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

2005－ＳＬＤＭ－119（11）
2005／3／17

Responsive Linkを用いたリアルタイム通信ライブラリの設計および実装

佐々木貴宏† 小林 秀典† 山崎 信行†

† 慶應義塾大学大学院 理工学研究科 開放環境科学専攻
〒 223-8522 横浜市港北区日吉 3-14-1
E-mail: †{sasaki,kobahide,yamasaki}@ny.ics.keio.ac.jp

あらまし *Responsive Link* は，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，パケットに優先度を付加し高優先度パケットが低優先度パケットを追い越し，ノードごとに優先度を付け替えることができるなどの特徴を持つ．本研究では，*Responsive Link* の特徴を活かしてリアルタイム通信を実現し，なおかつ，その特殊性を隠蔽する通信ライブラリの設計および実装を行う．本通信ライブラリはどの通信リンクを使用するかをデータサイズによって決定し，パケットの優先度を通信周期によって決定する．本通信ライブラリを用いる事によって，使用する通信リンクの種類およびパケットの優先度をユーザに陽に指定させること無く，リアルタイム通信を実現する．

キーワード リアルタイム通信

Design and Implementation of Real-Time Communication Library Over Responsive Link

Takahiro SASAKI†, Hidenori KOBAYASHI†, and Nobuyuki YAMASAKI†

† Keio University
3-14-1 Hiyoshi, Kouhoku-ku, Yokohama, Kanagawa 223-8522 Japan
E-mail: †{sasaki,kobahide,yamasaki}@ny.ics.keio.ac.jp

Abstract *Responsive Link* has many unique features including separation of soft real-time communication (data link) and hard real-time communication (event link), priority based packet overtaking and priority replacement on each node. In this paper, we design and implement the real-time communication libraries which exploit features of *Responsive Link* and hide its speciality. Communication link to be used is decided by data size and priority of packet is decided by communication period. User can realize real-time communication without specifying communication link type to be used and priority of packet by usind these communication libraries.

Key words Real-Time Communication

1. はじめに

近年分散環境においてリアルタイムシステムを実現する要求が高まってきている．分散リアルタイムシステムでは，単一ノード内処理だけでなく，ノード間の通信においてもリアルタイム性を保証することが必要となる．

既存の通信インタフェースである Ethernet は，安価であり広く利用されている．しかし，Ethernet を用いてリアルタイム通信を実現することは困難である．Ethernet は CSMA/CD 方式を採用しており，データを送信したいノードはケーブルの通信状況を監視し，ケーブルが空くと送信を開始する．この際，複数のノードが同時に送信を開始するとデータが衝突するので，そのような場合には送信を中止し，ランダムな時間待って再送する．したがって，通信を行う際に全く衝突が起きなければ問題ないが，衝突が起こった際には，再送が行われるためにレイテンシの予測が困難となる．結果，リアルタイム性を必要とする通信が許容される時間内で通信される事を保証することができない．

一方，*Responsive Link* [1] [2] は分散環境においてリアルタイムシステムを実現するためのリアルタイム通信規格であり，通信のリアルタイム性を保証するための様々な特徴を持つ．*Responsive Link* は，画像や音声などのソフトリアルタイム通信用のデータリンクとハードリアルタイム用のイベントリンクを分離している．データリンクではパケットサイズを固定長で

大きめにしてバンド幅を保証するソフトリアルタイムを実現し，イベントリンクではパケットサイズを固定長で小さめにしてイベントのレイテンシを保証しハードリアルタイムを実現する．また，パケットに優先度を付加し，高優先度のパケットが低優先度のパケットを追い越すことができる．これにより，通信時間を予測することが可能となり，通信の時間制約を満たすことができる．さらに，各通信ノードでパケットの優先度のつけ替えが可能であり，パケットの加減速制御を行うことが可能である．

