

ネットワーク支援による階層化マルチキャストレート制御手法

中内清秀[†] 森川博之[†] 青山友紀[†]

インターネットのような不均一なネットワーク上でビデオなどのリアルタイムストリーム型データのマルチキャストを行う場合、階層符号化を用いた階層化マルチキャストが有効である。階層化マルチキャストにおいては、階層符号化の性質上基本階層のパケットロスがアプリケーションの品質を大きく劣化させるため、輻輳に迅速に対応できるレート制御を行うことが要求される。同時に、セッションの開始時刻によらないセッション間の公平な帯域割当の保証も要求される。しかしながら、従来の受信者駆動型レート制御ではこのような要求を満たすことができない。そこで、本稿では階層化マルチキャスト対応ルータによる支援により上記の要求を満足できるネットワーク支援型レート制御手法を示す。

A Rate Control Scheme for Layered Multicast with Network Support

KIYOHIDE NAKAUCHI,[†] HIROYUKI MORIKAWA[†]
and TOMONORI AOYAMA[†]

Layered Multicasting with layered source coding scheme enables efficient distribution of realtime streaming media over heterogeneous networks such as Internet. Since packet losses in the base layer lead great degradation of an application's QoS in layered multicast, rate control for layered multicast is required to cope with congestion quickly. At the same time, the rate control must assure fair bandwidth allocation among competitive layered multicast sessions at a congested link independent of their start time. Existing end-to-end rate control schemes, however, cannot meet these requirements. In this paper, we present efficient network supported rate control scheme for layered multicast which meets the requirements described above.

1. ま え が き

音声や映像などのストリームデータをネットワークを介してリアルタイムに多数の受信者に同時に配信する放送型アプリケーションや VoD (Video on Demand)、テレビ会議アプリケーションでは、マルチキャストを用いることによりデータ転送を効率良く行うことができる。この時、それぞれの受信者が利用可能帯域及び端末の性能に適した多様なレートで受信できることを保証する必要がある。しかしながら、単なるマルチキャストだけではこのようなネットワークの多様性に対処することができない。

ネットワークの多様性に対処するマルチキャスト手法として、受信者を利用可能帯域別にグループ化し、同一データをそれぞれのグループに対して異なるレートで送信するサイマルキャスト方式⁹⁾、階層符号化を用いて一つのセッションの送信データを複数のストリームに分割して送信する階層化マルチキャスト方式などが提案されている。本研究においては、帯域利用効率という点では多様性に対応しつつ同一のデータを複数回送信する必要がない階層化マルチキャスト方式が最も優れているため、階層化マルチキャストを対象とする。

階層化マルチキャストのレート制御においては、まず複数の階層化マルチキャストセッションが輻輳リンクを共有する場合にセッション間の公平な帯域割当が行われることが要求される。また、階層符号化の性質上、基本階層のパケットロスはアプリケーションの品質

を大きく劣化させる^{10),11)}ことから、レート制御にはネットワーク状態に関係なく基本階層のパケットロスを最小限に抑制することも要求される。なお、本稿では階層化マルチキャストにおいてそれぞれの受信者の受信階層数を決定することをレート制御と呼ぶ。

レート制御に関する従来の研究においては、レート制御の効率性の向上を目的とした手法^{1),2),5),7)}や階層化マルチキャストセッションと TCP セッションとの公平性を考慮した手法^{4),6)}が提案されているが、階層化マルチキャストセッション間の公平性を考慮した手法に関する研究はあまり行われていない。

また、ルータのパッファ管理手法として全フローを対象としてランダム廃棄を行う Drop Tail, RED などが実装されたランダム廃棄モデルのネットワークにおいては、輻輳時に基本階層のパケットロスが生じてしまうため、輻輳に迅速に対応しなければならない。しかしながら、従来のレート制御手法^{1)~6)}では、輻輳に迅速に対応することができない。

さらに、フローを考慮した優先廃棄を行うパッファ管理手法が実装された優先廃棄モデルのネットワークにおいても、ノード間で協調することなく優先廃棄のみによってレート制御⁹⁾を行った場合、基本階層のパケットロスは防止することができず、非輻輳リンクにおいては下流の輻輳リンクで廃棄されるパケットまで転送されるため帯域の浪費を招いてしまい、公平な帯域割当を保証することはできない。

このように従来のレート制御手法は階層化マルチキャストへの要求を満たすことができない。これは、従来のランダム廃棄モデルのネットワークを前提としたレート制御においては、受信者駆動型であり、さらにセッション間の公平性が考慮できないからであり、

