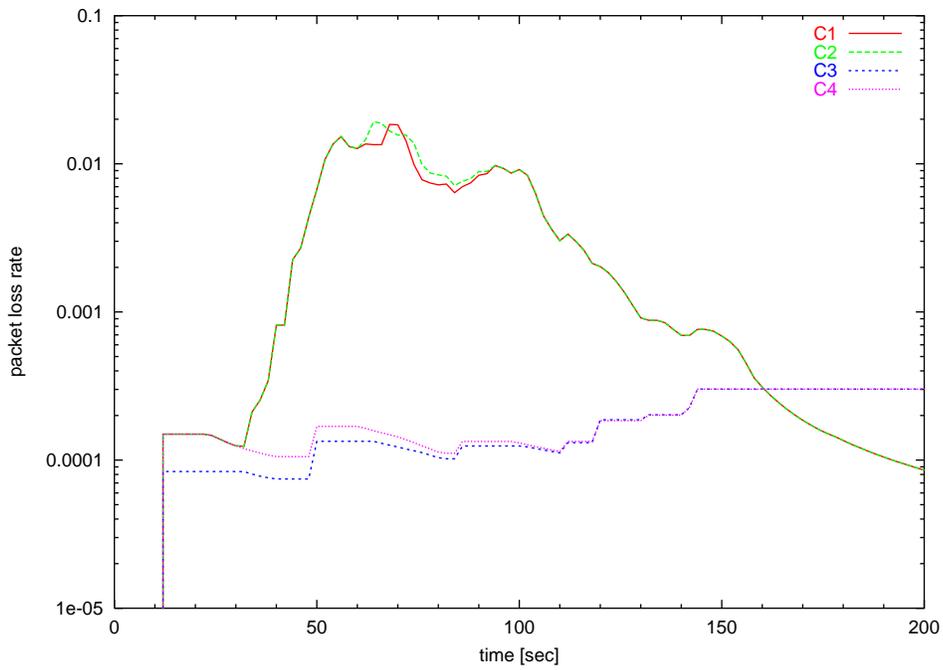


図 41: クライアントのパケット棄却率の変化 (split / merge)

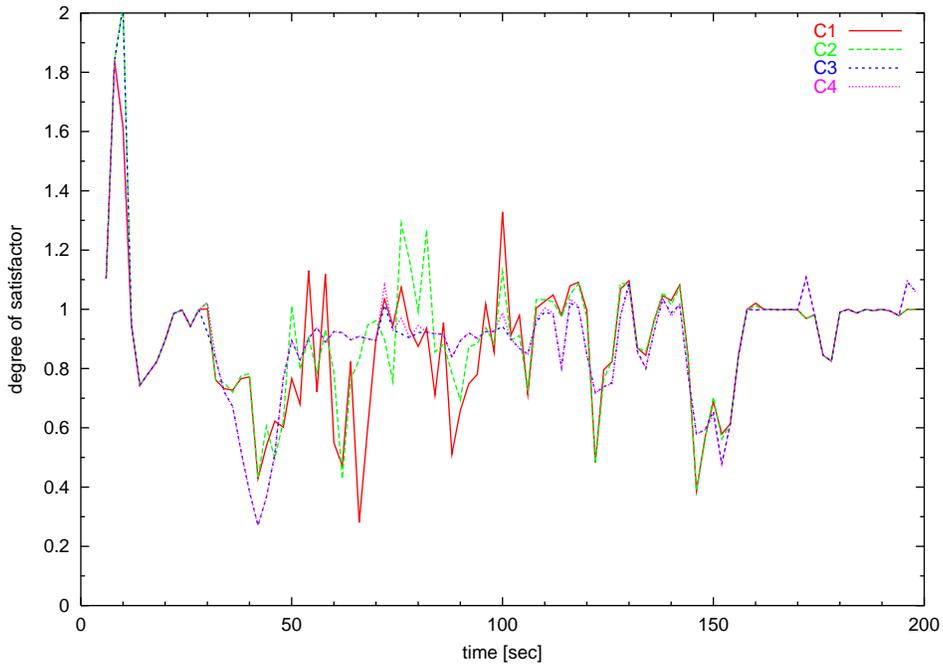


図 42: 要求完了度の変化 (split / merge)

