

2008年11月18日 セキュアVMワークショップ

BitVisorにおける 透過的なVPN処理の 設計と実装

ソフトイーサ株式会社
(筑波大学大学院システム情報工学研究科)

登 大遊

背景と目的

開発の背景

- 通信を保護するために VPN が必要
 - 通信の暗号化
 - ネットワークへの不正アクセスの防止
- 通常のコンピュータでの VPN 使用
 - セキュリティを高めるには、ユーザー自身で、OS や VPN ソフトウェアの設定・操作を適切にしなければならない。
 - 管理者はできれば強制的に VPN をかけたい。
 - VPN ゲートウェイでアクセス制御を行いたい。
- BitVisor
 - VPN を組み込めば、ゲスト OS がネットワーク通信を行う際にセキュリティを強制的にかけることができるはず。
 - BitVisor に VPN を組み込み VPN の利用を強制すればユーザーの設定にかかわらず全ての通信に VPN がかけられる。

VPN (Virtual Private Network) の利点

■ 通常、コンピュータがネットワーク上の他のコンピュータと通信しようとした場合は、

■ 通常のアプリケーション

■ データは暗号化されていないので誰でも盗聴できる

■ 暗号化機能が付いているアプリケーション

■ データは暗号化されているので盗聴できない

■ VPN を利用すると

■ すべてのアプリケーションで通信するデータは自動的に暗号化される

■ アプリケーションが暗号化に対応していなくても OK

開発の目的

- BitVisor のゲスト OS と外部との間のすべての通信を透過的に VPN を用いて暗号化する。
 - 「すべての通信」とは
 - IPv4、IPv6 のすべてのユニキャスト通信
 - 「透過的」とは
 - ユーザーやゲスト OS には VPN を使わず直接通信しているように見えること
 - 既存のアプリケーションやゲスト OS の変更が不要
 - ゲスト OS の VPN の機能を利用しなくても VPN 通信することが可能になる

BitVisor に VPN を組み込んだ場合、 大きなメリットがある

■ 情報漏洩・不正侵入の防止

■ すべての通信は VPN トンネル内を通る

- LAN 上での盗聴を防げる

- LAN に物理的に不正な PC が接続してもデータ漏洩を防げる

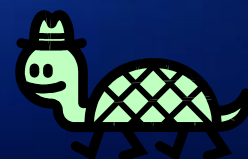
■ ユーザーは何もしなくても良い

■ VPN クライアント機能は BitVisor が提供

- ゲスト OS には VPN クライアントソフトが不要

■ ユーザーは VPN を意識しなくても利用可

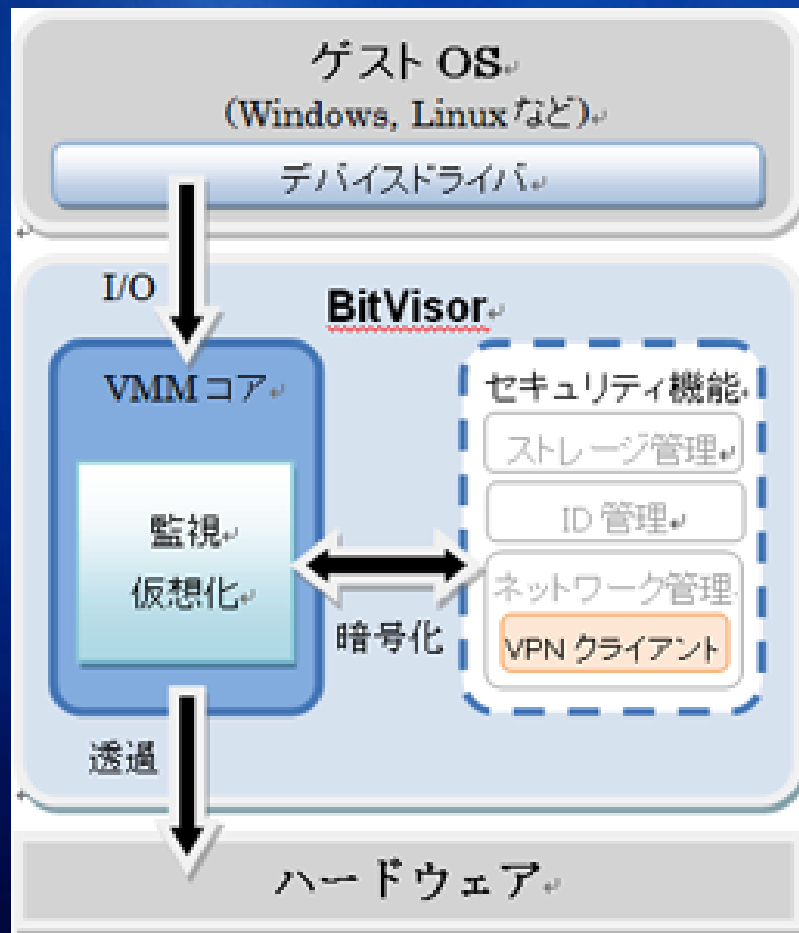
- 管理者による強制アクセス制御が実現



カメさんもびっくり！！

實現方法

BitVisor の構造



<http://www.securevm.org/bitvisor-abs.html> より

■ 準パススルー型

- BitVisor ではセキュリティ機能 (暗号化など) を実現するための最低限の I/O のみを管理し、それ以外の I/O はハードウェアに透過的にアクセス可能とする。