[†] 東京大学工学系研究科
School of Engineering, University of Tokyo

従来の優先廃棄モデルのネットワークを前提としたレート制御においては、ノード間の協調がないからである。以上のような観点から、輻輳に対する迅速な対応及び公平な帯域割当を可能とする階層化マルチキャストを実現するためには、ノード間で協調しつつ優先廃棄を行うようなネットワーク支援機構が必要であると考えられる。そこで、本稿ではネットワーク内に分散配置された階層化マルチキャスト対応ルータによるネットワーク支援型レート制御手法を示す。

2. 階層化マルチキャストのレート制御

2.1 受信者駆動型レート制御

従来の階層化マルチキャストは受信者駆動型のレート制御を行う。受信者駆動型レート制御は利用可能帯域の推定方法により、join experiment 型^{1)~5)}と帯域測定型^{6),7)}の2つに分類できる。

join experiment 型レート制御では、受信者は試験的に上位の階層を受信する「join experiment」とパケットロス率を用いて相対的に利用可能帯域を推定する。例えば、上位階層の受信を希望する場合、その階層に対して join して一定時間試験的に受信する(join experiment)。この期間中にパケットロスを検出しなければ、その階層の受信を継続する。

一方、帯域測定型レート制御では、受信者がパケットロス率などの現在の受信状態から方程式ベースで利用可能帯域を測定する。そして、総受信レートが測定帯域以下になるよう受信階層数を決定する。帯域測定方法としては、TCP の平均スループットとパケットロス率、RTT、MTU の関係式を用いる手法⁶⁾や、Packet Pair を用いる手法⁷⁾などが提案されている。

2.2 受信者駆動型レート制御の課題

上述の従来のレート制御手法は、輻輳に対する迅速な対応、及び輻輳リンクにおける公平な帯域割当という階層化マルチキャストに対する要求を満たすことができない。

従来の受信者駆動型レート制御が輻輳に対して迅速に対応することができない理由として、まず受信者のタイムスケール制御により、パケットロスを検出してから輻輳と判断するまでに時間を要する点が挙げられる。join experiment 型の RLM¹⁾ を例にとると、輻輳を検出するまでの時間はタイマーの初期値に依存する。RLM ではパケットロス検出後タイマーが切れた時に測定したパケットロス率が閾値を超えることにより輻輳を検出する。このタイマーの初期値は標準で数秒に設定される。基本的には PLM⁷⁾を除く他のレート制御手法^{2)~6)}においても同じ議論が成り立つ。

輻輳に迅速に対応できないもうひとつの理由として、一般的に知られる IGMP における leave 時の遅延が挙げられる。IGMP では受信者からの LeaveGroup メッセージの受信待ち時間が標準で数秒に設定される。このように、従来のレート制御手法では輻輳に迅速に対応できないため、基本階層のパケットロスを招いてしまう。さらに、従来のレート制御においてはパケットロス率が閾値以下であれば輻輳と認識しないが、パケットロス率がわずかであってもランダム廃棄が行われているため、基本階層のパケットロスが発生してしまう。

また、受信者駆動型レート制御は、複数の階層化マルチキャストセッションの存在を意識せず、セッション毎に独立に行われるため、文献³⁾の手法を除いてはセッションの獲得帯域は受信者の join 時刻に依存してしまい、輻輳リンクにおいて階層化マルチキャスト

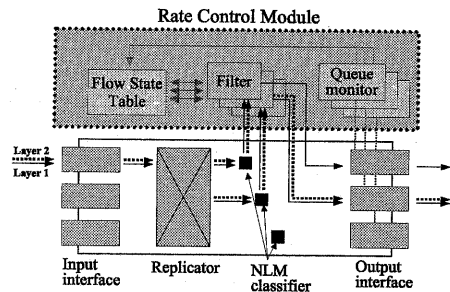


図1: LMRのアーキテクチャ

Fig 1: Architecture of LMR

セッション間の公平な帯域割当を保証することができない^{3),13)}。なお、ここでは multi-rate max-min fair¹²⁾であることを公平性の定義とする。

3. ネットワーク支援型レート制御手法

2.で述べた問題は、従来のレート制御が受信者駆動であり、セッション間の公平性が考慮できないことに起因する。従って、輻輳に対する迅速な対応及びセッション間の公平な帯域割当を実現するためには階層化マルチキャストセッションを統一的に管理するネットワーク支援機構が必要であると考えられる。そこで本節においてはネットワーク支援による階層化マルチキャスト (NLM: Network-supported Layered Multicast) のためのレート制御手法を示す。

3.1 ネットワークモデル

本稿では NLM の初期的性能評価を行うため、まず以下のようなネットワークモデルを前提とする。まず、サービスモデルはベストエフォート、ルータに実装されるバッファ管理方式は Drop Tail を想定する。さらに、ユニキャストとマルチキャストに割り当てる帯域を CBQ などにより静的に行うモデルを考える。従って、TCP などの他のユニキャストトラフィックとの公平な帯域割当に関しては議論しない。さらに、符号化方式としては各階層のレートが CBR となるような符号化方式を採用するものとし、すべてのセッションの基本階層及び拡張階層のレートは等しいものとする。

