社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
COMP2005-21(2005-06)

共有メモリ型スイッチにおけるオンラインバッファ管理アルゴリズムの競合比の改良

小林　浩二†　　宮崎　修一††　　岡部　寿男††

† 京都大学 情報学研究科　〒606–8501 京都府京都市左京区吉田本町
†† 京都大学 学術情報メディアセンター　〒606–8501 京都市左京区吉田本町
E-mail: †kobaya@net.ist.i.kyoto-u.ac.jp, ††{shuichi,okabe}@media.kyoto-u.ac.jp

あらまし　オンラインバッファ管理問題は，近年のネットワーク運用における主要な論点となっている QoS (Quality of Service) 保証実現のための，スイッチなどのキュー管理をオンライン問題として定式化した問題であり，様々なモデルが考案されている．本論文ではその中の 1 つである共有メモリ型スイッチを扱ったモデルを取り上げる．我々は，アルゴリズム Longest Queue Policy (LQD) の競合比の既知の上限を $2-1/N$ に改良した．ここで，N はスイッチの出力ポート数である．

キーワード　オンラインアルゴリズム，競合比解析，共有メモリ型スイッチ，バッファ管理問題

Improving Competitive Ratios of Online Buffer Management for Shared-Memory Switches

Koji KOBAYASHI†, Shuichi MIYAZAKI††, and Yasuo OKABE††

† Graduate School of Informatics, Kyoto University, Yoshida-honmachi, Sakyo-ku, Kyoto, 606–8501 Japan
†† Academic Center for Computing and Media Studies, Kyoto University, Yoshida-Hommachi, Sakyo-ku Kyoto 606–8501, Japan
E-mail: †kobaya@net.ist.i.kyoto-u.ac.jp, ††{shuichi,okabe}@media.kyoto-u.ac.jp

Abstract　The buffer management problem is a kind of online problems, which formulates the problem of queueing policies of network switches supporting QoS (Quality of Service) guarantee. For this problem, several models are considered, and in this paper, we focus on the model of shared memory switches. We improve the competitive ratio of the Longest Queue Policy (LQD) to $2-1/N$, where N is the number of output ports in a switch.

Key words　Online algorithms; Competitive analysis; Shared memory switches; Buffer management problem

1. はじめに

現行のネットワークのトラフィックの主要なボトルネックの 1 つに，ネットワークの接続点であるルータやスイッチのキュー管理を挙げることが出来る．それらルータやスイッチなどにおいては，複数の入力ポートから同時に到着するパケットを同一の出力ポートへ転送しなければならない場合，マシンの処理能力や帯域制限などのため，直ちに転送出来ない状況が発生する．このような状況で，パケットロスが発生する．

そこで，通常ルータやスイッチにはバッファを設け，パケットを一時的にバッファに蓄えておき，転送可能になった時に処理するという手段が取られている．しかし，ネットワークのトラフィックはバースト的に到着する傾向があり，バッファはパケットロスを緩和するものの，完全になくすことはできない．近年のネットワークにおいては，QoS (Quality of Service) 保証の考えが重要になってきており，優先度の高いパケットのロスをできるだけ少なくすることが課題である．そのため，到着するパケットの優先度に応じて，パケットをバッファに受理するか，もしくは，将来到着するであろう更に優先度の高いパケットのために破棄するかを決定する，いわゆるバッファ管理問題が重要となってくる．

近年，これらの問題をオンライン問題として定式化し，それに対するオンラインアルゴリズムの研究が盛んに行われている．これには種々のモデルがあるが，基本的な問題設定は以下の通りである．各パケットは何らかの価値を持っており，スイッチは有限サイズのバッファを持っている．スイッチは，到着するパケットを，バッファのサイズ分だけバッファ内に格納しておくことができる．スイッチは一定時刻ごとにパケットを送出す

ることができ，バッファ内のパケットが 1 個送られる．(通常，バッファは FIFO キューで実現されており，キューの先頭にあるパケットが送出される.) オンラインアルゴリズムのポリシーは，将来到着するパケットの情報を知らずに，現在のパケットを受理するか破棄するかである．また，既にバッファ内に取り込んでいるパケットを破棄することが許されているモデルもある．これをプリエンプションと言う．

