計算機アーキテクチャ 144－26
（２００１．７．２７）

レスポンシブリンクの通信遅延管理を実現する動的経路制御機構

加藤 孝一　小林 秀典　山崎 信行　安西 祐一郎

慶應義塾大学大学院 理工学研究科 開放環境科学専攻

E-mail: {katokoh, kobahide, yamasaki, anzai}@ayu.ics.keio.ac.jp

Responsive Link は、データリンクとイベントリンクの２系統を持ち、ハードウエア的に実時間通信をサポートするネットワークである。この特徴を利用することで既存のネットワークの問題点である制御情報の時間制約を満たすことが可能である。本機構では、データと共に一定間隔で、遅延検出用のコントロールパケットを送出する。各ノードにおいてその受信間隔を計測し、ネットワーク上で許容値を超える遅延を生じている区間 (*Critically Congested Section*) の検出、さらに代替経路探索を行なう。この機構を本研究室で開発された *μ-PULSER* 上で設計・実装する。

実時間通信、 Quality-of-Service, ルーティング、*Responsive Processor*, *Responsive Link*

Delay Sensitive Adaptive Routing for *Responsive Link*

Kohichi Katoh, Hidenori Kobayashi, Nobuyuki Yamasaki, Yuichiro Anzai

School of Science for Open and Environmental Systems
Graduate School of Science and Technology, Keio University

In this paper, we propose adaptive routing for guaranteeing Quality-of-Service in distributed envirment. Network has many problems in present condition, so we solved them using *Responsive Link* . *Responsive Link* supports real-time communications by hardware. In our system, control packets send at regular interval for detecting communication delay. Every node measures arrival time interval of control packet.If a certain host detects packet arriving over permissible delay for data, there exists *Critically Congested Sections(CCS)*. In this case searching alternative routes of CCS and changing route.

Real-Time Communication, Quality-of-Service, routing, *Responsive Processor*, *Responsive Link*

1. はじめに

ネットワーク通信においては、ユーザもしくはデータが要求する Quality-of-Service(QoS) を保証する事が重要である。QoS を保証するためには、帯域保証・資源予約・通信遅延管理・ストリーム同期・バッファ制御等を行うことが一般的である。特に、ロボット等の組み込みシステムにおいては実時間性を持つデータ通信が度々発生する。さらに、定常的な通信に加えて、環境の変化に伴ってそれまでとは異なる通信が発生することもある。これらの通信においてデータの要求する時間制約が満たされない場合には、システムに重大な影響を与える可能性がある。その際、ネットワークの負荷状態は一定ではないため、通信遅延を考慮し、その変化に対して各ノードにおいて動的に対応する必要がある。しかし、既存のネットワーク通信媒体を用いた場合には、基本的に best-effort 型の通信であるため、ネットワークが混雑した場合に QoS を保証できないという問題がある。したがって、通信遅延の管理を行なうことが非常に重要である。

本研究では、通信の実時間性の保証を行うため、分散環境において各ノードでネットワークの通信遅延を検出し、動的にルーティングを行なう機構を設計、実装する。その際、ハードウエア側から実時間通信をサポートする *Responsive Link* をノード間の通信に用いる。

2. 背　景

2.1 ルーティング

実時間性を要求する通信において、ネットワークの状態を考慮したルーティングは有効な手段である。一般に、ルーティングは非適応型と適応型に分類されネットワーク上で発生する通信の要求する QoS を保証するもしくは、ネットワーク資源の有効利用のために行

われる。ネットワークの負荷状況は一定でないということを考えると、適応型ルーティングを行なうことが必要であると考えられる。

また、分散管理型ルーティングでは、*NP*-完全問題・遅延制約・コスト制約を分散的に解決し、マルチパスルーティングを行う[1)2)]。分散管理を行なうことにより、ルーティングの応答時間の短縮、システムのスケーラビリティを増加することが可能である。つまり、実時間性を保証するため、ルーティングを行なう際に必要となる遅延時間の管理に対して有効である。

ルーティングを行なう際にはまず、経路上のどこで通信の障害となるような遅延が発生しているのかということ検知しなければならない。更に、その遅延に対応可能なルーティングを行なうことが必要である。