また、NLM においては送信者は階層符号化を行った後すべての階層を同一マルチキャストアドレス宛に送信する。これにより、すべての階層が同一のルートを通過することが保証され、同一のルートを通過しない場合に従来のレート制御が効率良く機能しないという問題¹⁴⁾を回避することができる。送信者は、DATA パケットにそのパケットが属する階層を示す領域を設ける。

3.2 システムモデル

NLM では、ネットワーク内に分散するマルチキャストルータの一部を階層化マルチキャスト対応ルータ (LMR: Layered-Multicast-capable Router) に置換する。LMR は通常のマルチキャストルータに、階層化マルチキャストフローに対する付加的処理機構を組み込んだルータである。LMR に付加される機能は以下の通りである。

- (1) フロー状態表によるセッション情報の保持
- (2) アウトプットインターフェース (oif) における輻輳の検出
- (3) NLM 識別器 (classifier) による NLM パケットの選別

MGroup	Source	Session ID S_i	PrevID	oif I_j	S_i	$L_{max}(I_j, S_i)$	$L_{cur}(I_j, S_i)$
S1	Src1	1	LMR1	1	1	2	2
S2	Src2	2	LMR2	2	2	3	3
S3	Src3	3	LMR3	3	3	4	4
S1	Src4	4	LMR4	2	1	2	2
...	...	...	...	4	1	1	1
...	...	...	...	3	4	3	3
...	...	...	...	...	...	...	...

図 2: フロー状態表
Fig 2: Flow State Table

- (4) 特定の階層に属するパケットのフィルタリング
- (5) NLM ノード (送信者, 受信者及び LMR) 間のシグナリング

まず, LMR のアーキテクチャを図 1 に示す。LMR は図 1 に示すように, 従来のマルチキャストルータにレート制御モジュール及び NLM 識別器を付加した構成になる。

レート制御モジュールはフロー状態表, oif 毎のキューモニター及びパケットフィルタを具備する。フロー状態表は, oif 毎にその oif を通過する階層化マルチキャストセッションに関する情報を保持する。キューモニターは平均キュー長を計算する。また, パケットフィルタはフロー状態表をもとに特定の階層に属するパケットを選別してフィルタリングを行う。さらに, LMR にはシグナリング機能が付加されている。

また, LMR には oif 側に NLM 識別器が付加される。この NLM 識別器は NLM パケットだけを選別してレート制御モジュールに送り込む。フィルタを通過した NLM パケットは再び従来のプロセスラインに戻される。

以上のようなアーキテクチャにより, 従来のマルチキャストルータのアーキテクチャの変更を最小限に抑えたとともに, ルーティング及びフォワーディングに関する処理能力の低下を防ぐことができる。

NLM では, DATA, SESS, ADD_REQ, 及び DROP_REQ の 4 種類のタイプのパケットを使用する。これらのパケットは, マルチキャストアドレスフィールド MGroup, 送信者アドレスフィールド Source, 階層番号フィールド Layer を持ち, さらに SESS は前ホップ LMR アドレスフィールド PrevID を持つ。ADD_REQ, DROP_REQ はシグナリングにおいて, それぞれ指定した階層の転送開始要求, 転送中止要求に用いられる。また, SESS はセッション制御に使用される。

各 LMR が持つフロー状態表には, 図 2 に示すようにマルチキャストアドレス MGroup, 送信者アドレス Source, MGroup と Source により一意に決定するセッション ID S_i , 及び直上の LMR のアドレス PrevID が保持されている。本稿ではマルチキャストアドレス MGroup 及び送信者アドレス Source により一意に決まるホストの集合をセッションと呼ぶ。フロー状態表には, さらに oif 毎にその oif を通過する各セッションについて, 送信可能である最上位階層の番号 L_{max} , 及びフィルタリングされずに通過できる最上位転送階層の番号 L_{cur} が保持される。本稿では, oif I_j におけるセッション S_i の L_{max} , 及び L_{cur} の値をそれぞれ $L_{max}(I_j, S_i)$, $L_{cur}(I_j, S_i)$ と表すことにする。フロー状態表は, ADD_REQ / DROP_REQ 及び SESS により更新される。

3.3 NLM のレート制御の概要

NLM では, 輻輳を検出した LMR がフィルタリングを行うことにより輻輳に対する迅速な対応を可能と

し, さらにフィルタリングの対象となる階層をフロー状態表をもとに選択するとともに LMR 間のシグナリングを行うことにより輻輳リンクにおけるセッション間の公平な帯域割当を可能とする。

