

# OpenFlow と協調する仮想マシン環境におけるリアルタイム通信基盤の設計

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

**概要:** 音声通話や動画像のストリーミングではネットワークのリアルタイム性を確保する QoS が必要である。さらに、近年研究の進んでいるリアルタイム仮想マシンからのリアルタイム通信を行いたいという要求がある。しかし、VM 間での EndtoEnd のリアルタイム性の保証について課題がある。本論文では、VMM 層にリアルタイム通信機能を付加し、ネットワーク制御には OpenFlow を利用することにより、異なる計算機上で動作する VM 間のリアルタイム通信基盤を提案する。VMM 層でリアルタイム通信を管理することで、ゲスト OS がリアルタイム通信を管理する必要がなくなるため、ゲスト OS への修正を最小限に抑えられる。さらに、動的にネットワーク構成を変更可能である OpenFlow ネットワークを用い、帯域制御と優先制御を VMM と OpenFlow で協調して行うことで、ネットワーク全体のリアルタイム性の保証を実現する。RTvNIC システムの基礎評価として、パケットへのデッドライン時間のマーキングと EDF スケジューリングを用いることで、既存手法と比較し、最悪転送時間を約 92ms 削減できたことを確認した。

## Design of a Realtime Network in Virtual Machine Environment with OpenFlow

KENICHI SUZUKI<sup>1</sup> HIROSHI MIYATA<sup>1</sup> MIKIKO SATO<sup>1</sup> MITARO NAMIKI<sup>1</sup>

### 1. はじめに

近年、ネットワークは音声通信、動画像のストリーミング配信、企業の基幹ネットワーク、VPN などといった多種多様な利用目的を持つようになったことから、ネットワーク QoS(Quality of Service) の重要性が増している。さらに、産業用ネットワークでは、工場内で動く製造装置や組立装置などの制御のため、厳しいリアルタイム性を達成することが求められる。代表的な QoS 技術として DiffServ[1] と RSVP/IntServ[2][3] が挙げられるが、Diffserv は複数のフローをまとめたクラスに対して QoS 制御を行うため、EndtoEnd の通信のリアルタイム性を確保することが困難であり、RSVP/IntServ はフローごとに QoS 制御を行うためスケラビリティの面に大きな課題がある。

一方、クラウドなどに見られるように仮想マシン (VM) 技術も大きく進展している。従来はデータセンタ向けの

物理リソースの有効利用やオンデマンドでの生成・削除といった利点が重要視されていたが、最近では VM を管理する仮想マシンモニタ (VMM) 層に、セキュリティ機能やゲスト OS に依存しない画面表示機能を取り入れた BitVisor[5] や、組み込み向けの VMM の研究 [6] など VMM に新しい性質・機能を加える研究が行われている。この研究の中で、VM と VMM にリアルタイム性を持たせ、VM 上でリアルタイムアプリケーションを動作させる研究が進んでいる [4]。そして、その応用としてリアルタイム VM の間でネットワークを介したリアルタイム通信の実現が求められている。

VM 環境においてリアルタイムネットワークを実現する場合、従来のネットワーク仮想化技術では次のような課題がある。まず、ネットワークの構成や QoS を動的に変更することが難しいため、VM 環境で頻繁に起こりうる VM の生成や消滅による通信ノード数の変化へ追従できない。これに対しては、ネットワーク構成を動的に変更できる SDN(Software Defined Network) 技術の一つであ

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

る OpenFlow を用いることで解決できる可能性がある。また、ネットワークの QoS 制御においては上述の DiffServ と RSVP/IntServ の課題があるが、VMM の仮想ネットワークインターフェース (仮想 NIC) でゲスト OS の送受信するパケットを全て制御することを利用し、この仮想 NIC に QoS 補助機能を追加し、ネットワークと連携して QoS 制御を行うことで解決できる可能性がある。そこで本研究では、ネットワークの QoS 制御には DiffServ のアーキテクチャを用いてスケラビリティを確保しつつ、VMM と OpenFlow ネットワークの協調動作によるフロー単位の QoS 制御を行い、VM 間での EndtoEnd のリアルタイム性を保証することを目指す。

## 2. RTvNIC システムの概要

本章では、本論文で提案する VM 間のリアルタイム通信を実現する通信基盤である RTvNIC システムの技術課題と目標、およびシステムの概要を述べる。

### 2.1 課題と目標

DiffServ では、コアルータにおいて同じクラスのフローは同一ポリシーで転送されるため、フローごとの QoS 制御ができない。このため、同一クラスの他フローの影響を受け、自フローのリアルタイム性が確保できなくなる課題がある。また、VM 環境を対象としているため、頻繁に起こりうる VM の生成や消滅による通信ノード数の変化へ対応しなければならない。さらに、リアルタイム通信を必要としない VM のネットワーク通信も求められる。