問題の目的はスイッチが送信するパケットの価値の総和を最大化することであり，アルゴリズムの性能評価には競合比解析 [8], [17] が用いられる．競合比解析においては，オンラインアルゴリズムの得た価値を，入力が全て分かっている場合の，最適なオフラインアルゴリズム（これを以下では OPT と書く）の得る価値と比較する．任意の入力に対して，オンラインアルゴリズムが得る価値が，OPT の得る価値の $1/c$ 倍以上である場合に，そのオンラインアルゴリズムの競合比は c であると言う．

本研究の結果

本稿では，Aiello らが問題を提起し考案した一番最初のモデル [1] から派生した，Hahne らの共有メモリ型モデル [11] における既存の結果を改良した．この Hahne らのモデルでは，全てのパケットのサイズが 1 で，価値も 1 である．スイッチは 1 個の入力ポートと N 個の出力ポートを持ち，各パケットには目的の出力ポートがある．各出力ポートはバッファ（FIFO キュー）を持ち，その長さは可変長であるが，N 個のキューのサイズの総和は一定値 M に限定される．すなわち，目的ポートに関係なく，全部で M 個のパケットを保持しておくことが出来る．パケットが到着すると，オンラインアルゴリズムはそのパケットを受理して対応する目的ポートのキューに入れるか，受理せずに破棄するかを決定する．また，パケットのプリエンプションも許されている．パケットの送信は一定時刻ごとに可能であり，送出の可能な時刻には，全ての出力ポートから，そのキューの先頭にあるパケットが 1 個送出される．

Hahne らは [11] において，競合比の下限 4/3 を示した．また，オンラインアルゴリズム Longest Queue Drop Policy (LQD) の競合比の上限 2 を示した．我々はこのモデルにおいて，LQD の競合比の上限を $2-1/N$ に改良する．Hahne らは，OPT が余分に送信するパケットを LQD の送信するパケットと 1 対 1 マッチングさせ，そのパケット数の上限を見積もったが，我々は，そのマッチングを 1 対 $\frac{N}{N-1}$ で構成可能なことを示し，上限を改良した．

関連研究

オンラインバッファ管理問題は，Aiello ら [1] を皮切りに多くのモデルが考案され，研究されている．[1] においてバッファ管理の対象となるのは，プリエンプションを許さない 1 つの FIFO キューである．パケットの持つ価値は 2 種類（1 と $\alpha \geqq 1$）である．このモデルの下限として $2-\frac{1}{\alpha}$ を得ている．Andelman ら [4] は，[1] のモデルの上下限が $2-\frac{1}{\alpha}$ で一致することを示した．パケットの価値が 2 種類ではなく $[1, \alpha]$ の範囲の連続値とするモデルに対しては，競合比の上限 $\ln(\alpha)+2+O(\ln^2(\alpha)/B)$ [3] および下限 $\ln(\alpha)+1$ [4] が知られている．これらの結果を表 1 に示す．また，上記のモデルで，アルゴリズムにプリエンプションを許す場合の競合比の上下限を表 2 に示す．

表 1　プリエンプションを許さない FIFO キューモデルに対する競合比の上下限

	Lower Bound	Upper Bound
2 Values	$2-\frac{1}{\alpha}$ [1]	$2-\frac{1}{\alpha}$ [4]
Multiple Values	$\ln(\alpha)+1$ [4]	$\ln(\alpha)+2+O(\frac{\ln^2(\alpha)}{B})$ [3]

表 2　プリエンプションを許す FIFO キューモデルに対する競合比の上下限

	Lower Bound	Upper Bound
2 Values	1.28 [16]	1.30 [18]
Multiple Values	1.419 [15]	1.75 [6]

表 3　Bouded Delay モデルに対する競合比の上下限

		Lower Bound	Upper Bound
2-uniform	det.	1.377 [10]	1.377 [10]
	rand.	1.172 [7]	1.377 [10]
2-variable	det.	1.618 [12]	1.618 [13]
	rand.	1.25 [9]	1.25 [7]
k-uniform	det.	1	1.838 [10]
k-variable	det.	1.618 [12]	1.939 [10]
	rand.	1	1.582 [7]