LMR は oif 単位でレート制御を行う。oif I_j においてセッション S_i のパケットが転送される場合, そのパケットの属する階層が $L_{cur}(I_j, S_i)$ より大きければフィルタリングされる。従って, L_{cur} の値を増減させることによりレート制御が実現される。

平均キュー長が閾値 (q_{max}) を超えることにより輻輳発生を検出した LMR は, 輻輳が生じた oif I_j において, 最上位転送階層番号 ($L_{cur}(I_j, S_i)$) が最大であるセッション S_{i_0} を選択し, S_{i_0} の最上位転送階層に属するパケットのフィルタリングを開始する。つまり, $L_{cur}(I_j, S_{i_0})$ の値を 1 減少させる。また, I_j において平均キュー長が閾値 (q_{min}) 以下になることにより輻輳解消を検出した場合, 最上位転送階層番号 ($L_{cur}(I_j, S_i)$) が最小であるセッション S_{i_1} を選択し, S_{i_1} の最上位転送階層より 1 つ上位の階層に属するパケットのフィルタリングを解除する。つまり, $L_{cur}(I_j, S_{i_1})$ の値を 1 増加させる。本稿では前者のプロセスを DROP, 後者のプロセスを ADD と呼ぶ。フィルタリングされなかったパケットはそのままキューイングされる。以上に述べたように, 輻輳を検出した LMR がフィルタリングを行うことにより, NLM は輻輳に対する迅速な対応を可能とする。

さらに, LMR は DROP を行った時, 新たにフィルタリング開始の対象となった階層を他のどの oif から転送していない場合, その階層の受信は不要となる。従って, 帯域の浪費を防止するために転送が不要になった階層の転送中止を DROP_REQ を用いて直上の LMR に要求する。逆に, ADD を行った時, 新たにフィルタリング解除の対象となった階層を上流の LMR から受信していない場合, その階層の転送開始を ADD_REQ を用いて直上の LMR に要求する。このようなシグナリングにより, NLM は帯域の浪費を防止することができる。

以上のように, L_{cur} の値をもとにしたフィルタリング開始 / 解除の対象となる階層の選択, 及びシグナリングにより, セッション間の公平な帯域割当を実現し, multi-rate max-min fairness¹²⁾ を保証することができる。

また, 各受信者は join 後, 送信者とのパス上でフィルタリングされることなく受信できる階層数を把握し, その階層を受信するまで一定時間間隔で受信階層数を上げていく。それ以後は自ら受信階層数を上げることはなく, LMR が決定したレートで受信する。このような制御により, 各受信者は join 後, 受信レートを短時間でネットワーク状態に適した値に収束させることができる。

3.4 フィルタリングアルゴリズム

LMR は輻輳を検出した oif においてフィルタリングを行う。まず, LMR が使用するパラメタを表 1 に示す。

LMR は常に各 oif における平均キュー長 q_{len} を, その時点でのキュー長と直前の平均キュー長から q_{weight} を用いて相加的に求めている。キュー長ではなく平均キュー長を用いる理由は, 一時的な輻輳に対応して不要な DROP を行うことを防止するためである。

LMR は最初に $q_{len} \geq q_{max}$ となることにより輻輳を検出することを契機として, 輻輳を検出した oif において DROP を行い, レート制御を開始する。

表 1: LMR のパラメータ
Table 1: Parameters for LMR

Variables for LMR	
<i>add.intvl</i>	time interval for the next ADD
<i>drop.intvl</i>	time interval for the next DROP
<i>detect.period</i>	period for detecting congestion
<i>qlen</i>	average queue length
<i>qmax</i>	threshold for congested state
<i>qmin</i>	threshold for unloaded state
<i>qweight</i>	weight for calculating <i>qlen</i>
α	increase parameter (> 1)
β	decrease parameter (< 1)

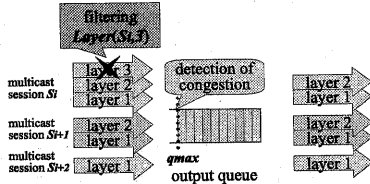


図 3: フィルタリングの様子
Fig 3: Overview of Filtering

まず、DROP プロセスについて述べる。oif I_j において $qlen \geq qmax$ となり、かつ最後に DROP を実行した時刻から一定時間 *drop.intvl* だけ経過していれば LMR は DROP を行う。この時、フロー状態表の oif I_j のエントリから、DROP の対象となるべき階層を以下のプロセスに従って探す。なお、本稿においてはセッション S_i の L 番目の階層のことを便宜上 $Layer(S_i, L)$ と表現し、DROP の対象となった階層の番号を L_{dt} と表す。