そこで本論文では、VMM の仮想 NIC に対してリアルタイム通信制御機能を付加した“リアルタイム仮想 NIC(RTvNIC)”と OpenFlow を用いて、異なる計算機上で動作する VM 間で EndtoEnd のリアルタイム通信を実現するリアルタイム通信の実行基盤「RTvNIC システム」を提案する。本 RTvNIC システムでは、次の三つの機能を提供し、上記に示した VM 環境におけるリアルタイム通信の課題を解決することを目標とする。

- VM 間の EndtoEnd の通信のリアルタイム性の保証
- 動的な VM 構成の変更への対処
- リアルタイム性を持たない仮想 NIC と RTvNIC の両方の提供

EndtoEnd のリアルタイム性の保証では、DiffServ と RSVP/IntServ に対する課題を解決するため VMM と OpenFlow によりネットワークの QoS 制御を行う。VMM 層ではパケットがデッドライン時間以内に転送されているか監視し、その情報を OpenFlow コントローラへフィードバックする。そして、デッドラインをミスしたフローに対して QoS の再設定を行い、遅延時間を短縮することで EndtoEnd のリアルタイム性を保証する。VM 構成の変化については、OpenFlow の動的な構成変更が容易である

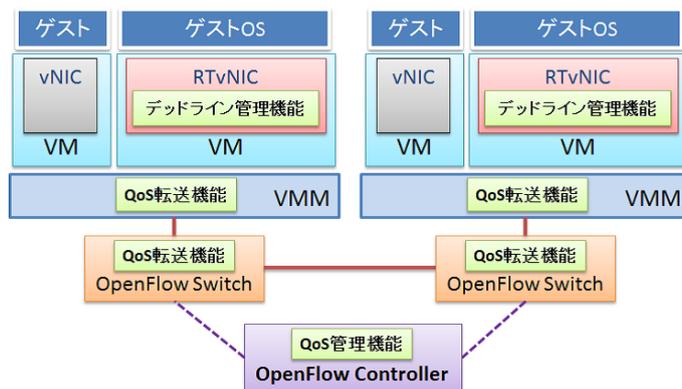


図 1 RTvNIC システムの全体構成

という特徴を利用して対応する。また、RTvNIC システムのリアルタイム制御処理を VMM 層で隠蔽することで、RTvNIC システムを利用しないゲスト OS に対しては、通常の仮想 NIC を提供する。

本論文では、リアルタイム性保証手法の設計、VMM と OpenFlow 連携機能とそのインターフェースを示し、パケットへのマーキングとリアルタイムパケットスケジューリングの試作を行った。そして、基礎評価を行い、RTvNIC システムのデッドライン時間のパケットマーキングに基づく EDF スケジューリングにより、リアルタイム性保証に必要な最悪転送遅延時間の制御効果を確認した。

### 2.2 全体構成

RTvNIC システムの全体構成を図 1 に示す。RTvNIC システムのリアルタイム通信制御は、大きく以下の三つの機能に分類される。

- QoS 管理機能
- QoS 転送機能
- デッドライン管理機能

QoS 管理機能は、OpenFlow コントローラが持つ。QoS 管理機能ではリアルタイム通信路の設定や、帯域や経路などのネットワーク資源の予約などを行う。VMM と OpenFlow フロースイッチは QoS 転送機能を持ち、パケットに対するマーキングやパケットスケジューリングによって QoS 制御を行う。RTvNIC はデッドライン管理機能を追加し、転送されたパケットが指定されたデッドライン時間以内に到達したかを確認する。

この三つの機能を連携させることでリアルタイム性の保証を実現する。まず、VMM はゲスト OS からのリアルタイム通信の開始要求を受け取ると OpenFlow コントローラに対して通知する。OpenFlow コントローラでは、VMM からリアルタイム通信の開始要求を受け取ると、通信経路を決定し、経路上の OpenFlow スイッチに対してネットワーク資源の予約や QoS 設定を行う。OpenFlow スイッチでは、指定された QoS 転送を行う。また、VMM でデッドラインミスが発生した場合も OpenFlow コントローラへ

の通知を行う。OpenFlow コントローラでは、デッドラインミスを起こしたフローに設定された帯域やスケジューリングのパラメータなどの再設定を行うことでデッドラインミスの再発を防止する。さらに、デッドラインミス通知はデッドラインミスを起こしたパケットの送信元 VMM へも通知され、パケットを送信したゲスト OS が指定したデッドラインミスハンドラを起動することができる。