表 4　単一価値 Multi-FIFO モデルに対する競合比の上下限

	Lower Bound	Upper Bound
det.	1.582 [2]	1.89 [2]
rand.	1.46 [5]	1.582 [5]

また，別の代表的なモデルとして，パケットが有効期限を持つモデルが考案されている [4], [13]．このモデルのキューは FIFO キューではなく，パケットの到着順序に関わりなく任意の順序で送信することが可能であるが，各パケットが持つ期限以内に送信することが出来なければ，そのパケットの価値を得ることが出来ない．パケットの価値は任意の値をとる．このモデルに対する決定性アルゴリズムと確率的アルゴリズムの，現在知られている最良の競合比を表 3 に示す．ただし，2-uniform は全てのパケットの有効期限が 2 のモデルであり，2-variable は，有効期限（2 以下）がパケットによって異なって良いモデルである．また，k-uniform および k-variable は，2 よりも大きい k に対して，同様に有効期限を定義されたモデルである．

表 4 に，複数の FIFO キューをもつモデルの結果を示す．このモデルでは，各キューに到着するパケットの受理・破棄の他に，毎時刻どのキューからパケットを送信するかという判断もポリシーの振る舞いに含まれている．

また，本研究で取り扱うモデルについて，各パケットの持つ価値が単一でない場合や，プリエンプションを許さないモデルについても研究されている [11], [14]．

2. オンラインバッファ管理問題

本節では，本研究で取り扱う Hahne らによるバッファ管理問題 [11] の定義と，LQD ポリシーの定義を与える．

2.1 問題の定義

本モデルにおける入力は，スイッチに到着する有限個のパケット列で与えられる．全てのパケットの価値は 1 である．各パケットには，送出されるべき出力ポートの番号 (1〜N) が記されている．パケットは，同一時刻に 2 個以上到着しないものとする．

オンラインアルゴリズムは，到着したパケットを受理するか破棄するかを，次のパケットが到着する前に決定する．ただし，送出ポート別にキューが用意されており，保持されているパケットは該当する出力キューに納められるものとする．スイッチのバッファサイズは M であるため，同時に M 個までしかパケットを保持できない．また，任意の時刻に，既にバッファに格納されているパケットを破棄することができる．これをプリエンプションという．

一定時間ごとに各出力ポートから 1 つずつ，キューの先頭にあるパケットが送出される．即ち，同一時刻に最大で N 個のパケットが送出される．出力ポートから送出したパケット数の総和が，入力に対するアルゴリズムの価値と定義される．

アルゴリズム A が入力 σ に対して得る価値の総和を $V_A(\sigma)$ と書く．オンラインアルゴリズム A が任意の入力 σ に対して $V_A(\sigma) \geqq V_{OPT}(\sigma)/c$ を満たすとき，A の競合比は c であると言う．

2.2 Longest Queue Drop(LQD) ポリシー

LQD ポリシーは以下のようなオンラインポリシーである．パケットが到着したとき，バッファに空きがあればそのパケットを受理する．もしバッファが満杯であれば，到着したパケットを受理したと仮定したときに，最も多くのパケットを保持するキューの，最も後ろにあるパケットをプリエンプションにより破棄してバッファにパケット 1 個分の空きを作り，到着したパケットを受理する．特に，空きを作るためにプリエンプションされるキューが，到着したパケットの目的キューと同一であれば，到着したパケットが単に破棄されることになる．プリエンプションの候補となるキューが複数ある場合は，任意のキューを選ぶ．以下では LQD ポリシーを LQD と表記する．

3. LQD ポリシーの競合比の改良

解析にあたって，いくつか定義を行う．パケットの到着と送信を，まとめてイベントと呼ぶ．イベントの起こる時刻 t に対して，$t-$ で，時刻 t のイベントが起こる直前で，その 1 つ前のイベントが起こった後の時刻を表す．同様に $t+$ で，時刻 t のイベントが起こった直後で，その 1 つ後のイベントが起こる前の時刻を表す．

スイッチの保持する k 番目のキューを $Q^{(k)}(1 \leqq k \leqq N)$ と表記する．また，時刻 t における，ポリシー A の $Q^{(k)}$ の保持するパケット数を，$h_A^{(k)}(t)$ で表す．引数となる時刻は，イベントの起こった時刻 t に対して，$t-$ や $t+$ を取る．