■ VPN クライアント機能

- 上記の設計思想に合わせた形で組み込まなければならない。

BitVisor とハードウェア

■ BitVisor によるハードウェア操作

■ ストレージ (HDD, USBメモリ)

- ゲスト OS と ATAコントローラとの間の通信に介入

■ ネットワーク通信

- ゲスト OS と NIC との間の通信に介入
(NIC: ネットワークインターフェイスカード)

■ スマートカード

- スマートカードに自力でアクセス

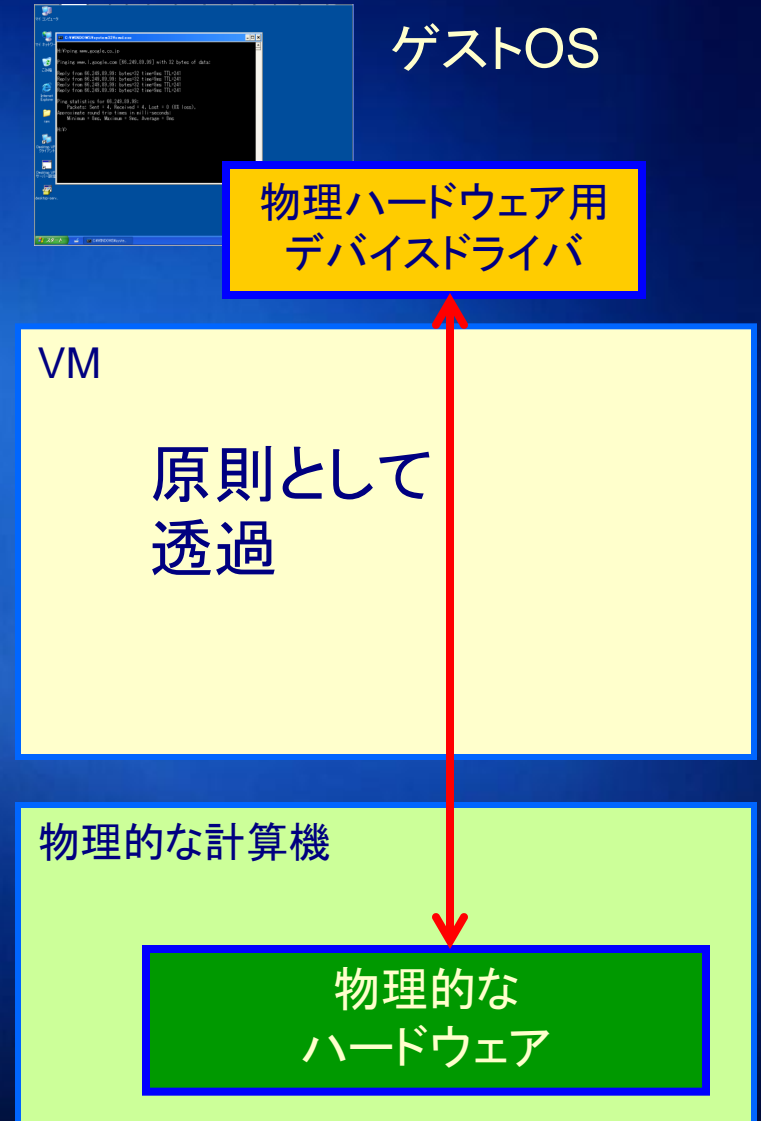
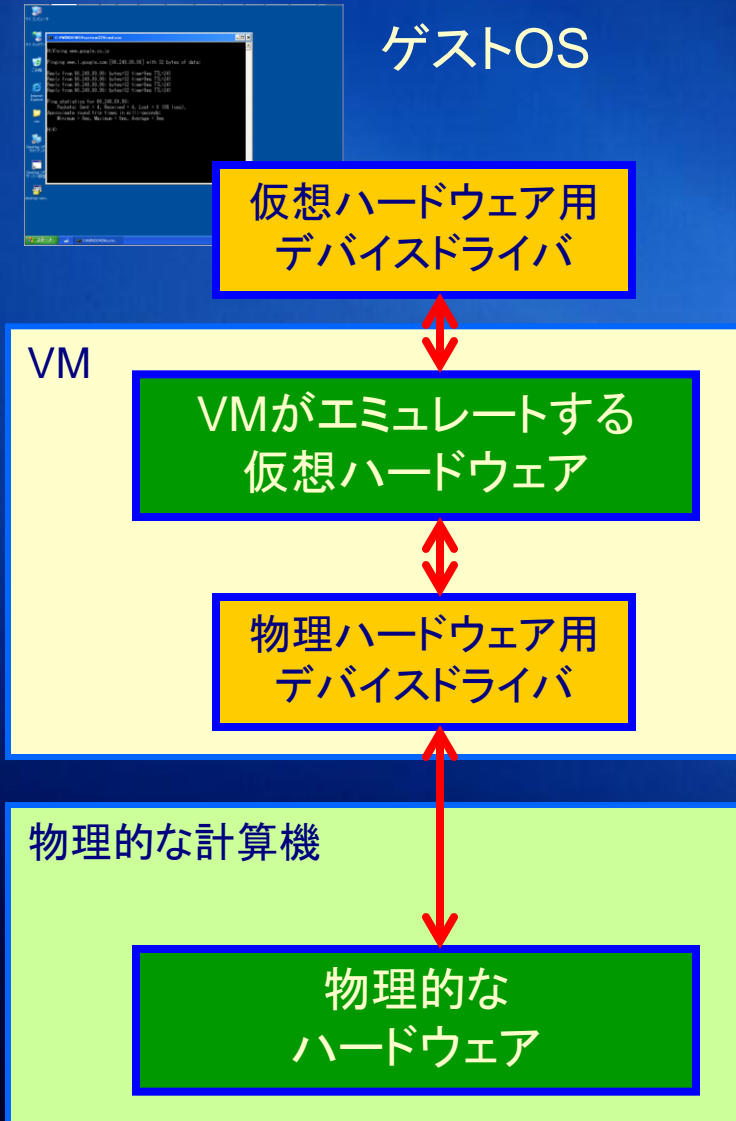
■ 上記以外のハードウェア

