社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2007－ＳＬＤＭ－129／2007－ＥＭＢ－4（6）
2007／3／15

Responsive Linkを用いた 実時間通信におけるアドミッションコントロール

藤田 裕士† 加藤 真平† 山崎 信行†

† 慶應義塾大学理工学部情報工学科
〒223-8522 横浜市港北区日吉3-14-1
E-mail: †{fujita,shinpei,yamasaki}@ny.ics.keio.ac.jp

あらまし 実時間通信では，新たなコネクションを確立する際に，既に確立されたコネクションの実時間性を保証するアドミッションコントロールが必要である．Responsive Linkを用いたマルチホップネットワークにおける従来の実時間通信では，送信ノードから受信ノードまでのデッドラインを，経路上の各ノード間で分割して設けた仮想的なデッドラインに基づいてスケジュール可能性判定を行い，コネクションの受入れを決定する．本研究で提案するアドミッションコントロールは，従来の手法では受入れられないコネクションでも，スケジュール可能性解析によって仮想デッドラインを再設定することで，受入れを可能にする．

キーワード 実時間通信，マルチホップネットワーク，コネクション指向，アドミッションコントロール

Admission Control in Real-Time Communication using Responsive Link

Yuji FUJITA†, Shinpei KATO†, and Nobuyuki YAMASAKI†

† Keio University
3-13-1 Hiyoshi, Kouhoku-ku Yokohama, Kanagawa 223-8522 Japan
E-mail: †{fujita,shinpei,yamasaki}@ny.ics.keio.ac.jp

Abstract In real-time communication, when a connection is established, the admission control is required in order to guarantee real-time performance of connections already established. In existing method of real-time communication with Responsive Link, the deadline from a source node to a destination node is divided between each node to set an virtual deadline, a schedulability test is performed based on those virtual deadlines, and acceptance of a connection is determined. It is possible to accept the connection that is not accepted in a existing method because of setting virtual deadline by schedulability analysis again with the admission control proposes by this research.

Key words Real-Time Communication, Multihop Network, Connection Oriented, Admission Control

1. はじめに

近年，一つのプロセッサで行う処理の限界や，システムの耐故障性等を向上するために，組込み実時間システムでも，プロセッサを複数用いた分散実時間処理への要求が高まっている．分散実時間処理を実現するためには，ノード内の処理に加え，ノード間の処理に必要となる通信の実時間性を保証しなければならない．特に，ロボット制御や自動車制御などでは分散実時間制御を行う必要があり，各ノード間で数100μsec単位の時間粒度の細かい実時間性の保証が必要になる．このような分散実時間制御用の通信規格としてResponsive Link [6] がある．Responsive Linkは各ノードで優先度によるパケットの追い越し機構を持っており，分散環境において実時間通信を実現するのに適している．また，パケットの追い越しをハードウェアで行うために，低遅延を実現しており，時間粒度の細かい実時間通信が実現可能である．しかしながら，Responsive Linkにおいて実時間通信を行うためには，ソフトウェアによる通信管理機構が必要になる．

実時間通信を行うために，実際の通信に先立ってコネクションを確立することによって実時間性を保証する．また，新たにコネクションを確立する際には，既に確立されたコネクションの実時間性を保証するために，アドミッションコントロールを行う必要がある．Responsive Linkを用いた従来の実時間通信では，送信ノードから受信ノードまでのデッドラインを経路上の各ノード間で分割して仮想的なデッドラインを設け，その仮想デッドラインに基づいてスケジュール可能性判定を行い，そ

の結果によって，コネクションを受入れを決定する．本研究では，Responsive Link を用いたマルチホップネットワークにおいて，実時間通信のためのアドミッションコントロールを提案する．本研究で提案するアドミッションコントロールは，コネクション確立の際，設定した仮想デッドラインに基づいてスケジュール可能性判定を行い，コネクションが受け入れられない場合でも，スケジュール可能性解析を行い，仮想デッドラインを再設定することで，コネクションの受入れを可能にする．

2. Responsive Link

Responsive Link は，我々の研究室において研究開発を行っている分散実時間処理用プロセッサ Responsive Multithread Processor [4] が持つ実時間通信規格である．