2.2 *Responsive Processor*

Responsive Processor[3)] はパーソナルロボット、オフィスオートメーション、マルチメディアサービス等における並列分散制御を目的として開発されたSystem-on-a-chip である[4)]。また、各ノード間の通信の際には次節で述べる *Responsive Link* と呼ばれるリンクを用いる。

2.3 *Responsive Link*

Responsive Link は、ノード間の通信において、ハードウエアの側からレスポンシブ性を実現することを支援するネットワークである。レスポンシブ性とは、リアルタイム性とリアクティブ性を兼ね備えていることをいう。以下に *Responsive Link* の特徴を示す。

- 1つのノードに対して4本のリンクを持つ
- 1ラインに対して、制御情報伝達用リンク (Event Link) とデータ伝達用リンク (Data Link) の2系統をもつ
- 固定パケットサイズ
  - データパケット:64byte, バンド幅保証
  - イベントパケット:16byte, レイテンシ保証
- データとイベントを独立にルーティング可能
- リンクの接続はPoint-to-Pointでありトポロジーフリーな形態でネットワークを構築可能
- パケットに対して4段階の優先度を持ち、ノードにおいて優先度の付け替えやパケットの追い越しが可能
- 1ホップ毎に前方エラー訂正可能

2.4 関連研究

Shinら[2)] は、遅延制約のある要求を満たすことを目的として、floodingによりルーティング要求メッセージを伝送する手法を用いた。パケットが、ディスティネーションに到着するまでにパケットが通ったパス上の遅延情報を記憶しておく。中間ノードでは、初めてメッセージが受信された場合もしくは以前に届いたメッセージに比べて遅延性能がよい場合のみ次のノードに対してfloodingを行う。

一方、Chenら[5)] は、probeパケットを用いて分散環境におけるQoS保証のためのルーティングを行った。しかしこれらの手法では、中間ノードにおいてパケットが到着した以外の全ての経路に対するfloodingを行うため、ネットワーク上のトラフィックが増加し、どの経路を用いても要求されるQoSを満たすことが出来ないという、いわゆるブロック状態を引き起こすことがある。このような欠点を改善するために次のような研究が行われている。

分散型ルーティングの評価基準としては、要求受け入れ率・要求セットアップ時間・ルーティング距離などがある。しかし、これまでのルーティングアルゴリズムはメッセージ遅延とシングルコネクションのルート距離の性能を良くしようというものが一般的であった。したがって、先に挙げた3つの評価を全て改善できるアルゴリズムは存在しなかった。それに対し、Manimaranら[6)] は、Parallel-probingアプローチを用いて3つの評価を全て改善している。この方法は、probeパケットを伝送して同時に複数のパスの探索を行う。途中に指定した中間ノードに最初に到着したprobeパケットが通って来た経路を、そのノードまでの経路として選択し、これをディスティネーション・ノードまで行うことでソース・ノードからのルーティングを行う。さらに、あるチャネルが確立された後は、必要以上に要求した資源の内で過剰なもの解放し資源の浪費防ぐ。しかしながら、この方法の場合も中間ノードの選択やその数の決定をいかに行うかという問題がある。また、中間ノードにおいてprobeパケットを送る先を最終的には、heuristicに決定しなければならない。

3. 動的経路制御機構の提案

本研究では、実時間性を要求する通信の障害となる、データの許容を超える遅延を発生させているネットワーク上の経路 (*Critically Congested Section*(以下、*CCS*)) を特定する機構、さらに、*CCS* を回避するための代替経路探索機構を設計・実装する。

3.1 QoS保証のための動的ルーティング機構

ネットワークのトラフィック状況は常に変化する。しかし、どの様な状況下においても個々のデータが持つ時間制約等のQoS要求は、データの種類に依存する。したがって、各データのQoSを保証するために