本研究では，*Responsive Link* を用いてリアルタイム通信を実現する通信ライブラリの設計および実装を行う．*Responsive Link* を用いる上で，パケットの優先度の決定および通信リンクの決定が問題となる．本通信ライブラリはデータリンクとイベントリンクのどちらの通信リンクを使用するかをデータサイズによって決定し，パケットの優先度を通信周期によって決定する．本通信ライブラリを用いる事によって，ユーザが使用する通信リンクの種類およびパケットの優先度を陽に指定すること無くリアルタイム通信を実現する．

2. *Responsive Link*

この章では，本機構の実装対象である *Responsive Link* について説明する．

2.1 ハードリアルタイム通信とソフトリアルタイム通信の分離

一般に，ソフトリアルタイム通信は，音声データや画像データなどのマルチメディアデータの通信などに主に用いられる．一方，ハードリアルタイム通信は，制御データやイベントの通知に主に用いられる．これらの通信には，以下のようなトレードオフがある．

- ソフトリアルタイム：スループットを上げたい．
- ハードリアルタイム：レイテンシを小さくしたい．

これらの要求を同時に満たす事は難しい．スループットを上げるためには，パケットサイズを大きくする必要があり，レイテンシを小さくするにはパケットサイズを小さくする必要がある．また，イベント通信をデータ通信と混在させるのは問題があり，これらを分離して通信する必要がある．

そこで *Responsive Link* では，ハードリアルタイム通信用の通信ラインとソフトリアルタイム通信用の通信ラインを分離している．以降，それぞれをデータリンク，イベントリンクと呼ぶ．

データリンクでは，パケットサイズを 64 バイト (ペイロードサイズ 56 バイト) と固定長で大きめにしてソフトリアルタイム通信に用い，イベントリンクでは，パケットサイズを 16 バイト (ペイロードサイズ 8 バイト) と固定長で小さめにしてハードリアルタイム通信に用いる．

2.2 優先度による追い越し機能

入力ポートから入力された通信パケットは，通信ノードで衝突しない場合，そのまま出力ポートへ出力される．異なる入力ポートから入力された通信パケットが同じ出力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ためて出力を待たせ，高い優先度の通信パケットを先に出力させる．高い優先度の通信パケットの出力の後に低い優先度の通信パケットを追い越し用バッファから出力ポートに出力し，優先度に従った通信パケットの追い越しを行う．

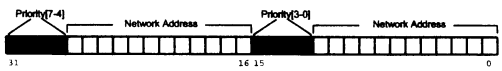

図 1　パケットのヘッダフォーマット

2.3 ルーティング

Responsive Link の経路制御は，図 2 に示すようなルーティングテーブルを設定することによって行われる．図 2 において，Reference 部はパケットのヘッダ部 (図 1) と同一であり，Referent 部に当該パケットに関する設定を行う．EE ビットおよび DE ビットは，それぞれそのラインがイベントリンク用の設定かデータリンク用の設定かを示す．L[0-4] は出力ポートを指し示す．

パケットのヘッダ部とルーティングテーブルの Referent 部の比較を行い，パケットのヘッダ部とネットワークアドレスおよび優先度が一致するルーティングテーブルの L[0-4] のビットが立っている出力ポートに送出される．その際，ルーティングテーブルの PE ビットが立っている場合，パケットの優先度が P[7-0] に付け替えられる．

3. 設計方針

本章では，*Responsive Link* を用いる上で重要となる優先度の決定方法および通信リンクの決定方法について述べる．

3.1 優先度決定

優先度の決定には以下の 2 つのパラメータを用いる事が考えられる．

- 許容できるレイテンシ
- 通信が行われる頻度 (通信周期)

まず，レイテンシを基に各ホップでの優先度を割り当てる場合，各ホップで図 3 のように仮想的なデッドラインを定める．複数ノードを経由する通信では，各ノードで優先度制御されるためそれぞれのホップでのレイテンシが異なる．そのため各ノードで異なるデッドラインを基に優先度を決定した方がより細やかな優先度制御を行うことができる．しかしながら，各ホップでの仮想デッドラインをどのように決定するか最適な方法を導き出すのは難しい．一つの方法として，各ホップでの通信状況を基に仮想デッドラインを割り振るという方法が考えられる．例えば，より通信が頻繁に行われているホップでは，より大きな仮想デッドラインを割り当て，あまり通信が行われていないホップでは，仮想デッドラインを小さく割り当てる．