Responsive Link は，画像や音声などのスループットを上げることが求められるソフトリアルタイム通信と制御コマンドや同期信号などのレイテンシを小さくすることを求められるハードリアルタイム通信の通信路を分離し，両方を同時に行うことが可能である．それぞれの通信路をデータリンク，イベントリンクと呼ぶ．通信パケットは固定長で，ソフトリアルタイム通信用のデータリンクのパケットサイズは 64byte，ハードリアルタイム通信用のイベントリンクのパケットサイズは 16byte となっている．各リンクは point-to-point で接続され，トポロジーフリーであることから，柔軟な接続形態でネットワークを構築することができる．各リンクは双方向の通信が可能であるように設計されている．

Responsive Link は，通信パケットに優先度を付け，高い優先度の通信パケットが低い優先度の通信パケットを通信ノードごとに追い越しを行う．ハードウェアによりサポートする優先度は 256 段階あり，ネットワークスイッチはこの優先度を用いて出力における衝突を瞬停する．各スイッチには追い越し用バッファと退避用外部記憶があり，通信パケットが衝突しない場合はそのまま出力されるが，異なるポートから入力された通信パケットが同じポートに出力を行う場合，通信パケットに付加された優先度に従い，低い優先度の通信パケットを追い越し用バッファに貯めて出力を待たせ，高い優先度の通信パケットを先に出力させる．追い越し用バッファが溢れそうになった場合には，退避用外部記憶に一時的に退避する．パケットの追い越しは，プロセスのコンテキストスイッチに相当し，スケジューラによる固定優先度タスクスケジューリングに置き換えて考えることができる．従って，Rate Monotonic [3] 等のリアルタイムスケジューリングの理論を応用することができる．また，優先度によって，同じ送信元アドレス及び送信先アドレスを持つ通信パケットでも別の経路に設定することが可能であり，専用回線や迂回路を設けることによって実時間通信を制御できる．通信ノードごとに優先度の付け替えができるので，分散管理による実時間通信の制御が可能となっている．

3. 関連研究

Responsive Link に関する研究に佐々木らによる実時間通信管理機構がある [5]．この研究では，Responisive Link を用いたマルチホップネットワークにおいて，実時間通信を実現しており，通信ライブラリの提供，コネクションの管理の設計と実装，送受信処理の設計と実装を行っている．

通信ライブラリを用いて，コネクション確立要求受付，コネクション確立要求，コネクション切断要求，データの送信，データの受信を行うことが可能である．コネクションの管理は，集中管理と比べて高いスケーラビリティとフォルトトレラントなシステムを構築可能するために分散管理を用いて行っている．コネクション確立は，リクエストフェーズ，ルーティングフェーズ，リザベーションフェーズ，リラクゼーションフェーズの 4 フェーズで行っている．文献 [2] では，コネクションの確立要求は送信ノードで出すが，この通信管理機構ではコネクションの確立はデータを受信する受信ノードが望むとして，受信ノードから送信ノードに対しコネクションの確立要求を出す．送受信処理は，Responsive Link の Dual Port Memory の制御レジスタを設定することによって行っている．

この通信管理機構では，動的な環境に適用するために，コネクション確立の際にアドミッションコントロールを行っている．コネクションの経由する各ノード間でスケジュール可能性判定を行い，既に確立されているコネクションの実時間性を侵害する可能性がある場合にはコネクションの確立を破棄し，確立しようとするコネクションの実時間性が保証できない場合にもコネクションの確立を破棄する．既に確立されているコネクションの実時間性を侵害しないこと，及び，確立しようとするコネクションの実時間性が保証できる場合にのみコネクションを確立する．このように，コネクションの確立を制御することによって各コネクションの通信遅延の保証を行っている．

Responsive Link を用いたマルチホップネットワークの通信では，コネクションは経由する各ノードにおいて，優先度の高いコネクションによって追い越しバッファに待たされる．図 1 は，ノード間の通信路である $Link_1$ から $Link_4$ を通って通信を行うコネクション M_i が、各通信路で優先度の高いコネクションに待たされている様子を示す．コネクションはノードをホップするたびにリンクを経由するので，ホップ数の多いコネクションほど待たされる時間が長くなる．周期を基にしてデッドラインを決定した場合，ホップ数に関わらず周期が短いコネクションに高い優先度が割り当てられる．そのため，ホップ数の多いコネクションは，デッドラインミスする可能性が高くなり，スケジュール可能性は低下する．例えば，周期が同じでも，ホップ数が多いコネクションは受入率が低い．この通信管理機構では，以下のようにホップ数の多いコネクションの優先度を上げ，スケジュール可能性を高めている．