5.4 マルチキャストグループの分割 / 統合 / 移動を行う場合の評価結果

マルチキャストグループの分割 / 統合とともに、マルチキャストグループの移動を行った際のシミュレーション結果を図 45 ~ 50 に示す。図 45 に示されるとおり、TCP セッションの開始により輻輳が発生し、分割 / 統合だけを行う場合と同様に、時刻 40 秒でマルチキャストグループの分割が行われる。その結果、受信状態の悪いクライアント C1 および C2 からなるマルチキャストグループ A と、輻輳の影響を直接受けていないクライアント C3 および C4 からなるマルチキャストグループ B が構成され (図 36)、クライアント C3 および C4 への動画データ配信レートは向上し、図 50 に示されるとおり要求充足率が改善される。

次に、TCP-friendly レートが低く、低品質な動画を提供されているクライアント C1 および C2 については、ローカルサーバ S1 は、時刻 60 秒でこれらのクライアントに動画データを配信しているマルチキャストグループ A の移動を試みる。ローカルサーバ S1 は、同一スタブ・ネットワーク内の別のローカルサーバ S2 に移動テストの準備を要求し、ローカルサーバ S2 は、この要求を受け付け、新たに移動テスト用のマルチキャストグループ C を設定し、準備を完了する。クライアント C1 および C2 は、マルチキャストグループ A での動画データ受信を継続したまま新たに設定されたマルチキャストグループ C に参加する。ローカルサーバ S1 からの要求にしたがってローカルサーバ S2 はマルチキャストグループ C での動画マルチキャスト配信を開始する。これら手順によって構築されたマルチキャストツリーを図 43 に示す。図 45 中の水色の線で示されるとおり、配信レート 0 [bps] から開始された移動テスト用マルチキャスト通信は、TFMCC アルゴリズムにしたがって配信レートを調整していく。

時刻 80 秒にはローカルサーバ S2 は移動テスト用の動画マルチキャスト通信の配信レートが安定したと判断し、ローカルサーバ S1 に配信レート R_c を通知する。ローカルサーバは、マルチキャストグループ A における配信レート $R_A = 287$ [bps] とマルチキャストグループ C における配信レート $R_c = 2959$ [bps] を比較し、4.3 節で述べたアルゴリズムにもとづいて、移動により動画データ受信状態が改善されると判断し、マルチキャストグループの移動を実行する。マルチキャストグループ A における動画配信は停止され、クライアント C1 および C2 はローカルサーバ S2 の管理下に移る。その結果、スタブ・ネットワークに

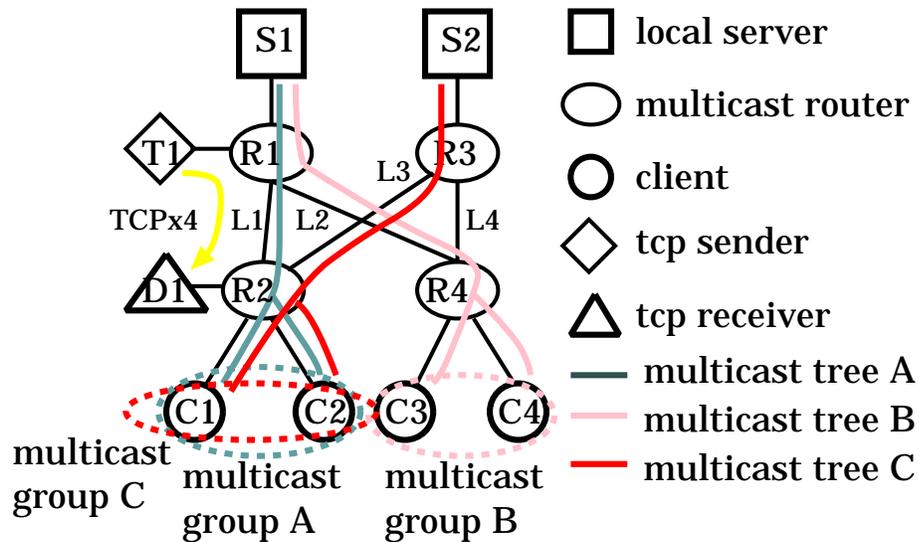


図 43: シミュレーションにおけるトポロジ (移動テスト中)