は、ネットワークの負荷状況を監視し、その状況に応じた対応を行う必要がある。その1つの手段として動的にルーティングを行うという方法が考えられる。

本研究では、*Responsive Processor* と、本研究室で開発されたリアルタイムOS *μ-PULSER* を用いてその開発を行う。

前述の *Responsive Link* が持つイベントリンクでは、その負荷の大小に対して無関係に、通信遅延をほぼ一定の値に抑えることが出来る[7]。そのため、リアクティブ性を実現することが可能である。そこで、ネットワークにおいて、許容不可能な遅延を検出した場合、優先度の変更や代替経路の探索、ルーティングテーブルの変更等の要求を送出する際にイベントリンクを用いることとする。

ネットワークの負荷状況や現在のシステムの状況に対応しながら、QoSを保証するためにルーティングの動的管理を行う。そのために、許容値以上の遅延を生じさせている経路の検出・ルーティングテーブルの動的管理・代替経路の探索などの機構が必要となる。

3.2 遅延の検出

QoSを保証するためには、各ノードにおいてネットワークの遅延を検出する必要がある。本研究においては、通常のデータと同じ経路上に、ある一定間隔で遅延検出用パケット (以下、コントロールパケット) を送出する。コントロールパケットはデータパケットを用いて送信する。各ノードでは、コントロールパケットの受信間隔を計測することにより、ネットワークの遅延検出を行う。

そこで、各ノードに許容可能な遅延 k という *threshold* を設けて判断の基準とする式 (1)。式 (1) において、$arrival\_time_{node(n)}$ はノード n におけるコントロールパケットの受信間隔である。式 (1) を満たしたノードでは、ソース・ノードに対して、データが通って来た経路上のイベントリンクを通して、遅延に対処するために、データの優先度を上げる要求を送出する。

$$arrival\_time_{node(n)} > k \qquad (1)$$

ここで、ネットワークの負荷の基準となる *threshold* は、アプリケーションに依存するものである。したがって、アプリケーションにより適切な *threshold* を指定する必要がある。

3.3 *Critically Congested Section* の検出

ネットワークに許容値以上の遅延が生じる場合でも、ソースからディスティネーションまでの経路全てにおいて許容値以上の遅延が生じているわけではない。基本的にはソース・ディスティネーション間の経路上の、あるノード間において遅延が生じていると考えることが出来る (図1)。

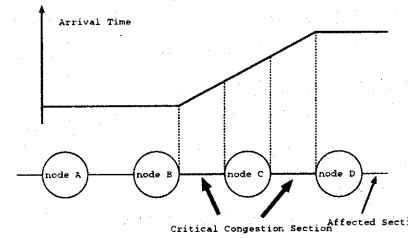

図1 *Critically Congested Section* のモデル

図1の様なノードAからノードDへの経路があった場合に、各ノードにおけるコントロールパケットの到着時刻間隔を検出してみると、図中のグラフのようになったとする。この場合には、ノードCよりも下流の全てのノードでは遅延の増大を検出するであろう。しかし、実際に、その原因となっているのはノードB・ノードD間の経路である。この様な区間を "*Critically Congested Section(CCS)*" とし、ノードD以降の許容値以上の遅延が生じた影響を受ける区間を "*Affected Section*" と定義する。

ここで、*Affected Section* における経路制御は遅延の解決にならない。すなわち、*CCS* によってもたらされる許容値以上の遅延の影響を回避するためには、*CCS* 全体を代替できるような経路を見つけることが望ましい。

本研究では、以下の手順により *CCS* を検出する。

(1) ソースノードから定期的にネットワークトラフィック遅延を検出するためのコントロールパケットをデータリンクを用いて送出する

(2) 各ノードでコントロールパケットの受信間隔を計測し、設定した *threshold* を超える遅延が生じているかどうか調べる

あるノード (例えば、図1ノードC) において許容値以上の遅延を検出した場合には、そのノードと経路上の一つ前のノード (ノードB) との間の経路は *CCS* の可能性があると言える。そこで、許容値以上の遅延を生じているノードの存在検出後には、どのノード間 (複数ノード間の場合も含め) が *CCS* であるかということを検出しなければならない。以下にその方法について述べる。

3.3節で述べたように、遅延を検出したノードはソースノードに対して、遅延を検出したというイベントを送信する。その場合1つ上流のノードの状況は、*threshold* を超える遅延を生じている場合と、*threshold* を

