

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

中内 清秀^{†a)} 森川 博之[†] 青山 友紀[†]

Network-supported Rate Control for Layered Multicast

Kiyohide NAKAUCHI^{†a)}, Hiroyuki MORIKAWA[†], and Tomonori AOYAMA[†]

あらまし インターネットのような多様なネットワーク上でビデオや音声などの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を含めた下位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、下位階層のパケットロスを最小限に抑制することが可能なレート制御が要求される。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ての保証も要求される。しかしながら、従来の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。そこで本論文では、レート制御機能を具備したルータが各階層の優先度を考慮したフィルタリング及びシグナリングを行うことにより、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。

キーワード 階層化マルチキャスト、レート制御、ネットワーク支援、ビデオ配信

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で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、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。このとき、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、単なるマルチキャストでは送信レートが単一であるため、受信者からのフィードバック情報をもとに送信者が適応的なレート制御 [1]~[3] を行っても、このようなネットワークの多様性に対処することはできない [4]。

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 [5]、ネットワーク内のビデオゲートウェイにより再符号化する方式 [6]、階層符号化を用いた階層化マルチキャスト方式 [7]~[13] が提案されている。階層化マルチキャストでは、

送信者は一つのセッションの送信データを階層符号化して基本階層と複数の拡張階層に分割し、それぞれの階層を一般に異なるマルチキャストグループを用いて送信する。階層符号化では、基本階層から順により上位の階層まで合わせて復号するほど、もとのデータにより近い品質が得られるため、受信者はネットワーク状態に応じて基本階層から順に上位階層に対して join する。階層化マルチキャストは、多様性が大きなネットワークにおいては帯域利用効率が高いという利点を有する。

階層化マルチキャストにおいて受信者の受信する階層数をネットワーク状態に適応的に決定することをレート制御と定義すると、レート制御には一般に以下の二つの点を満たすことが要求される。まず、階層符号化の性質上、下位階層のパケットロスであるほどアプリケーションの品質をより大きく劣化させることから、ネットワーク状態に関係なく下位階層のパケットロスを最小限に抑制することが要求される。また、複数の階層化マルチキャストセッションがリンクを共有する場合、開始時刻が異なるセッション間の公平な帯域割当てを保証することも要求される。ここで、本論文では階層化フローを考慮した max-min fairness [9], [14] を満たすことをセッション間の公平性と定義する。

従来の階層化マルチキャストのレート制御は、レー

[†] 東京大学工学系研究科, 東京都
School of Engineering, The University of Tokyo, Hongo 7-3-1
Bunkyo-ku, Tokyo, 113-8656 Japan

a) E-mail: nakauchi@mlab.t.u-tokyo.ac.jp

ト制御を行う主体から受信者駆動型 [7] ~ [12] とネットワーク支援型 [13] の二つに分類される。受信者駆動型では各受信者がネットワーク状態を推定し、受信階層数の増減を決定する。一方、ネットワーク支援型ではネットワーク内のルータやスイッチなどのノードが、各階層の優先度に基づいた優先廃棄によりレート制御を行う。送信者及び受信者には特にレート制御に関する機能は要求されない。

従来の受信者駆動型レート制御は、現在のインターネットのようにルータがふくそう時にフローや優先度を考慮せずにパケットを Drop Tail で廃棄するネットワークモデルを前提としている。また、ネットワークはふくそうの発生を明示的に受信者に伝える手段をもたないため、受信者はパケットロスからふくそうを判断しなければならない。Drop Tail による廃棄を行うネットワークにおいてパケットロスによるふくそうの検出を行えば必然的に下位階層のパケットロスをまねいてしまうため、従来のこのようなレート制御は階層化マルチキャストには本質的に適さない [15]。更に、受信者駆動型レート制御はふくそうに対して迅速に対応することができない [13], [16]。第 1 の理由として、受信者によるネットワーク状態の推定による遅延 (estimation delay) があげられる。受信者はパケットロスだけからネットワーク状態を推定しなければならないため、正確にふくそうの原因を判断するためには遅延が生じる。第 2 の理由として、一般に知られる IGMP (Internet Group Management Protocol) における leave メッセージの処理のための遅延 (leave delay) があげられる。このようにふくそうに迅速に対応できないため、下位階層のパケットロスをより多くまねいてしまう。また、文献 [7], [9], [17] に示されるように、複数のセッションがボトルネックリンクを共有する場合、先に定常状態に達したセッションが不公平に帯域を占有してしてしまうため、セッション間の公平性を保証することができない。このように、受信者駆動型は上述の要求をともに満たすことができない。

一方、従来のネットワーク支援型レート制御ではふくそうを検出したルータが優先廃棄を行うため、優先度の高い下位階層のパケットロスは防止することができる。しかし、下流のふくそうリンクで廃棄されてしまうパケットまで転送することにより帯域を浪費してしまい、セッション間の公平性を保証することができない。

以上のように、従来のレート制御手法は下位階層の

パケットロスの防止とセッション間の公平性という二つの要求を同時に満たすことはできない。インターネットのような Drop Tail による廃棄を行うネットワークにおいて受信者駆動型レート制御を行う限り、下位階層のパケットロスを防止することは不可能である。また、下流のリンクにおける利用可能帯域を把握せずに優先廃棄を行えば、セッション間の公平性は保証できない。以上のような観点から、下位階層のパケットロスの防止及びセッション間の公平性を保証できる階層化マルチキャストを実現するためには、下流における利用可能帯域を把握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。そこで本論文では、このようなネットワーク支援機構を具備し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におけるネットワーク支援

2.1 セッション間の公平性の保証

ネットワーク支援機構により階層化マルチキャストのレート制御を行う場合、一般にルータには優先廃棄機構が導入される [13], [18]。ルータはふくそうを検出したとき、優先度の高い下位階層のパケットを廃棄する前に優先度の低い上位階層のパケットを廃棄する。したがって、RED [19] (Random Early Detection) のようにバッファがオーバフローする前にふくそうを検出しパケットを優先度に応じて廃棄することで下位階層のパケットロスを抑制することができる。更に、各階層に対する優先度割当ポリシーをすべてのセッションで統一すれば、優先廃棄によりリンクを共有するすべてのセッションに対してリンク帯域を適切に分配することが可能となる。

しかしながら、優先廃棄のみによる従来のネットワーク支援型レート制御はセッション間の公平性を保証できない。max-min fairness の定義によれば、あるセッションがあるリンクにおいて等分に割り当てられた帯域を他のリンクにおける制限、または送信レートの制限により使い切ることができない場合、余った帯域は他の競合するセッションに等分して再分配される。従来のネットワーク支援型レート制御では、下流における割当帯域を把握する手段をもたないため、他のリンクにおける制限を把握することができない。そのため、不必要なパケットを転送して帯域を浪費してしまう。

例えば、従来のネットワーク支援型レート制御を行った場合の各セッションへの帯域割当ての様子、及び下流のリンクにおける割当帯域を把握して max-min fairness を実現する理想的な帯域割当ての様子をそれぞれ図 1、図 2 に示す。これらの図では Source # i と Receiver # i が一つのセッション S_i を形成している。また、Link #1 においては S_1, S_4 に、Link #2 においては S_1, S_2 及び S_3 にそれぞれ帯域がすべて割り当てられている。ここで、すべてのリンクの帯域は 6、それぞれの階層の帯域は 1、かつ同じレベルの階層の優先度は同等であるとする。また、 S_i に割り当てられる帯域を B_i とし、各リンクにおいて各セッションに割り当てられる帯域を (B_1, B_2, B_3, B_4) という形で表す。