■ ゲスト OS とハードウェアとの間の通信には介入しない

- ビデオカード、サウンドカード等はすべてゲスト OS で利用可能

ハードウェア仮想化型 (VMware ESX Server 等)

準パススルー型 (BitVisor)

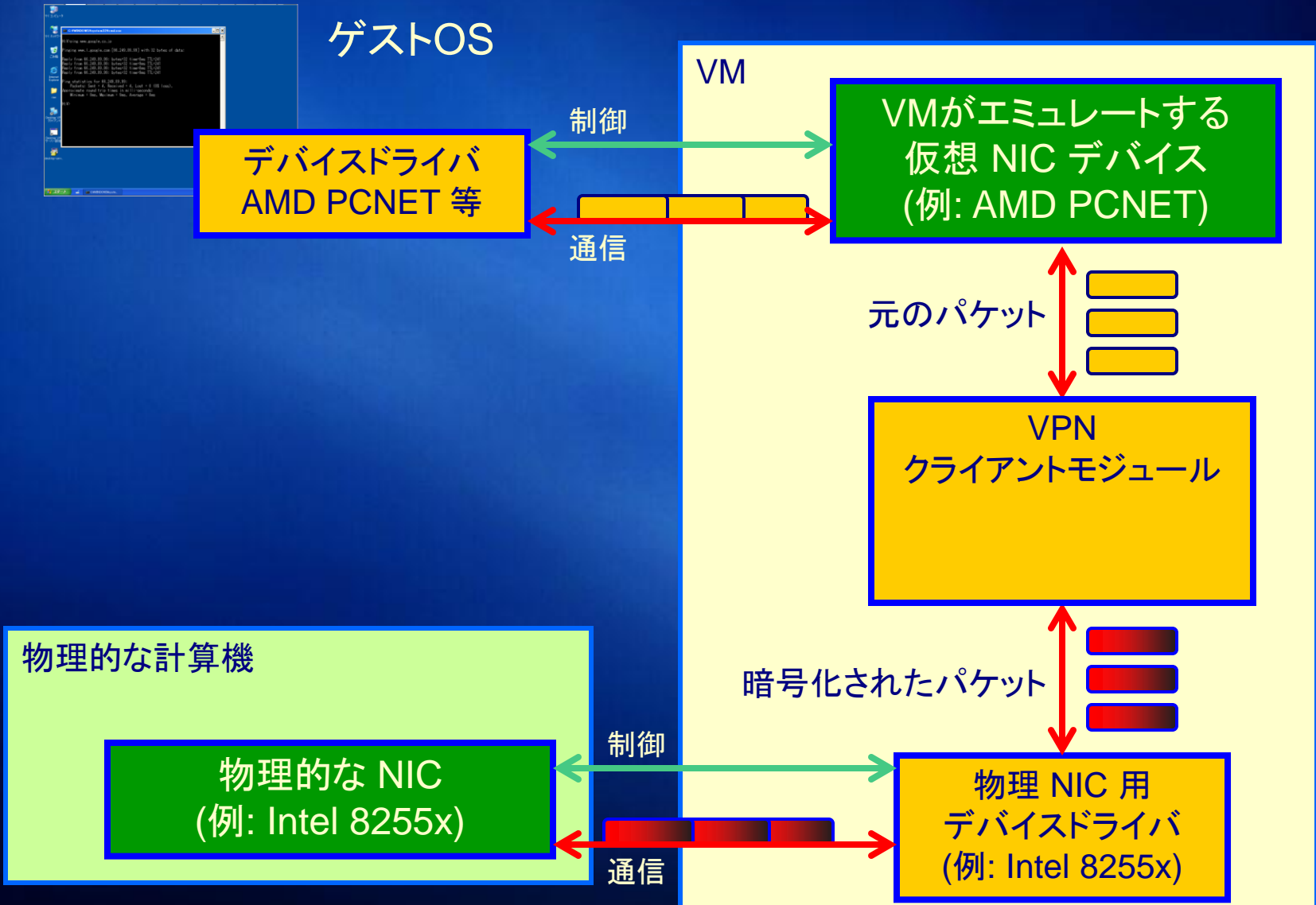


NIC のハードウェアとデバイスドライバとの間のコマンド

- NIC (LAN カード) の本体とデバイスドライバとの間のコマンドは以下のようなものがある。
 - 制御コマンド
 - NIC の初期化・シャットダウン
 - PHY の状態の検出
 - NIC 固有の機能 (LAN ケーブルテスター等) の使用
 - 通信コマンド
 - 送受信対象のパケットの受け渡し
- NIC によってコマンドの仕様は異なる。