超える遅延は検出していない場合がある。前者の場合には、*CCS* の開始ノードはさらに上流にあり、許容値を超える遅延を検出していない場合には *CCS* の開始ノードであるということが出来る。この方法により *CCS* の開始ノードを検出し、さらに *CCS* の終了ノードについては、ノード間でイベントリンクを用いて各ノードの遅延情報を交換することで検出を行う。

3.4 代替経路の探索

前節で述べた *CCS* を検出した場合に、代替経路の探索を行う。

ここで、代替経路の検出を行う場合には、ある程度の時間が必要となる。そこで *Responsive Link* の持つ優先度を用いる。ソースノードに *CCS* が発見されたというイベントパケットが届いた場合には、代替経路の探索が行われるまでの間、一時的に優先度を上げる(図2)。これは、優先度を上げない場合には、検索中においても、許容値を超えた遅延のもとで通信が行なわれてしまうため、その影響を小さくするためである。

図2は、*CCS* の検出から、ソース・ノードにおいてデータの優先度を上げるまでの流れを示したものである。

(1) ノード C で遅延を検出
(2) ノード C から遅延検出のイベントパケットを送出
(3) ソースノードで送信データ優先度をあげる
(4) *CCS* の開始ノード (ノード B) から代替経路の探索

更に以下では、図2の4で述べた代替経路に関してその探索方法を述べる。代替経路の探索は、*CCS* の開始ノード及び終了ノードから行う。その際に、前者は flooding、後者は selective probing を用いる。ただし本機構においては、元の経路上のノードに対しては、代替経路探索用パケットを送出しないものとする。経路探索は以下のような手順となる。なお図3の数字は以下のものと対応する。

(1) *CCS* の開始ノードから、現在の経路上以外のノードに対して flooding で代替経路探索用パケットを送出
(2) *CCS* の終了ノードから selective probing により代替経路探策用パケット送出
(3) 両方から送出されたパケットが同じノードに到達した場合、それらのパケットが通って来た経路を代替経路として採用、
(4) ルーティングテーブルの変更を行い発見した代替経路を用いた通信を行う

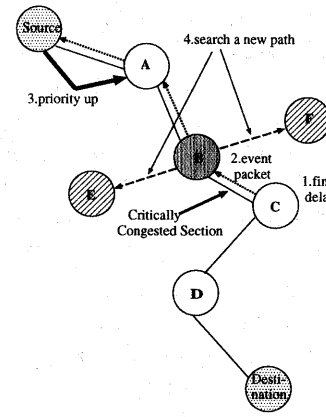

図2 *CCS* 発見後の priority up

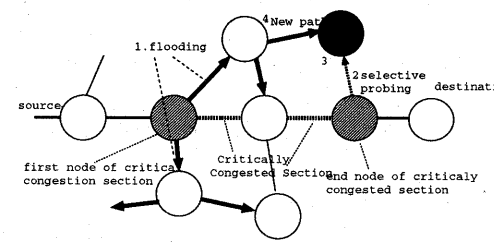

図3 代替経路探索

代替経路探索においては、以下のような問題が生じる可能性がある。

(1) flooding を行なうことで、代替用経路を検出できないパケットは、ネットワーク上で無限ループ状態に陥り、結果的にネットワーク上のトラフィクが増加する可能性がある
(2) flooding パケットと selective probing パケットの両方が、あるノードに到達するより前に元の経路上のノードに戻って来てしまうことがある
(3) 許容値を超える遅延を検出した場合には、代替経路の検出が行われない限り、遅延を検出し続ける可能性がある

これらの問題に対する対応を以下に述べる。

(1) ホップ数に上限を設け、上限を超えたパケットは破棄することによりネットワーク上の無駄なトラフィックを無くす
(2) 基本的には、*CCS* の途中のノードに到着した探索パケットが通って来た経路は採用しない
(3) 一端イベントパケットを送出したノードは、一定時間は次のイベントパケットの送出を行わない