- (1) oif I_j において $L_{cur}(I_j, S_i) \rightarrow max$ となるセッション S_{i_0} を選択する。条件を満たすセッションが複数存在する場合は、適宜セッション S_{i_0} を選択する。
- (2) ($L_{dt} =$) $L_{cur}(I_j, S_{i_0}) \geq 2$ であれば $Layer(S_{i_0}, L_{dt})$ を DROP の対象階層とする。条件を満たさなければ S_{i_0} を除いた S_i について 1, 2 を繰り返す。

DROP の対象階層 $Layer(S_{i_0}, L_{dt})$ が決定されれば、 $L_{cur}(I_j, S_{i_0})$ の値を 1 減少させることにより、 $Layer(S_{i_0}, L_{dt})$ をフィルタリングの対象に追加する。

また、LMR は DROP 後一定時間 *drop.intvl* 待機し、 $Layer(S_{i_0}, L_{dt})$ のフィルタリングの効果が $qlen$ に反映されるのを待つ。

次に、ADD プロセスについて述べる。oif I_j において $qlen \leq qmin$ となり、かつ最後に ADD を実行した時刻から *add.intvl* だけ時間が経過していれば LMR は ADD を行う。この時、フロー状態表の oif I_j のエントリから、以下のプロセスにより ADD の対象となるべき階層を決定する。なお、本稿においては ADD の対象となった階層の番号を L_{at} と表すことにする。

- (1) oif I_j において $L_{cur}(I_j, S_i) \rightarrow min$ となるセッション S_{i_1} を選択する。条件を満たすセッションが複数存在する場合は、適宜セッション S_{i_1} を選択する。
- (2) $L_{cur}(I_j, S_{i_1}) < L_{max}(I_j, S_{i_1})$ であれば S_{i_1} の階層 $L_{cur}(I_j, S_{i_1}) + 1 (= L_{at})$ を ADD の対象階層とする。条件を満たさなければ S_{i_1} を除いた S_i について 1, 2 を繰り返す。

ADD の対象階層 $Layer(S_{i_1}, L_{at})$ が決定されれば、 $L_{cur}(I_j, S_{i_1})$ の値を 1 増加させることにより、 $Layer(S_{i_1}, L_{at})$ をフィルタリングの対象から除く。

また、LMR は ADD 後一定時間 *add.intvl* 待機し、

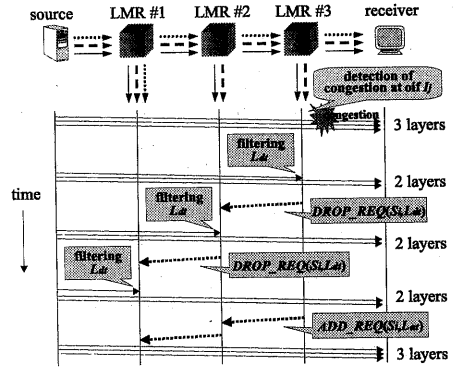


図 4: シグナリングの様子
Fig 4: Overview of Signaling

$Layer(S_{i_1}, L_{at})$ のフィルタリング解除の効果が $qlen$ に反映されるのを待つ。

さらに、NLM では ADD 及び DROP の繰り返しによるレート振動を防止するために *add.intvl* の値を適応的に変化させる。つまり、LMR は ADD 後 *detect.period* 以内にネットワーク内のいずれかの場所において輻輳が発生したことを検出した場合には、実行した ADD が輻輳の原因であると考え、*add.intvl* を $\alpha (> 1)$ 倍して大きくし、輻輳を検出しなかった場合には *add.intvl* を $\beta (< 1)$ 倍して小さくする。

3.5 シグナリング

シグナリングの目的は、DROP により転送が不要になった階層の転送を防止することである。不要な階層の転送が防止されれば、上述のフィルタリングアルゴリズムにより multi-rate max-min fairness が保証される。

シグナリングでは *ADD_REQ* パケットと *DROP_REQ* パケットが用いられる。これらのパケットは直上の LMR にユニキャストで送信される。ここでは便宜上 $Layer$ フィールドの値を L と表し、セッション S_i に属するそれぞれのパケットを *ADD_REQ*(S_i, L), *DROP_REQ*(S_i, L) と表すことにする。

oif I_j においてセッション S_i の階層 L_{at} を対象とした ADD、または L_{dt} を対象とした DROP を行った LMR は、「 $L_{at}(L_{dt}) < \max(L_{cur}(I_k, S_i))$, $k \neq j$ 」という条件を満たした場合に限り、それぞれ *ADD_REQ*(S_i, L_{at}) パケット、*DROP_REQ*(S_i, L_{dt}) パケットを生成し、一定時間の間、定期的に送信する。