送信ノードから受信ノードまでのデッドラインを相対デッドラインと呼び，式 (1) に示すように等分に分割し，その分割された値を基に優先度を決定する．分割されたデッドラインを仮想デッドラインと呼ぶ．

$$vd = \frac{D}{hops} \tag{1}$$

式 (1) では，vd は仮想デッドライン，D は相対デッドライン，$hops$ はホップ数を表す．

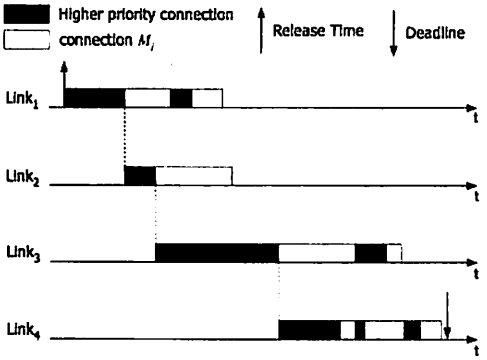


図 1 Responsive Link における通信の様子

仮想デッドラインを用いてスケジュール可能性判定を行い，各リンクにおいて確立しようとしているコネクションの実時間性と，既に確立されているコネクションの実時間性が保証されるときにコネクションを確立する．この仮想デッドラインに基づいた優先度決定を用いることにより，周期に基づいた優先度決定に比べ，高いコネクション受入率を実現している．さらに，相対デッドラインを等分に分割する手法を発展させたものとして，経由する各リンクの負荷を基にして仮想デッドラインを決定する手法を提案している．負荷を基にした仮想デッドラインでは，他のノードに邪魔されてもよい時間を負荷に応じて分割し，各リンクの仮想デッドラインに割り当てている．

$$vd = (D - C \times hops) \times \frac{load}{total\ load} + C \qquad (2)$$

式 (2) では，C は伝送時間，$(D - C \times hops)$ は他のコネクションに邪魔されてもよい時間を表す．$load$ はリンクの負荷を表し，リンク上で確立しているコネクションの周期 T_i，伝送時間式 C_i によって式 (3) として表される．

$$load = \sum \frac{C_i}{T_i} \qquad (3)$$

$total\ load$ は確立しようとするコネクションの経由する全てのリンクの負荷の合計を表す．従って，$\frac{load}{total\ load}$ は経由する各リンクの総負荷に対する割合である．ネットワークの負荷にばらつきがある場合，負荷に応じて分割した仮想デッドラインによる優先度決定法は，等分に分割した仮想デッドラインによる優先度決定法に比べ，コネクション受入率が向上する．

従来のアドミッションコントロールには，さらにコネクション受入率を向上させる余地があると考える．本研究ではこの手法について提案する．

4. 設　　計

この章では，Responisve Link を用いた実時間通信における本研究で提案するアドミッションコントロールについて述べる．

4.1 ネットワークモデル

本研究では Responsive Link によって point-to-point で結合され，通信パケットが複数のノードを経由するネットワークを想定する．各リンクは同じ帯域幅であるとし，ネットワークでは周期メッセージのみを扱う．ネットワーク上で実時間通信を行うために確立するコネクションのパラメータを表 1 に示す．相対デッドラインは周期と等しいとする．コネクションの優先度 p_i は仮想デッドラインを基に決定し，仮想デッドラインが小さいコネクションに高い優先度を割り当てる．負荷を基にした仮想デッドライン（式 2）を用いると，コネクションの経由するリンク間で優先度が変わる可能性がある．

表 1 パラメータ

記号	説明
M_i	コネクション
T_i	M_i の周期
C_i	M_i の伝送時間
D_i	M_i の相対デッドライン

送信ノードと受信ノード間には複数の経路の候補があり、個々の通信パケットが異なるノードを経由した場合，通信遅延を予測することが難しくなる．従って，確立したコネクションではメッセージは同じ経路を通るようにする．ルーティングには動的ルーティングを用いるとネットワークの状態にあわせて最適な経路を選択することができるが，メッセージが大量に発生するためオーバヘッドが大きい．従って，静的ルーティングである Shortest Path First (SPF) を用いる．

