

# 仮想マシン環境における実時間通信を実現する 仮想NICの研究

鈴木 健一<sup>1</sup> 佐藤 未来子<sup>1</sup> 並木 美太郎<sup>1</sup>

**概要:** 仮想マシン (VM) 環境では, 仮想マシンモニタ (VMM) が各 VM に対して CPU やメモリ, 通信などのリソースを必要な QoS(Quality Of Service) に応じて割り当てる。通信の QoS を確保する場合は, VM ごとに帯域を割り当てる手法が一般的である。しかし, この帯域割り当てでは通信の遅延時間を制御することができず, テレビ会議システムや制御通信などで必要な通信の実時間性が保証されない。これは VMM 内部の packets 制御と物理ネットワークの転送制御において遅延が発生するためである。そこで本研究では, VMM 内部にデッドライン保証型 packets スケジューラを追加し通信の優先制御をすることで, これらの課題を解決し End-to-End の通信の遅延時間に対する実時間性保証を実現している。本稿では, VMM の EDF packets スケジューラのデッドライン保証に対する有効性とスケジューリングの特徴を明らかにするため実装と評価を行った。評価の結果, 転送時間のデッドラインをそれぞれ 3ms, 5ms, 8ms に設定した三つのフローに対して, 擾乱フローがある状態でも最大 3.0ms, 4.9ms, 7.7ms の遅延時間に抑え, 仮想マシン間の実時間通信の保証を確認した。

**キーワード:** 仮想マシン, QoS, リアルタイム通信

## 1. はじめに

仮想マシン (VM) 環境では, 仮想マシンモニタ (VMM) が各 VM に対して CPU やメモリ, 通信などのリソースを必要な QoS(Quality Of Service) に応じて割り当てる。通信の QoS を確保する場合は, VM ごとに帯域を割り当てる手法が一般的である。しかし, この帯域割り当ての手法では, 各通信の遅延時間は考慮されない packets スケジューリングが用いられる, 帯域の増大に伴い通信のリアルタイム性が損なわれる課題がある。また, VM へ実行権を与えるための VMM による仮想 CPU スケジューリングでは, VM の実行待ち時間により packets の送受信処理が遅延することで, リアルタイム性が損なわれる課題がある。

本研究では, VMM に付加するリアルタイム通信制御機構によりこの課題を解決し, 物理的に異なる物理マシン上で動作する VM 間のリアルタイム通信を実現する通信基盤「RTvNIC システム」を提案している。本稿では, 提案する VMM の packets スケジューラのデッドライン保証に対する有効性と, スケジューリングの特徴を明らかにするため実装と評価を行った。

RTvNIC システムでは, ゲスト OS が VMM の提供するリアルタイム通信向け仮想 NIC(RTvNIC) を利用することで, ゲスト OS の改変無しに VM 間のリアルタイム通信を行うことができる。VM 管理者は, VM 起動時のパラメータとして各 RTvNIC に packets の転送遅延時間のデッドライン時間と最低帯域を設定することができる。RTvNIC を持つ VM 間の通信は, VMM の通信制御によりこの設定値を守った通信が保証される。

## 2. RTvNIC システムの概要

本章では, VM 間のリアルタイム通信を実現する通信基盤 RTvNIC システムの技術課題と目標, および概要を述べる。

### 2.1 課題

VM によるリアルタイム通信を実現するには, 仮想マシンに特有の遅延時間の課題を解決する必要がある。まず, VMM 内部の packets スケジューリングで発生する遅延の課題である。一つの物理マシン上に複数の VM が動作している場合, VM 上のゲスト OS の送信 packets を調停するために, VMM において packets スケジューリングが必要となる。このとき, ゲスト OS の送信した通信の帯域の合計が物理 NIC(Network Interface) を超える場合, VMM に

<sup>1</sup> 東京農工大学  
Tokyo University of Agriculture and Technology,  
Tokyo 184-8588, Japan