### 3. RTvNIC システムの設計

本章では、RTvNIC システムの設計方針と、VMM と OpenFlow の連携のインターフェースとリアルタイム性の保証手法について述べる。

#### 3.1 設計方針

RTvNIC システムの設計方針を述べる。本研究の目標である EndtoEnd のリアルタイム性を確保するため、宛先と送信元 IP アドレスの組でフローを区別し、そのフローごとにリアルタイム性を確保する。これにより、同じ DSCP 値を持つパケットを全て同様に扱う Diffserv では実現できない、各フローごとのリアルタイム性を確保する。IP アドレスは仮想 NIC と一対一で対応しているため、二つの仮想 NIC 間の通信のリアルタイム性を確保することと同じである。この仮想 NIC をリアルタイム仮想 NIC(RTvNIC) と定義し、リアルタイム性が確保された RTvNIC 間の通信路をリアルタイム通信路と定義する。RTvNIC システムでは IP/Ethernet プロトコルを用いたパケットを対象とする。

音声通信では、一定の割合でデッドラインミスが発生することが許容されるソフトリアルタイムであるが、制御通信などはデッドラインミスが致命的な損害が生じるハードリアルタイムである。このようにアプリケーションにより求めるリアルタイム性が異なるため、通信の利用目的に合わせ保証するリアルタイム性を選択できるようにする。パケットごとの転送遅延を測定するために、VMM 間の時刻同期が必要となるが、NTP,IEEE1588 PTP, GPS など既存技術の利用を想定し、本研究では対象としない。目的とする精度を持った時刻同期を用いる必要がある。

また、RTvNIC システムを用いない 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。

#### 3.2 リアルタイム通信路の設計

RTvNIC システムにおけるリアルタイム通信は、「リアルタイム通信路」と呼ばれる論理的な通信路によって実現する。リアルタイム通信路の両端はゲスト OS のリアルタイム向け仮想 NIC(RTvNIC) であり、送受信側の RTvNIC にそれぞれ割り当てられた IP アドレスと受信側のポート番号の組により識別される。RTvNIC システムでは、このリ

アルタイム通信路をリアルタイム性を確保するフローの単位として扱い、この通信路上で転送される片方向のパケットに対して EndtoEnd のリアルタイム性を保証する。ゲスト OS は、このリアルタイム通信路の設定を VMM へ要求することで通信のリアルタイム性を確保する事ができる。

RTvNIC システムでは、リアルタイム通信路の保証するリアルタイム性として、ハードリアルタイムとソフトリアルタイムに対応するために、ゲスト OS 上のアプリケーションが選択できるようにする。

##### (1) 資源予約型

最低帯域と最悪遅延時間が予測可能な通信路を提供する。モーション制御などデッドライン時間以内にパケットが到着しないと、致命的なダメージが発生する、もしくはパケットの価値が無くなるようなハード(ファーム)リアルタイムなパケットを対象としている。資源予約型のリアルタイム通信路の設定には、最低帯域、最大バースト長、デッドライン時間、デッドラインミス処理を指定する。この情報を元にネットワーク上の帯域に空きがあるか、最悪遅延時間の理論値は要求されたデッドライン時間以内であるかを確認した上で、ネットワーク資源を確保する。ネットワーク資源に余裕がない場合、資源予約型リアルタイム通信路は設定できない。

##### (2) デッドライン保証型

パケット転送のデッドライン時間を保証する通信路を提供する。動画や音声の配信など一定量のデッドラインミスやパケットロスが許容されるソフトリアルタイムなパケットを対象とし、最悪遅延時間を保証するのではなく、できるだけ多くのパケットがデッドライン時間以内に到着することを目標としている。デッドライン保証型の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。転送遅延が出来るだけ少なくなるようにスケジューリングされるが、デッドラインに対してパケット到着時間に余裕のない通信路は、他の余裕がある通信路よりも優先的にパケットスケジューリングされる。

### 3.3 RTvNIC の設計

本研究で用いる VMM は、既存 VMM の仮想 NIC に対して、パケットへのマーキング機能とパケットの転送遅延測定機能を追加したものである。本節では、RTvNIC へ追加したの設計について述べる。

#### 3.3.1 マーキング機能

ネットワーク内においてパケットを処理する際に、リアルタイムパケットとベストエフォートパケットを区別する必要がある。リアルタイム通信路の定義からは、宛先・送信元 IP アドレスを確認することで区別可能である。

しかし、ネットワーク上の全てのスイッチがすべてのリアルタイム通信路情報を持ち、そのつど IP アドレスを見て判定する方法をとると、スイッチが保持するデータ量が増加し、一パケットの処理に掛かる負荷が大きくなる。そこで、RTvNIC システムでは IP ヘッダの TypeOfService(ToS) フィールドにリアルタイム通信路ごとに固有の値を識別子として書き込み、これを利用することでこの問題を解決する。本設計では、特に、このリアルタイム通信路ごとに固有の識別子を DiffServ(Differentiated Services) の DSCP(Differentiated Services Code Point) 値と対応付けることにより、ネットワークを転送する際に、既存の DiffServ による QoS を流用できる形にした。送信側 VM から送信されたパケットの IP ヘッダ内の送信元 IP アドレス、宛先 IP アドレスを参照し、現在設定されているリアルタイム通信路の送信元・宛先 IP アドレスの組と一致していれば、IP ヘッダの DSCP フィールドにリアルタイム通信路ごとに固有に割り当てられた DSCP 値を書き込む。

