

エンドホストの動画画像フィルタリングによる アプリケーション層での QoS マルチキャストプロトコルの提案

中村 嘉隆[†] 山口 弘純[†] 廣 森 聡 仁^{††}
安本 慶一^{†††} 東野 輝夫[†] 谷口 健一[†]

本論文では、QoS を考慮したアプリケーション層マルチキャストプロトコル Emma/QoS を提案する。Emma/QoS は、アプリケーション層で構築された仮想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 制御を分散方式で実現する。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。このもとで、映像ストリームの配信レートをそれらのエンドホストが協調して動的に調整し、限られた仮想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の結果、Emma/QoS は高いユーザ満足度を実現できることが分かった。

An Application Layer Multicast Protocol for Providing Quality of Service by Using End System's Video Filtering

YOSHITAKA NAKAMURA,[†] HIROZUMI YAMAGUCHI,[†]
AKIHITO HIROMORI,^{††} KEIICHI YASUMOTO,^{†††} TERUO HIGASHINO[†]
and KENICHI TANIGUCHI[†]

In this paper, we propose a QoS-aware application layer multicast protocol called Emma/QoS. Emma/QoS provides QoS control in simultaneous transmission of multiple video sources in a fully distributed manner. Emma/QoS assumes that each end-host is capable of filtering video sources when forwarding them to the other end-hosts. Using this functionality, Emma/QoS (a) accommodates a request of a new video source by degrading the bit-rates of existing video streams, and (b) adjusts the transmission rates of existing video streams according to given user benefit functions, which can be application-specific. Simulation results have shown the efficiency of Emma/QoS.

1. はじめに

高速ネットワークの普及は、インターネットを用いた広域でのグループ通信の需要増加をもたらすと予想される。グループ通信は、主にグループ内の複数のメンバへの同報通信からなる。IP マルチキャスト¹⁾はそのようなグループ通信を実現する有用な基盤技術の1つである。しかし、現状ではドメイン内での利用が

主であり、今後いっそうの普及が期待されている段階である。一方、すべての2ユーザ間のユニキャストチャンネルを設立することは、グループの規模に関するスケラビリティが欠如する。

この問題に対する現実的な解決法として、アプリケーション層でマルチキャスト通信を行うアプリケーションレベルマルチキャスト(以下 ALM と略記する)が注目を集めている。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(オーバーレイネットワーク)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。たとえば、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回配送されているように、ALM では単一の実リンク上で

[†] 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University

^{††} 大阪大学大学院基礎工学研究科
Graduate School of Engineering Science, Osaka University

^{†††} 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology

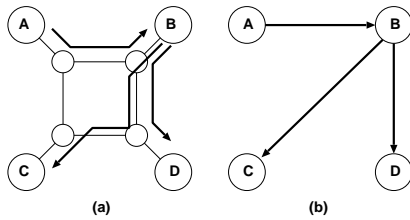


図1 アプリケーションレベルマルチキャスト
Fig. 1 Application level multicast.

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。

これまでの ALM の研究^{4)~8),11)} では、ALM のオーバー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。しかし、複数の映像が同時並行的にストリーミング配信されるようなアプリケーション（たとえばビデオ会議システム）では、オーバーレイリンク上の帯域や、ホストの転送能力などオーバー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。

これに対し、文献 15) では、ALM プロトコル Emma (End-user Multicast for Multi-party Applications) が提案されている。Emma では、オーバー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、限られたオーバー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。たと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ー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、プリファレンスがより高い映像ストリームの配信品質を保证する。

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¹⁶⁾ をエンドホストで行うことも十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。

本論文では、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 Emma と比較し、高いユーザ満足度を実現していることが確認できた。

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 18) などで行われている。文献 18) 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 18) 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と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いえる。

本論文は以下のように構成する。2 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 3 章で説明する。4 章で性能評価の結果と考察を与え、5 章でまとめと今後の課題を述べる。

2. 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を参加させる．

- (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，要求どおりにデータを配信してよいかを判断するアドミッション制御を行う．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どうしがなるべくリンクを共有しないような経路制御，(3)および(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ッション制御およ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)のアドミッション処理に着目し，その詳細について3章で述べる．

