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

2006－OS－102（8）
2006／5／13

TwinOSにおける通信の監視方式と基本性能評価

山本裕馬[†]　乃村能成[†]　谷口秀夫[†]

近年，OSに対する不正な攻撃が増加している．その対策として，計算機への不正侵入監視手法が多く提案されている．本稿では，２つLinuxを１台の計算機上で独立に走行させるTwinOSを利用して通信の監視を行う方式を提案する．TwinOSでは，一方のOSが占有するNICの機能を他方のOSに仮想化して提供する擬似NIC機能が実現されている．提案方式では，この擬似NIC機能を利用し，他方のOSが行う通信の監視を行う．監視対象となるOSの外部から監視を行うため，攻撃者による監視記録の改ざんを防ぐことが期待できる．また，提案方式をTwinOS上に実装し，基本性能を評価した結果を報告する．

Evaluation of communication monitoring method on TwinOS

Yu-ma YAMAMOTO,[†] Yoshinari NOMURA[†]
and Hideo TANIGUCHI [†]

Recentl, illegal attacks to OSs are increasing. A lot of intruder detection method have been proposed. In this paper, we propose a method for monitoring communication using TwinOS. TwinOS runs two Linux on a single PC, and has a virtual NIC that provides one Linux with the other Linux's NIC. Our proposed method monitors one Linux's communication from the other Linux via virtual NIC, and keep log from intruders.

1. はじめに

近年，インターネットが普及したことにより，オペレーティングシステム (以降，OSと呼ぶ) に対する不正な攻撃が増加している．具体的には，ウイルスに感染した計算機からファイル共有ソフトを通じて個人情報が流出するといった被害が発生し，大きな問題となっている．不正な攻撃への対策として，ウイルス対策ソフトやパーソナルファイアウォールソフトが用いられてきた．また，計算機への不正侵入を監視する手法も多く提案されている．しかし，不正な攻撃を行うソフトウェアの巧妙化，多様化により，従来の対策では検知できない攻撃が増加している．不正な侵入を許し，OSが乗っ取られた場合，情報が改ざんされ，侵入者の追跡が困難になる．この問題への対処として，攻撃対象となるOSの外部から監視する方法が有効である．対象となるOSの外部から監視するとき，複数OSが走行する環境を利用する方法が考えられる．この環境を構築する方式として，１台の計算機上に複数のOSを独立に走行させるTwinOS[1)]が提案されている．

TwinOSは，２つのLinuxが１台の計算機上で動作し，ハードウェアを各OSごとに分割し，占有させることによって，互いの処理の影響を受けない，および両OSとも入出力性能を十分に利用できるという特徴を

† 岡山大学大学院自然科学研究科
The Graduate School of Natural Science and Technology, Okayama University

持つ．TwinOS の利用法の１つとして，特定のハードウェア資源を選択し，共有させる手法が提案されている[2]．そして，TwinOS 上に PCI ハードウェアである NIC を対象として，本手法を実現した擬似 NIC 機能が実現されている．

ここでは，TwinOS 上に実現された擬似 NIC 機能を利用し，一方の OS が行う通信を他方の OS から監視する方式を提案する．さらに，提案方式に基づき，受信パケットのロギングを行う部分の実装を行い，提案方式の基本性能を測定した結果について述べる．

2. TwinOS と擬似 NIC

2.1 TwinOS

TwinOS は，以下の設計方針を持つ[1]．

(1) 相互に他 OS の処理負荷の影響を受けない．

(2) 両 OS とも入出力性能を十分に利用できる．

このため，１台の計算機ハードウェアにおいて，プロセッサやメモリ，および入出力機器といった各資源の効果的な共有と占有が必要である．２つの OS の独立性を保つために，共有するハードウェアを最小限とすることが有効であるため，プロセッサのみを共有させる．プロセッサ以外のハードウェアは分割し，分割したそれぞれを各 OS に占有させる．分割するハードウェアのうち，入出力機器は，OS の起動時に指定したものだけを占有させる．各ハードウェアの分割，共有方法を次に示し，**図 1** に TwinOS の構成を示す．