### 3.3.2 転送遅延時間の管理

ここでは、パケットの転送遅延測定機能について述べる。リアルタイムパケットが送信元 VMM から宛先 VMM に届くまでの転送遅延時間が、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ことで、EndtoEnd での通信のリアルタイム性が確保されていることを確認することができる。このパケットのデッドライン判定を行うためには、各パケットの送信側 VMM での送信時刻を宛先 VMM へ通知する必要がある。パケットのデッドライン時間を送信パケットと別のパケットで通知する方法では、通知パケットの到着遅延やロスによりデッドライン時間判定ができなくなる可能性がある。このため、デッドライン時間は送信パケットに付加して通知することとする。

パケットに付加するデッドライン時間は相対デッドライン時間と絶対デッドライン時間が考えられる。相対デッドライン時間を付加する場合は、送信パケットがネットワーク上の各スイッチを通るたびに、付加された相対デッドライン値からスイッチ内に滞在していた時間を減算し、値が 0 以下になったときデッドラインミスをしたと判定する。この方法では、送信元と宛先の時刻同期が必要ない点で優れているが、各遅延時間を減算して間接的にデッドライン時間を求めているため、実際のパケット転送時間とずれが生じる可能性がある。一方、絶対デッドライン時間を付加する場合は、送信側・受信側の時刻を同期し、同期した時計の時刻とパケットに付加された時刻を比較することにより、デッドラインを判定する方法である。この方法は実際のパケット転送時間を直接計測することができるが、計測の精度は時刻同期の精度に依存する。本研究では、相対デッドライン方式では計測誤差の発生要因が多く誤差の見積りがしにくい、絶対デッドライン方式では時刻同期の

表 1 リアルタイム通信路確保：要求メッセージ

Name	Description
dst_ip	宛先 IP アドレス
src_ip	送信元 IP アドレス
dst_port	宛先ポート番号 (オプション)
min_rate	最低帯域
max_burstlen	最大バースト長
deadline_time	パケット転送のデッドライン時間
deadline_handler	デッドラインミスハンドラ
rtpath_type	確保するリアルタイム通信路の型

精度のみに依存するために絶対デッドラインをパケットに付加し、同期した時刻によりデッドライン判定を行う。

パケットへのデッドライン時間の付加は、MPLS(Multi-Protocol Label Switching) のラベリングを流用する。パケットの L2 ヘッダと L3 ヘッダの間に MPLS ヘッダを挿入するフレームモードを利用する。送信側 VMM において、リアルタイムパケットに対して MPLS ラベルの挿入と MPLS ヘッダの Label フィールド (20bit) に絶対デッドライン時間を書き込む。なお、RTvNIC システムで付加した MPLS ヘッダはデッドライン時間を送信側 VMM から受信側 VMM へ通知するためのもので、この値によるラベルスイッチングは行わない。

MPLS ヘッダは 32bit(4byte) の長さであるため、リアルタイムパケットのパケット長はすべて元のパケットサイズより 4byte 大きくなる。RTvNIC システムの MPLS ラベリングは VMM によって、ゲスト OS に対して隠蔽されているため、ゲスト OS が送信する Ethernet フレームに MPLS ラベルを付加すると、Ethernet の最大フレームサイズである 1522 バイトを最大 4byte 超過する可能性がある。本システムでは、VMM が Ethernet フレームの最大フレームサイズを超過するパケットを受け取った場合、VMM がゲスト OS に対して、ICMP プロトコルの PATH MTU Discovery(type 3:code 4) を送ることで、ゲスト OS の送信する最大フレームサイズを 1518byte に設定する。

### 3.3.3 ゲスト OS と VMM 間の協調

ここでは、ゲスト OS が VMM に対してリアルタイム通信開始要求を行うためのインターフェースについて述べる。リアルタイム通信の開始にあたり、ゲスト OS は表 1 に示すパラメータを持つ、リアルタイム通信路確保要求メッセージを VMM に対して発行する。VMM は OpenFlow ネットワークにリアルタイム通信の開始の可否について問い合わせを行った後、その結果を表 2 に示す応答メッセージとしてゲスト OS に通知する。

リアルタイム通信開始後、送信したパケットがデッドラインミスを起こした場合、VMM からゲスト OS に対して表 3 のデッドラインミス通知メッセージが発行される。このメッセージによりゲスト OS はデッドラインミス処理を行う事ができる。