2.1 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（たとえば会議アプリケーションにおける1つの会議）をセッションと呼び，セッションごと独立にオーバ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など）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，あるオーバレイリンクで利用可能な帯域ユニット数をオーバレイリンク容量と呼ぶ．オーバ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．

2.2 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うる．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

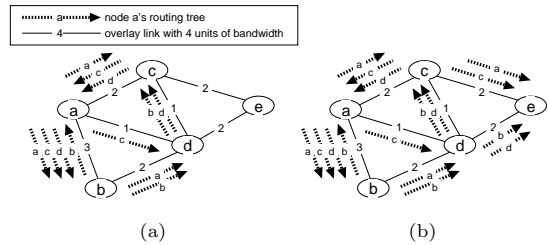


図2 ノードの参加にともなう経路木の拡張例

Fig. 2 Expansion of routing trees for new node e.

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ーと呼ぶ．セッションポリシー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．たと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，経路木どうし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* およびノード *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* および *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* および *c* の経路木には *c-e* を介して，ノード *b* および *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．

2.3 ストリーム要求のアドミッション制御

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

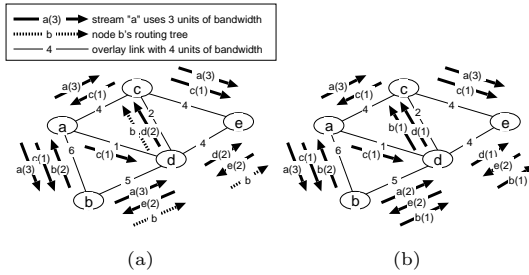


図 3 アドミッション制御例

Fig. 3 Admission control example.

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。たとえば文献 16) 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 (たとえばノード *c* はストリーム *a* および *d* をそれぞれ 3 および 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 3 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。たとえば、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* および *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* およびノード *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たかだか 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。

2.4 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、3 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. ストリーム配信時の QoS 制御

本章では、2.3 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。

3.1 ユーザ利得およびユーザ損失関数

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)

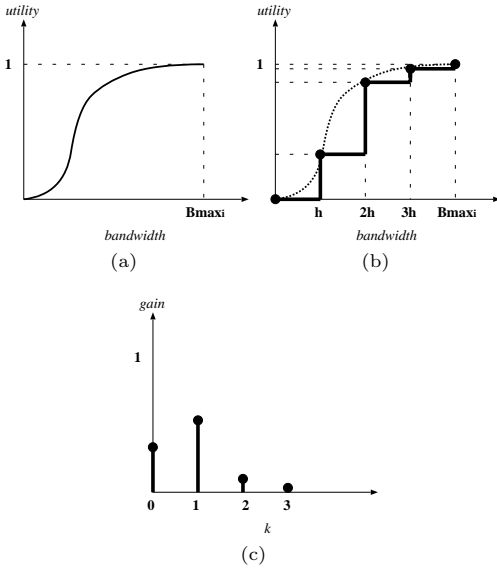


図 4 ユーティリティ関数からのユーザ利得関数の導出

Fig. 4 (a) Utility function, (b) Obtaining utility by every unit of bandwidth h and (c) Gaining function $gain_i(k, v)$.

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 ごとに利得およ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 17) 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および損失を定義する方法を述べる。

文献 17) 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リティ関数に対し、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ともな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、たとえば 2Mbps の MPEG2 と 512kbps 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、たとえば $h = 512\text{kbps}$ 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 512kbps 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。

3.2 ユーザ利得およびユーザ損失に基づいたアドミッション制御

3.2.1 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 およびノード 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 メッセージを受け取ったときに $Loss_i(v)$ 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。

たとえば、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 および 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。

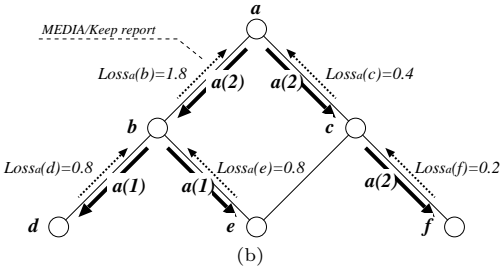
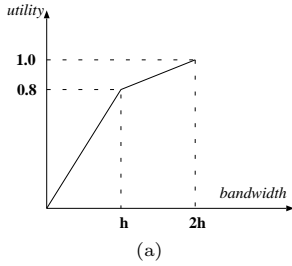


図5 MEDIA/Keep メッセージの転送
Fig. 5 Forwarding MEDIA/Keep messages.

