

研究会推薦論文

アイソクロナススケジューラを応用した
QoS 保証型通信の設計と実装竹内 理[†] 岩 寄 正 明[†]
中 原 雅 彦[†] 中 野 隆 裕[†]

近年のネットワークハードウェアの目覚ましい進歩にともない、インターネット上で高品質な連続メディアデータ転送を実現したいという要求が高まってきた。我々は上記要求を満たすべく、Ethernet セグメント間の通信における QoS を保証可能な通信方式 (QoS 保証型通信) の設計と実装を行った。本通信方式は、連続メディアデータのルーティング処理の実行時に必要となる資源をあらかじめ確保するためのシグナリングプロトコル (RTIPSIG プロトコルと RTIPSIG-multi プロトコル) と、確保した資源を過不足なく使用しつつルーティング処理を実行するルーティング実行方式 (周期ルーティング機構) を提供する。本稿では上記技術の概要とその評価結果につき述べる。評価の結果、本通信方式は、Fast Ethernet の物理限界とほぼ等しいスループットを持つ連続メディアデータを、その品質を低下させることなくルーティング可能であることを確認した。また、送信ノードや中途のルータの CPU を飽和することなく、十分実用的な数の受信ノードに対する連続メディアデータのマルチキャストが可能であることも確認した。

A Design and Implementation of QoS Assurable Communication
Using Isochronous SchedulerTADASHI TAKEUCHI,[†] MASAOKI IWASAKI,[†] MASAHIKO NAKAHARA[†]
and TAKAHIRO NAKANO[†]

The realization of continuous media data transfer through the internet is being expected as the development of high speed network. In order to satisfy this expectation, We designed and implemented original communication mechanism (*QoS assurable communication mechanism*) which can assure QoS (Quality of Service) of communication between nodes on different Ethernet segments. This communication mechanism provides two features. First, it provides signaling protocols (*RTIPSIG* and *RTIPSIG-multi* protocol) which reserves resources necessary to relay continuous media data. Second, it provides original routing mechanism which relay continuous media data using the resources reserved by those signaling protocols described above. This paper gives outlines and evaluation results of *QoS assurable communication mechanism*. We confirmed that *QoS assurable communication mechanism* enables relay of continuous media data at the rate nearly equal to Ethernet capacity without any quality loss. Besides, we confirmed that this mechanism enables multicast of continuous media data to ample receiving nodes because it can keep CPU load of sending node and routers low.

1. はじめに

近年、ギガビットイーサネットや ATM 網などの高速なネットワークハードウェアが数多く出現してきた。これらのネットワークは MPEG 形式などに圧縮されたビデオデータを転送するのに十分なハードウェア性能を備える。それにともない、ビデオデータや音声

データをはじめとする連続メディアデータをインターネット上で高品質に転送したいという要求が高まりつつある¹⁾。

上記要求を充足するためには、高性能なハードウェアだけではなく、高速 I/O や QoS (Quality of Service) 保証機構を提供する OS や通信プロトコルが不可欠となる。我々は連続メディア処理向きマイクロカーネル

[†] 日立製作所システム開発研究所

Systems Development Laboratory, Hitachi Ltd.

本論文の内容は 1998 年 11 月のマルチメディア通信と分散処理ワークショップにて報告され、同研究会主査により情報処理学会論文誌への掲載が推薦された論文である。

HiTactix, および Ethernet セグメント上におけるリアルタイム通信プロトコル「TTCP/ITM」の研究開発を推進してきた^{2)~5)}. これらにより, FastEthernet の物理限界とほぼ等しいスループットを達成する高速 I/O や Ethernet で接続されたセグメント内における通信の QoS 保証が可能となった. しかしながら, インターネット上での連続メディアデータ転送を実現するには, ルーティング処理の QoS 保証という課題が残されていた. 本研究ではさらにルーティング処理の QoS を保証可能な通信方式, 特に一般に QoS 保証が困難であるとされている⁵⁾ Ethernet セグメント間のルーティング処理の QoS を保証可能とする通信方式を新規に提案する. 上記既開発技術と本通信方式を組み合わせるにより, Ethernet セグメント内だけではなく Ethernet セグメント間においても高品質な連続メディアデータの転送が可能となる.

ルーティング処理の QoS を保証する通信方式は, 以下を提供する必要がある⁶⁾.

- (1) ルーティング実行時に必要となる資源(ネットワーク帯域やバッファなど)を連続メディアデータを格納したパケット(以後「リアルタイムパケット」と呼ぶ)の転送前にあらかじめ予約するためのシグナリングプロトコル
- (2) リアルタイムパケットを上記資源を用いながらアプリケーションが要求したレートでルーティングするルーティング実行方式

本研究では, ユニキャスト通信において(1)を実現する RTIPSIG プロトコルと(2)を実現する RTIP プロトコルのユニキャスト機構の概要とその定量的な性能評価結果につき示す. さらに, マルチキャスト通信において(1)を実現する RTIPSIG-multi プロトコルと(2)を実現する RTIP プロトコルのマルチキャスト機構の概要とその定量的な性能評価結果についてもあわせて示す.

2. QoS 保証型通信の概要

RTIP プロトコルのユニキャスト機構と RTIPSIG プロトコル, もしくは RTIP プロトコルのマルチキャスト機構と RTIPSIG-multi プロトコルを組み合わせて QoS 保証を実現している通信を本稿では以下「QoS 保証型通信」と呼ぶ. 本章では QoS 保証型通信の概要につき述べる.

2.1 設計目標

QoS 保証型通信は以下の実現を目標に設計した.

QoS の保証 Ethernet セグメント間において高品質な連続メディアデータの転送を実現するのに十分

な通信の QoS 保証を実現する. ここでいう「QoS 保証」とは, 送信ノード上のアプリケーションが, RTIPSIG プロトコルまたは RTIPSIG-multi プロトコルによる資源予約の際に宣言したレートでリアルタイムパケットを送信した場合, 上記レートと中途のルータにおけるルーティングレートとの一致を保証することをいう.

既存アプリケーションおよび OS の有効活用 受信ノード側のアプリケーションを改変することなく, 従来方式の通信から QoS 保証型通信への移行が可能である. また QoS 保証型通信は, 受信ノードに RTIPSIG プロトコルスタック, RTIPSIG-multi プロトコルスタック, RTIP プロトコルスタックが実装されていなくても実現可能である. すなわち, 従来方式の通信から QoS 保証型通信に移行する際に, 受信ノード上で動作する既存アプリケーションおよび OS を有効活用できる.

2.2 QoS 保証型通信の動作手順

QoS 保証型通信を用いた送信を実行するアプリケーションは, 以下の手順で QoS 保証型通信のアプリケーションインタフェースを呼び出し, RTIPSIG プロトコルや RTIP プロトコルなどの各プロトコルスタックを起動する.

- (1) リアルタイムパケットを送信する送信ノード上のアプリケーションは, 送信開始前に RTIPSIG プロトコルスタックまたは RTIPSIG-multi プロトコルスタックを用いた資源予約を要求するアプリケーションインタフェース(以後, 「allocate_resource 関数」と表記する)を呼び出す. RTIPSIG パケットまたは RTIPSIG-multi パケットが送信ノードや中途のルータの間で送受され, 後にアプリケーションが要求したレートでリアルタイムパケットを転送するのに十分な量の資源予約がなされる. また, 後のリアルタイムパケットのルーティング処理を実行する際に必要となるルーティング情報の構築も, 上記プロトコルを用いて行う.
- (2) 送信ノード上のアプリケーションは, 上記資源予約完了後, リアルタイムパケットの送信を要求するアプリケーションインタフェース(以後, 「rtsend 関数」と表記する)を呼び出す. リアルタイムパケットは RTIP プロトコルのユニキャスト機構またはマルチキャスト機構を用いて, 上記で確保した資源を過不足なく使用しつつ, アプリケーションが要求したレートで受信ノードまで到達する.