表 2 リアルタイム通信路確保：応答メッセージ

Name	Description
result	リアルタイム通信路の確保結果
rtpath_id	確保したリアルタイム通信路の識別番号

表 3 デッドラインミス通知メッセージ

Name	Description
rtpath_id	リアルタイム通信路の識別番号
exceed_time	デッドラインを超過した時間
ip_id	デッドラインミスをしたパケットの IP ヘッダの Identification 値

### 3.4 リアルタイム性の保証

VMM からリアルタイム通信路設定要求メッセージを受け取った OpenFlow コントローラは、要求されたリアルタイム性を保証できるだけネットワーク資源に余裕があるかを確認し、余裕が無い場合はリアルタイム通信路の設定が失敗したことを VMM へ通知する。余裕が無いときは要求を拒否することにより、リアルタイム性を確保出来ないフローが生じないようにする。

また、VMM からデッドラインミス発生通知メッセージを受け取った OpenFlow コントローラは、デッドラインミスをした対象フローに対して、QoS の再確保を行う。資源予約型リアルタイム通信路を実現するためのパケットスケジューリングアルゴリズムとして想定している WFQ(Weighted Fair Queuing) スケジューリングでは、EndtoEnd 間におけるあるフローの最大遅延時間は一般に式 (1) で表される。

$$Delay = \frac{(b - M)}{R} \times \frac{(p - R)}{(p - r)} + \frac{(M + C_{tot})}{R} + D_{tot} \quad (1)$$

ただし、

r: Token Bucket rate

b: Token Bucket Size

p: Peak Rate

M: Maximum Packet Size

R: Service Rate

$C_{tot}$ : Total Rate-Dependent Error Term

$D_{tot}$ : Total Rate-Independent Error Term

$0 \leq r \leq R \leq p, 0 \leq M \leq b, 0 \leq C_{tot}, 0 \leq D_{tot}$

これより、WFQ を用いているフローがデッドラインミスをした場合、そのフローの Service Rate(式 1 の R) を増加させることで遅延を削減し、デッドラインミスを回避する。

また、デッドライン保証型リアルタイム通信路を実現するためのパケットスケジューリングアルゴリズムとして、EDF アルゴリズムを用いる。各リアルタイム通信路で定義されたデッドライン時間と送信キュー内の滞留時間を元

に EDF(Earliest Deadline First) アルゴリズムを用いてパケットスケジューリングを行う。パケットのデッドライン時間を経路上のスイッチ SW1 から SWn に配分して、パケットのキュー内滞在時間でスケジューリングを行う。この EDF(Earliest Deadline First) を用いて、パケット到着のデッドライン時間を保証する。

### 3.5 VMM と OpenFlow の連携

本節では、ネットワークのリアルタイム性の保証を実現するための OpenFlow と VMM の連携のインターフェースについて述べる。

#### 3.5.1 VMM と OpenFlow の連携内容

OpenFlow と VMM はリアルタイム通信路の確保時とデッドラインミス時にやりとりを行う。リアルタイム通信路の確保では、まず、VMM が OpenFlow コントローラに対してデッドライン時間など要求するリアルタイム性を通知する。OpenFlow コントローラではその要求を満たすリアルタイム通信路を確保可能か判定し、可能であれば経路上のスイッチに帯域予約を設定する。不可能であれば、確保失敗通知を送信側 VMM へ返す。

デッドラインミス時には、デッドラインミス判定を行っている VMM が OpenFlow コントローラとデッドラインミスをしたパケットの送信 VMM へデッドラインミスの発生を通知する。OpenFlow コントローラでは、一度デッドラインミスが発生したネットワークの QoS 設定のままでは、デッドラインミスが頻発する可能性があるため、3.4 で述べたリアルタイム性保証より、通信路の転送遅延を削減しデッドラインミスが再発しないように、ネットワークの QoS を変更を行う。

#### 3.5.2 VMM と OpenFlow 間で連携するためのインターフェース

ここでは、VMM と OpenFlow が連携し、ネットワーク資源の確保とデッドラインミス対処を行うためのインターフェースを述べる。まず、ゲスト OS からリアルタイム通信路確保要求メッセージを受け取った VMM は、表 4 のリアルタイム通信路設定要求メッセージを OpenFlow コントローラに対して通知する。OpenFlow コントローラは、要求されたリアルタイム通信路が設定可能かを確認した後、その結果を表 5 の応答メッセージとして通知する。また、受信側 VMM で行う受信パケットのデッドラインミス判定により、デッドラインミスの発生を受け、受信側 VMM から OpenFlow コントローラと送信側 VMM に対して表 6 のデッドライン発生通知メッセージを発行する。これらのメッセージは「制御通信型」として、最高優先度で転送される。