3.2.2 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 がすで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、2.3 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。

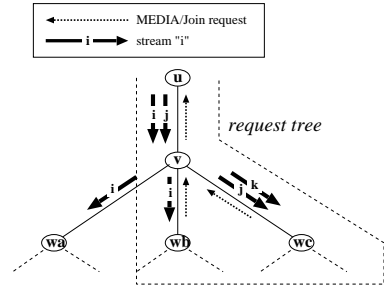


図6 $MinLoss_i(v)$ の計算
Fig. 6 Calculation of $MinLoss_i(v)$.

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

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 および 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

$$\begin{aligned} Gain_i(v) &= gain_i(v, 0) \\ &+ \sum_w Gain_i(w) \end{aligned} \quad (5)$$

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。

3.3 議 論

3.3.1 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 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うる。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もな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。

3.3.2 オーバレイリンク容量の調整

オーバーレイリンク容量はリンクの両エンドノード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が、この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うえで、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。

3.3.3 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(それはたとえばセッションポリシ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 MEDIA/Keep report を使い、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。

4. 性能評価

オブジェクト指向スクリプト言語 Ruby³⁾ 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。

4.1 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 200 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 6 Mbps, 50 Mbps, 100 Mbps 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 1 Mbps とし、ノードは 256 kbps ごと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 256 kbps とした。オーバーレイノード数が約 55 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 帯域ユニットを超えないようにした。各ストリームのユーザ利得/損失関数は 3.1 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。

なお、実際のビデオ会議を想定し、セッション開始直後には多数のノードが参加し、ノード数が安定した後はランダムに選択された 10% のノード(おおよそ 5 から 6)が参加および離脱を繰り返すシナリオを用いた。シミュレーション時間にもなうユーザ数の変動は図 11 の棒グラフで示されているため参照されたい。

4.2 測定項目

本論文で評価する項目は以下のとおりである。

- (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) ユーザ利得の時間変動

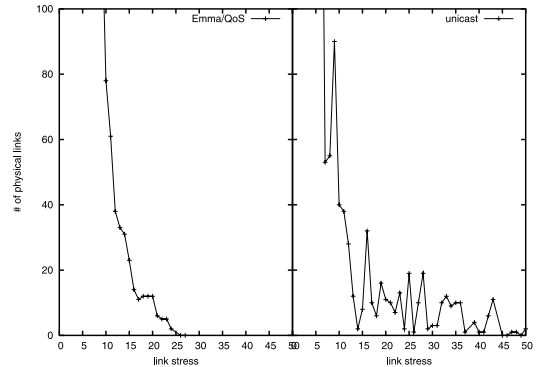
リンクストレスとパスストレッチは他の ALM でも多く用いられており、これらには相関関係があると考えられる（たとえばリンクストレスを高くすればパスストレッチは短くなり、ユニキャストに近づく）。しかし一般に対象アプリケーションも異なる既存の ALM と比較することは容易ではないため、本論文では Emma/QoS におけるこれらの値が、実用的に妥当でかつ適切なバランスを達成しているかを測定した。なお、1 つの指標としてユニキャストを用いた。

また、ツリーストレスは Emma/QoS（および 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 ではストリーム間なるべく帯域が競合しないよう、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。

同様にユーザの要求満足率およ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、そのような制御をまったく行わない到着順方式（First-come-first-serve、以下 FCFS）と比較した。なお、ユーザ利得に関しては、Emma/QoS のユーザ利得/損失関数から Emma の優先度であるプリファレンス値を設定し、その充足度を測定して比較することで Emma/QoS の優位性を確認した。

