Vol.2012-SLDM-155 No.24
Vol.2012-EMB-24 No.24
2012/3/3

Responsive Linkを用いた分散リアルタイムシステムにおけるルーティング手法

吉住 修[†1] 松谷宏紀[†1] 山崎信行[†1]

分散リアルタイムシステムではデータ転送において時間制約があるため，時間制約を満たすようにパケットをルーティングする必要がある．従来の Responsive Link を用いた分散リアルタイムシステムにおける通信では，実行前に送信元から宛先までの最短経路を設定した上で，各パケットに対してスケジュール可能性判定を行い受け入れを決定してきた．このとき，パケット同士の依存関係によってデッドロックが発生するため，各パケットの経路を設定する際に循環依存が生じないことを保証する必要がある．本論文では，Responsive Link を用いた分散リアルタイムシステムにおいて，リアルタイム性およびデッドロックフリーを保証するルーティング手法を提案する．シミュレーションによる評価の結果，提案手法が従来のルーティング手法と比較して最大 45% 受け入れ可能な転送データ量を向上させることができた．

Packet Routing for Distributed Read-Time System using Responsive Link

Osamu Yoshizumi,[†1] Hiroki Matsutani[†1]
and Nobuyuki Yamasaki[†1]

Since distributed real-time systems require strict time constraints on communication, packets must be routed so as to meet the real-time constraints. In our previous communication schemes using Responsive Link for distributed real-time systems, first, a minimal path for each source destination pair is selected, and then its schedulability is verified if the selected path meets the constraints. Since such routing schemes may form cyclic dependencies across multiple packets, routing schemes must be designed so as to guarantee deadlock-freedom. In this paper, we propose routing schemes of Responsive Link that guarantee both real-time capability and deadlock-freedom for distributed real-time systems. Simulation results show that the proposed routing schemes improve the schedulability of communications by up to 45%.

1. はじめに

ロボット制御や自動車制御などに代表される分散リアルタイムシステムでは，リアルタイム性を保証するため通信遅延の削減が求められる．このような用途では，各ノード間で周期的に転送されるデータに対して数 $100\mu sec$ 単位の時間粒度の細かいリアルタイム性の保証が必要になる．

分散リアルタイムシステム用の通信規格として Responsive Link[1] がある．Responsive Link は分散リアルタイムシステムにおける通信を実現するために，通信のプリエンプション（横取り）をパケット追い越し機能によって実現している．さらに，柔軟なリアルタイム通信を実現するために，ソフトリアルタイム通信（データリンク）とハードリアルタイム通信（イベントリンク）の分離，異なる優先度を持つパケットに対する別経路（専用回線や迂回路等）の実現，ノード毎の優先度の付け替えなどで分散管理型でパケットの加減速制御などの機能を実現する．また，Responsive Link は Virtual Cut Through(VCT) 方式[2] でのパケット転送が可能である．VCT 方式では，出力ポートが開いていればパケット全体の到着を待つことなく即座にスイッチされるため，低い通信遅延が実現可能である．

しかし，閉路が存在するシステムにおいて VCT 方式による転送を行った場合，デッドロックが発生する可能性がある．デッドロックが発生した場合，システム内のパケットが互いのバッファを占有し合い，パケットの転送が不可能となるため，デッドロックが発生しない経路群の設定が必要となる．

Up/Down ルーティング[3] は，システム内で設定可能な経路を限定するデッドロックフリーなルーティング手法である．Up/Down ルーティングはシステム内の 1 ノードをルートノードとした後に，ルートノードの位置を考慮した経路を生成する．分散リアルタイムシステムにおいて Up/Down ルーティングを利用する場合，システム内の各通信パケットに対する時間制約を保証する必要があるため，適切なルートノードを選択する必要がある．そこで，本研究では Up/Down ルーティングを用いて時間制約を考慮した経路群の選択を行うアルゴリズムを提案する．本論文の構成は次の通りである．まず 2 章では本論文で対象とする通信リンクである Responsive Link について述べる．3 章では，本論文の関連研究について述べる．4 章では，本論文で提案するルーティング手法について述べる．5 章では，従来