4.2 スケジュール可能性判定

コネクションの実時間性を保証するためにはスケジュール可能性判定を行う必要がある．スケジュール可能性判定は，Response Time Analysis (RTA) [1] を基にして行う．RTA では最悪応答時間とデッドラインを比較してスケジュール可能性が判定される．隣接ノード間のパケットスケジューリングにおける応答時間は，ノードにメッセージが到着してから，出力先のノードにメッセージを送信し終わるまでに当たる．最悪応答時間は最も時間がかかる場合を想定したときの応答時間である．

パケットの到着は経由してきた通信路の状態に依存するため，いつ到着するのかを正確に知ることはできない．この状況は，タスクスケジューリングにおけるジッタを考慮した最悪応答時間として考えることができ，高優先度 M_j のメッセージ到着から送信開始されるまでの許容される最大の遅れ $(T_j - C_j)$ をジッタと見なして，M_i の $Link_n$ における最悪応答時間 WRT_i^n を以下のように表すことができる．

$$WRT_i^n = \sum_{p_j > p_i} (\lceil \frac{WRT^n + T_j - C_j}{T_j} \rceil \times C_j) + C_i \qquad (4)$$

式 (4) では，両辺に WRT_i^n が含まれており解くことができないが，以下の漸化式を解くことによって WRT_i^n 求めることができる．

$$Z_i^{m+1} = \sum_{p_j > p_i} (\lceil \frac{Z_i^{m+1} + T_j - C_j}{T_j} \rceil \times C_j) + C_i \qquad (5)$$

式 (5) で $Z_i^0 = C_i$ とし $Z^{m+1} = Z^m$ となるまで再帰する．そのときの Z^m が最悪応答時間 WRT_i^n となる．

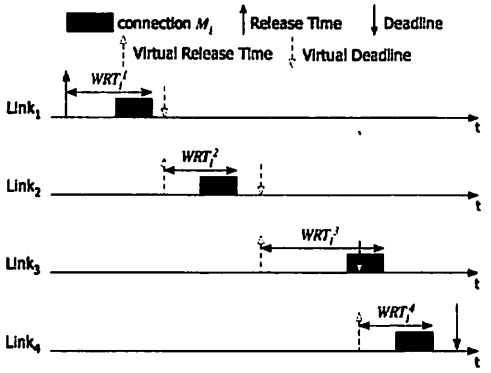


図 2　従来手法では受入不可能なコネクション

4.3 設計方針

従来のアドミッションコントロールは，コネクション確立時に経由する各リンク毎に行う．新たに確立しようとするコネクションの実時間性と既に確立されたコネクションの実時間性が保証される場合に，新しいコネクションを受け入れる．新たに確立しようとするコネクションの実時間性を保証しようとするとき，スケジュール可能性判定を用いて，確立しようとするコネクションの最悪応答時間を求める．そして，最悪応答時間と仮想デッドラインを比較し，最悪応答時間が大きければ，コネクションの実時間性を保証不可能として，コネクションを破棄する．このとき，相対デッドラインまでに通信メッセージをスケジュール可能だとしても，コネクションの経由するリンク上で一ヶ所でも最悪応答時間が仮想デッドラインを超えてしまった場合，コネクションは破棄される．

図 2 は，従来のアドミッションコントロールでは受入不可能なコネクションを示す．M_i は送信ノードから受信ノードまでに，$Link_1$ から $Link_4$ を経由するコネクションである．M_i はリンク 3 において，最悪応答時間は仮想デッドラインをミスしている．従って，M_i は実時間性を保証不可能としてコネクションを破棄する．しかし，図 2 の場合，相対デッドラインまでには受信ノードに全てのメッセージを送ることが可能である．

以上のようなコネクションを受け入れるため，仮想デッドラインをミスしても，コネクションを破棄しないようにする．コネクションの経由する全てのリンクについて，スケジュール可能性判定を行った後，相対デッドラインを満たせないならば，コネクションを破棄する．

仮想デッドラインをミスしたが受入可能なコネクションは，仮想デッドラインを再設定し，スケジュール可能性判定を行えるようにする．再設定によって，仮想デッドラインの値を変更するが，優先度を変更すると，他のコネクションの実時間性を侵害する恐れがある．従って，元の仮想デッドラインに基づいた優先度を用いる．