3.1 解析の概要

[定義 1] OPT がパケットを送信する時に，LQD は同じキューからはパケットを送信しなかったとする．この時，OPT が送信したパケットを extra パケットと呼ぶ．

extra パケットの総数を EX とすると，任意の入力 σ に対して $V_{OPT}(\sigma) \leqq V_{LQD}(\sigma) + EX$ が成り立つ．以下では EX の値の上限を見積もることにより，競合比の上限を与える．その方針を以下に述べる．

入力が全て終了した時点で，OPT により送信されたパケットと，LQD により送信されたパケットのみを考える．OPT の送信した extra パケット 1 個に対して，LQD の送信したパケット $\frac{M}{CHG}$ 個 (CHG は後で定義する) をマッチングさせる (ただし，$\frac{M}{CHG}$ は整数であるとは限らない)．このマッチングを，以下に出てくるマッチングと区別して F-マッチング (Final-マッチングの略) と呼ぶことにする．3.2 節と 3.3 節において，F-マッチングが構成可能であることを証明する (補題 1, 2)．従って，$EX \leqq \frac{CHG}{M} V_{LQD}(\sigma)$ となり，上記の考察より $V_{OPT}(\sigma) \leqq (1 + \frac{CHG}{M}) V_{LQD}(\sigma)$ である．また，3.4 節で，$CHG \leqq \frac{N-1}{N} M$ を示す (補題 3)．これにより，以下の定理を得ることが出来る．

[定理 1] LQD の競合比は $2 - 1/N$ 以下である．

3.2 T-マッチング

上記の F-マッチングを保証するために，任意の時刻において，マッチング (Temporary-マッチング，T-マッチングと書く) を構成する．以下，その具体的方法について述べる．

[定義 2] A を任意のポリシーとし，t をイベントの起こらない時刻とする．時刻 t にパケット p が，ポリシー A のキューの先頭から j 番目に位置するとき，$\ell_A(p,t) = j$ と書く．便宜上，p が時刻 t に既に送信済であれば，$\ell_A(p,t) = 0$ とする．

[定義 3] t をイベントの起らない時刻とする．

$$h_{OPT}^{(j)}(t) - h_{LQD}^{(j)}(t) > 0$$

が成立するキュー $Q^{(j)}$ において，

$$h_{LQD}^{(j)}(t) + 1 \leqq \ell_{LQD}(p,t) \leqq h_{OPT}^{(j)}(t)$$

を満たす OPT のパケット p を，時刻 t における charge パケットと呼ぶ．

charge パケットは，OPT の方が LQD よりも多くパケットを保有しているキューにおいて，OPT がキューの後方に余分に持っているパケットである．定義より，時刻 t がパケットの送信時刻ならば，時刻 $t-$ に OPT のキューの先頭にある charge パケットは，extra パケットとなる．

T-マッチングは，以下のタイプ (a) のパケットとタイプ (b) のパケットとの間のマッチングである．

タイプ (a)

(a-1) 現在 OPT のキューの中にある charge パケット

(a-2) OPT が既に送信した extra パケット

タイプ (b)

(b-1) 現在 LQD のキューの中にあるパケット

(b-2) LQD が既に送信したパケット

ただし，マッチングの条件は，以下の通りである．

条件 1： タイプ (a) のパケット 1 個に対してタイプ (b) のパケット $\frac{M}{CHG}$ 個をマッチさせる．

条件 2： 時刻 t のマッチングで，タイプ (a-1) の charge パケット p にタイプ (b) の LQD のパケット q がマッチさせられるとする．このとき，$\ell_{OPT}(p,t) \geqq \ell_{LQD}(q,t)$ が成立している．

条件 2 は，マッチング可能性の証明を容易にするための条件である．

アルゴリズムが終了した時点では，タイプ (a-1) や (b-1) のパケットは存在しない．すなわち，その時点の T-マッチングは F-マッチングの条件を満たしている．従って，以下の補題が成り立つ．