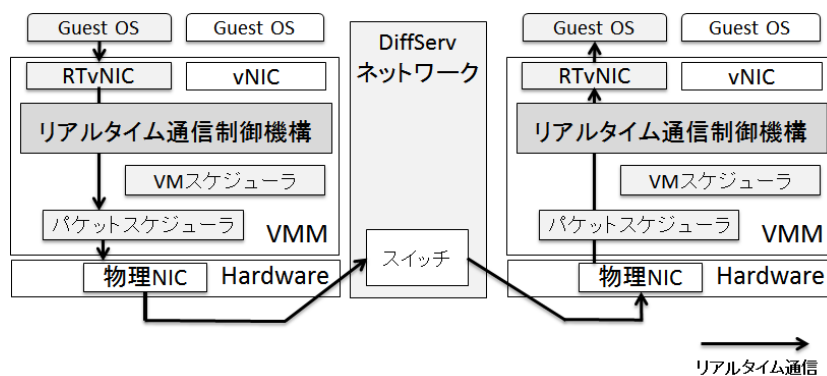


図 1 RTvNIC システムの全体構成

おけるパケットスケジューリングのキューイング遅延の増大に伴い通信の遅延時間が大きくなることで、通信のリアルタイム性が損なわれる。

次に、VM の仮想 CPU スケジューリングにおける遅延の課題である。VMM では、VM の仮想 CPU に実行権を割り当てるための仮想 CPU スケジューリングを行っている。このとき、実行権を割り当てていない VM は、次に実行権を割り当てられるまでの間、パケットの送受信処理を行うことができない。この実行権の割り当て待ち時間により、パケットの送受信処理が遅延することで、通信のリアルタイム性が損なわれる。

本論文で提案する RTvNIC システムでは、VMM のパケットスケジューラと仮想 CPU スケジューラに対して、リアルタイム通信の許容できる遅延時間であるデッドライン時間に応じた優先制御を付加することで、VMM における遅延の課題を解決する。

## 2.2 全体構成

提案する RTvNIC システムにおいては、リアルタイム通信制御機構が VMM における通信制御の中心となる。本システムの全体構成を図 1 に示す。リアルタイム通信制御機構は、RTvNIC を介してゲスト OS から受け取るリアルタイム通信制御のための情報とネットワークの QoS 情報を元に、VMM における RTvNIC、パケットスケジューラ、VM スケジューラを制御することで、リアルタイム通信の QoS を確保する。また、リアルタイム通信の端点のリアルタイム通信制御機構間でデッドラインミス情報を交換することで、End-to-End でのリアルタイム性の保証をする。

パケットスケジューラでは、リアルタイム通信制御機構から受け取った各通信のデッドライン情報に応じて、リアルタイム通信のデッドラインを守るパケットスケジューリングを行う。また、VM スケジューラでは、パケット受信時にパケットを受け取る VM に対する実行待ち時間を少なくするため、RTvNIC を持つ VM を優先する仮想 CPU スケジューリングを行う。

RTvNIC は、ゲスト OS の送信するパケットに対してリ

アルタイム通信制御機構において優先制御に利用する QoS 制御のための情報を付加する。この QoS 制御のための情報は、VM 起動時のパラメータなどにより設定された通信のデッドライン時間や帯域により VM ごとに異なるため、リアルタイム通信を行う各 VM に備えられている RTvNIC において付加する必要がある。また、デッドラインミスの発生に応じた処理を行うための、ゲスト OS の受信パケットのデッドライン時間以内の到着を判定する。検出したデッドラインミス情報は、パケットの送信元 VMM のリアルタイム通信制御機構に対して通知され、ゲスト OS の設定したデッドラインミスハンドラが起動される。さらに、ゲスト OS が VMM のリアルタイム通信制御機構に対してリアルタイム通信の設定を伝えるためのインターフェースの役割も持つ。

## 3. RTvNIC システムの設計

本章では、提案する RTvNIC システムの設計方針、および、VMM におけるリアルタイム通信を制御するためのパケットスケジューリング、VM スケジューリング、リアルタイム通信制御機構の設計について述べる。

### 3.1 設計方針

提案する RTvNIC システムでは、ゲスト OS の送信する通信を、送信元と宛先の IP アドレスの組で区別しリアルタイム性を確保する。また、必要に応じて送信元と宛先のポート番号も対象とすることで、アプリケーション単位でリアルタイム通信を利用できるようにする。IP アドレスを用いて通信を区別することから、RTvNIC システムでは IP/Ethernet プロトコルを用いた通信がリアルタイム性保証の対象となる。

RTvNIC システムでは、VM 間のリアルタイム通信の遅延時間を計測し、End-to-End での遅延時間を保証する。この遅延時間の計測には VMM 間の時刻同期が必要となるが、本研究では NTP, IEEE1588 PTP, GPS など目的とする精度を持った既存技術の利用を想定する。

また、リアルタイム通信を用いない VM や OS も RTvNIC

システムを用いた VM と通信可能な設計にすることで、相互接続性を確保する。ただし、それらの通信はリアルタイム性の保証の対象外とする。

通信の遅延はネットワークにおいても発生する可能性がある。本稿では、ネットワークが DiffServ など既存技術を用い、適切に QoS 保証されていることを前提とする。

