スケーラブルなクラウドネットワークを実現するホストベース論理分離技術

尾上　浩一[†]　　松岡　直樹[†]　　田中　淳[†]

異なる学術機関や企業が計算資源を共有するクラウドデータセンタでは，セキュリティの観点から計算環境毎に分離されたネットワークを提供する必要がある．既存のシステムでは IEEE 標準の分離手法を応用し，ネットワークスイッチで論理ネットワークを提供している．しかし，この場合には提供できる論理ネットワークの数の制限や専用のネットワーク機器を要することによる高い導入コストのような課題が生じる．本論文では，データセンタ内のエンドホストサーバ上でこれらの課題を解決する論理ネットワーク分離技術 *HostVLAN* を提案する．HostVLAN では各エンドホストサーバに MAC アドレスとそれに関連付ける論理ネットワーク識別子を含む分離情報を持たせる．ネットワークデータの受信時に，エンドホストサーバはこの分離情報に基づき，適切な送信先のみに受信データを転送する．従来のネットワーク機器での手法と異なり，提案手法は MAC フレームヘッダに分離情報を含めることなく，エンドホストサーバ上で計算環境単位でネットワークを論理分割する．

Host-based Logical Isolation Technology for Scalable Cloud Networks

Koichi Onoue,[†] Naoki Matsuoka[†] and Jun Tanaka[†]

Computing resources in a cloud data center are shared among different academic institutions and/or enterprises. From the viewpoint of security, the data center networks need to have a network isolation function. To provide private networks, several conventional systems support isolation schemes based on IEEE standards on network switches. However, these systems impose a limitation on the number of isolated logical networks and offer the high cost because of adopting the dedicated hardware. In this paper, we propose *HostVLAN*, a network isolation technology to overcome the challenges on scalability and adoption cost. To provide logical isolated networks, end-host servers deployed in a data center have isolation information containing MAC addresses and logical network IDs associated with them. The end-host servers forward received network data to designated VMs based on the isolation information. Unlike conventional schemes at the network switch level, HostVLAN provides private networks at the end-host server without adding logical network IDs to MAC frame header.

1. はじめに

データセンタを利用して必要に応じてサーバやストレージを利用できる，Amazon EC2[1] のような Infrastructure as a Service (IaaS) は，複数の学術機関や企業の間で計算資源を共有できるため有用である．以降では，学術機関毎や企業毎等に分離される計算環境の単位をテナントと呼ぶ．

一般的に，IaaS ではテナントが利用可能な計算資源を必要に応じて増加できるように，L2 ネットワーク単位で余剰計算資源を確保している．多数の仮想マシン（VM）による計算環境からなるデータセンタで，L2 ネットワークが小規模である場合，複数の L2 ネットワークを組み合わせることが必要となる．この場合，各 L2 ネットワークで余剰計算資源の領域を確保するため，分割損が生じてしまう．さらに負荷分散，耐障害性，計算環境の移動も考慮すると[2]，データセンタでは大規模な L2 ネットワークが必要となる．

また，IaaS では異なるテナント間でデータが漏洩しないように，テナント毎に分割されたネットワークを提供しなければならない．以降では，テナント単位での分離機能をマルチテナンシと呼ぶ．ネットワークの規模の拡大に付随して，データセンタ内に分離できるテナント数も増加できなければならない．さらに，データセンタ内では，異なる種類の仮想ネットワーク

† 株式会社 富士通研究所
Fujitsu Laboratories Ltd.

を提供する必要もある．例えば，データセンタ内の管理用およびストレージ用の仮想ネットワーク，テナント毎の仮想ネットワークが含まれる．

これらのことから，データセンタでは大規模 L2 ネットワークとともに，マルチテナンシのスケーラビリティも必要となる．これまで大規模 L2 ネットワークやクラウドデータセンタ用のネットワーク基盤が提案されている[3)~7)]が，マルチテナンシは述べられていない．

本論文では，クラウドデータセンタにおいてマルチテナンシを提供するためのネットワーク分離技術 *HostVLAN* を提案する．提案手法は MAC フレームヘッダに分離識別子を付与することなくエンドホストサーバでマルチテナンシを実現させるため，特定の MAC フレームフォーマットに依存しない．これにより HostVLAN をクラウドデータセンタに適用することで，汎用的な物理スイッチを用いて，理論的に上限のない多数の論理ネットワークを提供できる．

我々はエンドホストサーバ上で仮想マシンモニタ（VMM）KVM[8)] を用いて，KVM が提供する仮想ネットワーク機能を拡張し，HostVLAN の設計および実装を行った．HostVLAN の評価では，(1) 評価系の上でマルチテナンシを実現できていることを確認し，(2) 実行時オーバーヘッドを測定した．

以降，2 章で既存の論理ネットワーク分離技術について述べる．3 章で提案技術 HostVLAN について説明する．4 章と 5 章で，各々HostVLAN の実装と評価について述べる．最後に，まとめを 6 章で述べる．

2. 関連研究

L2 ネットワークでマルチテナンシを提供するためには，マルチキャストフレームや unknown ユニキャストフレームをテナント毎に転送する必要がある．ここでは，主に物理スイッチでマルチテナンシを実現するための 2 つの既存の IEEE 標準技術である，VLAN tagging と Provider Backbone Bridge（PBB）を用いる手法について述べる．