表 4 リアルタイム通信路設定：要求メッセージ

Name	Description
qos_param	ゲスト OS からのリアルタイム通信路確保要求メッセージの内容

表 5 リアルタイム通信路設定：応答メッセージ

Name	Description
result	リアルタイム通信路の設定結果
rtpath_id	設定したリアルタイム通信路の識別番号

表 6 デッドラインミス発生通知メッセージ

Name	Description
rtpath_id	リアルタイム通信路の識別番号
exceed_time	デッドラインを超過した時間
ip_id	デッドラインミスをしたパケットの IP ヘッダの Identification 値

## 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。RTvNIC システムの基礎評価として、パケットスケジューリングに EDF を用いて QoS を確保したネットワークを実装し、パケットのリアルタイム性の評価を行った。

### 4.1 実装

RTvNIC システムの実装には、仮想マシンモニタとして Linux のカーネルモジュールである KVM/qemu-kvm(1.2.0) を利用する。この内、qemu-kvm のエミュレートする仮想 NIC である NE2000\_PCI にパケットへの MPLS ラベルの挿入と DSCP 値の付加を実装し、これを RTvNIC とした。OpenFlow コントローラには trema を利用した。OpenFlow スイッチには仮想スイッチソフトウェアの Open vSwitch(1.9.0) を利用し、PC ルータとして利用した。仮想マシンモニタと OpenFlow スイッチにおいて、EDF パケットスケジューリングをを実現するために、Linux カーネルの機能である Traffic Control を改変し、EDF スケジューラを実装した。

### 4.2 評価方法

本論文では基礎評価として、リアルタイム性の保証に重要な最悪転送時間を確認するために、RTvNIC システムで想定しているデッドライン保証型の EDF スケジューリングを用いた場合の評価を行った。評価に用いた VMM・ネットワーク構成を図 2 に示す。本実装で用いた物理 NIC の最大帯域は 100Mbps のものを用いた。本実装では複数の VMM 間の時刻同期を実装しなかったため、VM が送信したパケットが自 VMM のもう一方の VM に戻ってくる構成を取ること、時刻同期の実装なしにデッドライン判定を行った。VM の構成は、RTvNIC を一つ持った VM を二つ配置し、この VM 間で OpenFlow ネットワークを

介してリアルタイム通信を行う。さらに、RTvNIC を持たない VM を一つ配置し、ネットワークを輻輳状態にするためのベストエフォート (BE) パケットを送信する VM とした。OpenFlow ネットワークには、二台の OpenFlow スイッチを配置し、VMM と OpenFlow スイッチでそれぞれで EDF スケジューリングによる QoS を実施する。QoS はリアルタイムパケットを優先的に送信する優先度キューを用いる。

評価として、送信側 VM から受信側 VM に対して計測用 UDP パケット送信を行い、そのパケット転送遅延時間を測定する。UDP パケットは、RTvNIC システムによりリアルタイム性を確保したパケットである。計測用 UDP パケットのフレームサイズは 1500byte である。

本評価ではパケットのデッドライン時間  $D=10\text{ms}$  とし、経路上の二つのスイッチスイッチに対して、等しく  $5\text{ms}$  ずつデッドライン時間  $D_i(i=1,2)$  を配分する。各スイッチは、パケットの送信キュー内滞留時間を  $5\text{ms}$  以内に送信するように、デッドラインに近いパケットから送信を行う。

次に実験内容を示す。ネットワークを輻輳させた場合とさせない場合、計測用パケットに対して RTvNIC システムのリアルタイム性保証を適用した場合としていない場合を組み合わせた 4 通りの実験を行った。

実験 (1) 計測用パケットに対してデッドライン時間を設定しない、かつ、ネットワークの帯域に余裕がある場合計測用パケットに対してデッドライン時間を設定をせず、RTvNIC システムの制御対象外として計測を行う。各スイッチの EDF スケジューリングにおいて計測用パケットは、ベストエフォートパケットとしてデッドライン時間が無限大でスケジューリングされる。ネットワークには計測用パケット以外のパケットは送信しない。

実験 (2) 計測用パケットに対してデッドライン時間を設定する、かつ、ネットワークの帯域に余裕がある場合計測用パケットに対して、 $10\text{ms}$  のデッドライン時間を設定し、RTvNIC システムの制御対象として計測を行う。各スイッチの EDF スケジューリングにおいて計測用パケットは、 $5\text{ms}$  のデッドライン時間を持つリアルタイムパケットとしてスケジューリングされる。ネットワークには計測用パケット以外のパケットは送信しない。

実験 (3) 計測用パケットに対してデッドライン時間を設定しない、かつ、ネットワークを輻輳させた場合実験 (1) と同様に、計測用パケットに対してデッドライン時間を設定をせずに計測を行う。ただし、ネットワークには RTvNIC を持たない VM から、物理 NIC の最大帯域一杯の  $100\text{Mbps}$  のベストエフォートパケット (UDP パケット) が送信されている状態で計測を行う。実験 (4) 計測用パケットに対してデッドライン時間を設定