### 3.2 リアルタイム通信路

提案する RTvNIC システムにおいては、リアルタイム性を確保する通信の単位として「リアルタイム通信路」と呼ぶ論理的な通信路を用いる。リアルタイム通信路の端点は VMM における RTvNIC であり、送信元と宛先の IP アドレス、および、必要に応じて送信元と宛先のポート番号により区別される。本システムでは、リアルタイム通信路上の片方向の通信に対して、端点の RTvNIC 間のリアルタイム性を保証する。ゲスト OS は、RTvNIC を介してリアルタイム通信路の設定を VMM へ要求することで、リアルタイム通信を利用することが出来る。なお、本稿では、リアルタイム通信路上の packets をリアルタイム packets と呼び、リアルタイム性を保証しない通信をノンリアルタイム通信、また、その通信路上の packets をノンリアルタイム packets と呼ぶ。

RTvNIC システムのリアルタイム通信路では、動画や音声の配信など一定量のデッドラインミスや packet loss が許容されるソフトリアルタイムな通信を想定したリアルタイム性を保証する。リアルタイム通信路の設定には、最低帯域、デッドライン時間、デッドラインミス処理を指定する。デッドラインに対して余裕のない通信は、デッドラインまで余裕がある通信より優先的に転送することで、できるだけ多くのリアルタイム packets のデッドラインを守る。

### 3.3 パケットスケジューリング

2.1 節で述べた VMM におけるパケットスケジューリングの遅延の課題を解決し、RTvNIC システムにおけるリアルタイム通信路を実現するため、パケットスケジューリングに EDF (Earliest Deadline First) アルゴリズムを用いる。VMM はゲスト OS の packet 送信の時刻や帯域を事前に知ることができない点を考慮し、各 packet のデッドラインまでの余裕時間に応じて動的に優先度を変更できる EDF アルゴリズムを選択した。EDF パケットスケジューリングでは、デッドライン時間がもっとも近い packet から優先的に送信するスケジューリングを実施する。

ノンリアルタイム通信については最も低い優先度としてスケジューリングするが、リアルタイム通信のデッドライン時間までに余裕がある場合に限り、ノンリアルタイム通信を優先して送信する。これは、ノンリアルタイム通信で利用が想定される TCP 通信において、通信の遅延時間の

増加によるスループットの低下を軽減するためである。また、リアルタイム通信の帯域が大きく増加した場合、ノンリアルタイム通信の帯域が圧迫される問題がある。これについては、必要に応じてリアルタイム通信に対して帯域制限をかけることで対応する。

### 3.4 VM スケジューリング

2.1 節で述べた VM の仮想 CPU スケジューリングにおける遅延の課題を解決するため、RTvNIC を持つ VM を優先的に実行権を割り当てる VM スケジューリングを行う。まず、まず、VMM の packet 受信後から宛先 VM が packet 受信処理を開始するまでの遅延時間に対しては、packet 受信をイベントとした実行権の割り当てを行う。受信した packet のデッドラインに応じて、VM のスケジューリング優先度を変更することで、遅延時間を低減する。また、packet 送信の処理に必要な実行権の割り当てについては、送信処理には外部イベントが発生しないため、RTvNIC を持つ VM に対して周期的な実行権の割り当てを行う。一定時間以内に必ず実行権が割り当てられるようにすることで、実行待ち時間による packet 送信の遅延を低減する。

これを実現するため、Cheng[1] らの手法のランキューを拡張したものを利用する。本研究で適用する VMM のランキューを図 2 に示す。Cheng[1] らは、VMM の一つである Xen[10] を対象として、仮想 CPU スケジューリングによる I/O 遅延を削減するため、EDF と Credit アルゴリズムを用いた二つのランキューを持つ仮想 CPU スケジューラを提案している。この手法では、リアルタイム性を必要とする VM に対しては、packet 送信処理のための周期的な実行と packet 受信などの外部イベントの発生時の実行のリアルタイム性を確保するために、EDF ランキューを用いることで実行待ち時間を削減した。また、リアルタイム性を必要としない VM は Credit ランキューを用い、実行時間の公平性を確保した。

RTvNIC システムでは、Cheng らの提案するダブルランキューのうち、EDF ランキューを拡張し、RTvNIC を持つ VM のスケジューリングに適用することで、VM のスケジューリングの遅延の課題を解決する。さらに、Cheng らの Credit ランキューを拡張し、RTvNIC を持たない通常の VM のスケジューリングに適用することで、VM の実行時間の公平性を確保する。