許容できるレイテンシを基に優先度を割り当てた場合の長所として，各ホップでより細やかな優先度の設定ができ，スケジューラビリティが高まるということである．

一方欠点として，各ホップでの通信状況は刻々と変化するために，最適に設定されていた仮想デッドラインが最適ではなく

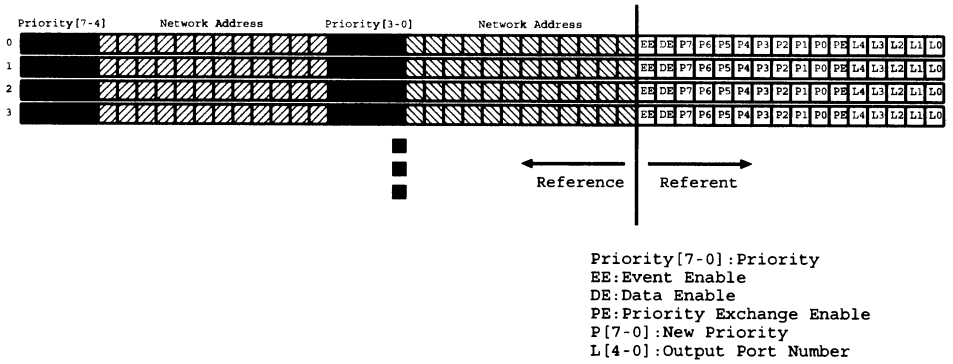

図 2 *Responsive Link* のルーティングテーブル

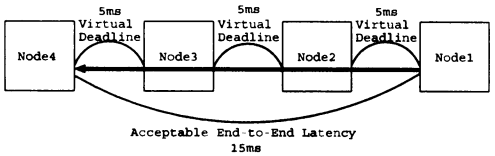

図 3 仮想デッドラインの割り当て

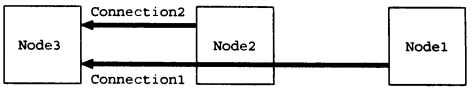

図 4 ネットワーク構成例

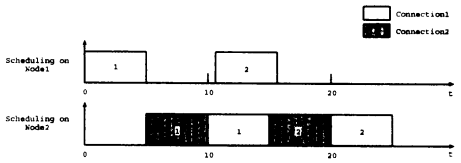

図 5 周期を用いて優先度を決定した場合の各ノードでのパケットスケジューリング例

なってしまい，各ホップでの仮想デッドラインを再計算する必要がある．例えば，仮想デッドラインを設定した時点ではホットスポットであったホップが，あるコネクションが開放されてホットスポットでは無くなった場合に，設定されていた各ホップでの仮想デッドラインは最適ではなくなり再計算が必要となる．仮想デッドラインの決定ポリシにもよるが，頻繁に通信状況が変化するネットワークでは，各ホップでの仮想デッドラインの再計算によるオーバヘッドが無視できない．また，仮想デッドラインを割り当てるためには各通信ノードでの通信状況を把握しておく必要があるので，このためにノード間でメッセージのやり取りをしておく必要がある．

一方，周期を基に優先度を割り当てる場合，周期が短い，つまり，頻繁に通信が行われるコネクションの優先度を高く設定する．

この方法の長所として，前者に比べ単純であり優先度を決定する際の計算コストを抑えることができる．優先度の決定は各コネクションの周期によりソートを行うだけである．また，あるホップでの通信状況が変化した場合に，レイテンシを用いて割り当てた場合のように全ホップに影響を与えるのではなく，そのホップに影響が出るだけであるのでオーバヘッドは少なく，システムを単純化できる．