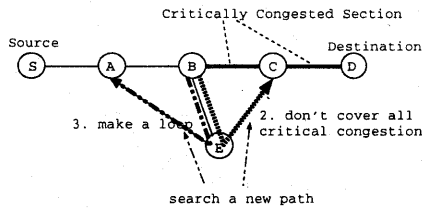

図4　代替経路探索時における問題

3.4.1　隣接ノードとの接続管理

システムにスケーラビリティを持たせるための補助として、各ノードに各リンクの接続先ノード番号、相手ノードとの間のネットワーク遅延情報をテーブルとして持たせる。

新たに他のノードと接続された場合には、自分のノード番号とノード間の遅延のデータを持ったパケットを接続された相手先のノードに送る。

3.4.2　コントロールパケット

データ通信を行う際に3.2節で述べたように、データと共に、ある一定時間間隔でコントロールパケットを送出する。このコントロールパケットは経路上の全てのノードにおいて受信される必要がある。したがって各ノードにおいて、通常のデータとコントロールパケットが区別可能でなければならない。パケットのヘッダにおけるアドレスフィールドは図5の様になっている。そこで、コントロールパケットを通常のデータと区別するため、コントロールパケットの送信側のノードID、及びポートIDは0とする。また、16ビット目・32ビット目はハードウエアにより優先度として用いられる。

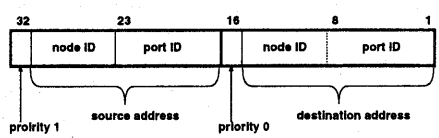

図5　パケットヘッダフォーマット

4. 評　　価

4.1　評価環境

Responsive Processor の評価用基板を対象として、実装および測定を行なった。評価用基板では、コアプロセッサは100MHzで動作するが、*Responsive Link* は6.25Mbaudで動作する。

4ノードを用いて測定を行なった。各ノードにおける許容遅延のthresholdは実際にはアプリケーションに依存するものである。そこで、今回の測定では画像通信を想定した。動画の転送においては、1秒間に30フレームの品質が要求される。バッファリングをしない場合を想定しても、2フレーム以下の遅延である必要がある。1フレームを送信する時間が約$33ms$であるため、許容遅延のthresholdを$33ms$に指定した。また、$330ms$をコントロールパケットの送信間隔とした。したがって、コントロールパケットの到着間隔が$360ms$以上になった時に許容できない遅延が発生しているとみなす。

4.2　過負荷・無負荷状態における評価

基本性能として、無負荷状態と過負荷状態におけるコントロールパケットの受信間隔の計測を行なった。粒度が、$10.2\mu s$のタイマを用いて計測したときの結果を図6に示す。ただし、縦軸は到着時間を、横軸はコントロールパケット数を表す。

図6において、過負荷時・無負荷時共に、16番目と32番目のデータが到着した時に受信間隔が大きくなっている。これは、DPMのバッファサイズによるもので、このときにDPMのバッファの値がメモリに退避されるために現れる。したがって、この遅延は本質的なものではない。

高負荷状態においては、コントロールパケットの遅延状態にばらつきが見られる。これは、他のパケットとの衝突に原因がある。*Responsive Link* において、衝突はスイッチ部において生じる。したがって、スイッチ部において他のパケットとの衝突が起こると、パケット送出の際の遅延が増大し、結果として受信側での到着間隔は図6の様になる。

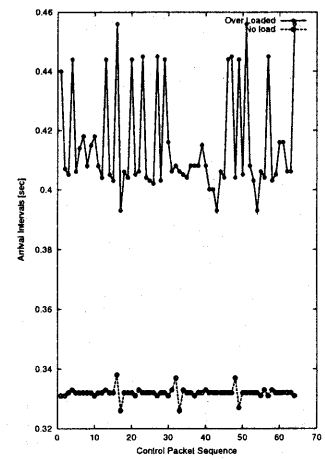

図6　高負荷時・無負荷時のコントロールパケットの到着時間

4.3 代替経路探索時における評価

データが許容できないような遅延を検出した場合に、代替経路を探す本機構の評価を flooding のみを用いて行なった。測定の結果を図7に示す。縦軸は到着時間間隔、横軸はコントロールパケットの受信番号を表す。