次章以降で、RTIP SIG プロトコル、RTIP プロトコルのユニキャスト機構、RTIP SIG-multi プロトコル、RTIP プロトコルのマルチキャスト機構のそれぞれの概要につき順に述べる。

3. RTIP SIG プロトコル

RTIP SIG プロトコルは、異なる Ethernet セグメントに属する送受信ノード間におけるリアルタイムパケットのユニキャスト通信を実行する際に、中途のルータにおいて必要となる資源をあらかじめ予約することを目的としたシグナリングプロトコルである。RTIP SIG プロトコルは、2.1 節で示した設計目標を達すべく、パケット折り返し機構と資源予約機構を保持する。パケット折り返し機構は受信ノード上の既存アプリケーションおよび OS の有効活用を可能にし、資源予約機構は QoS の保証を実現するための機構である。以下各機構の概要につき述べる。

3.1 パケット折り返し機構

パケット折り返し機構は、allocate_resource 関数がアプリケーションより発行された際の RTIP SIG パケット (REQ パケットおよび RESV パケット) のフローを図 1 に示すように制御する。

- (1) 送信ノードで allocate_resource 関数が発行されると、送信ノードおよび各ルータは受信ノードに向かって REQ パケットを送信、中継する。REQ パケットには送受信ノードの IP アドレス、送信元ポート番号、受信先ポート番号、QoS 情報 (リアルタイムパケットの送信周期、1 周期あたりのパケット数、パケットサイズ) などが含まれる。REQ パケットは受信ノードと同一セグメントに属するルータまで到達する。
- (2) 受信ノードと同一セグメントに属するルータが REQ パケットを受信すると、REQ パケットの送信経路と逆方向に RESV パケットを送信する^{*}。この RESV パケットには、REQ パケットと同様の情報が格納されている。RESV パケットを受信した各ルータは、該当パケットを

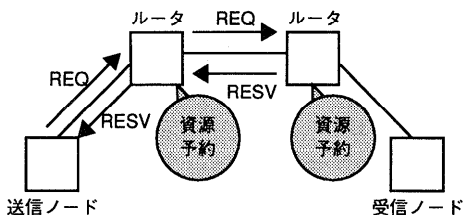


図 1 REQ パケット, RESV パケットの流れ

Fig. 1 Flows of REQ and RESV packets.

前段ホップノード (自ルータが受信した REQ パケットの送信元ノードまたは送信元ルータ) に転送するほかに、以下を実行する。

- (a) RESV パケットを受信した各ノードまたはルータは RESV パケット内に含まれている QoS 情報を参照し、該当 QoS を保証するのに十分な資源予約を実行する。この資源予約の実行方法は 3.2 節で述べる。
- (b) さらに、RESV パケット内に含まれている送受信ノードの IP アドレスを用いて、後のリアルタイムパケットのルーティング処理を実行する際に必要となるルーティング情報の構築を行う。このルーティング情報の詳細は 4.2 節で述べる。

送信ノードが RESV パケットを受信し、かつ上記で述べた資源予約とルーティング情報の構築に成功すると allocate_resource 関数は正常終了する。

上記から明らかなように、REQ パケット、RESV パケットはいずれも受信ノードまで到達しない。そのため、受信ノードに RTIP SIG プロトコルスタックは必要ない。また、allocate_resource 関数は送信ノード上のアプリケーションで発行され、受信ノード上のアプリケーションの改変は必要ない。これにより、RTIP SIG プロトコルは受信ノード上で動作する既存アプリケーションおよび OS の有効活用を可能にしている。

3.2 資源予約機構

3.1 節で述べたように、RESV パケットを受信した各ルータおよび送信ノードは、リアルタイムパケットのルーティング処理実行時に必要となる CPU 時間、バッファ、ネットワーク帯域を予約する。具体的には以下を実行する。

- (1) 各ルータは、自ルータに到達すべく予約されたリアルタイムパケットの到達レートの総計を管理する。この到達レートの総計は、RESV パケットに含まれる QoS 情報から算出可能である。RESV パケットによる資源予約要求を受け取った際に、総計が一定値を超えるか否かをチェックする。この値を超えた場合、該当パケット

^{*} ルータは REQ パケットを受信した際に、該当 REQ パケットの送信元ノードまたは送信元ルータの IP アドレスを記録しておく。各ルータは RESV パケットの送信先ノードまたは送信先ルータを上記情報に基づき決定する。そのため RESV パケットの送信経路は、REQ パケットの送信経路の逆方向となることを保証できる。

トのルーティング処理実行に十分な CPU 時間またはバッファを割り当てられないとして、該当資源予約要求は拒否される。

- (2) 各ルータは、該当通信が要求する自ルータと次段ホップノード（自ノードまたは自ルータが送信した REQ パケットの送信先ルータ^{*}）との間のネットワークの帯域を予約する。この予約は、以下に示す帯域管理サーバとの TTCP パケットの送受により実現する⁵⁾。
- (a) 帯域管理サーバは Ethernet セグメントにつき 1 つ存在する。該当セグメントに属するすべてのノードまたはルータは、ネットワーク帯域を使用するに先立ち、帯域予約要求を帯域管理サーバに対して送信する。そのため帯域管理サーバは、該当 Ethernet セグメントを流れるトラフィックの総量を把握可能である。
- (b) ルータは帯域管理サーバに対して帯域予約要求を示す TTCP パケットを送信する。帯域管理サーバは、要求された帯域を割当て可能か否かを判断する。割当て可能である場合には、帯域使用許可を通知する TTCP パケットをルータに返信する。
- (3) RTIPSIG プロトコルスタックはストリーム（送受信ポート間を接続する論理的な伝送路。送受信ノードの IP アドレスおよび送受信ポート番号により一意に識別可能である）ごとに状態を管理する。さらに、その状態の 1 つとして上記帯域予約処理実行中状態も定義している。そのため、(2) の TTCP パケットの送受中に他ストリームの RTIPSIG 制御パケットに対する処理も並行実行可能である。

4. RTIP プロトコルのユニキャスト機構

RTIP プロトコルは、異なる Ethernet セグメントに属する送受信ノード間におけるリアルタイムパケットのユニキャスト通信を実現する。特に RTIP プロトコルは、リアルタイムパケットのルーティングレートと、送信ノード上のアプリケーションが RTIPSIG プロトコルを用いた資源予約の際に宣言した送信レートが一致することを保証する。

本章では、RTIP プロトコルで用いるパケット形式

（すなわち、リアルタイムパケットのパケット形式）およびルーティング情報、ルーティング処理手順の概要について説明した後、上記保証を実現するための周期ルーティング機構について述べる。

4.1 パケット形式

本節では、RTIP プロトコルで使用するパケット形式の概要につき述べる。

4.3 節で述べるように、周期ルーティング機構はリアルタイムパケットのルーティング処理の QoS を保証可能とすべく、

- (1) リアルタイムパケットと非リアルタイムパケットを判別可能であること
- (2) リアルタイムパケットがどのストリームに属するかを判別可能であること

を要求する。

また、受信ノード上で動作する既存アプリケーションおよび OS を有効活用保するために、RTIP プロトコルで用いるパケット形式は、既存プロトコルのパケット形式と同一である必要がある。

上記要求を満たすべく、RTIP プロトコルで用いられるパケット形式は、従来の IP プロトコルパケット形式と同一とした。そのため、受信ノードは従来の IP プロトコルスタックのみを備えていれば、リアルタイムパケットの受信が可能となる。リアルタイムパケット、非リアルタイムパケットの判別は IP ヘッダの TOS フィールド^{**}により行う。リアルタイムパケットには 4.2 節で説明する非 0 のストリーム ID の値が、非リアルタイムパケットには 0 が格納されている。

4.2 ルーティング情報とルーティング処理手順

RTIP プロトコルで使用する主要なルーティング情報は、

- (1) 次段ホップノードの IP アドレス
- (2) 前段ストリーム ID
- (3) 次段ストリーム ID

である。3.1 節で述べたとおり、送信ノードおよびルータは RESV パケットを受信した際に、上記各ルーティング情報を構築する。

次段ホップノードの定義は 3.2 節で述べたとおりである。次段ホップノードは、IP ルーティングテーブルと RESV パケットに含まれている受信ノードの IP アドレスを検索することにより決定できる。

ストリーム ID は、各ルータを経由するストリームを識別するための ID である。この ID は図 2 に示すと

^{*} ただし、受信ノードと同一セグメントに属するルータに限り、その次段ホップノードは受信ノードであると定義する。

^{**} 従来の IP プロトコルスタックの実装では TOS フィールドを参照しない。そのため、従来の IP プロトコルでは TOS フィールドを事実上使用していない。

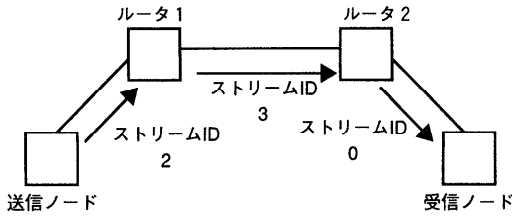


図 2 ストリーム ID
Fig. 2 Stream ID.

