

修士学位論文

題目

P2P ネットワーク上で実時間ストリーミングを実現するための
分散制御プロトコルの提案

指導教官

東野 輝夫

報告者

中村 嘉隆

平成 16 年 2 月 13 日

大阪大学 大学院情報科学研究科

情報ネットワーク学専攻

P2P ネットワーク上で実時間ストリーミングを実現するための
分散制御プロトコルの提案

中村 嘉隆

内容梗概

近年，クライアント同士がサーバを介さずに自律分散的に働くことによってサービスを提供する P2P (Peer-to-Peer) と呼ばれる接続方式が注目されている．この P2P 方式を用いて実現するマルチキャストは，アプリケーション層マルチキャスト (ALM) と呼ばれ，従来の IP マルチキャストと比べ，プロトコル設計が自由であるなどの利点がある．

アプリケーション層マルチキャストに関する従来の研究では，ネットワーク資源の利用コストやエンド間遅延を考慮した経路木の生成，エンドホストの非安定性への対応を課題として扱ってきた．しかし，電子会議等，実時間ストリーミングを必要とするグループ通信に利用するには，エンドホスト付近の帯域制約や受信したいメディアの偏りといった問題がある．本研究が提案する QoS を考慮した ALM プロトコル Emma/QoS では，アプリケーション層で構築された P2P ネットワーク上での複数の映像のストリーミング配信における QoS 制御を実現する．Emma/QoS では，各エンドホストが映像ストリームをマルチキャスト転送する際に映像の解像度を下げるなどのフィルタリングが可能であると仮定する．この仮定のもとで，映像ストリームの配信レートを各エンドホストが協調して動的に調整し，限られた P2P ネットワーク帯域のもとでユーザのサービス要求をより満足するストリーミング配信を実現する．シミュレーション実験により，従来の代表的な ALM プロトコル Narada などとの比較を行った結果，より高い満足度を達成していることが分かった．

主な用語

P2P ネットワーク，アプリケーション層マルチキャスト，オーバーレイネットワーク，ユーザプリファレンス

目次

1	まえがき	4
2	研究背景	7
2.1	対象とするアプリケーション	7
2.2	アプリケーション層マルチキャスト	7
2.3	関連研究	8
3	Emma/QoS の概要	10
3.1	オーバレイネットワークの構築	10
3.2	経路木の構築	11
3.3	ストリーム要求のアドミSSION制御	12
3.4	ノードの離脱時の回復処理	14
3.5	プリファレンスの設定	14
3.6	異なる複数の受信要求による競合の回避	15
3.7	ポリシー管理	15
4	ストリーム配信時の QoS 制御	16
4.1	ユーザ利得及びユーザ損失関数	16
4.2	ユーザ利得及びユーザ損失に基づいたアドミSSION制御	18
4.2.1	ユーザ損失の定期的な公告	18
4.2.2	受信要求の処理手順	18
4.3	フィルタリングが不可能なノードへの対応	21
4.4	利用帯域変動への対応	21
5	性能評価	22
5.1	測定項目	22
5.2	シミュレーションシナリオ	23
5.3	リンクストレスとパスストレッチの評価	24
5.4	ツリーストレスの評価	26
5.5	ユーザ要求満足率の評価	26
5.6	ユーザ利得値の分布の測定	27
5.7	グループサイズにおけるユーザ利得値総和の測定	28
6	あとがき	29
	謝辞	30

1 まえがき

高速ネットワークの普及は、近い将来、インターネットを用いたグループ通信アプリケーションの普及をもたらすと予想される。このグループ通信は、主にグループ内のメンバへの同報通信からなり、ビデオチャット等では実時間ストリーミングの配信も要求される。しかし、そのようなグループ通信が多数同時に行われる場合、従来のクライアント・サーバ型通信によって、全グループのデータ交換処理を限られた数のサーバが行うことは、サーバ側のネットワーク資源及びコネクション数の観点から現実的ではない。このクライアント・サーバ型通信形態と対になる概念として、近年のネットワークの広帯域化を背景として、分散型ファイル共有アプリケーションに代表される Peer-to-Peer (P2P) という新たな通信形態が注目を集めている [4, 5, 6, 7]。P2P では、全てのホストが対等な関係であり、ホスト同士がサーバを介さず直接通信を行っている。

一方、グループ通信において大規模なファイル配信やストリーム配信などを行うアプリケーションを実現するためには、同一のデータを同時に複数のホストに配信する技術が必要となっている。この 1 対多型通信形態はマルチキャストと呼ばれる。マルチキャスト技術の 1 つとして、IP マルチキャストがある [8, 9, 10]。IP マルチキャストは現在ほとんどのルータに実装されており、利用は可能である。しかし、ドメイン間ルーティング、マルチキャストアドレスの割り当て、グループ管理などが困難である点、スケーラブルなマルチキャストルーティングプロトコルでない点、また、マルチキャスト対応ルータへのサポートが必要であるなど、現状では利用に問題が多い。また、これらを解決して用いるためには、複雑なプロトコルが必要となり、現実的ではない。

そこで、P2P ネットワークを用いることによって、アプリケーション層でマルチキャスト機能を実現するアプリケーション層マルチキャスト（以下 ALM と呼ぶ）を利用することが考えられる。ALM は、エンドホスト間の P2P リンクによって構成される仮想ネットワーク（P2P ネットワーク）上で、エンドホスト自身がマルチキャスト転送を行う通信形態である。ALM では単一の実リンク上での同一パケットの重複や、パケットが複数の P2P リンクを経由することによる遅延の増大などのオーバーヘッドが生じる。一方で、特定のインフラストラクチャに依存せず、既存のインターネット上で実現可能であり、ユニキャストと

比較して資源の利用効率が高いなど利点も多い。

従来の ALM の研究 [11, 12, 13, 14, 15, 21] では、ALM のオーバヘッドや不安定性の解消、スケーラビリティの追求など様々な設計目標に基づいたプロトコルが提案されてきている。一方、複数の映像が同時並行的にストリーミング配信されるアプリケーション（例えばビデオ会議システム）では、オーバレイリンク上の帯域や、ホストの転送能力などオーバレイネットワーク上の限られた資源が競合する場合があるが、その場合についての効率的な制御法は考慮されていない。例えば、文献 [14, 15] は大容量ファイル配信アプリケーション向けに、帯域利用効率のよい経路木構築が目標となっている。また、ALMI[13] では、少人数向けの低遅延マルチキャスト木の構築を目標としている。一方、Narada[11, 12] では、ビデオ会議向けの ALM を提案している。この場合も、同時に複数のストリームを表示するような状況は想定しておらず、経路重複を回避するような経路木を構築し、利用できる帯域と配信ストリームの帯域に差がある場合は場合は該当リンクをもつノードでパケットを落としてレートを下げるといった制御を行っているのみである。

これに対し、私の所属する研究グループでも ALM プロトコル Emma (End-user Multicast for Multi-party Applications) を提案してきた [23]。Emma では、オーバレイネットワーク上に配信木を構築し、各ホストが映像ストリームに対して指定する優先度要求（プリファレンス）に基づき、限られたオーバレイリンク容量や各ホストのストリーム転送能力のもとで、どのデータストリームを優先的に配信するかを、動的かつ分散で制御する。例えばビデオ会議システムでは、会議の参加者は中心となる話者達の映像に対してはより高いプリファレンスを指定し、他の聴衆や会場全体などの映像には比較的低いプリファレンスを指定する。これに対し、複数の映像ストリームがオーバレイネットワークの帯域を競合した場合、プリファレンスがより低い映像ストリームの配信を強制的に停止し、より高いプリファレンスをもつ映像ストリームの配信品質を保証する。

一方、最近の計算機の著しい能力向上により、映像の配信を強制的に停止するのではなく、ビットレートを実時間で下げるフィルタリング処理 [24] をエンドホストで行うことも充分可能になってきた。したがって、各エンドホストがこのような処理能力を持つと仮定し、多くの映像をなるべく高品質で配信する、より柔軟な制御を行うことで、ユーザの満足度を向

上させる高品質のサービスを提供できる可能性があると考えられる。

本研究では、複数の映像ストリーム配信に対する QoS 制御機能を提供する ALM プロトコル Emma/QoS を提案する。Emma/QoS では、各エンドホストが映像を他のエンドホストへマルチキャスト転送する際にその転送レートを調整する機能を持つと仮定する。そのもとで、既存の映像ストリームの転送レートをエンドホスト間で協調して調整することで新たな映像ストリームの受信要求を受け入れるアドミッション制御を行う。シミュレーション実験により、Emma/QoS が従来の ALM プロトコルと比較し、高いユーザ満足度を実現していることなどが確認できた。

なお、P2P ネットワークにおけるフィルタリングの研究は、文献 [26] などで行われている。文献 [26] では、トランスコーディングやフィルタリングのようなサービスを行ういくつかのプロキシノードが P2P ネットワークに配置されており、帯域制約、トランスコーディングのコスト、品質要求などを満たしながらユーザが要求するサービスを提供するための最適な P2P 経路を見つけるアルゴリズムが提案されている。この意味で文献 [26] の手法は静的 QoS ルーティングに比較的近いといえる。しかし、この手法ではサービス要求が P2P ネットワークのリソースを競合する場合については考慮されていない。一方、Emma/QoS は、そのような競合に対し、ユーザの利得を全体としてなるべく大きくするよう、動的に競合を解消する制御を行うため、ユーザの参加離脱が比較的多い P2P ネットワークのユーザのサービス要求に対し、動的にかつより柔軟に対応できると言える。

