



今わかっている NVGRE の
すべて見せます。
～デモ付き～



後藤 諭史 (Satoshi GOTO)
三井情報株式会社
Microsoft MVP - SCCDM



自己紹介

- 後藤 諭史 (Satoshi GOTO)
- 三井情報株式会社 IT技術基盤本部 クラウドサービス技術部 所属
- 仮想化製品が主な専門分野です。
 - Hyper-V や SCVMM 等々の Microsoft 仮想化製品
 - XenApp や XenDesktop といった Citrix 社製品
 - あと、ネットワーク関連もそれなりにやっています
- Microsoft MVP - System Center Cloud and Datacenter Management
(Jul.2012 - Jun.2013)

はじめに

- これからお話する Windows Server 2012 Network Virtualization の技術詳細は、おそらく（というか、確実に）氷山の一角です。
- 不明点や調査しきれていない個所があるかもしれないことを、あらかじめお詫びしておきます。
- この資料やセッション内で触れなかったこと、間違っている事柄をご存じの方がいらっしゃいましたら、是非情報交換をお願いいたします。

目的とゴール

- セッションの目的
 - Windows Server 2012の新機能である『Network Virtualization』の概要や、検証を通して確認した機能詳細を解説します。
 - SystemCenter 2012 Virtual Machine Manager SP1 の機能概要、検証を通して確認した機能詳細を解説します。
- セッションのゴール
 - 『Network Virtualization』の概要と特徴、詳細が説明できる。
 - SC2012 VMM SP1 を利用する事のメリットが説明できる。
 - （個人的なゴール）自分が持っている情報のすべてを、皆様と共有する。

アジェンダ

- NVGRE とは？
- NVGRE におけるパケットの流れ
- NVGRE におけるパケットサイズおよびフラグメンテーション処理
- PowerShell による Network Virtualization 実装
- System Center 2012 Virtual Machine Manager SP1
- Windows Network Virtualization における IP アドレス設定
- Broadcast over NVGRE
- Network Virtualization Gateway
- NVGRE ホスト側負荷評価
- まとめ
- Q & A
- Appendix A : IP Rewrite とは？（軽く）
- Appendix B : Network Virtualization の PowerShell での実装例

公式リファレンス

NVGRE draft RFC

<http://tools.ietf.org/html/draft-sridharan-virtualization-nvgre-02>

Hyper-V ネットワーク仮想化の概要

<http://technet.microsoft.com/ja-jp/library/jj134230.aspx>

Simple Hyper-V Network Virtualization Demo

<http://gallery.technet.microsoft.com/scriptcenter/Simple-Hyper-V-Network-d3efb3b8>

Simple Hyper-V Network Virtualization Script with Gateway

<http://gallery.technet.microsoft.com/scriptcenter/Simple-Hyper-V-Network-6928e91b>

MMS2013 How to Design and Configure Networking in VMM and HyperV

<http://channel9.msdn.com/Events/MMS/2013/WS-B312> (Part 1 of 2)

<http://channel9.msdn.com/Events/MMS/2013/WS-B313> (Part 2 of 2)

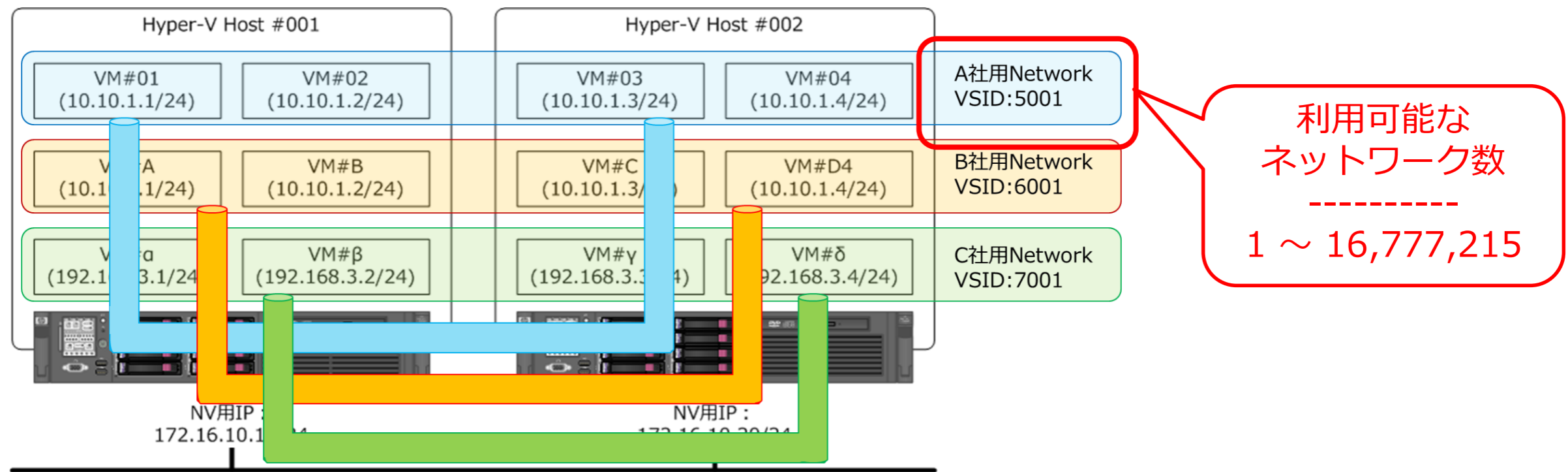
Hyper-V Network Virtualization Packet Flow

<http://www.microsoft.com/en-us/download/details.aspx?id=34782>

NVGRE とは？

NVGRE とは？

- 仮想マシンの通信（Packet）を GRE（Generic Routing Encapsulation）プロトコルでカプセル化し、物理 Network ではカプセル化した状態（GRE Packet）で通信を行う、カプセル方式のトンネル技術
- トンネル（カプセル）の識別には 24bit の Virtual Subnet ID（VSID）を使用



- アクセススイッチ（仮想化モジュール）でカプセル化処理を行う為、仮想マシンは仮想ネットワークを全く意識しない

NVGRE のポイント

- L2 over L3
 - GRE で L2 フレームをカプセル化してしまう為、オリジナルは完全に隠ぺいされる
→ 但し、GRE はカプセル化するだけであり、Packet の暗号化は行わない
 - カプセル化のオーバーヘッドは 42byte
 - Layer3 でのカプセル化である為、WAN 越えが容易
- 24 bit の Virtual Subnet ID (VSID)
 - 1-16,777,215 までの仮想ネットワークが設定可能
 - 但し、Windows Server 2012 の仕様により、使用できる VSID は 4,097 から 16,777,214 の範囲
→ 16,777,215 (FFFFFFFF) はシステムが予約しているため、使用不可
 - Packet Capture すると **Flow ID (8bit)** との組み合わせで、32 bit (4byte) の Key として表示
- 『 FlowID 』 とは？
 - マルチパス ネットワークで負荷分散を行う為の NVGRE 固有の実装
 - NVGRE 対応 Router であれば、等コストマルチパス (ECMP) バランシング可能

使い分けガイドライン（TechNet ※より）

NVGRE

- スケーラビリティに優れているため、ほとんどのシナリオに推奨
- 現在のネットワークインフラストラクチャハードウェアと互換性がある
- 1 ホストにつき 1 つの IP アドレスで済む為、スイッチの負荷が低い
- 標準ベース: RFC 2784 および 2890 と業界サポート
→ NVGRE ドラフト RFC の共同作成者:
Arista, Broadcom, Dell, Emulex, HP, Intel
- 完全な MAC ヘッダーと明示的な VSID マーキングにより、マルチテナントのトラフィック分析、メータリング、制御がサポートされる
- NVGRE 対応ハードウェアは IP Rewriteと同程度のパフォーマンスを提供する

IP Rewrite

- 現時点では、10Gbps を必要とする仮想マシンなどの高パフォーマンスシナリオに適している
- ※ NVGRE 対応ハードウェアが市販されるまで待てないという特殊なシナリオを想定

SC2012 VMM SP1 では未サポート

※ <http://technet.microsoft.com/ja-jp/library/jj134174.aspx>

NVGRE パケット構造

Outer Ethernet Header (VLAN Tag あり・ 18byte / VLAN Tag なし・ 14byte) :

送信先 MAC Address (48bit)	送信元 MAC Address (48bit)	VLAN タグ (32bit)	Ethertype (16bit)
---------------------------------	---------------------------------	----------------------	------------------------

Outer IPv4 Header (20byte) :

Version (4bit)	IHL (4bit)	ToS (8bit)	Total Length (16bit)	ID (16bit)	Flags (3bit)	Fragment Offset (13bit)	TTL (8bit)	Protocol 0x2F (8bit)	Header Checksum (16bit)	送信元 IP Address (32bit)	送信先 IP Address (32bit)
---------------------	-----------------	-----------------	------------------------------	-----------------	-------------------	---------------------------------	-----------------	---------------------------------------	---------------------------------	--------------------------------	--------------------------------

GRE Header (8byte) :

Flags and Version (16bit)	Protocol Type 0x6558 (16bit)	VSID (24bit)	FlowID (8bit)
-----------------------------------	--------------------------------------	---------------------------	----------------------------

Inner Ethernet Header :

送信先 MAC Address (48bit)	送信元 MAC Address (48bit)	Ethertype (16bit)
---------------------------------	---------------------------------	------------------------

.....

0x2F = GRE

NVGRE パケットキャプチャ

Realtek PCIe GBE Family Controller: ¥Device¥NPF_{57A4061A-5571-4639-86F9-05B1BEC85729} [Wireshark 1.8.4 ...]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
17	56.1327540	192.168.1.104	192.168.1.106	SMB2	201	SessionSetup Response, Unknown message type
18	56.1333320	192.168.1.106	192.168.1.104	SMB2	212	TreeConnect Request Tree: \\192.168.1.104\IPC\$
19	56.1335880	192.168.1.104	192.168.1.106	SMB2	180	TreeConnect Response
20	56.1341380	192.168.1.106	192.168.1.104	SMB2	258	Ioctl Request DFS Function:0x0065, File: \

Frame 18: 212 bytes on wire (1696 bits), 212 bytes captured (1696 bits) on interface 0

Ethernet II, Src: Intel_ (68:05:ca:), Dst: Cisco_ (00:18:19:)

Internet Protocol Version 4, Src: 10.1.2.107 (10.1.2.107), Dst: 10.1.1.143 (10.1.1.143)

Version: 4
Header length: 20 bytes

Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not F...
Total Length: 198
Identification: 0x0700 (1792)
Flags: 0x02 (Don't Fragment)
Fragment offset: 0
Time to live: 128
Protocol: GRE (47)

Header checksum: 0xdb0d [correct]
Source: 10.1.2.107 (10.1.2.107)
Destination: 10.1.1.143 (10.1.1.143)
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]

Generic Routing Encapsulation (Transparent Bridging)

Flags and Version: 0x2000
Protocol Type: Transparent Ethernet bridging (0x6558)

Key: 0x1bb1ce71

Ethernet II, Src: Microsof_b7:1c:16 (00:1d:d8:b7:1c:16), Dst: Microsof_b7:1c:12 (00:1d:d8:b7:1c:12)

Internet Protocol Version 4, Src: 192.168.1.106 (192.168.1.106), Dst: 192.168.1.104 (192.168.1.104)

0000 00 10 15 01 51 00 00 00 00 00 00 00 00 00 00 ...@.../...k..
0010 00 c6 07 00 40 00 80 2f db 0d 0a 01 02 6b 0a 01 ...@.../...k..
0020 01 8f 20 00 65 58 1b b1 ce 71 00 1d d8 b7 1c 12 ...ex...q.....
0030 00 1d d8 b7 1c 16 08 00 45 00 00 9c 00 af 40 00E.....@..
0040 80 06 75 8a c0 a8 01 6a c0 a8 01 68 c0 05 01 bd ...u...j...h....
0050 14 aa 63 16 c7 7d 27 0a 50 18 01 fe 47 1e 00 00 ...C...}'...P...G...

The Key field contains a four octet num... Packets: 119 Displayed: 119 Marke... Profile: Default

Key: 0x1bb1ce71

VSID
(1814990)

FlowID

NVGRE パケット構造：注意点

No.	Time	Source	Destination	Protocol	Length	Info
77	2013-03-12 19:57:02.279892000	192.168.1.106	192.168.1.104	TCP	108	49157 > microsoft-ds [SYN] Seq=
78	2013-03-12 19:57:02.280178000	10.1.1.143	10.1.2.107	ICMP	178	Destination unreachable (Host a

+	Frame 77: 108 bytes on wire (864 bits), 108 bytes captured (864 bits) on interf
+	Ethernet II, Src: Intel_ (68:05:ca:), Dst: Cisco_
+	Internet Protocol Version 4, Src: 10.1.2.107 (10.1.2.107), Dst: 192.168.1.104 (192.168.1.104)
-	Generic Routing Encapsulation (Transparent Ethernet bridging)
+	Flags and version: 0x2000
	Protocol Type: Transparent Ethernet bridging (0x6558)
	Key: 0xa81bb1ce
+	Ethernet II, Src: Microsof_b7:1c:16 (00:1d:d8:b7:1c:16), Dst: Microsof_b7:1c:12 (00:1d:d8:b7:1c:12)
+	Internet Protocol Version 4, Src: 192.168.1.106 (192.168.1.106), Dst: 192.168.1.104 (192.168.1.104)

KB2779768 適用前

No.	Time	Source	Destination	Protocol	Length	Info
17	2013-03-12 21:16:04.091083000	192.168.1.104	192.168.1.106	SMB2	201	SessionSetup Response, Unknown
18	2013-03-12 21:16:04.091661000	192.168.1.106	192.168.1.104	SMB2		

+	Frame 18: 212 bytes on wire (1696 bits), 212 bytes captured (1696 bits) on i
+	Ethernet II, Src: Intel_ (68:05:ca:), Dst: Cisco_
+	Internet Protocol Version 4, Src: 10.1.2.107 (10.1.2.107), Dst: 10.1.1.143 (10.1.1.143)
-	Generic Routing Encapsulation (Transparent Ethernet bridging)
+	Flags and version: 0x2000
	Protocol Type: Transparent Ethernet bridging (0x6558)
	Key: 0x1bb1ce71
+	Ethernet II, Src: Microsof_b7:1c:16 (00:1d:d8:b7:1c:16), Dst: Microsof_b7:1c:12 (00:1d:d8:b7:1c:12)
+	Internet Protocol Version 4, Src: 192.168.1.106 (192.168.1.106), Dst: 192.168.1.104 (192.168.1.104)

KB2779768 適用後

KB2779768 のポイント

- KB2779768 を適用すると、GRE Header （ 8byte ） の Format が RFC Draft 準拠に変更されます
 - KB2779768 は 2012/12/15 に Windows Update サイトに登録された模様
 - 『 Wnv.sys 』 『 Wnvapi.dll 』 というファイルが更新されます
 - KB2779768 で修正された内容が書かれた KB は見つかりませんでした
 - KB2779768 で置き換わるファイルのリスト → <http://support.microsoft.com/kb/2791465>

- KB2779768 が適用済みホストと未適用ホスト間では NVGRE 通信不可
 - icmp Type3 Code10 （ Destination host administratively prohibited ） が通知され、通信不可

```
Internet Control Message Protocol
Type: 3 (Destination unreachable)
Code: 10 (Host administratively prohibited)
Checksum: 0xb02a [correct]
```

- これから検証を開始する場合、 3rd Party 実装の NVGRE 対応機器と接続試験をする場合、
必ず最新のパッチを適用してから開始してください

KB2779768 のポイント

- TechEd 2013 NA : How to Design and Configure Networking in Microsoft System Center - Virtual Machine Manager and HyperV (Part 2 of 2) より

Network fabric configuration



- Enabling network virtualization
 - WS 2012 R2 no longer requires NV filter enablement
- Configuring provider address space
 - Must have static IP pool
 - Must enable network virtualization on logical network for provider addresses

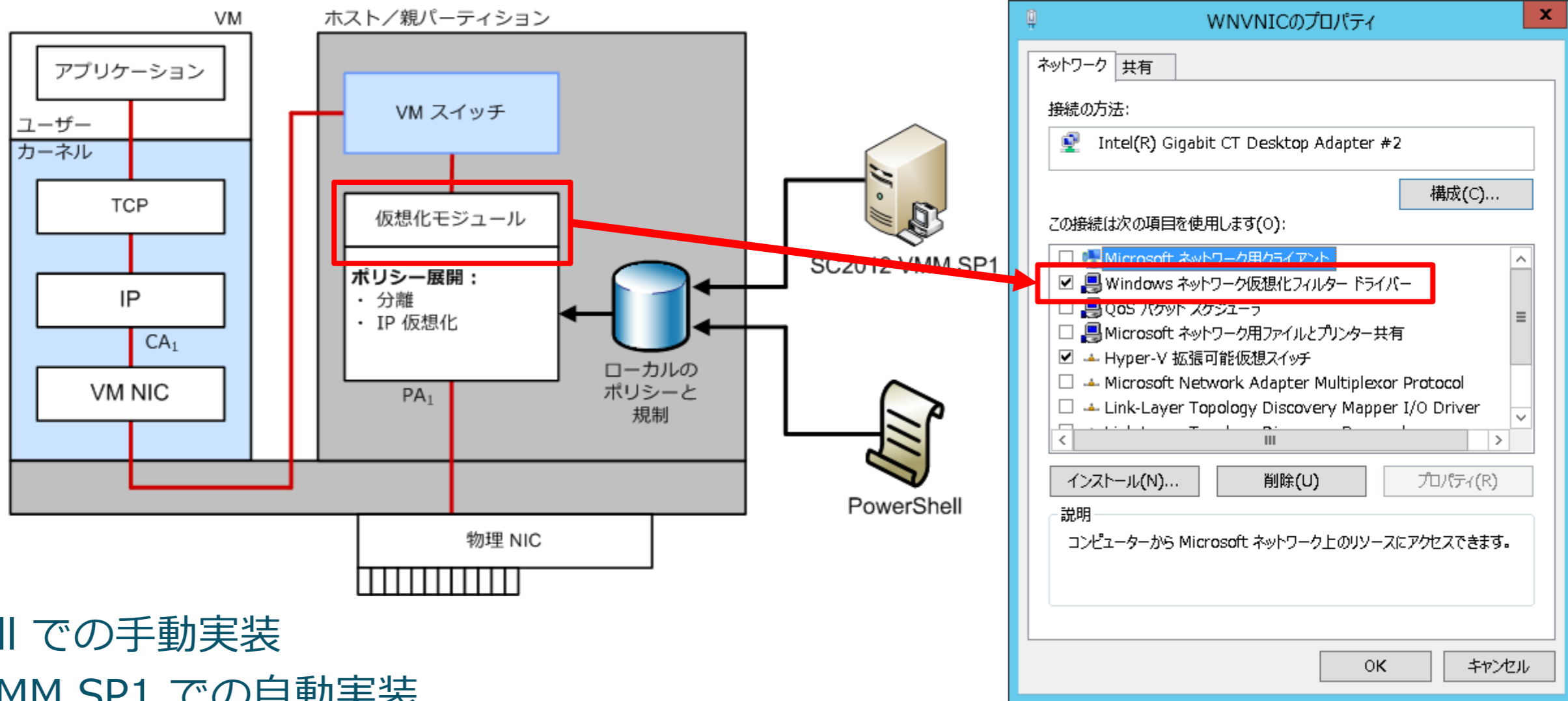
If mixing 2012 and 2012 R2 hosts, must have KB2779768 on 2012 hosts

- Windows Server 2012 R2 と混在させる場合にも、**必ず最新のパッチを適用してください**

Network Virtualization 用語の整理

CustomerAddress (CA)	仮想マシンの IP Address 。 テナントの IP Address とも。
ProviderAddress (PA)	トンネリング通信の終端 IP Address 。 データセンター内の IP Address とも。
VirtualSubnetID (VSID)	Network Virtualization における同一セグメントの範囲 (Virtual Subnet) を表す ID 。 古いRFC Draft (Ver.00) では『 Tenant Network ID 』と表記されている。
RoutingDomainID (RDID)	ルーティング可能 (パケット交換可能) な範囲を表す ID 。 VirtualSubnetID が異なっても、RoutingDomainID が同一であれば通 信可能。 同一 Network (同一の テナント) かを識別する ID といいかえる事も可能。

アーキテクチャー（ TechNet ※より）



- PowerShell での手動実装
- SC2012 VMM SP1 での自動実装

→ Software Defined Networking（SDN）

※ <http://technet.microsoft.com/ja-jp/library/jj134174.aspx>

【参考】 SDN を簡単に……（１）

- Software Defined Networking の略
 - ネットワークの構成をプログラム（＝ソフトウェア）で定義する、という思想／概念
 - 個々のネットワーク機器それぞれをコンフィグレーションするのではなく、ネットワーク全体の構成やトラフィックフローを統一されたプログラム手法で構成／管理してしまおうという仕組み
- 具体的な実装例としては、最近有名な『 OpenFlow 』
 - 但し、SDN は概念であり、 OpenFlow は実装の一形態である為イコールではない
- NVGRE を用いて、SC2012 VMM で『ネットワークを』『ソフトウェア的に』『定義できる』ので、 NVGRE + SC2012 VMM は SDN の実装の一つである

【参考】 SDN を簡単に……（２）

- SDN には『オーバーレイ型』と『ホップバイホップ型』の二種類がある
- 『ホップバイホップ型』の代表例が『 OpenFlow 』
 - 『ホップバイホップ型』は途中経路の Router / Switch に至る全ての Network 機器が対応している必要がある
 - OpenFlow でいうと、Network 機器の全てが OpenFlow を喋れる必要がある
 - 導入するには、既存機器のリプレイス（もしくは対応 OS への入れ替え）が必要
 - **実は、ものすごく敷居が高い**
- Windows Server 2012 の Network Virtualization は『オーバーレイ型』
 - 『オーバーレイ型』では NVE （ Network Virtualization Endpoint ）で Network Virtualization （カプセル化）が行われる為、途中経路は NVGRE に『必ずしも』対応している必要なし
 - 対応していれば、 ECMP のような高付加機能が利用可能
 - 従来の L3 Network にそのままボルトオン可能
 - 『ホップバイホップ型』に比べて、**低コストで導入可能**