における動画配信の様子は図 44 のようになる。

時刻 110 秒ではマルチキャストグループ B に対して、また、時刻 160 秒ではマルチキャストグループ C に対して、それぞれ移動テストが実行されるが、移動による動画データ受信状態の向上が見込まれず、移動は行われなかった。

図 50 に示されるクライアントの要求充足率について、マルチキャストグループの分割/統合だけを行う場合 (図 42) と比較すると、移動を行わない場合には分割や統合による一時的な要求充足率の低下が認められるものの、平均的な要求充足率はそれほど変わらない。しかしながら、図 45 に示されるとおり、図 37 と比較して、クライアント C1 および C2 に対する動画データ配信レートは非常に高く、クライアント C3 および C4 への配信レートも高く保たれている。したがって、マルチキャストグループの移動を行うことにより、クライアントの要求どおりで高品質な動画マルチキャスト配信が実現されると言える。

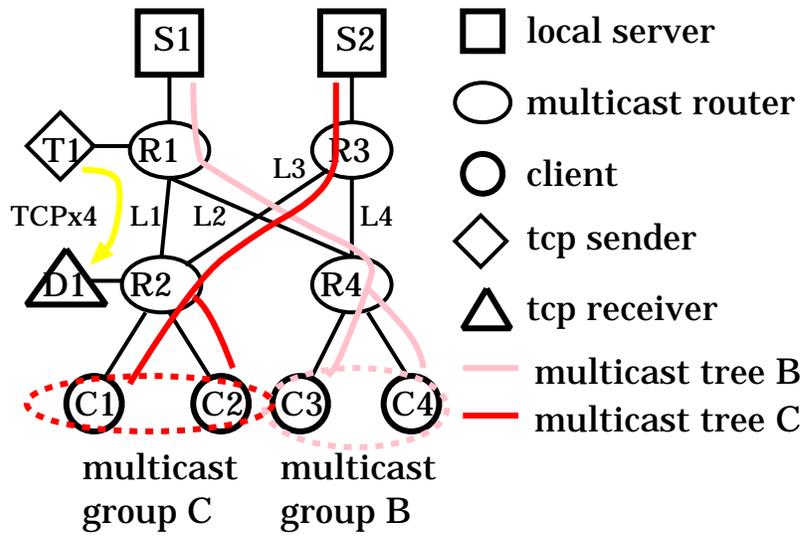


図 44: シミュレーションにおけるトポロジ (移動後)