一方欠点として，前者に比べ，スケジューラビリティが低いということが挙げられる．例えば，図 4 のように 2 つのコネクションが確立されており，それぞれ周期が 10ms，11ms，許容できるレイテンシが 10ms，11ms，各ホップでのレイテンシ (他に何も通信が行われていない状況で) がそれぞれ 5ms であるとする．コネクション 1 とコネクション 2 はノード 2 で同一経路にパケットが送出されるため衝突が生じる．

図 5 は，通信周期を基に優先度を決定した場合の最悪の状況での各コネクションのデータ送信スケジュールを示している．コネクション 1 の周期が 11ms であり，コネクション 2 の周期が 10ms であるので，ノード 2 での優先度はコネクション 2 の方が高くなる．時間 0 でコネクション 1 がデータ送信を開始し，時間 5 でコネクション 2 がデータ送信を開始する．時間 5 で各コネクションのパケットが同一経路にパケットを送信しようとするため衝突が起きる．ここで，コネクション 2 の優先度の方が高いためにコネクション 2 のパケットの送信が行われる．そのためコネクション 1 のパケットは待たされて，時間 10 から送信が開始される．コネクション 1 のデータ送信は時間 15 で終了するが，これは許容されるレイテンシである 11ms を越えてしまっており，時間制約を満たせない．

図 6 は，レイテンシを基に優先度を決定した場合の最悪の状況での各コネクションのデータ送信スケジュールを示している．なお，コネクション 1 の各ホップでの相対デッドラインをそれぞれ 5.5ms と設定し，ノード 2 での優先度はコネクション 1 の方が高くなる．この例でも時間 0 でコネクション 1 がデータ送信を開始し，時間 5 でコネクション 2 がデータ送信を開始する．時間 5 で各コネクションのパケットが同一経路にパケットを送信しようとするため衝突が起きる．ここで，コネクション 1 の方が優先度が高いためにコネクション 1 のパケットの送信

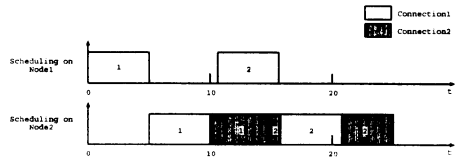

図 6　通信遅延を用いて優先度を決定した場合の各ノードでのパケットスケジューリング例

が開始される．コネクション 1 のパケット送信後，コネクション 2 のパケット送信が開始され，時間 15 でコネクション 2 のパケット送信は完了する．この場合，両コネクションとも時間制約を満たすことができる．

以上のように，これら二つのアプローチにはトレードオフがある．通信遅延を基に優先度を決定する方法は，スケジューラビリティを高くすることができるが，仮想デッドラインの最適な割り当てが難しく，また，仮想デッドラインの計算が必要であり，さらに，ノード間で通信状況についてのメッセージのやり取りが必要になり，システムが複雑になってしまい実用的ではない．したがって，本研究では，通信周期を基に各通信ノードでの優先度を決定する．

3.2　通信リンクの決定

イベントリンクはパケットサイズが小さいため，より低遅延で通信を行うことができる．しかしながら，イベントリンクでヘッダとトレイラがパケット中に占める割合は，16 バイト中 8 バイトでありデータリンクが 64 バイト中 8 バイトであるのに比べオーバヘッドが大きい．したがって，大きなデータをイベントリンクで送る場合，ヘッダとトレイラのオーバヘッドにより，データリンクで送信した場合より通信遅延は大きくなる．

通信帯域を B[bytes/s]，データサイズを S[bytes] とした時にイベントリンクを用いた場合のレイテンシ L_{event}[sec] は式 (1) となる．また，データリンクを用いた場合のレイテンシ L_{data}[sec] は式 (2) となる．

$$L_{event} = \frac{\lceil \frac{S}{8} \rceil \times 16}{B} \quad (1)$$

$$L_{data} = \frac{\lceil \frac{S}{56} \rceil \times 64}{B} \quad (2)$$

ここで，式 (1) および式 (2) より，データサイズが 32 バイトより大きい場合にはデータリンクを用いた場合の方が低レイテンシとなる．

また，イベントリンクを用いてデータを送信した場合のレイテンシとデータリンクを用いてデータを送信した場合のレイテンシを実測した結果を図 7 に示す．なおこのレイテンシはソフトウェアのオーバヘッドを含み，通信バンド幅が 16Mbps，CPU の動作速度は 125MHz で測定した．図 7 より，理論的に得た結果と同様に 32 バイトより大きいデータの場合は，イベントリンクを用いた場合よりデータリンクを用いた場合の方が低レイテンシとなることが確認できる．