おり、同一ストリームであっても各 Ethernet セグメントごとに異なる値をとる。そして、自ノードまたは自ルータと次段ホップノードが属する Ethernet セグメントにおいて使用されるべきストリーム ID を「次段ストリーム ID」、自ノードまたは自ルータと前段ホップノードが属する Ethernet セグメントにおいて使用されるべきストリーム ID を「前段ストリーム ID」と呼ぶ。たとえば図 2 では、ルータ 1 の前段ストリーム ID は 2 であり、次段ストリーム ID は 3 である。

各ルータは、前段ストリーム ID から次段ホップノードの IP アドレスおよび次段ストリーム ID の情報が取得可能になるようにルーティング情報を構築する[☆]。各ルータは、リアルタイムパケットを受信した際に、該当パケットの TOS フィールドを参照することにより前段ストリーム ID を取得する。そして上記ルーティング情報から、次段ホップノードの IP アドレスおよび次段ストリーム ID を取得する。取得した次段ストリーム ID を該当パケットの TOS フィールドに設定し、該当パケットを次段ホップノードに転送することにより、ルータは該当パケットのルーティング処理を完了する。

4.3 周期ルーティング機構

周期ルーティング機構は、RTIPSIG プロトコルの資源予約機構により確保した資源を過不足なく使用することにより、Ethernet セグメント間でのルーティング処理の QoS (結果として異なる Ethernet セグメントに存在する送受信ノード間通信の QoS) を保証する。具体的には以下を実現する。

- (1) リアルタイムパケットのルーティング処理のために予約した CPU 時間を該当処理に割り当てることを保証すること。
- (2) リアルタイムパケットのルーティング処理のためにつねに一定数以上のバッファを確保すること。
- (3) リアルタイムパケットのルーティング処理実行

[☆] 前段ストリーム ID は、該当 ID を TOS フィールドに格納したパケットを受信するルータにおいて一意となるよう、RTIPSIG パケットの送受時に決定される。

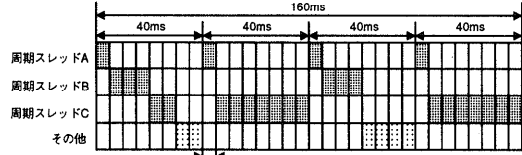


図 3 タイムスロット・テーブル
Fig. 3 A timeslot table.

表 1 周期動作の要求例

Table 1 An example of requests for interval and CPU-time.

	T	L
A	40 ms	5 ms
B	80 ms	15 ms
C	160 ms	80 ms

により使用する帯域が、予約した帯域と一致すること。

- (4) Fast Ethernet の物理限界と等しいスループットで到達するパケットをパケット喪失なくルーティング可能な処理能力を持つこと。

上記実現のため、周期ルーティング機構は、CPU 時間保証機構、バッファ保証機構、帯域保証機構、一括送受信機構を備える。このうち周期ルーティング機構、帯域保証機構はアイソクロナススケジューラ³⁾を応用して実現した。本節では、アイソクロナススケジューラの概要をのべた後、上記各機構の概要につき述べる。

4.3.1 アイソクロナススケジューラの概要

アイソクロナススケジューラは、周期動作を要求するスレッド (以後、「周期スレッド」と呼ぶ) の厳密な周期駆動と、一定割合の CPU 時間の占有保証の実現を目的としたスケジューラである。本スケジューラは、タイムスロット割当て部とディスパッチ部からなる。

タイムスロット割当て部では、本スケジューラは周期スレッドが発行する周期動作要求を受理する。本要求受理時に要求駆動周期 (T) と 1 周期あたりに占有したい CPU 時間 (L) もあわせて受理する。要求を受理すると本スケジューラは、各周期スレッドからの要求を充足するようなタイムスロットテーブルを作成する。タイムスロットテーブルとは図 3 に示すように、タイム割込み発生時刻を境界に時間を分割し (以後、この分割単位をタイムスロットと呼ぶ)、各タイムスロットごとにディスパッチすべき周期スレッドを記したテーブルである。図 3 は表 1 に示す周期動作要求を充足するタイムスロットテーブルである。

タイムスロットテーブルの作成を完了すると、アイ

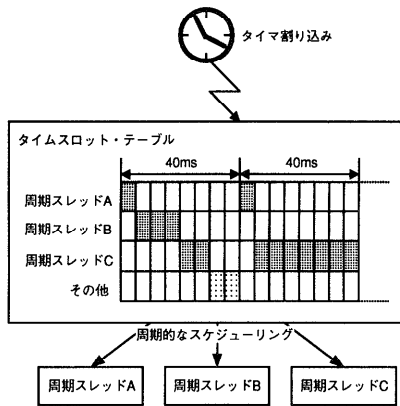


図4 周期スレッドの起動

Fig. 4 Cyclic thread driving based on timeslot table.

ソクロナススケジューラのディスパッチ部は、図4に示すようにタイム割り込みを契機に起動し、タイムスロットテーブルに記された周期スレッドを順番にディスパッチする。

アイソクロナススケジューラは、各周期スレッドのためのCPU時間をあらかじめ一定周期ごとに確保する。そのため、ある周期スレッドの実行は、他の周期スレッドの実行により妨げられない。また本スケジューラは、周期動作を要求しない通常のスレッドを、タイムスロットテーブルで周期スレッドが割り当てられていない時間を用いてラウンドロビン方式でスケジューリングする。そのため、周期スレッドの実行が、通常スレッドの実行により妨げられることもない。すなわち本スケジューラは、周期スレッドの厳密な周期駆動と、一定割合のCPU時間の占有保証が実現できる[☆]。

4.3.2 CPU 時間保証機構

既存のルーティング処理方式では、非リアルタイムパケットとリアルタイムパケットのIPルーティング処理は同一スレッドで実行する。そのため非リアルタイムパケットが大量に到達しCPU負荷が高くなった場合、リアルタイムパケットのルーティング処理実行が遅延し、該当ルーティング処理のQoS保証が不可能となる。上記課題を解決するため、周期ルーティング機構は図5に示すスレッド構成をとる。

- (1) Ethernetドライバは、Ethernetボードに到達したパケットをリアルタイムパケットと非リア

ルタイムパケットに振り分ける(IPヘッダのTOSフィールドが0であるか否かにより判別する)。前者であればIPスレッド、後者であればRTIPスレッドがルーティング処理を実行する。そして、ルーティング処理が完了したパケットをITMスレッド、およびEthernetドライバがネットワークに送信する。

- (2) Ethernetドライバは最高優先度のスレッドとして実現している。また、RTIPスレッド、ITMスレッドはアイソクロナススケジューラを用いた周期スレッドとして実現している。これらのスレッドは厳密な周期駆動(10ms周期)、および一定割合のCPU時間の占有(それぞれ1周期につき6ms、1msのCPU時間を占有可能)を保証されている。IPスレッドの優先度は図示されているスレッドの中では最低となる。

リアルタイムパケットのルーティング処理は、最高優先度のEthernetドライバ、および一定割合のCPU時間の占有を保証されているRTIPスレッド、ITMスレッドのみによって実行される。したがって、非リアルタイムパケットが大量に到達しCPU負荷が高くなっても、リアルタイムパケットのルーティング処理に、各10ms周期において一定割合以上のCPU時間を割り当てることを保証できる。

4.3.3 バッファ保証機構

Ethernetはハードウェアレベルにおけるパケット属性の区別(リアルタイムパケットか非リアルタイムパケットかの区別)が不可能である。既存のルーティング実行方式を用いると、非リアルタイムパケットが大量に到達した際にルータのバッファが枯渇し、リアルタイムパケットが喪失する可能性がある。

上記課題を解決するため、周期ルーティング機構は図5に示すように、

- (1) IPプロトコルスタックが使用する非リアルタイム受信キュー、および非リアルタイム送信キューにキューイング可能なパケット数の上限を設ける。
- (2) 上限を超えるパケットキューイングを行おうとした場合には、ただちに該当パケットを破棄(バッファを解放)する。