図 1 においては各ルータは下流における割当帯域を把握していないため、リンク帯域を各セッションに等分に割り当て、Link #1 における帯域割当ては $(3, 0, 0, 3)$ 、Link #2 における帯域割当ては $(2, 2, 2, 0)$ となる。一方、各ルータ（及び送信者）が下流における割当帯域を把握することができれば、Link #2 において $B_1 = 2$ と制限されていることから、図 2 に示すように Link #1 における帯域割当ては $(2, 0, 0, 4)$ となり、 S_4 は等分された帯域以上の帯域を獲得できる。このように、帯域を浪費することがないため max-min fairness を達成することができる。したがって、セッション間の公平性を保証するためには、フィードバックにより下流のリンクにおける割当帯域を把握しなければならない。

2.2 フィルタリングの必要性

ルータにおいて図 2 に示すような max-min fairness を満たす帯域割当てを優先廃棄により行った場合、上位の階層はその一部だけが転送される場合がある。例えばある受信者においては第 1 階層、第 2 階層はすべて受信されるが、第 3 階層は 50% だけ受信されるという状況が発生する。階層符号化されたデータの場合、符号化方式にも依存するが、一般にある階層のデータはパケットロス率が一定値以下で受信されなければ復号できない。したがって、一部だけ受信された第 3 階層は復号されずにむだになり、max-min fairness を満たす帯域割当てが結果として帯域の浪費をまねいてしまう。このような帯域の浪費を防いで帯域利用効率を向上させるためには、ルータは単なる優先廃棄ではなく、優先度に基づいて低優先度の階層に属するパケットをすべて廃棄するフィルタリングを行う必要

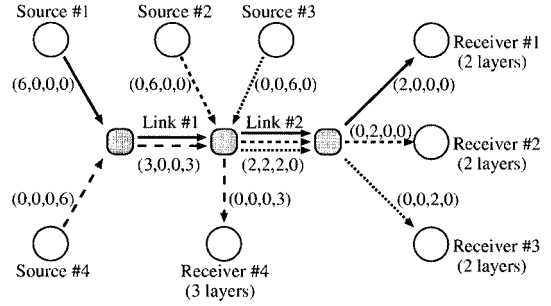


図 1 下流リンクにおける割当帯域を考慮しない従来のネットワーク支援型レート制御による帯域割当て
Fig. 1 Network-supported rate control without the knowledge of the limitation of bandwidth on downstream links cannot achieve max-min fairness.

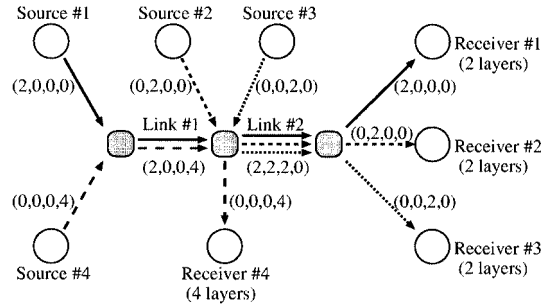


図 2 下流リンクにおける割当帯域を考慮した max-min fairness を満たす帯域割当て
Fig. 2 Network-supported rate control based on the knowledge of the limitation of bandwidth on downstream links can assure max-min fairness.

がある。この場合、フィルタリングを行うことにより max-min fairness を達成することができなくなるが、一つのセッションのレートが段階的であるという制約のもとで max-min fairness に最も近い現実解を求めることは可能である。このように、階層化マルチキャストにおいては max-min fairness と帯域利用効率はトレードオフの関係にあるといえる。

以上のような観点から、下位階層のパケットロスの防止とセッション間の公平性という二つの条件を同時に満たすためには、ルータは各セッションへの割当帯域を上流のルータに報告するシグナリング機構とともに、階層の優先度及び下流における割当帯域を考慮したフィルタリング機構が必要となる。

3. ネットワーク支援型階層化マルチキャスト (NLM)

本章においては、ネットワーク支援機構としてフィルタリング及びシグナリングを用いてレート制御を行う階層化マルチキャスト (NLM: Network-supported Layered Multicast) アーキテクチャを示す。NLM は帯域保証などを行わない既存の IP ネットワークに導入することを前提としており、IP マルチキャスト機構に対するいっさいの変更を要求しない。

3.1 システムモデル

本論文では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。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。

NLM のシステムモデルを以下に示す。まず、送信者は階層符号化を行った後、すべての階層が同一の経路を選択するよう、すべての階層を同一マルチキャストアドレスにて送信する。このとき、送信する DATA パケットにそのパケットが属する階層を示すタグを付加する。

次に、階層化マルチキャスト対応ルータ (LMR: Layered-Multicast-capable Router) はフィルタリング及びシグナリングによりレート制御を行う。NLM ではすべてのルータを LMR に置換する必要はなく、LMR はネットワーク内に分散的に導入される。理想的には LMR はレート制御が必要になる場所、すなわち帯域の差が大きいネットワークの境界、及びふくそうが頻繁に発生するポイントに導入される。

最後に、受信者は LMR 間で発生したふくそうを検出し、シグナリングにより上流の LMR にふくそうの発生を報告する機能を具備する。LMR は LMR 間で発生したふくそうを検出することはできないからである。

3.2 レート制御の概要

NLM では、ふくそうを検出した LMR がフィルタリングを行うことによりふくそうに対する迅速な対応を可能とする。また、各階層の優先度及びそれぞれのセッションの下流における割当帯域に基づいて、すべてのセッションのなかからフィルタリング開始/解除の対象となる階層を一つだけ選択するとともに、LMR

<i>MGroup</i>	<i>Source</i>	Session ID S_i	<i>PrevID</i>	<i>Oif I_j</i>	S_i	$L_{max}(S_i)$	$L_{cur}(I_j, S_i)$
S1	Src1	1	LMR1	1	1	2	2
S2	Src2	2	LMR2		2	3	3
S3	Src3	3	LMR3		3	4	4
S1	Src4	4	LMR4	2	1	2	2
⋮	⋮	⋮	⋮		4	1	1
⋮	⋮	⋮	⋮		3	4	3
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

図3 セッション状態表
Fig. 3 Session state table.

間のシグナリングを行って不要な階層の転送を防止することによりセッション間の公平な帯域割当てを可能とする。このように、NLM はフィルタリングとシグナリングによりレート制御を行う。NLM はアウトプットインタフェース (oif) ごとに独立にレート制御を行う。NLM のレート制御は以下の三つの機能から構成される。

- (1) oif におけるふくそうの検出、及び特定の階層のフィルタリング (フィルタリング機能)。
- (2) NLM ノード (送信者、受信者及び LMR) 間のシグナリング (シグナリング機能)。
- (3) セッション状態表による oif 単位のセッション情報の保持 (セッション制御機能)。

各 LMR は図 3 に示されるセッション状態表を保持している。セッション状態表には、マルチキャストアドレス *MGroup*、送信者アドレス *Source*、*MGroup* と *Source* により一意に決定されるセッション ID S_i 、及び直上の LMR のアドレス *PrevID* が保持されている。ただし、本論文ではセッション S_i が形成するマルチキャストツリーにおいて、ある LMR (LMR1 とする) と S_i の送信者を結ぶパス上に存在する LMR で、LMR1 に論理的に最も近い LMR を、LMR1 の直上の LMR と呼ぶ。NLM ではシグナリングパケットは直上の LMR にユニキャストで送信されるため *PrevID* が必要となる。セッション状態表には、更に oif I_j を通過するセッション S_i について、フィルタリングせずに転送している最上位階層の番号 $L_{cur}(I_j, S_i)$ 、及び転送可能である最上位階層の番号 $L_{max}(S_i) (= \max_j L_{cur}(I_j, S_i))$ が保持される。セッション状態表は、フィルタリング、シグナリング及びセッション制御により更新される。 $L_{cur}(I_j, S_i)$ の初期値は 0 である。