LMR における *ADD_REQ* 及び *DROP_REQ* の転送条件は基本的には IGMP の *JoinMessage*, *LeaveMessage* とそれぞれ同じである。LMR が oif I_j から *ADD_REQ*(S_i, L) を受信した場合、その *ADD_REQ*(S_i, L) を直上の LMR に転送すべきかどうかを調べる。ここでは I_j 以外の oif についての $L_{cur}(I_k, S_i)$ ($k \neq j$) の最大値と L を比較する。前者の値の方が小さければ (条件 1-a)、当該 LMR は $Layer(S_i, L)$ を受信していないことになるので、この階層の転送を要求するために *ADD_REQ*(S_i, L) を直上の LMR に転送しなければならない。条件 1-a が満たされなければ、 $L_{cur}(I_j, S_i)$ を L に更新してからこのパケットを廃棄する。

条件 1-a が満たされた場合、LMR は次に L と $L_{cur}(I_j, S_i)$ を比較する。もし前者の方が大きければ (条件 1-b)、この *ADD_REQ* が $Layer(S_i, L)$ の転送を要求する最初の (最初に受信した) *ADD_REQ* であ

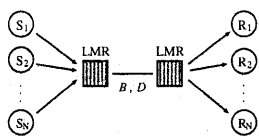


図 5: シミュレーショントポロジー

Fig 5: Simulation topology

ることを意味するので、 $L_{cur}(I_j, S_i)$ の値を L に更新し、 $ADD_REQ(S_i, L)$ を直上の LMR に転送する。条件 1-b を満たさなければ、重複する ADD_REQ とみなすことができるのでそのまま転送する。

次に、LMR が oif I_j から $DROP_REQ(S_i, L)$ を受信した場合、まずこの $DROP_REQ(S_i, L)$ を直上の LMR に転送すべきかどうかを調べる。LMR は ADD_REQ の場合と同様に I_j 以外の oif についての $L_{cur}(I_k, S_i)$ ($k \neq j$) の最大値と L を比較する。前者の値の方が小さければ (条件 2-a)、当該 LMR は $Layer(S_i, L)$ を oif I_j だけから転送していたことになるので、 $DROP_REQ(S_i, L)$ を直上の LMR に転送しなければならぬ。条件 2-b が満たされなければ、 $L_{cur}(I_j, S_i)$ を $L-1$ に更新してからこの $DROP_REQ$ を破棄する。

条件 2-a が満たされた場合、LMR は次に L と $L_{cur}(I_j, S_i)$ を比較する。もし、 L が $L_{cur}(I_j, S_i)$ の値以下であれば (条件 2-b)、このパケットが $Layer(S_i, L)$ の転送中止を要求する最初の (最初に受信した) $DROP_REQ$ であることを意味するので、 $L_{cur}(I_j, S_i)$ の値を $L-1$ に更新し、直上の LMR に転送する。条件 2-b を満たさなければ、重複する $DROP_REQ$ とみなすことができるのでそのまま転送する。以上のシグナリングの様子を図 4 に示す。

3.6 セッション制御

NLM では送信者が一定時間間隔 (ss_intvl) で送信する $SESS$ パケットを用いてセッション制御を行う。セッション制御の目的は、分散配置された LMR 間でフロー状態表の一貫性を保つことである。フィルタリングやシグナリングにより一部の LMR においてフロー状態表の L_{cur} の値が変更されるため、変更された LMR より下流の LMR に、その時点での最上位転送階層の番号を正確に把握させることにより L_{cur} の値について一貫性を保つことができる。

また、LMR が $SESS$ の $PrevID$ フィールドに順次自身のアドレスを書き込んでいくことにより LMR は直上の LMR のアドレスを把握することができる。

4. シミュレーションによる性能評価

NLM のレート制御の性能を、文献¹⁾で用いられているネットワークシミュレータ *ns* を用いて行った。シミュレーションは図 5 に示すトポロジーにおいて行った。一つの送信者と一つの受信者から構成されるセッションが複数存在し、ボトルネックリンクを共有している。なお、今回のシミュレーションでは IGMP における leave 遅延の影響は考慮しなかった。

主なシミュレーションパラメータは、キューサイズ = 20 [packets], $qmax = 15$ [packets], $qmin = 3$ [packets], $weight = 0.05$, $add_intvl_min = 5$ [s], $add_intvl_max = 80$ [s], $drop_intvl = 0.5$ [s], $detect_period = 5$ [s], $\alpha = 2.0$, $\beta = 0.75$ と設定し、セッション数 (N)、ボトルネックリンク帯域 (B [Mbps]) 及びボトルネックリンク遅延 (D [ms]) はシミュレーション毎に変化させた。また、階層符号化の最大階層数は 5 とし、各階層のレートは CBR で、基