†1 慶應義塾大学大学院理工学研究科開放環境科学専攻
Department of Computer Science, Graduate School of Science and Technology, Keio University

Vol.2012-SLDM-155 No.24
Vol.2012-EMB-24 No.24
2012/3/3

手法と提案手法の比較評価について述べる．最後に 6 章で結論を述べる．

2. Responsive Link

Responsive Link は，ISO/IEC で国際標準化されている分散リアルタイムシステム向けの通信規格であり，分散リアルタイム処理用プロセッサ Responsive Multithreaded Processor[4] に実装されている．

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通信遅延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を分離した通信規格である．ソフトリアルタイム通信，及びハードリアルタイム通信が行われる通信路をそれぞれデータリンク，イベントリンクと呼ぶ．通信パケットは固定長で，データリンクは 64bytes，イベントリンクは 16bytes である．各通信リンクは point-to-point で接続され，トポロジーフリーである．各通信リンクは双方向の通信が可能となるように設計されている．

Responsive Link は，通信パケットに対して優先度を付与し，各ノードにおいて高優先度パケットが低優先度パケットの追い越しが可能であり，Rate Monotonic[5] 等のリアルタイムスケジューリング理論が適用可能である．また，Responsive Link は優先度に応じた経路の変更が可能であり，同じ送信元アドレスと送信先アドレスを持つ通信パケットに対して異なるルーティングが可能であるため，専用回線や迂回路を設けることによってリアルタイム性が求められる通信の制御が可能である．Responsive Link は，Store and Forward 方式[2] と VCT 方式の 2 種類のパケット転送方式が適用可能である．Store and Forward 方式は，ノードに通信パケット全体が到着した時点で次ノードへの転送が開始される．一方，VCT 方式は，ノードに通信パケットのヘッダ部が到着した時点で次ノードへの転送が開始されるため，低い通信遅延が実現可能である．本研究では，VCT 方式で通信パケットの転送を行うこととする．

3. 関 連 研 究

トポロジが不規則なネットワークにおけるデッドロックフリーなルーティング手法として，Spanning Tree Protocol(STP)[6] がある．

STP は，システムを点と線で構成されるグラフとみなし，システム内の一部を利用禁止とすることで，システム内の閉路を回避するアルゴリズムである．STP を利用することで，デッドロックが発生しないルーティングが可能である．STP は，ID によって選択されたルートノードを根とした深さが最小となる木を構成した後，木に含まれない通信リンクを使用禁止とする．図 1 に，ノード数 8 の分散システムにおいて STP を用いてノード G からノード B に通信パケットを転送する場合のルーティング例を示す．図 1 では，通信パケットが利用禁止である通信リンクを利用せずに通信している．しかし STP では，システム内に通信パケットが使用できない通信リンクが生じてしまうため，システム内の通信パケットの転送ホップ数が増加してしまうと同時に各ノードにおける通信パケットの衝突が増加してしまう．そのため，通信パケットを効率的に転送可能なルーティング手法が必要となる．

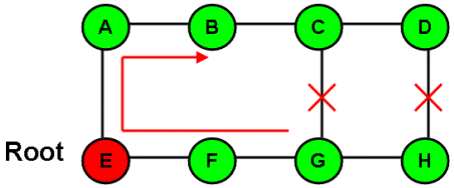
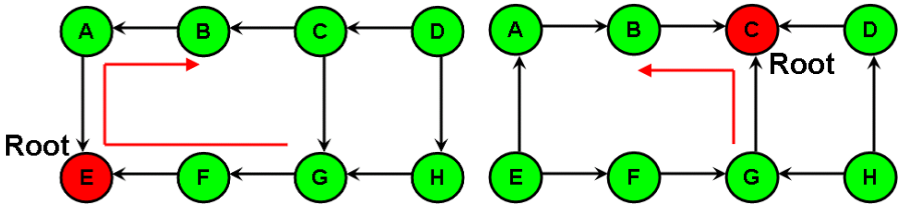