これにより、残りのバッファは、

- (1) フリーバッファキューに接続されている
 - (2) リアルタイム受信キューに接続されている
 - (3) リアルタイム送信キューに接続されている
 - (4) Ethernetドライバにより使用されている
- のいずれかであることが保証される。すなわち一定数

[☆] 現在の実装では、Lの値をタイム割り込み発生間隔より小さい単位で(1ms単位で)指定可能である。タイムスロット割当て部は、1msごとにスケジューリングすべき周期スレッドを記したタイムスロットテーブルを作成する。またディスパッチ部は、タイム割り込み発生ばかりでなく、カーネルモード走行中に行う経過時間チェックをも契機にして起動する。

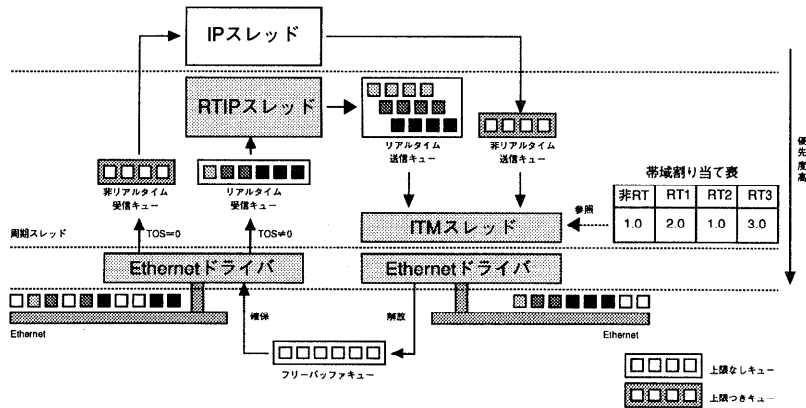


図5 周期ルーティング機構のスレッド構成およびデータ構造
 Fig. 5 Threads and data structures for isochronous routing mechanism.

以上のバッファがつねにリアルタイムパケットのルーティング処理のために確保されていることが保証され、バッファ枯渇によるリアルタイムパケットの喪失を防ぐことができる。

4.3.4 帯域保証機構

ルーティング処理の QoS を保証するためには、CPU 時間を確保するばかりでなく、RTIPSIG プロトコルの資源予約機構により確保したネットワーク帯域を過不足なく使用しながらルーティング処理を実行する必要がある。

この要求を充足するため周期ルーティング機構は、

- (1) RTIP スレッドは 10ms 周期で周期駆動する。そして、10ms の間に到達したリアルタイムパケットをストリームごと（すなわち IP ヘッダの TOS フィールドの値ごと）に別キューにして、ITM スレッドに受け渡す。
- (2) IP スレッドは、非リアルタイム送信キューを用い ITM スレッドに送信すべき非リアルタイムパケットを受け渡す。
- (3) ITM スレッドは 10ms 周期で周期駆動する。そして、各送信キューから帯域割当て表に記された量のパケットをデキューする。デキューしたパケット群を Ethernet ドライバに受け渡す。Ethernet ドライバはただちにパケット送信を実行する。

帯域割当て表に登録されているリアルタイムパケット用の帯域は、RTIPSIG プロトコルによる帯域予約に成功した際に、予約した帯域に従って設定される。この際に非リアルタイムパケット用の帯域も決定される。リアルタイムパケット用の帯域と非リアルタイムパケット用の帯域の総計^{*}はネットワークの物理限界を超えないように設定される。かつ、ITM スレ

ッドは帯域割当て表に見合った量のパケットを周期的に Ethernet ドライバに受け渡す。上記手順でパケット送信を実行することにより、リアルタイムパケットのルーティング実行時に使用する帯域は予約帯域と一致することを保証できる。

4.3.5 一括送受信機構

周期ルーティング機構は Fast Ethernet に対する送受信処理を一括化して実行し、スレッドスイッチや割込みハンドラ起動のオーバーヘッドを低減している。これによりルーティング処理能力の向上を実現している。

Ethernet ドライバは以下の手順で受信処理を実行する。

- (1) Ethernet ドライバは複数の受信コマンドからなるコマンドチェーンを Ethernet ボードに対し発行する。
- (2) Ethernet ボードは上記受信コマンドの数だけパケットが到達するか、もしくはタイムアウト（5ms）により外部割込みを発生する。この外部割込みを契機に Ethernet ドライバは再起動し、受信処理を一括実行する。そして再び上記 (1) を実行する。

ITM スレッドおよび Ethernet ドライバは以下の手順で送信処理を実行する。

- (1) ITM スレッドは、10ms 周期に駆動する。そして、各送信キューから帯域割当て表に従った数のパケットをデキューする。デキューしたパ

^{*} ここでいう「総計」は、同一セグメントに属するすべてのノードにおいて割り当てられたリアルタイムパケット用の帯域と非リアルタイムパケット用の帯域の総計を示す。したがって、Ethernet のような伝送路を複数ノードで共有するネットワークの場合でも、周期ルーティング機構を用いればネットワークが輻輳状態に陥ることはない。

ケット群を1つのキューにして Ethernet ドライバに受け渡す。そして Ethernet ドライバを起動する。

- (2) Ethernet ドライバはコマンドチェーンを利用して、受け渡されたパケット群の一括送信を実行する。
- (3) 上記一括送信実行完了後に外部割込みが発生する。この外部割込みを契機に Ethernet ドライバはバッファの解放処理を実行する。

5. RTIPSIG-multi プロトコル

RTIPSIG-multi プロトコルは、送信ノードから複数の受信ノードに対するリアルタイムパケットのマルチキャスト通信を実行する際に、中途のルータにおいて必要となる資源をあらかじめ予約することを目的としたシグナリングプロトコルである。

RTIPSIG-multi プロトコルは、送信ノードがあらかじめリアルタイムパケットの配送先となる受信ノード群を指定して、上記資源予約を実行する（通信コネクションを確立する）ことを仮定する。配送先に新たな受信ノードを追加する場合には、送信ノード上のアプリケーションは、新たな通信コネクションを確立し、以後、複数のコネクションに対して同一のリアルタイムパケットの配送処理を実行しなくてはならない[☆]。一方、配送先から特定の受信ノードを削除する場合には、以下にのべる「単一受信ノード配送停止機構」を利用する。

RTIPSIG-multi プロトコルは以下の機能を備える。

- (1) 送信ノードと、送信ノード上のアプリケーションが `allocate_resource` 関数を発行する際に指定した複数の受信ノードとの間におけるリアルタイムパケットのマルチキャスト通信を実現する際に、中途のルータにおいて必要となる資源を一括して予約する機能。
- (2) 上記で予約した資源を一括して解放する機能。
- (3) `allocate_resource` 関数で指定した複数の受信ノードのうち、送信ノード上のアプリケーションが指定した特定の受信ノードに対するリアルタイムパケットの配送のみを停止する機能。

RTIPSIG-multi プロトコルは、上記 (1) を「一括資源予約機構」で、(2) を「一括資源解放機構」で、(3) を「単一受信ノード配送停止機構」で実現する。本章ではこのうち、「一括資源予約機構」と「単一受信

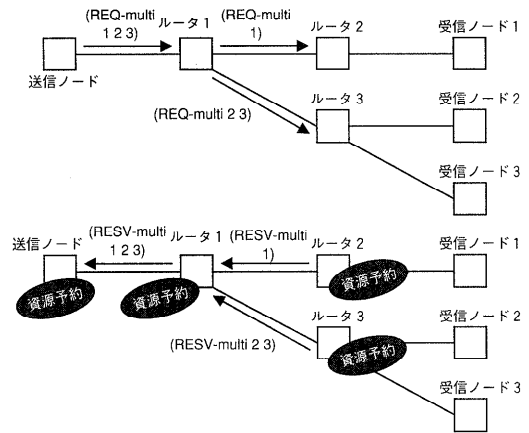


図6 一括資源予約機構における RTIPSIG-multi パケットのフロー

Fig. 6 Flows of RTIPSIG-multi packets in executing collective resource reservation.