本階層から順に 100, 100, 200, 400, 800 [kbps] とした。従って最高階層までの合計レートは 1.6 [Mbps] となる。

$N = 1$, $B = 1.5$, $D = 10$ とし、受信者が 20 [s] に join した時の NLM のレート制御の様子を図 6 に示す。 add_intvl が指数関数的に大きくなるのがわかる。

$N = 1$, $B = 2.0$, $D = 10$ とし、20 [s] にセッションを開始させ、100 [s] に 1.8 [Mbps] の妨害トラフィックをボトルネックリンクに発生させた時のスループットを図 7 に示す。NLM では 2 秒程度で最適レートに収束しているのに対し、RLM は 40 秒近く要している。さらに、この間のパケットロス率についても、NLM が 2 秒間だけ 10 % を記録したのに対し、RLM は 40 秒間、数十 % の高いパケットロス率を維持した。このように、たとえ巨大なバーストラフィックが発生するような状況においても、NLM は低パケットロス率を維持することができる。

$N = 1$, $B = 1.6$, $D = 10$ とした時、100 [s] から 200 [s] まで 0.333 [s] 間隔で測定したグッドプットを図 8 に示す。グッドプットとは実際に復号できるデータの量を表す。ここでは一定時間間隔において階層毎のパケットロス率を測定し、基本階層から順にパケットロス率が 20 [%] 以下で受信できた階層までの合計レートをグッドプットと定義する。図 8 に示すように、NLM ではグッドプットが 1.6 [Mbps] で一定に保たれるのに対し、RLM では大きく変動する。この結果より、パケットロス率が閾値以下であれば輻輳と認識しないレート制御手法は階層化マルチキャストには適さないと考えられる。

$N = 2$, $B = 1.7$, $D = 10$ とし、2 つのセッションをそれぞれ 20 [s] と 120 [s] に開始させた時のスループットを図 9 に示す。NLM ではセッション間の公平な帯域割当が実現されるのに対し、RLM では最初に開始したセッションが帯域の大部分を占有している。

$N = 3$, $B = 3.0$ とし、 D を 1 から 10000 [ms] まで変化した時、セッション毎に 1.0 [s] 間隔で測定したパケットロス率の最大値を図 10 に示す。NLM では遅延時間に関係なく最大パケットロス率が 2 % で一定であるのに対し、RLM では遅延時間が大きくなるにつれて最大パケットロス率も大きくなる。

$B = 1.0 \times N$, $D = 10$ とし、 N を 1 から 100 まで変化した時、セッション毎に 1.0 [s] 間隔で測定したパケットロス率の最大値を図 11 に示す。NLM ではセッション数に関係なく最大パケットロス率が 1 % 以下で一定であるのに対し、RLM ではセッション数が多くなるにつれて最大パケットロス率も大きくなる。

5. むすび

本稿では、輻輳に対して迅速に対応して基本階層のパケットロスを抑制し、かつ輻輳リンクにおいて階層化マルチキャストセッション間の公平な帯域割当を可能とする、階層化マルチキャストのためのネットワーク支援型レート制御手法を示した。

参考文献

- 1) S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven Layered Multicast". In *Proc. of ACM SIGCOMM '96*, Aug. 1996.
- 2) X. Li, S. Paul and M. Ammar, "Layered Video Multicast with Retransmissions (LVMR): Evaluation of Hierarchical Rate Control". In *Proc. of IEEE INFOCOM '98*, March 1998.
- 3) X. Li, S. Paul, and M. H. Ammar, "Multi-session Rate Control for Layered Video Multicast". In *Proc.*

of Multimedia Computing and Networking 1999, Jan. 1999.