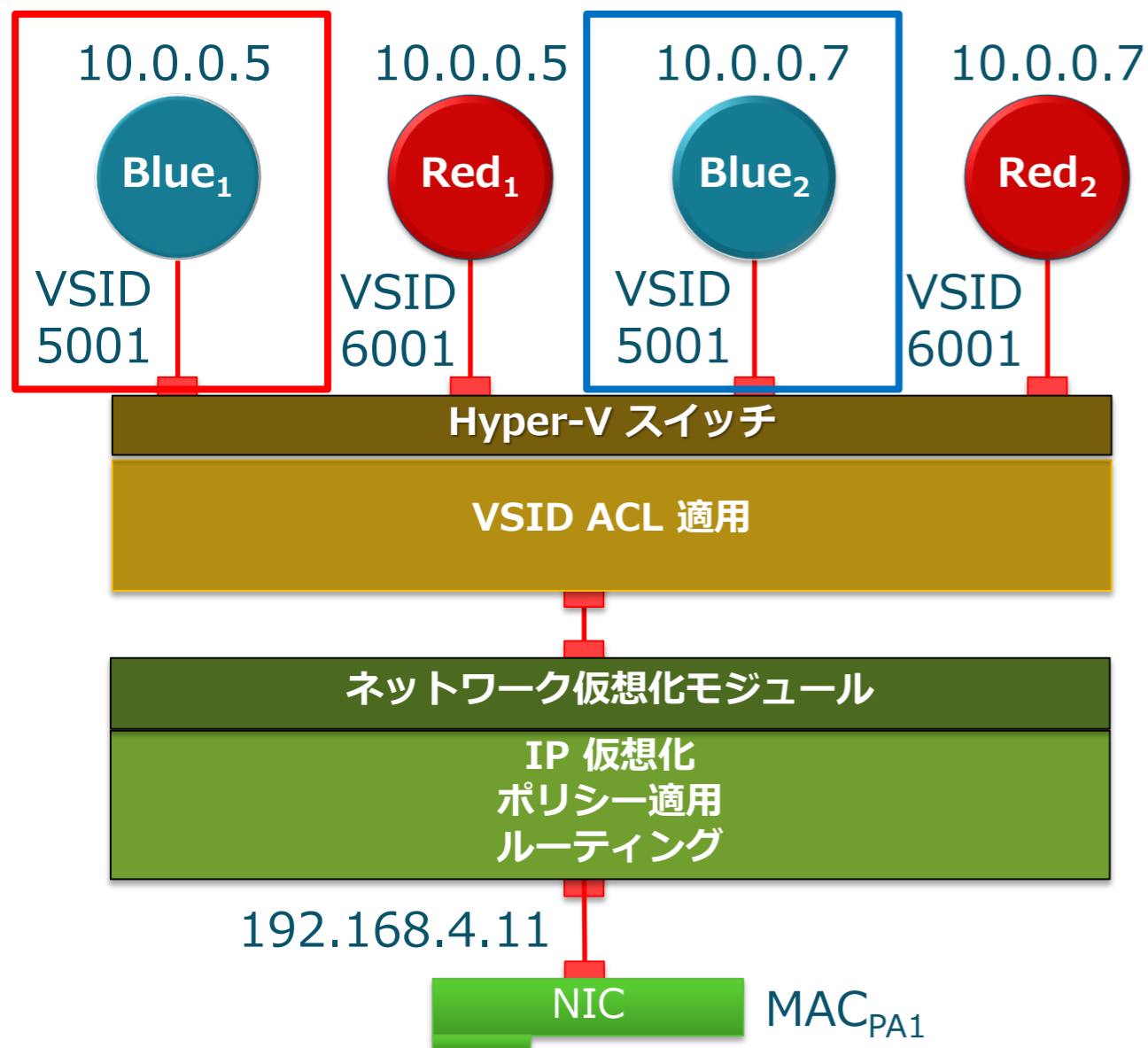
NVGRE におけるパケットの流れ

※オリジナル（英語）

<http://www.microsoft.com/en-us/download/details.aspx?id=34782>

同じサブネットで、同じホストの場合

パケットの流れ：Blue1 から Blue2



10.0.0.7 と通信したい

ARP リクエスト：10.0.0.7



Hyper-V スイッチがARPを転送
(ブロードキャスト転送)

1. VSID 5001 に属するローカル VM
2. ネットワーク仮想化モジュール

Blue₁ が Blue₂ の MAC を学習

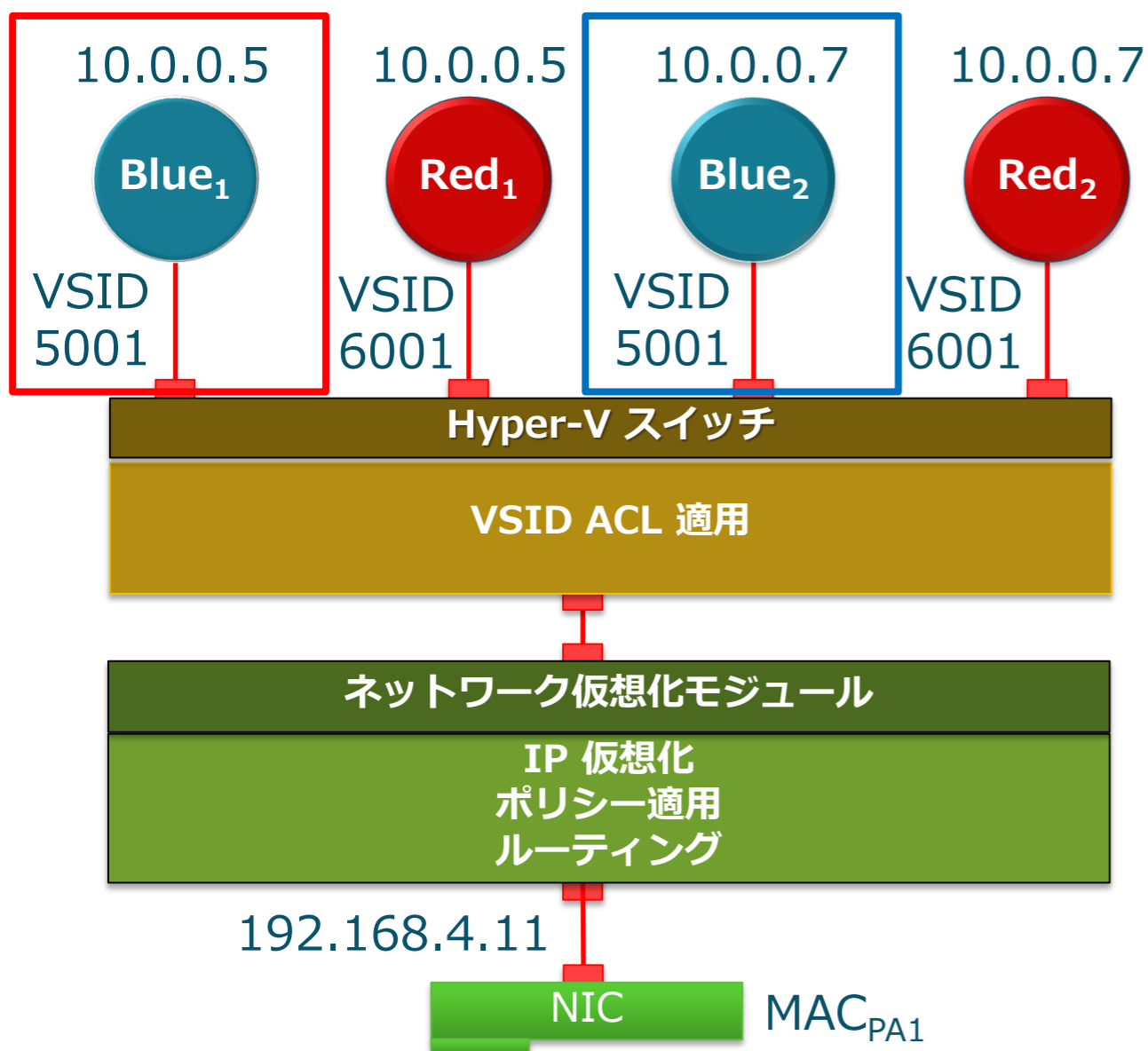
10.0.0.7 is at MAC_{B2}



Blue₂ がARP リプライ
IP 10.0.0.7 is at Blue₂ MAC
(VSID 5001)



パケットの流れ：Blue1 から Blue2（送信）



Blue₁ から送信

MAC_{B1}→MAC_{B2} | 10.0.0.5 → 10.0.0.7



Hyper-V Switch 受信

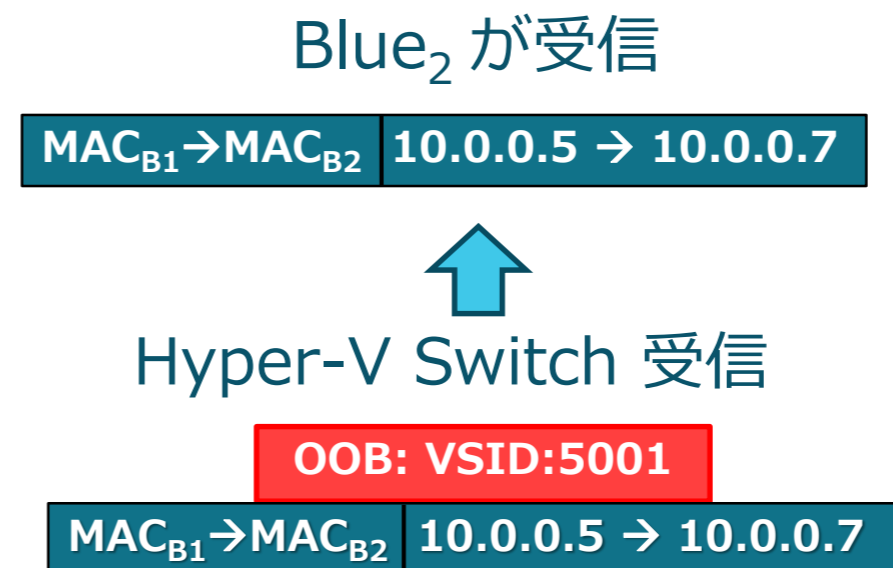
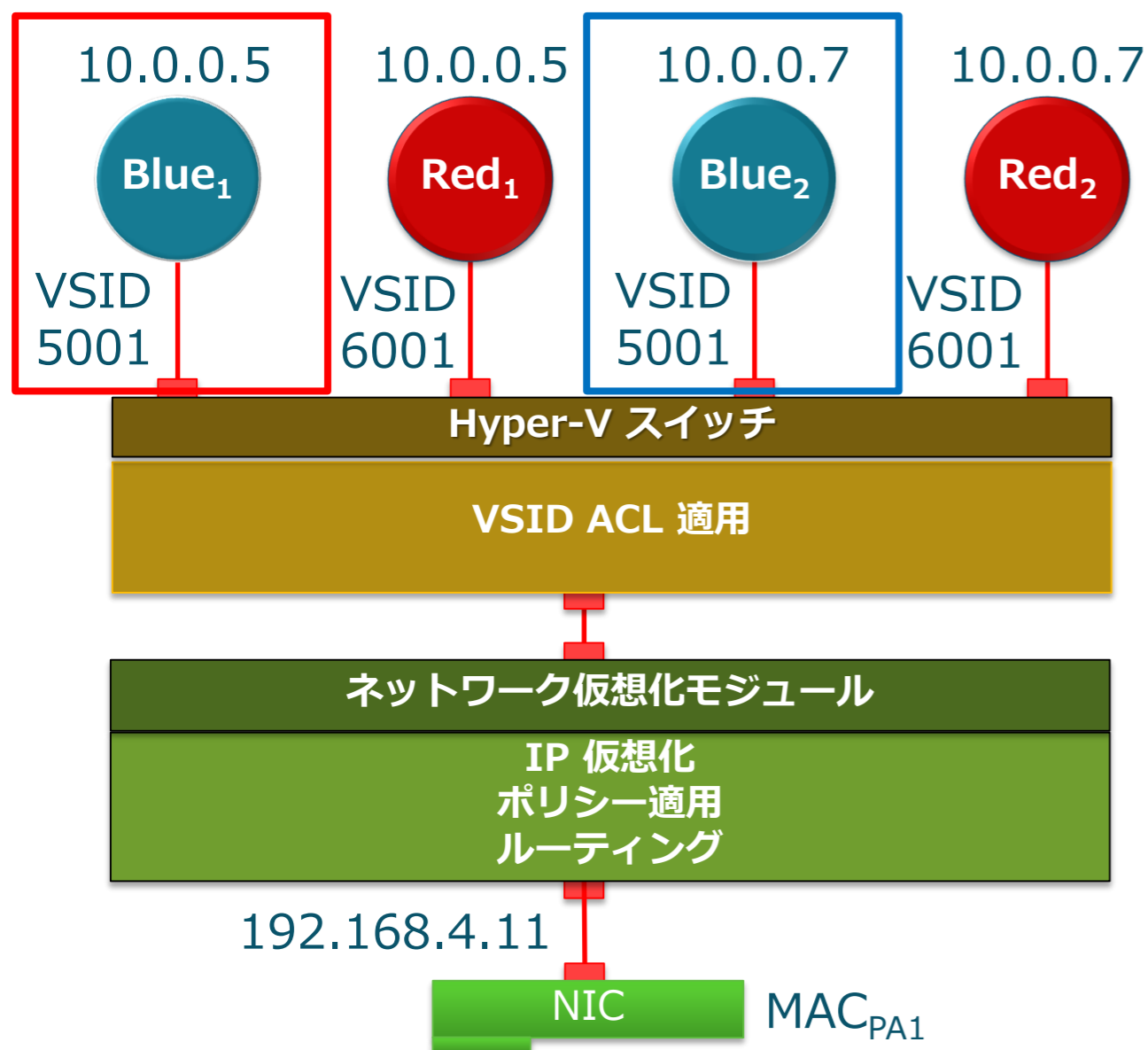
OOB: VSID:5001

MAC_{B1}→MAC_{B2} | 10.0.0.5 → 10.0.0.7

- OOB データ = 帯域外データ

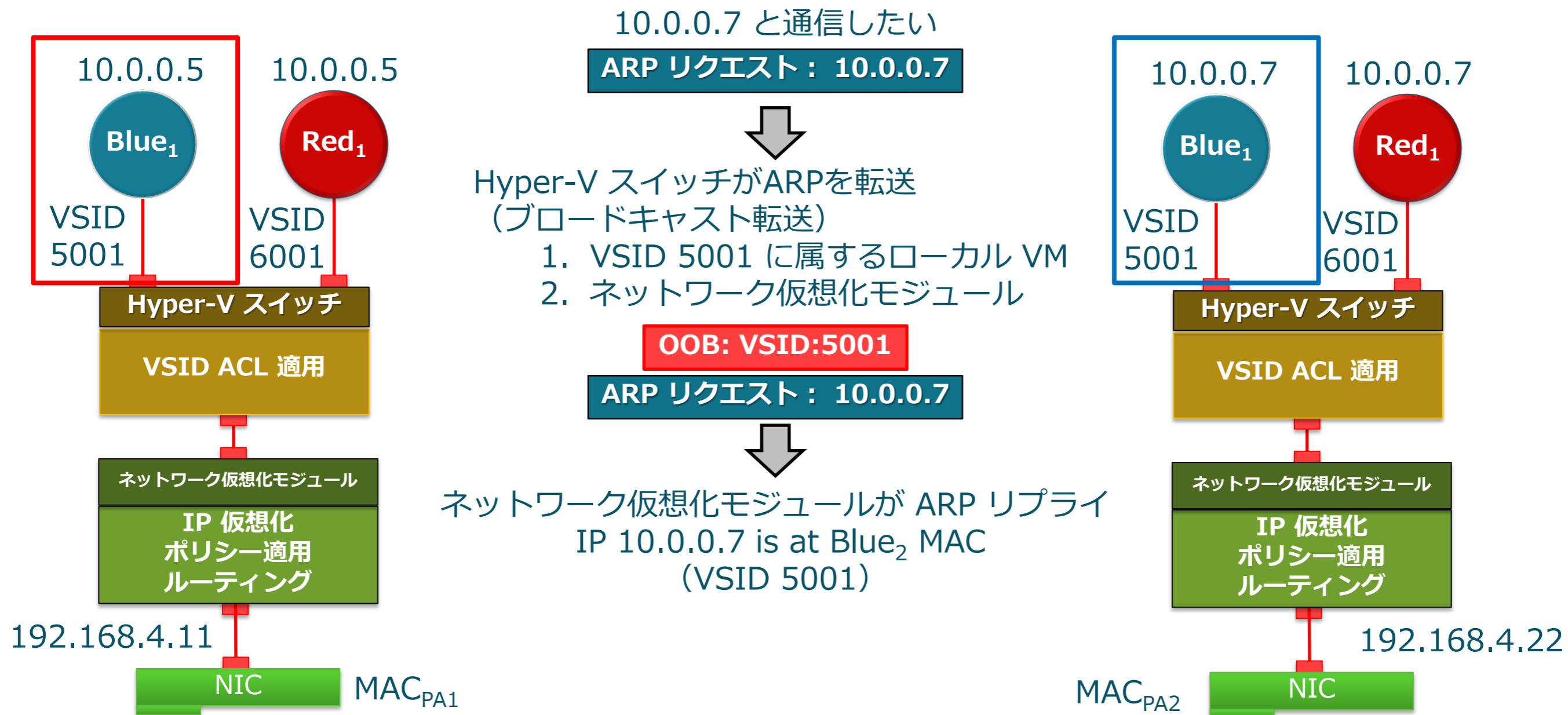
- パケットの外にあって、パケットに関連付けられたデータ
- 仮想化フィルターと Hyper-V スイッチの間での、パケットの識別に用いられる