本論文は以下のように構成する。2 章では研究の背景を述べる。次に、3 章では Emma/QoS の概要を述べ、Emma/QoS におけるアドミッション制御については 4 章で説明する。5 章で性能評価の結果と考察を与え、6 章でまとめと今後の課題を述べる。

2 研究背景

2.1 対象とするアプリケーション

本研究では、ビデオ会議等、実時間でのデータ交換を伴うコミュニケーションシステムを対象アプリケーションとしている。このようなアプリケーションでは、アプリケーションの参加者が他の参加者に映像ストリームを配信し、他の参加者からの映像ストリームを受信する。従って、送受信される映像ストリームには帯域、遅延双方の保証が必要となる。これらの映像ストリームはなるべく多くの参加者に配信されるのが望ましいが、ネットワーク資源や計算機資源の制約から、他の全ての参加者に配信することは難しく、ストリーム配信の制御を行う必要がある。

2.2 アプリケーション層マルチキャスト

2.1 で述べたようなアプリケーションを実現するための現実的な手段としては、アプリケーション層マルチキャストが挙げられる。ALM はアプリケーション層でマルチキャストを実現する技術であり、各ホストがパケットの複製やマルチキャストルーティング、グループ管理などを行うものである。また、IP マルチキャストと異なり、ALM では全ての通信をユニキャストで行う。そのため、信頼性や輻輳制御、レート制御などトランスポートプロトコルの制御機能が利用可能であるという利点がある。オーバーレイネットワーク構築時に適当なメトリックを選ぶことにより、アプリケーションに特化した ALM を構築できる。

ALM の例を、図 1 に挙げる。図 1(a) では、ホスト *B* がホスト *A* からパケットを受け取り、複製してホスト *C* とホスト *D* に転送している（図 1(b) はマルチキャスト配信木を表す）。図 1(a) のホスト *B* 付近では同一のパケットが 3 回配送されている。

このように、データ配送時に単一の物理リンク上で同一パケットの重複が発生する。また各ホスト間は end-to-end の P2P リンクであるために、ホスト付近での帯域制約が生じる、リンクが帯域変動を受けやすい、複数の P2P リンクを経由することが避けられないために遅延の増大を招く等の問題も生じる。従って、これらの問題を解決できるようなプロトコルを設計することが必要となる。

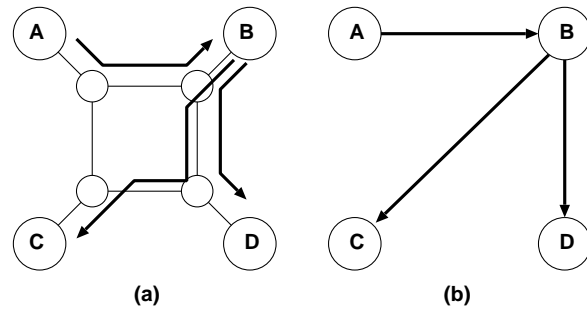


図 1: アプリケーションレベルマルチキャスト

2.3 関連研究

近年，ALM に関して異なる設計目標に基づく多くの研究がなされている．

これらのうち Overcast[14]，NICE[21] などは，メッシュ型のオーバーレイネットワークを作らず，直接マルチキャスト木を構築する tree-first アプローチをとる．Overcast では大容量のファイル配信アプリケーションの実現を目指して，帯域利用効率を考慮して共有経路木構築する方法を提案している．また NICE では多数の受信者を持つアプリケーション向けに制御オーバーヘッドの低減を目指して，階層構造の共有経路木を構築している．一方，HBM[15] はランデヴポイントを用いてマルチキャスト木を構築し，マルチキャスト経路木の安定性のため，バックアップリンクをどのように構成するかの方法論を提案している．これらの tree-first アプローチを取った場合，マルチキャスト木構築までのプロトコルは比較的容易になるが，ループ検出やツリー分断時の再構築時間のオーバーヘッドが大きくなる．

Narada[11, 12]，ALMI[13] はメッシュ型のオーバーレイネットワークを構築し，その上にマルチキャスト木を構築する．ALMI では低遅延での配信を目指し，総遅延が最小である被覆木を構築する．このとき，セッションの管理者が全てのグループメンバの情報を中央集権的に管理するため，最適なマルチキャスト木を構築することが可能であるが，大人数ではスケールしないという問題がある．Narada[11, 12] はビデオ会議などのアプリケーション向けに設計された ALM である．文献 [11] では，ping で得られた遅延のみをメトリックとしてオーバーレイネットワークを構築し，その上で DVMRP (Distance Vector Multicast Routing Protocol) 方式の最短経路木を構築している．これを会議システム向けに改良し，

帯域と遅延双方を考慮して shortest widest path tree を構築しているのが、文献 [12] である。これらは各経路木を分散制御で構築・維持しているために複数データストリーム間の経路重複をある程度静的に回避できる。しかし、オーバーレイネットワーク上の限られた資源を競合する際、その制御を動的に行うことはできない。

本研究が対象とするアプリケーションでは、

- 多対多のストリーム配信が必要である
- 配信には広帯域、低遅延のリンクが必要である
- グループメンバーの参加、離脱は頻繁に発生する
- オーバーレイネットワーク上の資源を適当に制御することにより資源競合を回避する

の要求を満たす必要がある。このため、Emma/QoS では、メッシュライクなオーバーレイネットワークを構築し、その上に経路重複を考慮しながら遅延をメトリックとした最短経路木を構築する方法を用いた。

3 Emma/QoS の概要

本章では, Emma/QoS の基本動作について述べる. Emma/QoS はオーバーレイネットワークの構成要素である各エンドホスト (以下ノード) の振舞いに基づき以下の動作を行う.

1. ノードのアプリケーション参加要求に対し, そのノードをオーバーレイネットワークに参加させる.
2. 参加したノードが自身の映像データを他ノードにストリーミング配信するためのマルチキャスト経路木 (被覆木) をオーバーレイネットワーク上に構成する. また, ストリーミングを受信するため, 既参加ノードの経路木にそのノードに参加させる.
3. ノードの映像ストリーム (以下, ストリーム) 受信要求に対し, 要求通りにデータを配信してよいかを判断するアドミSSION制御を行う. この際, 可能な限り要求を受け入れるために既存のストリームのレート制御を行う.
4. ノードの離脱により切断されたストリームを可能な限り回復する.

Emma/QoS では, 上記の (2) において異なる経路木同士がなるべくリンクを共有しないような経路制御, (3) 及び (4) において, ストリームに指定されたユーザ利得と呼ばれる評価値をなるべく大きくするようなアドミSSION制御及び回復制御をそれぞれ行う. これらの制御を Emma/QoS における QoS 制御と総称するが, 本論文では主に (3) のアドミSSION処理に着目し, その詳細について 4 章で述べる.

3.1 オーバレイネットワークの構築

Emma/QoS ではあるノード集合によるアプリケーションの実行単位 (例えば会議アプリケーションにおける一つの会議) をセッションとよび, セッションごと独立にオーバーレイネットワークを構築する.

セッションに参加するノードは, まず始めに既参加ノードのプロファイル (IP アドレス, ポート番号, ノード ID 等) を, それらを管理するサーバ (以下ロビーサーバと呼ぶ) から取得する. 新規参加ノードは, 取得したプロファイルを用いて既参加ノードとの遅延あるい

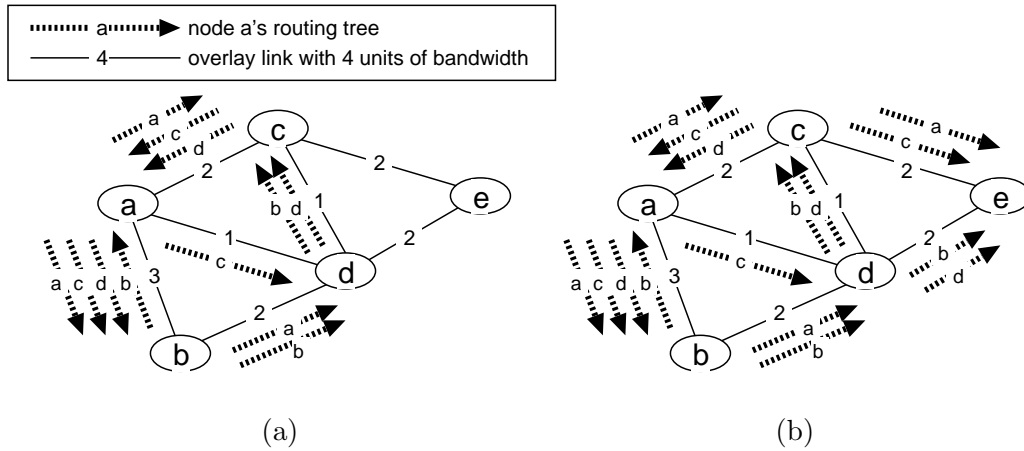


図 2: ノードの参加に伴う経路木の拡張例

は RTT を計測し，計測した遅延がなるべく小さいいくつかのノードと P2P リンク（以下オーバーレイリンクと呼ぶ）を構築する．なお，Emma/QoS では，帯域ユニットと呼ばれる一定量の単位で帯域を扱い，あるオーバーレイリンクで利用可能な帯域ユニット数をオーバーレイリンク容量と呼ぶ．オーバーレイリンク構築の際には構築相手のノードと交渉を行い，自身と相手の計算機資源や LAN の帯域，測定した遅延などからそのオーバーレイリンク容量を決定する．