VLAN tagging は MAC フレームのヘッダに VLAN ID を含む VLAN tag を挿入する．論理ネットワーク ID として VLAN ID（12 ビット）を用いることができる．既存の L2 スイッチの多くが VLAN tagging に対応しているため，物理ネットワークへの適用が容易である．このため，既存の多くの VMM の仮想ネットワークが VLAN tagging に基づくネットワーク分離機能を提供している[9)~11)]．

しかし，VLAN tagging を用いた手法には論理ネットワークの ID 空間が制限されてしまうという課題が生じる．論理ネットワーク ID として用いられる VLAN ID は 12 ビットであり，予約済 ID 0，4095 を除く 4,094 の論理ネットワークしか提供できない．近年盛んなクラウドサービスの提供に伴い，学術機関や企業等の利用者が増大し，データセンタの規模が大きくなってきている．また，データセンタネットワークでは，負荷分散，耐障害性，計算環境の移動，ネットワーク関連の設定の利便性の観点から，大規模 L2 ネットワークも必要となってくる[2)]．これらに伴い，マルチテナンシのスケーラビリティも必要になってくる．

論理ネットワークの ID 空間の制限を課題とした，マルチテナンシを提供する研究がこれまで提案されている[12),13)]．SEC2[12)] や VSITE[13)] は，データセンタネットワークを 2 階層に分類し，VM が配置される複数のネットワーク（エッジネットワーク）をスイッチからなるネットワーク（コアネットワーク）で接続させる．データセンタ単位ではなく，エッジネットワーク単位で VLAN ID が一意に決まるようにする．エッジネットワーク間の接続は，コアネットワーク・エッジネットワーク境界に配置されたスイッチが，MAC-in-MAC または MAC-in-IP でエッジネットワーク間をトンネリングする．これによってデータセンタ全体で VLAN ID 空間を拡大できる．ただし，エッジネットワーク毎の VLAN ID 空間の制限は 12 ビットのままとなってしまい，トンネリングに伴うフレームヘッダも付与されてしまう．

PBB では MAC フレームが PBB 専用の MAC フレームヘッダでカプセル化される（MAC-in-MAC 方式）．この場合，PBB 専用の MAC フレームヘッダの I-TAG の I-SID （24 ビット）が論理ネットワーク ID として用いられる．I-SID の ID 空間は VLAN ID の ID 空間より大きいため，クラウドデータセンタ内により多くのテナントを分離できる．

しかし，PBB を用いた手法には次の 2 つの課題がある．1 つ目の課題は，専用の物理スイッチを設置する必要があり，導入コストがかかってしまう．2 つ目の課題は，18〜22 バイトの MAC フレームヘッダを付与するため，ネットワーク中に流れるデータサイズが増加し，ネットワーク帯域を浪費してしまう．

3. 提案技術：*HostVLAN*

3.1 設計方針

HostVLAN では以下の目標を同時に達成させる．

多数の論理ネットワークを提供できる VLAN tagging（12 ビットの ID 空間）よりも多くの論理ネットワークを提供できる．

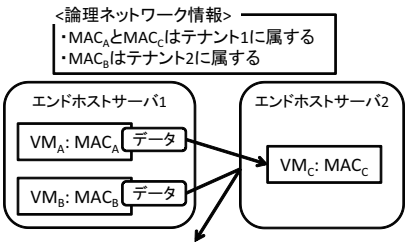

図 1　動作原理

導入コストが低い　PBB のように専用のスイッチを用意する必要がなく，導入コストがかからない．

専用の **MAC** フレームフォーマットを用いない　エンドホストサーバ上での分離制御として実現させることで，PBB の MAC フレームヘッダの追加のようなネットワークデータサイズに関するオーバーヘッドが生じない．また，エンドホストサーバで IP ヘッダをカプセル化する手法[14] と異なり，VM の送受信 MAC フレームに対し，checksum offloading のような NIC のハードウェアオフロード機能も利用できる．

容易でかつ柔軟な管理・運用ができる　データセンタでネットワーク分離を一元的に管理・運用でき，かつその操作が容易である．また，1 つの仮想 NIC に対して複数の論理ネットワークを割り当てたり，動的に MAC アドレスと論理ネットワークの関連付けを変更できる．

3.2　動作原理

HostVLAN では基本的に，ネットワークデータ（MAC フレーム）中の送信元アドレスが属するテナントと送信先アドレスが属するテナントが一致した場合のみデータ転送を許可することにより，ネットワークを分離する（図 1）．

3.3　Logical Network ID (LNID)

HostVLAN では分割される論理ネットワークの識別子として *Logical Network ID (LNID)* を用いる．構築できる論理ネットワークの柔軟な設定を可能にするために，MAC アドレスと LNID は多対多の関係にしている．まず，複数の MAC アドレスと 1 つの LNID が関連付けさせる．さらに，1 つの MAC アドレスに複数の LNID を関連付けることもできる．これは 1 つの VM（仮想 NIC）は複数の論理ネットワークに属することを意味する．また，すべての論理ネットワークに属することに相当する特別な LNID として *global LNID* を定義する．