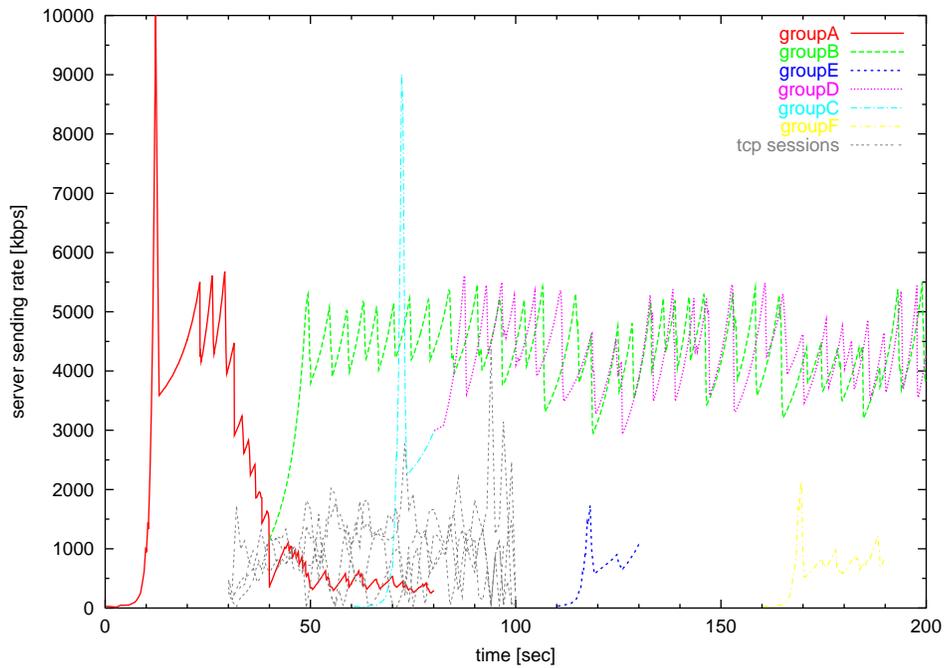


図 45: 配信レートの変化 (split / merge / move)

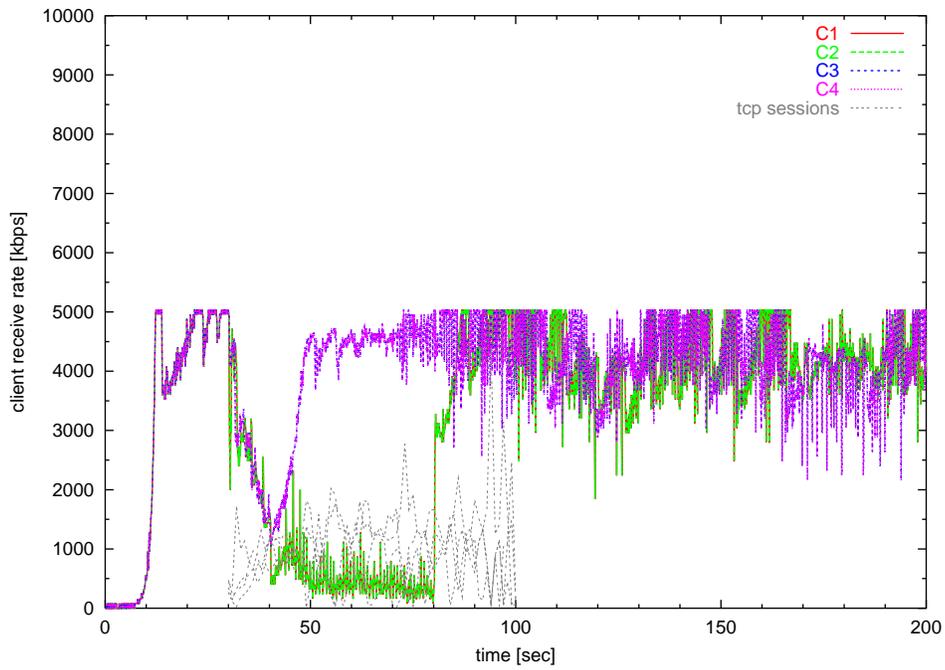


図 46: クライアントの受信レートの変化 (split / merge / move)