なお，各ノードはロビーサーバへの定期的な報告を行うことでプロファイルを最新のものに更新する．

3.2 経路木の構築

Emma/QoS が対象とする会議アプリケーションでは，各ユーザが自身の映像などをストリーミングする可能性があるため，すべてのノードはストリームの送信者となり得る．このため，新規参加ノードは，自身を送信者とするストリーム配信のため，他のすべてのノードを含む被覆木をメッセージブロードキャストによる経路探索によりあらかじめ形成しておき，これを経路木として用いる．また，既参加ノードのストリームを受信可能とするため，それらの経路木に参加する．

なお，セッションで適切な品質を実現するために管理者が Emma/QoS の振舞いをいくつか規定できるものとし，これをセッションポリシと呼ぶ．セッションポリシはロビーサーバ

が管理し、各ノードは参加時のノードプロファイル取得と同時にこれを取得する。例えば会議アプリケーションでは、セッションにおける最大のノード間遅延がなるべく小さいことが望ましい。したがって、Emma/QoS では送信者ノードからの経路木のホップ数上限がセッションポリシーに指定されているとし、これをホップ数制約と呼ぶ。このホップ数制約を満たしながら、経路木同士がなるべく重ならないように、新規ノードは既参加ノードの経路木に参加する。

既参加ノードの経路木への参加例を図 2 に示す。図 2(a) は新規ノード e がノード c 及びノード d とオーバレイリンクを構築し、オーバレイネットワークに参加した直後の様子を表す。ノード a, b, c, d はそれぞれすべての他ノードを含む被覆木である経路木（破線）をすでに構築している。ここでホップ数制約が 2 であるとする。ノード e はオーバレイリンク $c-e$ 及び $d-e$ の容量がいずれも 2 であることを考慮し、経路木の重なりがなるべく少なく、かつホップ数制約を満たすよう、ノード a 及び c の経路木には $c-e$ を介して、ノード b 及び d の経路木には $d-e$ を介して参加する（図 2(b)）。また、ノード e 自身の経路木はノード e からのメッセージブロードキャストによる経路探索により決定するが、この図では省略している。

3.3 ストリーム要求のアドミッション制御

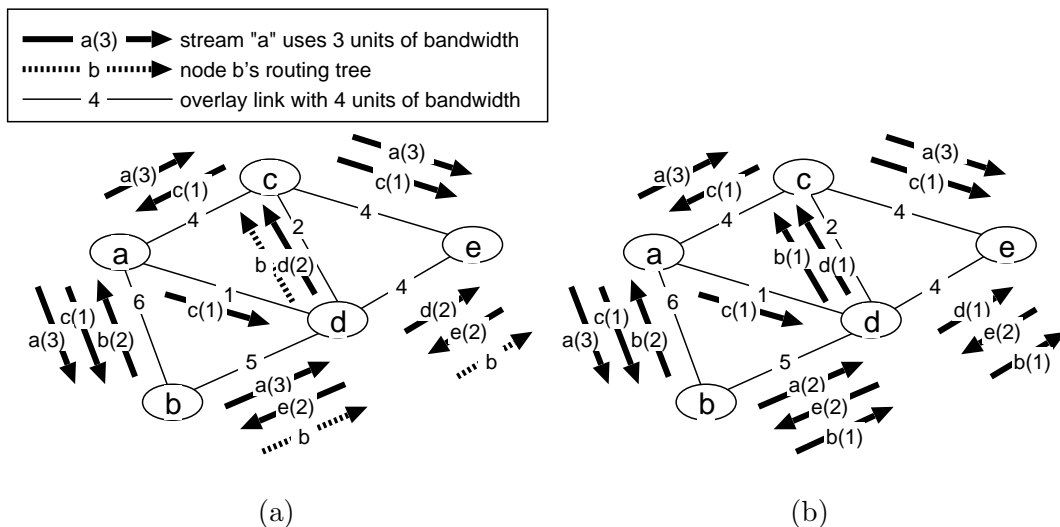


図 3: アドミッション制御例

ノード v の経路木に参加したノードは、ノード v を送信者とするストリーム（これを以下ストリーム v と呼ぶ）の受信を要求できる。Emma/QoS はこの要求を許可するか否かのアドミッション制御を行う。各ストリームは 1 以上の帯域ユニットを消費するとし、送信者からのストリームは受信を許可されたノード群に対し、経路木上をマルチキャスト配信される。

Emma/QoS では、ストリームを中継する各ノードは、フィルタリングやトランスコーディングなどによりストリームが利用する帯域ユニット数を下げることができると仮定する。例えば文献 [24] ではいくつかのフィルタリング技術が紹介されている。さらに各ノードは、ストリームの受信レートが 1 帯域ユニット増加（減少）した場合の自身の利得（損失）を値で表現する。これは各ノードが各ストリームに対して設定したプリファレンスに基づいて算出される。このもとで、あるノードがストリームの受信や既受信ストリームの品質向上のために 1 帯域ユニットを要求した場合、Emma/QoS のアドミッション制御は、フィルタリングにより既使用の帯域ユニットを適切に融通することで満足されるユーザ利得が、融通することによるユーザ損失を上回る場合のみ、その要求を許可する。

図 3 にアドミッション制御の一例を示す。この図では読みやすさのため、ノード b の経路木以外の経路木は表記していない。図 3(a) では各ノードがすでにいくつかのストリームを受信している様子を示している（例えばノード c はストリーム a 及び d をそれぞれ 3 及び 2 帯域ユニットで受信している）。ここで、ノード c が新たにストリーム b の受信を要求したとする。ノード c が発行したストリーム b の受信要求は b の経路木を遡って転送され、ノード b まで到着する。要求が転送される間に、経路 $c-d-b$ 上に空き帯域ユニットがないことが確かめられ、同時に、既使用の帯域ユニットをどのストリームから融通すればユーザ損失が最小に抑えられるかが計算される（これらの処理については 4 章で詳しく述べる）。ここで、ノード c が与えるユーザ利得が、計算されたユーザ損失の最小値を上回らなかったとする。ノード c の要求は棄却されるが、経路上のノード b, d, c はこの要求を一定期間キャッシュする。その間に同じストリームを要求する別のノードからの要求が到着した場合はそれらを集約することで、本来個別に扱われる要求を集約し要求を受け入れやすくする集団効果が期待される。例えば、ノード c の要求棄却後にノード e がストリーム b を要求し

たとする．ノード e の要求はノード d に転送され，キャッシュされているノード c の要求と合わせてノード b に到着する．この場合，リンク $b-d$ ， $d-c$ 及び $d-e$ での帯域ユニットの融通方法が計算される．このように要求のキャッシュによる集約を考えた場合，ストリーム要求が転送される経路はそのストリームの経路木の部分木となる．これを要求木と呼ぶ．今，ノード c 及びノード e のユーザ利得が帯域ユニット融通によるユーザ損失を上回ったとし，ユーザ損失を最小とできる融通方法はノード b がストリーム a から，ノード d がストリーム d からそれぞれ 1 帯域ユニットを減少させることであるとする．それらのノードは指定された帯域ユニット削減を行い，ストリーム b の 1 帯域ユニットでの配信を開始する（図 3(b)）．

なお，Emma/QoS では，ノードが一度の受信要求で得られる帯域ユニットは高々 1 であるとし，それ以上を要求する場合は新たな帯域追加要求を発行するものとする．

3.4 ノードの離脱時の回復処理

Emma/QoS は，セッション途中の参加ノードの離脱などでストリーム配信が中断された場合も，その配信を回復し，ユーザ利得の維持に努める．

具体的には，ストリーム v を転送する各ノード（それぞれを u とする）は，4 章で述べる周期メッセージを用い，自身の子ノード v が離脱した場合にノード v の子ノード w が再接続可能なノードの候補の情報を収集しておく．ここで，ノード w は単純にノード u に再接続するといった選択肢も考えられるが，ノード v の子ノード数が多い場合やノード u の帯域ユニットに余裕がない場合などの状況も考えられるため，ノード u が新たな接続先を指示することで，ストリーム配信を継続できる可能性を向上させることができると考えられる．

3.5 プリファレンスの設定

Emma/QoS におけるプリファレンスは，あるエンドホストがストリームを要求する際に，どの程度優先的に要求するかを表す値である．プリファレンス値の設定方法は利用するアプリケーションごとに任意に決定することができる．ビデオ会議アプリケーションにおいては，各ユーザに一定の持ち点を与え，それを要求ビデオに配分するチケット方式のプリファ