(1) プロセッサ
時分割することによって共有させる．このため，OS の起動処理は順番に行い，走行中はタイマ割込みを契機に OS を切替える．

(2) メモリ
上位と下位に２分割する．そして，先に起動する OS(以降，先行 OS と呼ぶ) にメモリの老番アドレス空間を，後から起動する OS(以降，共存 OS と呼ぶ) に若番アドレス空間を割当てる．

(3) 入出力機器
起動時に指定されたものを占有する．入出力機器からの割込みに対しては，その入出力機器をを占有する OS の割込み処理ルーチンを実行する．このため，走行していない OS が占有する入出力機器からの割込みの場合は，OS を切替える．

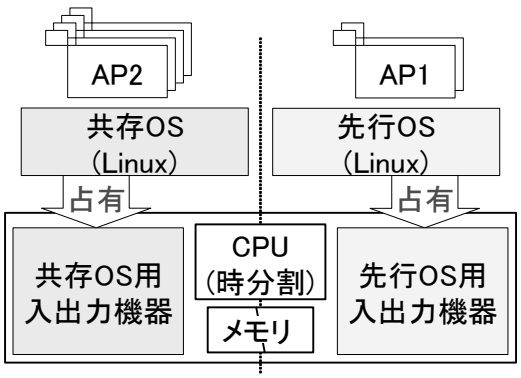

図 1 TwinOS の構成

TwinOS では，自分が占有していないハードウェアを利用することはできない．しかし，次のような場合には，２つの OS にハードウェア資源を共有させたい．

(1) １台の計算機上に１つしかハードウェアを用意できない場合

(2) １つのハードウェアを２つの OS で使用すると利点が生じる場合

そこで，TwinOS の用途を拡大させるために，PCI ハードウェアである NIC を対象として，一方の OS が占有する NIC の機能を他方の OS に提供する手法である擬似 NIC 機能が提案，および実現されている．なお，PCI ハードウェアは，現在主流なハードウェア間の通信規格であり，本手法を PCI ハードウェアに適用することによって様々なハードウェアに適用可能である．