LMR は各 oif のキュー長をモニタリングしておりパケットをキューに入れるたびに平均キュー長 ($qlen$) を計算する。 $qlen$ が上位のしきい値 ($qmax$) を超え

ることにより LMR はふくそうの発生を検出し, $qlen$ が下位のしきい値 ($qmin$) を下回ることによりふくそうの解消を検出する。ここで, キュー長ではなく平均キュー長を用いるのは, パケットロスを生じさせない程度の一時的なバーストラヒックによりキュー長が $qmax$ を超えても転送階層数を減少させないようにするためである。

LMR はセッション状態表に基づいて oif ごとに独立にフィルタリングを行う。oif I_j において LMR がセッション S_i のパケットを受信した場合, そのパケットの属する階層 L が $L_{cur}(I_j, S_i)$ より大きければフィルタリングする。したがって, $L_{cur}(I_j, S_i)$ の値を増減させることにより oif I_j における S_i のレート制御が実現される。oif I_j においてふくそう発生を検出した LMR は, $L_{cur}(I_j, S_{i_0})$ 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。逆に, oif I_j においてふくそう解消を検出した場合, $L_{cur}(I_j, S_{i_1})$ 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一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。本論文では前者のプロセスを DROP, 後者のプロセスを ADD と呼ぶ。ADD 及び DROP の様子を図 4 に示す。

更に, LMR は DROP を行った後, 必要に応じてシグナリングを行う。LMR は DROP を行った後, DROP の対象となって新たにフィルタリングを開始した階層を他のどの oif からも転送していない場合, その階層の受信は不要となるので帯域の浪費を防止するために $DROP_REQ$ パケットを用いて直上の LMR にその階層の転送中止を要求する。逆に, ADD を行った後, ADD の対象となって新たにフィルタリングを解除した階層を他のどの oif からも転送していない場合, つまり上流の LMR から受信していない場合, ADD_REQ パケットを用いて直上の LMR にその階層の転送開始を要求する。このようなシグナリングの様子を図 5 に示す。シグナリングにより, NLM は下流における割当帯域を把握することができ, 下流における割当帯域と優先度に基づいたフィルタリングを行うことにより max-min fairness を保証することができる。

また, NLM では $SESS$ パケットを用いてセッション制御を行う。セッション制御により, 分散配置された LMR 間でセッション状態表中の L_{cur} 及び L_{max}

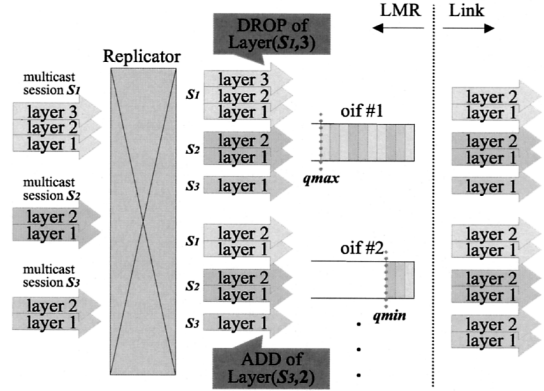


図 4 フィルタリングの概要
Fig. 4 Overview of filtering mechanism.

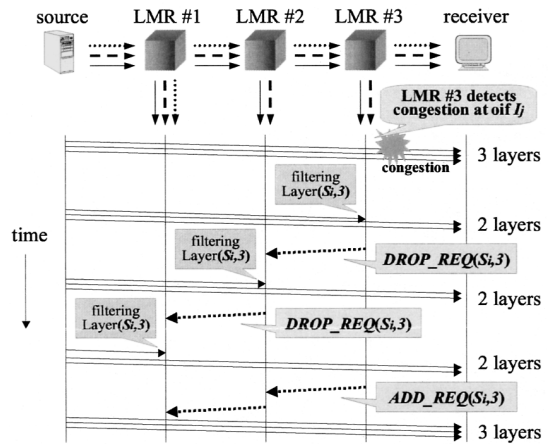


図 5 シグナリングの概要
Fig. 5 Overview of signaling mechanism.

の一貫性の保持, 及び $PrevID$ の更新を行う。

以下, フィルタリングアルゴリズム, シグナリングアルゴリズム, セッション制御アルゴリズムを 3.3-3.5 においてそれぞれ説明する。

3.3 フィルタリングアルゴリズム

LMR はふくそうを検出した oif においてフィルタリングを行う。まず, フィルタリングアルゴリズムにおいて使用するパラメータ及び定数を表 1 に示す。 add_intvl^j , add_time^j , $drop_time^j$, $qlen^j$ は変数であり, 初期値は 0.0 とする。その他は定数ですべての oif に共通である。

フィルタリングアルゴリズムを図 6 に示す。LMR は oif I_j においてパケットをキューに入れるたびに

表1 フィルタリングアルゴリズムに用いるパラメータ及び定数

Table 1 Parameters for filtering algorithm.

add_intvl^j	interval for the next ADD at I_j
add_intvl_min	minimum value of add_intvl^j
add_intvl_max	maximum value of add_intvl^j
add_time^j	last ADD time at I_j
$drop_intvl$	interval for the next DROP
$drop_time^j$	last DROP time at I_j
$detect_period$	period for detecting congestion
$qlen^j$	average queue length at I_j
$qmax$	upper threshold
$qmin$	lower threshold
$qweight$	weight for calculating $qlen^j$
$\alpha (> 1)$	increase constant for add_intvl^j
$\beta (< 1)$	decrease constant for add_intvl^j

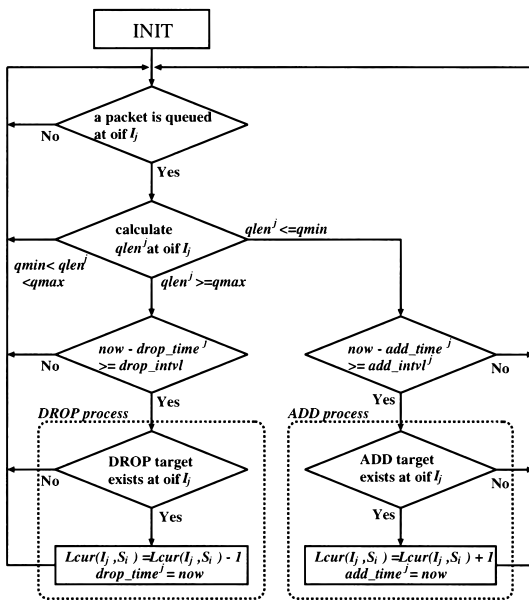


図6 フィルタリングアルゴリズム
Fig.6 Filtering algorithm.

$qlen^j$ を以下の式に従って求めている .

$$qlen^j = qweight \times qcur^j + (1 - qweight) \times qlen^j$$

ここで、 $qcur^j$ とはその時点におけるキュー長、 $qweight$ とは平均キュー長を求めるための重みをそれぞれ表す .