ハードウェア仮想化型の場合 VPN 処理を透過的に入れるのは容易



ハードウェア仮想化型の問題点

■ パフォーマンスの低下

- VM がゲスト OS に対して仮想 NIC をエミュレーションする処理はオーバーヘッドがかかる

■ 対応する物理的な NIC の種類の数だけ、VM 内に NIC のドライバが必要

- ゲスト OS が Windows 等の場合は問題ないが BitVisor の VMM 等の独自 OS の場合は対応ドライバを一から作る必要がある
 - NIC 用のドライバを作るのは大変

■ NIC 固有の機能がゲスト OS から利用できない

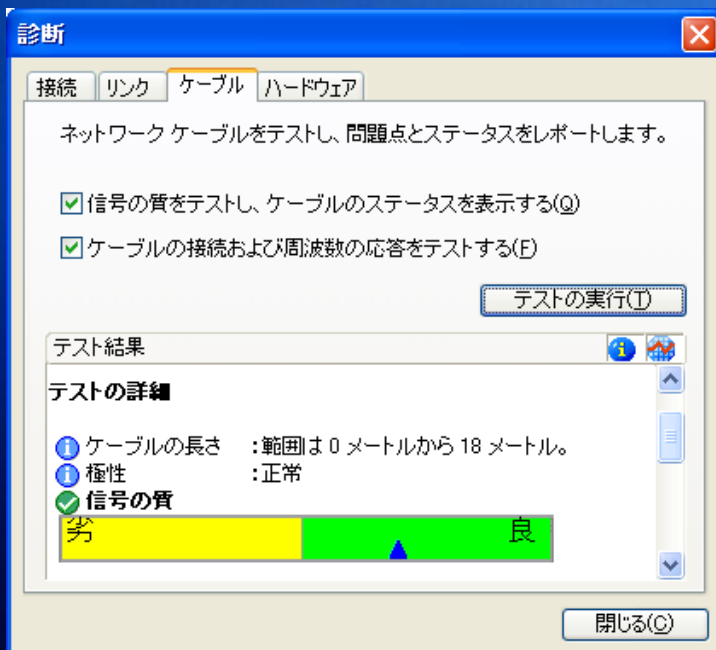
- ネットワーク診断機能
- NIC の状態の表示機能
- タグ付き VLAN 機能

ハードウェア仮想化型の問題点

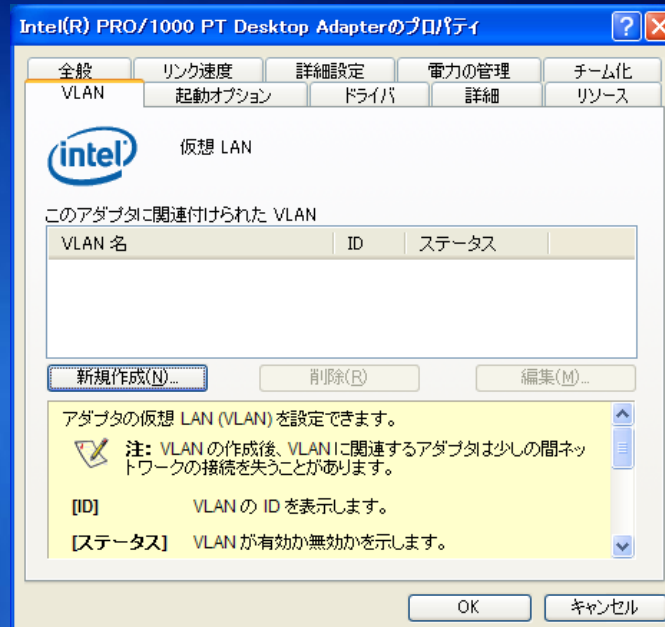
- NIC 固有の機能がゲスト OS から利用できない



NIC の PHY の 状態の検査機能



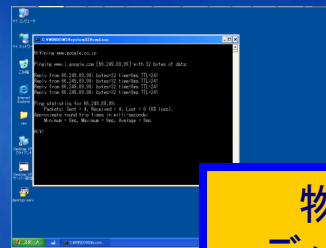
LAN ケーブルの品質の診断
や通信テスト機能



ハードウェアによる タグ付き VLAN 機能

Intel 8254x というチップを
搭載した NIC の機能の例

準パススルー型の場合 VPN 処理を透過的に入れる (本研究)



ゲストOS

物理 NIC 用
デバイスドライバ
(例: Intel 8255x)

- デバイスドライバとNICとの間のコマンドについて
 - 制御コマンドはそのまま通す
 - 通信コマンドは対象となるパケットの VPN による暗号化 / 復号化処理を実施する

物理的な計算機

物理的な NIC
(例: Intel 8255x)