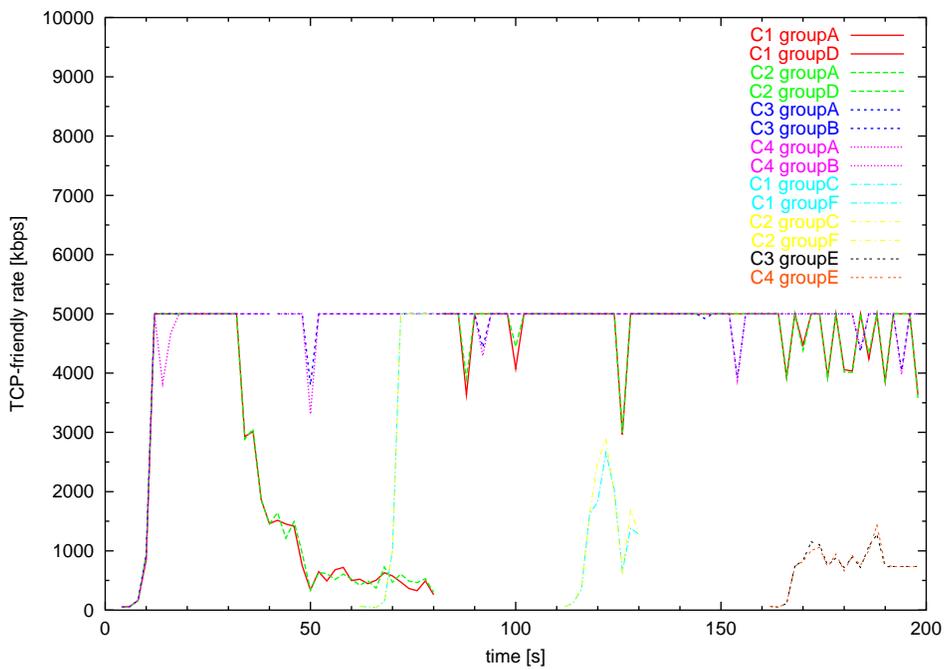


図 47: クライアントのフィードバックした TCP-friendly レート (split / merge / move)

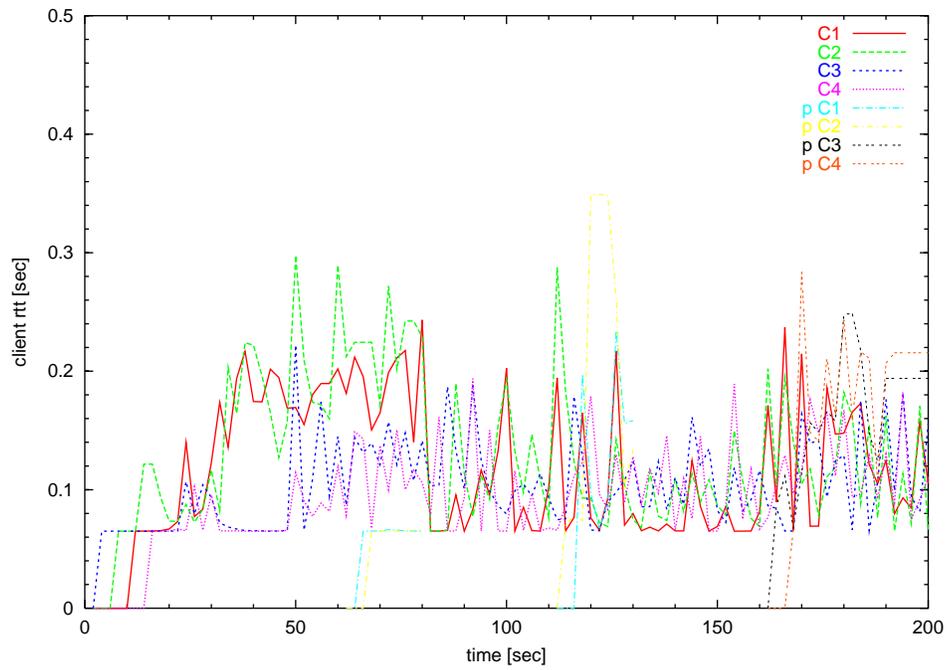


図 48: クライアントの計測しん RTT の変化 (split / merge / move)