図 1 Spanning Tree Protocol(STP)
Fig. 1 Spanning Tree Protocol(STP)

STP を基にしたルーティングアルゴリズムは並列計算機の分野でも利用されてきた．例えば，Up/Down ルーティングは Autonet[3] や Myrinet[7] などのネットワークで既に利用されている，デッドロックフリーなルーティング手法である．Up/Down ルーティングでは，分散システム内の 1 ノードをルートノードと定義し，各ノードにおいてルートノードに近づく方向を Up，ルートノードから離れる方向を Down と定義する．Up/Down ルーティングでは，Up 方向に進行した後に Down 方向に進行する，または Down 方向，Up 方向のみに進行することを許可する．Up/Down ルーティングでは Down 方向に進行した後に Up 方向に進行することを許可しないことで，デッドロックを回避する．

図 2 に，ノード数 8 の分散システムにおいて Up/Down ルーティングを用いてノード G からノード B に通信パケットを転送する場合のルーティング例を示す．図 2(a) では，ルートノードをノード E としておりノード F，E，A を経由してノード B に到達する経路のみ許可させるが，ノード C を経由してノード B に到着する経路は Down 方向に進行した後に Up 方向に進行するため許可しない．一方，図 2(b) では，ルートノードをノード C としており，ノード C を経由してノード B に到着する経路を Up 方向に進行した後に Down 方向に進行するため許可する．図 2 に示すように，ルートノードの選択によって通信パケットが経由するホップ数が変化する．

Vol.2012-SLDM-155 No.24
Vol.2012-EMB-24 No.24
2012/3/3



(a) ルートノード:E　　(b) ルートノード:C

図 2 Up/Down ルーティング
Fig. 2 Up/Down Routing

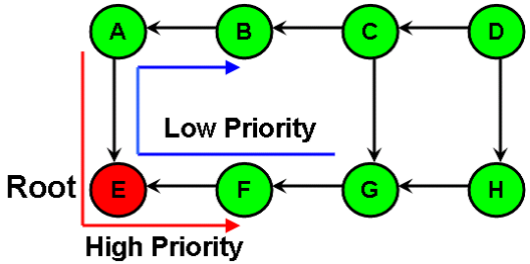


図 3 提案手法
Fig. 3 Proposed routing scheme

分散リアルタイムシステムでは，時間制約が厳しい通信パケットに対して少ないホップ数で転送することで転送データの受け入れ可能性が向上するため，システム内の通信パケットの時間制約を考慮してルートノードを選択する必要がある．

4. 提 案 手 法

本章では，Responsive Link を用いた分散システムにおいてリアルタイム性を保証しつつデッドロックフリーなルーティング手法を提案する．

4.1 ネットワークモデル

本研究では，Resposive Link によって point-to-point で結合され，通信パケットが複数ノードを経由するネットワークを構成する分散システムを想定する．分散システム内のノードの集合を $node$ と定義する．今回想定する分散システムは，N 個のノードで構成され，i 個目のノードを $node_i$ と表現する．各通信リンクが単位時間に転送可能なデータ量を a と定義する．システム内のデータは周期的に転送される．システム内において，各周期で転送されるデータ群をメッセージセット m と定義し，l 個目のメッセージを m_l と定義する．m_l が各周期で転送するデータ量の最大を c_l と表す．メッセージセット m は M 個のメッセージで構成される．m_l は周期と相対デッドラインを持ち，それぞれ T_l，D_l と表す．また，$D_l \leq T_l$ とする．m_l の転送ホップ数は h_l，最短経路を取る場合の転送ホップ数は $h_{l,min}$ で表す．m_l が経由するノードをパス P_l と定義し，P_l 内の各ノードで同一の固定優先度 $priority_l$ を持つ．各メッセージ m_l の優先度 $priority_l$ は $D_l - (c_i \times h_{l,min})$ の値が小さいほど高優先度であるとする．また，システム内のメッセージはプリエンプティブとする．よって低優先度からの影響は考慮しない．