### 3.5 リアルタイム通信制御機構の設計

本節では、リアルタイム通信を制御する VMM 内部のリアルタイム通信制御機構における、ゲスト OS や VM から情報を元にしたリアルタイム通信の設定、ゲスト OS と VMM の協調、および、通信制御のための packet に対するマーキングについて述べる。

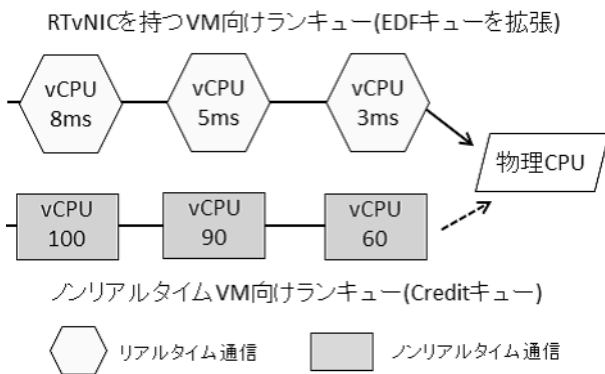


図 2 VM 実行のリアルタイム性を保証するダブルランキュー

### 3.5.1 リアルタイム通信の設定

リアルタイム通信を利用するためには、VMM のリアルタイム通信制御機構に対して、リアルタイム通信のための情報を設定する必要がある。本設定は、VM の起動時のパラメータとして、もしくは、ゲスト OS が RTvNIC 通してリアルタイム通信制御機構に伝える。リアルタイム通信を確保するために必要な情報は、送信元と宛先の IP アドレス、通信のデッドライン時間である。また、必要に応じて、リアルタイム通信を確保する送信元と宛先のポート番号、最低帯域、デッドラインミス・ハンドラを設定する。

また、リアルタイム通信を確保した後、リアルタイム通信路でデッドラインミスを起こした場合、VMM に設定したデッドラインミス・ハンドラが起動される。また、VMM からゲスト OS に対して、デッドラインミスを起こしたリアルタイム通信路の識別番号とデッドラインを超過した時間が通知される。このメッセージによりゲスト OS はデッドラインミスに対応する処理を行う事ができる。

### 3.5.2 ゲスト OS と VMM の協調

RTvNIC システムでは、利用するリアルタイム通信の設定に応じてゲスト OS と VMM の協調方法が変わる。まず、ゲスト OS 全体で同一のリアルタイム通信の設定を実施する場合、ゲスト OS とアプリケーションは透過的にリアルタイム通信を利用できる。リアルタイム通信の設定やデッドラインミスに対応する処理などの制御は完全に VMM に隠蔽されるため、ゲスト OS とアプリケーションに対する修正は必要ない。リアルタイム通信の設定は VM 起動時のパラメータとして設定することで、特定の VM 間でのアプリケーションのリアルタイム通信を実現する。

次に、アプリケーションごとにリアルタイム通信の設定を実施するためには、ゲスト OS に対して、アプリケーションが VMM とリアルタイム通信の設定を交換するためのシステムコールを提供する修正が必要である。アプリケーションは、ゲスト OS の提供するシステムコールを用い、リアルタイム通信の設定を VMM へ伝えることで、リアルタイム通信を利用できる。ゲスト OS のシステムコールとしては、既存の Socket にオプションを追加する方法

や、RTvNIC デバイスに対する読み書き (UNIX における `ioctl`) などを想定している。

最後に、アプリケーションがデッドラインミスに対応した処理を行うためには、ゲスト OS に対して、アプリケーションにデッドラインミス情報を通知するためのアップコールを提供する修正が必要である。アプリケーションに対するアップコールとしては、シグナル割り込みによる通知などを想定している。アプリケーションは、これらのゲスト OS の提供するシステムコールとアップコールを用いて、リアルタイム通信の設定とデッドラインミスに対する処理を行う。

### 3.5.3 マーキングによるデッドラインミス検出とネットワーク QoS 制御

提案する RTvNIC システムでは、リアルタイム通信の packets について、送信元 VMM から宛先 VMM に届くまでの遅延時間が指定されたデッドライン時間以内であることを確かめるために、各リアルタイムパケットの転送遅延時間を計測する。RTvNIC では、このデッドライン判定のために、ゲスト OS の送信パケットに対して、対象パケットのデッドライン時刻を付加する。デッドライン時刻が付加されたリアルタイムパケットを受信した RTvNIC では、付加された時刻と現在時刻からデッドラインミス判定を行う。ここで、デッドラインミスを検出した場合は、パケットの送信元 VMM に対して、デッドラインミスの発生を通知する。なお、デッドライン時刻を付加するパケットの領域は、IPv6 の宛先オプションヘッダや MPLS (Multi Protocol Label Switching) ヘッダなどの拡張ヘッダやイーサネットフレームのトレーラなどを想定している。