3.4　HostVLAN を用いたシステム構成

HostVLAN を利用したシステムはデータセンタ内

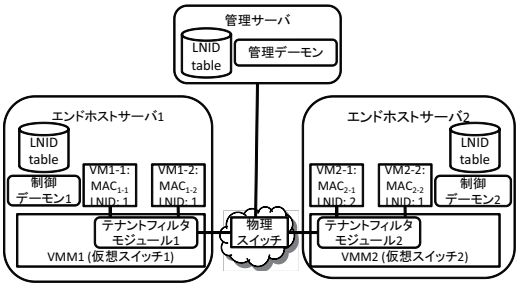

図 2　HostVLAN を適用したシステム

の論理ネットワーク情報を含む *LNID table* を集中管理する管理サーバと，ネットワークの分離を行うエンドホストサーバ からなる（図 2）．LNID table は管理サーバによって各エンドホストサーバ上に配置される．LNID table には MAC アドレスとそれに関連付けられたテナントの識別子（LNID）が含まれる．また，VMM に含まれる仮想スイッチにより物理・仮想 NIC 毎に割り当てられた仮想ポートにも MAC アドレスと LNID が関連付けられている．以降では，この情報を仮想ポート情報と呼ぶ．エンドホストサーバは MAC フレームを受信した際，LNID table と仮想ポート情報の LNID に基づいて適切な送信先 MAC アドレスに受信データを転送する．ネットワーク分離に関連する，LNID table と仮想ポート情報を以降では分離情報と呼ぶ．

3.4.1　管理サーバ

管理サーバは LNID table を管理する．データセンタの管理者は管理サーバ上でのみ管理コマンドを用いて分離情報の管理操作をすればよい．管理コマンドが発行されたとき，管理デーモンがエンドホストサーバと連携して分離情報を更新する．

3.4.2　エンドホストサーバ

エンドホストサーバでは VMM が稼働し，その上で稼働する VM 単位（仮想 NIC 単位）でネットワークが分離される．エンドホストサーバは制御デーモンとテナントフィルタモジュールからなる．制御デーモンは管理デーモンと連携してエンドホストサーバで保持する分離情報を更新する．テナントフィルタモジュールは分離情報に基づいて，送受信データを転送する（フィルタリング）．

3.5　論理分離機構：テナントフィルタモジュール

3.5.1　受信データの転送

図 3 の流れで，HostVLAN はネットワークデータ（MAC フレーム）の受信時に LNID table の要素とのパターンマッチによって，適切な送信先にデータを転送する．このフィルタリングは仮想スイッチの拡張機

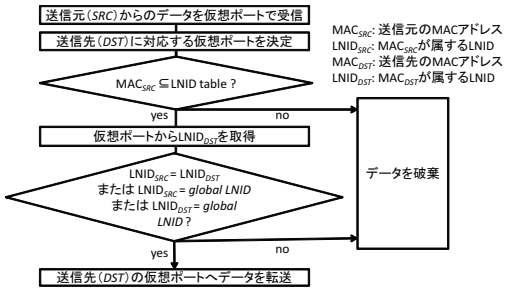

図 3 受信 MAC フレームのフィルタリングの流れ

能として実現させる．

まず，仮想スイッチの転送機能により，VM または物理 NIC とつながっている各仮想ポートから受信した MAC フレームを，送信先アドレスに基づいて適切な転送先仮想ポートが決定される．その後，仮想スイッチの拡張機能として実現したテナントフィルタモジュールでは，LNID table から送信元 MAC アドレス（MAC_{SRC}）の LNID を取得する．LNID table を走査するための MAC_{SRC} には MAC フレームヘッダの MAC アドレスが用いられる．もし取得できなければ，その MAC フレームが属するテナントではないと判断し，これを破棄する．

他方，送信先 MAC アドレス MAC_{DST} が属する LNID は，MAC フレームヘッダの MAC アドレスを用いて LNID table から取得するのではなく，転送先仮想ポートに関連付けられた LNID を用いる．もし，MAC フレームヘッダの送信先 MAC アドレスを用いたとすると，MAC_{DST} がマルチキャストアドレスである場合に，LNID table から MAC_{DST} の LNID を取得できない．HostVLAN では転送先仮想ポートに関連付けられた LNID 情報を用いることにより，マルチキャストフレームに対応できる．

最後に，MAC_{SRC} と MAC_{DST} が関連付けられている LNID が同じであるか，または関連付けられた LNID のどちらかが global LNID であった場合に，該当する仮想ポートに MAC フレームを転送する．

3.5.2 送信データの検査

HostVLAN は MAC アドレスに基づいて論理的にネットワークを分離する．このため，もし VM が攻撃者に奪取されてしまった場合，仮想 MAC アドレスが変更された偽装データを送信できる．攻撃者は元々属していた ID とは異なる LNID のネットワークにデータを送信することで，異なる論理ネットワークへの DoS 攻撃を実現できてしまう．この偽装データによる DoS 攻撃を防ぐために，HostVLAN では仮想 NIC のデータ送信時に送信元 MAC アドレスを検査する．送信元仮想ポートに関連付けられている MAC アドレスが MAC フレームヘッダの送信元 MAC アドレスと異なる場合，その MAC フレームが偽装されているとみなし，これを破棄する．