LMR は oif I_j において最初に $qlen^j \geq qmax$ となることにより検出したふくそうを契機として、oif I_j においてフィルタリングによるレート制御を開始する . ここでは便宜上、セッション S_i の L 番目の階層を $Layer(S_i, L)$ と表す . また、DROP 及び ADD の対象

となった階層の番号をそれぞれ L_{dt} 、 L_{at} と表し、現在時刻を now と表す .

まず、フィルタリングアルゴリズムにおける DROP プロセスについて述べる . oif I_j においてふくそうの発生を検出した LMR は DROP を行う . このとき、セッション状態表の oif I_j のエントリから、以下の条件をともに満たす階層 $Layer(S_{i_0}, L_{cur}(I_j, S_{i_0}))$ を探し、DROP の対象階層とする .

- (1) $i_0 = i$,
- if $L_{cur}(I_j, S_i) = \max_k \{L_{cur}(I_j, S_k)\}$.

- (2) $L_{cur}(I_j, S_{i_0}) \geq 2$.

なお、対象候補が複数存在する場合は適宜選択する . ただし、基本階層だけはふくそうが生じても受信を継続すると考え、oif I_j を通過するすべてのセッションについて $L_{cur}(I_j, S_i) = 1$ であれば DROP の対象はないものとする .

DROP の対象階層 $Layer(S_{i_0}, L_{dt})$ ($L_{dt} = L_{cur}(I_j, S_{i_0})$) 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、 $Layer(S_{i_0}, L_{dt})$ のフィルタリングを開始する . 同時に、現在の時刻を $drop_time^j$ に記録する . 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 $qlen^j$ 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 においてふくそうの解消を検出した LMR は ADD を行う . このとき、セッション状態表の oif I_j のエントリから、以下の条件をともに満たす階層 $Layer(S_{i_1}, L_{cur}(I_j, S_{i_1}))$ を探し、ADD の対象階層とする .

- (1) $i_1 = i$,
- if $L_{cur}(I_j, S_i) = \min_k \{L_{cur}(I_j, S_k)\}$.

- (2) $L_{cur}(I_j, S_{i_1}) < L_{max}(S_{i_1})$.

なお、対象候補が複数存在する場合は適宜選択する . ここで、上記の条件 (1) を満たすセッション S_{i_1} が条件 (2) を満たさなければ、この S_{i_1} を除いたすべてのセッションについて再び条件 (1) (2) を確認し、条件をともに満たすセッションが見つかるまで条件の確認を繰り返す .

ADD の対象階層 $Layer(S_{i_1}, L_{at})$ ($L_{at} = L_{cur}(I_j, S_{i_1}) + 1$) 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、 $Layer(S_{i_1}, L_{at})$ のフィルタリングを解除する . 同時に、現在の時刻を add_time^j に記録する . ここで、LMR は ADD 後一定時間 add_intvl^j 待機し、 $Layer(S_{i_1}, L_{at})$ のフィルタリング解除の効果が $qlen^j$ に反映されるのを待つ .

add_intvl^j の制御に関しては 3.6 において述べる .

3.4 シグナリングプロトコル

NLM では $DROP_REQ$ パケットと ADD_REQ パケットを用いてシグナリングを行う . これらのパケットは直上の NLM ノードあてにユニキャストで送信される . 本論文では $Layer(S_i, L)$ のフィルタリングの開始を要求するパケット , $Layer(S_i, L)$ のフィルタリングの解除を要求するパケットをそれぞれ $ADD_REQ(S_i, L)$, $DROP_REQ(S_i, L)$ と表す .

まず , シグナリングパケットの生成条件について述べる . oif I_j において $Layer(S_i, L)$ を対象とした DROP または ADD を行った LMR は , 以下の条件を満たす場合にシグナリングを行う .

$$L > \max_{k \neq j} \{L_{cur}(I_k, S_i)\}.$$

すなわち , oif I_j において $Layer(S_i, L)$ をフィルタリングの対象とした DROP を行った LMR は , $Layer(S_i, L)$ を oif I_j からのみ転送していた場合 , DROP により $Layer(S_i, L)$ を受信する必要がなくなるため , $DROP_REQ(S_i, L)$ を直上の LMR にユニキャストで送信する . 一方 , oif I_j において $Layer(S_i, L)$ をフィルタリングの解除の対象とした ADD を行った LMR は , $Layer(S_i, L)$ を oif I_j からのみ転送することになった場合 , ADD により $Layer(S_i, L)$ を受信することが必要となったため , $ADD_REQ(S_i, L_{at})$ を直上の LMR にユニキャストで送信する .

LMR はシグナリングを行うとき , シグナリングパケットを一定時間 ($detect_period$) の間 , ss_intvl 間隔で定期的に送信する . 同じシグナリングパケットを複数送信することにより , ふくそうが発生してパケットロスが生じてもこれらのパケットが上流の LMR に到達することを保証することができる .

次に LMR におけるシグナリングパケットの処理について述べる . まず , LMR が oif I_j から $DROP_REQ(S_i, L)$ を受信した場合 , 以下に示すアルゴリズムに従って指定された $Layer(S_i, L)$ のフィルタリングを開始する .

$$\begin{aligned} \text{if } (L \leq L_{cur}(I_j, S_i)) \{ & \quad (1a) \\ & L_{cur}(I_j, S_i) = L - 1; \\ & drop_time^j = now; \\ \} \end{aligned}$$

$$\begin{aligned} \text{if } (L > \max_{k \neq j} \{L_{cur}(I_k, S_i)\}) & \quad (1b) \\ \text{forward } DROP_REQ(S_i, L); \end{aligned}$$

else

discard $DROP_REQ(S_i, L)$;

条件 (1a) が満たされた場合 , 受信した $DROP_REQ(S_i, L)$ が最初の要求であることから , セッション状態表中の $L_{cur}(I_j, S_i)$ を $L - 1$ に更新する . 条件 (1a) が満たされなければ , 重複する $DROP_REQ(S_i, L)$ を受信したこととなるため , $L_{cur}(I_j, S_i)$ の更新は行わない .

次に条件 (1b) が満たされた場合 , DROP の対象となった $Layer(S_i, L)$ は oif I_j からしか転送されていないことになる . したがって , $DROP_REQ(S_i, L)$ を受信した LMR にとっても $Layer(S_i, L)$ の受信が不要となるため , $DROP_REQ(S_i, L)$ を直上の LMR に転送する . 条件 (1b) が満たされなければ $DROP_REQ(S_i, L)$ は廃棄され , シグナリングが終端される .

一方 , LMR が oif I_j から $ADD_REQ(S_i, L)$ を受信した場合の処理は , まず条件 (1a) が $L > L_{cur}(I_j, S_i)$ となる . この条件が満たされた場合 , $L_{cur}(I_j, S_i)$ が $L_{cur}(I_j, S_i) = L$ と更新され , $drop_time^j$ ではなく add_time^j が now に更新される . それ以外の処理は $DROP_REQ(S_i, L)$ と同様である .