また、ネットワークにおける通信の QoS を確保する上で、パケットに QoS 情報を付加する必要がある場合も、RTvNIC において付加を行う。DiffServ でネットワークが QoS 制御されている場合は、適切な DSCP (Differentiated Services Code Point) 値をパケットに付加することなどを想定している。

## 4. 実装と評価

本章では、RTvNIC システムの実装と評価について述べる。本稿の評価では、RTvNIC システムにおける EDF パケットスケジューリングの特徴を明らかにするため、EDF パケットスケジューラの実装を行いリアルタイム通信の遅延時間を計測する実験を行った。

### 4.1 実装

RTvNIC システムの実装には、VMM として Linux のカーネルモジュールである KVM/QEMU-KVM(1.2.0) を利用した。この内、QEMU-KVM のエミュレートする仮想 NIC である e1000 を利用し、デッドライン時間の付加のための MPLS ラベルの挿入と DSCP 値のマーキングを実

表 1 実験 1,2 で用いる通信のパラメータ

	送信帯域	デッドライン
リアルタイム通信 1	2Mbps	3ms
リアルタイム通信 2	2Mbps	5ms
リアルタイム通信 3	2Mbps	8ms
リアルタイム通信 4	2Mbps	10ms
リアルタイム通信 5	2Mbps	15ms
リアルタイム通信 6	2Mbps	20ms
ノンリアルタイム通信	1Gbps	*

装し、これを RTvNIC とした。VMM において、EDF パケットスケジューリングを実現するために、Linux カーネルの機能である Traffic Control を改変し、EDF パケットスケジューラを実装した。また、EDF パケットスケジューラに、リアルタイム通信のパケットがデッドラインまで設定した時間以上の余裕時間がある場合には、ノンリアルタイム通信を優先して送信する機能を実装した。

仮想 CPU スケジューラは理想的なスケジューリングが行われる状態を想定し、RTvNIC を持つ VM の仮想 CPU を高優先度とする優先度スケジューラを実装した。評価に用いたマシンは、CPU として Intel Core i7 980、メモリは 24GB、NIC は 1Gbps のものを用いた。

## 4.2 評価

複数のリアルタイム通信をスケジューリングした場合の、VMM の EDF パケットスケジューリングにおけるパケット遅延の特徴を実験により明らかにする。また、このため、次の三つの実験を行った。

- 実験 1: VMM における EDF パケットスケジューリングの評価
  - 実験 2: VM 間通信の転送遅延時間の評価
  - 実験 3: RTvNIC システムによるオーバーヘッドの評価
- 次節から評価の詳細について述べる。

### 4.2.1 VMM における EDF パケットスケジューリングの評価

本 RTvNIC システムにおける VMM の EDF パケットスケジューリングにより、送信パケットの遅延時間をデッドライン時間以内に抑えられることを示すため、各パケットのパケットスケジューリングによる遅延時間を計測した。

#### (1) 評価方法

実験 1 では、表 1 で示す異なるデッドラインを持つ六つのリアルタイム通信と擾乱として輻輳を起こすためのノンリアルタイム通信を用いて計測を行う。いずれもネットワークベンチマーク iperf による UDP 通信を利用し、ペイロードを 1470Byte とする。また、リアルタイム通信のデッドラインに対して 2ms 以上の余裕時間がある場合は、ノンリアルタイム通信を優先して送信するよう設定した。同一物理マシン上で VM を七台起動し、一つの VM から一つのリアルタイム通信またはノンリアルタイム通信を送

信する。この構成で、VMM における EDF パケットスケジューラのキューにパケットがエンキューされた時刻から、デキューされるまでの遅延時間を計測する。計測は、擾乱のためのノンリアルタイム通信を送信している状態で、リアルタイム通信 1 から 6 までを約 30 秒ごとに順に追加していく方法で行う。