- 4) L. Vicisano, L. Rizzo, and J. Crowcroft, "TCP-like congestion control for layered multicast data transfer". In *Proc. of IEEE INFOCOM'98*, March 1998.
- 5) L. Wu, R. Sharma, and B. Smith, "Thin Streams: An Architecture for Multicasting Layered Video". In *Proc. of NOSSDAV'97*, May 1997.
- 6) T. Tuletto, S.F. Parisi and J.-C. Bolot, "Experiments with a Layered Transmission Scheme over the Internet". INRIA Research Report No. 3296, Nov. 1997.
- 7) A. Legout and E. W. Beiersack, "Fast Convergence for Cumulative Layered Multicast Transmission Schemes". In *Proc. of ACM SIGMETRICS 2000*, June 2000.
- 8) S.Y. Cheung, M. Ammar, and X. Li, "On the Use of Destination Set Grouping to Improve Fairness in Multicast Video Distribution". In *Proc. of IEEE INFOCOM'96*, March 1996.
- 9) S. Bhattacharjee, K. L. Calvert, and E. W. Zegura, "Network Support for Multicast Video Distribution". Technical Report (GIT-CC-98-16), 1998.
- 10) S. Bajaj, L. Breslau, and S. Shenker, "Uniform versus Priority Dropping for Layered Video". In *Proc. of ACM SIGCOMM'98*, Sept. 1998.
- 11) M. Grossglauser and J.-C. Bolot, "On Service Models for Multicast Transmission in Heterogeneous Environments". In *Proc. of IEEE INFOCOM 2000*, March 2000.
- 12) D. Rubenstein, J. Kurose, and D. Towsley, "The Impact of Multicast Layering on Network Fairness". In *Proc. of ACM SIGCOMM'99*, Sept. 1999.
- 13) R. Gopalakrishnan, et al., "Stability and Fairness Issues in Layered Multicast". In *Proc. of NOSSDAV'99*, June 1999.
- 14) チャン ハグエン, 川田 雅人, 中内 清秀, 森川 博之, 青山 友紀, "スパースモードに適した階層型マルチキャスト輻輳制御手法". 2000年電子情報通信学会総大会, B-7-44, Mar. 2000.
- 15) UCB/LBNL/VINT Network Simulator - ns (version 2) (1999). <http://www-mash.cs.berkeley.edu/ns/>.

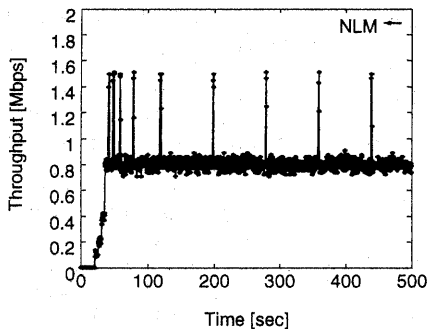


図 6 : レート制御の様子
Fig 6 : Behavior of Rate Control

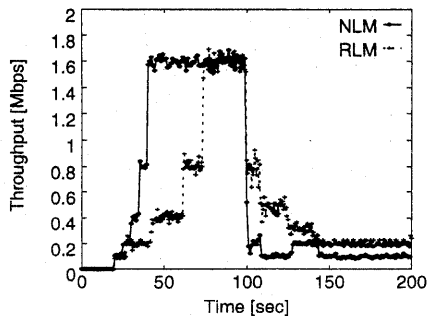


図 7 : 輻輳に対する迅速な対応
Fig 7 : Quick Response to Congestion

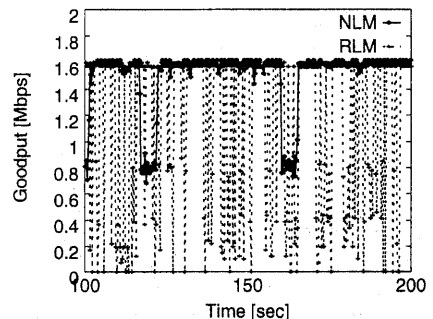


図 8 : グッドブット
Fig 8 : Goodput

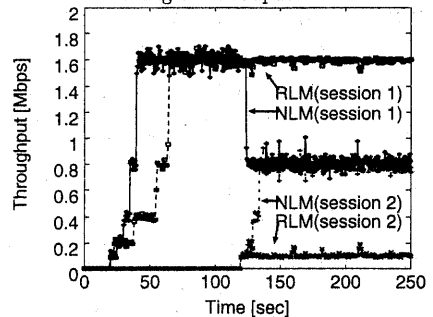


図 9 : セッション間の公平な帯域割当
Fig 9 : Fair Bandwidth Allocation

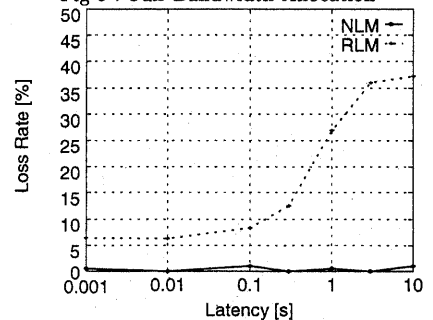


図 10 : 遅延スケーラビリティ
Fig 10 : Delay Scalability

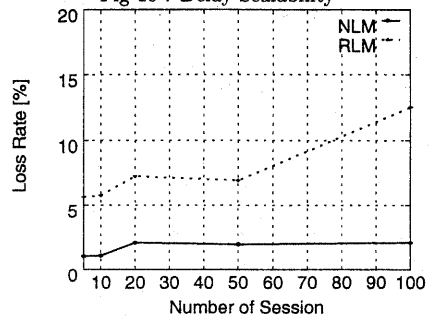


図 11 : セッションスケーラビリティ
Fig 11 : Session Scalability