レンス設定方法が最も適当であると考えられる。このチケットはセッション参加時に他ノードのプロファイル取得した際に同時に入手する。また、3.7 章で後述するポリシー管理によって、不正なプリファレンス値の出現を防止する。

3.6 異なる複数の受信要求による競合の回避

Emma/QoS のノードは任意のストリームの受信要求を行えるため、互いに異なるストリームを要求する複数のストリーム要求 (MEDIA/Join request) メッセージが同じオーバーレイリンクを転送されることもあり得る。それらを同時並行的に処理することによる帯域の競合とそれに伴う矛盾の発生を回避するため、MEDIA/Join request メッセージを転送するノードは自身の状態を“予約状態”としておき、予約状態であるノードに後から到着した MEDIA/Join request メッセージは棄却するようにする。これにより、そのような競合の発生を避けることができる。また、予約状態は MEDIA/Join request メッセージの応答メッセージを転送する際に解除する。

3.7 ポリシ管理

不正な利得値や損失値を用いるノードの出現を防ぐために、隣接ノードの送出した値を監視し、お互いにそれらの上限値 (それは例えばセッションポリシーに指定することができる) を守っているかをチェックしあう方法が考えられる。このような方法はストリーム維持要求 (MEDIA/Keep report) メッセージを用い、新たなメッセージを導入することなく組み入れることができると考えられる。

また、受信要求が受け入れられるまで、同じ受信要求を頻繁に送出するノードが現れる可能性もあり、そのような場合はトラヒックと処理能力を浪費する。この問題に対しては、前述した要求のキャッシュが有効であると考えられる。同じ要求がキャッシュに残っている間は MEDIA/Join request を転送しないことで同じ要求が繰り返し送出されることを避けることができる。

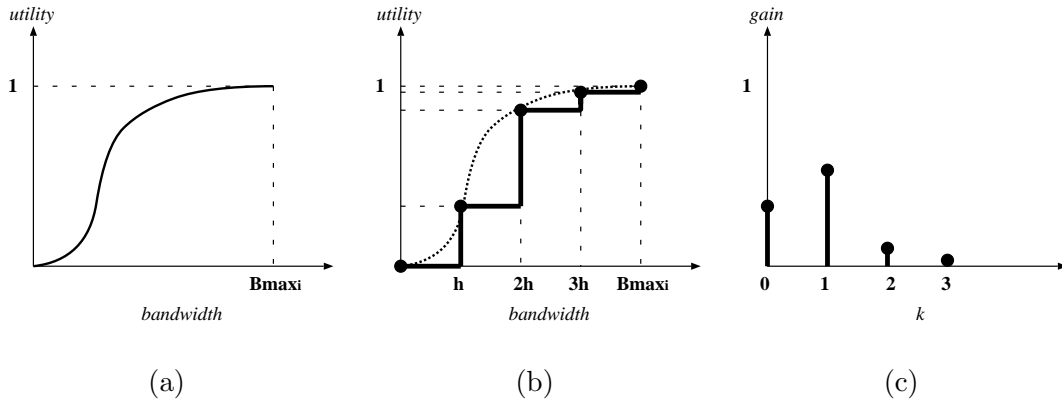


図 4: ユーティリティ関数からのユーザ利得関数の導出

4 ストリーム配信時の QoS 制御

本章では, 3.3 節で述べたストリームのアドミッション制御の詳細について述べる.

4.1 ユーザ利得及びユーザ損失関数

Emma/QoS では, 帯域ユニットによるストリーム配信制御を行うため, プリファレンスを帯域ごとに定める必要がある.

ノード v におけるストリーム i の受信レートが k 帯域ユニットであるとする (受信していない場合は $k = 0$). このとき, ストリーム i に 1 帯域ユニットを追加した場合のノード v が得る利得 (ユーザ利得) を $gain_i(v, k)$ で表す. また 1 帯域ユニットが削減された場合の損失 (ユーザ損失) を $loss_i(v, k)$ で表す. Emma/QoS はこれらの値を用いてアドミッション制御を行う.

しかし, ノードが k ごとに利得及び損失を逐一定義することは一般に煩雑である. そこで, 帯域を追加することで得る品質向上の程度を, 文献 [25] で紹介されている帯域対品質関数 (ユーティリティ関数) を用いて表し, それを元にノードの利得及び損失を定義する方法を述べる.

文献 [25] では, レート適応型アプリケーションにおけるユーティリティ関数が図 4(a) のように定義されている. $Bmax_i$ はストリーム i の最大レートである. Emma/QoS では, ある帯域幅 h を単位とした帯域ユニットで帯域を扱うとしているため, 与えられたユーティ

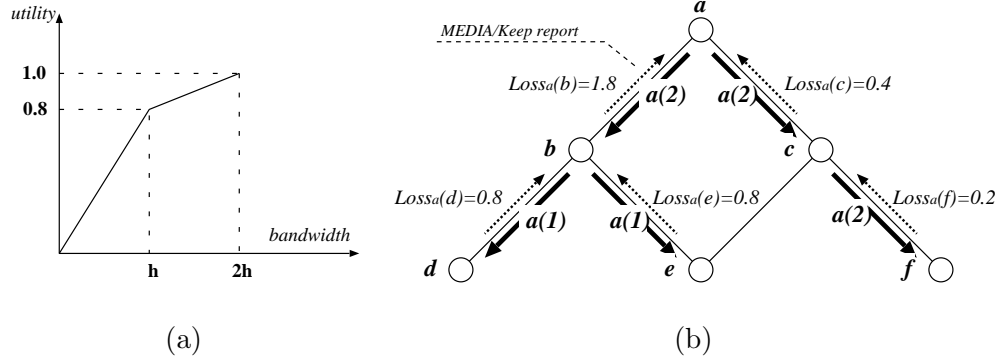


図 5: MEDIA/Keep メッセージの転送

リイティ関数に対し，図 4(b) のように帯域ユニットごとのユーティリティ値を得る．これを用い関数 $gain_i(v, k)$ は帯域ユニットの追加に伴うユーティリティの増加量として以下のように定義できる（図 4(c)）．

$$gain_i(v, k) = \begin{cases} U_i(h(k+1)) - U_i(hk) & (k = 0, 1, \dots, \lceil \frac{Bmax_i}{h} \rceil - 1) \\ 0 & (k = \lceil \frac{Bmax_i}{h} \rceil) \end{cases} \quad (1)$$

ここで， U_i はストリーム i のユーティリティ関数である．同様に， $loss_i(v, k)$ は以下のように定義できる．

$$loss_i(v, k) = \begin{cases} U_i(hk) - U_i(h(k-1)) & (k = 1, \dots, \lceil \frac{Bmax_i}{h} \rceil) \\ 0 & (k = 0) \end{cases} \quad (2)$$

なお，例えば 2 Mbps の MPEG2 と 512 kbps の MPEG4 など異なる符号化方式が混在する場合，各ストリーム i ごとの $Bmax_i$ も大きく異なる．そのような場合，例えば $h = 512$ kbps とし， $k = 0$ の場合はいずれも同じ大きい利得を，MPEG2 の $k > 0$ に対しては $k = 0$ の場合と比較して小さい利得を与える利得関数を定義し，“MPEG2 か MPEG4 かにかかわらずストリームの基本帯域 512 kbps は優先し，帯域に余裕がある場合は MPEG2 ストリームにより多くの帯域を与える”といったポリシーを反映させることも可能である．

4.2 ユーザ利得及びユーザ損失に基づいたアドミッション制御

4.2.1 ユーザ損失の定期的な公告

ノード u からストリーム i を受信するノード v は、ノード v 及びノード v の下流でストリーム i を受信しているすべてのノードが、ストリーム i の現在の受信レートからそれぞれ 1 帯域ユニットを失った場合のユーザ損失の総和（以下 $Loss_i(v)$ で表す）を保持する。ノード v は MEDIA/Keep report と呼ばれるメッセージにより、この値をノード u に送信する。 $Loss_i(v)$ は、以下のように定義できる。

$$Loss_i(v) = loss_i(v, k) + \sum_w Loss_i(w) \quad (3)$$

定義より、ストリーム i をノード v から直接受信する各子ノード w から MEDIA/Keep report メッセージを受けとったときに $Loss_i(v)$ が計算でき、その値を含む MEDIA/Keep report メッセージを親ノードに送ることができる。

例えば、図 5(a) のようなユーティリティ関数を、すべてのノードがストリーム a に対し使用すると仮定し、ストリーム a は図 5(b) のように配信されているとする。ノード b はノード d 及び e からそれぞれ、 $Loss_a(d) = loss_a(d, 1) = 0.8$ と $Loss_a(e) = loss_a(e, 1) = 0.8$ を受け取る。その後、ノード b はこれらの値と自身のユーザ損失 $loss_a(b, 2) = 0.2$ の和を計算し、その総和（この場合は 1.8）を MEDIA/Keep report メッセージに含ませて上位ノード（ノード a ）に送る。

4.2.2 受信要求の処理手順

各ノードは MEDIA/Join request メッセージと呼ばれるメッセージを送信することで、ストリーム s の受信を要求できる。このメッセージはノード s に向けて s の経路木を遡り、ストリーム s が既に配信されているノード（ノード s' とする）に到着するまで転送される。