[補題 1] 任意の時刻に T-マッチングが構成可能であれば，F-マッチングが構成可能である．

3.3 T-マッチング可能性の証明

ここでは，任意の時刻に T-マッチングが構成可能であることを示す．それに先立ち，いくつか定義を行う．

[定義 4] 時刻 t に charge パケットの存在するキューの番号集合を $\mathcal{G}(t)$ とする．即ち，

$$\mathcal{G}(t) = \{j \mid h^{(j)}_{OPT}(t) - h^{(j)}_{LQD}(t) > 0,\ 1 \leqq j \leqq N\}$$

である．

[定義 5] charge パケット数が最大となる時刻の charge パケット数を CHG とする．即ち，

$$CHG = \max_t\{\sum_{j\in\mathcal{G}(t)} (h^{(j)}_{OPT}(t) - h^{(j)}_{LQD}(t))\}$$

である．

[補題 2] 任意の入力に対し，任意の時刻に T-マッチングが構成可能である．

証明. 時刻による帰納法で証明する．初期状態においては，どちらのアルゴリズムもパケットを保持していないので，T-マッチングは成立している．イベントの起こる任意の時刻を t とする．時刻 $t-$ で T-マッチングが構成されていると仮定し，時刻 $t+$ における T-マッチングの具体的な構成方法を示すことにより，補題を証明する．その際，時刻 $t+$ におけるマッチングでは，タイプ (a-2) のパケットに対しては時刻 $t-$ の時のマッチングをそのまま適用させる．これは，条件 2 より，タイプ (a-2) のパケットにマッチされているパケットはタイプ (b-2) のパケットであり，LQD のプリエンプションによりなくなることはないので可能である．そこで，タイプ (a-2) のパケットの $t+$ でのマッチングについては，以後述べないことにする．

時刻 $t-$ に T-マッチングが構成されていると仮定する．時刻 t のイベントとして考えることが出来るのは，パケットの送信かパケットの到着である．

時刻 t のイベントがパケットの送信であった場合，新たな charge パケットは生じない．従って，時刻 $t-$ のマッチングをそのまま時刻 $t+$ のマッチングとすれば，全ての条件は満たされる．

次に，時刻 t のイベントがパケットの到着である場合を考える．この時の OPT の動作としては，パケットの受理と非受理が考えられる．(一般性を失うことなく，OPT はプリエンプションをしないと仮定して良い．もしもプリエンプションを行い，既にキュー内にあるパケットを破棄するなら，前もってそのパケットを受理しなければ良い．) また，LQD の動作としては，受理，非受理，プリエンプション後の受理（この動作は，今後単にプリエンプションと呼ぶ）が考えられる．以下，6 つの場合について見ていく．一般性を失うことなく，到着するパケットは目的ポートが $Q^{(2)}$ であるとし，LQD がプリエンプションを行う場合は，パケットを $Q^{(1)}$ から破棄することにする．

場合 1. OPT：受理，LQD：受理

このとき，$2 \notin \mathcal{G}(t-)$ であれば，パケット受理の前後で charge パケットは変化しない．従って，時刻 $t-$ のマッチングをそのまま時刻 $t+$ のマッチングとすれば良い．

$2 \in \mathcal{G}(t-)$ であれば，パケットを受理した際，OPT のキューで $h^{(2)}_{LQD}(t+)$ の位置のパケット（これを p とする）が charge パケットでなくなり，新しく到着して受理した $h^{(2)}_{OPT}(t+) = h^{(2)}_{OPT}(t-) + 1$ の位置のパケット（これを q とする）が新たに charge パケットとなる．即ち，受理の前後で charge パケット数は変化しないが，charge パケットは変化している．そこで，時刻 $t-$ にパケット p が構成していたマッチングを，時刻 $t+$ ではパケット q と構成させる．$\ell(p,t+) < \ell(q,t+)$ であり，時刻 $t-$ で p に対するマッチングが条件 2 を満たしていたことから，時刻 $t+$ で q についても条件 2 が満たされることが分かる．また，条件 1 が満たされていることは明らかである．p 以外のタイプ (a) のパケットについては，時刻 $t-$ の時と同じマッチングを与えれば良い．

場合 2. OPT：受理，LQD：非受理

$h^{(2)}_{LQD}(t-) - h^{(2)}_{OPT}(t-) > 0$ ならば，時刻 $t+$ での charge パケットは時刻 $t-$ での charge パケットと同一である．よって，時刻 $t+$ のマッチングは時刻 $t-$ のマッチングと同一のものにすれば良い．