3.5.3 動 作 例

図 4 には，エンドホストサーバ 1 上で稼働する VM1-1 から送信されたブロードキャストフレームに対するフィルタリングの動作例が示されている．

まず，エンドホストサーバ 1 で，テナントフィルタモジュール 1 が送信データの検査のため，VM1-1 から送信されたブロードキャストフレームの送信元 MAC アドレスと仮想ポート VP1-1 に関連付けられた MAC アドレスを比較する．ここでは同じ MAC_{1-1} であるためデータ転送が許可される．

次に，エンドホストサーバ 2 で，仮想スイッチにより，ブロードキャストフレームの転送先仮想ポートとして VP2-1，VP2-2，VP2-3，VP2-4 が決定される．その後，テナントフィルタモジュール 2 により，送信元 MAC アドレス MAC_{1-1} に関連付けられた LNID 1 と各仮想ポートに関連付けられた LNID を比較する．これにより，同じ LNID 1 に属する VP2-1 と VP2-4 にはデータが転送され，異なる LNID 2 に属する VP2-2 と VP2-3 にはブロードキャストフレームは転送されない．

3.6 分 離 情 報

3.6.1 LNID table

LNID table の要素には，VM の仮想 MAC アドレスと LNID のタプルのペアを含む．例えば，ある VM の仮想 NIC（MAC アドレスが 00:11:22:33:44:55）を LNID 1 と 2 に属するように設定する場合，(00:11:22:33:44:55, {1,2}) が LNID table に含まれる．

管理サーバが持つ LNID table には，データセンタ内のすべての VM の仮想 MAC アドレスが含まれる．他方，エンドホストサーバが持つ LNID table には，エンドホストサーバ上の VM が属している LNID と同じデータセンタ内の VM の仮想 MAC アドレスのみ含まれればよい．例えば，データセンタ内に，MAC_A，MAC_B，MAC_C の 3 つの VM が存在し，LNID が，各々，1, 2, 1 であると仮定する．さらに，エンドホストサーバ 1 では MAC_A の VM が稼働していると仮定する．この場合，エンドホストサーバ 1 では，LNID 1 に関連する MAC_A と MAC_C の 2 つの MAC アドレスを保持するだけでよい．各物理サーバでデータセンタ全体で管理する LNID の一部だけが存在することは少なくない．このため，各エンドホストサーバで管理する LNID table のサイズを小さくできる．

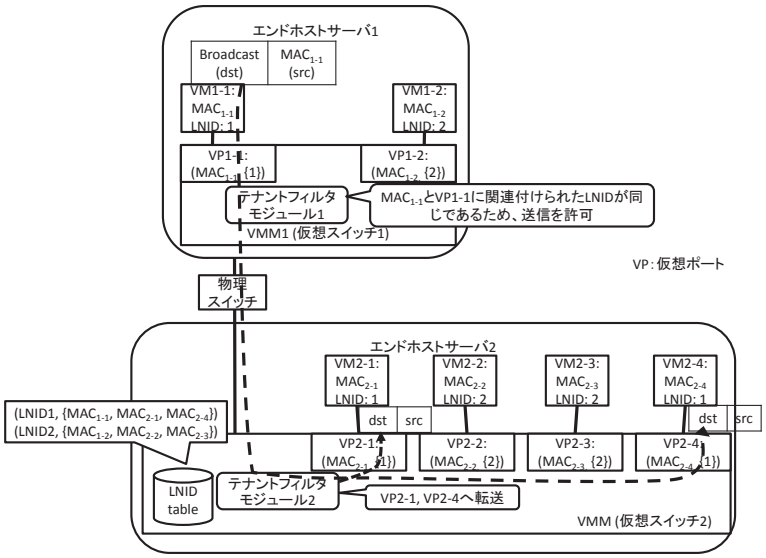

図 4　ブロードキャストフレームに対する HostVLAN フィルタリングの動作例

3.6.2　仮想ポート情報

仮想スイッチによって管理される仮想ポート毎に，それにつなげられている仮想 NIC の MAC アドレスと LNID が関連付けられる．MAC アドレスはデータ送信時の偽装データを検査するために利用される．一方，LNID はデータ受信時のフィルタリングで送信先の LNID として用いられる．

3.6.3　VM の移動に伴う分離情報の更新

VM を異なる物理サーバに移動させる場合には，すべてのエンドホストサーバの LNID table を更新する必要はない．移動先の LNID table に移動させる VM の仮想 NIC に関連付けられた LNID が含まれていれば，LNID table の更新は必要ない．LNID が含まれていなければ，管理コマンド経由で移動元エンドホストサーバと移動先エンドホストサーバで管理されている LNID table に対してのみ更新手続きが必要となる．他方，仮想ポートに関しては移動毎に更新される必要がある．