なお、3.3 節の例で述べたように、あるストリーム s の受信を要求して棄却された MEDIA/Join request メッセージはそれが転送された経路上でキャッシュされ、同じストリーム s を要求する他のノードの MEDIA/Join request メッセージと集約される。結果として、複数のノードが s の経路木の部分木上で帯域ユニットを要求することになる。このように

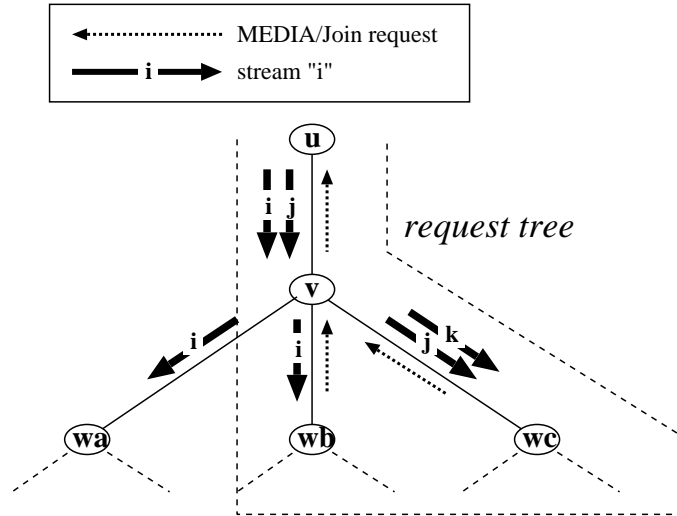


図 6: $MinLoss_i(v)$ の計算

キャッシュにより集約されたすべての MEDIA/Join request の転送経路を表す部分木をストリーム s の要求木と呼ぶ。Emma/QoS は、ストリーム s の要求木に 1 帯域ユニットを確保する際のユーザ損失の総和が受信要求を満足することによるユーザ利得より小さければそれらの要求を受け入れるアドミッション制御を行う。

ここで、要求木におけるノード v の親ノードをノード u とする。今、ノード u を根としノード v を経由する、要求木の部分木で 1 帯域ユニットを確保する方法を考える。ノード u はノード v へ転送しているいずれかのストリームから 1 帯域ユニットを削減する。このストリームを i としたときのユーザ損失総和の最小値を $MinLoss_i(v)$ で表す。 $MinLoss_i(v)$ は次のように定義される。

$$\begin{aligned}
 MinLoss_i(v) &= loss_i(v, k) \\
 &+ \sum_{w_a} Loss_i(w_a) \\
 &+ \sum_{w_b} MinLoss_i(w_b) \\
 &+ \sum_{w_c} \min_j \{MinLoss_j(w_c)\}
 \end{aligned} \tag{4}$$

ここで、ノード w_a はストリーム s の要求には関係がない（要求木上にない）が、ストリーム i を受信している各ノードを表す。 w_a はノード u がストリーム i の帯域を削減した場合

にユーザ損失 $Loss_i(v)$ を生じるため，そのような w_a に関する総和により第 2 項を得る．ノード w_b はストリーム s の要求木上にあり，かつストリーム i を受信している各ノードを表す．したがって，ノード u がストリーム i の帯域を削減した場合，その下流であるリンク $v-w$ でも自動的にストリーム i の帯域が削減される．このとき，ノード v を根とし，ノード w_b を経由する要求木の部分木に，1 帯域ユニットを確保する際のユーザ損失総和の最小値は $MinLoss_i(w_b)$ であり，そのような w_b に関する総和により第 3 項を得る．最後に，ノード w_c はストリーム s の要求木上にあり，かつストリーム i を受信していない各ノードを表す．したがって，ノード v を根としノード w_c を経由する，要求木の部分木に 1 帯域ユニットを確保する際のユーザ損失総和の最小値は，別のストリーム j から帯域を削減する必要があることから $\min_j \{MinLoss_j(w_c)\}$ となり，そのような w_c に関する総和により第 4 項を得る．図 6 に，ノード w_a , w_b 及び w_c がそれぞれ 1 つ存在する簡単な例を示す．

ノード v がノード u に送信する MEDIA/Join request メッセージには，ノード u がノード v に転送している各ストリーム i ごと $MinLoss_i(v)$ を含めておく．これらの値と MEDIA/Keep report メッセージにより計算された損失の総和から，要求木上の各ノードは再帰的に最小のユーザ損失総和を計算できる．また，要求したノードが与える利得の総和

$$Gain_i(v) = gain_i(v, 0) + \sum_w Gain_i(w) \quad (5)$$

も同様に MEDIA/Join request メッセージに含めておく．

要求されたストリーム s がすでに配信されているノード s' は MEDIA/Join request メッセージを受け取り，ユーザ損失の総和の最小値とユーザ利得の総和とを比較する．後者が前者より大きい値であれば，要求を許可する MEDIA/Join acceptance メッセージを，そうでなければ MEDIA/Join rejection メッセージを要求木上に送信する．MEDIA/Join acceptance メッセージを受け取った要求木上のノードは，指示されたストリームの帯域を削減し，要求されたストリーム s の転送を開始する．

なお，すでに受信しているストリームに対し帯域ユニットを追加する要求も，MEDIA/Join

request と同様に扱われる。

4.3 フィルタリングが不可能なノードへの対応

セッションに参加したノードの中には、能力的にフィルタリングを行えないものもある。これらのノードを通過して配信した場合には、Emma/QoS の行う配信制御が適用することができないため、現在のレートで下流ノードに配信できない場合は、配信を停止しなくてはならない。これを回避するために、セッション参加時の各配信経路木構築の際、該当ノードは子ノードを接続しないようにする。

4.4 利用帯域変動への対応

Emma/QoS ではビデオの配信を RTP (Real-time Transport Protocol) を用いて行い、このときのセッション維持を RTCP (RTP Control Protocol) を用いている。この RTCP によって送信品質を検出できる。

オーバーレイリンク容量はリンクの両エンドホストが独自に決定するため、実際に利用可能な帯域(実帯域)との差異も生じる。ここで、リンク容量が実帯域を上回ると輻輳によるストリームのデータ損失などが発生する可能性がある RTCP によって、このような状況を検出した場合は、そのオーバーレイリンク上のいずれかのストリームの帯域を削減した上で、リンク容量を再設定することで対応できる。なお、その際にはユーザ損失が最小となるストリームを選択する。また、帯域を削減した後もなおパケットロスが検出される場合は、そのオーバーレイリンクを用いた配信を回避するために、リンク切断メッセージを出し、該当リンクでのストリーム配信を停止する。その後、そのリンク以外のリンクへ新たにストリーム受信要求を出すことによって、新しい経路を構築する。

また、オーバーレイリンクはエンドホスト付近(LAN)の帯域が最も狭くなっていると考えられるため、各エンドホストは自らの LAN 帯域を監視し、1 帯域ユニット分の帯域増加が見られた場合はオーバーレイリンク容量を増加させる。

5 性能評価

オブジェクト指向スクリプト言語 Ruby[3] を用いて Emma/QoS のシミュレータを実装し、性能評価を行った。

5.1 測定項目

本論文で評価する項目は以下の通りである。

1. リンクストレス：実リンク上を配送される同一パケット数
2. パスストレッチ：2 ノード間のユニキャストの実ホップ数に対するオーバーレイネットワーク上での実ホップ数の比
3. ツリーストレス：オーバーレイリンク上の経路木の数
4. ユーザの要求満足率：ノードごと、そのノードが送信した要求のユーザ利得総和に対する受け入れられた要求のユーザ利得総和の比)
5. 満足されたユーザ利得分布のシナリオによる時間変動
6. 全ユーザで満足されるユーザ利得総和

一般に ALM プロトコルとしての性能を評価するには、構築される配信木の品質が問題となる。リンクストレスとパスストレッチは他の ALM でも多く用いられており、これらには相関関係があると考えられる。例えばリンクストレスを高くすればパスストレッチは短くなり、ユニキャストに近づく。本論文では Emma/QoS におけるこれらの値が、実用的に妥当でかつ適切なバランスを達成しているかをユニキャスト、代表的な ALM である Narada[11, 12] の場合と比較することで測定した。また、ソースごとに配信木を構築するツリーストレスは Emma/QoS (及び Emma) 独自の評価基準であり、共有木を用いる他の多くの ALM では扱われていない。Emma/QoS ではストリーム間でなるべく帯域を競合しないよう、あらかじめ経路木の重なりがなるべく少なくなるようにしているため、この値を測定することでそれがどの程度達成されているかを調べた。シナリオに従って参加離脱が繰