一方，$h^{(2)}_{OPT}(t-) - h^{(2)}_{LQD}(t-) \geqq 0$ である場合，時刻 $t-$ から時刻 $t+$ への変化で charge パケットが新たに 1 個増える（p とする）．それ以外の charge パケットは時刻 $t-$ の時と変わらない．従って，p を新たにマッチさせ，p 以外の charge パケットのマッチングは変更しないことにする．

まず，p に条件 1 を満たす数のパケットをマッチさせるために，時刻 $t+$ において LQD のパケットが十分にあることを示す．LQD は到着したパケットを受理しなかったので，時刻 $t+$ でバッファは一杯 ($\sum_{j=1}^{N} h^{(j)}_{LQD}(t+) = M$) であり，時刻 $t+$ での charge パケットの数は定義より高々 CHG 個である．故に，時刻 $t+$ において，LQD のバッファ内には，マッチングを構成するのに十分な数のパケットが存在している．

次に条件 2 について考える．p 以外のパケットは，時刻 $t-$ と同一のマッチングであるため，条件 2 を満たしている．LQD

は p を非受理としているので，LQD の定義より，

$$h^{(2)}_{LQD}(t+) \geqq h^{(j)}_{LQD}(t+) \quad (1 \leqq j \leqq N)$$

が成立している．また，場合分けの条件より $h^{(2)}_{OPT}(t-) - h^{(2)}_{LQD}(t-) \geqq 0$ であり，OPT は p を受理し，LQD は非受理としていることから，$h^{(2)}_{OPT}(t+) > h^{(2)}_{LQD}(t+)$ である．これら 2 つの関係式より，

$$h^{(2)}_{OPT}(t+) > h^{(j)}_{LQD}(t+) \quad (1 \leqq j \leqq N)$$

が成立している．よって，LQD の任意のパケット q に対して $\ell(p,t+) > \ell(q,t+)$ が成立する．従って，p に対するマッチングは条件 2 を満たす．

場合 3．OPT：受理，LQD：プリエンプション

$t+$ におけるマッチングの構成法は，$Q^{(2)}$ のパケットについては，OPT，LQD が共にパケットを受理した場合 1 と同様に考えれば良い．よって，プリエンプションの起こった $Q^{(1)}$ について考える．$Q^{(1)}$ では，OPT のキューの中身は変化なく，LQD はパケットを 1 個捨てている．そのため，もし $h^{(1)}_{LQD}(t-) > h^{(1)}_{OPT}(t-)$ であったなら，charge パケットは変わらないので，$t-$ のマッチングをそのまま採用してやれば良い．次に，$h^{(1)}_{LQD}(t-) \leqq h^{(1)}_{OPT}(t-)$ の場合を考える．LQD はプリエンプションしているので，定義より，時刻 $t+$ において LQD のバッファは満杯である．また，プリエンプションにより LQD は $Q^{(1)}$ のパケットを破棄したので，$Q^{(1)}$ が LQD において最も多くパケットを保持しているキューである．このことから，場合 2 と同様の議論を行うことができ，全ての条件を満たすマッチングが，時刻 $t+$ においても構成可能であることが分かる．

場合 4．OPT：非受理，LQD：受理または非受理

時刻 $t-$ から $t+$ にかけて，新しい charge パケットは生じない．従って，$t+$ のマッチングは $t-$ のマッチングと同一にすれば良い．

場合 5．OPT：非受理，LQD：プリエンプション

場合 3 と同様に考えることが可能である．すなわち，新たにマッチングを考えなければならないのは，$Q^{(1)}$ で LQD がパケットを 1 個破棄したことにより生じる可能性のある charge パケットであるが，LQD が $Q^{(1)}$ からパケットを破棄したという事実から，その charge パケットをマッチさせることが可能である．

以上より，時刻 $t+$ においても T-マッチングが構成可能であることが示された． □

3.4 CHGの評価

最後に本節で，CHG の値の上限を証明する．

［補題 3］ 任意の入力に対し，$CHG \leqq \frac{N-1}{N}M$ が成立する．

証明． 時刻に関する帰納法で示す．

初期状態においては，OPT も LQD もパケットを保持していないので，明らかに条件は成立している．イベントの起こる時刻を t とする．$t-$ で $CHG \leqq \frac{N-1}{N}M$ が成立していると仮定し，$t+$ でも成立することを示す．