4.2 通信リンクの選択

Responsive Link は，イベントリンクとデータリンクの 2 つの独立した通信リンクが存在し，通信パケット長はそれぞれ 16bytes，64bytes と異なる．各メッセージ m_i が用いる通信リンクは，各周期での最大転送データ量 c_i および最短経路をとる場合の転送ホップ数 $h_{l,min}$ によって決定される．c_i が小さいメッセージ m_i が用いる通信リンクをイベントリンク，c_i が大きいメッセージ m_i をデータリンクで通信することで，c_i が小さいメッセージ m_i は細粒度の通信，c_i が大きいメッセージ m_i はオーバーヘッドが小さい通信が可能である．イベントリンクとデータリンクの使用率を等しくするため，以下の式 (1) が成り立つメッセージ m_i はイベントリンクでパケット転送を行い，式 (1) が成り立たないメッセージ m_i はデータリンクでパケット転送を行う．

$$\sum_{c_j \leq c_i} 2 \times \frac{h_{j,min} \times c_j}{D_i} \leq \sum_{m_j \in m} \frac{h_{j,min} \times c_j}{D_i} \tag{1}$$

式 (1) を用いて通信リンクを選択することで，各メッセージ m_i の転送データ量 c_i とデータリンクとイベントリンクの使用率を考慮した通信リンクの選択が可能である．

4.3 経路の選択

提案手法では，イベントリンクとデータリンクそれぞれで Up/Down ルーティングにおけるルートノードを設定する．ルートノードは高優先度のメッセージが経由するノード数を小さくするように設定する．

ルーティングアルゴリズムをアルゴリズム 1 に示す．アルゴリズム 1 では，ルートノード

となりうるノードの集合を $roots$ とする．アルゴリズム 1 によって $roots$ より 1 つのルートノードを選択する．$roots$ の初期値としてシステム内の全ノードを含み，高優先度のメッセージに対して最短経路を選択できないルートノードを $roots$ から削除することで，ルートノードとなるノードを 1 つ選択する．ルートノードとなりうるノードが複数存在する場合は，ノードの ID によりルートノードを選択する．ルートノードを選択した後，システム内の各メッセージに対して Up/Down ルーティングを用いてルーティングを行う．

図 3 に，ノード数 8 の分散システムにおける，ノード A からノード F へ転送する高優先度パケットとノード G からノード B へ転送する低優先度パケットのルーティング例を示す．図 3 の例では，高優先度パケットが少ないホップ数で転送可能となるルートノードを設定しているため，低優先度パケットの転送ホップ数が増加している．時間制約が厳しい通信パケットが少ないホップ数での通信が可能であるため，受け入れ可能なデータ量が増加する．

Algorithm 1 Select *root*

```
sort M based on priority;
roots = node;
for i = 0; i < M; i++ do
  remove the node can't achieve the shortest path for m_i from roots;
  if Number of nodes in roots is one then
    root = the node in roots;
    break;
  end if
end for
if Number of nodes in roots > 1 then
  root = Select the node which ID is smallest in roots;
end if
```