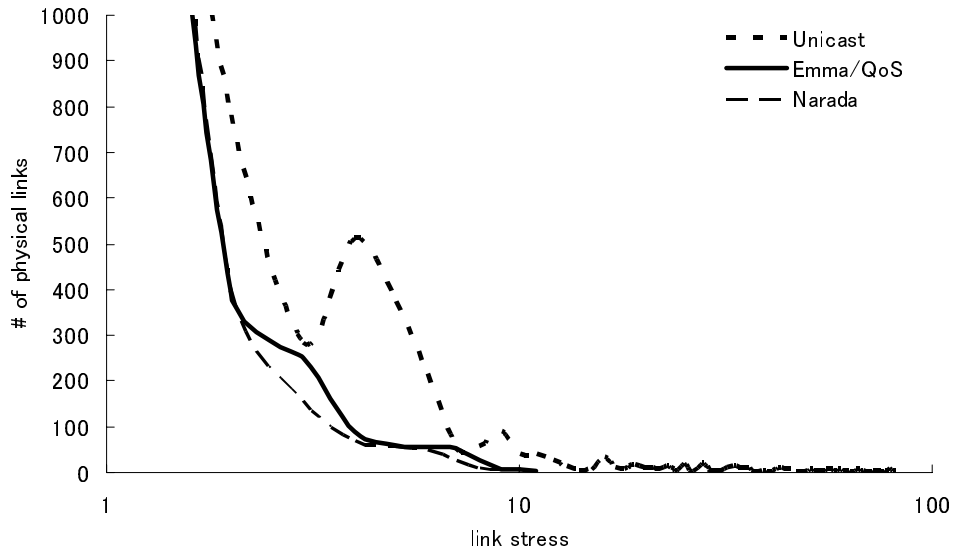
り返された場合に、離脱処理によってこれら各値が悪化していないことを確認するために、各値の時間変動も測定した。

ユーザの要求満足率及びユーザ利得の時間変動も、ユーザ利得という概念を用いる Emma/QoS 独自の評価基準である。Emma/QoS では、ユーザのセッション途中の受信要求に対しても動的かつ柔軟なアドミッション制御を行う点が最大の特長であるため、この制御がどの程度の効果をあげているかを確認するためにそれらの値を測定した。この値を、代表的な ALM である Narada において同一のシナリオを実行した場合に得られる値と比較した。なお、Narada では同時に複数のストリームを表示することを想定していない。しかし、利用できる帯域がなくなった場合は、中継ホストにおいてパケットを落とすことにより配信ストリームのレートを下げる仕組みが備えられている。これを利用して、要求を満たすことができなかった場合は配信中の全ストリームのレートを下げ、新たな要求が受け入れるような形で実装した。また、4.1 章のユーティリティ関数とビデオに関するプリファレンス値を Emma/QoS と同様に適用して、配信されているストリームのユーザ利得を計算した。

グループサイズが増大した場合に、ユーザ利得値の総和がどのような変化をするかを調べ、スケーラビリティを確かめた。

5.2 シミュレーションシナリオ

ネットワークは LAN, MAN, WAN より生成されるルータ数 400 の階層型ネットワークを tiers モデル [2] に基づいて生成し、LAN, MAN, WAN の帯域はそれぞれ 6Mbps, 50Mbps, 100Mbps とした。これはユーザ付近のリンクの帯域がボトルネックリンクとなることを表す。各ストリームの最大レートは 1Mbps とし、ノードは 256kbps 毎にストリームをフィルタリングできる、すなわち 1 帯域ユニットは 256kbps とした。オーバーレイノード数が約 60 である場合について評価を行った。ノードはセッション参加時に 3 本のオーバーレイリンクを構築し、1 ノードからのオーバーレイリンク容量の総和が、LAN の帯域の半分、すなわち 12 帯域ユニットを越えないようにした。各ストリームのユーザ利得 / 損失関数は 4.1 節で定義した関数に従い定義した。なお、セッションポリシのホップ数制約は 4 とした。また、各ユーザがビデオ要求時に要求ビデオに与えるプリファレンスは zipf にしたがって



59 users (187 node networks)

図 7: リンクストレス分布

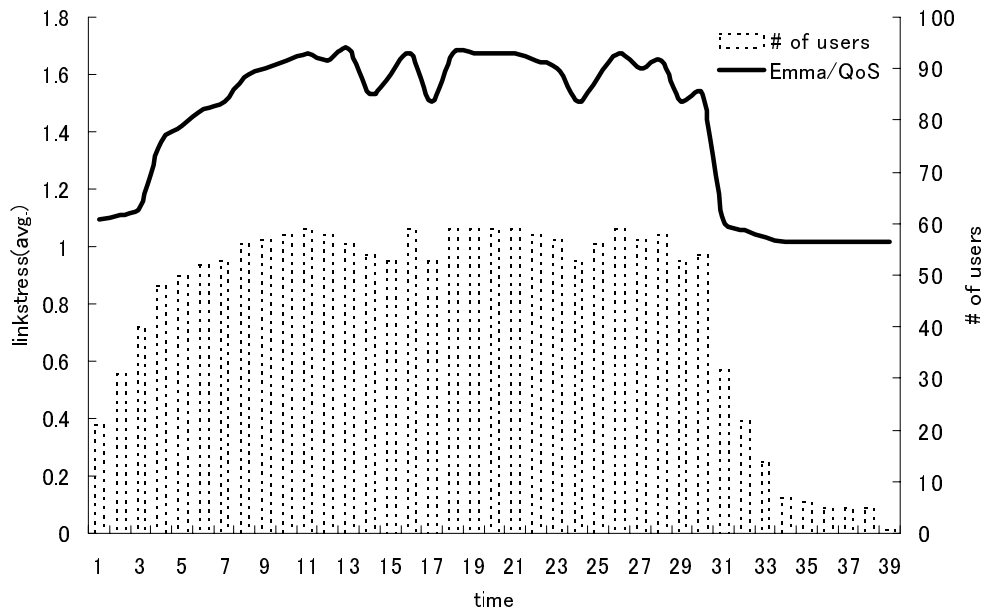
各ビデオごとに設定されるものとする。

シミュレーションシナリオは実際のビデオ会議を想定し，参加，要求，要求変更，離脱の 4 つのフェイズに分ける．このとき，例えば，参加フェイズではアクションを行うノードの約 50% が参加アクションを行い，他のアクションは約 10% が行うとしており，他のフェイズでも同様である．これら 4 つのフェイズを連続して実行することによって，ビデオ会議を想定したシナリオでシミュレーションを行うことができる．シミュレーション時間に伴うユーザ数の変動は図 8 等の棒グラフの変動によって表しているため参照されたい．

5.3 リンクストレスとパスストレッチの評価

まず，要求フェイズに入った時点での（図 13 参照）のリンクストレス及びパスストレッチを測定した．これは，要求フェイズに入った時点で，参加・離脱アクションが少なくなるため，オーバレイネットワークポロジが最も安定していると考えられるためである．

図 7 にリンクストレスの分布を示す．ここで，X 軸はリンクストレス値を対数表示した

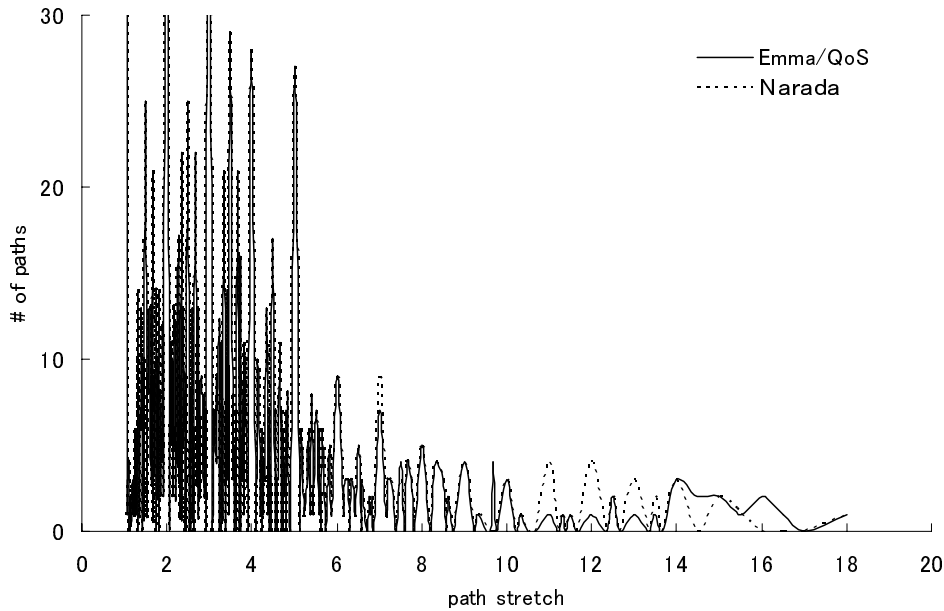


59 users (187 node networks)

図 8: リンクストレス変動

もの、Y 軸はそのリンクストレスを持つ物理リンクの数を表している。また比較のため、ユニキャストでの値、Narada での値を併記した。Emma/QoS はマルチキャストであるため、ユニキャストより良い値が得られる。また、最大リンクストレスがユニキャストの場合の十分の一程度で収まっており、ALM によるリンクストレス削減効果は十分であると考えられる。Narada は全てのノードからランダムに数ノードを選んでオーバーレイネットワークを構築し、その上に shortest widest path tree を構築しているため、Emma/QoS ともそれほど差はないことがわかる。次にシナリオによって変動したリンクストレスの平均値変動を図 8 に示す。ここでは、X 軸がシミュレーション時間、Y 軸が各ノードでの平均リンクストレスである。これによると、ユーザ数変動した場合も、リンクストレスの極端な悪化は見られないことが分かる。