したがって，本研究では 32 バイト以下の場合はイベントリンクを用い，32 バイトより大きいデータの場合はデータリンクを用いることにする．

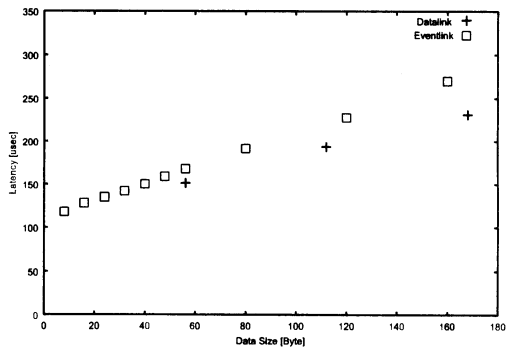

図 7　各通信リンクのレイテンシ

4.　リアルタイム通信ライブラリ

この章では，提供する通信ライブラリについて述べる．*Responsive Link* を用いて通信を行うにあたり，パケットの優先度の設定，使用する通信リンクの種類およびルーティングテーブルの設定が必要となる．本ライブラリは，これらの処理を隠蔽する．また，ユーザが使用しやすい様に一般に利用されているソケットライブラリに似た構造とする．

各通信ライブラリの機能と概要を以下に示す．

- rl_connect():受信側から送信側に接続要求を出す
- rl_accept():受信側からの接続要求を待つ
- rl_send():データ送信
- rl_recv():データ受信

4.1　コネクション確立

コネクション確立に使用するライブラリとして rl_connect() および rl_accept() を提供する．rl_connect() は受信側が送信側に対し接続要求を出すライブラリであり，rl_accept() は送信側が受信側から要求される接続要求を待つライブラリである．

```
rtn = rl_connect(size_addr src_addr,
        size_port port_number,
        size_period period, size_t size);
rtn:
        確立したコネクションの ID
        -1:コネクション確立エラー
```

```
rtn = rl_accept(size_port port_number);
rtn:
        確立したコネクションの ID
        -1:コネクション確立エラー
```

コネクション確立には *responsive Link* を用いて通信を行う際に必要となる，ルーティングテーブルの設定が必要となる．送信ノードと受信ノード間の各通信ノードで周期によりソートを行い，コネクションの優先度を決定し，ルーティングテー

ブルの登録を行う．決定された優先度は優先度のつけ替えのため一つ手前のノードに必要となるので，決定された優先度情報を含め次のノードにメッセージを送信する．周期によりソートを行った際，すでに確立されているコネクションの優先度が変更する可能性があり，その場合一つ手前のノードにおいてルーティングテーブルの優先度つけ替えビットの変更を行う必要がある．そのためにそのノードにルーティングテーブルの変更を要求するメッセージを送信する．

またコネクションを確立する際，ユーザに指定されたサイズによって使用する通信リンクの種類を決定する．

これらの処理により，ユーザは *Responsive Link* の特殊性を意識すること無く，コネクションを確立することが可能となる．

なお，ルーティングは送信ノードと受信ノードの最短経路で行われる．

4.2 データ送受信

データの送受信を行う際に使用するライブラリとして rl_send() および rl_recv() を提供する．

```
rtn = rl_send(size_id connection_id,
        const void *buf, size_t size);
rtn:
        送信したデータサイズ (バイト)
        -1:送信エラー
```

```
rtn = rl_recv(size_id connection_id,
        const void *buf, size_t size);
rtn:
        受信したデータサイズ (バイト)
        -1:受信エラー
```