#### (2) 実験結果と考察

実験 1 の計測結果について、縦軸に EDF スケジューリングにおける遅延時間、横軸にリアルタイム通信 1 の開始時刻を 0 とした経過時間を取ったグラフを図 3 に示す。リアルタイム通信 1 から 6 までの最大遅延時間は、それぞれ 3.00ms, 4.98ms, 6.39ms, 8.67ms, 14.44ms, 19.28ms となり、全ての通信で表 1 で示した通信のデッドライン時間を下回る遅延時間でパケットスケジューリングを行うことができた。これより、VMM の EDF パケットスケジューリングが送信パケットのデッドライン保証に対して有効であることを示した。

リアルタイム通信の遅延の特徴として、デッドライン時間からノンリアルタイム通信を優先するパラメータとして与えた 2ms を引いた遅延時間の近くで推移している部分と、遅延が増大し鋭い山型となっている部分の二つの部分がある。鋭い山型となっている原因としては、リアルタイム通信のバースト的な転送により、キューイングされているパケット数が増大したことにより、キューの最後尾に近いパケットの送信待ち時間が長くなったことが挙げられる。パケットの遅延時間が大きくなるにつれ、EDF によるパケットスケジューリングの優先度も上昇することから、デッドラインミスが発生する前に転送を行うことができたと考える。この結果から、EDF パケットスケジューリングにより各パケットのキューイング遅延時間に応じたパケット送信を行えたことを確認した。

### 4.2.2 VM 間通信の転送遅延時間の評価

提案手法が VM 間の通信のリアルタイム性保証に対して有効であることを示すため、異なる物理マシン上で動作する VM 間の通信の転送遅延時間について評価を行った。

#### (1) 評価方法

評価に用いた VMM・ネットワーク構成を図 4 に示す。VM の構成は、各物理マシンに RTvNIC を持つ VM を配置し、この VM 間でリアルタイム通信を行う。物理ネットワークには一台の L2 スイッチを配置し、VMM においては EDF スケジューリングによる QoS を実施する。

この構成を用い、End-to-End の転送遅延時間として、送信側の RTvNIC がパケットを送信した時から、受信側の RTvNIC に到達するまでの転送遅延時間を計測する実験 2 を行う。この実験では、表 1 で示すリアルタイム通信 1 から 3 とノンリアルタイム通信を利用して計測する。この評価では、計測フローに対して RTvNIC システムのリアルタイム性保証を適用する場合としない場合で比較する。

表 2 実験 2 : VM 間の End-to-End の通信の遅延時間 (RTvNIC あり)

	平均遅延時間 (ms)	最大遅延時間 (ms)	最小遅延時間 (ms)
リアルタイム通信 1(DL:3ms)	2.8	3.0	2.6
リアルタイム通信 2(DL:5ms)	4.7	4.9	4.4
リアルタイム通信 3(DL:8ms)	7.5	7.7	7.2

表 3 実験 2 : VM 間の End-to-End の通信の遅延時間 (RTvNIC なし)

	平均遅延時間 (ms)	最大遅延時間 (ms)	最小遅延時間 (ms)
リアルタイム通信 1(DL:3ms)	14.2	15.0	3.5
リアルタイム通信 2(DL:5ms)	14.7	14.9	6.2
リアルタイム通信 3(DL:8ms)	13.3	15.7	4.2

本実験では、NTP(Network Time Protocol)により二台の物理マシン間で時刻同期を行い、リアルタイム通信のパケットの転送遅延時間を計測した。二台の物理マシン間では、NTP による時刻同期専用の LAN を構築した。本実験を実施中においても、時刻の誤差は最大約 300ns であったため、本評価で用いるには十分な精度である。

#### (2) 実験結果と考察

実験 2 の計測結果を表 2, 表 3 に示す。実験 2 では、最大転送時間について、デッドラインをそれぞれ 3ms, 5ms, 8ms を設定したリアルタイム通信 1 から 3 に対して、RTvNIC を適用しない場合では 15.0ms, 14.9ms, 15.7ms とデッドライン時間を超えているのに対して、RTvNIC を適用した場合には、3.0ms, 4.9ms, 6.7ms となり、最大転送時間をデッドライン時間以内に抑えることができた。このとき、RTvNIC を適用しない場合のパケットスケジューリングである FIFO アルゴリズムではデッドラインを超えるスケジューリングが存在していたのに対し、RTvNIC を適用した場合の EDF では、各パケットのデッドラインを考慮したスケジューリングを行ったことにより、デッドラインを守るパケットスケジューリングとなっている。