図 9 では、パスストレッチの分布とその累積分布を示す。X 軸はパスストレッチ値、Y 軸はそのパスストレッチを持つパスの数を表している。パスストレッチも Narada と同程度の結果が得られた。



59 users (187 node networks)

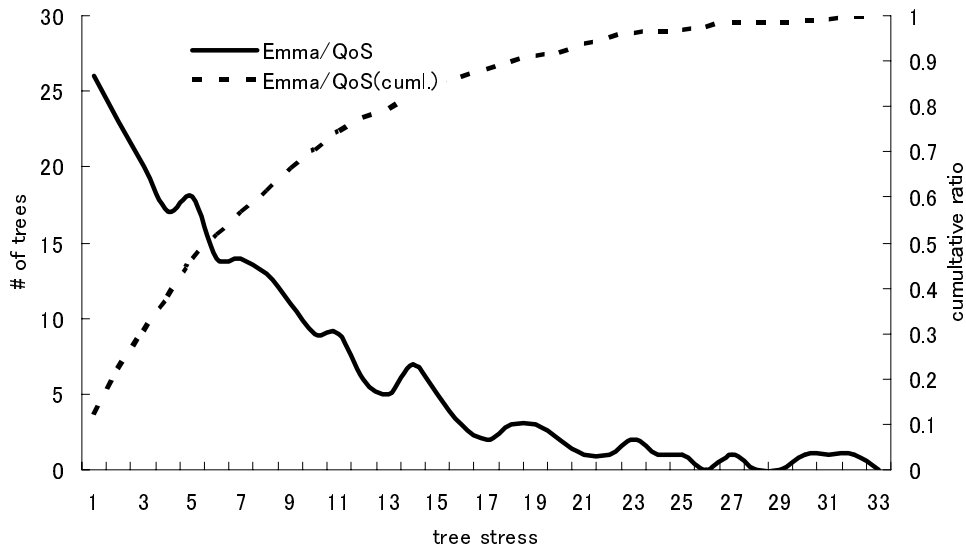
図 9: パスストレッチ分布

5.4 ツリーストレスの評価

リンクストレス、パスストレッチと同様、要求フェイズに入った直後のツリーストレスの分布及び累積分布を図 10 に示す。複数の映像ストリームを配信する対象アプリケーションでは、ツリーストレスは一様に分布していることが望ましい。グラフより、ツリーストレスは 90% の値が最悪値の半分以下に収まっている。したがって、おおよそ一様に分布していると見なすことができ、重複の少ない経路木構築の効果が得られているといえる。また、シナリオによって変動したツリーストレスの平均値の変動を図 11 に示す。この場合もリンクストレス、パスストレッチと同様に、ユーザ数変動に対して値の悪化は見られない。

5.5 ユーザ要求満足率の評価

ノードごと、そのノードが送信した要求のユーザ利得総和に対する受け入れられた要求のユーザ利得総和の比をユーザ要求満足率と定義し、この値を測定した。図 12 に値の分布を示す。ここで、X 軸は要求満足率、Y 軸は該当するユーザ数を表している。なお、従来の代



59 users (187 node networks)

図 10: ツリーストレス分布

表的な ALM である Narada との比較を行った。また参考として、Emma/QoS の方式で構築した配信木上で、リンクに空きがある限り要求順に受け入れ、空きがない場合は受け入れない到着順方式 (FCFS) でビデオの配信を行った場合の値も計測した。Emma/QoS では競争が発生した際に、より低い優先度を持つストリームのレートを下げるという制御を行っているため、他の方式と比べ、要求が全く満たされないユーザの数が少なくなっている。ビデオ会議システムなどでは単純な動画ストリーミングと異なり、ユーザは高品質な単一のビデオよりは、低品質でも複数のビデオを要求する傾向がある。この点で、Emma/QoS のアドミッション制御機能はそのようなアプリケーションに有効であると言える。

5.6 ユーザ利得値の分布の測定

また、シミュレーション時間でのユーザ利得の各ユーザ平均値の変動を測定した。その結果を図 13 に示す。X 軸にシミュレーション時間、Y 軸にその時間で満たされているユーザ利得の各ユーザ平均値とその時間でのユーザ数をとっている。比較対象として Narada を用いた。この結果から、セッション開始後、ユーザの参加・離脱、ビデオに対するプリファレ



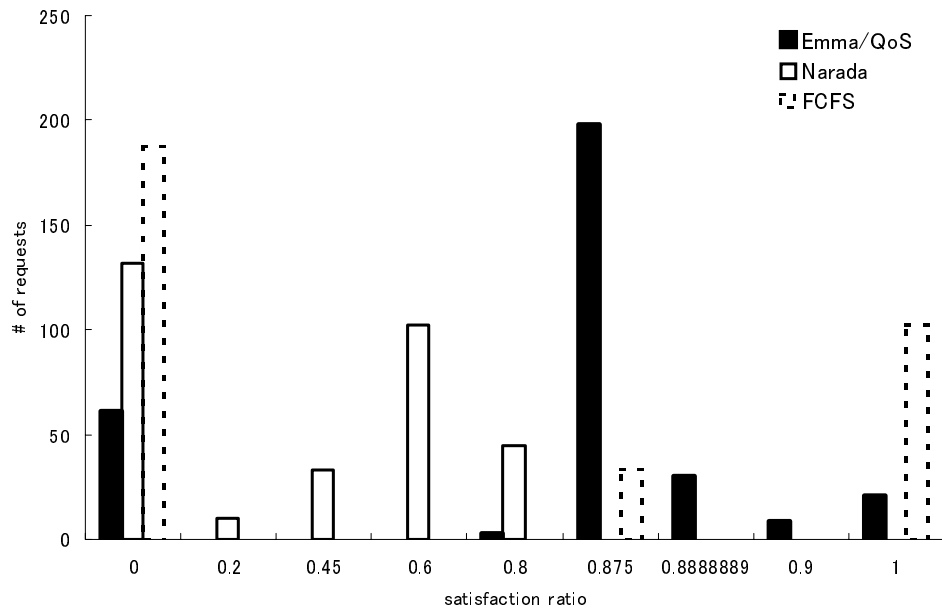
59 users (187 node networks)

図 11: ツリーストレス変動

ンスの変更を行った場合のいずれも、Narada と比較して高いユーザ利得を達成していることが分かった。これは、優先度要求をもとにした Emma/QoS ストリーム制御の効果であり、ユーザに対し従来より高い満足度を与えているといえる。

5.7 グループサイズにおけるユーザ利得値総和の測定

また、グループサイズ(参加する総ユーザ数)が増大した場合におけるユーザ利得値総和を測定した。その結果を図 14 に示す。この場合も同様に、比較対象として Narada, FCFS を用いた。この結果から、グループサイズが大きくなった場合もユーザ利得に関する Emma/QoS の優位性は保たれており、優先度要求をもとにした Emma/QoS のストリーム制御は有効であると考えられる。



59 users (187 node networks)

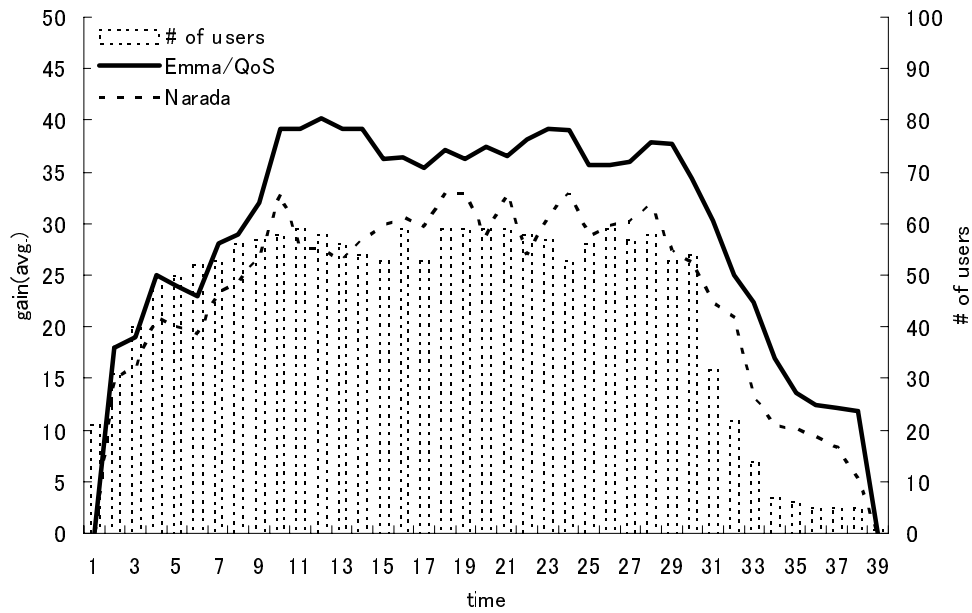
図 12: ユーザ要求満足率分布

6 あとがき

本研究では、P2P ネットワークにおける複数の実時間ストリームの同時配信時に、ストリーミングの QoS を分散制御で実現する ALM プロトコル Emma/QoS を提案した。また、Emma/QoS, Narada[11, 12] を実装したシミュレータを作成し、これらと比較することによってプロトコルの性能を評価した。