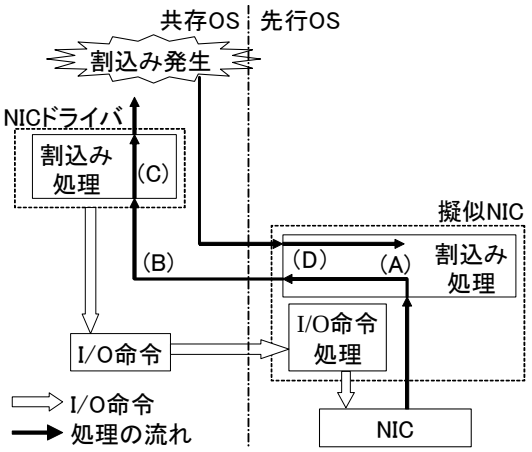

図 2 入出力処理の流れ

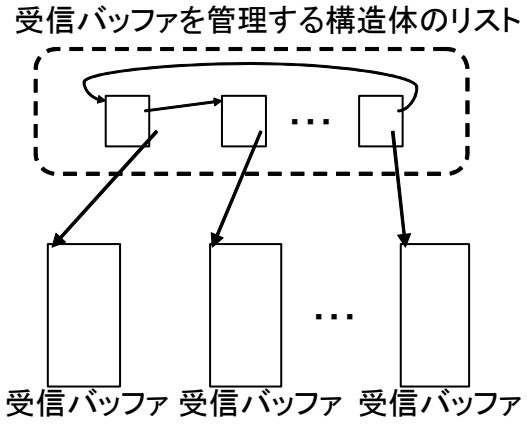

図 3 受信バッファと受信バッファ管理構造体

2.2 擬 似 NIC

擬似 NIC 機能は，TwinOS の用途を拡大させるために，先行 OS が占有する NIC の機能を共存 OS に提供する手法である．OS はデバイスドライバを用い，I/O 命令を発行することによってハードウェアを制御する．また，ハードウェアは割込みによって自身の状態を OS に通知する．このため，擬似 NIC では NIC と NIC ドライバの間で I/O 命令と割込みを監視する．TwinOS で対象としている NIC は 3com 3c905-TX であり，ドライバは 3c905x である．**図 2** に NIC から割込みが入った場合の入出力処理の流れを示し，I/O 命令と割込みの監視方法について以下に説明する．共存 OS の NIC ドライバにより発行された I/O 命令は擬似 NIC が横取りし処理を行う．この際，必要に応じて NIC に対して I/O 命令を発行する．NIC からの割込みに対しては，図 2 の (A) から (D) の処理が行われる．これらの処理を次に示す．

(A) NIC が発行した割込みを受け，擬似のための処理を行う．

(B) OS を切替えて共存 OS に割込みを通知する．

(C) 共存 OS の NIC ドライバでの割込みに対する処理を行う．そして，このとき発行される I/O 命令は擬似 NIC が横取りし，処理を行う．

(D) 共存 OS からの復帰処理である．

2.3 擬似 NIC を用いた通信

TwinOS において，擬似 NIC 機能を用いて先行 OS が占有する NIC を共存 OS に提供し，共存 OS が擬似 NIC 機能を経由して外部と行う通信について説明する．

2.3.1 バッファとパケット

TwinOS で対象とする NIC のドライバは **図 3** に示す構造の受信バッファと受信バッファを管理する構造体のリスト (以降，受信バッファ管理構造体と呼ぶ) を持っている．擬似 NIC 機能を用いた通信では，共存 OS のドライバは受信バッファと受信バッファ管理構造体を持ち，擬似 NIC は受信バッファ管理構造体のみを持つ．パケットを受信する時の NIC と受信バッファの関係を**図 4** に示し，NIC が受信バッファにパケットを書込む流れを以下に説明する．

(1) NIC は擬似 NIC 上の受信バッファ管理構造体を参照し，パケットを書込む受信バッファを調べる．

(2) NIC は共存 OS 上の受信バッファに DMA 転送でパケットを書込む．

(3) 擬似 NIC 上の管理構造体の情報を更新し，受信完了の割込みを発生させる．

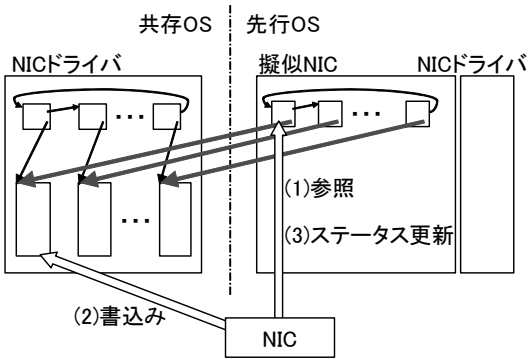

図 4 受信バッファと NIC

この後は図 2 に示した流れとなる．

本稿で提案する方式は，共存 OS が擬似 NIC 機能を経由して通信を行っている状況で，先行 OS 側から擬似 NIC 機能を利用して通信を監視するものである．擬似 NIC 機能を利用し，先行 OS 側からパケットの監視を行うことによって以下の利点が生じる．

(利点 1)監視処理の隠蔽

共存 OS は先行 OS から独立しているため，共存 OS 側から先行 OS の存在を知ることはできない．このため，外部の攻撃者も先行 OS の存在を知ることができず，監視を行っていることを隠すことが可能になりログの改竄を防ぐことができる．

(利点 2)効率のよい監視

一般的なパケットフィルタリングでは，受信したパケットに対して，ドライバ内のバッファからカーネルへパケットがコピーされた後にフィルタリングが行われる．提案方式では，ドライバ内のパケットに対してフィルタリングを行うことで，効率の良いパケットフィルタリングが可能になる．また，提案方式を用いたロギングでは，NIC がパケットを書込んだ OS とは異なる OS 上にログを残すため，通信を行っている OS に与える影響が小さい．