4.4 スケジュール可能性判定

各メッセージに対して，経路内の各ノードにおける高優先度メッセージからの遅延を定義する．各メッセージは各ノードにおいて，入力ポート，又は出力ポートが同一の高優先度メッセージに遅延される．以下にメッセージ m_i のノード $node_j$ における高優先度メッセージからの遅延の最大 $d_{i,j}$ を定義する．ノード $node_j$ を経由する高優先度メッセージの集合を $M_{i,j,hp}$ とする．

$$d_{i,j} = \sum_{m_k \in M_{i,j,hp}} \lceil \frac{T_i}{T_k} \rceil c_k \tag{2}$$

m_i と高優先度メッセージが同時に $node_j$ へ到着する場合の遅延が最大となる．よって，メッセージ m_i の最悪到着時間 W_i はメッセージ m_i の転送時間と，高優先度メッセージからの遅延の合計となり，式 (3) で定義される．

$$W_i = \sum_{node_j \in P_i} (c_{i,j} + d_{i,j}) \tag{3}$$

システム内の全メッセージに対して最悪到着時間 W_i を計算する．メッセージ m_i に対して常に $W_i \leq D_i$ が成り立つ場合スケジュール可能である．

4.5 多重化したネットワークへの適用

分散システム内の通信リンクを多重化 (マルチリンク) することで，データリンクとイベントリンクそれぞれで複数のルートノードを選択可能である．図 4 では，通信リンクを 2 重にした分散システムにおいてノード D とノード E をルートノードとした場合の例を示す．

図 4 で 2 重にした通信リンクのうち，1 つの通信リンクをルートノード E とした Up/Down ルーティング，1 つの通信リンクをルートノード D とした Up/Down ルーティングに使用することで，デッドロックフリーな経路を複数本形成可能である．データリンクまたはイベントリンクで通信するメッセージは，多重化されたリンクから 1 つを選択する．式 (4) によって各通信リンクに割当てられたメッセージの使用率 U を示し，多重化された通信リンクから U が一番低い通信リンクを選択する．

$$U = \sum_{priority_j < priority_i} \frac{h_j \times c_j}{D_j} \tag{4}$$

式 (4) を用いて多重化された通信リンクから 1 つを選択することで，通信リンク間の使用率を考慮しつつ高優先度メッセージの転送ホップ数を削減することができる．

図 4 に，ノード数 8 の分散システムにおける，ノード A からノード F へ転送する高優先度パケットとノード G からノード B へ転送する低優先度パケットのルーティング例を示す．図 4 の例では，高優先度パケットはノード E をルートノードとする経路が設定され，低優先度パケットはノード D をルートノードとする経路が設定される．図 3 で示す多重化されていない通信リンクで構成される分散システムでは低優先度パケットが複数ホップの転送が

Vol.2012-SLDM-155 No.24
Vol.2012-EMB-24 No.24
2012/3/3

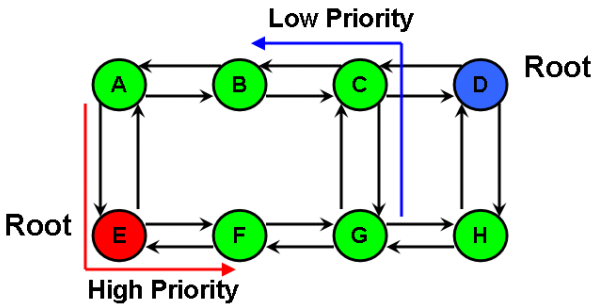


図 4 マルチリンクにおけるルーティング例
Fig. 4 Routing example on multilink

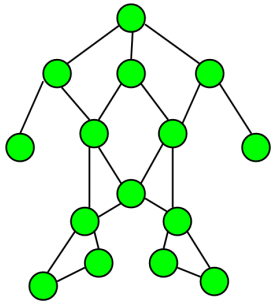


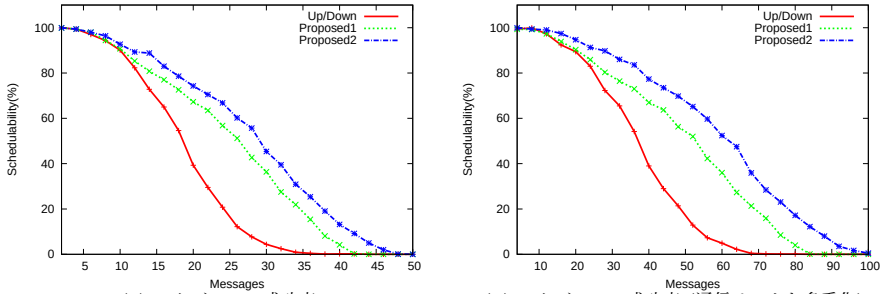
図 5 シミュレーションで想定するトポロジ
Fig. 5 Topology used in simulation