3.6.4　分離情報の更新の流れ

管理デーモンは自身の管理する LNID table の更新とともに，更新が必要となる分離情報を管理する制御デーモンへの通知も行う．

分離情報の更新手続きの流れを図 5 に示す．図 5 (a) はデータセンタの初期状態を表しており，エンドホストサーバ 1 で VM1（MAC アドレス：MAC_1，LNID：1）が稼働している．このとき，エンドホストサーバ 2 で VM2 を稼働させた後，その MAC アドレス MAC_2 と LNID 1 を関連付けるため，データセンタの管理者が管理サーバでコマンドを発行する (図 5 (b))．このとき，HostVLAN では，まず管理サーバが持つ LNID table に MAC アドレス MAC_2 と LNID 1 のペアを追加する (図 5 (c))．その後，管理デーモン が制御デーモン 2 に 2 つの LNID 情報（MAC アドレス MAC_2 と LNID 1 のペアと MAC アドレス MAC_1 と LNID 1）をエンドホストサーバ 2 の LNID table に追加するように通知する．制御デーモン 2 は，仮想ポート 2 と MAC アドレス MAC_2，LNID 1 を関連付ける．さらに，管理デーモンは，制御デーモン 1 にも追加された LNID 情報（MAC アドレス MAC_2 と LNID 1 のペア）をエンドホストサーバ 1 の LNID table に追加するように通知する．これにより，管理サーバとエンドホストサーバ 1，2 のもつ LNID table が同期され，エンドホストサーバ 2 の仮想ポート 2 に VM2 の MAC アドレスと LNID が関連付けられる（図 5 (d)）．

3.7　HostVLAN の適用範囲

HostVLAN は，MAC フレームヘッダに分離識別子を付与する必要がなく，物理ネットワーク内のルータやスイッチでは通常の MAC フレームとして見える．このため，HostVLAN は既存の L2 ネットワーク技術[4)~7)] と併用できる．また，VM の利用者は VLAN 等の L2 ネットワーク技術を併用することもできる．

HostVLAN は大規模 L2 ネットワーク[3)] のような物理スイッチで構成された単一の L2 ネットワークを想定しており，ルータを含む異なる L2 ネットワークには適用できない．異なる論理ネットワークからルータ経由で送信される L2 フレームの送信元はルータの

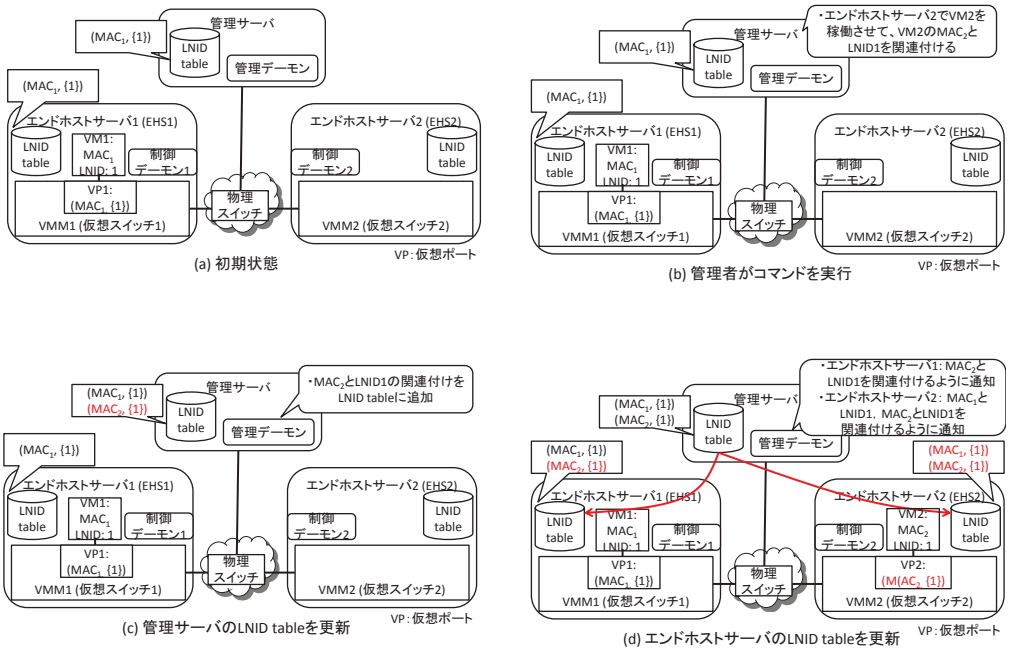

図 5　分離情報の更新の流れ

MAC アドレスとなり，異なる論理ネットワークから送信されたデータを区別できない．だたし，Virtual Routing and Forwarding (VRF) 機能 ☆を有するルータに HostVLAN を適用することで，異なる L2 ネットワーク間でもマルチテナンシを利用できる．

また，HostVLAN は MAC アドレスに基づき論理ネットワークを分離するため，異なる論理ネットワークで同じ MAC アドレスの利用を許可するデータセンタにも適用できない．既存のデータセンタの多くは，管理者によってデータセンタ 内の MAC アドレスが管理されている．この場合には，管理者がデータセンタ内の MAC アドレスを一意に設定することができ，この制約が緩和される．

4. 実　　装

4.1 LNID table のデータ構造

HostVLAN では LNID table のデータ構造として MAC アドレスをハッシュ値とするハッシュテーブルを用いる．さらに，各 MAC アドレスに対して複数の LNID を関連付けできるように，各ハッシュテーブルの要素は LNID のリストを持つ．