グラフが大きく上下しているのは、タイマの影響があると考えられる。本機構では、受信側でコントロールパケットを受け取る間隔を計測しているため、スイッチにおいて大きな遅延を生じた場合に、次のパケットがすぐ後に続いてしまうという状況が起こる可能性がある。したがって、衝突によりコントロールパケットの到着間隔が大きくなった後に、それに続くコントロールパケットの到着間隔が小さくなるという事が起きている。実際には、コントロールパケットの受信間隔が大きく変動する現象は、*Critically Congested Section* の存在により引き起こされている。高負荷状態を検出した場合には、新たな経路を探索する。その間、従来の経路では、優先度を上げることにより、混雑の影響を緩和した。いずれの場合でも、経路の変更後は、コントロールパケットの受信間隔に変動は見られない。

過負荷状態におけるコントロールパケットの到着時間平均値と、経路変更後のコントロールパケットの到着時間の平均値を比べると、経路変更により負荷の影響を回避したと結論づけることが出来る。

以上では、QoS を保証するために障害となる遅延を発生させる経路上の区間 (*Critically Congested Section*) を発見し、その代替経路を探索し、経路の変更を行なった。その結果、本機構は QoS を実現するための遅延管理において有効であった。

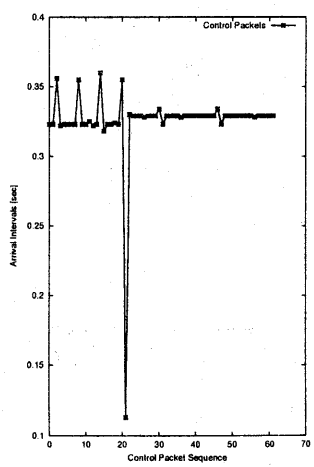

図7 代替経路探索におけるコントロールパケットの到着時間間隔1

5. 結　　論

通信を行う際に、送信ノードから、通常のデータと共に一定時間間隔でコントロールパケットを送出し、各ノードにおいてその受信間隔を監視し遅延管理を行なった。その際に *Responsive Link* の持つ、データの優先度を上げる機構を用いることにより、経路探索時の QoS の低下を避ける事に対する有効性を示した。また、個々のノードが管理しているリンクの接続状況の基づき、*CCS* の開始ノードから代替経路の探索を行うことで、動的に QoS を保証するために有効な経路変更が行われることを示した。本機構全体の更なる詳しい検証と、優先度を考慮した経路変更機構の提案が今後の課題である。

参 考 文 献

1) Wang, Z. and Crowcroft, J.: QoS Routing for Supporting Resource Reservation, in *IEEE Journal on Selected Areas Communications* (1996).
2) Shin, K. G. and Chou, C. C.: A Distributed Routing Scheme for Establishing Real-Time Channel , in *Sixth IFIP Int'l Conf. on High Performance Networking Conf.(HPN'95)*, pp. 319–329 (1995).
3) 山崎信行, 松井俊浩：並列分散リアルタイム制御用レスポンシブプロセッサ, 日本ロボット学会誌, Vol. 19, No. 3, pp. 352–361 (2001).
4) Yamasaki, N. and Matsui, T.: A Functionally Distributed Responsive Micro Controller for Distributed Real-Time Processing, in *Proceedings of the 1998 IEEE/RSJ International Conference on Intlligent RObоit and Systems. Innovative Robotics for Real-World Applications.IRos '97*, pp. 793–798, IEEE (1997).
5) Chen, S. and Nahrsted, K.: "Distributed Quality-of-Service Routing in High-Speed Networks Based on Selective Probing", in *IEEE Seventh International Conference on Computer, Communication and Networks*, IEEE (1998).
6) Manimaran,G., Rahul,H.S. and Murthy,C.R.: A New Distributed Route Selection Approach for Channel Establishment in Real-Time Networks, in *IEEE/ACM TRANSACTIONS ON NETWORKING*, Vol. 7, pp. 698–709, IEEE (1999).
7) 奥埜貢士：μ-PULSER における実時間通信機構の設計と実装, Master's thesis, 慶應義塾大学 (1999).