パケットの流れ：Blue1 から Blue2（受信）

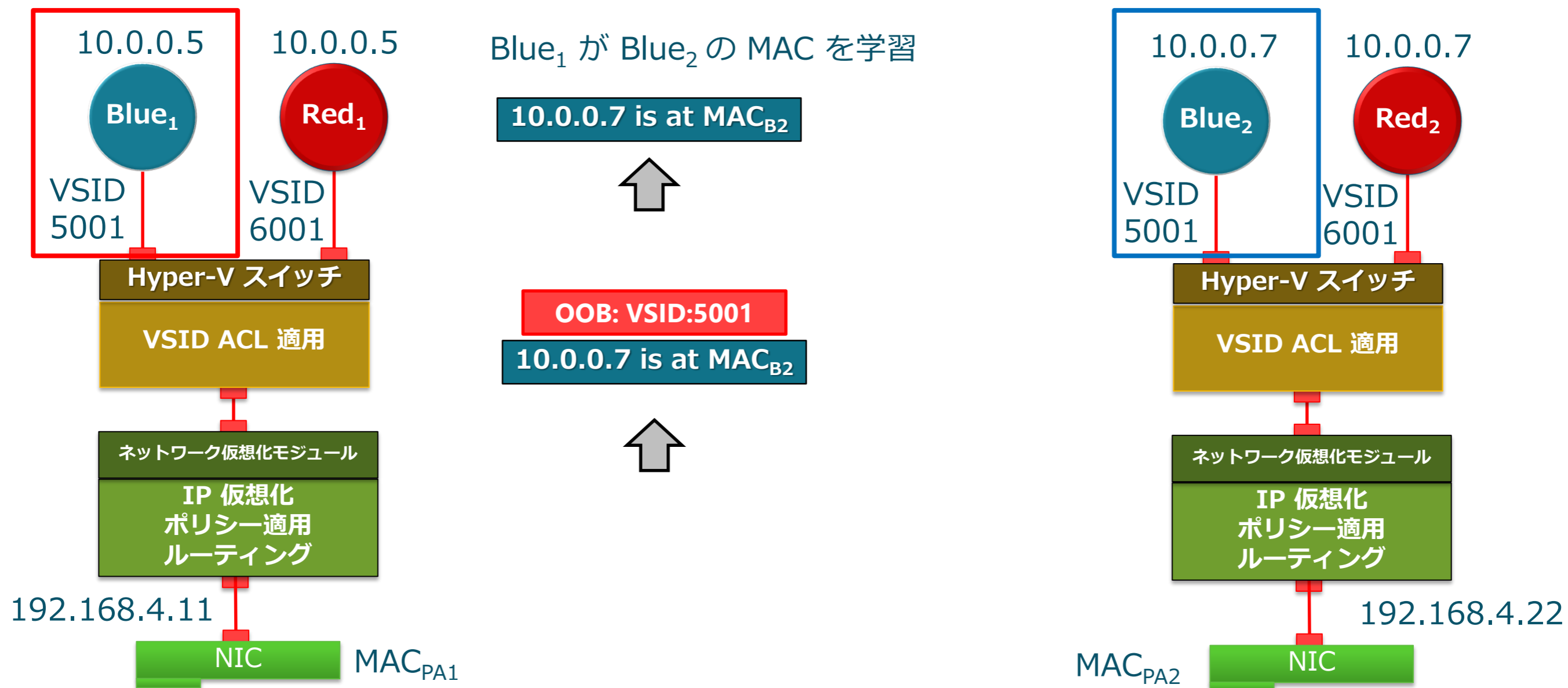


同じサブネットで、異なるホストの場合

パケットの流れ：Blue1 から Blue2

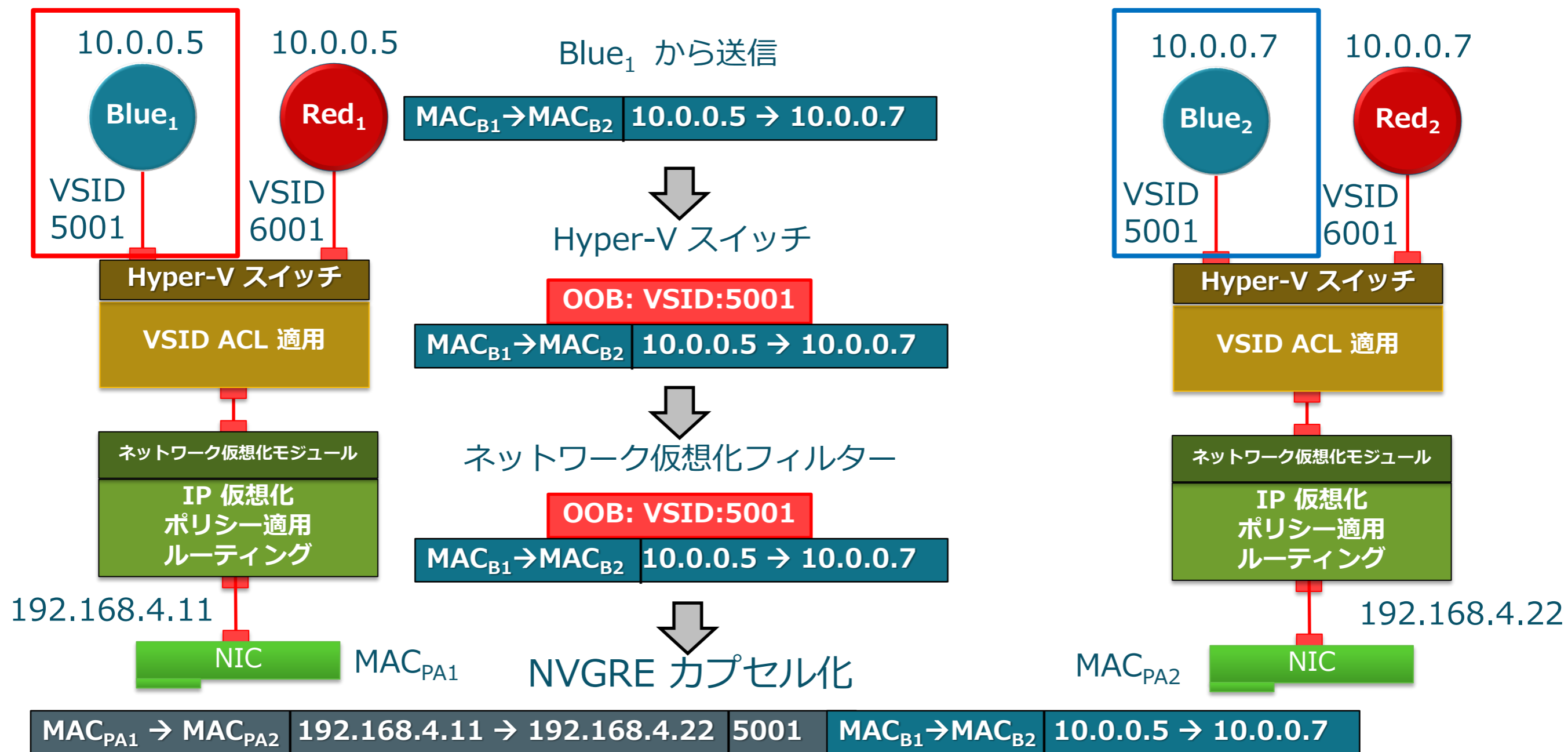


パケットの流れ：Blue1 から Blue2

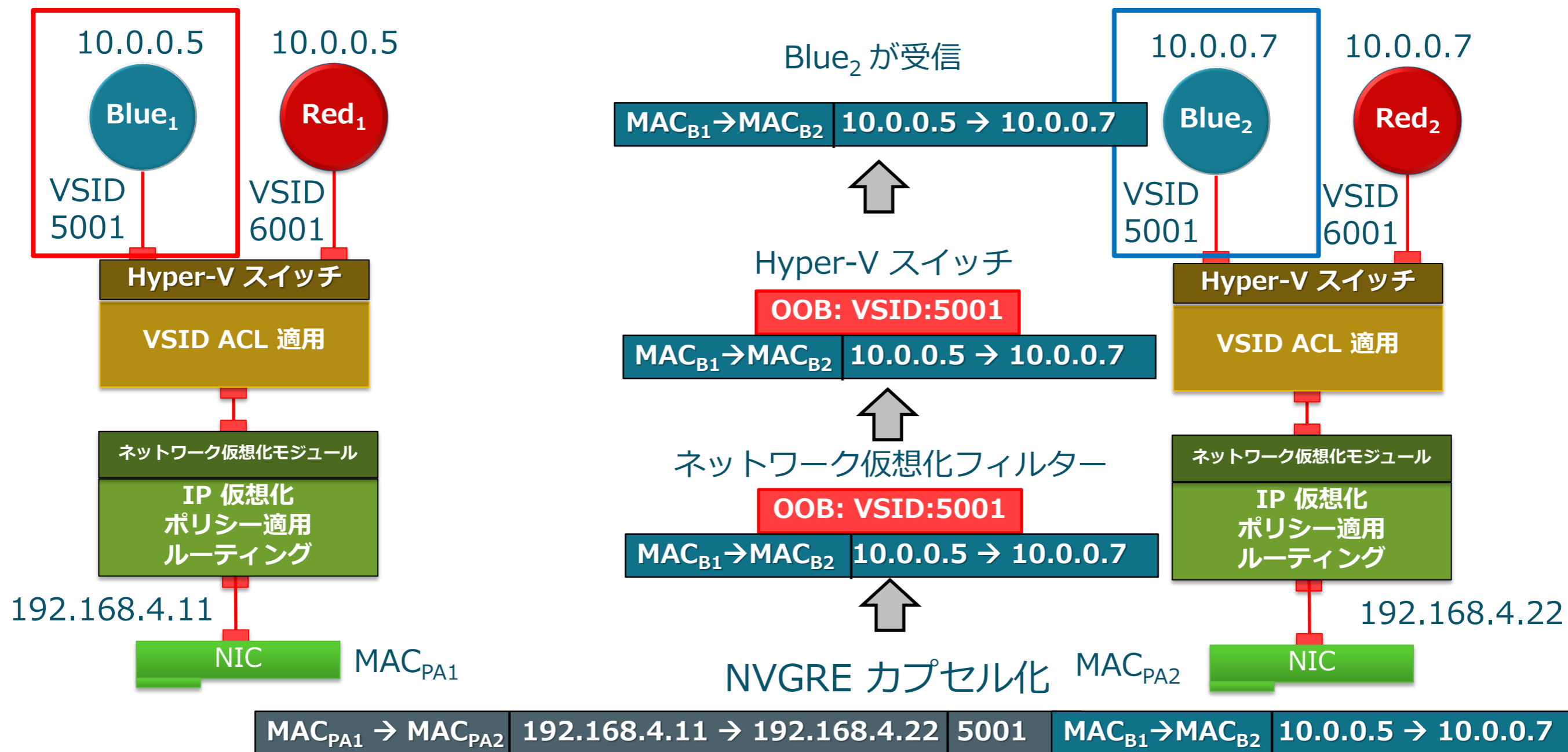


ARP パケットは物理ネットワークにブロードキャストされません

パケットの流れ：Blue1 から Blue2

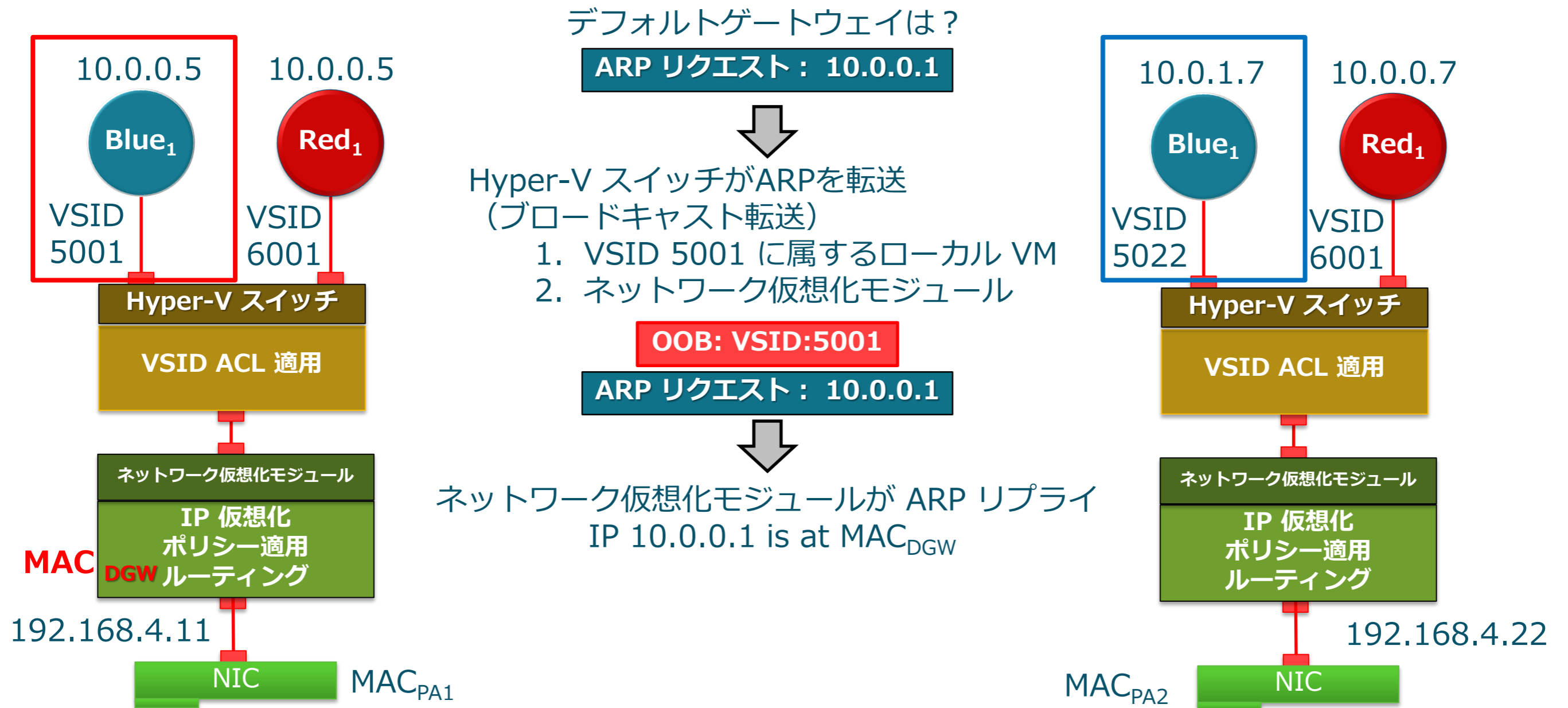


パケットの流れ：Blue1 から Blue2



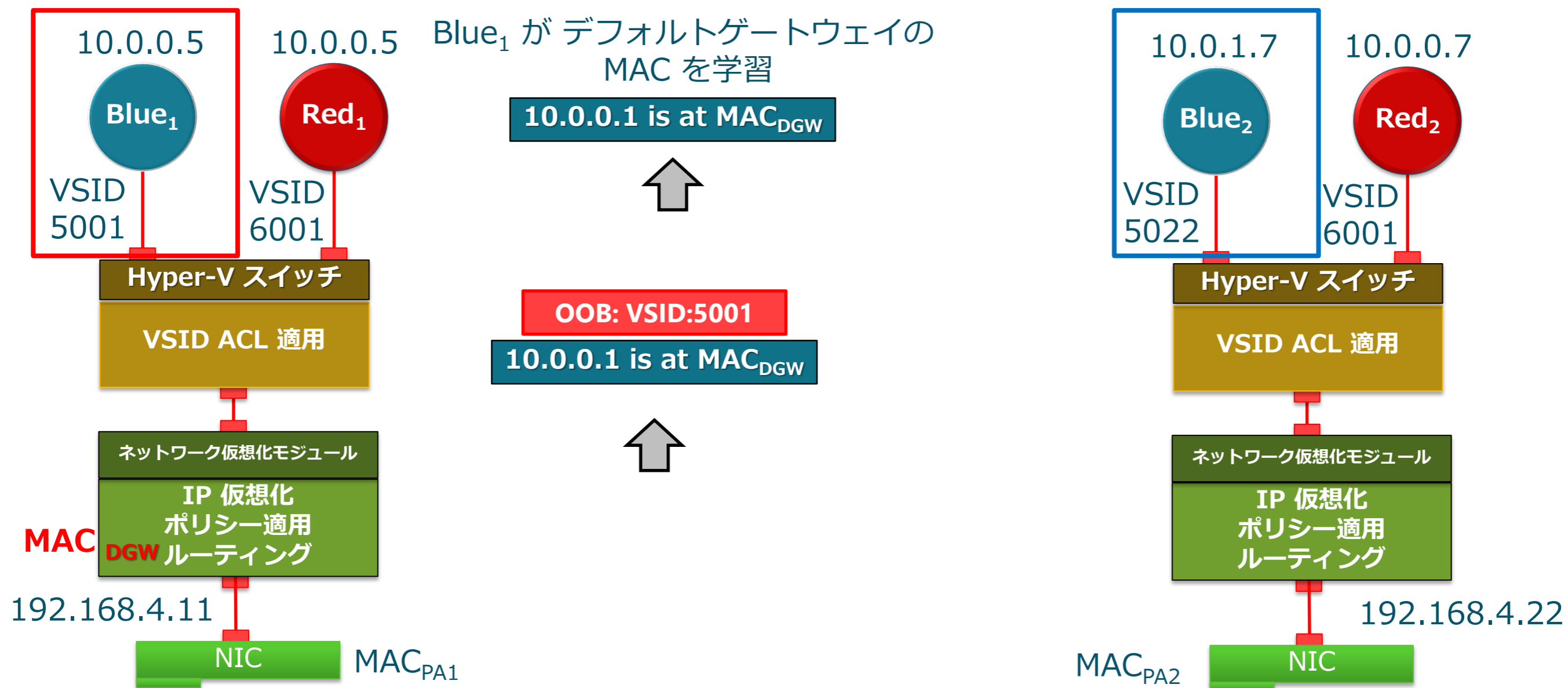
**異なるサブネット（同じ RDID ）で、
異なるホストの場合**

パケットの流れ：Blue1 から Blue2



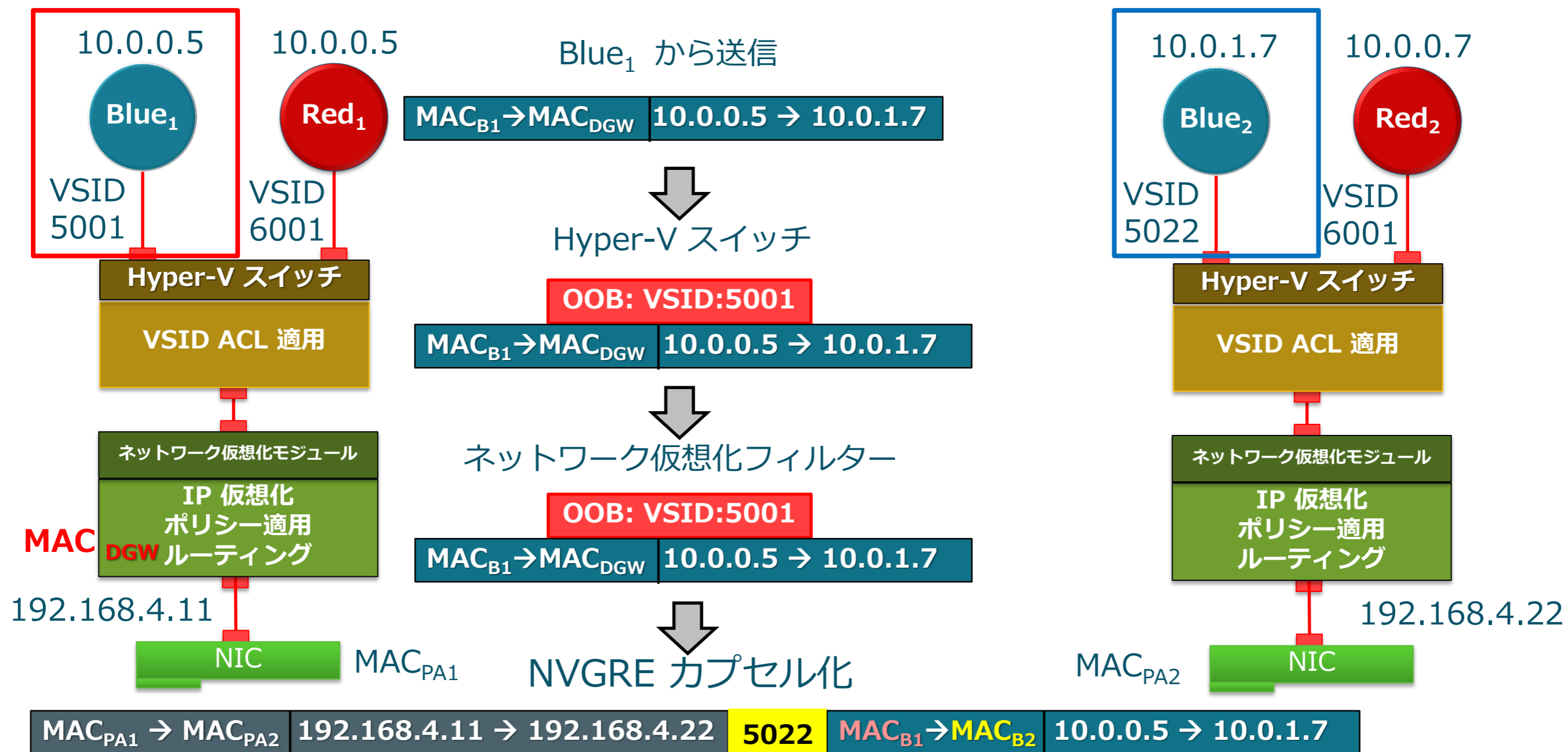
ARP パケットは物理ネットワークにブロードキャストされません

パケットの流れ：Blue1 から Blue2

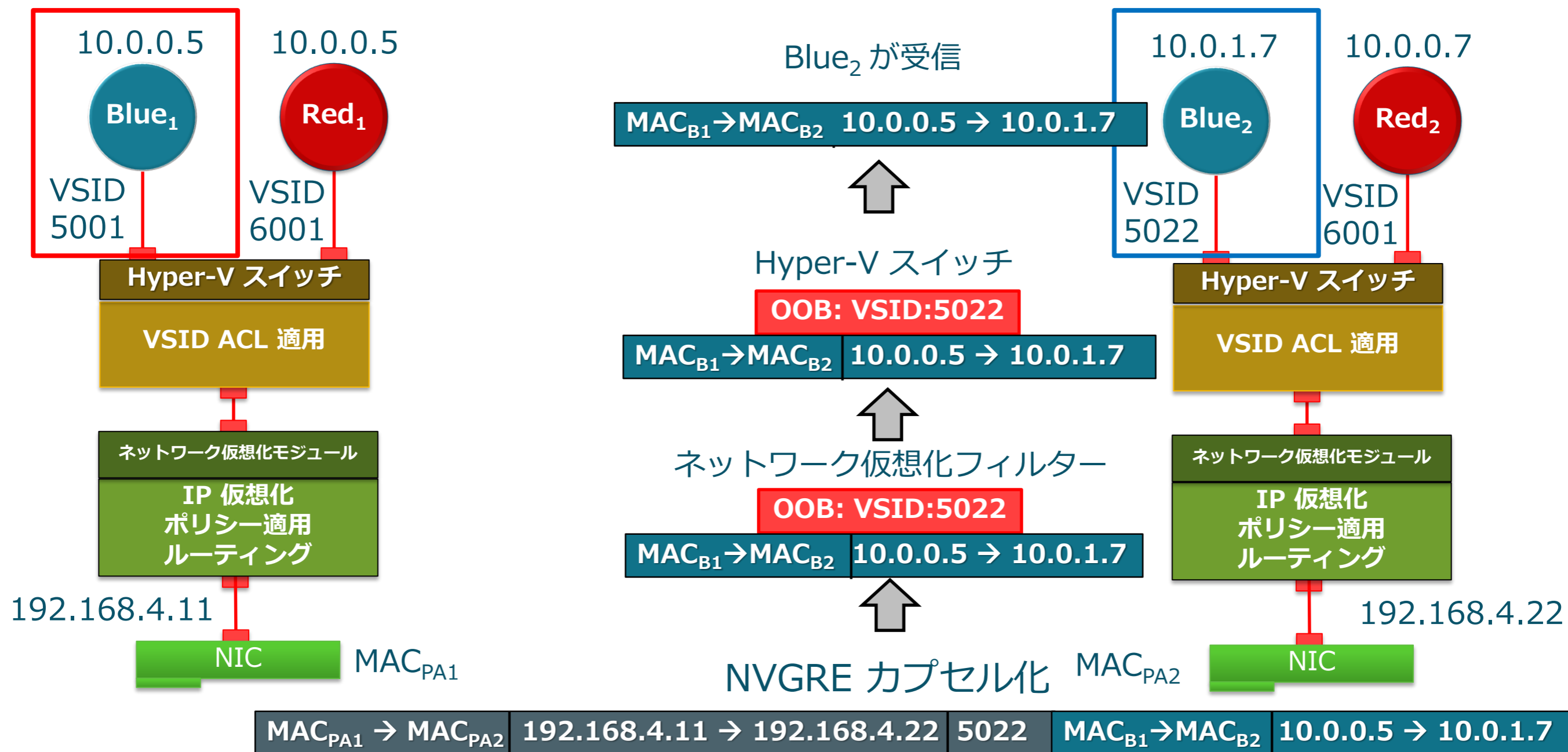


ARP パケットは物理ネットワークにブロードキャストされません

パケットの流れ：Blue1 から Blue2



パケットの流れ：Blue1 から Blue2



MAC アドレスの状態（ex：物理ホスト間の通信）

No.	Time	Source	Destination	Protocol	Length	Info
42	9.92468600	192.168.250.10	10.10.1.10	SMB2	160	TreeConnect Request Tree: \\10.10.1.10\c\$
43	9.92535800	10.10.1.10	192.168.250.10	SMB2	138	TreeConnect Response
44	9.92554900	192.168.250.10	10.10.1.10	SMB2	210	Ioctl Request NETWORK_FILE_SYSTEM Function: C
45	9.92609000	10.10.1.10	192.168.250.10	SMB2	131	Ioctl Response, Error: STATUS_FILE_C
46	9.92666200	192.168.250.10	10.10.1.10	SMB2	234	Create Request File:

+	Frame 42: 160 bytes on wire (1280 bits), 160 bytes captured (1280 bits) on interface 0
-	Ethernet II, Src: Hewlett-_e1:c4:ef (2c:27:d7:e1:c4:ef), Dst: Cisco_df:9f:95 (00:18:19:df:9f:95)
+	Destination: Cisco_df:9f:95 (00:18:19:df:9f:95)
+	Source: Hewlett-_e1:c4:ef (2c:27:d7:e1:c4:ef)
	Type: IP (0x0800)
+	Internet Protocol Version 4, Src: 192.168.250.10 (192.168.250.10), Dst: 10.10.1.10 (10.10.1.10)
+	Transmission Control Protocol, Src Port: 54382 (54382), Dst Port: microsoft-ds (445), Seq: 998, Ack: 1221, Len: 1
+	NetBIOS Session Service
+	SMB2 (Server Message Block Protocol version 2)

送信先はルーター MAC アドレス

サブネット越え通信：送信側

送信元がルーターの MAC アドレス

No.	Time	Source	Destination	Protocol	Length	Info
31	12.34567800	192.168.250.10	10.10.1.10	SMB2	160	TreeConnect Request Tree: \\10.10.1.10\c\$
32	12.34567800	10.10.1.10	192.168.250.10	SMB2	138	TreeConnect Response
33	12.34567800	192.168.250.10	10.10.1.10	SMB2	210	Ioctl Request NETWORK_FILE_SYSTEM Function: C
34	12.34567800	10.10.1.10	192.168.250.10	SMB2	131	Ioctl Response, Error: STATUS_FILE_CLOSED
35	12.34567800	192.168.250.10	10.10.1.10	SMB2	234	Create Request File:

+	Frame 29: 160 bytes on wire (1280 bits), 160 bytes captured (1280 bits) on interface 0
-	Ethernet II, Src: Cisco_df:9f:94 (00:18:19:df:9f:94), Dst: Aopen_60:e4:21 (00:01:80:60:e4:21)
+	Destination: Aopen_60:e4:21 (00:01:80:60:e4:21)
+	Source: Cisco_df:9f:94 (00:18:19:df:9f:94)
	Type: IP (0x0800)
+	Internet Protocol Version 4, Src: 192.168.250.10 (192.168.250.10), Dst: 10.10.1.10 (10.10.1.10)
+	Transmission Control Protocol, Src Port: 54382 (54382), Dst Port: microsoft-ds (445), Seq: 1070, Ack: 1293, Len: 1
+	NetBIOS Session Service
+	SMB2 (Server Message Block Protocol version 2)

サブネット越え通信：受信側

MAC アドレスの状態（ex：物理ホスト間の通信）

No.	Time	Source	Destination	Protocol	Length	Info
42	9.92468600	192.168.250.10	10.10.1.10	SMB2	160	TreeConnect Request Tree: \\10.10.1.10\c\$
43	9.92535800	10.10.1.10	192.168.250.10	SMB2	138	TreeConnect Response
44	9.92554900	192.168.250.10	10.10.1.10	SMB2	210	Ioctl Request NETWORK_FILE_SYSTEM Function:0
45	9.92609000	10.10.1.10	192.168.250.10	SMB2	131	Ioctl Response, Error: STATUS_FILE_CLOSED
46	9.92666200	192.168.250.10	10.10.1.10	SMB2	234	Create Request File:

+	Frame 42: 160 bytes on wire (1280 bits), 160 bytes captured (1280 bits) on interface 0
-	Ethernet II, Src: Hewlett-_e1:c4:ef (2c:27:d7:e1:c4:ef), Dst: Cisco_df:9f:95 (00:18:19:df:9f:95)
+	Destination: Cisco_df:9f:95 (00:18:19:df:9f:95)
+	Source: Hewlett-_e1:c4:ef (2c:27:d7:e1:c4:ef)
	Type: IP (0x0800)
+	Internet Protocol Version 4, Src: 192.168.250.10 (192.168.250.10), Dst: 10.10.1.10 (10.10.1.10)
+	Transmission Control Protocol, Src Port: 54382 (54382), Dst Port: microsoft-ds (445), Seq: 998, Ack: 1221, Len: 1
+	NetBIOS Session Service
+	SMB2 (Server Message Block Protocol version 2)

送信先はルーター MAC アドレス

送信パケット

送信元がルーターの MAC アドレス

No.	Time	Source	Destination	Protocol	Length	Info
44	9.92554900	192.168.250.10	10.10.1.10	SMB2	210	Ioctl Request NETWORK_FILE_SYSTEM Function:0
45	9.92609000	10.10.1.10	192.168.250.10	SMB2	131	Ioctl Response, Error: STATUS_FILE_CLOSED
46	9.92666200	192.168.250.10	10.10.1.10	SMB2	234	Create Request File:

+	Frame 43: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits) on interface 0
-	Ethernet II, Src: Cisco_df:9f:95 (00:18:19:df:9f:95), Dst: Hewlett-_e1:c4:ef (2c:27:d7:e1:c4:ef)
+	Destination: Hewlett-_e1:c4:ef (2c:27:d7:e1:c4:ef)
+	Source: Cisco_df:9f:95 (00:18:19:df:9f:95)
	Type: IP (0x0800)
+	Internet Protocol Version 4, Src: 10.10.1.10 (10.10.1.10), Dst: 192.168.250.10 (192.168.250.10)
+	Transmission Control Protocol, Src Port: microsoft-ds (445), Dst Port: 54382 (54382), Seq: 1221, Ack: 1104, Len: 1
+	NetBIOS Session Service
+	SMB2 (Server Message Block Protocol version 2)

折り返し受信パケット

MAC アドレスの状態（バーチャルマシン上での確認）

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	192.168.2.104	192.168.1.105	TCP	66	49157 > microsoft-ds [SYN] Seq=0 win=8192 Len=
2	0.01388600	192.168.1.105	192.168.2.104	TCP	66	microsoft-ds > 49157 [SYN, ACK] Seq=0
3	0.01400700	192.168.2.104	192.168.1.105	TCP	54	49157 > microsoft-ds [ACK] Seq=1 Ack=1
4	0.01444100	192.168.2.104	192.168.1.105	SMB	213	Negotiate Protocol Request
5	0.01541000	192.168.1.105	192.168.2.104	SMB2	228	NegotiateProtocol Response

Frame 4: 213 bytes on wire (1704 bits), 213 bytes captured (1704 bits) on interface 0

Ethernet II, Src: Microsof_b7:1c:07 (00:1d:d8:b7:1c:07), Dst: Vmware_00:00:01 (00:50:56:00:00:01)

Destination: Vmware_00:00:01 (00:50:56:00:00:01)

Source: Microsof_b7:1c:07 (00:1d:d8:b7:1c:07)

Type: IP (0x0800)

Internet Protocol Version 4, Src: 192.168.2.104 (192.168.2.104), Dst: 192.168.1.105 (192.168.1.105)

Transmission Control Protocol, Src Port: 49157 (49157), Dst Port: microsoft-ds (445), Seq: 1, Ack: 1, Len: 159

NetBIOS Session Service

SMB (Server Message Block Protocol)

送信先はDGW MAC アドレス

サブネット越え通信：送信側

送信元がバーチャルマシン MAC アドレス

No.	Time	Source	Destination	Protocol	Length	Info
52	14.50000000	192.168.2.104	192.168.1.105	TCP	54	49157 > microsoft-ds [ACK] Seq=1 Ack=1 win=1
53	14.50000000	192.168.2.104	192.168.1.105	SMB	213	Negotiate Protocol Request
54	14.50000000	192.168.1.105	192.168.2.104	SMB2	228	NegotiateProtocol Response

Frame 53: 213 bytes on wire (1704 bits), 213 bytes captured (1704 bits) on interface 0

Ethernet II, Src: Microsof_b7:1c:07 (00:1d:d8:b7:1c:07), Dst: Microsof_b7:1c:06 (00:1d:d8:b7:1c:06)

Destination: Microsof_b7:1c:06 (00:1d:d8:b7:1c:06)

Source: Microsof_b7:1c:07 (00:1d:d8:b7:1c:07)

Type: IP (0x0800)

Internet Protocol Version 4, Src: 192.168.2.104 (192.168.2.104), Dst: 192.168.1.105 (192.168.1.105)

Transmission Control Protocol, Src Port: 49157 (49157), Dst Port: microsoft-ds (445), Seq: 1, Ack: 1, Len: 159

NetBIOS Session Service

SMB (Server Message Block Protocol)

サブネット越え通信：受信側

MAC アドレスの状態 (バーチャルマシン上での確認)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	192.168.2.104	192.168.1.105	TCP	66	49157 > microsoft-ds [SYN] Seq=0 win=8192 Le
2	0.01388600	192.168.1.105	192.168.2.104	TCP	66	microsoft-ds > 49157 [SYN, ACK] Seq=0
3	0.01400700	192.168.2.104	192.168.1.105	TCP	54	49157 > microsoft-ds [ACK] Seq=1 Ack=1
4	0.01444100	192.168.2.104	192.168.1.105	SMB	213	Negotiate Protocol Request
5	0.01541000	192.168.1.105	192.168.2.104	SMB2	228	NegotiateProtocol Response

+	Frame 4: 213 bytes on wire (1704 bits), 213 bytes captured (1704 bits) on interface 0
-	Ethernet II, Src: Microsof_b7:1c:07 (00:1d:d8:b7:1c:07), Dst: Vmware_00:00:01 (00:50:56:00:00:01)
+	Destination: Vmware_00:00:01 (00:50:56:00:00:01)
+	Source: Microsof_b7:1c:07 (00:1d:d8:b7:1c:07)
	Type: IP (0x0800)
+	Internet Protocol Version 4, Src: 192.168.2.104 (192.168.2.104), Dst: 192.168.1.105 (192.168.1.105)
+	Transmission Control Protocol, Src Port: 49157 (49157), Dst Port: microsoft-ds (445), Seq: 1, Ack: 1, Len: 159
+	NetBIOS Session Service
+	SMB (Server Message Block Protocol)

送信先はDGW MAC アドレス

送信パケット

送信元がバーチャルマシン MAC アドレス

No.	Time	Source	Destination	Protocol	Length	Info
3	0.01400700	192.168.2.104	192.168.1.105	TCP	54	49157 > microsoft-ds [ACK] Seq=1 Ack=1 win=1
4	0.01444100	192.168.2.104	192.168.1.105	SMB	213	Negotiate Protocol Request
5	0.01541000	192.168.1.105	192.168.2.104	SMB2	228	NegotiateProtocol Response

+	Frame 5: 228 bytes on wire (1824 bits), 228 bytes captured (1824 bits) on interface 0
-	Ethernet II, Src: Microsof_b7:1c:06 (00:1d:d8:b7:1c:06), Dst: Microsof_b7:1c:07 (00:1d:d8:b7:1c:07)
+	Destination: Microsof_b7:1c:07 (00:1d:d8:b7:1c:07)
+	Source: Microsof_b7:1c:06 (00:1d:d8:b7:1c:06)
	Type: IP (0x0800)
+	Internet Protocol Version 4, Src: 192.168.1.105 (192.168.1.105), Dst: 192.168.2.104 (192.168.2.104)
+	Transmission Control Protocol, Src Port: microsoft-ds (445), Dst Port: 49157 (49157), Seq: 1, Ack: 160, Len: 174
+	NetBIOS Session Service
+	SMB2 (Server Message Block Protocol version 2)

折り返し受信パケット

※ 送信先と 受信元の MAC アドレスが異なる

異なる VSID 間の通信（Routing）

- VSID が異なる VM Network であっても、Routing Domain ID が同一であれば通信可能。
- Routing は **仮想化モジュール** が実施。その Subnet の Gateway Address は『New-NetVirtualizationLookupRecord』で設定された仮想 MAC Address 及び仮想 IP Address となります。

管理者: Windows PowerShell

```
RoutingDomainID : {82B741DB-8D6E-411C-8F01-4747CA91FF42}
VirtualSubnetID : 14132563
DestinationPrefix : 192.168.100.0/24
NextHop : 0.0.0.0
Metric : 256

RoutingDomainID : {82B741DB-8D6E-411C-8F01-4747CA91FF42}
VirtualSubnetID : 9554512
DestinationPrefix : 10.254.254.0/29
NextHop : 0.0.0.0
Metric : 256

RoutingDomainID : {82B741DB-8D6E-411C-8F01-4747CA91FF42}
VirtualSubnetID : 9554512
DestinationPrefix : 0.0.0.0/0
NextHop : 10.254.254.2
Metric : 256

RoutingDomainID : {33FBC914-9770-47E5-A3AC-35CE5A3D170E}
VirtualSubnetID : 510845
DestinationPrefix : 192.168.2.0/24
NextHop : 0.0.0.0
Metric : 256

RoutingDomainID : {33FBC914-9770-47E5-A3AC-35CE5A3D170E}
VirtualSubnetID : 12911665
DestinationPrefix : 192.168.1.0/24
NextHop : 0.0.0.0
Metric : 256
```

管理者: Windows PowerShell

```
CustomerAddress : 0.0.0.0
VirtualSubnetID : 4378131
MACAddress : 00cafedec0c0
ProviderAddress : 10.1.1.200
CustomerID : {33FBC914-9770-47E5-A3AC-35CE5A3D170E}
Context : GATEWAY-MANAGED
Rule : TranslationMethodEncap
VMName :
UseVmMACAddress : False

CustomerAddress : 192.168.1.103
VirtualSubnetID : 12911665
MACAddress : 001dd8b71c04
ProviderAddress : 10.1.2.101
CustomerID : {33FBC914-9770-47E5-A3AC-35CE5A3D170E}
Context : GATEWAY-MANAGED
Rule : TranslationMethodEncap
VMName :
UseVmMACAddress : False

CustomerAddress : 192.168.2.1
VirtualSubnetID : 510845
MACAddress : 0a0a07cb7d01
ProviderAddress : 169.254.254.254
CustomerID : {33FBC914-9770-47E5-A3AC-35CE5A3D170E}
Context : GATEWAY-MANAGED
Rule : TranslationMethodEncap
VMName :
UseVmMACAddress : False
```

Routing Domain ID が異なる為疎通不可

Network の Default Gateway

Subnet の Gateway Address

NVGRE におけるパケットサイズ およびフラグメンテーション処理

NVGRE の Packet Size

- 仮想マシン間の通信は NVGRE でカプセル化する為、何も処理を行わなければ物理 Network 上に流れる Packet Size は $1518\text{byte} + 42\text{byte} = 1560\text{byte}$ であるはず。
 - ※ Wireshark で Packet キャプチャを実施すると、L2 Frame の最後に挿入される FCS (Frame Check Sequence : 4byte) をキャプチャできない為、キャプチャ結果とは 4byte の差異が出ます。
- いや、L2 Frame を丸ごとカプセル化するのであれば、Outer Frame にも FCS がつくはず。なので、物理 Network 上に流れる Packet Size は 1564byte ではないか？
- 仮想 Network で 802.1q (VLAN Tag) の使用が許容されるのであれば、さらに 4byte が追加されるはず。
- いずれにせよ、1522byte を超える場合、全 Network で Jumbo Frame の設定が必要であるはず。
- 実際のところはどうなのか？ 確認してみました。

NVGRE での FCS の扱い

Internet-Draft NVGRE February 2013

- Virtual Subnet ID (VSID): The first 24 bits are used for VSID as shown in Figure 1.
- FlowID: The last 8 bits of the Key field are (optional) FlowID, which can be used to add per-flow entropy within the same VSID, where the entire Key field (32-bit) is used for ECMP purposes by switches or routers in the physical network infrastructure. If a FlowID is not generated, the FlowID field MUST be set to all zero.

o The protocol type field in the GRE header is set to 0x6558 (transparent Ethernet bridging) [ETHTYPES].

The inner headers (headers of the GRE payload):

o The inner Ethernet frame comprises of an inner Ethernet header followed by the inner IP header, followed by the IP payload. The inner frame could be any Ethernet data frame not just IP. Note that the inner Ethernet frame's FCS is not encapsulated.

o Inner VLAN tag: The inner Ethernet header of NVGRE SHOULD NOT contain inner VLAN Tag. When an NVE performs NVGRE encapsulation, it SHOULD remove any existing VLAN Tag before encapsulating NVGRE headers. If a VLAN-tagged frame arrives encapsulated in NVGRE, then the decapsulating NVE SHOULD drop the frame.

The inner Ethernet frame comprises of an inner Ethernet header followed by the inner IP header, followed by the IP payload.

The inner frame could be any Ethernet data frame not just IP.

Note that the inner Ethernet frame's FCS is not encapsulated.

2013/02 版 (Ver.02)

- 『インナーイーサネットフレームの FCS はカプセル化されない事に注意してください』との注意書きもあるところから、FCS が外された状態でカプセル化されます。つまり、1514byte の L2 Frame がカプセル対象となります。

NVGRE での 802.1q (VLAN Tag) の扱い

Internet-Draft NVGRE February 2013

- Virtual Subnet ID (VSID): The first 24 bits are used for VSID as shown in Figure 1.
- FlowID: The last 8 bits of the Key field are (optional) FlowID, which can be used to add per-flow entropy within the same VSID, where the entire Key field (32-bit) is used for ECMP purposes by switches or routers in the physical network infrastructure. If a FlowID is not generated, the FlowID field MUST be set to all zero.

o The protocol type field in the GRE header is set to 0x6558 (transparent Ethernet bridging) [ETHTYPES].

The inner headers (headers of the GRE payload):

o The inner Ethernet frame comprises of an inner Ethernet header followed by the inner IP header, followed by the IP payload. The inner frame could be any Ethernet data frame not just IP. Note that the inner Ethernet frame's FCS is not encapsulated.

o Inner VLAN tag: The inner Ethernet header of NVGRE SHOULD NOT contain inner VLAN Tag. When an NVE performs NVGRE encapsulation, it SHOULD remove any existing VLAN Tag before encapsulating NVGRE headers. If a VLAN-tagged frame arrives encapsulated in NVGRE, then the decapsulating NVE SHOULD drop the frame.

2013/02 版 (Ver.02)

Inner VLAN tag : The inner Ethernet header of NVGRE SHOULD NOT contain inner VLAN Tag.

インナー VLAN タグを NVGRE のインナーイーサネットヘッダーに含めないでください。

When an NVE performs NVGRE encapsulation, it SHOULD remove any existing VLAN Tag before encapsulating NVGRE headers.

エンドポイントで NVGRE カプセル化をする際、 NVGRE ヘッダーでカプセル化する前に、全ての VLAN タグを削除するべきです。

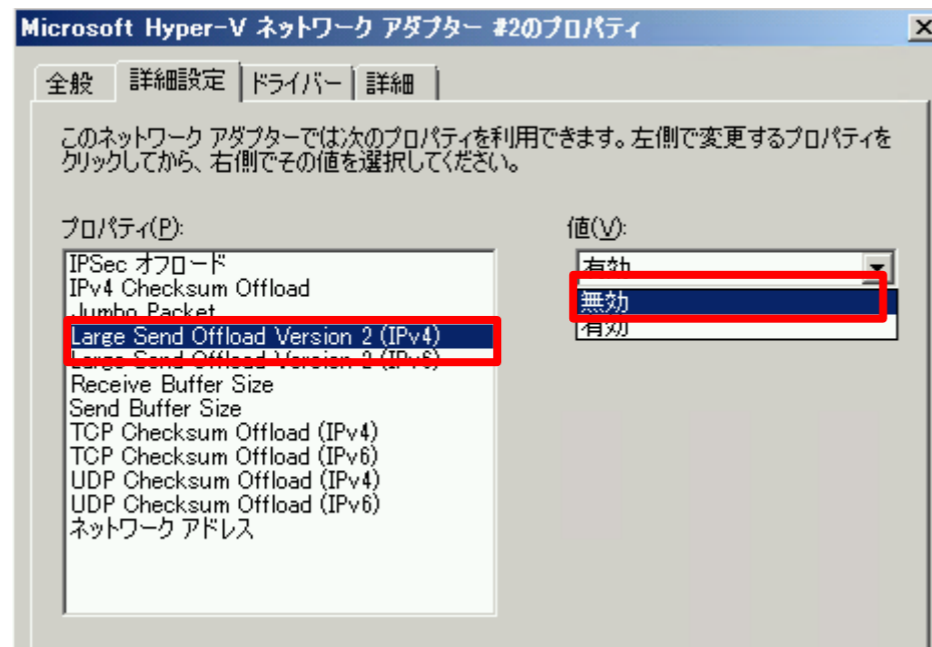
If a VLAN-tagged frame arrives encapsulated in NVGRE, then the decapsulating NVE SHOULD drop the frame.

もし、カプセル化された VLAN タグ付きフレームが到達した場合、カプセル化を解除した後に、そのフレームは破棄すべきです。

- VLAN Tag の使用は不可。
従って、最大 1514byte の L2 Frame がカプセル化対象になります。

NVGRE の Packet Size : 確認方法

- 仮想マシン上でカプセル化前の Packet を取得します。H/W オフロード処理が実施されないように、仮想マシンの Network Adapter でオフロード設定をオフにします。



No.	Time	Source	Destination	Protocol	Length	Info
43	4.74053400	192.168.101.104	192.168.101.105	TCP	54	5001 > 49165 [ACK] Seq
44	4.74055200	192.168.101.105	192.168.101.104	TCP	5726	49165 > 5001 [PSH, ACK
45	4.74739000	192.168.101.104	192.168.101.105	TCP	54	5001 > 49165 [ACK] Seq
46	4.74742600	192.168.101.105	192.168.101.104	TCP	7144	49165 > 5001 [PSH, ACK
47	4.74920700	192.168.101.104	192.168.101.105	TCP	54	5001 > 49165 [ACK] Seq
48	4.74922900	192.168.101.105	192.168.101.104	TCP	7144	49165 > 5001 [ACK] Seq
49	4.74926600	192.168.101.105	192.168.101.104	TCP	1188	49165 > 5001 [PSH, ACK
50	4.74996500	192.168.101.104	192.168.101.105	TCP	54	5001 > 49165 [ACK] Seq

オフロード有効の場合の Packet 長表示

- 同一のタイミングで Hyper-V Host の物理 NIC が接続されている Switch Port を通過する Packet を接続された Switch の SPAN Port から Capture を実施します。
- 確認する通信は http 通信 (80 / tcp) で、DF bit = 1 (Don't Fragment) が設定されています。

NVGRE の Packet Size : 結果

- 仮想マシン上で Packet を確認すると、同一サブネット上の通信であるにもかかわらず、Type3 / Code4 の ICMP Packet で MTU サイズの修正を求められている事を確認。以降 1472 (1458 + 14) byte Packet ※で通信しています。

The image shows a Wireshark packet capture of an ICMP packet. The packet list shows a packet of length 590 bytes. The packet details pane shows the following information:

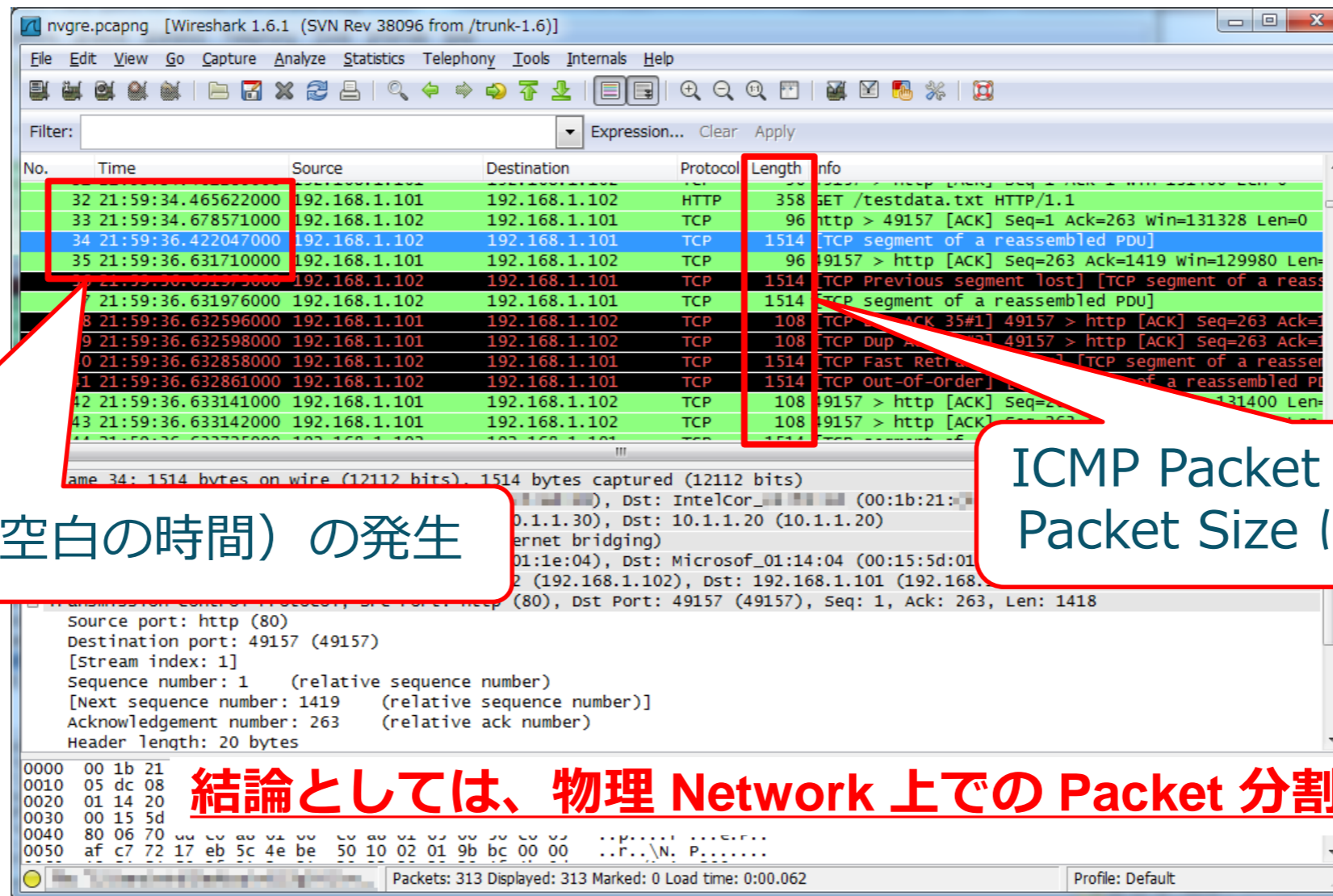
- Internet Control Message Protocol
- Type: 3 (Destination unreachable)
- Code: 4 (Fragmentation needed)
- Checksum: 0x46dc [correct]
- MTU of next hop: 1458

The packet bytes pane shows the raw data of the packet, including the ICMP header and the payload.