3.5 セッション制御

NLM では $SESS$ 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 間でセッション状態表中の L_{cur} 及び L_{max} の一貫性を保つこと , 及び直上の LMR のアドレスの把握である . S_i の送信者は送信階層数 L を示したセッション制御パケット $SESS(S_i, L)$ を一定時間間隔 (ss_intvl) で送信する . $SESS$ は直上の LMR のアドレスを表す領域 ($UpID$) も有する . LMR が oif I_j から $SESS(S_i, L)$ を転送する場合の処理を以下に示す .

```

if (  $S_i$  is not registered in  $I_j$  record )
  register  $S_i$  in  $I_j$  record;
if (  $PrevID$  in  $S_i$  record  $\neq UpID$  in  $SESS$  )
   $PrevID = UpID$ ;
 $UpID =$  This LMR Address;
 $L_{max}(S_i) = L$ ;
if (  $now - add\_time^j \geq detect\_period$  ) { (2)
  if (  $L > L_{cur}(I_j, S_i)$  ) (2a)
     $L = L_{cur}(I_j, S_i)$ ;
  elseif (  $L < L_{cur}(I_j, S_i)$  ) (2b)
     $L_{cur}(I_j, S_i) = L$ ;
}

```

forward $SESS(S_i, L)$;

LMR はまずセッション状態表において S_i が oif I_j のエントリに登録されているかどうかを調べる。もし登録されていないければ、新たに S_i のエントリを加える。次にセッション状態表の I_j , S_i のエントリにある $PrevID$ と $SESS$ の $UpID$ を比較する。もし両者が異なっていれば、マルチキャストツリーの変更により直上の LMR の位置が変化したとになるため、セッション状態表の $PrevID$ を $UpID$ に更新する。この後、 $SESS$ の $PrevID$ に自身の IP アドレスを書き込む。この処理により、ルート変更時においても直上の LMR の IP アドレスを把握することが可能となる。

次に LMR はセッション状態表の $L_{max}(S_i)$, $L_{cur}(I_j, S_i)$, 及び $SESS(S_i, L)$ の L を更新する。まず LMR は $L_{max}(S_i)$ を L に更新する。 $L_{max}(S_i)$ を L に制限することにより、DROP を行った LMR だけが DROP の対象となった階層に対する ADD を実行することができる。

更に条件 (2) により、oif I_j において $Layer(S_i, L)$ を対象とした ADD を実行した直後は $L_{cur}(I_j, S_i)$ 及び L の更新を行わない。これは、ADD を実行した直後は、 $ADD_REQ(S_i, L)$ の伝搬遅延のため上流の LMR の L_{cur} の値が変更されておらず、受信した $SESS(S_i, L)$ の L も上流の LMR の L_{cur} を反映していないためである。ADD を行った LMR は上流の LMR の L_{cur} が更新されるのを $detect_period$ だけ待つ。なお、 add_time^j の初期値は 0 であるため、ADD が一度も行われていない oif I_j においては、 I_j のエントリに S_i が登録されてから $detect_period$ だけ経過した後は、条件 (2) が常に真となる。

ADD 後 $detect_period$ 経過し、かつ条件 (2a) が満たされた場合、LMR は $SESS(S_i, L)$ の L を $L_{cur}(I_j, S_i)$ に更新する。これにより S_i について現在転送されている階層数を下流の LMR に把握させることができる。また、条件 (2b) が満たされた場合は、 $L_{cur}(I_j, S_i)$ を $SESS(S_i, L)$ の L に更新する。これは上流の LMR で DROP が行われ、転送階層数が減少したからである。

3.6 拡張フィルタリングアルゴリズム

LMR は同一階層に対する ADD 及び DROP が繰り返されることにより特定のセッションのレートが変動することを防ぐため、 add_intvl^j の適応的制御を行う。LMR は実行した ADD に起因するふくそうの発

生を検出した場合には add_intvl^j を以下のように更新する。

$$add_intvl^j = \min(add_intvl^j \times \alpha, add_intvl_max)$$

ここで、 $\alpha (> 1)$ は定数である。一方、ふくそうの発生を検出しなければ add_intvl^j を以下のように更新する。

$$add_intvl^j = \max(add_intvl^j \times \beta, add_intvl_min)$$

ここで、 $\beta (< 1)$ は定数である。

LMR は oif I_j において $Layer(S_i, L)$ を ADD 後、以下のいずれかの条件を満たした場合に、実行した ADD に起因するふくそうが発生したと判断する。

(1) $detect_period$ ($\leq add_intvl_min$) 以内に oif I_j においてふくそうを検出する。

(2) $detect_period$ 以内に下流の LMR から $DROP_REQ(S_i, L_{cur}(I_j, S_i))$ を受信する。

(3) $detect_period$ 経過後に上流から $SESS(S_i, L_{cur}(I_j, S_i) - 1)$ を受信する。

条件 (2) により、LMR は自身の oif I_j ではなく下流で発生したふくそうを検出する。例えば LMR # k が $Layer(S_i, L)$ を ADD したことにより下流の LMR # k' においてふくそうが発生した場合を考える。このとき LMR # k' が $Layer(S_i, L)$ を DROP すれば、上流に向けて $DROP_REQ(S_i, L)$ を送信する。したがって、LMR # k は ADD 後 $detect_period$ 以内に $DROP_REQ(S_i, L)$ を受信することにより、下流においてふくそうが生じたことを確認できる。

また、条件 (3) により LMR は上流で発生したふくそうを検出する。同様に LMR # k が $Layer(S_i, L)$ を ADD したことにより、 $ADD_REQ(S_i, L)$ を受信した上流の LMR # k'' においてふくそうが発生した場合を考える。このとき LMR # k'' が $Layer(S_i, L)$ を DROP すれば、LMR # k'' が下流に転送する $SESS(S_i, L)$ の L の値は LMR # k が ADD を行う前と同じになる。よって、LMR # k は自身の $L_{cur}(I_j, S_i)$ の値よりも 1 だけ小さい値が L に示された $SESS(S_i, L)$ を受信する。したがって LMR # k は ADD 後 $detect_period$ 経過後に $SESS(S_i, L_{cur}(I_j, S_i) - 1)$ を受信することにより、上流においてふくそうが生じたことを確認できる。

このような add_intvl^j の適応的制御により、利用可能帯域が定期的にな少ない場合に頻繁な ADD を抑制することができると同時に、利用可能帯域が増加した

場合にも適切なレートへの収束時間を短縮することが可能となる。

3.7 送信者及び受信者の動作

まず送信者の動作について説明する。送信者はセッション開始直後、及びシグナリングパケット受信時にレート制御を行う。送信者は図 3 に示したセッション状態表のサブセットである $(S_i, L_{max}(S_i), L_{cur}(S_i))$ のエントリを保持している。 S_i の送信者はセッション開始後、 $DROP_REQ$ を受信するまで一定時間間隔 (add_intvl_min) で ADD を繰り返す。送信者は ADD を行うたびに $L_{cur}(S_i)$ (初期値 0) を 1 だけ増加させる。

また、 S_i の送信者はシグナリングパケットを受信した場合、3.4 で述べたシグナリングパケットの処理と同様の処理を行い、送信階層数を増減させる。

次に受信者の動作について説明する。受信者は join 直後、及び LMR 間で発生し LMR では検出できないふくそうを検出した場合においてのみレート制御を行う。受信者は送信者同様、 $(S_i, L_{max}(S_i), L_{cur}(S_i))$ のエントリを保持している。 S_i の受信者は join 後、受信する $SESS(S_i, L)$ から、 S_i について受信できる階層数が L であると判断する。そこで、 $Layer(S_i, L)$ を受信するまで一定時間間隔 (add_intvl_min) で ADD を繰り返す。したがって、 add_intvl^j 間隔でしか ADD できない LMR と異なり、受信者は $Layer(S_i, L)$ を受信するまでの時間を短縮することができる。