次に，先行 OS 側から共存 OS が行う通信を監視するための実現方式について述べる．

3. 実 現 方 式

3.1 目的と方針

擬似 NIC を利用して先行 OS 上でパケットの監視を行う方式の実現に向けた方針を示す．

(方針 1)共存 OS の改造を行わない．

(方針 2)共存 OS が行っている通信に与える影響を最小限に抑える．

(方針 3)監視のための設定やログの解析，ログの保存処理は先行 OS 上のアプリケーションで行う．

図 4 より，NIC が受信したパケットは共存 OS 内に書込まれる．このため，先行 OS 内に共存 OS 内のパケットを先行 OS 側に提供するパケット取得部を用意する．また，(方針 3) より，先行 OS 上で通信を監視するアプリケーション (以降，監視 AP と呼ぶ) を用意する．提案する方式は上記の 2 つの部分からなり，監視 AP はログを一時的に保持するバッファ(以降，ログ域と呼ぶ) を持つ．以降ではパケット取得部と監視 AP の実現方式を受信パケットのロギングを行う場合を例として述べる．

3.2 パケット取得部

パケット取得部は先行 OS 内に位置し，共存 OS 上に存在するパケットから必要な部分を抽出したものを監視 AP に提供する．

受信バッファにあるパケットを監視 AP に提供する方法としてシステムコールを利用した方法が考えられる．この場合，共存 OS 内の受信バッファから先行 OS 内にパケットの必要な部分を複写しておき，監視 AP からシステムコールが発行されたときに先行 OS 内からログ域へ複写する．しかし，この方法では複写が二回発生するため，オーバヘッドが大きくなる．

そこで，パケット取得時に受信バッファから直接ログ域に複写することを考える．これにより複写回数の削減とシステムコール発行

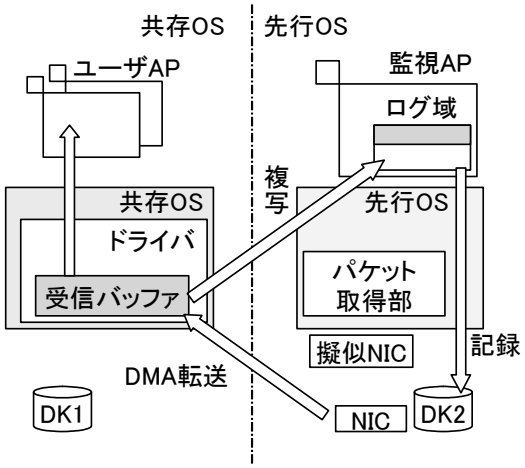

図 5　監視 AP とパケット取得部

によるオーバヘッドの削減ができる．このときのパケット取得部と監視 AP でのデータの流れを**図 5** に示す．パケット取得部を実装するための課題を以下に述べる．

(課題 1)パケットの取得契機
図 2 に示した入出処理の流れの中で，いつログ域への複写するか．

(課題 2)ログ域の位置の把握
先行 OS 内では通常は受信バッファとログ域の存在する領域を操作することができない．このため，この 2 つの領域を同時に操作可能にする必要がある．

(課題 3)ログ域上の書込み読出し位置の判断
ログ域へ直接複写するため，パケット取得部が複写した位置と監視 AP が処理した位置の情報を共有する必要がある．

(課題 4)パケットの抽出部分の決定
受信バッファからログ域へとパケットを複写する際に全てを複写していたのでは効率が悪い．

これらの課題に対する対処を以下に述べる．

(対処 1)パケットの取得は NIC が受信完了の割込みを入れた後から共存 OS の NIC ドライバでの受信処理が行われるまでに行う必要がある．これは，TwinOS で対象としている NIC のドライバでの受信パケットの処理後は，受信パケットの存在する領域がいつ上書きされるか先行 OS 側からわからないためである．よって，パケットの取得は擬似 NIC での割込みに対する処理中に行う．さらに，擬似 NIC では擬似のために必要な受信処理の一部で共存 OS の空間を操作している．これを利用するため，パケット取得を擬似 NIC での受信処理中に行う．