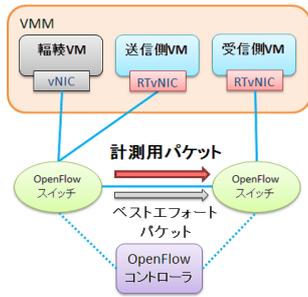


図 2 評価ネットワーク構成

する、かつ、ネットワークを輻輳させた場合  
 実験 (2) と同様に、計測用パケットに対して、4ms の  
 デッドライン時間を設定して計測を行う。ただし、  
 ネットワークには RTvNIC を持たない VM から、物  
 理 NIC の最大帯域一杯の 100Mbps のベストエフォ  
 ートパケット (UDP パケット) が送信されている状態  
 で計測を行う。

### 4.3 評価結果と考察

#### 4.3.1 転送遅延について

実験 (1)~(4) の計測結果を表 7 に示す。ネットワークの  
 帯域に余裕がある実験 (1),(2) では、計測用パケットに  
 デッドライン時間を設定してリアルタイム性を確保した場  
 合としなかった場合で、最小遅延時間と平均遅延時間につ  
 いて大きな差は見られなかった。これは計測用パケット以  
 外のパケットが存在せず、キューイング遅延が発生しない  
 ため、EDF スケジューリングによる遅延制御が必要ない  
 ためである。最大遅延時間に関しては、デッドライン時間  
 を設定しない実験 (1) の結果と比較して、実験 (2) の結果  
 は約 100us だけ増加している。これは評価用 OpenFlow ス  
 イッチのパケット処理時間のゆらぎに由来するものである  
 と考える。実験 (2) 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、実験 (1) のパ  
 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、100Mbps 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、 $4\text{byte} / 100\text{Mbps} = 0.32\text{us}$  であるため、  
 最大遅延時間増加の原因ではないと考える。

ネットワークを輻輳させた実験 (3),(4) では、測定用パ  
 ケットにデッドライン時間を設定しない実験 (3) の方に大  
 きな遅延が発生した。ネットワークの帯域に余裕がある  
 実験 (1),(2) と比較すると、実験 (3) では最大遅延時間で  
 約 98ms、平均遅延時間で約 51ms だけ遅延の増加が見ら  
 れる。これは、測定用パケットにデッドライン時間を設定  
 しなかったため、輻輳しているベストエフォートパケット  
 によるキューイング遅延の影響を受けたものであると考え  
 られる。一方で、測定用パケットにデッドライン時間を設

表 7 パケットの転送遅延時間の計測結果

遅延時間	実験 (1)	実験 (2)	実験 (3)	実験 (4)
最大 (us)	1203	1359	99553	7008
最小 (us)	1004	994	996	987
平均 (us)	1085	1090	52401	1026

表 8 オーバーヘッド

パケットサイズ	100byte	1500byte
最大処理時間 (us)	289	288
最小処理時間 (us)	271	273
平均処理時間 (us)	278	279

定し EDF スケジューリングを行う実験 (4) では、最大遅  
 延時間で約 6ms の増加が見られるものの、最小遅延時間と  
 平均遅延時間にはほとんど変化がなく、ネットワークの帯  
 域に余裕があるときと同じ程度の遅延時間に抑えられてい  
 る。これは、測定用パケットのデッドライン時間に基づい  
 て EDF スケジューリングが行われ、輻輳しているベスト  
 エフォートパケットよりも優先的に送信された結果、遅延  
 時間を抑えられたといえる。また、約 6ms 増加した最大  
 遅延時間は、特異点的な遅延の発生パターンで、多くのパ  
 ケットの遅延時間は平均遅延時間の周辺に分布している。  
 最大遅延時間の増加の原因として考えられるのは、他プロ  
 セスへのコンテキストスイッチによる処理の中断である。  
 実験 (4) では、ネットワークを輻輳させるパケット送信の  
 ために、RTvNIC を持たない VM から帯域一杯のパケッ  
 トを送信する処理が行われている。このプロセス (VM) へ  
 のコンテキストスイッチが発生し、最大遅延時間が増加し  
 たものと考えられる。実験 (1),(2) では、この輻輳パケッ  
 トを送信する VM がなかったため、コンテキストスイッチ  
 による遅延が発生していないものとする。この結果は、  
 VM まで含めた EndtoEnd のリアルタイム性を確保するた  
 めには VM スケジューリングまで制御する必要があること  
 を示唆している。

実験 (3) と実験 (4) より、輻輳したネットワークにおい  
 て、RTvNIC システムでパケットへ付加したデッドライン  
 時間に基づく EDF スケジューリングによる最悪転送時間  
 の削減効果が確認できた。ネットワークを設計する際に、  
 帯域に余裕を持たせた設計を行う事が遅延時間を短く抑  
 える上で重要であるが、物理的な帯域一杯まで輻輳する可  
 能性のあるネットワークでは、RTvNIC システムによるパ  
 ケット転送遅延時間の制御が有効であると言える。