実験 2 の RTvNIC を適用した場合のパケットは、デッドライン時間を管理の MPLS ラベル (4byte) を付加したため、RTvNIC を適用しないパケットと比較して 4byte フレームサイズが大きい。このため、パケット長が長くなったことにより物理 NIC のパケット送信処理時間は増加しているはずであるが、1Gbps の帯域を持つ物理 NIC が 4byte 分のパケットを送信するのにかかる時間は、 $4\text{byte} / 1\text{Gbps} = 0.032\mu\text{s}$  であるため、最大遅延時間増加のオーバーヘッドは小さい。

#### 4.2.3 RTvNIC システムのオーバーヘッドの評価

RTvNIC システムを追加したことによるオーバーヘッドを検証するため実験 3 を行った。本実験では、パケット遅延の増加要因となる、パケットへの MPLS ラベルの挿入/除去と EDF パケットスケジューリングの処理時間について計測を行った。計測の結果、MPLS ラベルの挿入と除去の最大処理時間はそれぞれ 80us, 72us で、EDF パケットスケジューリングの最大処理時間は 160us であった。この

結果から RTvNIC システムによりリ秒オーダのリアルタイム性の保証を実現するためには、このパケットマーキングと EDF パケットスケジューリングの処理時間は十分に小さいと言える。

#### 4.3 評価結果のまとめ

実験 1 では、VMM の EDF パケットスケジューリングにおいて、異なるデッドラインを持つ 6 つのリアルタイム通信を転送した結果、それぞれのデッドライン以内の送信を確認した。また、実験 2 では、別物理マシンの VM 間で End-to-End のパケット転送において、EDF のパケットスケジューリングによる異なるデッドラインを持つ三つのフローの転送遅延時間を計測した。この結果、EDF パケットスケジューリングの End-to-End の遅延に対する有効性を示した。実験 3 では、RTvNIC システムのオーバーヘッドを検証し、保証するデッドライン時間に対して小さいオーバーヘッドでリアルタイム通信制御を実現したことを示した。以上より、提案している RTvNIC システムによる VMM のパケット制御で、VM 間の End-to-End のリアルタイム通信の保証を実現した。

### 5. おわりに

本論文では、VMM のリアルタイム通信制御により、異なる物理マシン上で動作する VM 間のリアルタイム通信を実現する通信基盤である RTvNIC システムを提案した。また、パケットスケジューラとして EDF アルゴリズムを用いたスケジューラを実装し、RTvNIC システムによるリアルタイム通信の実験を行った。実験の結果、ノンリアルタイムパケットで輻輳させたネットワーク中でも、RTvNIC システムによりリアルタイム性を確保したパケットが優先的に送信され、設定したデッドライン時間以内に到達することを確認した。この結果より、RTvNIC システムによるパケットスケジューリングによるパケット遅延時間の低減効果を示した。