また、 S_i の受信者は測定したパケットロス率がしきい値 ($loss_th$) を超えたとき、ふくそうが発生したと判断する。このとき、最上位受信階層 ($Layer(S_i, L_{cur}(S_i))$) の転送中止要求 $DROP_REQ(S_i, L_{cur}(S_i))$ を 3.4 で述べた LMR と同様に直上の LMR に送信する。同時に $L_{cur}(S_i)$ を 1 減少させる。また、受信者が $SESS(S_i, L)$ を受信したときの処理は 3.5 で述べた LMR における処理と同様である。

3.8 LMR におけるアウトプットインタフェースの状態遷移

LMR は oif ごとに状態を保持しており、状態遷移を行う。oif の状態は *init* 状態、 *unloaded* 状態、 *loaded* 状態、 *congested* 状態、 *drop* 状態、 *tmp* 状態の六つに分類される。LMR の状態遷移の様子を状態が遷移するときの条件も併せて図 7 に示す。

oif I_j の初期状態は、 *init* 状態である。oif I_j は、 $qlen^j \geq qmax$ となるまで *init* 状態を保ち、

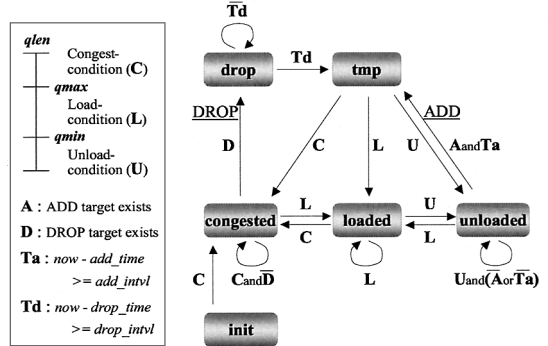


図 7 LMR におけるアウトプットインタフェースの状態遷移図

Fig. 7 State transition of an oif at LMR.

congested 状態に移る。*congested* 状態に移ると、DROPP の対象が存在すればすぐに DROPP を行い、3.3 で述べたようにレート制御を開始する。oif I_j は DROPP 後 $drop_intvl$ の間 *drop* 状態を保ち、その後 *tmp* 状態に移る。

また、 *unloaded* 状態において、ADD の対象が存在し、かつ直前の ADD から add_intvl^j 以上経過している場合、oif I_j において ADD が行われ、oif I_j は *tmp* 状態に移る。

oif I_j は *tmp* 状態に移るとすぐに、 $qlen^j$ の大きさに応じて *unloaded* 状態、 *loaded* 状態、 *congested* 状態のいずれかの状態に移る。

3.9 LMR のアーキテクチャ

LMR のアーキテクチャを図 8 に示す。LMR は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セッション状態表、及び oif ごとのキューモニタとフィルタから構成される。oif I_j に対応するキューモニタはふくそうを検出した場合、oif I_j に対応するフィルタにふくそうの発生を通知する。このとき、フィルタは 3.3 で述べたアルゴリズムに従って DROPP を行うかどうかを判断し、DROPP を行った場合はセッション状態表を更新する。フィルタは ADD を実行した後もセッション状態表を更新する。更に、フィルタは $SESS$ を受信した場合においても、3.5 で述べたセッション制御に従ってセッション状態表を更新する。フィルタは $DATA$ を受信した場合、セッション状態表を参照しながら廃棄するかどうかを決定する。

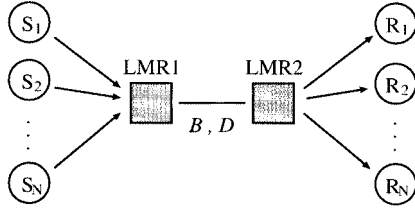


図9 シミュレーショントポロジー
Fig.9 Simulation topology.

表2 シミュレーションパラメータ
Table 2 Simulation parameters.

DATA packet size	1,024	[bytes]
Maximum number of layer	5	
Rate of a base layer	100	[Kbps]
Rate of the l -th layer ($2 \leq l \leq 5$)	$100 \times 2^{l-2}$	[Kbps]
Router queue size	20	[packets]
Number of sessions	N	
Bottleneck link bandwidth	B	[Mbps]
Access link bandwidth	4.0	[Mbps]
Bottleneck link delay	D	[ms]
Access link delay	10	[ms]
q_{max}	15	[packets]
q_{min}	3	[packets]
q_{weight}	0.05	
add_intvl_min	5	[s]
add_intvl_max	80	[s]
$drop_intvl$	0.5	[s]
$detect_period$	5	[s]
α	2.0	
β	0.75	
$loss_th$	0.25	
ss_intvl	0.1	[s]

に劣化し、IGMP をレート制御に利用しない NLM との性能の差はますます大きくなることに注意されたい。

4.2 シミュレーション結果

まず、NLM のレート制御の様子を図 10 に示す。 $N = 1, D = 10\text{ms}$ とし、ふくそうを生じさせるためにボトルネックリンク帯域が第 5 階層までの合計レートよりも 0.1 Mbps 小さくなるよう $B = 1.5\text{Mbps}$ とした。受信者は 20s に join した後 add_intvl_min (5s) 間隔で受信階層数を増加させる。受信者が 40s に第 5 階層の受信を開始した後、LMR 1 においてふくそうが生じるため、LMR 1 はレート制御を開始し、DROP を行う。DROP 後ふくそうはすぐに解消されるため、LMR 1 は、直前の ADD の add_intvl (5s) 後に ADD を行う。しかし、ADD 後 $detect_period$ 以内に再び LMR 1 においてふくそうが発生するため、 add_intvl は 3.6 で示したアルゴリズムに従い α 倍

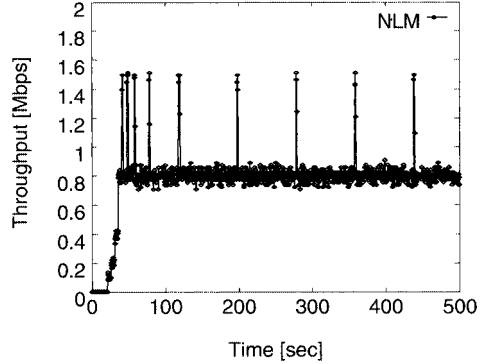


図10 レート制御の振舞い
Fig.10 Overview of rate control.

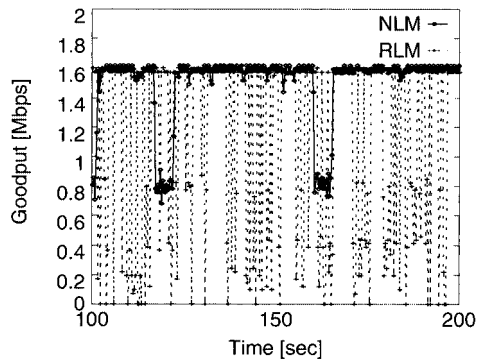


図11 グッドプット
Fig.11 Goodput.