4.3 リンクストレスとパスストレッチの評価

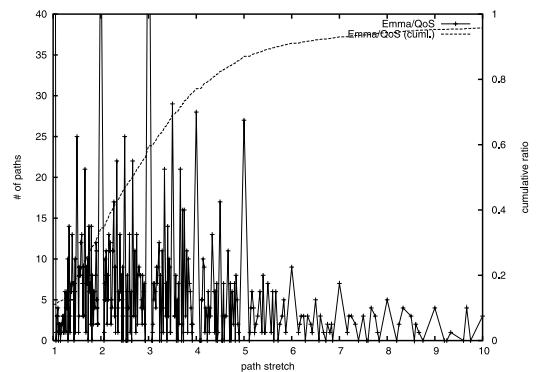
一定数の参加・離脱が繰り返された 40 シミュレーション時間後（図 11 参照）のリンクストレスおよびパスストレッチを測定した。



55 users (200 node networks)

図 7 リンクストレス分布

Fig. 7 Distribution of link stresses.

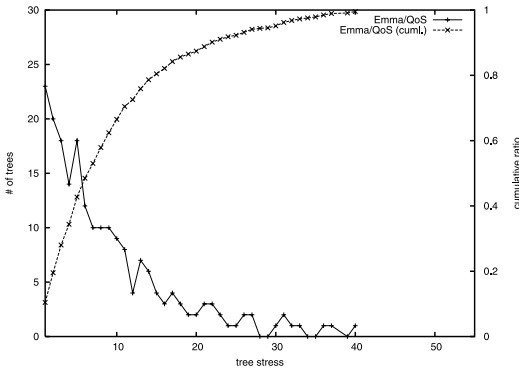


55 users (200 node networks)

図 8 パスストレッチ分布

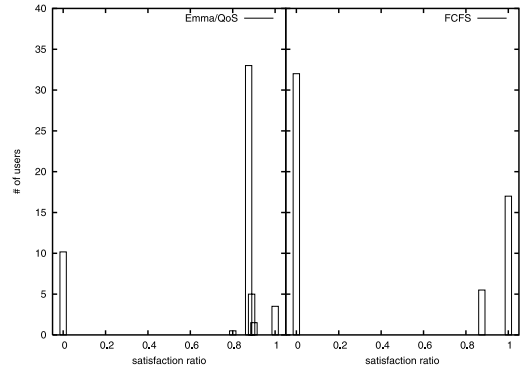
Fig. 8 Distribution of path stretches.

図 7 にリンクストレスの分布を示す。x 軸はリンクストレスの値を、y 軸はその値を持つ実リンクの数を表す。Emma/QoS では、最大のリンクストレスがユニキャストの場合の半分に収まっており、リンクストレスの削減効果は十分現れていると考えられる。一方で、図 8 にパスストレッチの分布とその累積分布を示す。近年のインターネット環境では、国内のエンド間遅延は数十ミリ秒程度である。たとえばこの 4 倍程度の遅延を仮定してもたかだか数百ミリ秒程度であり、ビデオ会議のようなインタラクティブアプリケーションにおいて十分実用的な値といえる。グラフより、Emma/QoS における 80% の経路のパスストレッチは 4 以下で収まっている（すなわち実ネットワーク上の遅延の 4 倍以内に収まっている）ことが読み取れ、Emma/QoS が実用的な遅延を実現できることが分かった。



55 users (200 node networks)

図 9 ツリーストレス分布
Fig. 9 Distribution of tree stresses.



55 users (200 node networks)

図 10 ユーザ要求満足率分布
Fig. 10 Distribution of users' satisfaction ratios.