ノード配送停止機構」の概要につき説明し、RTIPSIG-multi プロトコルが既存アプリケーションおよび OS の有効活用を可能にしている（受信ノード上のアプリケーションの変更が必要ないこと、および RTIPSIG-multi パケットが受信ノードに到達しないこと）ことを明らかにする。

5.1 一括資源予約機構

一括資源予約機構は、送信ノードと複数の受信ノードとの間に介在するルータにおいて、送受信ノード間におけるマルチキャスト通信を実行する際に必要となる資源を一括予約する。

一括資源予約機構を実行する際の RTIPSIG-multi パケットのフローの例を図 6 に示す。

- (1) 送信ノード上で動作するアプリケーションが `allocate_resource` 関数を発行すると、送信ノードから `allocate_resource` 関数で指定された受信ノード群に向かって REQ-multi パケットが流れる。REQ-multi パケットには REQ パケットと同様の情報が格納される。図 6 では、受信ノード 2 および 3 の IP アドレスが格納された REQ-multi パケットを (REQ-multi 2 3) と表記している。
- (2) 送信ノードおよび REQ-multi パケットを受信した各ルータは、IP ルーティングテーブル、および REQ-multi パケットに格納されている受信ノード群の IP アドレスを参照し、該当受信ノード群にリアルタイムパケットを配送する際の次段ホップノード群を決定し、決定した次段ホップノード群に REQ-multi パケットを転送する。転送する REQ-multi パケットには、

[☆] 通信コネクションの数が増大した場合、複数の通信コネクションを1つの通信コネクションに束ねる処理を送信ノード上のアプリケーションが実行することも可能である。

REQ-multi パケットの送信先となるルータを次段ホップノードに持つ受信ノード群の IP アドレスが格納される。たとえば図 6 のルータ 1 は、ルータ 2 に (REQ-multi 1) を、ルータ 3 に (REQ-multi 2 3) を転送する。

- (3) 受信ノードと同一セグメントに属するルータは、REQ-multi パケットを受信すると、REQ-multi パケットの送信経路と逆方向に RESV-multi パケットを送信する。RESV-multi パケットは REQ-multi パケットと同様の情報が格納されている。
- (4) (2) で決定したすべての次段ホップノードから RESV-multi パケットを受信した各ルータは、RESV-multi パケットを 1 パケットだけ前段ホップノードに転送する。転送する RESV-multi パケットには、受信した各 RESV-multi パケットに格納された受信ノードの IP アドレスをすべて格納する。たとえば図 6 のルータ 1 は、ルータ 2 から (RESV-multi 1) を、ルータ 3 から (RESV-multi 2 3) を受信し、送信ノードに (RESV-multi 1 2 3) を転送する。その他にルータは以下を実行する。
- (a) RESV-multi パケットに含まれている QoS 情報を参照し、該当 QoS を保証するのに十分な資源予約を実行する。この資源予約の方法は 3.2 節で述べた方法と同様である。
- (b) RESV-multi パケットに含まれている送受信ノードの IP アドレスを用いて、後のリアルタイムパケットのルーティング処理を実行する際に必要となるルーティング情報の構築を行う。このルーティング情報の詳細は 6.2 節で述べる。

送信ノードが RESV-multi パケットを受信し、かつ上記で述べた資源予約とルーティング情報の構築に成功すると allocate_resource 関数は正常終了する。

5.2 単一受信ノード配送停止機構

単一ノード配送停止機構は、送信ノード上のアプリケーションが allocate_resource 関数を発行した際に指定した受信ノード群のうち、特定の 1 つの受信ノードに対するリアルタイムパケットの配送のみを停止することを要求する。RTIP SIG-multi プロトコルが本機構を実行した後にアプリケーションが rtsend 関数を発行した場合、上記 1 つの受信ノードを除いた残りの受信ノードに対してのみリアルタイムパケットが配

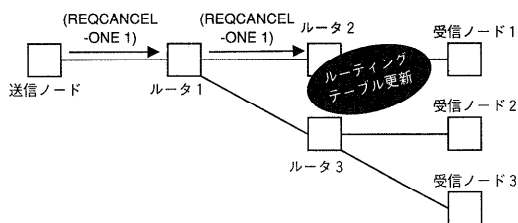


図 7 単一受信ノード配送停止機構における RTIP SIG-multi パケットのフロー

Fig. 7 Flows of RTIP SIG-multi packets in requesting to stop RTIP packets transfer for specified receiving node.

送される。

単一受信ノード配送停止機構を実行する際の RTIP SIG-multi パケットのフローの例を図 7 に示す。

- (1) 送信ノード上で動作するアプリケーションが単一受信ノードに対するリアルタイムパケットの配送停止を要求する関数を発行すると、REQCANCEL-ONE パケットが上記関数発行時に指定された受信ノードに向かって流れる。REQCANCEL-ONE パケットには、上記受信ノードの IP アドレスが格納される。
- (2) REQCANCEL-ONE パケットに IP アドレスが格納されている受信ノードと同一セグメントに属するルータが上記パケットを受信すると、該当ルータに保存されているルーティング情報のうちの配送停止ビットをオンにする。6.2 節で述べるように、該当ビットがオンである場合、RTIP マルチキャスト機構は、該当受信ノードに対するリアルタイムパケットの転送を行わない。

なお、本機構は一括資源予約機構で予約した資源の解放はいっさいしない。資源解放は一括資源解放機構を用いて行われる。

6. RTIP プロトコルのマルチキャスト機構

RTIP プロトコルのマルチキャスト機構は、送信ノードから異なる Ethernet セグメントに属する複数の受信ノード群に対してリアルタイムパケットのマルチキャスト通信を実現する。

本章では、RTIP マルチキャスト機構で用いられるパケット形式およびルーティング情報、ルーティング処理手順の概要を述べる。

6.1 パケット形式

RTIP マルチキャスト機構は、4.1 節で述べたパケット形式を用いる。ただし、IP ヘッダの宛先アドレスフィールドの値に、マルチキャストアドレス (クラス D の IP アドレス) が格納されている点が異なる。

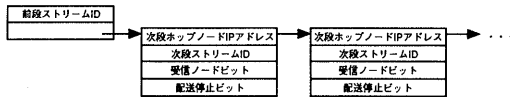


図 8 RTIP マルチキャスト機構で用いられるルーティング情報
Fig. 8 Routing information used by RTIP multicast mechanism.

通常の RTIP プロトコルのユニキャスト機構およびマルチキャスト機構を用いて配送されるパケットは、ともにルータで Ethernet ドライバから RTIP スレッドに受け渡される。RTIP スレッドはその IP ヘッダ部の宛先アドレスフィールドを参照して、4.2 節で述べた処理手順または 6.2 節で述べる処理手順でルーティング処理を実行するかを決定する。

6.2 ルーティング情報とルーティング処理手順

RTIP マルチキャスト機構で用いる主要なルーティング情報を図 8 に示す。図中で用いられている前段ストリーム ID、次段ストリーム ID、次段ホップノードの用語定義は 4.2 節で述べたとおりである。受信ノードビットは次段ホップノードが受信ノードであるか否かを示す。配送停止ビットは次段ホップノードへのリアルタイムパケット配送を実行すべきか否かを示すビットである。

ルータは 1 つのストリームにつき複数の次段ホップノードを持つ可能性がある。そのため本ルーティング情報では、各次段ホップノードごとの情報（次段ホップノード IP アドレス、次段ストリーム ID、受信ノードビット、配送停止ビット）をリスト状に接続している。

ルーティング情報は前段ストリーム ID からその他すべての情報が取得可能になるように構築される。RTIP スレッドは、RTIP マルチキャスト機構で配送されるリアルタイムパケットを受信すると、該当パケットの TOS フィールドを参照することにより前段ストリーム ID を取得する。そしてこの前段ストリーム ID から、図 8 に示したルーティング情報を取得する。そして取得した各次段ホップノードに対して以下を実行する。

- (1) 取得した次段ホップノードに対応する配送停止ビットがオフである場合にのみ、リアルタイムパケットを該当次段ホップノードに転送する。
- (2) 取得した次段ホップノードに対応する受信ノードビットがオフである（次段ホップノードが受信ノードではない）場合、受信したリアルタイムパケットの TOS フィールドを対応する次段ストリーム ID に書き換えたパケットを該当次段ホップノードに転送する。
- (3) 取得した次段ホップノードに対応する受信ノード

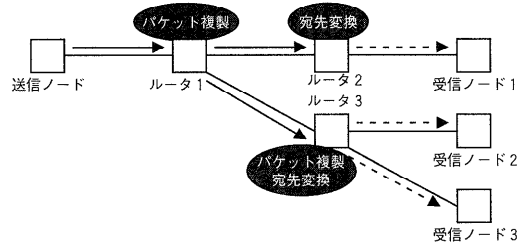


図 9 RTIP マルチキャスト機構におけるリアルタイムパケットの流れ
Fig. 9 Real-time packets flow in executing RTIP multicast mechanism.