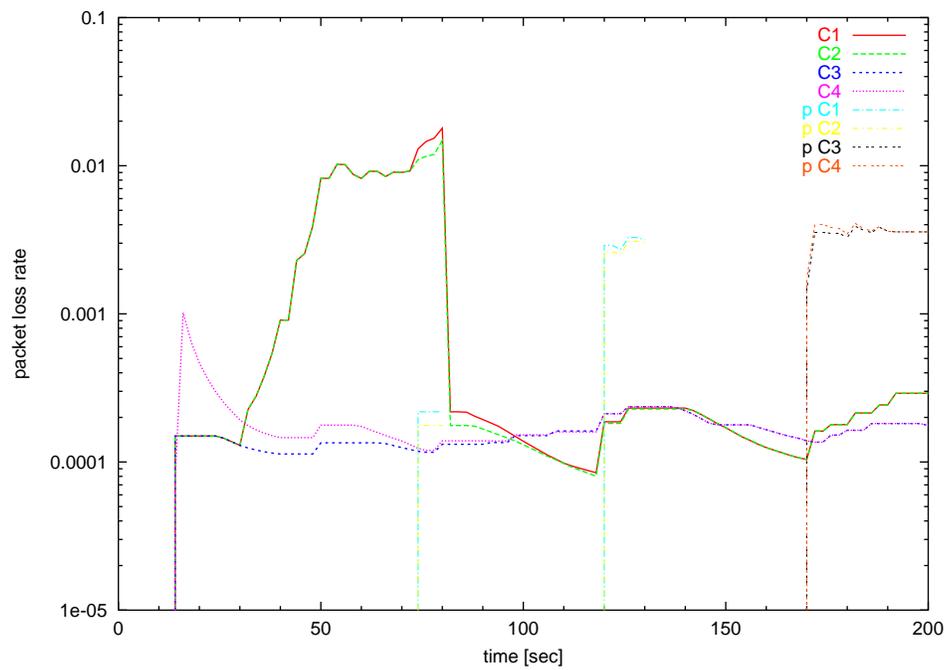


図 49: クライアントのパケット棄却率の変化 (split / merge / move)

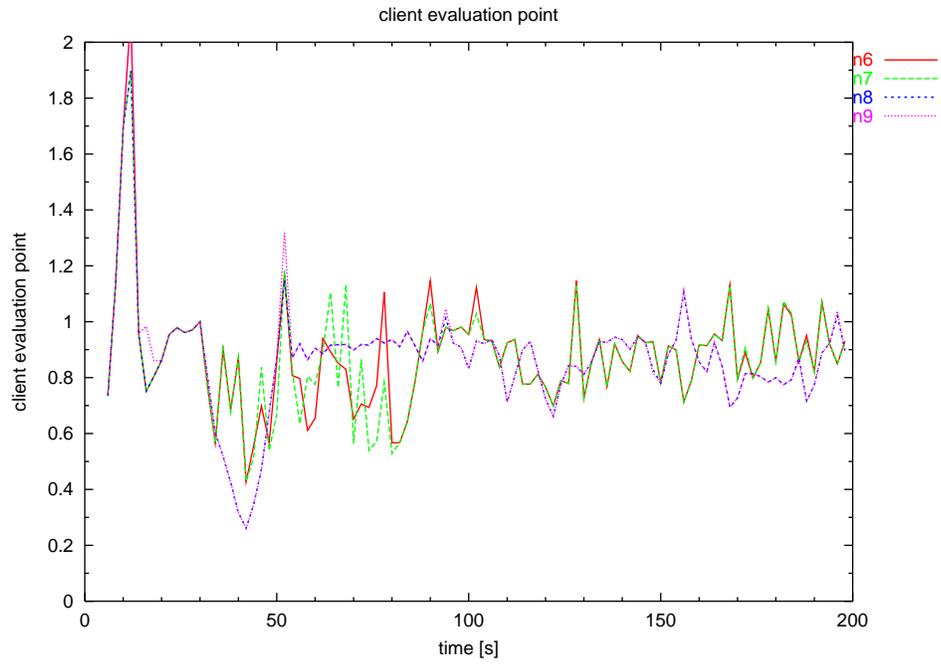


図 50: 要求完了度の変化 (split / merge / move)

6 おわりに

本報告では、実時間動画像マルチキャスト配信サービスにおいて、ネットワーク環境やシステム環境、またはユーザの好みなどによりさまざまに異なるクライアントの要求品質や動画像データ受信状態の変化に応じて動的にマルチキャストグループを再構成する手法として、我々の研究グループがこれまでに提案したマルチキャストグループの分割、統合に加えて、マルチキャストグループの管理を移す手法を提案した。また、シミュレーションによる評価によって、マルチキャストグループの移動によってマルチキャストツリーのトポロジが変化し、クライアントの動画像データ受信状態が向上することにより、サービス品質が高められることを示した。

一方で、一時的な動画像データ受信状態の変動により不必要な再構成が行われることが分かった。また、本報告では、他のトラヒックの流入やマルチキャストグループの再構成がネットワーク全体に非常に大きな影響を与える小規模なネットワークを対象にシミュレーション評価を行ったが、多数のクライアントが存在し、ネットワークの負荷変動が激しく、あるいは緩やかに変動するような大規模なネットワークにおいても、マルチキャストグループの再構成が適切に動作するよう、制御パラメータの設定法を含めたより効果的な再構成手法を今後検討する必要がある。