4.4 ツリーストレスの評価

40 シミュレーション時間後のツリーストレスの分布および累積分布を図 9 に示す。複数の映像ストリームを配信するビデオ会議などでは、ツリーストレスが特定のリンクに集中することなく一様に分布していることが望ましい。累積分布から、ツリーストレスの約 90% は最大ストレス値の半分程度とおおよそ一様に分布していると見なすことができ、重なりが少ない経路木構築の効果が得られているといえる。

4.5 ユーザ要求満足率の評価

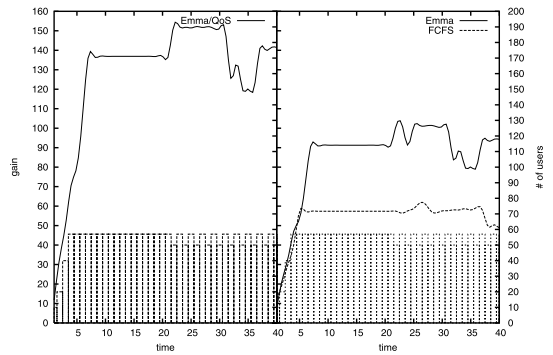
ノードごと、そのノードが送信した要求のユーザ利得総和に対する受け入れられた要求のユーザ利得総和の比をユーザ要求満足度と定義し、この値を測定した。図 10 に値の分布を示す。なお、リンクに空きがある限り要求順に受け入れ、空きがない場合は受け入れない到着順方式 (FCFS) と比較した。

結果より、FCFS 方式では半分以上のノードの要求満足率が 0 であり、ストリームを受信できなかった。一方、Emma/QoS ではほとんどのノードが 0.8 以上の要求満足率を達成できている。

ビデオ会議システムなどでは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いえる。

4.6 ユーザ利得値の時間変動の測定

最後に、シミュレーション時間におけるユーザ利得値総和の変動を測定した。その結果を図 11 に示す。比較対象として Emma および FCFS を用いた。セッション開始段階、ノードの参加・離脱時のいずれもユーザ利得は Emma、FCFS いずれと比較しても高い値を達成していることが分かった。



55 users (200 node networks)

図 11 ユーザ利得の変動
Fig. 11 Variation of user's gain.

5. ま と め

本論文では、オーバーレイネットワークにおける複数ビデオの同時配信時に QoS を分散制御で実現する ALM プロトコル Emma/QoS を提案した。

オーバーレイネットワーク上の帯域はエンド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である。

参 考 文 献

- 1) Diot, C., Dabbous, W. and Crowcroft, J.: 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) Calvert, K.L., Doar, M.B. and Zegura, E.W.: Modeling Internet Topology, *IEEE Communications Magazine*, pp.160–163 (1997).
- 3) Ruby Home Page.
<http://www.ruby-lang.org/en/>
- 4) Chu, Y.-H., Rao, S.G., Seshan, S. and Zhang, H.: Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture, *Proc. ACM SIGCOMM*, pp.55–67 (2001).
- 5) Chu, Y.-H., Rao, S.G. and Zhang, H.: A Case for End System Multicast, *Proc. ACM SIGMETRICS*, pp.1–12 (2000).
- 6) Pendarakis, D., Shi, S., Verma, D. and Waldvogel, M.: ALMI: An Application Level Multicast Infrastructure, *Proc. 3rd Usenix Symp. on Internet Technologies & Systems*, pp.49–60 (2001).
- 7) Jannotti, J., Gifford, D.K., Johnson, K.L., Kaashoek, M.F. and O’Toole, J.W.: Overcast: Reliable Multicasting with an Overlay Network, *Proc. 4th Usenix Symp. on Operating Systems Design and Implementation (OSDI)*, pp.197–212 (2000).
- 8) Roca, V. and El-Sayed, A.: A Host-Based Multicast (HBM) Solution for Group Communications, *Proc. 1st IEEE Int. Conf. on Networking (ICN’01)*, pp.610–619 (2001).
- 9) Cohen, R. and Kaempfer, G.: A Unicast-based Approach for Streaming Multicast, *Proc. IEEE INFOCOM 2001*, pp.440–448 (2001).
- 10) Ratnasamy, S., Handley, M., Karp, R. and Shenker, S.: Application-level Multicast using Content-Addressable Networks, *Proc. 3rd Int. Workshop on Networked Group Communication (NGC’01)*, pp.14–29 (2001).
- 11) Chawathe, Y., McCanne, S. and Brewer, E.A.: RMX: Reliable Multicast for Heterogeneous Networks, *Proc. IEEE INFOCOM 2000*, pp.795–804 (2000).
- 12) Zhuang, S.Q., Zhao, B.Y., Joseph, A.D., Katz, R.H. and Kubiatowicz, J.: Bayeux: An Architecture for Scalable and Fault-tolerant Wide-area Data Dissemination, *Proc. ACM NOSS-DAV 2001*, pp.11–20 (2001).
- 13) Francis, P.: Yoid: Extending the Internet Multicast Architecture, *Unrefereed Report* (2002).
<http://www.isi.edu/div7/yoid/>
- 14) Baccelli, F., Kofman, D. and Rougier, J.L.: Self Organizing Hierarchical Multicast Trees And Their Optimization, *Proc. IEEE INFOCOM2001*, pp.1081–1089 (2001).
- 15) Nakamura, Y., Yamaguchi, H., Hiromori, A., Yasumoto, K., Higashino, T. and Taniguchi, K.: On Designing End-user Multicast for Multiple Video Sources, *Proc. 2003 IEEE Int. Conf. on Multimedia & Expo*, Vol.III, pp.497–500 (2003).
- 16) Yeadon, N., García, F., Hutchison, D. and Shepherd, D.: Filters: QoS Support Mechanisms for Multipeer Communications, *IEEE Journal on Selected Areas in Communications*, Vol.14, No.7, pp.1245–1262 (1996).
- 17) Shenker, S.: Fundamental Design Issues for the Future Internet, *IEEE Journal on Selected Areas in Communications*, Vol.13, No.7, pp.1176–1188 (1995).
- 18) Xu, D. and Nahrstedt, K.: Finding Service Paths in a Media Service Proxy Network, *Proc. SPIE/ACM Conf. on Multimedia Computing and Networking (MMCN2002)*, pp.171–185 (2002).