(対処 2)そこで，監視 AP の初期化時にシステムコールを発行し，その中で監視 AP の空間を受信処理中に利用する空間に複写する．これにより，擬似 NIC の受信処理中に受信バッファからログ域への複写が可能になる．

(対処 3)パケット取得部と監視 AP で書込みと読出しの位置情報を共有するため，ログ域を管理する構造体 (以降，ログ域管理構造体と呼ぶ) を監視 AP 上に用意する．そして監視 AP の初期化時にパケット取得部に通知する．パケット取得部がログ域に書込んだ場合は，このログ域管理構造体を更新する．なお，ログ域の終端まで書込んだ場合は再び先頭から書込みを行い，次に書込む領域を監視 AP がまだ処理していない場合は上書きする．これは，複写処理が割込みに対する処理中に行われており，高速に行う必要があるためである．

(対処 4)パケット取得に関する処理を高速に行う必要があるため，複写する部分は静的に決定する．さらに，受信バッファにあるパケットから必要な部分だけを複写するために，受信パケットの先頭からのオフセットとサイズという形で指定する．監視 AP がこれらの値を決定し，システムコールを利用してパケット取得部に通知させる．こうすることで，必要な部分のみをログ域へと書込

むことができる．

3.3 監 視 AP

監視 AP は，ロギングやフィルタリングのためのルールの設定，取得したデータの解析，ログのディスクへの保存を行う．現在の監視 AP では，実行中の動作の変更には対応していない．しかし，実行中のルールの変更を考慮した構成にしている．監視 AP が行う処理を示す．

(処理 1)監視 AP の初期化処理
(処理 2)ルールの設定
(処理 3)ログの読出しと出力
(処理 4)監視 AP の終了処理

処理はこの順で行われ，(処理 3) は周期的に行われる．なお，ユーザからルール変更の要求があった場合には (処理 2) が呼び出される．以降では，それぞれの処理について説明する．

3.3.1 監視 AP の初期化処理

監視 AP の初期化時には以下の処理を行う．

(1) ログ域の確保
(2) ログ域管理構造体の初期化
(3) パケット取得部を初期化するシステムコールの発行

パケット取得部を初期化するシステムコールの中では監視 AP のマッピングの複写，ログ域とログ域管理構造体のアドレスの通知を行う．

3.3.2 ルールの設定

ユーザはルールの作成のために，あらかじめファイルにプロトコルやデータに対して取得するかどうかを記述しておく．監視 AP ではこのファイルを基にルールを作成し，システムコールを用いてパケット取得部に通知する．ルールの作成と同時にパケット取得部がログのヘッダとして書込む書込むための情報の作成する．なお，システムコールによるルールの設定中はパケット取得部による複写処理は停止する．

3.3.3 ログの読出しと出力

パケット取得部はログ域へのデータの書込みをログ域管理構造体を更新することによって監視 AP に通知する．このため，監視 AP は周期的にログ域管理構造体をチェックする．チェックした結果データが書込まれていれば，ログのヘッダ情報を基にパケットの情報を取り出す．その後，ログをファイルに出力し，最後にログ域管理構造体を更新する．

現在，ファイルへ出力するログの形式はユーザへの見せ方を考慮した独自の形式を使用している．しかし，保存されたログを基に解析することを考えると現在の形式では解析が困難である．一方，パケットのログをファイルに保存する形式として，PCAP 形式 (tcpdump 形式)，Sniffer 形式と呼ばれるものが存在している．そして，これらの形式で保存されたファイルを基に解析を行うツールも存在している．このため，後の解析を考慮した独自の形式の作成と上にあげた既存の形式の利用について検討する必要があり，今後の課題である．