謝辞

本報告を終えるにあたり，御指導，御教授を頂いた村田正幸教授に心より感謝申し上げます。また，本報告において終始御指導頂いた若宮直紀助教授に深く感謝致します。

並びに，日頃より適切な助言を頂いた宮原秀夫教授，大阪府立看護大学の菅野雅嗣助教授，大阪大学サイバーメディアセンターの馬場健一助教授，大崎博之助教授，村田研究室の長谷川剛助教授，大阪市立大学の阿多信吾助手，大阪大学経済学部の荒川伸一助手，宮原研究室の牧一之進助手，日立制作所の上岡功司氏に心から感謝致します。

最後に，本報告において直接御指導と適切な助言を頂いた Hector Akamine 氏，笹部昌弘氏，山田達也氏，谷口義明氏，侍建港氏，並びに御協力頂いた村田研究室および宮原研究室の皆様心から御礼申し上げます。

参考文献

- [1] Steven McCanne, Van Jacobson and Martin Vetterli, “Receiver-driven Layered Multicast,” *Proc. ACM SIGCOMM*, vol. 26,4, pp. 117–130, August 1996.
- [2] Kentarou Fukuda, Naoki Wakamiya, Masayuki Murata and Hideo Miyahara, “Real-time Video Multicast with Hybrid Hierarchical Video Coding in Heterogeneous Network and Client Environments,” in *Proceedings of Second IRIP/IEEE International Conference on Management of Multimedia Networks and Service 98’ (MMNS 98)*, October 1998.
- [3] Héctor Akamine, Naoki Wakamiya, and Hideo Miyahara, “Congestion-adaptive video multicast in an active network,” in *Technical Report of the IEICE (NS 2002-59)*, pp. 27–30, June 2002.
- [4] Jörg Widmer and Mark Handley, “TCP-friendly multicast congestion control (TFMCC): protocol specification,” *IETF Internet Draft (draft-ietf-rmt-bb-tfmcc-01.txt)*, November 2002.
- [5] S.F.Bush and A.B.Kulkarni, *Active Networks and Active Network Management*. Kluwer Academic/Plenum Publishers, 2001.
- [6] Mark Handley and Van Jacobson, “SAP: Session Anounce Protocol,” *IETF Internet Draft* (<http://www.ietf.org/rfc/rfc2974.txt>), October 2000.
- [7] Mark Handley and Van Jacobson, “SDP: Session Description Protocol,” *IETF Internet Draft* (<http://www.ietf.org/rfc/rfc2327.txt>), April 1998.
- [8] Jonathan M. Smith and Kenneth L. Calvert and Sandra L. Murphy and Hilarie K. Orman and Larry L. Peterson, “Activating Networks: A Progress Report,” *IEEE Computer*, vol. 32, no. 4, pp. 32–41, 1999.

- [9] Héctor Akamine, Naoki Wakamiya, and Hideo Miyahara, “An Approach for Heterogeneous Video Multicast Using Active Networking,” in *Proceedings of Second International Working conference on Active Networks, Musashino*, pp. 157–170, October 2000.
- [10] Jörg Widmer and Mark Handley, “Extending equation-based congestion control to multicast applications,” in *Proceedings of SIGCOMM '01*, August 2001.
- [11] Injong Rhee, Nallathambi Ballaguru and George N. Rouskas, “MTCP: Scalable TCP-like Congestion Control for Reliable Multicast,” Tech. Rep. TR-98-01, North Carolina State University, Department of Computer Science, February 1998.
- [12] Sally Floyd, Mark Handley, Jitendra Padhye and Jörg Widmer, “Equation-Based Congestion Control for Unicast Applications,” in *Proceedings of SIGCOMM 2000*, August 2000.
- [13] Jitendra Padhye and Victor Firoiu and Don Towsley and Jim Krusoe, “Modeling TCP Throughput: A Simple Model and its Empirical Validation,” in *Proceedings of ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication*, pp. 303–314, September 1998.
- [14] *The Network Simulator*. <http://www.isi.edu/nsnam/ns/>.