☆ 仮想ネットワーク（論理ネットワーク）単位で経路を制御する機能．IP-VPN 等で用いられる．

4.2 仮想ネットワーク機能の拡張

我々は KVM で利用できる 3 種類の仮想ネットワーク機能（bridging, Open vSwitch, macvtap）に対してマルチテナンシを実装した．

4.2.1 bridging への適用

bridging は IEEE 802.1d で標準化され，データリンク層で 2 つ以上のネットワークをつなげる機能である．KVM で bridging を利用する場合，仮想スイッチによるデータの受信処理が発生したときに Linux のカーネルレベルで bridging の hook が挿入される．我々は bridging の転送処理部分にフィルタリング機能を追加した．

bridging 上でマルチテナンシを実現するために，VM の仮想 NIC の LNID 情報に加えてさらに 2 つの情報が必要となる．1 つ目の情報は，仮想スイッチで管理される物理 NIC の LNID 情報である．これは，VM から外部の物理サーバへ MAC フレームを転送するときに必要となる．HostVLAN はこの転送処理時にフィルタリングを行い，送信先 MAC アドレスとして転送先仮想ポートに関連付けられた MAC アドレスを用いる．VM から外部の物理サーバへ MAC フレームが送信される場合には，この MAC アドレスが物理 NIC の MAC アドレスとなる．このため，仮想スイッチが保持する LNID table では，物理 NIC の MAC

アドレスとそれに関連付ける global LNID を含む必要がある．ただし，この情報は，仮想 NIC に関する情報のようにすべてのエンドホストサーバで管理する LNID table で共有する必要はない．各エンドホストサーバの LNID table には，そのエンドホストサーバ自身がもつ物理 NIC の情報だけ含まれていればよい．

必要となる 2 つ目の情報は仮想ポートと仮想 NIC の関係である．たとえば，VM の仮想ポートとして tap デバイスを用いる場合，tap デバイス（仮想ポート）と仮想 NIC に設定される MAC アドレスが異なる．VM からの送信データを仮想スイッチが受信したとき，MAC フレームの送信元の MAC アドレスは tap デバイスの MAC アドレスとなり，偽装データの検査により，正常なデータが破棄されてしまう．これを解決するため，HostVLAN では tap デバイス情報に仮想 NIC の MAC アドレスを付加した．

4.3 Open vSwitch への適用

Open vSwitch[15] は，openflow switching protocol[16] に基づき，MAC アドレス，IP アドレスやポート番号等の組み合わせによって定義される OpenFlow ベースでネットワークデータを制御する仮想スイッチである．Open vSwitch では Linux カーネルモジュール（datapath）を用いて，bridging 処理を OpenFlow ベースの処理で置き換える．我々は datapath によるデータ転送時にフィルタリング機能を追加した．

bridging へ適用するときと同様，2 つの LNID 情報（仮想スイッチで管理される物理 NIC の MAC アドレスと LNID，仮想ポートと仮想 NIC の関連付け）の追加が必要となる．

4.3.1 macvtap への適用

macvtap は tap 機能と bridge 機能を組み合わせた仮想ネットワーク機能である．我々は macvtap の受信処理部分にフィルタリング機能を追加した．macvtap の場合，macvtap デバイスと仮想 NIC の MAC アドレスが同じである．このため，bridging や Open vSwitch を利用する場合と異なり，仮想ポートと仮想 NIC の関連付け処理が必要ない．

また，VM からデータが送信されるとき，bridging や Open vSwitch のように，仮想スイッチの受信に伴う hook は生じない．このため，仮想スイッチで管理される物理 NIC の MAC アドレスと LNID の登録も必要ない．しかし，このことによって，同じ物理サーバ上で稼働する VM 間のデータ転送において，受信時のフィルタリングが適用されなくなってしまう．これを解決するため，我々は macvtap の送信処理時に，同じ物理サーバ上で稼働する VM へのデータ転送のときのみ，フィルタリングが実行されるようにした．

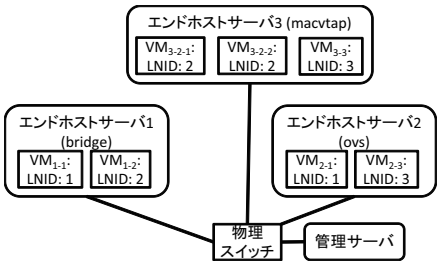

図 6 評価系

5. 評　　価

我々は VMM として KVM 0.13 を，host/guest OS カーネルとして Linux 2.6.36 を用いて HostVLAN の実装および評価を行った．さらに Open vSwitch 1.1.0pre2 も用いた．

5.1 評価系を用いた動作確認

図 6 のような評価系を用いて，ネットワークが論理的に分離できることを確認した．

この評価系では 1 つの管理サーバと 3 つのエンドホストサーバを稼働させた．エンドホストサーバ 1，エンドホストサーバ 2，エンドホストサーバ 3 上で稼働する KVM では仮想ネットワーク機能として各々，bridging (bridge)，Open vSwitch (ovs)，macvtap を利用した．

エンドホストサーバ 1 では LNID 1，2 に属する VM をそれぞれ 1 つずつ（VM_{1-1}，VM_{1-2}）稼働させた．エンドホストサーバ 2 では LNID 1，3 に属する VM をそれぞれ 1 つずつ（VM_{2-1}，VM_{2-3}）稼働させた．エンドホストサーバ 3 では LNID 2 に属する VM を 2 つ（VM_{3-2-1}，VM_{3-2-2}）と LNID 3 に属する VM を 1 つ（VM_{3-3}）稼働させた．