されて 10s となる。以後このように LMR 1 において DROP と ADD が繰り返され、 add_intvl は指数関数的に大きくなり、115s 付近において add_intvl_max (80s) に収束する。

次に、NLM と RLM のグッドプットを図 11 に示す。 $N = 1, D = 10\text{ms}$ とし、パケットロスがわずかであればふくそうとは判断しない RLM ではグッドプットの低下が非常に大きくなることを示すために、ボトルネックリンク帯域 B を第 5 階層までの合計レートと等しい 1.6 Mbps とした。ここで、グッドプットとは実際に復号に利用可能なデータの量を表し、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スループットと定義する。下位階層が受信されなければ上位階層の復号ができないという階層間に依存関係がある場合は、スループットよりもグッドプットのほうがアプリケーションの品質を正確に反映

する．図 11 にはセッション開始後 100～200s までのグッドプットを示している．この間，RLM の受信者は測定したパケットロス率がふくそうを判断するしきい値（25%）よりも常に低いため第 5 階層までの受信を続ける．したがって，スループットは 1.6 Mbps に保たれる．しかし，ルータにおいて下位階層のパケットがわずかにランダム廃棄されるため，その時点のグッドプットは瞬間的に低下してしまう．基本階層のパケットが廃棄されるという最悪の場合，グッドプットは 0 Mbps になってしまう．

このように，パケットロス率からしかふくそうを検出できないためグッドプットが低下してしまう RLM に対して，NLM ではアウトプットバッファの平均キュー長からふくそうを検出し，バッファのオーバーフローが起こる前にふくそうと判断して DROP を行う．したがって，NLM ではアウトプットバッファにおいてパケットがランダム廃棄される確率は非常に低く，グッドプットはスループットと同じ値に保たれる．

次に，妨害トラヒックによりふくそうが発生したときのレート制御の様子を図 12 に示す． $N = 1$ ， $B = 1.6$ Mbps， $D = 10$ ms とし，90s にレート制御を行わない 1.3 Mbps の CBR の妨害トラヒックをボトルネックリンクに発生させた．したがって，残っている利用可能帯域は 0.3 Mbps となり，第 2 階層までしか受信することはできない．図 12 に示されるように，90s にふくそうが発生した後，RLM は 98s 付近でふくそうを検出して第 5 階層に対して leave を行うものの，ふくそうは解消されない．次に，114s 付近，130s 付近でふくそうを検出し，それぞれ第 4 階層，第 3 階層に対して leave を行い，ようやくふくそうが解消される．このように RLM ではふくそうが解消されるまで 40 秒近くも要している．これは，ふくそうの原因を判断するために一定時間（十数秒）間隔でしか leave を実行できない RLM の性質に起因する．更にふくそうが解消されるまでは約 50% から 10% のパケットロス率が観測されており，図 11 と同様にこの間のグッドプットは非常に低くなる．一方，NLM ではアウトプットバッファの平均キュー長からふくそうを判断することができるため $drop_intvl$ (0.5s) 間隔で DROP を行うことができる．したがって，ふくそう発生後約 2s でふくそうを解消することができる．更に，ふくそうが解消されるまでの最大パケットロス率も 1% であった．このように，たとえ巨大なバーストラヒックが発生するような状況においても，NLM

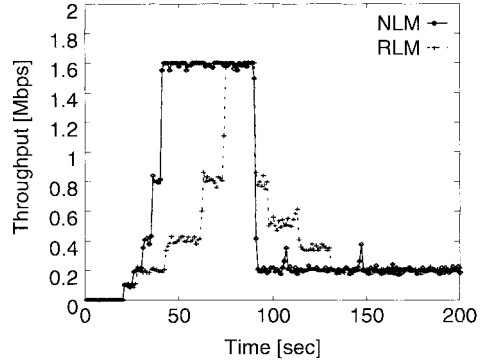


図 12 妨害トラヒックによりふくそうが発生したときのレート制御の振舞い
Fig. 12 Response to the congestion caused by interruption traffic.

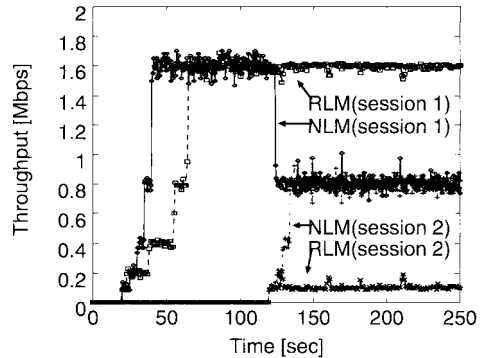


図 13 競合するセッションに対する帯域割当ての様子
Fig. 13 Bandwidth allocation to competitive sessions.

は迅速に対応することができ，低パケットロス率を維持することができる．

ボトルネックリンクで二つのセッションが競合する場合の帯域割当ての様子を図 13 に示す． $N = 2$ ， $D = 10$ ms とし，二つのセッションの受信者がともに第 4 階層まで受信できる帯域として $B = 1.7$ Mbps と設定した．図 9 に示すトポロジーにおいては二つのセッションの受信者がともに第 4 階層まで受信した場合に max-min fairness が達成される．セッションをそれぞれ 20s と 120s に開始させたとき，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しまうため公平な帯域割当てが行われぬ．これに対し，NLM ではすべてのセッションを通じて最も優先度が低い階層がフィルタリングされるため，セッショ

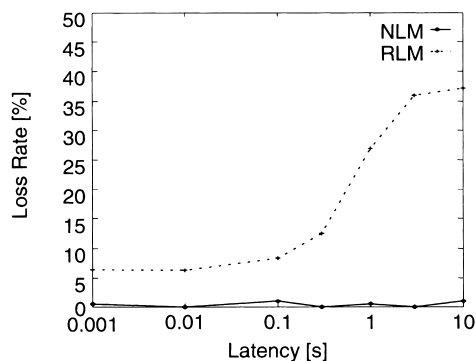


図 14 伝搬遅延と最大パケットロス率の関係
Fig. 14 Relation between propagation delay and maximum loss rate.

ン間の公平な帯域割当てが実現される。

伝搬遅延と最大パケットロス率の関係を図 14 に示す。 $N = 3$, $B = 3.0$ Mbps とし, D を 1 から 10,000 ms まで変化させた。RLM では, IGMP leave メッセージが送信者に到達するまでの時間が伝搬遅延に依存するため, ふくそうが解消されるまでの時間も伝搬遅延に依存してしまう。したがって, ふくそうの解消が遅れ, 最大パケットロス率は大きくなる。一方, NLM ではふくそうが発生した LMR においてフィルタリングが行われるため, ふくそうが解消されるまでの時間は伝搬遅延には依存しない。したがって NLM では最大パケットロス率が一定となる。更に, NLM では上述したようにふくそうが発生しても低パケットロス率を維持しながらふくそうを解消できるため, 最大パケットロス率を 1% 以下に維持することができる。

最後に, セッション数と最大パケットロス率の関係を図 15 に示す。 $B = 1.0 \times N$ Mbps, $D = 10$ ms とし, N を 1 ~ 100 まで変化させた。また, セッションの開始時刻は 20 ~ 120 s の間でランダムに決定した。RLM ではセッション数が多くなるにつれて異なるセッションにおいて同時に join が行われる確率が高くなるため最大パケットロス率も大きくなる。これに対し, NLM ではすべてのセッションを通じて転送階層数が最少であるセッションについてのみ ADD を行うため, セッション数に関係なくふくそうの発生頻度を少なくすることができる。更にふくそうが発生した場合においても, 上述したように低パケットロス率を維持しながらふくそうを解消することができるため, 最大パケットロス率を 2% 以下に維持することができる。

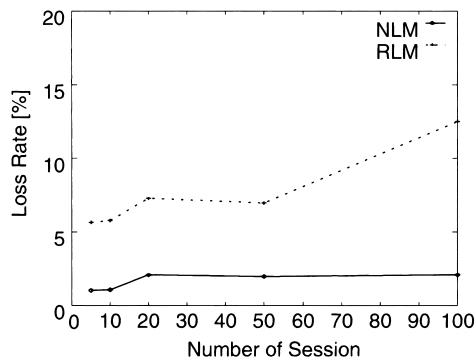


図 15 セッション数と最大パケットロス率の関係
Fig. 15 Relation between propagation delay and maximum loss rate.