必要であるが，図 4 では 2 つの通信パケットが最短ホップ数で通信可能であるため，受入れ可能な転送データ量が増加する．

5. 評　　価

ネットワークのメッセージ数を変化させ，従来のルーティング手法を適用した場合と提案手法を用いてルーティングした場合で，受け入れ可能なタスク数がどのように変化するのかを，シミュレーションによって比較する．このとき，通信リンクを多重化した場合と，しない場合の 2 種類のネットワークで評価を行なった．

評価で想定する分散システムのネットワークトポロジを図 5 に示す．図 5 は，ヒューマノイドロボットのトポロジを想定しており，複数環状な部分が存在する．また，通信リンクを多重化したネットワークに対する評価では，図 5 の各通信リンクは 2 重であるとする．通信リンクが転送可能なデータ量 $a = 100$ とする．メッセージ m_i の周期 $T_i = \{5000, 5500, ..., 10000\}$，各周期で転送するデータ量の最大 $c_i = \{50000, 55000, ..., 100000\}$，送信元ノード，送信先ノードはランダムに選択する．メッセージの集合をメッセージセットと定義する．通信リンクが多重化されていないネットワークに対してはメッセージセット内のメッセージ数 $\{2, 4, ..., 50\}$，通信リンクが 2 重となるネットワークに対してはメッセージセット内のメッセージ数 $\{4, 8, ..., 100\}$ で測定する．指標であるスケジュール成功率を以下は，評価で実行する全メッセージセットのうち，受け入れ可能であるメッセージセットの割合と定義する．各メッセージ数に対してメッセージセットを 1000 個ずつ用意して用いてスケジュール成功率を測定する．



(a) スケジュール成功率　(b) スケジュール成功率 (通信リンクを多重化)

図 6 シミュレーション結果
Fig. 6 Simulation results

図 6(a) は，通信リンクを多重化しないネットワークにおいてメッセージ数を変化させた場合のスケジュール成功率を示す．図 6(a) の *UpDown* はデータリンク及びイベントリンクのルートノードを ID によって指定した評価，*Proposed*1 は提案手法を適用してイベントリンクとデータリンクで共通のルートノードを選択した場合の評価，*Proposed*2 は提案手法を適用してイベントリンクとデータリンクそれぞれでルートノードを選択した場合の評価を示す．*UpDown* と *Proposed*1 を比較して，時間制約を考慮したルートノードの選択をすることによってスケジュール成功率が向上している．メッセージ数 26 において，*UpDown* のス

ケジュール成功率は 11.1% であるのに対して，*Proposed*1 のスケジュール成功率は 52.3% であり，スケジュール成功率が 41.2% 向上している．優先度が高いメッセージが少ないホップ数で転送可能となるルートノードを選択することでスケジュール成功率が向上したことがわかる．*Proposed*1 と *Proposed*2 を比較して，データリンクとイベントリンクそれぞれでルートノードの選択をすることによってスケジュール成功率が向上している．メッセージ数 28 において，*Proposed*1 のスケジュール成功率は 42.0% であるのに対して，*Proposed*2 のスケジュール成功率は 56.6% であり，スケジュール成功率が 14.6% 向上している．データリンクとイベントリンクそれぞれでルートノードを設定して高優先度メッセージの転送ホップ数が削減できたことで，スケジュール成功率が向上したことがわかる．