※ FCS 含まず

NVGRE の Packet Size : 結果

- 物理 Network 上で Packet を確認すると、ICMP Packet は流れていないので、Hyper-V の仮想 Switch（仮想化フィルタードライバー？）が ICMP を返していると推測されます。



NVGRE の Packet Size : 追加確認

- 同一の環境で、UDP 通信を確認してみました。
- iperf.exe にて datagram 1470byte 、 DF bit = 0 の UDP トラフィックを発生させ、仮想マシン上及び物理 Network 上で確認しました。

NVGRE の Packet Size : 追加結果

- 仮想マシン上の Packet で、icmp（Path MTU Discovery）を確認。次の Packet から MTU サイズを調整／分割（1466byte + 80byte ※）して送信している事も確認しました。
- 物理 Network 上でも 1508byte + 122byte（NVGRE オーバーヘッド 42byte）Packet ※で通信している事を確認しました。

The left screenshot shows a packet capture in Wireshark with the following details:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	Microsof_01:1e:04	Broadcast	ARP	42	who has 192.168.100.103? Tell 192.168.100.102
2	0.00017900	Microsof_01:1e:04	Microsof_01:1e:04	ARP	42	192.168.100.103 is at 00:15:5d:01:14:04
3	0.00019100	192.168.100.102	192.168.100.103	UDP	1512	Source port: 51167 Destination port: 5001
4	0.00026200	192.168.100.103	192.168.100.102	ICMP	590	Destination unreachable (Fragmentation needed)
5	0.01828500	192.168.100.102	192.168.100.103	IPv4	1466	Fragmented IP protocol (proto=UDP 17, off=0, ID=122) source port: 51167 Destination port: 5001
6	0.01829500	192.168.100.102	192.168.100.103	UDP	80	Source port: 51167 Destination port: 5001
7	0.03368800	192.168.100.102	192.168.100.103	IPv4	1466	Fragmented IP protocol (proto=UDP 17, off=0, ID=122) source port: 51167 Destination port: 5001
8	0.03369500	192.168.100.102	192.168.100.103	UDP	80	Source port: 51167 Destination port: 5001
9	0.04926800	192.168.100.102	192.168.100.103	IPv4	1466	Fragmented IP protocol (proto=UDP 17, off=0, ID=122) source port: 51167 Destination port: 5001
10	0.04927500	192.168.100.102	192.168.100.103	UDP	80	Source port: 51167 Destination port: 5001
11	0.05074300	192.168.100.102	192.168.100.103	IPv4	1466	Fragmented IP protocol (proto=UDP 17, off=0, ID=122) source port: 51167 Destination port: 5001
12	0.05075000	192.168.100.102	192.168.100.103	UDP	80	Source port: 51167 Destination port: 5001

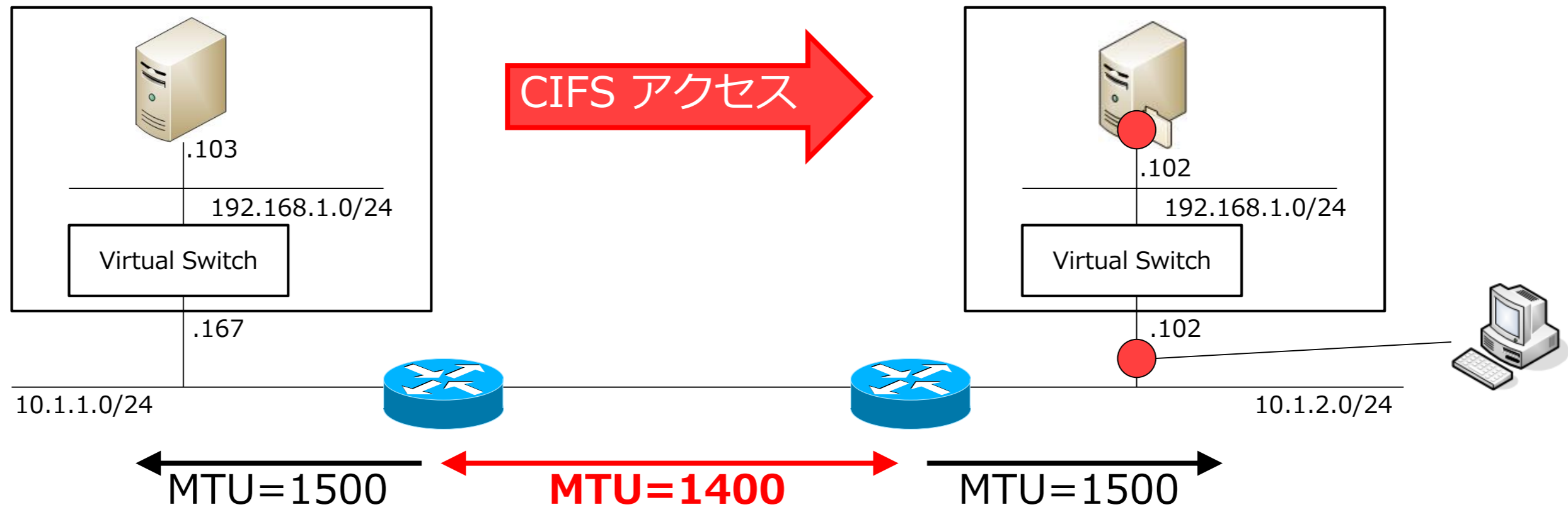
The right screenshot shows a packet capture in Wireshark with the following details:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	192.168.100.102	192.168.100.103	IPv4	1508	Fragmented IP protocol (proto=UDP 17, off=0, ID=122) source port: 51167 Destination port: 5001
2	0.00002200	Intel_06:da:62	Broadcast	ARP	60	who has 10.1.2.250? Tell 10.1.2.101
3	0.00050800	Cisco_df:9f:95	Intel_06:da:62	ARP	60	10.1.2.250 is at 00:18:19:df:9f:95
4	0.00072900	192.168.100.102	192.168.100.103	UDP	122	Source port: 51167 Destination port: 5001
5	0.01535800	192.168.100.102	192.168.100.103	IPv4	1508	Fragmented IP protocol (proto=UDP 17, off=0, ID=122) source port: 51167 Destination port: 5001
6	0.01538300	192.168.100.102	192.168.100.103	UDP	122	Source port: 51167 Destination port: 5001
7	0.03095200	192.168.100.102	192.168.100.103	IPv4	1508	Fragmented IP protocol (proto=UDP 17, off=0, ID=122) source port: 51167 Destination port: 5001
8	0.03097200	192.168.100.102	192.168.100.103	UDP	122	Source port: 51167 Destination port: 5001
9	0.03240500	192.168.100.102	192.168.100.103	IPv4	1508	Fragmented IP protocol (proto=UDP 17, off=0, ID=122) source port: 51167 Destination port: 5001
10	0.03241600	192.168.100.102	192.168.100.103	UDP	122	Source port: 51167 Destination port: 5001
11	0.04212700	192.168.100.102	192.168.100.103	IPv4	1508	Fragmented IP protocol (proto=UDP 17, off=0, ID=122) source port: 51167 Destination port: 5001
12	0.04214300	192.168.100.102	192.168.100.103	UDP	122	Source port: 51167 Destination port: 5001

※ FCS 含まず

経路上でのフラグメンテーション

- 以下のような環境で、NVGRE の疎通試験、並びにパケットキャプチャを実施しました。
- 試験時に用いた通信は Windows ファイル共有（CIFS : 445/TCP）です。



● : パケットキャプチャ実施ポイント

経路上でのフラグメンテーション：結果

- 結果としてはアクセス不可でした。
- 経路上でフラグメントが発生した場合、Router が ICMP を返す先は PA であり、バーチャルマシンまでリダイレクトされないようです。
- 経路上に MTU=1500 以下の回線（VPN 等）がある場合は、注意が必要です。

物理ネットワーク上のキャプチャ結果

No.	Time	Source	Destination	Protocol	Length	Info
27	8.46055100	192.168.1.102	192.168.1.103	SMB2	173	Create Response
28	8.46113200	192.168.1.103	192.168.1.102	SMB2	372	Create Request
29	8.46164200	192.168.1.102	192.168.1.103	SMB2	340	Create Response File:
30	8.46223800	192.168.1.103	192.168.1.102	SMB2	198	Find Request File: SMB2_FIND_ID_BOTH_DIRECTORY_INFO Pattern: *
31	8.47225700	192.168.1.102	192.168.1.103	SMB2	340	[TCP Previous segment not captured] Create Response File:
32	8.47225800	192.168.1.102	192.168.1.103	SMB2	173	Notify Response, Error: STATUS_PENDING
33	8.47303300	192.168.1.103	192.168.1.102	TCP	108	[TCP Dup ACK 30#1] 49163 > microsoft-ds [ACK] Seq=2422 Ack=1691 wi
34	8.47303400	192.168.1.103	192.168.1.102	TCP	108	[TCP Dup ACK 30#2] 49163 > microsoft-ds [ACK] Seq=2422 Ack=1691 wi
35	8.77717800	192.168.1.102	192.168.1.103	TCP	1514	[TCP Retransmission] [TCP segment of a reassembled PDU]
36	8.77773000	10.1.2.250	10.1.2.102	ICMP	70	Destination unreachable (Fragmentation needed)
37	9.40162200	192.168.1.102	192.168.1.103	TCP	1514	[TCP Retransmission] microsoft-ds > 49163 [ACK] Seq=1691 Ack=2422
38	9.40222200	10.1.2.250	10.1.2.102	ICMP	70	Destination unreachable (Fragmentation needed)

Frame 36: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
Ethernet II, Src: Cisco_ab:af:71 (00:19:06:ab:af:71), Dst: IntelCor_10:35:cf (68:05:ca:10:35:cf)
Internet Protocol Version 4, Src: 10.1.2.250 (10.1.2.250), Dst: 10.1.2.102 (10.1.2.102)
Internet Control Message Protocol
Type: 3 (Destination unreachable)
Code: 4 (Fragmentation needed)
Checksum: 0x5b9a [correct]
MTU of next hop: 1400
Internet Protocol Version 4, Src: 10.1.2.102 (10.1.2.102)
Generic Routing Encapsulation (Transparent Ethernet bridg

VM上のキャプチャ結果

No.	Time	Source	Destination	Protocol	Length	Info
28	5.58969600	192.168.1.102	192.168.1.103	SMB2	131	Create Response, Error: STATUS_PENDING
29	5.59058500	192.168.1.103	192.168.1.102	SMB2	330	Create Request File:
30	5.59085700	192.168.1.102	192.168.1.103	SMB2	298	Create Response File:
31	5.59165900	192.168.1.103	192.168.1.102	SMB2	156	Find Request File: SMB2_FIND_ID_BOTH_DIRECTORY_INFO Pattern: *
32	5.59185600	192.168.1.102	192.168.1.103	TCP	1514	[TCP segment of a reassembled PDU]
33	5.59186500	192.168.1.102	192.168.1.103	SMB2	500	Find Response
34	5.59309300	192.168.1.103	192.168.1.102	ICMP	590	Destination unreachable (Fragmentation needed)
35	5.60127100	192.168.1.102	192.168.1.103	SMB2	298	Create Response File:
36	5.60130300	192.168.1.102	192.168.1.103	SMB2	131	Notify Response, Error: STATUS_PENDING
37	5.60248500	192.168.1.103	192.168.1.102	TCP	66	[TCP Dup ACK 31#1] 49163 > microsoft-ds [ACK] Seq=2422 Ack=1691 wi
38	5.60248600	192.168.1.103	192.168.1.102	TCP	66	[TCP Dup ACK 31#2] 49163 > microsoft-ds [ACK] Seq=2422 Ack=1691 wi
39	5.90629400	192.168.1.102	192.168.1.103	TCP	1472	[TCP Retransmission] microsoft-ds > 49163 [ACK] Seq=1691 Ack=2422
40	6.53079000	192.168.1.102	192.168.1.103	TCP	1472	[TCP Retransmission] microsoft-ds > 49163 [ACK] Seq=1691 Ack=2422

PowerShell による Network Virtualization 実装

PowerShell での実装（１）

- PowerShell での実装は、大きく分けて 4 ステップ

1. CA と PA 、仮想マシンの MAC Address 、 VSID の組み合わせを定義。
また、トンネル化方式を指定

- 使用コマンド： New-NetVirtualizationLookupRecord
- コマンド使用例：

```
New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.101" -ProviderAddress "10.1.1.20" -MACAddress "00155D011404" -Rule "TranslationMethodEncap" -VMName "hv3-blue01"
```

- ポイント：『 -Rule 』でトンネル方式を指定
 - ✓ -Rule "TranslationMethodEncap" ⇒ NVGRE
 - ✓ -Rule "TranslationMethodNat" ⇒ IP Rewrite
- ポイント：『 -UseVmMACAddress \$True 』を指定すると、 IP Rewrite でも仮想マシンの MAC Address を使用可能

PowerShell での実装 (2)

2. RoutingDomain を定義して、同一 RoutingDomain の VSID と CA の送信先セグメントアドレスの組み合わせを定義

- 使用コマンド : New-NetVirtualizationCustomerRoute
- コマンド使用例 :

```
New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}"  
-VirtualSubnetID "5001" -DestinationPrefix "192.168.1.0/24" -NextHop "0.0.0.0" -Metric 255
```

```
New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}"  
-VirtualSubnetID "5001" -DestinationPrefix "0.0.0.0/0" -NextHop "192.168.1.250" -Metric 255
```

- ポイント : 仮想マシンの通信先として、宛先セグメント (DestinationPrefix) 単位で、
全ての Route (Default Route 含む) を記述。
『 RoutingDomainID 』は UUID 形式で指定し、同一物理 Network 中で重複が発生しないよう注意

PowerShellでの実装 (3)

3. Hyper-V の物理 NIC（仮想スイッチ）と PA の紐づけを定義。また、PA が複数サブネットに存在する場合には PA の Routing（Default Route）を定義

- 使用コマンド : `New-NetVirtualizationProviderAddress`
`New-NetVirtualizationProviderRoute`
- コマンド使用例 :

```
$iface = Get-NetAdapter WNVNIC
```

```
New-NetVirtualizationProviderAddress -InterfaceIndex $iface.InterfaceIndex -ProviderAddress "10.1.1.20"  
-PrefixLength 24
```

```
New-NetVirtualizationProviderRoute -InterfaceIndex $iface.InterfaceIndex -DestinationPrefix "0.0.0.0/0"  
-NextHop "10.1.1.1"
```

- ポイント : PA のサブネットマスクは『PrefixLength』で指定する。CIDR 形式でない事に注意。
PA の Routing（Default Route）を指定する場合は CIDR 形式である事に注意。

PowerShellでの実装 (4)

4. Hyper-V の物理 NIC (仮想スイッチ) と仮想マシンの MAC Address 、 VSID の組み合わせを定義

- 使用コマンド : Set-VMNetworkAdapter
- コマンド使用例 :

```
$cred = Get-Credential "dob1¥administrator"
```

```
Invoke-Command -ComputerName "ml110g6-01" -Credential $cred {  
  Get-VMNetworkAdapter "hv3-blue01" | where {$_.MacAddress -eq "00155D011404"} | Set-VMNetworkAdapter  
  -VirtualSubnetID 5001;  
}
```

- ポイント : 実行に管理者権限が必要な為、あらかじめ『 Get-Credential 』 コマンドレットにて資格情報を取得
指定 MAC Address が接続された仮想 Switch のポート (?) に対して、 VSID を割り当てるイメージ
⇒ VSID ACL ?

PowerShellでの実装（結果）

```
管理者: Windows PowerShell
PS C:\Users\administrator.DOB1> Get-NetVirtualizationLookupRecord

CustomerAddress : 192.168.1.109
VirtualSubnetID : 3631299
MACAddress      : 001dd8b71c06
ProviderAddress : 10.1.1.117
CustomerID      : {CD3E5A73-C1D6-4C41-9B20-EDF79E34EA9F}
Context         : SCVMM-MANAGED
Rule            : TranslationMethodEncap
VMName          : hv3-red01
UseVmMACAddress : False

CustomerAddress : 192.168.1.111
VirtualSubnetID : 3631299
MACAddress      : 001dd8b71c0e
ProviderAddress : 10.1.1.117
CustomerID      : {CD3E5A73-C1D6-4C41-9B20-EDF79E34EA9F}
Context         : SCVMM-MANAGED
Rule            : TranslationMethodEncap
VMName          : hv3-red02
UseVmMACAddress : False

CustomerAddress : 192.168.1.1
VirtualSubnetID : 1814990
MACAddress      : 005056000001
ProviderAddress : 1.1.1.1
CustomerID      : {43277961-6F08-45E6-B9B8-9AE2A1F3A51D}
Context         : SCVMM-MANAGED
Rule            : TranslationMethodEncap
VMName          : GW
UseVmMACAddress : False
```

```
管理者: Windows PowerShell
PS C:\Users\administrator.DOB1> Get-NetVirtualizationCustomerRoute

RoutingDomainID : {43277961-6F08-45E6-B9B8-9AE2A1F3A51D}
VirtualSubnetID : 1814990
DestinationPrefix : 192.168.1.0/24
NextHop           : 0.0.0.0
Metric            : 0

RoutingDomainID : {CE7A90E2-A052-4461-A6D2-AE103E4CB0A1}
VirtualSubnetID : 1872518
DestinationPrefix : 192.168.102.0/24
NextHop           : 0.0.0.0
Metric            : 0
```

```
管理者: Windows PowerShell
PS C:\Users\administrator.DOB1> Get-NetVirtualizationProviderAddress

ProviderAddress : 10.1.1.121
InterfaceIndex  : 13
PrefixLength    : 0
VlanID          : 0
AddressState    : Preferred

ProviderAddress : 10.1.1.114
InterfaceIndex  : 13
PrefixLength    : 0
VlanID          : 0
AddressState    : Preferred
```



Network Virtualization PowerShell での実装

DEMO

PowerShellによる手動設定時の課題

- 全物理ホストに対して、 PowerShell による設定を実施する必要がある。
 - PA、CA、Mac Addressの組み合わせを仮想マシン単位で設定する必要あり。
 - 仮想マシン追加の都度、手動にて追加設定する必要あり。
- Live Migration に自動追従できない為、 Migration 後 PowerShell による再設定実施完了まで仮想マシンは通信不可。
- 物理ホストを再起動すると、その物理ホストに設定されていた Network Virtualization に関する設定が全て初期化されてしまう。
 - 再起動毎に PowerShell による再設定が必要。

System Center 2012 Virtual Machine Manager SP1

Network Virtualization を中心に

まず最初に……

- 手元に、このファイルをダウンロードすることを強くお勧めします。

Cmdlet Reference for Virtual Machine Manager in System Center 2012 SP1

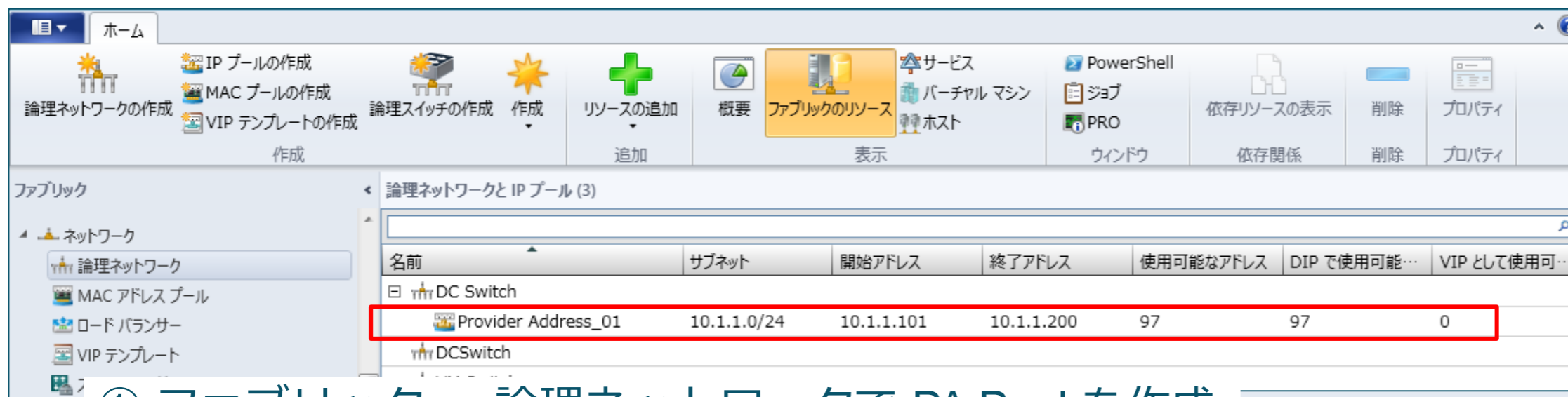
URL : <http://www.microsoft.com/en-au/download/details.aspx?id=6346>

- GUIで設定できない項目があった場合、 PowerShell で設定できないかを調べる上で有用です。
- 但し、 PowerShell で設定可能であっても、サポート外となる項目もありますので、注意が必要です。

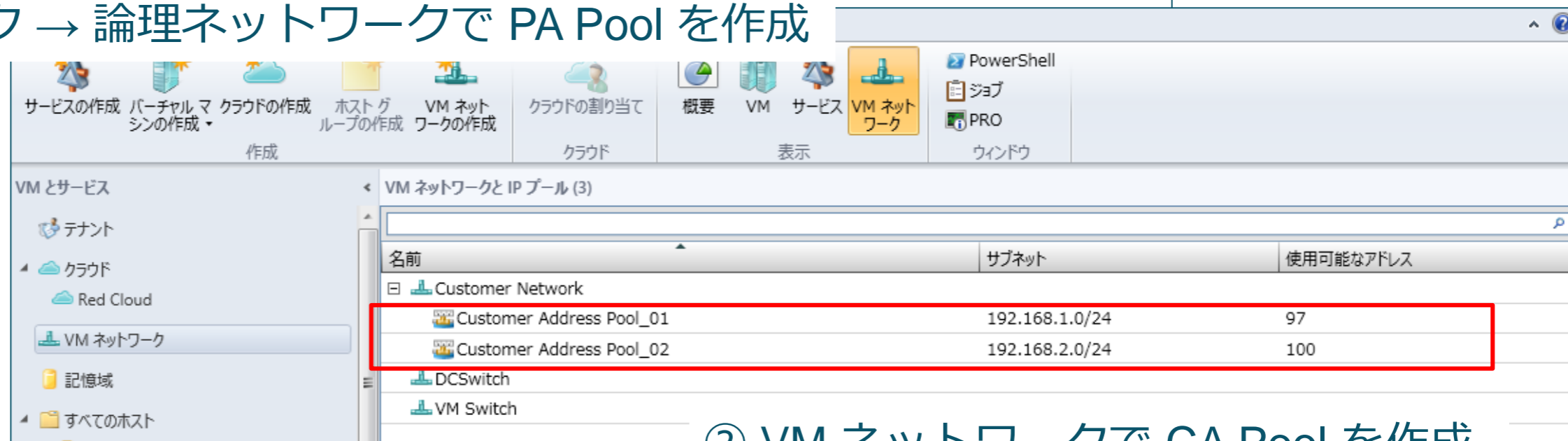
SC2012 VMM SP1 での Network Virtualization