4.4 アドミッションコントロール

仮想デッドラインを変更し，コネクションを受け入れるために，相対デッドラインまでにコネクションの通信メッセージを

表 2　スケジュール可能性解析に用いる値

記号	説明
$over_i^n$	最悪応答時間が仮想デッドラインから超過した時間
$remain_i^n$	仮想デッドラインから最悪応答時間を引いた余裕時間
$total\ over$	$over_i^n$ の合計
$total\ remain$	$remain_i^n$ の合計

```
1: if(WRT_i^n > vd)
2:     over_i^n = WRT_i^n − vd
3:     total over = total over + over
4: else
5:     remain_i^n = vd − WRT_i^n
6:     total remain = total remain + remain
7: endif
```

図 3　スケジュール可能性解析

```
1: if(total remain < total over)
2:     return FALSE
3: return SUCCESS
```

図 4　全ノードで最悪応答時間計算後のアドミッションコントロール

スケジュール可能性解析を行う必要がある．表 2 で示す値を用いる．ノード内のテーブルに $over_i^n$ と $remain_i^n$ を保持し，コネクション確立のためにノード間で受け渡すメッセージに $total\ over$ と $total\ remain$ を保持する．

コネクションを確立する際に経由する各リンクで行うアドミッションコントロールの流れを示す．

RTA を用いて確立しようとするコネクションの最悪応答時間を算出する．次にスケジュール可能性解析を行うために，図 3 で示す疑似コードで示すように，最悪応答時間と仮想デッドラインの比較を行い，表 2 の値に代入する．ノード内に保持されている $over_i^n$ と $remain_i^n$ の値は，コネクションが破棄されたら削除する．$total\ over$ と $total\ remain$ は，コネクション確立のためのメッセージとして次のノードに渡す．ここで確立しようとするコネクションが既に確立されているコネクションの実時間性を侵害する時は，コネクションは破棄される

次に，全ノードで最悪応答時間を計算した後に行うアドミッションコントロールの流れを示す．図 4 に疑似コードを示す．

$total\ remain$ と $total\ over$ を比較し，$total\ over$ の方が大きいならば，コネクションを破棄する．$total\ remain$ の方が大きければ，ネットワークはコネクションを確立する．このアドミッションコントロールによって，仮想デッドラインをミスしても，送信ノードから受信ノードまでの最悪応答時間が相対デッドラインよりも小さければコネクションを確立できることを実現する．

4.5 デッドラインの再設定

提案するアドミッションコントロールを用いて，はじめに設定した仮想デッドラインを超えるコネクションまで受け入れると，新たにコネクションを確立するときにスケジュール可能性判定を行うことができない．経由する各リンクで最悪応答時間が仮想デッドラインを超えてしまった場合に，仮想デッドライ

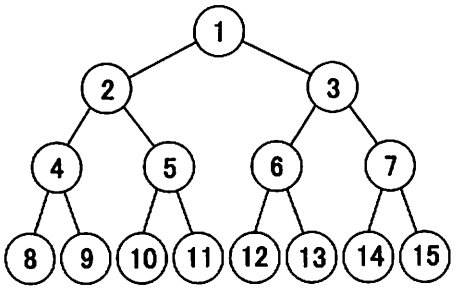


図 5　シミュレーションを行ったネットワークトポロジー
（ツリー構造）

ンの再設定を行う．このアドミッションコントロールの時に得た値を用いる．アドミッションコントロールにより受け入れ可能ならば，超過したノード間では仮想デッドラインを（式 6）で示すように再設定した．仮想デッドラインをミスしているので超過した分だけ足した．

$$vd = vd + over_i^n \tag{6}$$

受入可能な時間があるノード間では，$remain_i^n$ は各リンクによってばらつきがあると考え，(式 7) で示すように仮想デッドラインから残っている時間に応じて時間を減らした．

$$vd = vd - \frac{remain_i^n}{total\ remain} \times over_i^n \tag{7}$$

$Over_i^n$ は M_i が $link_n$ で仮想デッドラインから超過した時間を表す．

5. 評　　価

本研究では，ネットワーク利用率を変化させ，従来のアドミッションコントロールを行った場合と，提案するアドミッションコントロールを行った場合で，コネクション受入率がどのように変化するのか，シミュレーションによって比較を行った．

ネットワークトポロジーにはツリー構造（図 5）及び 4 × 4 の二次元トーラス（図 6）の 2 種類を用いた．制御用の分散システムではツリー構造をとる場合が多い．また，ツリー構造とは別の典型的なネットワークトポロジーとして二次元トーラスを用いた．図 5, 6 では，ノード間は双方向の通信が行ことが可能である．