VM

制御

通信

元の
パケット

VPN
クライアントモジュール

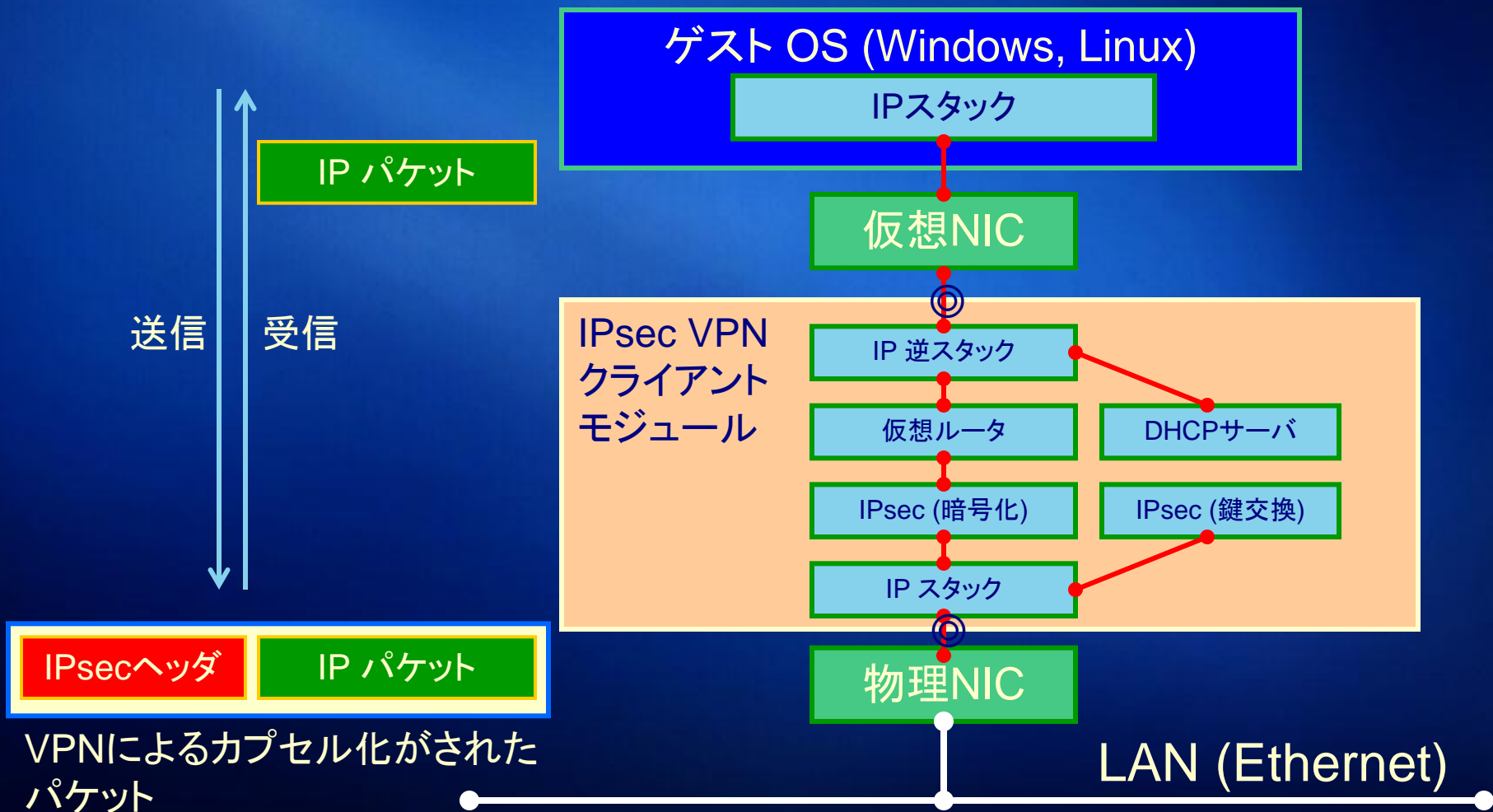
暗号化
された
パケット

通信

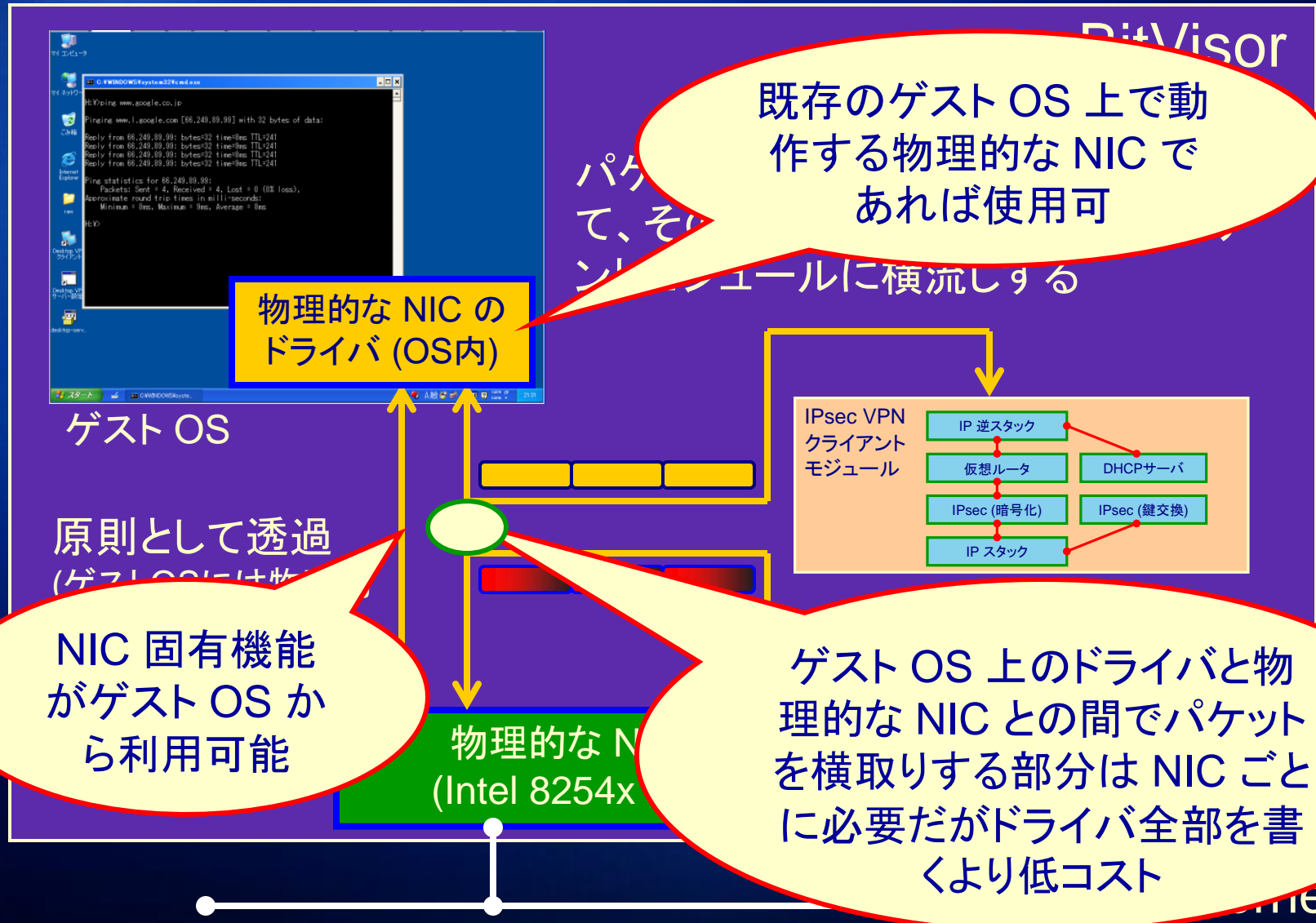
制御

VPN クライアントモジュールの構造

- IPsec VPN クライアント機能 (IPv4, IPv6) を実現するクライアントモジュールのコードは 2006 年度に完成。



準パススルー型の処理の詳細



従来の方法と比べた本研究の利点のまとめ

	従来の方法	本研究の方法
ゲスト OS に見えるもの	VM がエミュレーションした仮想 NIC デバイス	コンピュータにささっている物理的な NIC
デバイスドライバ	ホスト OS (または VM) 上	ゲスト OS 上
NIC 固有の機能	利用できない	利用できる
パフォーマンス	エミュレーションによる低下	エミュレーションが不要
複数 NIC への対応	開発コストが高い (NIC の初期化・制御・パケット送受信をするドライバの組み込みが必要)	開発コストが低い (パケットの横取り・書き換え部分のみ開発すればよい)

設計と実装

最初に対応したデバイス

■ Intel 8255x Fast Ethernet Adapter

- 『Intel PRO/100 シリーズ』として広く流通
- 書籍『Ethernetのしくみとハードウェア設計技法 - プロトコルの詳細からネットワーク対応機器の作成まで』で詳細な利用方法が解説されている
- 特性
 - 100Base-TX, 10Base-T
 - PCIバスマスタによるパケット転送



設計 (初期化)

■ BitVisor の初期化時

- PCI デバイスを列挙しベンダ ID やハードウェア ID が対象 NIC のものがあるか調べる

- 例: `PCI\VEN_8086&DEV_1229&SUBSYS_00408086&REV_0C\4&24B99E98&0&48F0`

- コンフィグレーションレジスタのI/Oポート範囲を取得
- コントロール/ステータスレジスタ (CSR) のI/Oポート範囲を取得

■ 以降、ゲスト OS と対象 NIC との間での I/O 通信を監視

- ゲスト OS が対象 NIC の初期化処理を完了するまで何もしない
- 初期化処理はゲスト OS に任せ、完了したら動作開始

設計 (パケットの送信)