- SC2012 VMM SP1 からサポート
- VM Networks 単位で Network を論理分割
 - VM Networks が異なると、RoutingDomainID が異なる
 - 異なる VM Networks の場合、同一 Cloud であっても疎通不可
 - 同一の VM Networks に属する VMSubnet であれば、疎通可能
- SC2012 VMM SP1 では、NVGRE のみサポート
 - CTP2 の時は IP Rewrite も使用可能でした（というか、Default が IP Rewrite ）
 - PowerShell Cmdlet (New-SCVMSubnet) から IP Rewrite を設定する為のオプションが消えました
 - TechNet Document ※ の 2012/12/21 版を確認すると、『 In this release, you can virtualize the IP address of a virtual machine by using Network Virtualization with Generic Routing Encapsulation (NVGRE) . 』と記述されています
 - ~~➤ また、『 Not all of the capabilities of network virtualization in Windows Server 2012 are supported in this release. 』とも記述されています~~ → 2013/04/24版では記述が消えました
- ~~• Static IP で VM を展開する場合は、テンプレートからの展開が必須~~
→ PowerShell で設定可能です（後述します）

具体的な SC2012 VMM SP1 ネットワーク設定

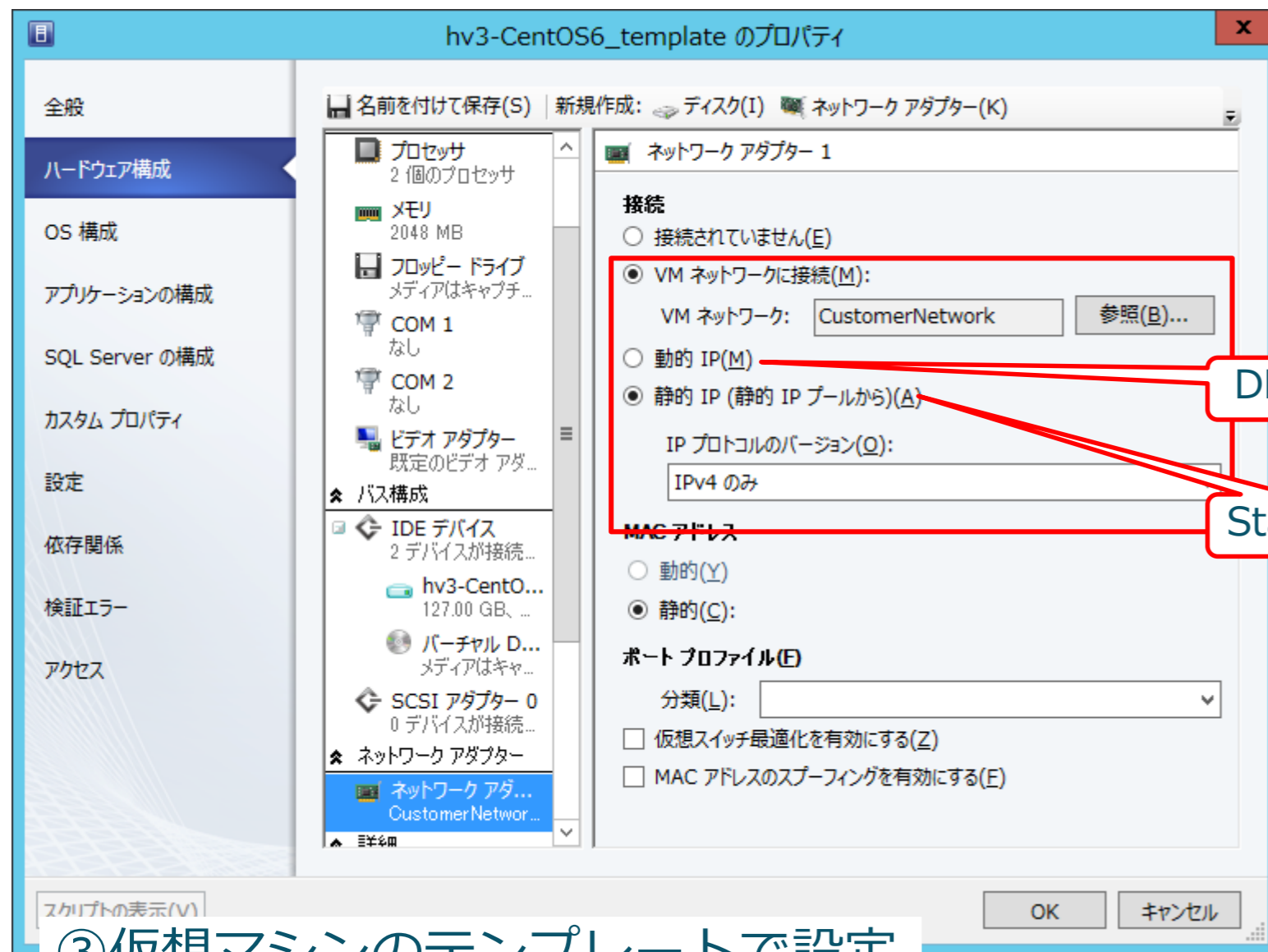


① ファブリック → 論理ネットワークで PA Pool を作成

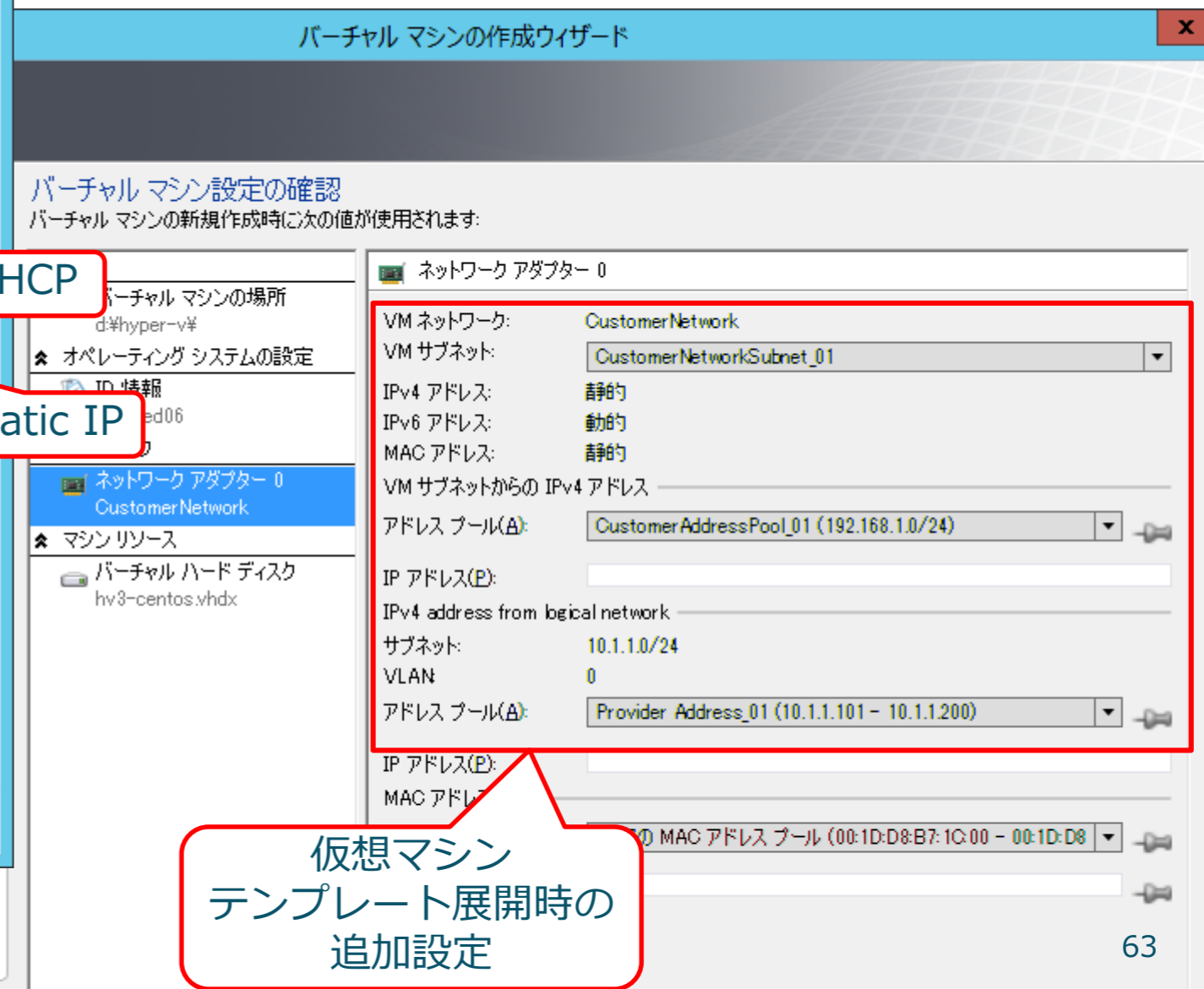


② VM ネットワークで CA Pool を作成

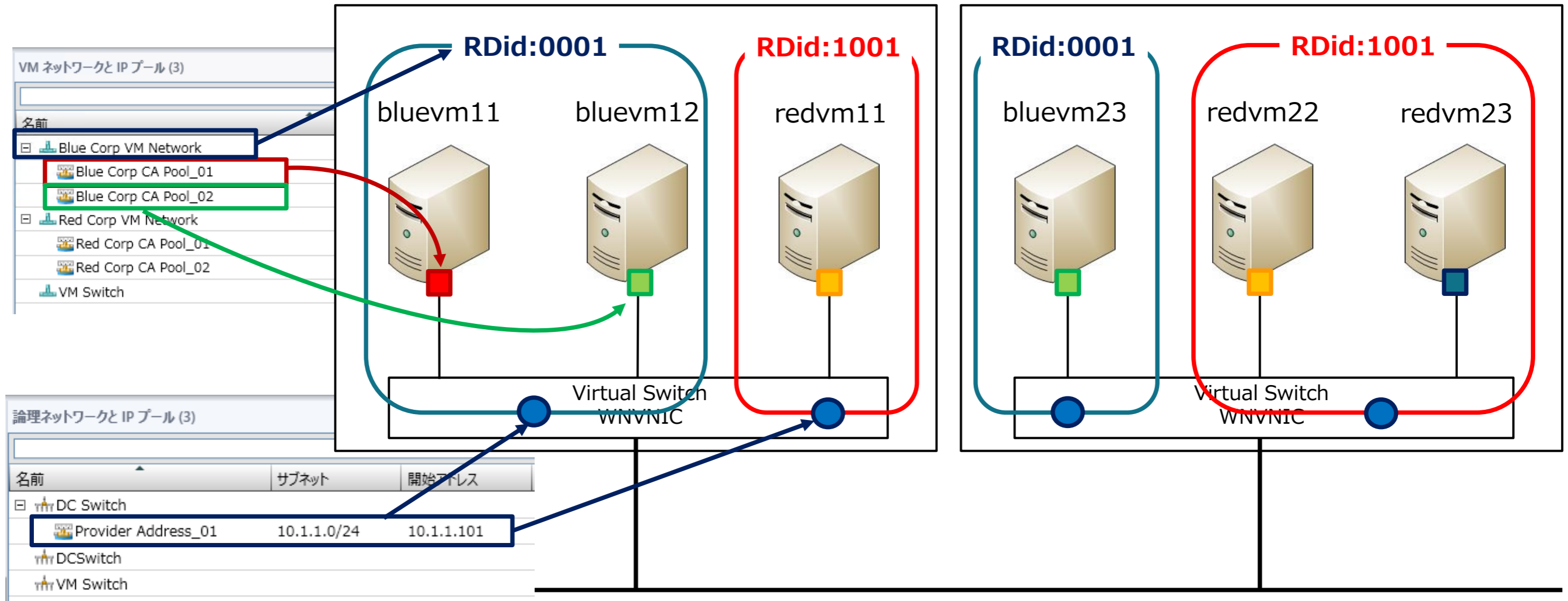
具体的な SC2012 VMM SP1 ネットワーク設定



③仮想マシンのテンプレートで設定



VMM SP1における 論理ネットワークと VM ネットワークの関係



PAは、同一ホスト内であっても、Routing Domain ID単位で個別にアサインされる。

複数サブネット構成の VM ネットワークの注意点

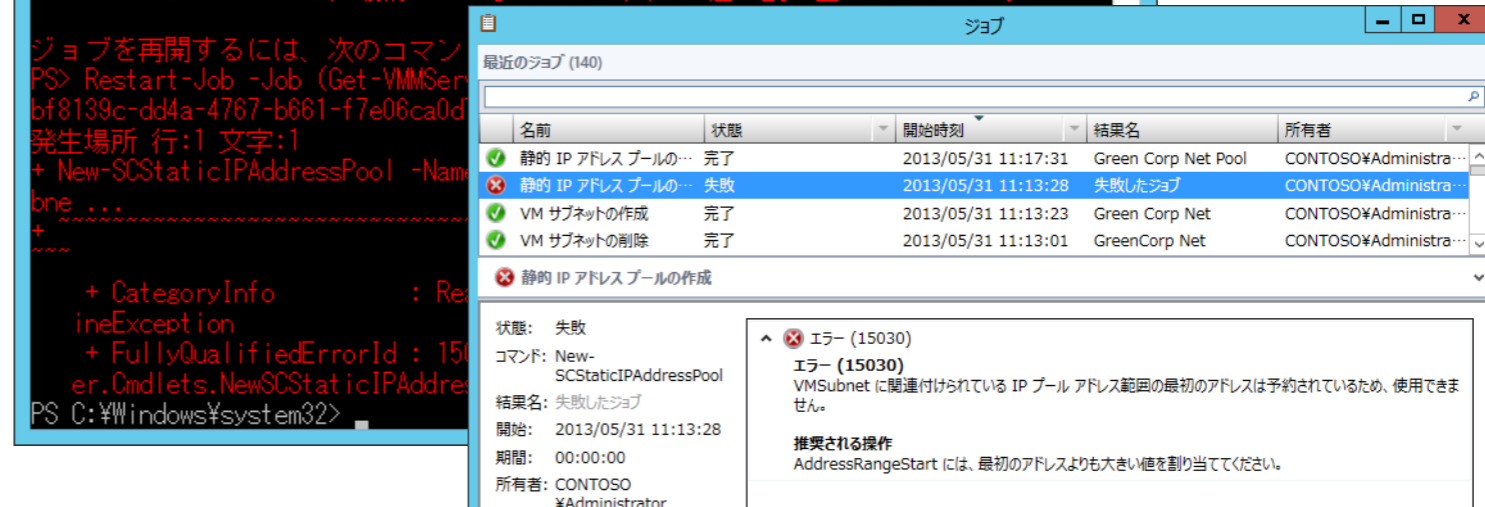
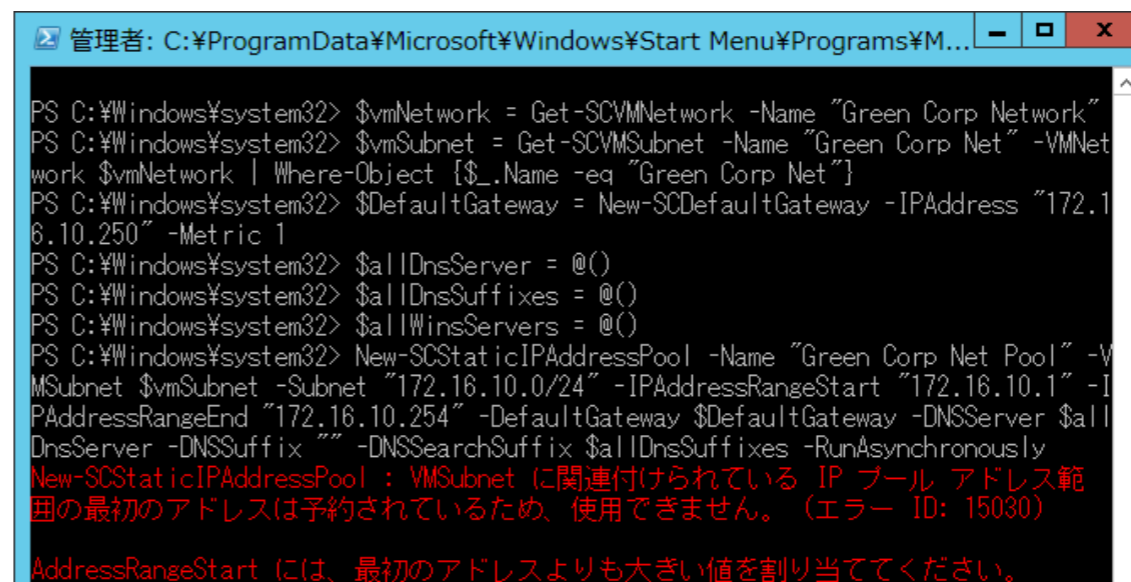
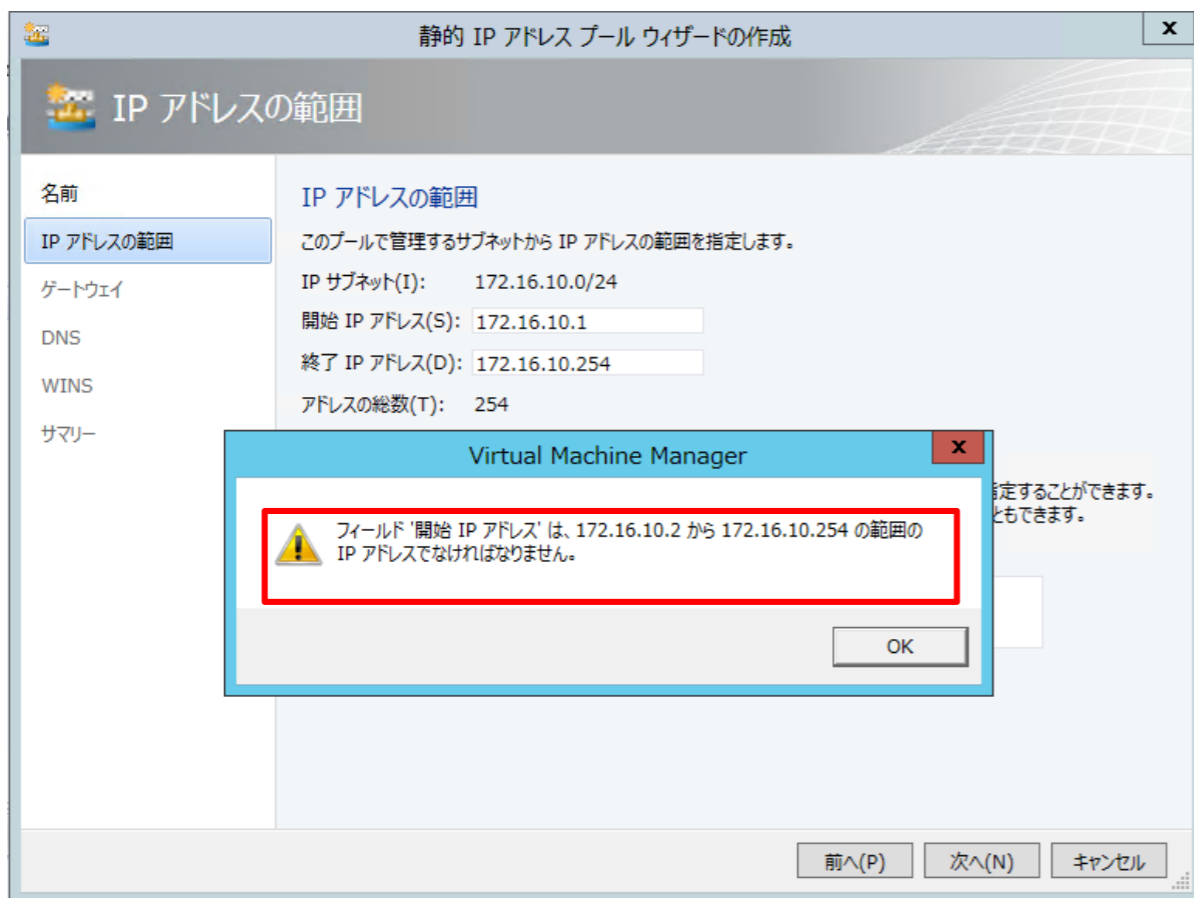
- 一つの VM ネットワーク内に複数のサブネットを構成した場合、サブネット間の Routing は仮想スイッチが実施します。

VM ネットワークと IP プール (3)		
名前	サブネット	使用可能なアドレス
Customer Network		
Customer Address Pool_01	192.168.1.0/24	97
Customer Address Pool_02	192.168.2.0/24	100
DCSwitch		
VM Switch		

- この場合、各サブネットの Gateway Address は SC2012 VMM が自動的に作成し、各サブネットの Host Address 『 1 』 が使用されます
 - 上記例の場合 『 192.168.1.1 』 『 192.168.2.1 』 が Gateway の Address になります
 - 自動割り当ての為、変更不可
- 既存環境を移行する場合には、注意が必要

複数サブネット構成の VM ネットワークの注意点

- ホストアドレス『1』をプールが変更できないか、確認してみました。
- システム予約アドレスとのことで、GUI、PowerShell ともに指定不可でした。



複数サブネット構成の VM ネットワークの注意点

```
管理者: Windows PowerShell
Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. All rights reserved.

PS C:\Users\administrator.DOB1> Get-NetVirtualizationCustomerRoute

RoutingDomainID : {D0CFFFE5-3A24-48DE-BC19-D99E071082FA}
VirtualSubnetID : 1122534
DestinationPrefix : 192.168.10.0/24
NextHop          : 0.0.0.0
Metric           : 0

RoutingDomainID : {D0CFFFE5-3A24-48DE-BC19-D99E071082FA}
VirtualSubnetID : 7696957
DestinationPrefix : 192.168.1.0/24
NextHop          : 0.0.0.0
Metric           : 0

PS C:\Users\administrator.DOB1>
```

```
管理者: Windows PowerShell
PS C:\Users\administrator.DOB1> Get-NetVirtualizationLookupRecord

CustomerAddress : 192.168.10.51
VirtualSubnetID : 1122534
MACAddress      : 001dd8b71c01
ProviderAddress : 10.1.1.54
CustomerID      : {D0CFFFE5-3A24-48DE-BC19-D99E071082FA}
Context        : SCVMM-MANAGED
Rule           : TranslationMethodEncap
VMName         : hv3-red02
UseVmMACAddress : False

CustomerAddress : 192.168.10.1
VirtualSubnetID : 1122534
MACAddress      : 005056000000
ProviderAddress : 1.1.1.1
CustomerID      : {D0CFFFE5-3A24-48DE-BC19-D99E071082FA}
Context        : SCVMM-MANAGED
Rule           : TranslationMethodEncap
VMName         : GW
UseVmMACAddress : False

CustomerAddress : 192.168.1.54
VirtualSubnetID : 7696957
MACAddress      : 001dd8b71c00
ProviderAddress : 10.1.1.55
CustomerID      : {D0CFFFE5-3A24-48DE-BC19-D99E071082FA}
```

複数サブネット構成の VM ネットワークの注意点

- Prefix 24 以下のサブネットを作成した場合、使用可能なホストアドレスの最初のアドレスがゲートウェイアドレスに採用されることを確認しました。

The image shows two windows from a Windows environment. The left window is a PowerShell console titled '管理者: Windows PowerShell' showing the output of the command `Get-NetVirtualizationLookupRecord -CustomerID "[D3C89404-A56B-4BAB-AC51-97DF60C66C6E]"`. The output lists two network configurations for a VM named 'purple-pc01'. The first configuration has a CustomerAddress of 10.10.1.129 and a VirtualSubnetID of 11802626. The second configuration has a CustomerAddress of 10.10.1.130 and a VirtualSubnetID of 8751366. The right window is the 'ジョブ' (Jobs) window in Hyper-V Manager, showing a list of recent jobs. The job '静的 IP アドレス プールの作成' (Static IP Address Pool Creation) is highlighted, and its details are shown in the lower pane. The details pane shows the properties of the '静的 IP アドレス プール - Purple Corp Net Pool#2'. The 'DefaultIPGateway' property is highlighted with a red box, showing a value of 10.10.1.129.