(平成 15 年 5 月 19 日受付)

(平成 15 年 12 月 2 日採録)



中村 嘉隆 (学生会員)

平成 14 年大阪大学基礎工学部情報科学科卒業。現在、同大学大学院情報科学研究科博士前期課程在学中。分散協調アプリケーションの実現に関する研究に従事。



山口 弘純 (正会員)

平成 6 年大阪大学基礎工学部情報工学科卒業。平成 10 年同大学大学院基礎工学研究科博士後期課程修了。同年オタワ大学客員研究員。平成 11 年大阪大学大学院基礎工学研究科助手。平成 14 年より同大学院情報科学研究科助手 (工学)。分散システムや通信プロトコルの設計および実装に関する研究に従事。



廣森 聡仁(学生会員)

平成 11 年大阪大学基礎工学部情報科学科中退。平成 13 年同大学大学院基礎工学研究科博士前期課程修了。同年同大学院基礎工学研究科博士後期課程入学。現在、同課程在学中。平成 15 年より日本学術振興会特別研究員。ネットワーク QoS の研究に従事。



安本 慶一(正会員)

平成 3 年大阪大学基礎工学部情報工学科卒業。平成 7 年同大学大学院基礎工学研究科博士後期課程退学後、滋賀大学経済学部助手。平成 9 年モントリオール大学客員研究員。平成 14 年より奈良先端科学技術大学院大学情報科学研究科助教授。博士(工学)。分散システム、マルチメディア通信システムに関する研究に従事。IEEE/CS 会員。



東野 輝夫(正会員)

昭和 54 年大阪大学基礎工学部情報工学科卒業。昭和 59 年同大学大学院基礎工学研究科博士後期課程修了。同年同大学助手。平成 2 年、6 年モントリオール大学客員研究員。現在、同大学大学院情報科学研究科教授。博士(工学)。分散システム、通信プロトコル等の研究に従事。電子情報通信学会、ACM 各会員。IEEE Senior Member。



谷口 健一(正会員)

昭和 40 年大阪大学工学部電子工学科卒業。昭和 45 年同大学大学院基礎工学研究科博士課程修了。同年同大学助手。現在、同大学大学院情報科学研究科教授。博士(工学)。この間、計算理論、ソフトウェアやハードウェアの仕様記述・実現・検証の代数的手法および支援システム、関数型言語の処理系、分散システムや通信プロトコルの設計・検証法等に関する研究に従事。