rl_send() は，buf から size バイトのデータを rl_accept() によって確立されたコネクションを用いて送信する．この際，コネクション確立時に決定された通信リンクを用いてデータは送信される．

rl_recv() は，rl_connect() によって確立されたコネクションを用いて送信側から送信されたデータを size バイト分 buf に格納する．

5. 評　　価

本研究では，*Responsive Link* を有した *Responsive Processor* [3] を実装対象とし，その上で動作する RT-OS である RT-Frontier [4] に本通信通信ライブラリを実装し，リアルタイム通信を実現できるか評価を行った．評価対象として，既存の通信インターフェースで最もよく用いられる Ethernet を用いた．*Responsive Link*，Ethernet の通信帯域をそれぞれ 16Mbps，10Mbps として実験を行った．また，Ethernet の上位プロトコルとして TCP/IP を用いた．

Motion-JPEG によりノード間で JPEG データの通信を行い，その JPEG データ通信のリアルタイム性が保証できているか評価した．動画配信では，各 JPEG フレームの通信にリアルタイム性が求められる．

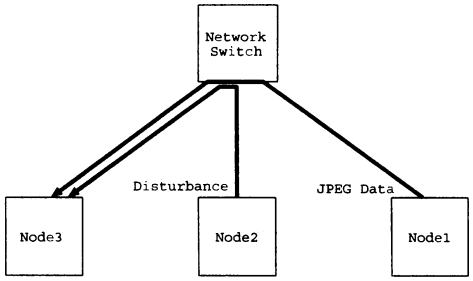

図 8　Ethernet の実験環境

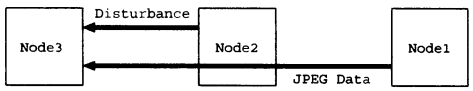

図 9　Responsive Link の実験環境

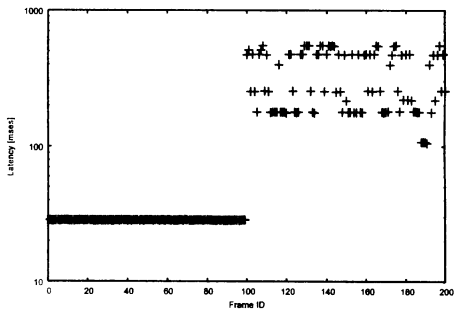

図 10　Ethernet を用いた場合の実験結果

通信している最中に邪魔となるパケットの送信を開始し，そのパケットが JPEG データパケットに与える影響を *Responsive Link*，Ethernet の両者で比較を行った．それぞれノード 1 からノード 3 に平均して約 32 キロバイトの JPEG データを送信し，ノード 2 からノード 3 にベストエフォートで送れるだけの邪魔パケットを送信する (図 9，図 8)．

Ethernet を用いた場合の実験結果を図 10 に示す．図 10 で，横軸はフレーム ID，縦軸は JPEG データ通信のレイテンシを示す．このレイテンシはソフトウェアでの送受信処理を含む．フレーム ID が 100 となる時点から，邪魔となるパケットの送信を開始した．図 10 より邪魔パケットの送信が開始された途端に JPEG データパケットのレイテンシが増大している．これは，邪魔パケットと JPEG データパケットが衝突し，再送が行われているためだと考えられる．この結果より，Ethernet を用いる場合，レイテンシの予測が困難となりリアルタイム通信を実現するのは難しいということがわかる．

Responsive Link の実験では，JPEG データパケットの優先度を邪魔パケットより優先度が高くなるように設定した場合と，比較のため JPEG データパケットの優先度が邪魔パケットより低くなるように設定した場合の 2 パターンで実験を行った．そ