まず，時刻 t のイベントがパケットの送信である場合を考える．パケットの送信では，charge パケット数は増加しないので，仮定より $CHG \leqq \frac{N-1}{N}M$ が時刻 $t+$ でも成立している．

次に，時刻 t のイベントがパケットの到着である場合を考える．補題 2 と同様に，OPT と LQD の動作に応じて場合分けする．また補題 2 と同様に，到着するパケットの目的ポートは $Q^{(2)}$ であり，LQD がプリエンプションを行う場合は，パケットを $Q^{(1)}$ から破棄することにする．

場合 1．OPT：受理，LQD：受理

キュー内の charge パケット数は，時刻 t のイベントの前後で変化しないので，$CHG \leqq \frac{N-1}{N}M$ が成立している．

場合 2．OPT：受理，LQD：非受理

まず，$h^{(2)}_{OPT}(t-) \geqq h^{(2)}_{LQD}(t-)$ の場合を考える．LQD は新しく到着したパケットを非受理としたので，その方針から，時刻 $t+$ においてバッファは満杯であり，$Q^{(2)}$ が最長のキューである．よって，

$$h^{(2)}_{LQD}(t+) \geqq \frac{1}{N}M$$

が成り立つ．また，仮定より $h^{(2)}_{OPT}(t-) \geqq h^{(2)}_{LQD}(t-)$ であり，OPT は新しくパケットを受理したので，OPT はキュー $Q^{(2)}$ に，LQD より多くパケットを持っていることになる．すなわち，charge パケットでないパケットを $\frac{1}{N}M$ 個以上保持している．OPT の保持できるパケット数は高々 M なので，時刻 $t+$ で $CHG \leqq \frac{N-1}{N}M$ が成立している．

次に，$h^{(2)}_{OPT}(t-) < h^{(2)}_{LQD}(t-)$ の場合を考える．このとき，時刻 t に $Q^{(2)}$ に到着したパケットを OPT が受理しても，charge パケットの数は増加しない．よって，時刻 $t-$ での仮定から，$CHG \leqq \frac{N-1}{N}M$ が成立している．

場合 3．OPT：受理，LQD：プリエンプション

まず，$Q^{(2)}$ について見ると，OPT も LQD も到着するパケットを受理するので，$Q^{(2)}$ の charge パケット数は変化しない．

続いて LQD によるプリエンプションの発生する $Q^{(1)}$ について考える．最初に，$h^{(1)}_{OPT}(t-) \geqq h^{(1)}_{LQD}(t-)$ の場合を考える．LQD は時刻 $t-$ に保持していたパケットを 1 つ破棄しており，その方針から，

$$h^{(1)}_{LQD}(t+) \geqq \frac{1}{N}M$$

である．$h^{(1)}_{OPT}(t-) \geqq h^{(1)}_{LQD}(t-)$ なので，時刻 $t+$ に $Q^{(1)}$ にあるパケット数は OPT の方が LQD よりも多い．場合 2 と同様に考えることにより，$CHG \leqq \frac{N-1}{N}M$ が成立する．

$h^{(1)}_{OPT}(t) < h^{(1)}_{LQD}(t)$ の場合も，場合 2 と同様に考えることができる．すなわち，LQD が $Q^{(1)}$ からパケットを破棄しても $Q^{(1)}$ の charge パケット数は増加しない．よって，時刻 $t-$ での仮定から，$CHG \leqq \frac{N-1}{N}M$ が成り立つ．

場合 4．OPT：非受理，LQD：受理または非受理

場合 1 と同様に，キュー内の charge パケット数は，パケットの到着の前後で変化しないので，$CHG \leqq \frac{N-1}{N}M$ が成立している．

場合 5. OPT：非受理，LQD：プリエンプション

この場合は，OPT が受理し，LQD がプリエンプションする場合 3 と同様に考えることができる．

上記の議論より，$t+$ においても，$CHG \leqq \frac{N-1}{N}M$ が成立しており，帰納的に，任意の時刻において $CHG \leqq \frac{N-1}{N}M$ が成立していることが言えた． □