——▶ マルチキャストアドレスを宛先に持つパケットの流れ

---▶ ユニキャストアドレスを宛先に持つパケットの流れ

ドビットがオンである場合（次段ホップノードが受信ノードである）場合、受信したリアルタイムパケットの TOS フィールドを対応する次段ストリーム ID に書き換え、かつ宛先アドレスフィールドを対応する次段ホップノードの IP アドレスに書き換えたパケットを該当次段ホップノードに転送する。

上記ルーティング処理手順を用いて各ルータはリアルタイムパケットのルーティング処理を実行することにより、図 9 に示すように、送信ノードから複数の受信ノードに対するリアルタイムパケットのマルチキャスト通信が実現可能となる。

上記手順 (3) により、受信ノードと同一セグメントに属するルータがリアルタイムパケットを通常の IP パケットと同一形式に変換する。そのため、受信ノードは RTIP プロトコルのマルチキャスト機構を備える必要はなく、既存アプリケーションおよび OS の有効活用を可能にしている。

また、RTIP マルチキャスト機構を用いてリアルタイムパケットのマルチキャスト通信を行う際にも、ルータは 4.3 節で述べた周期ルーティング機構を用いてリアルタイムパケットのルーティング処理を実行する。そのため上記通信の QoS も保証可能となる。

7. 性能評価実験

本章では、QoS 保証型通信の性能評価実験の概要およびその結果について示す。実験は、QoS 保証型通信の QoS 保証性能を評価する実験と、RTIPSIG-multi プロトコルと RTIP プロトコルのマルチキャスト機構を用いた通信のスケラビリティを評価する実験の 2 つを行った。

7.1 QoS 保証性能の評価実験

図 10 に示すシステムのルータにおける、リアルタ

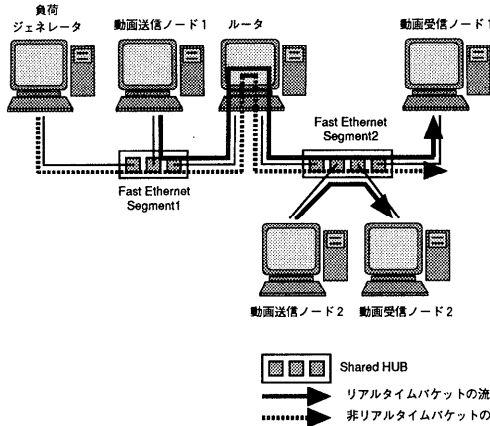


図 10 QoS 保証性能測定実験システム概要

Fig. 10 Experimental system for evaluating QoS ability.

リアルタイムおよび非リアルタイムパケットの入出力量の時間変動およびパケット喪失率を測定することにより、周期ルーティング機構の QoS 保証性能を評価した。

図 10 に示すシステムは以下の構成をとる。

- (1) Fast Ethernet により構築されたセグメント 1 およびセグメント 2 をルータ (Pentium Pro[☆] 200 MHz 搭載マシン) で接続した。ルータには RTIP プロトコルおよび RTIP SIG プロトコルが実装されている。
- (2) セグメント 1 には、動画送信ノード 1、負荷ジェネレータを置く。セグメント 2 には、動画受信ノード 1、動画送信ノード 2、動画受信ノード 2 を置く。
- (3) 動画送信ノード 1 は動画受信ノード 1 に向かって、約 48 Mbps (180 パケット/40 ms) のリアルタイムパケット (ビデオデータ) を送信する。
- (4) 負荷ジェネレータは、セグメント 1 からセグメント 2 に向かって約 24 Mbps (80 パケット/40 ms) の非リアルタイムパケットを送信する。
- (5) 動画送信ノード 2 は動画受信ノード 2 に向かって約 8 Mbps のリアルタイムパケットを送信する。この負荷によりセグメント 2 がセグメント 1 より高負荷となる。

入出力量の時間変動の測定結果を図 11 および図 12 に、パケット喪失率の測定結果を表 2 に示す。グラフの横軸は経過時間を、縦軸は 40 ms の間に入出力が行われたパケット総数を表す。これらから以下が分かる。

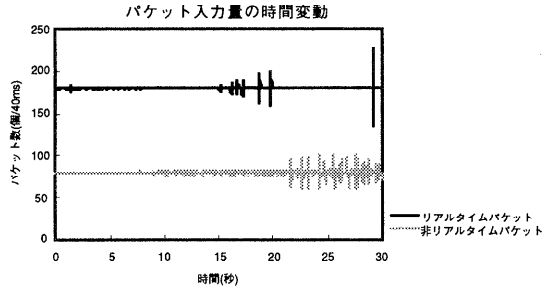


図 11 パケット入力量の時間変動

Fig. 11 A change of the number of input packets.

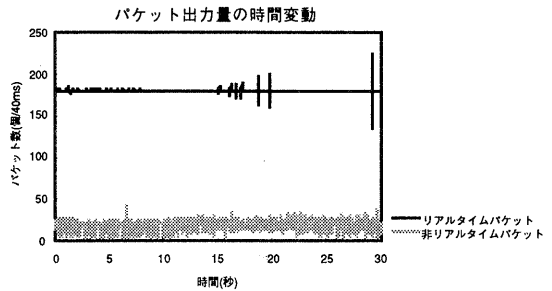


図 12 パケット出力量の時間変動

Fig. 12 A change of the number of output packets.

表 2 パケット喪失率
Table 2 Packet loss ratio.

パケットの種類	パケット喪失率
リアルタイム	0.00%
非リアルタイム	75.00%

- (1) セグメント 2 には動画送信ノード 2 によりネットワーク負荷がかけられている。そのためルータは一部パケットを破棄せざるをえない。ルータは 75.00% の非リアルタイムを破棄しているにもかかわらず、リアルタイムパケットは破棄していない。
- (2) 非リアルタイムパケットの出力量のゆらぎは平均 29.41% に達しているのに対し、リアルタイムパケットの出力量のゆらぎは平均 2.51% に抑えられている。

上記 (1) は、リアルタイムパケットのルーティング処理用につねに一定量以上のバッファが確保されていることを示す。また上記 (2) は、ルータに 70 Mbps 以上のレートでパケット到達する状況下においても、リアルタイムパケットのルーティング処理用の CPU 時間や送信帯域が確保されていることを示している。以上より、QoS 保証型通信は Fast Ethernet の物理限界の 70% を超えるレートで到達する連続メディアデータを、その品質を低下させることなくルーティング可

☆ Pentium Pro は Intel の登録商標です。

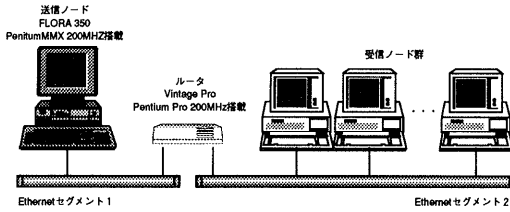


図 13 予備実験システム

Fig. 13 A system for preliminary experiments.

能であることが分かる。

7.2 スケーラビリティの評価実験

本節では、RTIP-multi プロトコルおよび RTIP プロトコルのマルチキャスト機構を用いたリアルタイムパケットのマルチキャスト通信のスケラビリティを評価するために行った実験の概要と、その結果につき示す。本実験では、マルチキャスト通信のスケラビリティを、マルチキャストの対象となる受信ノードの数を変動させた場合の、送信ノードおよび中途のルータの CPU 負荷の変動を測定することにより評価した。

実験は、実機を用いた予備実験と、予備実験結果を用いたシミュレーションからなる。以下、行った予備実験およびシミュレーションにつき述べる。

7.2.1 実機による予備実験

予備実験システムの構成を図 13 に示す。

- (1) 送信ノードには Pentium MMX * 200 MHz 搭載マシンを、ルータには Pentium Pro 200 MHz 搭載マシンを使用する。
- (2) ルータは 2 つの Ethernet セグメント (Ethernet セグメント 1 および 2) を接続する。

実験方法を以下に示す。

- (1) 送信ノードから可変個の受信ノード群に向かって、1 つの受信ノードにつき 8 Mbps の動画データを以下の 2 通りの方法を用いて行った。
 - (a) RTIP SIG プロトコルによる資源予約を実行した後に、RTIP プロトコルのユニキャスト機構を用いて配送する方法 (以後「RTIP SIG を用いた方法」と表記する)。
 - (b) RTIP SIG-multi プロトコルによる資源予約を実行した後に、RTIP プロトコルのマルチキャスト機構を用いて配送する方法 (以後「RTIP SIG-multi を用いた方法」と表記する)。
- (2) 上記 2 通りの方法を用いた場合のそれぞれにつき、受信ノード数を可変にした際の、送信ノード

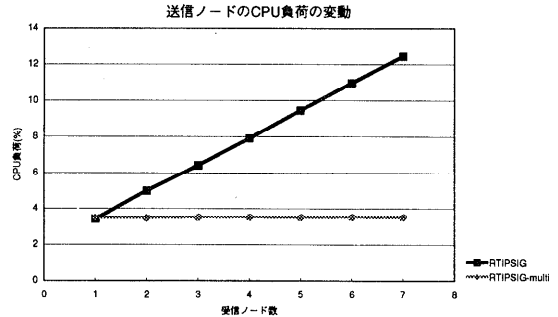


図 14 送信ノードの CPU 負荷の変動

Fig. 14 CPU load variation of sending node.