#### 4.3.2 パケットへのマーキングのオーバーヘッドについて

RTvNIC システムで追加したことによるパケット転送遅  
 延の増加要因である、パケットへの MPLS ラベルの挿入  
 のオーバーヘッドについて計測した。計測結果を表 8 に示  
 す。RTvNIC システムはミリ秒オーダーのリアルタイム性の  
 保証を目指しているため、表 8 で示したパケットマーキン  
 グの遅延時間は十分に小さいといえる。

#### 4.4 おわりに

本論文では、OpenFlow と VMM が協調動作することにより、ネットワークを介した 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、OpenFlow と VMM の協調動作のインターフェースの設計と、通信のリアルタイム性を確認、保証するための機構の設計を示した。そして、パケットスケジューラとして EDF スケジューリングを実装し、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、ベストエフォート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設定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、本研究が提案する RTvNIC システムによるパケットの転送遅延の計測により、EndtoEnd の通信でリアルタイム性が確保されていることを確認できた。また、パケットへのマーキングとネットワークでのパケットスケジューリングにより、パケットのリアルタイム性が確保できることを示した。

本論文での評価は、RTvNIC システムで保証するリアルタイム性の一部を確認する基礎的な評価である。今後は、デッドラインミス時の QoS 再確保などの実装を進め、QoS 再確保の効果に対する評価、また、リアルタイム性を確保するフロー数を増やしての評価など実ネットワーク環境に近い条件で評価を行い、RTvNIC システムによるリアルタイム通信の保証を確認する必要があると考えている。

#### 参考文献

- [1] S.Blake: An Architecture for Differentiated Services, IETF RFC Standard 2475, 1998
- [2] L. Zhang: RSVP: A New Resource Reservation Protocol, IEEE Network, vol. 7, no. 5, pp.8-18 1993
- [3] R.Braden, D.Clark, and S.Shekner: Integrated services in the Internet architecture: an overview, RFC 1633, June 1994
- [4] 太田貴也, 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), 1-8, 2012-11-28
- [5] 品川 高廣, 榮樂 英樹, 谷本 幸一, 面 和成, 長谷川 晶一, 保理江 高志, 平野 学, 光来 健一, 大山 恵弘, 河合 栄治, 河野 健二, 千葉 滋, 新城 靖, 加藤 和彦: 準バススルー型仮想マシンモニタ BitVisor の設計と実装, OS-4:仮想化,2008年並列/分散/協調処理に関する『佐賀』サマー・ワークショップ (SWoPP 佐賀 2008), <http://www.bitvisor.org/>
- [6] 鈴木 章浩, 追川 修一: ARM アーキテクチャ用仮想マシンモニタの開発情報処理学会研究報告. [システムソフトウェアとオペレーティング・システム] 2009-OS-111(12), 1-8, 2009-04-15 ARM アーキテクチャ用仮想マシンモニタの実装情報処理学会研究報告. [システムソフトウェアとオペレーティング・システム] 2010-OS-113(1), 1-8, 2010-01-20
- [7] Kukreja, G.; Singh, S. : Optimization of KVM Network Based on CPU Affinity on Multi-cores, Information Technology, Computer Engineering and Management Sciences (ICM), 2011 International Conference on, Date of Conference: 24-25 Sept. 2011 Volume: 4 Page(s): 347 - 351
- [8] Jiuxing Liu, T J. Watson Res. Center, IBM, Hawthorne, NY, USA : Evaluating standard-based self-virtualizing devices: A performance study on 10 GbE NICs with SR-IOV support, Parallel & Distributed Processing (IPDPS), 2010 IEEE International Symposium on, Date of Conference: 19-23 April 2010, Page(s): 1 - 12
- [9] Yuan Tang Sch. of Comput. Sci. & Eng., Univ. of Electron. Sci. & Technol. of China, Chengdu, China Jian-Ping Li : VON/KVM: A high performance virtual overlay network integrated with KVM , Apperceiving Computing and Intelligence Analysis (ICACIA), 2010 International Conference on, Date of Conference: 17-19 Dec. 2010 Page(s): 129 - 132
- [10] Civanlar, S. ARGELA Technol., Istanbul, Turkey Parlakisik, M. ; Tekalp, A.M. ; Gorkemli, B. ; Kaytaz, B. ; Onem, E. : A QoS-enabled OpenFlow environment for Scalable Video streaming, GLOBECOM Workshops (GC Wkshps), 2010 IEEE, Date of Conference: 6-10 Dec. 2010, Page(s): 351 - 356
- [11] 戸田巖, ネットワーク QoS 技術, Ohmsha, 2001