今後は、ネットワークの QoS 制御を含めたリアルタイム通信保証の実現、また、リアルタイム通信のフロー数や仮想マシン数のスケーラビリティの検証など実運用を想定し

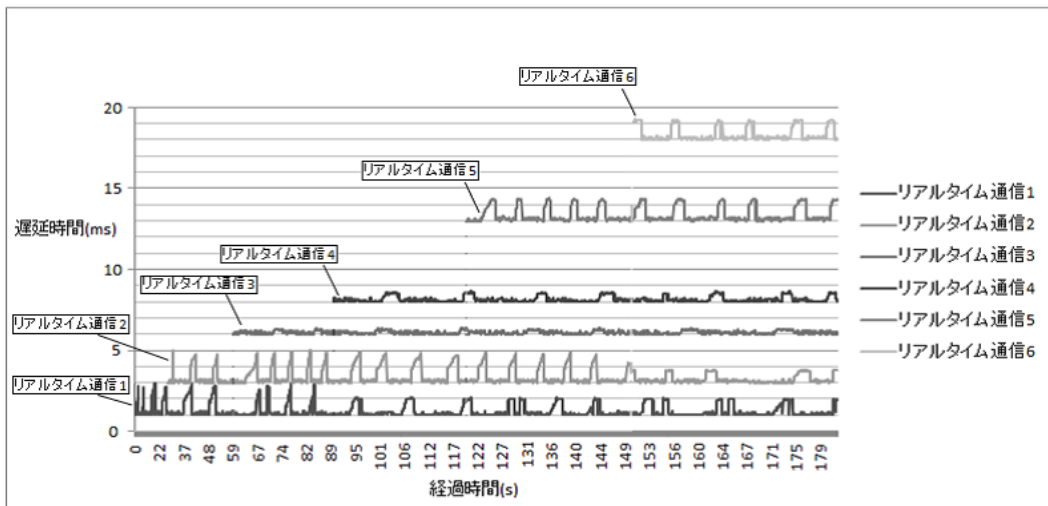


図 3 実験 1 : EDF パケットスケジューリングにおける各パケットの遅延時間

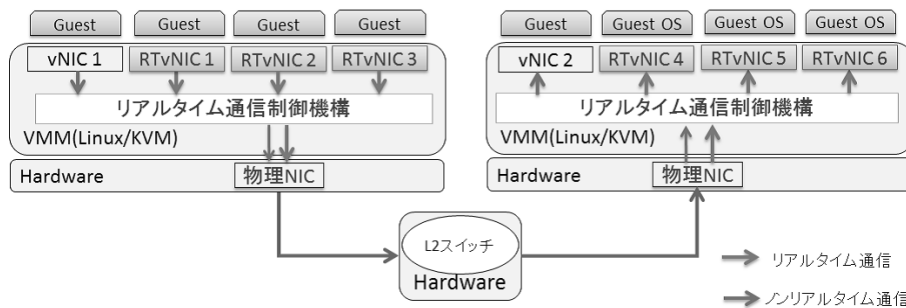


図 4 実験 2 : 評価用ネットワーク構成

た環境で提案手法の適用可能な範囲を明らかにする評価、分析を行っていく。

### 参考文献

- [1] Cheng L., Wang C., and Di S.: Defeating network jitter for virtual machines, Proc. *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference*, pp.65–72, (2011).
- [2] 太田貴也, DanielSangorin, 本田晋也, 高田広章: 組込みマルチコア向け仮想化環境における性能低下抑止手法, 情報処理学会研究報告. *EMB, 組込みシステム 2012-EMB-27(12)*, pp. 1–8, (2012).
- [3] Blake S., Black D., Carlson M., and et al.: An architecture for differentiated services, *IETF RFC Standard 2475*, (1998).
- [4] Zhang L., Deering S., Estrin D., and et al.: RSVP: a new resource reservation protocol, *Network, IEEE*, Vol. 7, No. 5, pp. 8–18, (1993).
- [5] Braden R., Clark D., and Shekner S.: Integrated services in the internet architecture: an overview, *RFC 1633*, (1994).
- [6] Guo Z. and Hao Q.: Optimization of kvm network based on cpu affinity on multi-cores, Proc. *Information Technology, Computer Engineering and Management Sciences (ICM), 2011 International Conference*, Vol. 4, pp. 347–351, (2011).
- [7] Liu J.: Evaluating standard-based self-virtualizing devices: A performance study on 10 gbe nics with sr-ioV support, Proc. *Parallel Distributed Processing*

- (*IPDPS*), *2010 IEEE International Symposium*, pp.1–12, (2010).
- [8] Tang Y. and Li J.: Von/kvm: A high performance virtual overlay network integrated with kvm, Proc. *Apperceiving Computing and Intelligence Analysis (ICACIA), 2010 International Conference*, pp. 129–132, (2010).
- [9] McKeown N., Anderson T., Balakrishnan H., and et al.: Openflow: Enabling innovation in campus networks, *SIGCOMM Comput. Commun. Rev.*, Vol. 38, No. 2, pp. 69–74, (2008).
- [10] Barham P., Dragovic B., Fraser K., and et al.: Xen and the art of virtualization, Proc. *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles (SOSP '03)*, pp. 164–177, (2003).
- [11] Liu C.L. and Layland J.W.: Scheduling algorithms for multiprogramming in a hard-real-time environment, *J. ACM*, Vol. 20, No. 1, pp. 46–61, (1973).
- [12] Paul Schnackenburg: Virtualization: What's new with Hyper-V, Microsoft TechNet, available from <http://technet.microsoft.com/en-us/magazine/dn235778.aspx> (accessed 2014-07-28)
- [13] VMware: vSphere 5.5 document center, available from <http://pubs.vmware.com/vsphere-55/index.jsp#com.vmware.vsphere.networking.doc/GUID-35B40B0B-0C13-43B2-BC85-18C9C91BE2D4.html> (accessed 2014-07-28)
- [14] Citrix: Xen: Virtual network QoS settings, *XenServer Administrator's Guide*, available from <http://docs.vmd.citrix.com/XenServer/4.0.1/reference/reference.html>

(accessed 2014-07-28)

- [15] Sharma S., Katramatos D., Yu D., and Shi L.: Design and Implementation of an Intelligent End-to-End Network QoS System Proc. the International Conference on High Performance Computing, Networking, Storage and Analysis (SC '12), Article 68, (2012).