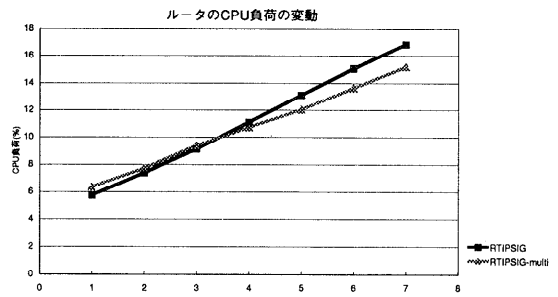


図 15 ルータの CPU 負荷の変動

Fig. 15 CPU load variation of router.

ドおよびルータの CPU 負荷の変動を測定した。

送信ノードの CPU 負荷の測定結果を図 14 に、ルータの CPU 負荷の測定結果を図 15 に示す。それぞれのグラフの横軸は受信ノード数を、縦軸は送信ノードまたはルータの CPU 負荷を示している。

上記実験結果から、RTIP SIG を用いた方法、RTIP SIG-multi を用いた方法とも、送信ノードが N ストリームの動画データを配送する場合の CPU 負荷 $L_s(N)$ が、以下の式で近似可能であることが分かった[☆]。

$$L_s(N) = 1.92 + 1.50 \times N \quad (1)$$

また、RTIP SIG を用いた方法および RTIP SIG-multi を用いた方法でルータが N ストリームの動画データを配送する場合の CPU 負荷 $L_{ru}(N)$, $L_{rm}(N)$ は、以下の式で近似可能であることも分かった。

$$L_{ru}(N) = 3.68 + 1.88 \times N \quad (2)$$

$$L_{rm}(N) = 4.87 + 1.46 \times N \quad (3)$$

7.2.2 シミュレーションによる実験

前項で得られた実験結果をもとに、より現実に近いネットワーク環境における RTIP SIG-multi を用いた

[☆] 本予備実験における RTIP SIG-multi を用いた方法では、受信ノード数にかかわらず、送信ノードはつねに 1 ストリームのみの動画データ配送を実行している。

[☆] Pentium MMX は Intel の登録商標です。

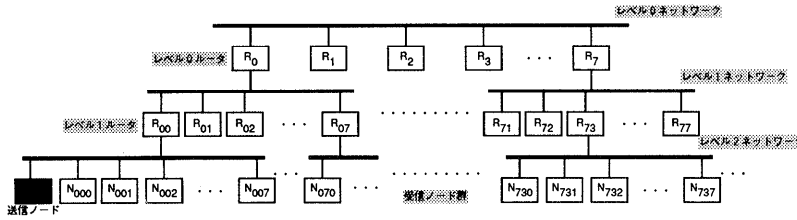


図 16 仮定したネットワーク構成

Fig. 16 Assumed network topology.

方法のスケラビリティをシミュレーションにより評価した。

本シミュレーションで仮定したネットワーク構成を図 16 に示す。

- (1) ネットワークを形成するルータおよび送信ノード群はツリー状構成をとる。
- (2) ツリーの先頭に位置するネットワークを「レベル 0 ネットワーク」と名付ける。レベル 0 ネットワークには、8 台のルータが接続されている。この 8 台のルータを「レベル 0 ルータ」と名付ける。
- (3) 各レベル 0 ルータには、レベル 0 ネットワークの他にもう 1 つのネットワークが接続されている。このネットワーク群を「レベル 1 ネットワーク」と名付ける。また、各レベル 1 ルータの他に 8 台のルータが接続されている。このルータ群を「レベル 1 ルータ」と名付ける。
- (4) 同様に各レベル 1 ルータには、レベル 1 ネットワークの他にもう 1 つのネットワークが接続されている。このネットワーク群を「レベル 2 ネットワーク」と名付ける。また各レベル 2 ネットワークには、レベル 1 ルータの他に 8 台の受信ノード群が接続されている。さらに、レベル 2 ネットワークのうちの 1 つのネットワークには、レベル 1 ルータと 8 台の受信ノード群の他に送信ノードが接続されている。

上記ネットワーク構成において、送信ノードから可変個の受信ノード群（受信ノード群のネットワーク上での存在位置はランダムであると仮定する）に対して、1 つの受信ノードにつき 8 Mbps の動画データを配送する。配送は「RTIPSIG を用いた方法」と「RTIPSIG-multi を用いた方法」の 2 通りの方法で行った。各方法において受信ノード数を変動させた場合の、

- (1) 送信ノードの CPU 負荷の変動
 - (2) ルータの平均 CPU 負荷の変動
 - (3) ルータの最大 CPU 負荷の変動
- をシミュレーションにより求めた。ここでいう「ルー

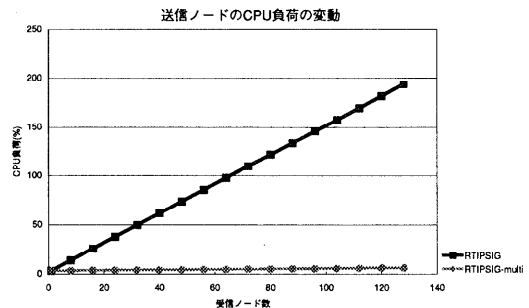


図 17 送信ノードの CPU 負荷の変動

Fig. 17 CPU load variation of sending node.

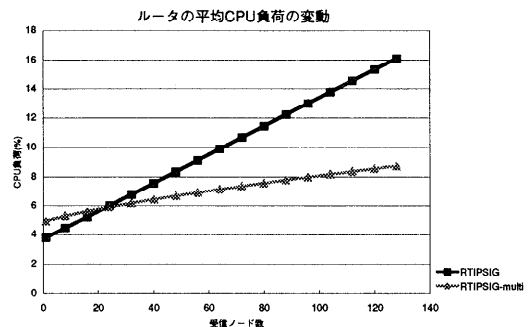


図 18 ルータの平均 CPU 負荷の変動

Fig. 18 Average CPU load variation of router.

タの平均 CPU 負荷」とは、各レベル 0 ルータおよび各レベル 1 ルータの CPU 負荷の平均値を指す。また「ルータの最大 CPU 負荷」とは、各レベル 0 ルータおよび各レベル 1 ルータの CPU 負荷の最大値を指す。送信ノードおよびルータの CPU 負荷は、前項で示した $L_s(N)$, $L_{ru}(N)$, $L_{rm}(N)$ に従うと仮定した。

シミュレーションの結果得られた送信ノードの CPU 負荷変動を図 17 に、ルータの平均 CPU 負荷の変動を図 18 に、ルータの最大 CPU 負荷の変動を図 19 に示す。それぞれグラフの横軸は受信ノード数を、縦軸は各 CPU 負荷を示している。

シミュレーションの結果、以下が明らかになった。

- (1) RTIPSIG を用いた方法では、受信ノード数が 65 を超えると送信ノードの CPU が飽和

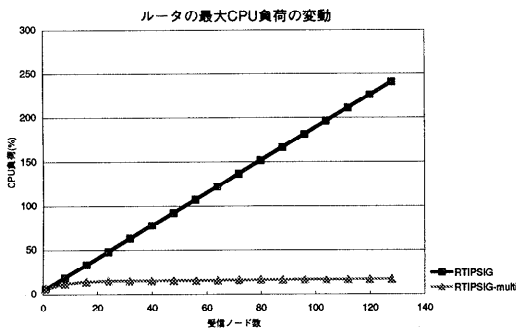


図 19 ルータの最大 CPU 負荷の変動

Fig. 19 Maximum CPU load variation of router.