ここでの評価では `ping` コマンドを用いてブロードキャストフレームを各 VM から送信した．各 VM 上で `tcpdump` コマンドを用いて，仮想 NIC で受信した MAC フレームを標準出力に表示させ，異なる LNID に属する VM から MAC フレームを受信していないことを確認した．

また，初期設定後に仮想 NIC に LNID を追加した場合の動作も確認した．管理サーバから管理コマンドを用いて VM_{1-1} の LNID に 3 を追加した後，VM_{2-3} と VM_{3-3} と通信できるようになったこと確認した．

さらに，MAC アドレスの偽装に対する対応も確認した．各 VM から送信元 MAC アドレスを偽装した MAC フレームを送信し，テナントフィルタモジュールが送信された偽装 MAC フレームを検出し，破棄で

表 1　想定したデータセンタの構成

エンドホストサーバ（エンドホスト）数	8,192
エンドホストサーバ当たりの VM 数	128
テナント当たりの VM 数	64
データセンタ内のテナント数	16,384
データセンタ内の VM 数	1,048,576

きることを確認した．

5.2　実行時オーバーヘッドの計測

5.2.1　実 験 環 境

エンドホストサーバの実験環境は CPU Quad-Core Intel Xeon 2.93 GHz が 2 つ，48 GB メモリ，1 Gbps NIC である．VM は 1 つの仮想 CPU を割り当て，メモリサイズは 4 GB に設定し，仮想 NIC として virtio-net を用いた．クライアント側のベンチマークプログラムは同じスイッチとつながっている物理サーバ（クライアントサーバ）上で実行した．クライアントサーバは CPU Quad-Core Intel Xeon 2.93 GHz が 2 つ，24 GB メモリ，1 Gbps NIC である．

以下の 3 つ仮想ネットワーク機能に対して，HostVLAN の有無の場合における実行時オーバーヘッドを計測した．VMM で VLAN tagging を用いた場合には，物理 NIC のハードウェア機能により VLAN ID の追加・削除が行われる．これに伴う性能低下は非常に小さいため，ここでは VLAN tagging を用いた場合の実行時オーバーヘッドを HostVLAN の無の場合に相当するものとみなした．

bridging　bridging の利用．KVM のネットワーク I/O 性能を改善する vhost-net 機能を用いた．

Open vSwitch　Open vSwitch の利用．

macvtap　macvtap の利用．

ここでは表 1 のような，VLAN ID 数の限度である 4,094 を越す数の論理ネットワークを提供できるデータセンタを仮定し，実行時オーバーヘッドを計測した．ここでは，既存の Dell PowerEdge R610（1U ラックサーバ）のメモリサイズ 192 GB と今後のエンドホストサーバの性能向上を考慮し，エンドホストサーバのメモリサイズを 512 GB として仮定した．この仮定とこの実験で VM に割り当てたメモリサイズ 4 GB から，各エンドホストサーバ上で稼働する VM 数は 128 となる．さらに，テナント当たりの VM 数を 64 とした．これらより，このデータセンタ内では 16,384（2^{14}）の異なるテナント（LNID）を提供でき，1,048,576（2^{20}）の VM を分離できる．

各エンドホストサーバがもつ LNID table に含まれるテナントは，エンドホストサーバ上の VM が属するテナントのみでよい．表 1 では，エンドホストサーバがもつ LNID table のサイズが最大となる最悪ケースを想定し，エンドホストサーバ内で稼働する VM がすべて異なるテナントに属するように設定した．これは，エンドホストサーバがもつ LNID table に属するテナント数が 128 となることを意味する．この最悪ケースでは，エンドホストサーバのもつ LNID table に含まれる VM 数（仮想 MAC アドレス数）は，8,192（2^{13}）となる．

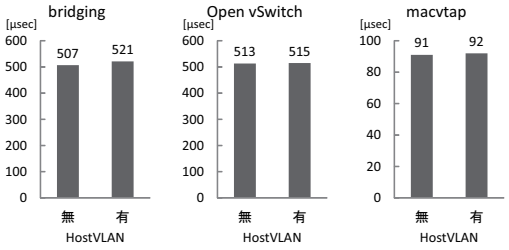

図 7　UDP 通信の遅延

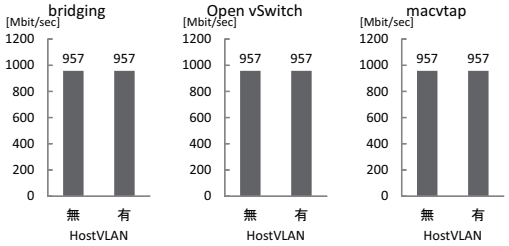

図 8　UDP 通信のスループット

この実験における LNID table のデータ構造は，仮想 MAC アドレスの下位 13 ビットをハッシュ値とするハッシュテーブルを用いた．

5.2.2　マイクロベンチマーク

上述の実験環境でネットワークの遅延とスループットを測定した．まず，*lmbench* の *lat_udp* を用いて UDP 通信の遅延を測った．次に *netperf* を用いて UDP 通信のスループット（UDP データサイズ：1470 B）を測った．lat_udp と netperf のサーバプログラムをエンドホストサーバの VM 上で，各クライアントプログラムをクライアントサーバ上で稼働させた．