3.3.4 監視 AP の終了処理

監視 AP の初期化時に監視 AP 走行中の空間を複写している．このままでは問題が発生する可能性があるため，初期化時に監視 AP の空間を複写した部分をもとの状態に戻す処理を行う．

4. 評　　価

本稿で述べた実現方式を TwinOS 上に実装した．ここでは，その基本性能について述べる．

4.1 評価の観点と測定環境

NIC からの割込みに対して擬似 NIC での受信処理中に行うパケット取得のオーバヘッドが共存 OS の通信に与える影響を明らかにする．そして，得られた結果を基にパケット監視処理下でのスループットを計算により明らかにする．以下に測定する項目を示す．

(1) 基本となる受信処理時間
(2) パケットデータ複写処理時間
パケット取得部によるパケットデータ複写処理を停止した状態で，共存 OS が通信を行う場合の基本となる受信処理時間を測定する．比較対象として改造を行っていない Linux(以降，オリジナルと呼ぶ) での受信処理時間を測定する．そして，パケットデータ複写処理時間を複写データ長を変化させて測定する．

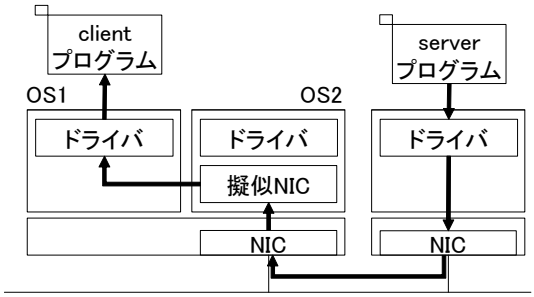

図 6　測定の様子

測定環境は，CPU : Pentium4　3.0GHz, OS : Linux Kernel 2.4.7 , NIC : 3com 3c905-TX である．伝送路は 100Mbps であり，測定に使用した計算機の MTU (Maximum Transmission Unit) は 1500 バイトである．測定を行う様子を**図 6** に示す．図 6 に示すように，TwinOS の他に通信を行う計算機を用意し，共存 OS 上に client プログラム，通信を行う計算機上に server プログラムを用意する．client プログラムの要求に対して server プログラムが共存 OS にパケットを送信する．測定は CPU のタイムスタンプカウント値を出力する rdtsc 命令を用い，得られた値を CPU の周波数で割ることによって処理時間を求めた．なお，計算機は，すべてシングルユーザモードで起動させ，他の計算機が接続されている一般の LAN は使用せずに，専用の LAN を構築し測定を行った．

4.2 基本となる受信処理時間

基本となる受信処理時間として，擬似 NIC が割込みを受けてから復帰処理を行うまでの時間を測定した．これは，図 2 の (A) から (D) に対応する部分である．オリジナルの受信処理時間は，NIC ドライバが割込みを受けてから割込みに対する処理が終了するまでの時間を測定した．測定結果を**表 1** に示す．

表 1 より，基本となる受信処理時間は 14.05μs となり，オリジナルと比較して 9.92μs 遅くなっている．共存 OS の NIC ドライバでの処理における 3.29μs の遅延は，NIC ドライバが発行する I/O 命令を擬似 NIC が横取りし，処理を行っているためである．具体的には，NIC ドライバから 5 回の I/O 命令が発行されており，1 つの I/O 命令あたりの遅延が約 0.65μs となっているためである．