する (CPU 負荷が 100% を超える)。しかし、RTIPSIG-multi を用いた方法では、受信ノード数がいくら増大しても送信ノードの CPU は飽和しない。これは、RTIPSIG-multi を用いた方法では、送信ノードは自ノードに隣接しているルータおよび受信ノード (図 16 で仮定したネットワーク構成では 9) を超えるストリーム数の動画データを配送することがありえないためである。

- (2) RTIPSIG を用いた方法では、受信ノード数が 52 を超えると一部のルータの CPU が飽和する (最大 CPU 負荷が 100% を超える)。しかし、RTIPSIG-multi を用いた方法では、受信ノード数がいくら増大しても CPU が飽和するルータは出現しない。これは (1) と同様の理由による。
- (3) 受信ノード数が 24 を超えると、RTIPSIG-multi を用いた方法は、RTIPSIG を用いた方法と比べてルータの平均 CPU 負荷が低くなる。これは、RTIPSIG-multi を用いた方法では、ネットワーク上を流れる連続メディアデータの総量を低く抑えられるためである。

8. 関連研究

QoS 保証型通信を実現するためのシグナリングプロトコルとしては RSVP^{6),7)}、ルーティング実行方式としては、WFQ^{8),9)} および CBQ¹⁰⁾ が知られている。

RSVP では受信ノードが資源予約要求を発行する。そのため、RSVP を用いた通信を実現するためには受信ノード上のアプリケーションの変更を必要とする。そのため従来方式の通信からの移行の容易さという点で RTIPSIG プロトコルおよび RTIPSIG-multi プロトコルの方が優れる。また、RSVP ではネットワーク帯域予約処理実行中の状態が定義されていない。その

ため TTCP⁵⁾ のように帯域予約のために通信が必要となる処理の実行中に、他ストリームの RSVP パケットに対する処理を実行できない。RTIPSIG プロトコルおよび RTIPSIG-multi プロトコルでは上記が可能である。

WFQ, CBQ は、ネットワーク帯域の飽和によるルーティング処理の QoS 喪失防止を目的としたルーティング実行方式である。そのため CPU 能力飽和時やバッファ枯渇時には、ルーティング処理の QoS を保証できない。また、Ethernet のような伝送路を共有するネットワークにおいて衝突が多発した場合、それにとまなう伝送遅延により通信の QoS が損なわれる可能性がある。QoS 保証型通信が提供する周期ルーティング機構は、CPU 時間保証機構、バッファ保証機構、帯域保証機構を備えることにより、上記課題をすべて解決している。

9. おわりに

本稿では、Ethernet セグメント間における高品質な連続メディアデータ転送を実現する QoS 保証型通信を設計、実装し、その性能の定量的評価を実施した。評価の結果、本通信方式は、Fast Ethernet の物理限界の 70% を超えるレートで到達する連続メディアデータを、(非リアルタイムトラフィックが混在する状況下においても) その品質を低下させることなくルーティング可能であることを確認した。また、送信ノードや中途のルータの CPU を飽和することなく、十分実用的な数の受信ノードに対する連続メディアデータのマルチキャストが可能であることも確認した。

参考文献

- 1) 松原 敦：インターネットを中心に家庭の情報化が始まる，日経バイト，11月号，pp.140-143 (1996)。
- 2) Iwasaki, M., Takeuchi, T., Nakamura, S., Nakano, T., Nakahara, M., Taguchi, S. and Serizawa, K.: An Experiment of Isochronous Video Data Transfer on Network, *Worldwide Computing and Its Applications*, pp.334-349 (1997)。
- 3) 竹内ほか：連続メディア処理向き OS の周期駆動保証機構の設計と実装，情報処理学会論文誌，Vol.40, No.2, pp.1-13 (1999)。
- 4) Iwasaki, M., Takeuchi, T., Nakahara, M. and Nakano, T.: Isochronous Scheduling and its Application to Traffic Control, *IEEE Real-Time System Symposium*, pp.14-25 (1998)。
- 5) 中野ほか：Ethernet 上で QoS を保証する通信

方法の設計と実装, 情報処理学会コンピュータシステムシンポジウム, pp.35-42 (1997).

- 6) Zhang, L., Deering, S., Estrin, D., Shenker, S. and Zappala, D.: RSVP: A New Resource ReSerVation Protocol, *IEEE Network*, Vol.7, No.5, pp.8-18 (1993).
- 7) Braden, R., Zhang, L., Berson, S., Herzog, S. and Jamin, S.: Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification, RFC-2205 (1997).
- 8) Parekh, A.: A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks, MIT PhD dissertation (1992).
- 9) Bennett, J.C.R. and Zhang, H.: WF2Q: Worst-case Fair Weighted Fair Queueing, *IEEE INFOCOMM '96* (1996).
- 10) Floyd, S. and Jacobson, V.: Link-Sharing and Resource Management Models for Packet Networks, *IEEE/ACM Trans. Networks* (1996).
- 11) 竹内ほか: アイソクロナススケジューラを応用した QoS 保証型ルーティング方式の設計と実装, 情報処理学会マルチメディア通信と分散処理ワークショップ論文集, pp.119-126 (1998).
- 12) Topolcic, C.: Experimental Internet Stream Protocol, Version 2 (ST-II), RFC-1190 (1990).
- 13) Diot, C., Dabbous, W. and Crowcroft, J.: Multipoint Communication: A Survey of Protocols, Functions and Mechanisms, *IEEE Journal on Selected Areas in Communications*, Vol.15, No.3 (1997).
- 14) Deering, S.: Multicast Routing in a Datagram Internetwork, Stanford University PhD dissertation (1991).

(平成 11 年 4 月 19 日受付)

(平成 11 年 7 月 1 日採録)

推薦文

インターネット上での連続メディアデータ転送という今日重要とされている課題に対して, 資源の予約プロトコルと, 確保した資源を用いて周期的にルーティング処理を実行する機構とによる解決策を提案している. QoS 保証可能なルーティング方式として, 非常に興味深い研究である. 定性的な部分に関して, 他の方式と比べての優位性が示されており, また, シミュレーション環境においても, 良好な結果を示されている. (マルチメディア通信と分散処理研究会主査 滝沢 誠)



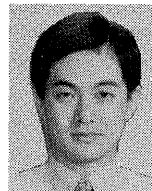
竹内 理 (正会員)

昭和 44 年生. 平成 4 年東京大学理学部情報科学科卒業. 平成 6 年同大学大学院理学系研究科情報科学専攻修士課程修了. 同年 (株) 日立製作所システム開発研究所入社. 連続メディア処理向きマイクロカーネルの研究, 特にリアルタイムスケジューリング方式, リアルタイム通信方式, ヘテロジニアス OS アーキテクチャに関する研究に従事.



岩寄 正明 (正会員)

昭和 33 年生. 昭和 56 年九州工業大学工学部電子工学科卒業. 昭和 58 年九州大学大学院総合理工学研究科情報システム学専攻修士課程修了. 同年 (株) 日立製作所中央研究所入所, 平成 5 年より同社システム開発研究所勤務. 並列推論マシン, 金融 OLTP システム, 並列計算機 SR2201 の OS 研究開発を経て, HiTactix の研究を開始, 現在に至る.



中原 雅彦 (正会員)

昭和 40 年生. 昭和 63 年東京農工大学工学部数理情報工学科卒業. 平成 2 年同大学大学院工学研究科修士課程修了. 同年 (株) 日立製作所システム開発研究所入社. ワークステーションの性能評価, 並列計算機用オペレーティングシステム, 連続メディア処理向きマイクロカーネル等の研究・開発に従事.



中野 隆裕 (正会員)

昭和 44 年生. 平成 5 年電気通信大学電気通信学部情報工学科卒業. 平成 7 年同大学大学院電気通信学研究科情報工学専攻修士課程修了. 同年 (株) 日立製作所システム開発研究所入社. 連続メディア処理向きマイクロカーネルの研究, 特に入出力方式やリアルタイム通信方式に関する研究に従事.