図 7 に遅延に関する実験結果を示した．bridging，Open vSwitch，macvtap に対し，各々，約 3%，約 0.3%，約 1%以下の遅延のオーバーヘッドが生じた．図 8 で示されているスループットの結果から，この実験ではスループットの低下はみられなかった．

5.2.3　アプリケーションベンチマーク

クライアントサーバから *ApacheBench* を用いて，エンドホストサーバの VM 内で稼働する Web サーバ *Apache* のスループットを計測した．ここでは，

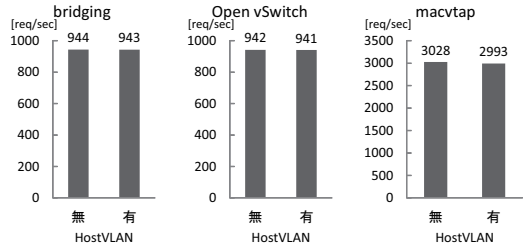

図 **9** Web サービススループット

ApacheBench から要求数が 1,024 の場合に対する静的コンテンツ（ファイルサイズ：1 KB）の要求を発行した．

図 9 に示されているように，Web サービススループットは各々，約 0.1%，0.1%，1%以下の低下で抑えることができた．アプリケーションベンチマークの性能低下はマイクロベンチマークにおける遅延とスループットを合わせた性能低下となっている．

表 1 で想定したデータセンタにエンドホストサーバを追加し，提供できる論理ネットワーク数を増加させたとしても，エンドホストサーバがもつ LNID table の大きさは変わらない．このことと上述の実験結果より，HostVLAN は VLAN ID 数の上限の 4,094 より多くの論理ネットワークを提供でき，かつネットワーク分離に伴う性能低下を小さく抑えられるため，有用なネットワーク分離技術であるとみなせる．

6. ま と め

本論文では，クラウドデータセンタで多数の論理ネットワークを提供することができるエンドホストでの技術 HostVLAN を提案した．我々は KVM の 3 種類の仮想ネットワーク機能に対してマルチテナンシを実装し，評価した．十分多くの論理ネットワークを想定した計算環境下で，HostVLAN による Web スループットを約 1%以下の低下で抑えることができた．我々はこの実験結果を実用の範囲内であるとみなしている．

参 考 文 献

1) amazon.com: *Amazon Elastic Compute Cloud*. `http://aws.amazon.com/ec2`.
2) Dunbar, L. and Hares, S.: Address Resolution for Large Data Center Problem Statement (2010).
3) Kim, C., Caesar, M. and Rexford, J.: Floodless in SEATTLE: A Scalable Ethernet Architecture for Large Enterprises, *Proc. ACM SIGCOMM*, Seattle (2009).
4) Mysore, R., Pamboris, A., Farrington, N., Huang, N., Miri, P., Radhakrishnan, S., Subramanya, V. and Vahdat, A.: PortLand: A Scalable Fault-Tolerant Layer 2 Data Center Network Fabric, *Proc. ACM SIGCOMM*, Barcelona (2009).
5) Greenberg, A., Hamilton, J., Jain, N., Kandula, S., Kim, C., Lahiri, P., Maltz, D., Patel, P. and Sengupta, S.: VL2: A Scalable and Flexible Data Center Network, *Proc. ACM SIGCOMM*, Barcelona (2009).
6) Guo, C., Lu, G., Li, D., Wu, H., Zhang, X., shi, Y., Tian, C., Zhang, Y. and Lu, S.: BCube: A High Performance, Server-centric Network Architecture for Modular Data Centers, *Proc. ACM SIGCOMM*, Barcelona (2009).
7) Guo, C., Wu, H., Tan, K., Shi, L., Zhang, Y. and Lu, S.: DCell: A Scalable and Fault-Tolerant Network Structure for Data Centers, *Proc. ACM SIGCOMM*, Seattle (2008).
8) KVM: *Kernel Based Virtual Machine*. `http://www.linux-kvm.org/page/Main_Page`.
9) VMware, I.: *VMware ESX Server 3: 802.1Q VLAN Solutions* (2006). `http://www.vmware.com/resources/techresources/412`.
10) Open vSwitch: *VLANs*. `http://openvswitch.org/?page_id=146`.
11) Wood, T., Gerber, A., Ramakrishnan, K., Shenoy, P. and Merwe, J.: The Case for Enterprise-Ready Virtual Private Clouds, *Proc. HotCloud*, San Diego (2009).
12) Hao, F., Lakshman, T., Mukherjee, S. and Song, H.: Secure Cloud Computing with a Virtualized Network Infrastructure, *Proc. HotCloud*, Boston (2010).
13) Li, L. and Woo, T.: VSITE: a scalable and secure architecture for seamless L2 enterprise extension in the cloud, *Proc. NPSec*, Kyoto (2010).
14) 菊池, 今井, 福井, 小田部: クラウドデータセンターにおけるネットワーク仮想化方式（L2 トンネル方式）の提案と評価, *SWoPP*, 金沢 (2010).
15) Pettit, J., Gross, J., Pfaff, B., Casado, M. and Crosby, S.: Virtual Switching in an Era of Advanced Edges, *Proc. DC CAVES*, Amsterdam (2010).
16) McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S. and Turner, J.: OpenFlow: Enabling Innovation in Campus Networks, *ACM SIGCOMM Computer Communication Review*, Vol. 38, No. 2, pp. 69–74 (2008).