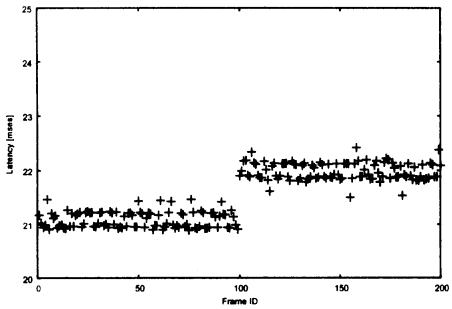

図 11 *Responsive Link* を用いた場合の実験結果 (高優先度)

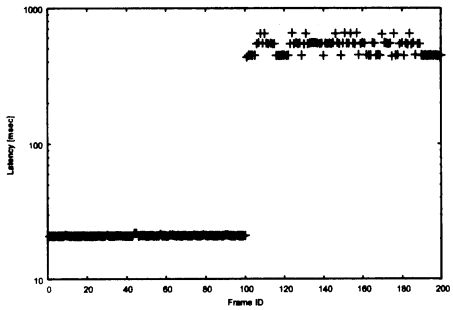

図 12 *Responsive Link* を用いた場合の実験結果 (低優先度)

れぞれの実験結果をそれぞれ図 11，図 12 に示す．

図 11，図 12 も図 10 と同様に横軸はフレーム ID，縦軸は JPEG データ通信のレイテンシを示す．フレーム ID が 100 となる時点から，邪魔となるパケットの送信を開始した．

図 11 より，邪魔パケットの送出が開始されても JPEG データパケットはほとんど影響を受けていない事がわかる．邪魔パケットの送信開始後平均して 1ms ほどレイテンシは増大しているが，これは，ソフトウェアでの処理において影響を受けてしまった結果だと考えられる．

図 12 より，JPEG データパケットが低優先度の場合，邪魔パケットのが高優先度となるために JPEG データパケットは追い越されてしまい，レイテンシが増大する．しかしながら，本来，邪魔パケットのような大量のデータ通信はリアルタイム性が求められない非周期通信であり，動画通信のような周期通信より優先度は低く設定される．今回の実験では，意図的に邪魔パケットの優先度が高くなるように設定している．

6. ま と め

本研究では，*Responsive Link* を用いたリアルタイム通信ライブラリを設計および実装した．本ライブラリは，*Responsive Link* を用いる上で重要となる優先度および通信リンクをユーザに指定させる事なく，ライブラリ内部で適切に設定する．実験結果より，*Responsive Link* を用いる事によってリアルタイム性が求められるデータ通信の時間制約を保証することがわかる．

今後の課題として，優先度の決定ポリシの改善が必要である．周期を基に優先度を決定した場合，スケジューラビリティが低いという問題点があり最適ではない．また通信を行う上でルーティングは非常に重要な要素である．現状では最短経路でコネクションを確立している．他の通信が行われていない場合には，最短経路でのコネクション確立はベストであるが，最短経路で通信が頻繁に行われている場合には，レイテンシが大きくなる可能性がある．したがって，通信状況により通信経路を決定する事が望ましい．さらに，本研究ではアドミッションコントロールを行っていないために，低優先度のパケットは高優先度のパケットに追い越され，通信の時間制約を満たせない可能性がある．このような状況に対処するために，遅延管理機構を実装する必要がある．

謝辞 本論文の研究は，文部科学省の科学技術振興調整費の支援による．また本研究の一部は，科学技術振興機構 CREST の支援による．

文 献

[1] "Responsive Link(IPSJ-TS 0006:2003)", http://www.itscj.ipsj.or.jp/ipsj-ts/02-06/toc.htm.

[2] 山崎 信行："分散制御用リアルタイム通信 Responsive Link の設計および実装"，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), pp. 50–63 (2004).

[3] N. Yamasaki: "Design and Implementation of Responsive Processor for Parallel/Distributed Control and Its Development Environments", Journal of Robotics and Mechatronics, **13**, 2, pp. 125–133 (2001).

[4] H. Kobayashi and N. Yamasaki: "An Integrated Approach Implementing Imprecise Computations", IEICE Transactions on Information and Systems, **E86-D**, pp. 2040–2048 (2003).