図 6(b) は，通信リンクを 2 重にしたネットワークにおいてメッセージ数を変化させた場合のスケジュール成功率を示す．図 6(b) の *UpDown* はシステム内の全通信リンクのルートノードを ID によって指定した評価，*Proposed*1 は提案手法によって全通信リンクで共通のルートノードを選択した場合の評価，*Proposed*2 は提案手法を適用してイベントリンクとデータリンクの各通信リンクそれぞれでルートノードを選択した場合の評価を示す．*UpDown* と *Proposed*1 を比較して，時間制約を考慮したルートノードの選択をすることによってスケジュール成功率が向上している．メッセージ数 56 において，*UpDown* のスケジュール成功率は 7.3% であるのに対して，*Proposed*1 のスケジュール成功率は 41.8% であり，スケジュール成功率が 34.5% 向上している．多重化した通信リンクを持つネットワークにおいても，優先度が高いメッセージが少ないホップ数で転送可能となるルートノードを選択することでスケジュール成功率が向上したことがわかる．*Proposed*1 と *Proposed*2 を比較して，データリンクとイベントリンクの各通信リンクでルートノードの選択をすることによってスケジュール成功率が向上している．メッセージ数 64 において，*Proposed*1 のスケジュール成功率は 28.0% であるのに対して，*Proposed*2 のスケジュール成功率は 47.1% であり，スケジュール成功率が 19.1% 向上している．データリンクとイベントリンクそれぞれで 2 つのルートノードを設定していることで高優先度メッセージの転送ホップ数が削減でき，スケジュール成功率が向上したことがわかる．

6. 結　　論

本研究では，Resposive Link を用いたマルチホップネットワークにおいて，リアルタイム性を考慮しつつデッドロックフリーを保証するルーティング手法を提案した．提案手法は，時間制約が厳しい通信パケットに対してホップ数が少ない通信路を設定しつつ，デッドロックフリーなルーティングが可能である．また，提案手法が Responsive Link を多重化したネットワークに対しても適用可能であることを示した．

シミュレーションによる評価の結果，イベントリンクとデータリンクそれぞれでシステム内のメッセージの優先度を考慮してルートノードを選択することで，受け入れ可能な転送データ量を高めることに成功した．提案手法を用いることで，Up/Down ルーティングと比較してスケジュール成功率を最大約 45% 向上した．また，Responsive Link を多重化したネットワークで適用した場合においても提案手法が有効であることを示した．

今後の課題として，提案手法を Responsive Multithreaded Processor に対して実装を行い評価を取ることが挙げられる．また，非周期的に転送される通信パケットを考慮する必要がある．

謝辞　本研究は科学技術振興機構 CREST の支援によるものであることを記し，謝意を表す．また，本研究の一部は文部科学省グローバル COE プログラム「環境共生・安全システムデザインの先導拠点」に依るものであることを記し，謝意を表す．

参 考 文 献

1) 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．pages 50–63, 2004.
2) Kleinrock L and Kermani P. Virtual Cut-Through : A New Computer Communication Switching Technique. *Computer Networks*, 3, September 1979.
3) MichaelD. Schroeder, AndrewD. Birrell, Michael Burrows, Hal Murray, RogerM. Needham, and ThomasL. Rodeheffer. Autonet: A High-speed, Self-configuring Local Area Network Using Point-to-point Links. *IEEE Journal on Selected Areas in Communications*, 9(8):1318–1335, October 1991.
4) N. Yamasaki. Responsive processor for parallel/distributed real-time control. *IEEE/RSJ International Conference on Intelligent Robots and SystemsE*, pages 1238–1244, 2001.
5) C.L.Liu and J.W. Layland. Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. *Journal of the ACM (JACM)*, pages 20(1):46–61, 1973.
6) IEEE 802.1D. IEEE standard for local and metropolitan area networks–Common specifications–Media access control (MAC) Bridges.
7) N.J. Boden, D.Cohen, R.E. Felderman, A.E. Kulawik, C.L. Seitz, J.Seizovic, and W.Su. Myrinet: A Gigabit Per Second Local Area Network. *IEEE Micro*, pages 29–36, February 1995.