コネクションの送信ノードと受信ノードはランダムに決定する．ネットワーク利用率 U は，確立しているコネクション M_i の周期 T_i，伝送時間 C_i，ネットワークトポロジーの通信路の数 $links$ より式 (8) として定義した．

$$U = \sum \frac{C_i}{T_i} \times \frac{1}{links} \tag{8}$$

コネクションの確立によってネットワーク利用率を調整する．評価を行う手法を用いてコネクションを確立し，ネットワーク利用率を上げる．コネクションを確立してもすぐに解放することによって，ネットワーク利用率を一定にしてコネクション受入率を測定する．コネクションの優先度は，相対デッドライン

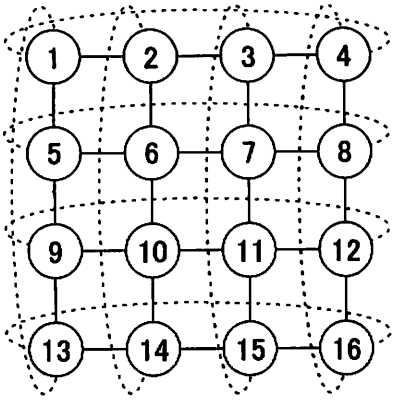


図 6　シミュレーションを行ったネットワークトポロジー
（4 × 4 の二次元トーラス）

を等分に分割した仮想デッドライン（式 1）と，負荷に応じて分割した仮想デッドライン（式 2）を基に決定する手法をそれぞれ用いた．確立するコネクションは，周期は [100,1000] の一様分布，伝送時間は [10,50] の一様分布である．ネットワーク利用率に対して，コネクション受入率はどのようになるかを評価する．

ツリー構造において，従来のアドミッションコントロールを行った場合と，提案するアドミッションコントロールを行った場合のコネクション受入率を図 7 に示す．

図 7 より，ネットワーク利用率が 0.05 から 0.2 の時において，提案手法と負荷に応じた優先度決定法を用いたコネクション確立手法が最も高いコネクション受入率となっている．また，負荷を考慮した従来手法より，最大で約 10 %の受入率向上を実現できている．また，相対デッドラインを等分に分割する優先度決定法を用いたコネクション確立法でも，提案手法を用いたときの方がコネクション受入率は最大で約 5%向上した．ツリー構造では負荷のばらつきが生じやすく，特にルートノードとのリンクに負荷が高くなりやすい．負荷に応じたデッドライン分割による優先度決定でもコネクションを確立できなかった場合でも．提案した手法がうまく働きコネクション受入率を向上させることができた．ネットワーク利用率が高くなると，どの手法もコネクションは受入率は低下し，どの手法もほぼ同じ値となった．

トーラスにおいて，従来のアドミッションコントロールを行った場合と，提案するアドミッションコントロールを行った場合のコネクション受入率を図 8 に示す．

図 8 より，トーラスでの評価は，ネットワーク利用率が 0.15 のときに，優先度決定法による差はあるものの，それ以外には大きな差は見られない．これは送信ノードと受信ノードをランダムに決定していることや，トーラスは通信経路が多く存在するので，負荷が分散され，各リンクの負荷がほぼ均一になったからだと考えられる．

提案したアドミッションコントロールは，ネットワークの負

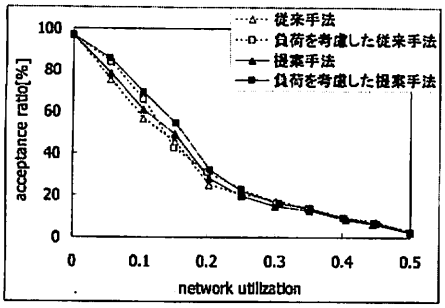


図 7　ツリー構造の評価結果

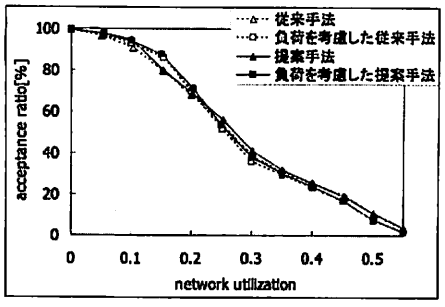


図 8　トーラスの評価結果

荷にばら付きがあり，仮想デッドラインをミスするリンクとミスしないリンクが同じコネクション上に存在するときに効果があると考える．また，仮想デッドラインをミスする状況は，高優先度のコネクションを確立したネットワークにおいてコネクションの確立を試みるときに発生する可能性が高い．従って，従来手法と提案手法で差が見られなかったトーラスに対して，特定のリンクに高優先度のコネクションを確立し，その後で，低い優先度のコネクションを確立し，コネクション受入率を評価した．