なお, 本論文では図 9 に示す最も単純なトポロジーにおいてシミュレーションを行って NLM の基本特性を評価したが, ランダムトポロジーにおいても特性の評価を行っている。送信者からある受信者までのパス上で獲得できる帯域がクリティカルリンク (パス上で最も帯域が小さいボトルネックリンク) における LMR のフィルタリングだけに依存すること, 及び NLM の特性がエンド・エンドの遅延に依存しないことから, ランダムなトポロジーであっても図 9 に示すトポロジーに単純化することができる。したがって, ランダムトポロジーにおいても同様の特性を得ることができる。

5. む す び

本論文では, 下位階層のパケットロスの防止, 及びセッション間の公平な帯域割当て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機構 (NLM) を示した。NLM では, ふくそうを検出した階層化マルチキャスト対応ルータ (LMR) が優先度の低い階層に対してフィルタリングを行うことにより下位階層のパケットロスを防止することができ, パケットロスが生じる前にフィルタリングを開始することにより, ふくそうに対する迅速な対応を可能とする。更に, シグナリングにより得られる下流における割当帯域と階層の優先度をともに考慮してフィルタリングの対象を選択することによりセッション間の公平性の保証を可能とする。シミュレーションにより, NLM がスループットと同等でかつ安定したグッドプット, ふくそうへの迅速な対応, 低パケットロス率, セッション間の公平性を同時に達成できることを示した。

本論文ではすべてのセッションにおいて同じ階層付

号化方式を用いることを前提としたが、各階層のレートがセッションごとに異なるような他の階層符号化方式にも対応することが可能である。この場合、すべてのセッションの送信者は共通の優先度割当ポリシーに従って各階層に優先度を付与する。このようにセッションごとに階層に付与する優先度が異なる方式は、課金量に応じて階層に付与する優先度を高くするというモデルにも応用することができる。また、本論文ではバッファ管理方式として Drop Tail を採用したが、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な帯域割当てを可能とする方式 [21] や種々のスケジューリング方式を採用することも可能であり、様々なサービスモデル上において NLM を展開することが可能である。

文 献

- [1] J.-C. Bolot and T. Turletti, "A rate control scheme for packet video in the Internet," Proc. IEEE INFOCOM'94, June 1994.
- [2] J.-C. Bolot, T. Turletti, and I. Wakeman, "Scalable feedback control for multicast distribution in the Internet," Proc. ACM SIGCOMM'94, Sept. 1994.
- [3] I. Busse, B. Defner, and H. Schulzrinne, "Dynamic QoS Control of Multimedia Applications based on RTP," Proc. of First International Workshop on High Speed Networks and Open Distributed Platforms, June 1995.
- [4] T. Turletti and J.-C. Bolot, "Issues with multicast video distribution in heterogeneous networks," Proc. of 6th International Workshop on Packet Video, Oct. 1994.
- [5] S.Y. Cheung, M. Ammar, and X. Li, "On the Use of Destination Set Grouping to Improve Fairness in Multicast Video Distribution," Proc. of IEEE INFOCOM'96, March 1996.
- [6] E. Amir, S. McCanne, and H. Zhang, "An Application-level Video Gateway," Proc. of ACM Multimedia'95, Nov. 1995.
- [7] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven Layered Multicast," Proc. of ACM SIGCOMM'96, Aug. 1996.
- [8] X. Li, S. Paul, and M. Ammar, "Layered Video Multicast with Retransmissions (LVMR): Evaluation of Hierarchical Rate Control," Proc. of IEEE INFOCOM'98, March 1998.
- [9] X. Li, S. Paul, and M.H. Ammar, "Multi-session Rate Control for Layered Video Multicast," Proc. of Multimedia Computing and Networking 1999, Jan. 1999.
- [10] L. Vicisano, L. Rizzo, and J. Crowcroft, "TCP-like congestion control for layered multicast data transfer," Proc. of IEEE INFOCOM'98, March 1998.
- [11] L. Wu, R. Sharma, and B. Smith, "Thin Streams: An Architecture for Multicasting Layered Video," Proc. of NOSSDAV'97, May 1997.
- [12] T. Turletti, S.F. Parisi, and J.-C. Bolot, "Experiments with a Layered Transmission Scheme over the Internet," INRIA Research Report, no.3296, Nov. 1997.
- [13] S. Bhattacharjee, K.L. Calvert, and E.W. Zegura, "Network Support for Multicast Video Distribution," Technical Report (GIT-CC-98-16), 1998.
- [14] D. Rubenstein, J. Kurose, and D. Towsley, "The Impact of Multicast Layering on Network Fairness," Proc. of ACM SIGCOMM'99, Sept. 1999.
- [15] M. Grossglauser and J.-C. Bolot, "On Service Models for Multicast Transmission in Heterogeneous Environments," Proc. of IEEE INFOCOM 2000, March 2000.
- [16] C. Albuquerque, B. Vickers, and T. Suda, "An End-to-End Source-Adaptive Multi-layered Multicast (SAMM) Algorithm," Proc. of 9th International Packet Video Workshop, April 1999.
- [17] R. Gopalakrishnan, J. Griffioen, G. Hjalmtysson, and C. Sreenan, "Stability and Fairness Issues in Layered Multicast," Proc. of NOSSDAV'99, June 1999.
- [18] S. Bajaj, L. Breslau, and S. Shenker, "Uniform versus Priority Dropping for Layered Video," Proc. of ACM SIGCOMM'98, Sept. 1998.
- [19] S. Floyd and V. Jacobson, "Random Early Detection gateways for Congestion Avoidance," IEEE/ACM Trans. Networking, vol.1, no.4, pp.397-413, Aug. 1993.
- [20] "UCB/LBNL/VINT Network Simulator - ns (version 2)," <http://www-mash.cs.berkeley.edu/ns/>, Sept. 1997.
- [21] 佐々木克博, 中内清秀, 森川博之, 青山友紀, "マルチキャストフローに適したパケットドロップアルゴリズムの設計; 情処学マルチメディア, 分散, 協調とモバイルシンポジウム (DICOMO '99), June 1999.

(平成 12 年 8 月 4 日受付, 10 月 26 日再受付)



中内 清秀 (学生員)

平 10 東大・工・電子情報卒。平 12 同大大学院修士課程了。現在、同博士課程在学中。マルチキャスト通信に関する研究に従事。IEEE, 情報処理学会各会員。



森川 博之 (正員)

昭 62 東大・工・電子卒・平 4 同大学院
博士課程了。現在, 同大・創成・基盤情報
助教授。工博。コンピュータネットワーク,
モバイルコンピューティング, 無線ネット
ワーク, QoS, ネットワークサービス, コ
ラボレーション等の研究に従事。平 3 年度
丹羽記念賞, 平 3 年度本会学術奨励賞等受賞。IEEE, ACM,
情報処理学会, テレビジョン学会, 情報理論とその応用学会各
会員。



青山 友紀 (正員)

昭 44 東大大学院電気工学科修士課程了。
同年日本電信電話公社入社。以来, 電気通
信研究所において, 主として情報通信シス
テム, 通信ネットワークの研究に従事。昭
48 より 1 年間 MIT に客員研究員として滞
在。平 7 NTT 光ネットワークシステム研
究所長に就任。平 9 東大に転職, 現在, 同大学院工学系研究
科電子情報工学専攻教授。工博。情報通信に関する著書多数。
IEEE 会員。