4. おわりに

本論文では，共有メモリ型のバッファ管理オンライン問題に対して，既存の競合比の上限を改良した．今後は，上下限を一致させることが重要な課題である．また，このモデルに対する他の設定（価値を複数認めるモデルやプリエンプションを許さないモデル）についての競合比の改良も行いたい．

文　献

[1] W. Aiello, Y. Mansour, S. Rajagopolan, and A. Rosen, "Competitive queue policies for differentiated services," *IEEE INFOCOM*, pp. 431–440, 2000.

[2] S. Albers and M. Schmidt, "On the performance of greedy algorithms in packet buffering," *In Proc. of 36th annual ACM Symposium on Theory of Computing*, pp. 35–44, 2004.

[3] N. Andelman and Y. Mansour, "Competitive management of non-preemptive queues with multiple values," *In Proc. of 17th International Symposium on Distributed Computing*, pp. 166–180, 2003.

[4] N. Andelman, Y. Mansour and A. Zhu, "Competitive queueing policies for QoS switches," *In Proc. of 14th annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 761–770, 2003.

[5] Y. Azar and Y. Richter, "Management of multi-queue switches in QoS networks," *In Proc. of 35th annual ACM Symposium on Theory of Computing*, pp. 82–89, 2003.

[6] N. Bansal, L. Fleischer, T. Kimbrel, M. Mahdian, B. Schieber and M. Sviridenko, "Further improvements in competitive guarantees for QoS buffering," *In Proc. of the 2004 International Colloquium on Automata, Languages, and Programming*, pp. 196–207, 2004.

[7] Y. Bartal, F. Chin, M. Chrobak, S. Fung, W. Jawor, R. Lavi, J. Sgall and T. Tichý, "Online competitive algorithms for maximizing weighted throughput of unit jobs," *In Proc. of 21st International Symposium on Theoretical Aspects of Computer Science*, pp. 187–198, 2004.

[8] A. Borodin and R. El-Yaniv, "Online Computation and Competitive Analysis," *Cambridge University Press*, 1998.

[9] F. Chin and S. Fung, "Online scheduling for partial job values: Does timesharing or randomization help?," Algorithmica, Vol.37, pp, 149–164, 2003,

[10] M. Chrobak, W. Jawor, J. Sgall and T. Tichý, "Improved online algorithms for buffer management in QoS switches," *In Proc. of 12th annual European Symposium on Algorithms*, pp. 204–215, 2004.

[11] E. Hahne, A. Kesselman and Y. Mansour, "Competitive buffer management for shared-memory switches," *In Proc. of 13th annual ACM Symposium on Parallel Algorithms and Architectures*, pp. 53–58, 2001.

[12] B. Hajek, "On the competitiveness of on-line scheduling of unit-length packets with hard deadlines in slotted time," *In Proc. 35th annual Conference on Information Sciences and Systems*, pp. 434–439, 2001.

[13] A. Kesselman, Z. Lotker, Y. Mansour, B. Patt-Shamir, B. Schieber, and M. Sviridenko, "Buffer overflow management in QoS switches," *In Proc. of 33rd annual ACM Symposium on Theory of Computing*, pp. 520–529, 2001.

[14] A. Kesselman and Y. Mansour, "Harmonic buffer management policy for shared memory switches," *Theoretical Computer Science*, Vol. 324, No. 2-3, pp. 161–182, 2004.

[15] A. Kesselman, Y. Mansour and R. Stee, "Improved competitive guarantees for QoS buffering," *In Proc. of 11th annual European Symposium on Algorithms*, pp. 361–372, 2003.

[16] Z. Lotker and B. Patt-Shamir, "Nearly optimal FIFO buffer management for DiffServ," *In Proc. of 21st annual ACM Symposium on Principles of Distributed Computing*, pp. 134–142, 2002.

[17] D. Sleator and R. Tarjan, "Amortized efficiency of list update and paging rules," *CACM 28*, pp. 202–208, 1985.

[18] M. Sviridenko, "A lower bound for on-line algorithms in the FIFO model," unpublished manuscript, 2001.