管理者: Windows PowerShell

```
PS C:\Users\administrator.CONTOSO> Get-NetVirtualizationLookupRecord -CustomerID "[D3C89404-A56B-4BAB-AC51-97DF60C66C6E]"
```

CustomerAddress : 10.10.1.129
VirtualSubnetID : 11802626
MACAddress : 005056000002
ProviderAddress : 1.1.1.1
CustomerID : {D3C89404-A56B-4BAB-AC51-97DF60C66C6E}
Context : SCVMM-MANAGED
Rule : TranslationMethodEncap
VMName : GW
UseVmMACAddress : False

CustomerAddress : 10.10.1.130
VirtualSubnetID : 8751366
MACAddress : 001dd8b71c17
ProviderAddress : 10.10.1.52
CustomerID : {D3C89404-A56B-4BAB-AC51-97DF60C66C6E}
Context : SCVMM-MANAGED
Rule : TranslationMethodEncap
VMName : purple-pc01
UseVmMACAddress : False

ジョブ

最近のジョブ (58)

名前	状態	開始時刻	結果名	所有者
静的 IP アドレス プールの...	完了	2013/06/03 16:01:09	Purple Corp Net Pool#2	CONTOSO\Administr...
静的 IP アドレス プールの...	完了	2013/06/03 15:59:31	Purple Corp Net Pool#1	CONTOSO\Administr...

静的 IP アドレス プールの作成

状態: 完了
コマンド: New-SCStaticIPAddressPool
結果名: Purple Corp Net Pool#2
開始: 2013/06/03 16:01:09
期間: 00:00:00
所有者: CONTOSO
¥Administrator

プロパティ	以前の値	新しい値
静的 IP アドレス プール - Purple Corp Net Pool#2		
Name	(なし)	Purple Corp Net Pool#2
StartingIPAddress	(なし)	10.10.1.130
EndingIPAddress	(なし)	10.10.1.254
AccessVlan	(なし)	0
DefaultIPGateway	(なし)	10.10.1.129
IPSubnet	(なし)	10.10.1.128/25
NetBIOSEnabled	(なし)	False

サマリー 詳細 変更履歴

☒ 新規オブジェクトが作成されたときにこのウィンドウを表示する(S)

再開(R) キャンセル(C)

VSID に関して

- VSID は VM サブネットを作成した段階で、VMM によって自動採番（ランダム割り当て）されます。
- VSID を（運用上の理由等で）明示的に指定したい場合、PowerShell から VM サブネットを作成することにより、希望の ID を割り当てることができます。
 - 使用コマンド： New-SCSubnetVLan
New-SCVMSubnet
 - コマンド使用例：

```
$vmNetwork = Get-SCVMNetwork -Name "Green Corp Network"
```

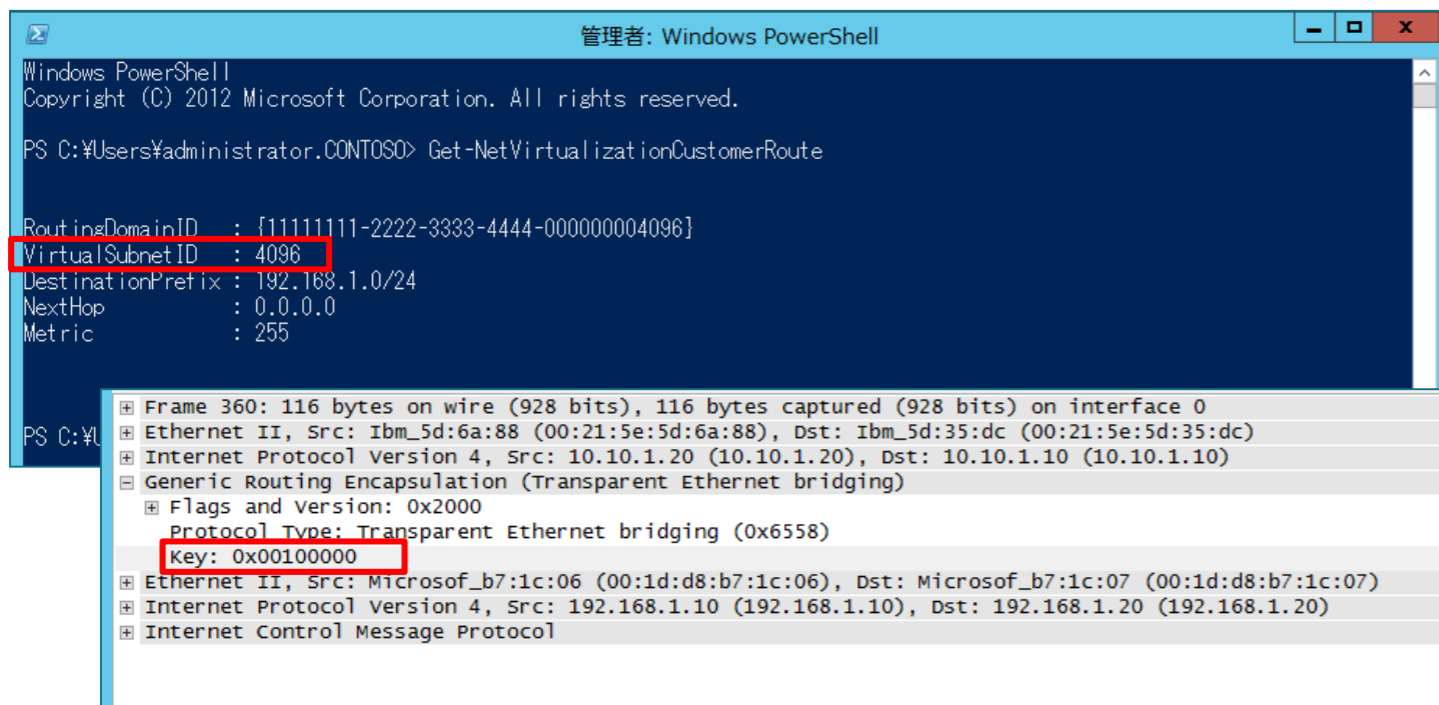
```
$subnet = New-SCSubnetVLan -Subnet "172.16.10.0/24"
```

```
New-SCVMSubnet -Name "GreenCorp Net" -VMNetwork $vmNetwork -SubnetVLan $subnet -VMSubnetID 5001
```

- ポイント：『 -VMSubnetID 』で割り当てたい VSID を指定する。
指定可能な範囲は 4,097 から 16,777,214 。

VSID に関して

- 『 New-NetVirtualizationCustomerRoute 』 Cmdlet では、 VSID は 4,096 から 16,777,215 の範囲で指定可能ですが、『 New-SCVMSubnet 』 Cmdlet を使用した場合、下限が 4,097 になりますので、注意が必要です。



```
管理: Windows PowerShell
Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. All rights reserved.

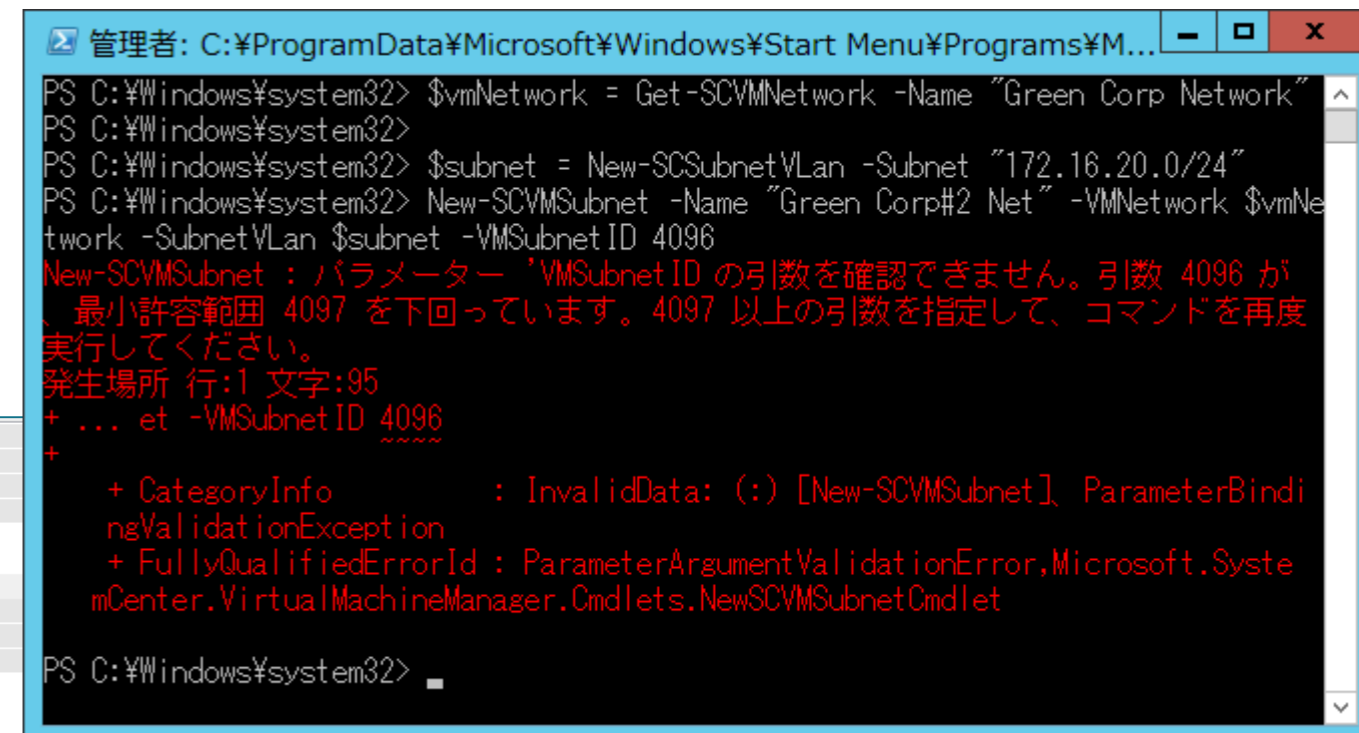
PS C:\Users\administrator.CONTOSO> Get-NetVirtualizationCustomerRoute

RoutingDomainID : {11111111-2222-3333-4444-0000000004096}
VirtualSubnetID  : 4096
DestinationPrefix : 192.168.1.0/24
NextHop           : 0.0.0.0
Metric            : 255
```

Network packet details:

- Frame 360: 116 bytes on wire (928 bits), 116 bytes captured (928 bits) on interface 0
- Ethernet II, Src: Ibm_5d:6a:88 (00:21:5e:5d:6a:88), Dst: Ibm_5d:35:dc (00:21:5e:5d:35:dc)
- Internet Protocol Version 4, Src: 10.10.1.20 (10.10.1.20), Dst: 10.10.1.10 (10.10.1.10)
- Generic Routing Encapsulation (Transparent Ethernet bridging)
- Flags and Version: 0x2000
- Protocol Type: Transparent Ethernet bridging (0x6558)
- Key: 0x00100000
- Ethernet II, Src: Microsof_b7:1c:06 (00:1d:d8:b7:1c:06), Dst: Microsof_b7:1c:07 (00:1d:d8:b7:1c:07)
- Internet Protocol Version 4, Src: 192.168.1.10 (192.168.1.10), Dst: 192.168.1.20 (192.168.1.20)
- Internet Control Message Protocol

『 New-NetVirtualizationCustomerRoute 』 Cmdlet を使用した場合



```
管理: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Microsoft Management Console\cmd.exe
PS C:\Windows\system32> $vmNetwork = Get-SCVMNetwork -Name "Green Corp Network"
PS C:\Windows\system32>
PS C:\Windows\system32> $subnet = New-SCSubnetVlan -Subnet "172.16.20.0/24"
PS C:\Windows\system32> New-SCVMSubnet -Name "Green Corp#2 Net" -VMNetwork $vmNetwork -SubnetVlan $subnet -VMSubnetID 4096
New-SCVMSubnet : パラメーター 'VMSubnetID' の引数を確認できません。引数 4096 が、最小許容範囲 4097 を下回っています。4097 以上の引数を指定して、コマンドを再度実行してください。
発生場所 行:1 文字:95
+ ... et -VMSubnetID 4096
+
+ CategoryInfo          : InvalidData: (:) [New-SCVMSubnet], ParameterBindingValidationException
+ FullyQualifiedErrorId : ParameterArgumentValidationError,Microsoft.SystemCenter.VirtualMachineManager.Cmdlets.NewSCVMSubnetCmdlet

PS C:\Windows\system32>
```

『 New-SCVMSubnet 』 Cmdlet を使用した場合

VSID 16,777,215

- エラーパケット処理など、システムメッセージ交換用として使用される模様
 - KB2779768 問題でみられた icmp Type3 Code10 (Destination host administratively prohibited) のパケットは、VSID FFFFFFFF (16,777,215) のパケットでした

The image shows a Wireshark packet capture of an ICMP Echo (ping) request and response. The packet list on the left shows four packets:

- No. 461: 1591.05489 IntelCor_10:35:cf Broadcast ARP 60 who has 10.1.2.250? Tell 10.1.2.107
- No. 462: 1591.05542 Cisco_df:9f:95 IntelCor_10:35:cf ARP 60 10.1.2.250 is at 00:18:19:df:9f:95
- No. 463: 1591.05565 192.168.1.106 192.168.1.104 ICMP 116 Echo (ping) request id=0x0001, seq=1
- No. 464: 1591.05586 10.1.1.143 10.1.2.107 ICMP 186 Destination unreachable (Host administratively prohibited)

The packet details pane on the right shows the structure of the selected packet (No. 464). The 'Internet Control Message Protocol' section is expanded, showing:

- Type: 3 (Destination unreachable)
- Code: 10 (Host administratively prohibited)
- Checksum: 0xd3e1 [correct]
- Key: 0x001bb1ce

The packet bytes pane at the bottom shows the raw data of the packet, with the key field highlighted in blue.

- 従って、ユーザー利用不可です。

VSID 16,777,215

- しかしながら、VSID 16,777,215 をアサインすることが可能です。
- 但し、NIC に関連付けを行う際に失敗します。
- Technet の『 New-NetVirtualizationCustomerRoute 』 Cmdlet のリファレンスには、『 -VirtualSubnetID 』の引数の許容範囲として 4,096-16,777,214 と記載されていますが、実際には 16,777,215 も許容されてしまうので、注意が必要です。

```
管理者: Windows PowerShell
PS C:\Users\administrator.CONTOSO> New-NetVirtualizationLookupRecord -VirtualSubnetID "16777215" -CustomerAddress "192.168.1.20" -ProviderAddress "10.10.1.10" -MACAddress "001DD8B71C07" -Rule "TranslationMethodEncap" -VMName "red-pc01" -CimSession "cluster01"

CustomerAddress : 192.168.1.20
VirtualSubnetID : 16777215
MACAddress      : 001dd8b71c07
ProviderAddress : 10.10.1.10
CustomerID      : {00000000-0000-0000-0000-000000000000}
Context         :
Rule            : TranslationMethodEncap
VMName          : red-pc01
UseVmMACAddress : False
PSComputerName  : cluster01

PS C:\Users\administrator.CONTOSO>
PS C:\Users\administrator.CONTOSO> New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000016777215}" -VirtualSubnetID "16777215" -DestinationPrefix "192.168.1.0/24" -NextHop "0.0.0.0" -Metric 255 -CimSession "cluster02"

RoutingDomainID : {11111111-2222-3333-4444-000016777215}
VirtualSubnetID : 16777215
DestinationPrefix : 192.168.1.0/24
NextHop           : 0.0.0.0
Metric            : 255
PSComputerName    : cluster02
```

```
管理者: Windows PowerShell
PS C:\Users\administrator.CONTOSO>
PS C:\Users\administrator.CONTOSO> $cred = Get-Credential "contoso\administrator"
PS C:\Users\administrator.CONTOSO> $WNVNIC = "WNVNIC"
PS C:\Users\administrator.CONTOSO>
PS C:\Users\administrator.CONTOSO> $iface = Get-NetAdapter $WNVNIC -CimSession "cluster01"
PS C:\Users\administrator.CONTOSO> New-NetVirtualizationProviderAddress -InterfaceIndex $iface.InterfaceIndex -ProviderAddress "10.10.1.10" -PrefixLength 24 -CimSession "cluster01"

ProviderAddress : 10.10.1.10
InterfaceIndex  : 15
PrefixLength    : 24
VlanID          : 0
AddressState    : Preferred
PSComputerName  : cluster01

PS C:\Users\administrator.CONTOSO>
PS C:\Users\administrator.CONTOSO> Invoke-Command -ComputerName "cluster01" -Credential $cred [
>> Get-VMNetworkAdapter "red-pc01" | where {$_.MacAddress -eq "001DD8B71C07"} | Set-VMNetworkAdapter -VirtualSubnetID 16777215;
>> ]
>> ]
>> ]
操作に失敗しました。
スイッチ 'Vswitch1' のスイッチ ポート設定 'Ethernet Switch Port Security Settings' を適用する際にエラーが発生しました:
無効な引数があります (0x80070057)。
無効なパラメーターが操作に渡されました。
+ CategoryInfo          : InvalidArgument: (Microsoft.HyperV.PowerShell.VMTask:VMTask) [Set-VMNetworkAdapter], VirtualizationOperationFailedException
+ FullyQualifiedErrorId : InvalidParameter,Microsoft.HyperV.PowerShell.Commands.SetVMNetworkAdapterCommand
+ PSComputerName        : cluster01
```

VSID 16,777,215

- 同様に、SC 2012 VMM の『 New-SCVMSubnet 』 Cmdlet においても、16,777,215 を VSID として指定することが可能です。

The screenshot displays two windows from a System Center 2012 VMM environment. The left window is a PowerShell console running commands to create a new VM subnet. The right window is the VMM console showing the configuration of the newly created subnet.

PowerShell Console Output:

```
PS C:\Windows\system32> $vmNetwork = Get-SCVMNetwork -Name "Blue Corp Network"
PS C:\Windows\system32>
PS C:\Windows\system32> $subnet = New-SCSubnetVlan -Subnet "192.168.2.0/24"
PS C:\Windows\system32> New-SCVMSubnet -Name "Blue Corp#2 Net" -VMNetwork $vmNetwork -SubnetVlan $subnet -VMSubnetID 16777215
```

VMM Console Output (Blue Corp#2 Net Pool):

名前	サブネット	使用可能なアドレス
Blue Corp Network		
Blue Corp Net Pool	192.168.1.0/24	249
Blue Corp#2 Net Pool	192.168.2.0/24	253
Green Corp Network		

Blue Corp#2 Net Pool Details:

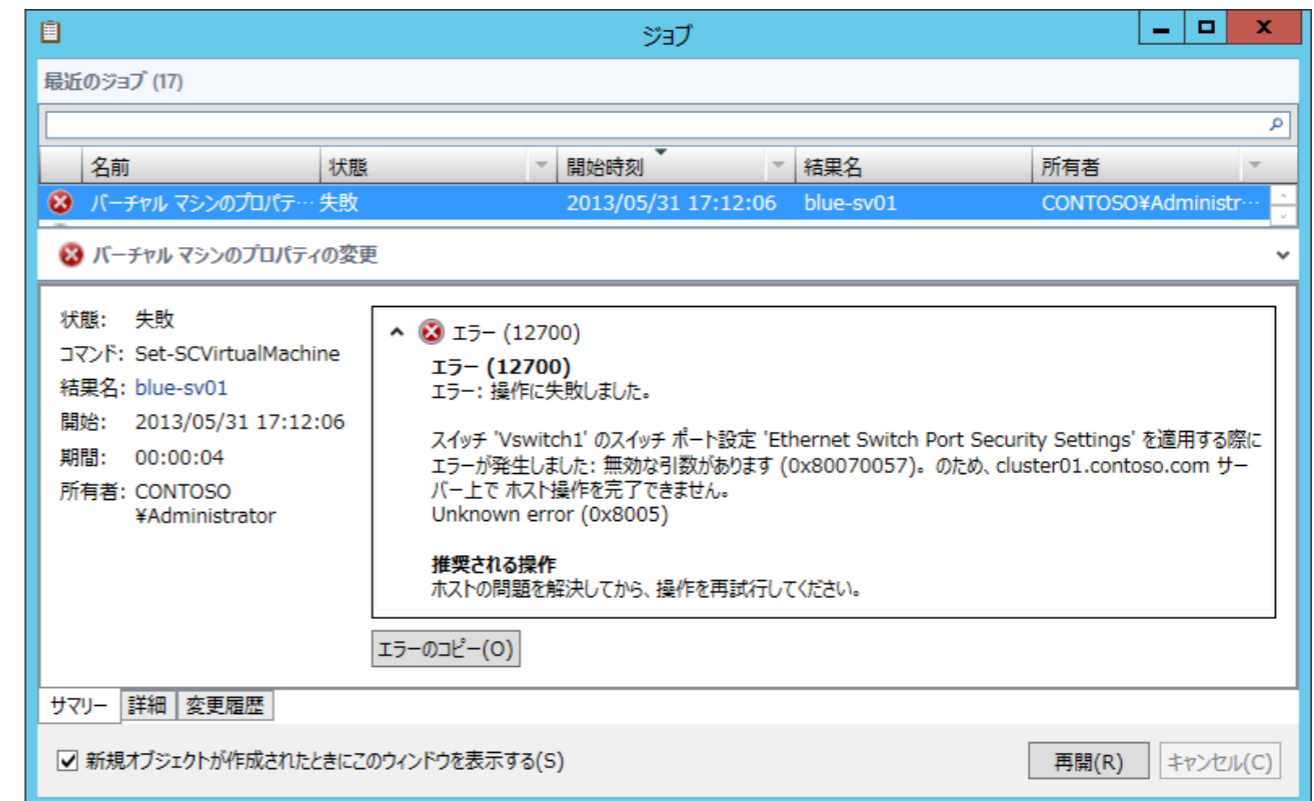
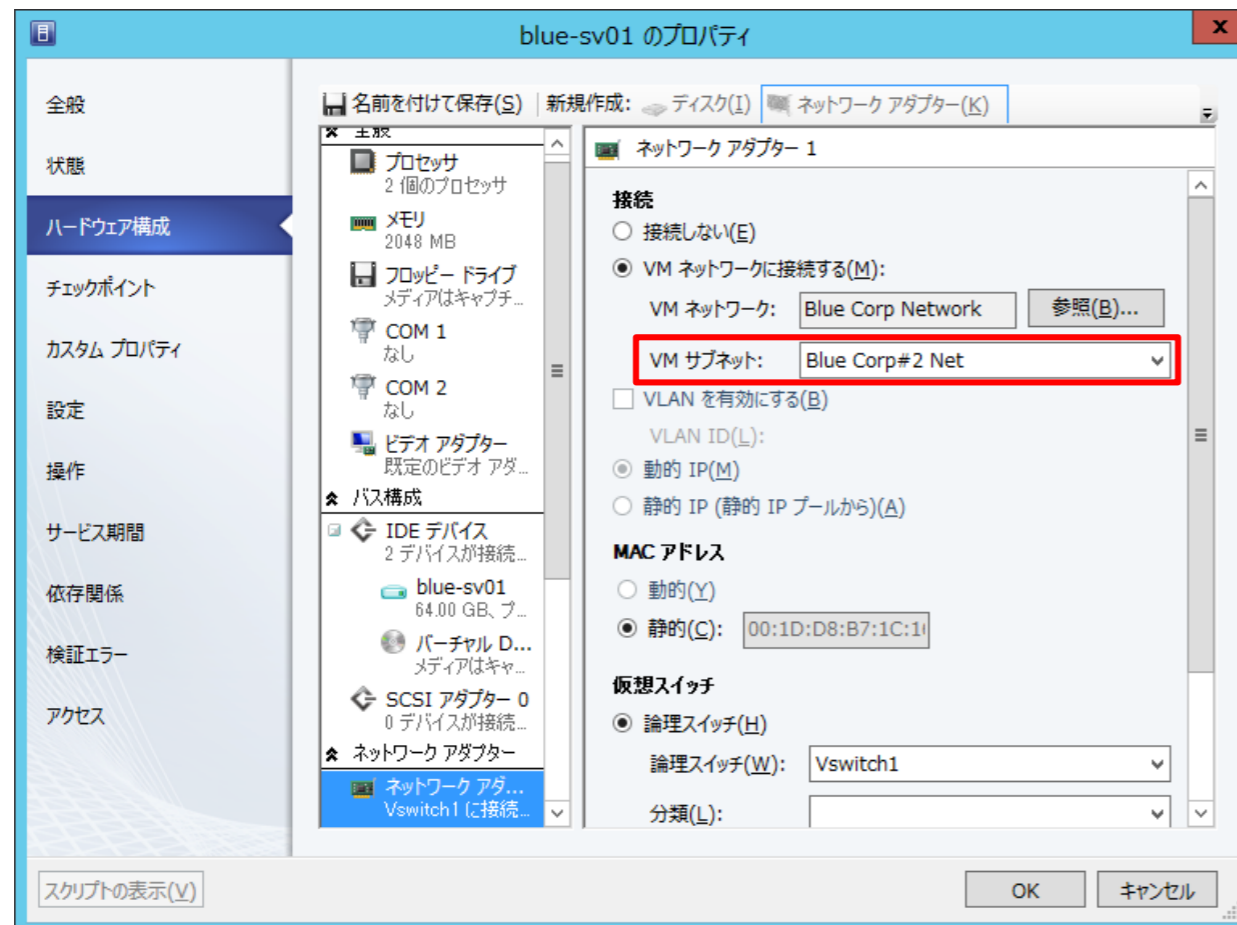
静的 IP アドレス プールの情報		IP アドレスの使用状況	
開始アドレス:	192.168.2.2	使用可能なアドレス:	253
終了アドレス:	192.168.2.254	アドレスの総数:	253
予約されたアドレス:	0		