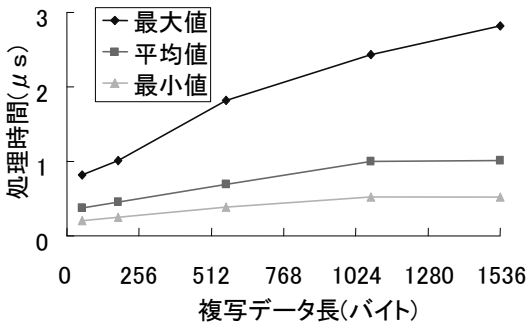

図 7　パケットデータ複写処理時間

4.3 パケットデータ複写処理時間

パケットデータ複写時間を**図 7** に示す．図 7 は，複写データ長を 53 バイトから 1536 バイトの間で変化させながら測定を行った場合の処理時間である．このとき，53 バイトはパケットのヘッダのみをログの対象とした場合であり，1536 バイトは NIC ドライバで定義されている 1 パケットの最大長である．複写データ長が 53 バイトの場合，複写時間の最小値が 0.2μs，平均値が 0.37μs，最大値が 0.82μs となっている．また，複写データ長が 1536 バイトの場合，複写時間の最小値が 0.52μs，平均値が 1.01μs，最大値が 2.82μs となっている．複写データ長が大きくなるにつれて，平均値と最大値の差が大きくなって

表 1 基本となる受信処理時間

測定項目	オリジナル	擬似 NIC
擬似 NIC での処理	–	約 1.99μs
OS 切替え処理	–	約 3.99μs
共存 OS の NIC ドライバでの処理	約 4.13μs	約 7.32μs
復帰処理	–	約 0.75μs

いる．

4.4 監視処理化の通信性能

これまでの測定結果をもとに，擬似NIC機能を用いてパケットのロギングを行っている状況でのスループットを計算により求める．スループットはパケットサイズ (MTU) を基本となる受信処理時間とパケットデータ複写処理時間の和で割ることによって計算できる．MTU は現在の場合 1500 バイトである．基本となる受信処理時間は表 1 の擬似 NIC での処理，OS 切替え処理，共存 OS の NIC ドライバでの処理に該当する．復帰処理が行われる前に次のパケットが到着した場合には処理が行われないため，復帰処理は計算に入れない．複写時間にヘッダのみ (53 バイト) の場合の平均を用いるとスループットは 879Mbps となる．また，複写データ長を最も大きくした場合 (1536 バイト) の平均を用いると，スループットは 838Mbps となる．

しかし，これらの計算結果は共存 OS 走行中に NIC から割込みを受けた場合に必要となる OS 切替えによる影響を考慮していない．

5. ま と め

ここでは，TwinOS 上で一方の OS が行う通信を他方の OS から監視する方式の実現に向け，受信パケットのロギングを行う部分を例として実現方式を述べた．実現においては，TwinOS の利用法の１つとして実装されている擬似 NIC 機能を利用した．提案方式ではパケットの監視を行う部分をパケット取得部と監視 AP に分割し，パケット取得部は監視を行う側の OS 内部で，監視 AP は OS 上で動作する．

パケット取得部では，オーバヘッドの削減を目的として，パケットを共存 OS 内に存在する受信バッファから監視 AP 上に用意したログ域へ直接複写する方法を用いた．この方法を用いる場合の実装における課題として，パケットの取得契機，ログ域の位置の把握，ログ域上の書込み読出し位置の判断，パケットの抽出部分の決定がある．これらの課題への対処として，擬似 NIC での処理中の仮想空間に変更を加え，割り込み処理の中でパケットを複写することで対応した．そして，ログ域を管理する構造体を用いて，ログ域内の位置の把握を行う．

最後に，評価としてパケット取得部での受信処理全体のオーバヘッドを測定し，オリジナルと比較して 9.92μs 遅くなることを示した．複写データ長が 1536 バイトの場合，複写処理時間の平均は 1.01μs となり，スループットは 838Mbps となることを示した．

今後の課題としては，ログの保存形式の検討，提案方式の送信パケットへの適応やフィルタリング機能の実現がある．

参 考 文 献

1) 田渕正樹，伊藤健一，乃村能成，谷口秀夫：二つの Linux を共存走行させる機能の設計と評価，電子情報通信学会論文誌，No.2, pp.251–262 (2005).
2) 桝本　圭，田渕正樹，伊藤健一，乃村能成，谷口秀夫：複数実計算機における非共有リソース利用方式の実装，情報処理学会研究報告， Vol.2003, No.80, pp.9–16 (2003).