オーバーレイネットワーク上の帯域はエンドホスト間の帯域だけでなく、LAN の帯域によっても制限される。一方でエンドホストの計算機資源は、近年の計算機能力の著しい向上により、多くのネットワークアプリケーションにおいて余剰していると考えられる。本論文ではそのようなエンドホストの計算能力を利用し、オーバーレイネットワーク上での QoS 制御を実現する新しい方法論を提案している。

今後の課題は、より大規模なグループを想定した場合の制御メッセージ総量などに関するシミュレーション実験を行い、規模適用性についての評価を行うことなどを考えている。



59 users (187 node networks)

図 13: ユーザ利得値平均の変動

謝辞

本研究にあたり、理解ある御指導、御鞭撻を賜わり、常に励まして頂きました東野輝夫教授に心から深く感謝致します。

本研究にあたり、適切な御指導、御助言を頂きました中田明夫助教授に心から深く感謝致します。

本研究にあたり、的確な御助言を頂きました吉岡敏明助手、梅津高朗助手に心から深く感謝致します。

また、本研究にあたり、親切な御指導、御助言を頂きました谷口研究室の山口弘純助手、博士後期課程3年生の廣森聡仁氏に心から深く感謝致します。さらに、本研究において、貴重なご意見などを頂きました奈良先端科学技術大学院大学情報科学研究科の安本慶一先生に心から感謝致します。

最後に、東野研究室の皆様のご心温まる御助言、御協力に感謝致します。

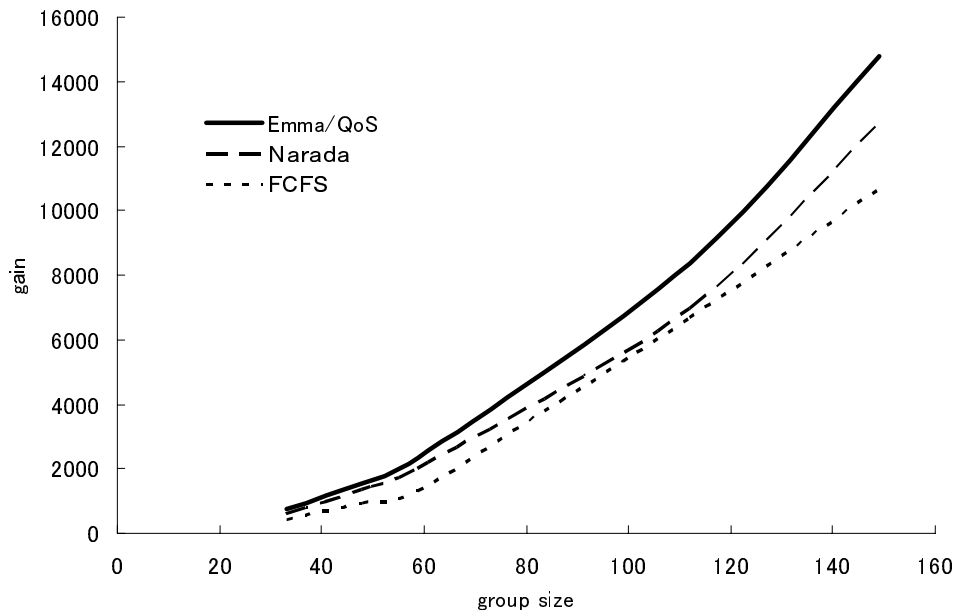


図 14: グループサイズ変化によるユーザ利得値総和の変化

参考文献

- [1] C. Diot, W. Dabbous and J. Crowcroft : “Multipoint Communication: A Survey of Protocols, Functions, and Mechanisms”, IEEE Journal on Selected Areas in Communications, Vol. 15, No. 3, pp. 277–290, 1997.
- [2] K.L. Calvert, M.B. Doar and E.W. Zegura : “Modeling Internet Topology”, IEEE Communications Magazine, pp. 160–163, 1997.
- [3] “Ruby Home Page”, <http://www.ruby-lang.org/en/>.
- [4] “Gnutella”, <http://www.gnutella.com/>.
- [5] I. Clarke, O. Sandberg, B. Wiley, and T.W. Hong : “Freenet: A Distributed Anonymous Information Storage and Retrieval System”, Lecture Notes in Computer Science, Vol. 2009, pp. 46–66, 2001.
- [6] “JXTA”, <http://www.jxta.org>.

- [7] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan : “Chord: A Peer-to-Peer Lookup Service for Internet Applications””, Proc. of ACM SIGCOMM 2001, pp. 149–160, 2001.
- [8] K. C. Almeroth : “A Long-Term Analysis of Growth and Usage Patterns in the Multicast Backbone (MBone)”, Proc. of IEEE INFOCOM 2000, pp. 824–833, 2000.
- [9] S. McCanne and V. Jacobson : “Receiver-driven Layered Multicast”, Proc. of ACM SIGCOMM ’96, pp. 117–130, 1996.
- [10] E. Amir, S. McCanne and R. Katz : “Receiver-driven Bandwidth Adaptation for Light-weight Session”, Proc. of ACM Multimedia, pp. 415–426, 1997.
- [11] Y.-H. Chu, S.G. Rao and H. Zhang : “A Case for End System Multicast”, Proc. of ACM SIGMETRICS 2000, pp. 1–12, 2000.
- [12] Y.-H. Chu, S.G. Rao, S. Seshan and H. Zhang : “Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture”, Proc. of ACM SIGCOMM 2001, pp. 55–67, 2001.
- [13] D. Pendarakis, S. Shi, D. Verma and M. Waldvogel : “ALMI: An Application Level Multicast Infrastructure”, Proc. of the 3rd Usenix Symp. on Internet Technologies & Systems, pp. 49–60, 2001.
- [14] J. Jannotti, D.K. Gifford, K.L. Johnson, M.F. Kaashoek and J.W. O’Toole : “Overcast: Reliable Multicasting with an Overlay Network”, Proc. of the 4th Usenix Symp. on Operating Systems Design and Implementation (OSDI), pp. 197–212, 2000.
- [15] V. Roca, A. El-Sayed : “A Host-Based Multicast (HBM) Solution for Group Communications”, Proc. of 1st IEEE Int. Conf. on Networking (ICN’01), pp. 610–619, 2001

- [16] R. Cohen and G. Kaempfer : “A Unicast-based Approach for Streaming Multicast”, Proc. of IEEE INFOCOM 2001, pp. 440–448, 2001.
- [17] S. Ratnasamy, M. Handley, R. Karp and S. Shenker : “Application-level Multicast using Content-Addressable Networks”, Proc. of 3rd Int. Workshop on Networked Group Communication (NGC’01), pp. 14–29, 2001.
- [18] Y. Chawathe, S. McCanne and E.A. Brewer : “RMX: Reliable Multicast for Heterogeneous Networks”, Proc. of IEEE INFOCOM 2000, pp. 795–804, 2000.
- [19] S.Q. Zhuang, B.Y. Zhao, A.D. Joseph, R.H. Katz and J. Kubiawicz : “Bayeux: An Architecture for Scalable and Fault-tolerant Wide-area Data Dissemination”, Proc. of ACM NOSSDAV 2001, pp. 11–20, 2001.
- [20] P. Francis : “Yoid: Extending the Internet Multicast Architecture”, <http://www.icir.org/yoid/>.
- [21] S. Banerjee, B. Bhattacharjee and C. Kommareddy : “Scalable Application Layer Multicast”, Proc. of ACM SIGCOMM 2002, pp. 205–217, 2002.
- [22] F. Baccelli, D. Kofman and J.L. Rougier : “Self Organizing Hierarchical Multicast Trees And Their Optimization”, Proc. of IEEE INFOCOM2001, pp. 1081–1089, 2001.
- [23] Y. Nakamura, H. Yamaguchi, A. Hiromori, K. Yasumoto, T. Higashino and K. Taniguchi : “On Designing End-user Multicast for Multiple Video Sources”, Proc. of 2003 IEEE Int. Conf. on Multimedia & Expo, Vol. III, pp. 497–500, 2003.
- [24] N. Yeadon, F. García, D. Hutchison and D. Shepherd : “Filters: QoS Support Mechanisms for Multipeer Communications”, IEEE Journal on Selected Areas in Communications, Vol. 14, No. 7, pp. 1245–1262, 1996.
- [25] S. Shenker : “Fundamental Design Issues for the Future Internet”, IEEE Journal on Selected Areas in Communications, Vol. 13, No. 7, pp. 1176–1188, 1995.

- [26] D. Xu and K. Nahrstedt : “Finding Service Paths in a Media Service Proxy Network,”
Proc. of SPIE/ACM Conf. on Multimedia Computing and Networking (MMCN2002),
pp. 171–185, 2002.
- [27] S. Chen, K. Nahrstedt and Y. Shavitt : “A QoS-Aware Multicast Routing Protocol”,
Proc. of IEEE INFOCOM2000, pp. 1594–1603, 2000.