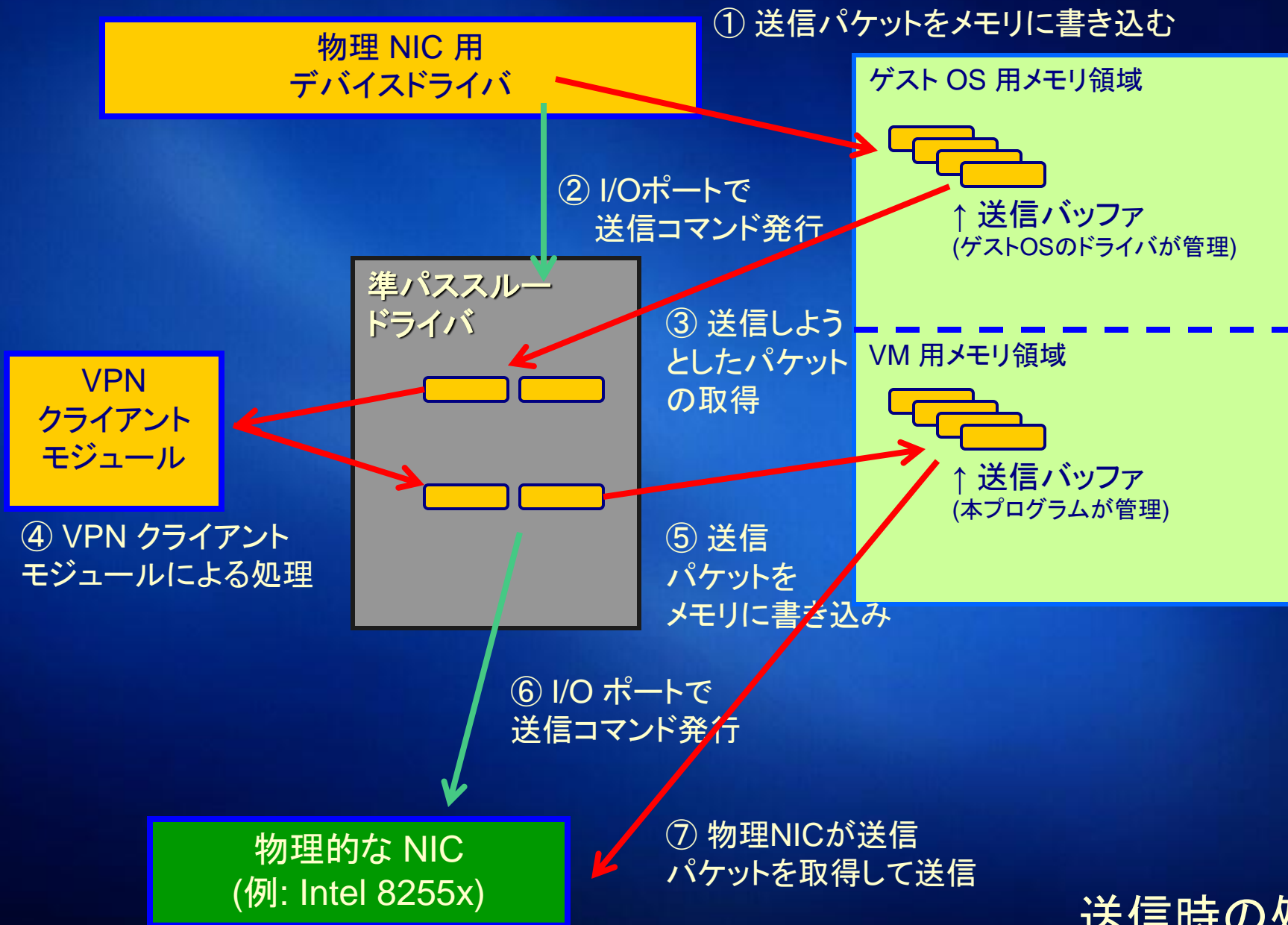
■ ゲスト OS -> VM -> NIC (ゲスト OS がトリガー)

- ゲスト OS はパケットチェーンをメインメモリに置いた後に NIC の CSR に I/O 書き込みで送信開始命令を出そうとする

- BitVisor が I/O をトラップしメインメモリからパケットデータを取得し送信完了マークを書き込む
 - ゲスト OS はパケットの送信が完了したと認識する
- パケットデータを VPN クライアントモジュールを用いて暗号化
- 暗号化されたパケットデータからパケットチェーンを生成してメインメモリ上に置き NIC の CSR に I/O 書き込みで送信開始命令を出す (パケット本体は DMA 転送されるのであって I/O ではない)

■ VM -> NIC (ゲスト OS がトリガーではない場合)

- 上記の最終ステップと同様にパケットチェーンを生成して NIC の CSR に I/O 書き込みし送信開始命令を出す



送信時の処理

設計 (パケットの受信)