高優先度のコネクションの送信ノードと受信ノードは図 6 の 5 から 12 の中からランダムに選ばれる．高優先度のコネクションの周期は [100,200] の一様分布で，伝送時間は [50,60] の一様分布とした．低優先度のコネクションの周期は [1000,1500] の一様分布で，伝送時間は [10,50] の一様分布であり，送信ノードと受信ノードはランダムに選んだ．

図 9 は，トーラスについて，あらかじめネットワークの一部に高優先度のコネクションを確立し，低優先度のコネクションの受入率を示すグラフである．図 9 では，ネットワーク利用率が 0.3 になるまでに，優先度の決定手法に関わらず，提案手法は従来手法に比べ，最大約 5 % の受入率が向上している．ネットワークにあらかじめ負荷をかけないときの評価と比べると，本手法によるコネクション受入率が上がっている．また，負荷を考慮しない従来手法よりも，負荷を考慮した提案手法では，最大で受入率が 20%向上している．一部に高優先度のコネクションが確立しているネットワークにおいて，低優先度のコネクションの受入率が向上した．提案手法は負荷を考慮した優先度決定を用いることにより，高いコネクション受入率を示している．

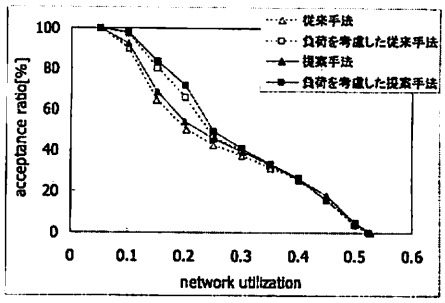


図 9　あらかじめ負荷をかけたときのトーラスの評価結果

6. 結　　論

本研究では，Responsive Link を用いたマルチホップネットワークにおいて，実時間通信を行うためのアドミッションコントロールについて述べた．本研究で提案するアドミッションコントロールは，コネクションが受け入れられない場合でも，スケジュール可能性解析を行い，仮想デッドラインを再設定することで，コネクションの受入率を向上することができる．

シミュレーションの評価により，ネットワークの負荷が均等でない場合に，コネクション受入率を高めることに成功した．提案手法を用いることで，同じ優先度決定法の従来手法に比べ，コネクションの受入率を最大約 10%向上した．また，負荷に応じた相対デッドライン分割による優先度決定法を同時に用いることにより高いコネクション受入率を示し，負荷を考慮しない従来手法と比べて最大で約 20%向上できた．

今後の課題として，実機に対して実装を行い評価をとる．また，今回は計算量を考慮していないので，スケジュール可能性判定に時間がかかる可能性がある．従って，オーバーヘッドを減らすように改良する必要がある．

謝辞　本研究は，科学技術振興機構 CREST の支援による．

文　　献

[1] N. Audsley, A. Burns, M. Richardson, K. Tindell, and A.J. Wellings. Applying New Scheduling Theory to Static Priority Pre-emptive Scheduling. *Software Engineering Journal*, 8(5):284–292, 1993.

[2] D. Ferrari and D.C. Verma. A scheme for real-time channel establishment in wide-area networks. *Selected Areas in Communications, IEEE Journal on*, 8(3):368–379, 1990.

[3] C.L. Liu and J.W. Layland. Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. *Journal of the ACM (JACM)*, 20(1):46–61, 1973.

[4] N. Yamasaki. Responsive Multithreaded Processor for Distributed Real-Time System. *Journal of Robotics and Mechatronics*, 17(2):130–118, 2005.

[5] 佐々木 貴宏. Responsive Link を用いたリアルタイム通信管理機構. Master's thesis, 慶應義塾大学大学院, 2006.

[6] 山崎信行. 分散制御用リアルタイム通信 Responsive Link の設計及び実装. 情報処理学会論文誌コンピューティングシステム, 45(SIG 3 (ACS 5)):50–63, 2004.