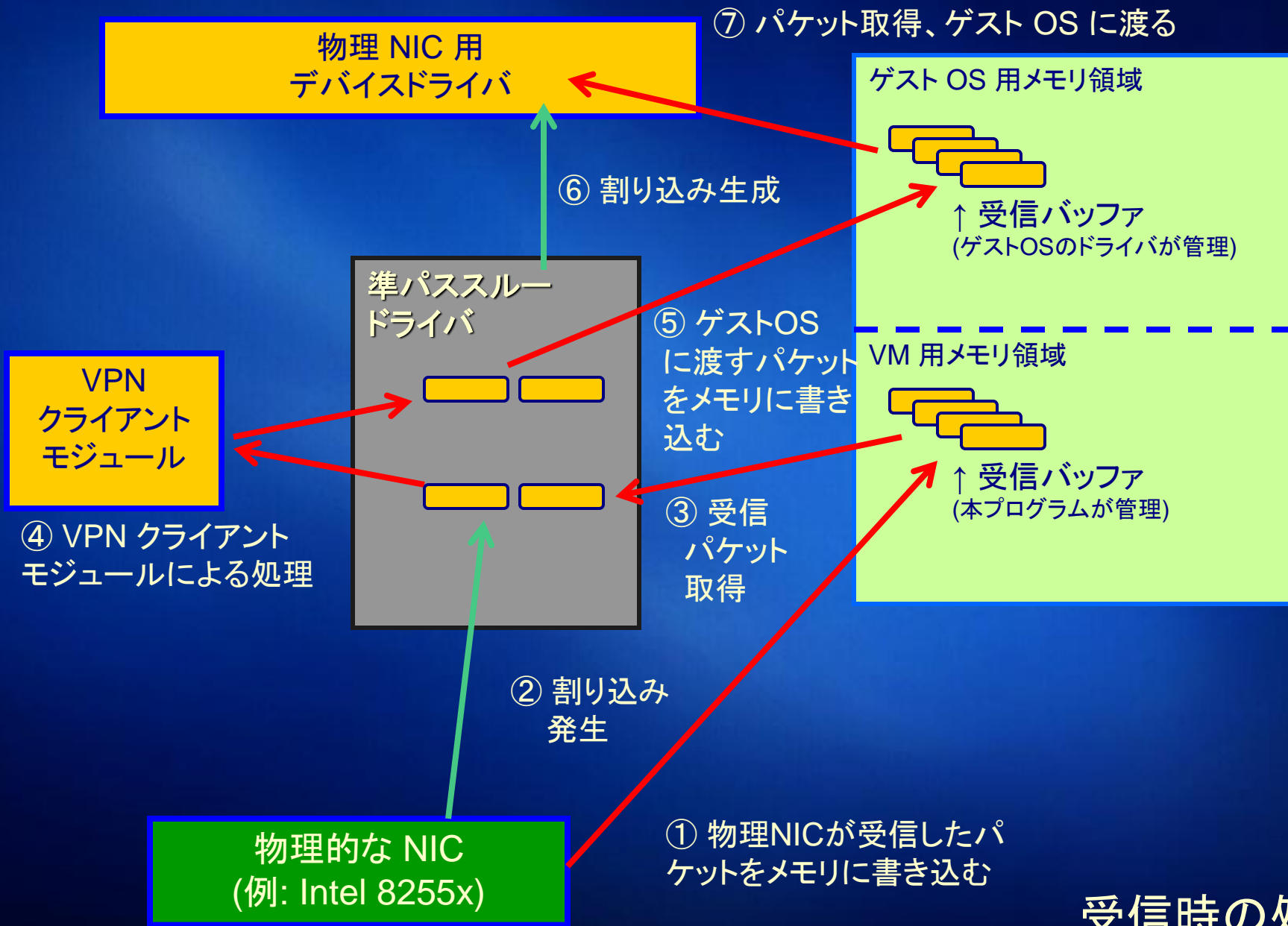
■ NIC -> VM -> ゲスト OS (NIC がトリガー)

■ NIC から割り込みが入るので BitVisor で受信

- あらかじめパケットチェーンを書き込むメモリ領域を確保し I/O 書き込みで NIC の CSR に登録
- 割り込まれた後にメモリ上のパケットチェーン領域を検査してパケットが届いていればそれを VPN クライアントモジュールを用いて復号化
 - NIC には受信処理が完了したように見せかけるためパケットチェーン領域はクリアしてその旨を NIC の CSR に書き込み
- 復号化されたパケットをゲスト OS が保有しているパケットチェーンに書き込みゲスト OS に対して割り込みを送信
- ゲスト OS はあたかも物理的な NIC からパケットが届いたと認識する

■ VM -> ゲスト OS (NIC がトリガーではない場合)

- 動作は (主に) タイマ割り込みで開始
- 上記の最後の 2 ステップと同様にゲスト OS にパケットを渡す



受信時の処理

設計 (その他の処理)

■ ゲスト OS が NIC の PHY の状態を確認しようとした場合

- 例: 現在のリンク速度 (100Mbps, 1000Mbps) をゲスト OS が取得しようとする場合

- それにかかる I/O 呼び出しはトラップしない

■ ゲスト OS が NIC 固有の機能を呼び出そうとした場合

- 例: LAN ケーブルチェック機能、自己診断機能

- それにかかる I/O 呼び出しはトラップしない

動作画面写真・ビデオ

まとめ

まとめ

■ 目的

- BitVisor のゲスト OS と外部との間のすべての通信を透過的に VPN を用いて暗号化する。
 - すべての通信の暗号化
 - 情報漏洩・不正侵入の防止
 - 誰でも意識せずに VPN 機能を利用可能

■ 実現方法

- ネットワーク処理
 - 従来の方法 (ハードウェア仮想化型) ではなく、「準パススルー型」で実施
- 利点
 - パフォーマンス向上
 - ゲスト OS の持つドライバが利用可
 - NIC 固有の機能がゲスト OS からそのまま利用可

■ 現状

- Intel 8255x で動作

■ 将来的には

- 技術的には、Intel 8254x や Broadcom, Realtek 等の他の NIC へも対応可能。NIC ごとのコマンドの差異によりプログラム修正は必要。

2008年11月18日 セキュアVMワークショップ

BitVisorにおける 透過的なVPN処理の 設計と実装

ソフトイーサ株式会社
(筑波大学大学院システム情報工学研究科)

登 大遊