PowerShell Output Properties:

- Name : Blue Corp#2 Net
- Description :
- VMSubnetID : 16777215**
- VMNetwork : Blue Corp Network
- SubnetVLans : {192.168.2.0/24}
- AllowsIntraPortCommunication : True
- MaxNumberOfPorts :
- VirtualSwitchExtensionManager :
- ExternalId :
- NetworkEntityAccessType : VmmManaged
- ExternalSyncTime :
- IsVMMInternal : False
- ServerConnection : Microsoft.SystemCenter.VirtualMachineManag...
- ID : 1fb8ae0d-3230-4180-8538-ded29b30210f
- IsViewOnly : False
- ObjectType : VMSubnet
- MarkedForDeletion : False
- IsFullyCached : True

VSID 16,777,215

- バーチャルマシンに作成した VSID 16,777,215 の VM サブネットをアサインすると、プロパティ変更ジョブで『Unknown error(0x8005)』が発生するので、注意が必要です。



VSID の結論

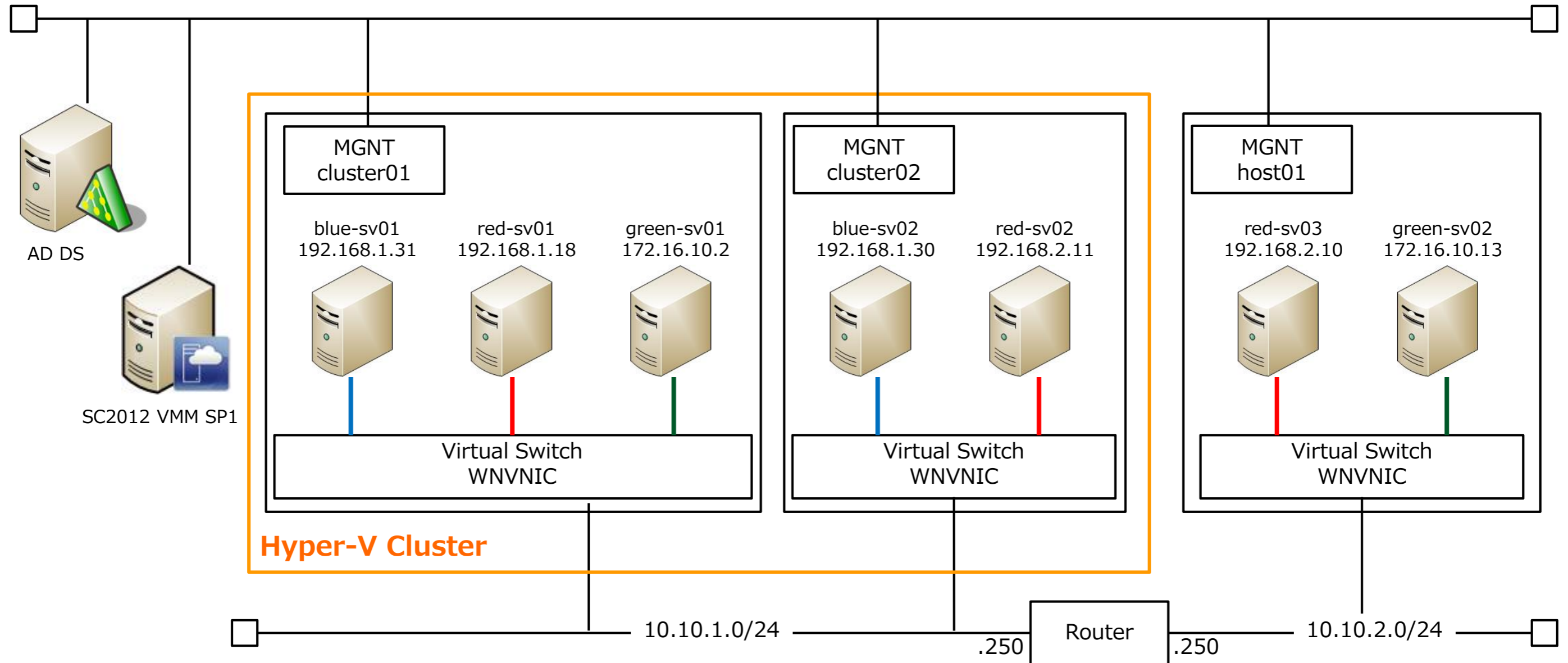
- SC2012 VMM の自動割り当てに依存してしまう、というのがお手軽ソリューションです。
- 手動にて割り当てる場合には、4,097 から 16,777,214 の範囲での割り当てを行うよう、運用回避してください。
- PowerShell による、Orchestrator 等での独自ロジックでの割り当てを行う場合、割り当て可能な VSID から 16,777,215 は、かならず除外してください。



Network Virtualization with SC2012 VMM SP1

DEMO

本日の Demo 環境



Windows Network Virtualizationにおける IP アドレス設定

IP アドレス割り当て方法（ SC2012 VMM 利用時）

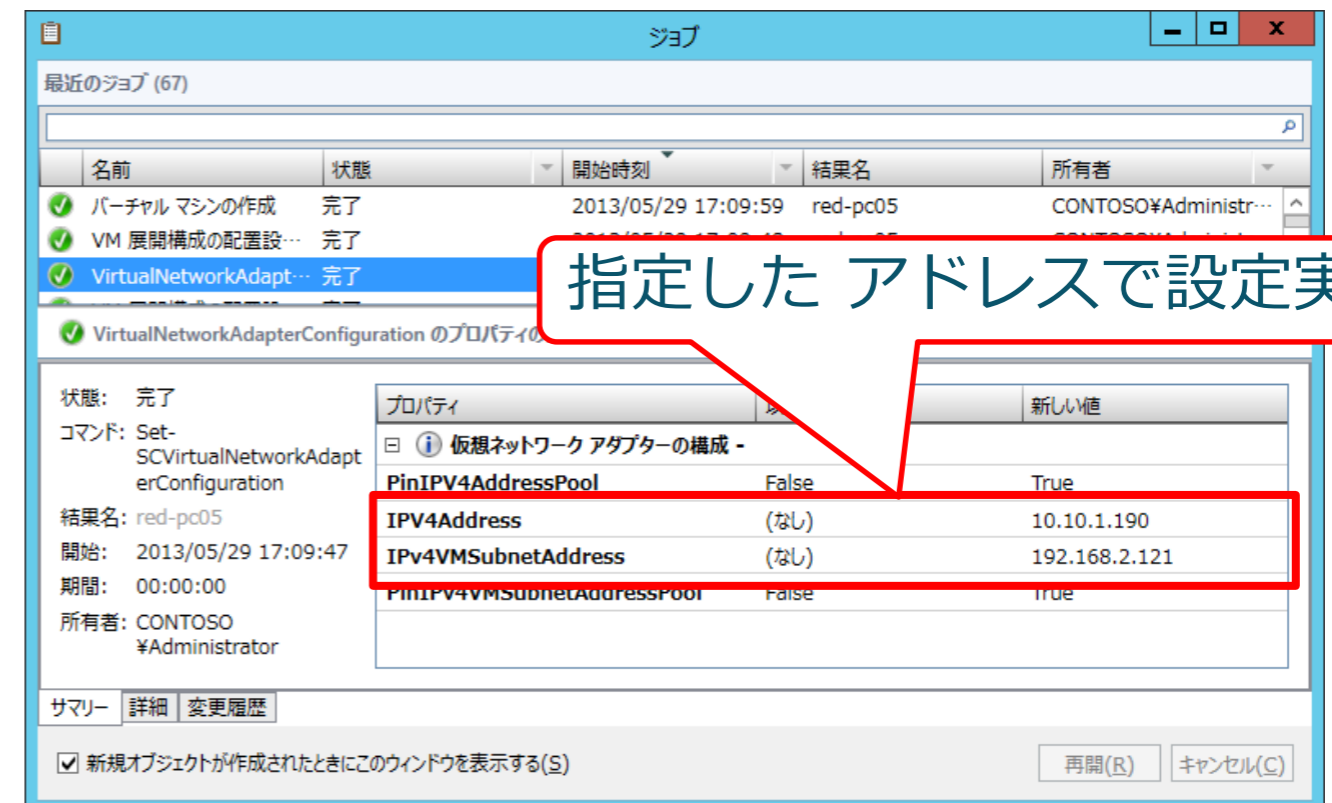
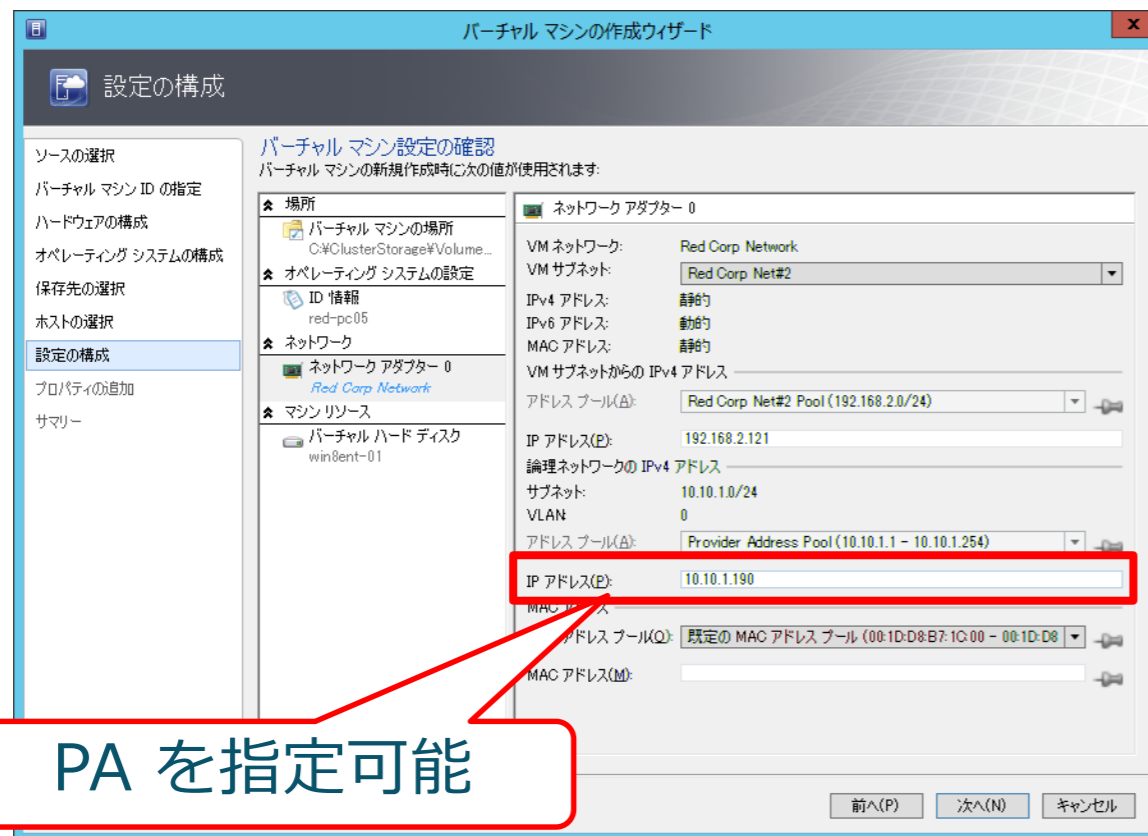
- Network Virtualization で利用される IP アドレスは2 種類。
 - Provider アドレス : 物理ネットワーク上で使用される IP アドレス。
 - Customer アドレス : 仮想ネットワーク上で使用される IP アドレス。
- PA は論理ネットワークの IP プールから自動割り当て
 - 既定では、ホスト毎に RDID 単位で割り当てを実施します。
 - 従って、 1 ホストに 20 RDID が存在する場合、 IP アドレスは 20 アドレス消費することになるので、アドレス設計には注意が必要です。
 - 割り当てられた PA は『 Get-NetVirtualizationProviderAddress 』 Cmdlet で確認可能です。
 - IP プールからの自動割り当てが基本ですが、条件付きで静的設定も可能です。

IP アドレス割り当て方法（ SC2012 VMM 利用時）

- CA は 2 つの割り当て方法が存在。
 - VM ネットワークの IP プールからの動的割り当て（バーチャルマシンでは DHCP 設定）
 - VM ネットワークの IP プールからの静的割り当て（バーチャルマシンでは Static 設定）
- VM ネットワークの IP プールからの静的割り当てを行うための方法は、 2 つ存在。
 - バーチャルマシン作成時に、テンプレートから展開することにより静的割り当てを実施
 - 展開済み（既存バーチャルマシン等）の場合、PowerShellによって静的割り当てを実施
- 既存バーチャルマシンの移行など、 IP アドレスが静的に割り当てられている場合でも、問題なく Network Virtualizationが利用可能。

PA 設定に関して

- PA は 条件付きで静的設定（管理者が任意でアドレスを割り当てる）が可能
 - 割り当てはテンプレート展開時に、GUIから指定可能
 - 割り当て可能なアドレスは、IP プールで設定されている範囲からの任意に指定可能

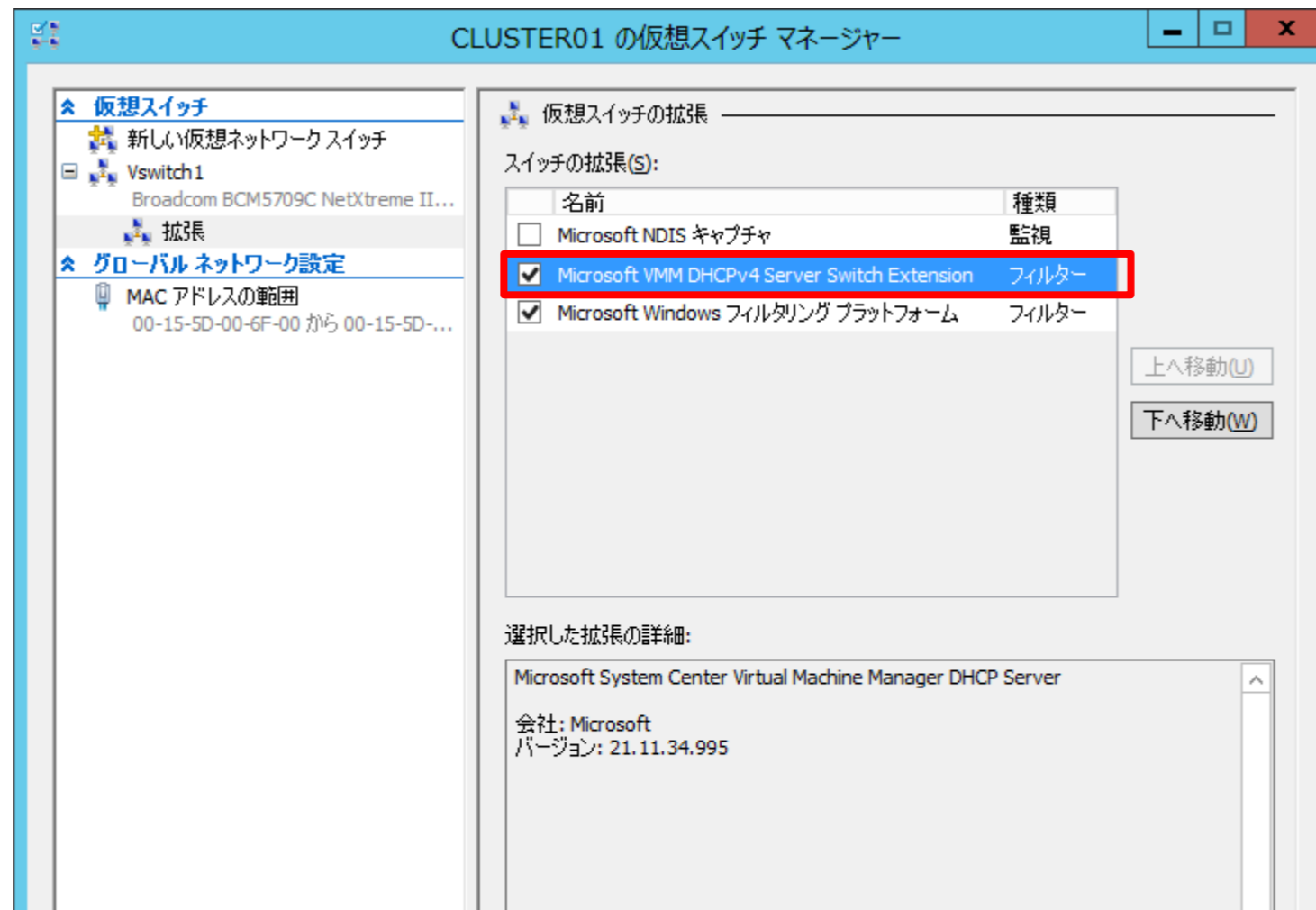


PA 設定に関して

- 静的設定できる条件は以下の通り
 - 静的設定を行うバーチャルマシンと同じ RDID が設定されているバーチャルマシンが、ホスト上に存在しない事
 - RDID 用の PA が既に存在する場合、その PA が自動的に使用されます。
- ライブマイグレーションなど、PA の再設定を伴う動作が発生した場合、IP プールからの自動採番が実施される為、静的設定は無効になるので注意
 - 当該ホストにその PA を使用するバーチャルマシンが存在しなくなった段階で、PA がプールに返却されるという挙動のため
 - クラスター環境では事実上意味をなさないという点に注意が必要
- PA の設定は SC2012 VMM に任せておいた方が無難

SC2012 VMM SP1 での DHCP 実装

- SC2012 VMM SP1 からサポート
- DHCP Extensions (Filter Driver) にて実装。従って、 Windows Server 2012 のみ対応
- SC2012 VMM SP1 エージェント導入時に自動的にインストール



SC2012 VMM SP1 での DHCP 実装

- 仮想マシンからの DHCP Discover を DHCP Extensions がフックし、SC2012 VMMと連携して IP Address を割り当てる模様
 - DHCP Server の Address は『 10.0.0.1 』と表示される
 - IP Pool で設定した IP Address / DNS Server Address などが DHCP のように割り当て可能
 - 一度設定された IP Address は、Release / Renew しても同じ Address が割り当てられる模様だが、VM Subnet の設定を変更すると異なる IP Address が割り当てられる模様。
 - これは、使用しなくなったらプールに戻し、必要になったらプールから再アサイン、という挙動によるものと考えられる。

```
管理者: コマンド プロンプト
C:\Windows\system32>ipconfig /all

Windows IP 構成

ホスト名 . . . . . : blue-pc02
プライマリ DNS サフィックス . . . . . :
ノード タイプ . . . . . : ハイブリッド
IP ルーティング有効 . . . . . : いいえ
WINS プロキシ有効 . . . . . : いいえ

イーサネット アダプター イーサネット:

接続固有の DNS サフィックス . . . . . :
説明 . . . . . : Microsoft Hyper-V ネットワーク アダプ
ター
物理アドレス . . . . . : 00-1D-D8-B7-1C-08
DHCP 有効 . . . . . : はい
自動構成有効 . . . . . : はい
リンクローカル IPv6 アドレス . . . . . : fe80::a81d:9de0:4676:dbca%13(優先)
IPv4 アドレス . . . . . : 192.168.1.17(優先)
サブネット マスク . . . . . : 255.255.255.0
リース取得 . . . . . : 2013年5月29日 10:42:54
リースの有効期限 . . . . . : 2013年5月30日 10:42:55
デフォルト ゲートウェイ . . . . . : 192.168.1.1
DHCP サーバー . . . . . : 10.0.0.1
DHCPv6 IAID . . . . . : 268440925
DHCPv6 クライアント DUID . . . . . : 00-01-00-01-19-22-55-25-00-1D-D8-B7-1C-08
DNS サーバー . . . . . : 192.168.0.10
```

SC2012 VMM SP1 での DHCP 実装

Microsoft Corporation: ¥Device¥NPF_{55C8420E-4153-44CF-AB77-99E7D096E77F} [Wireshark 1.8.7...]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
34	16.1699230	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xb2d43b7
35	16.1710630	10.0.0.1	192.168.1.17	DHCP	328	DHCP Offer - Transaction ID 0xb2d43b7
36	16.1715860	0.0.0.0	255.255.255.255	DHCP	357	DHCP Request - Transaction ID 0xb2d43b7
37	16.1722860	10.0.0.1	192.168.1.17	DHCP	328	DHCP ACK - Transaction ID 0xb2d43b7

Frame 35: 328 bytes on wire (2624 bits), 328 bytes captured (2624 bits) on interface 0

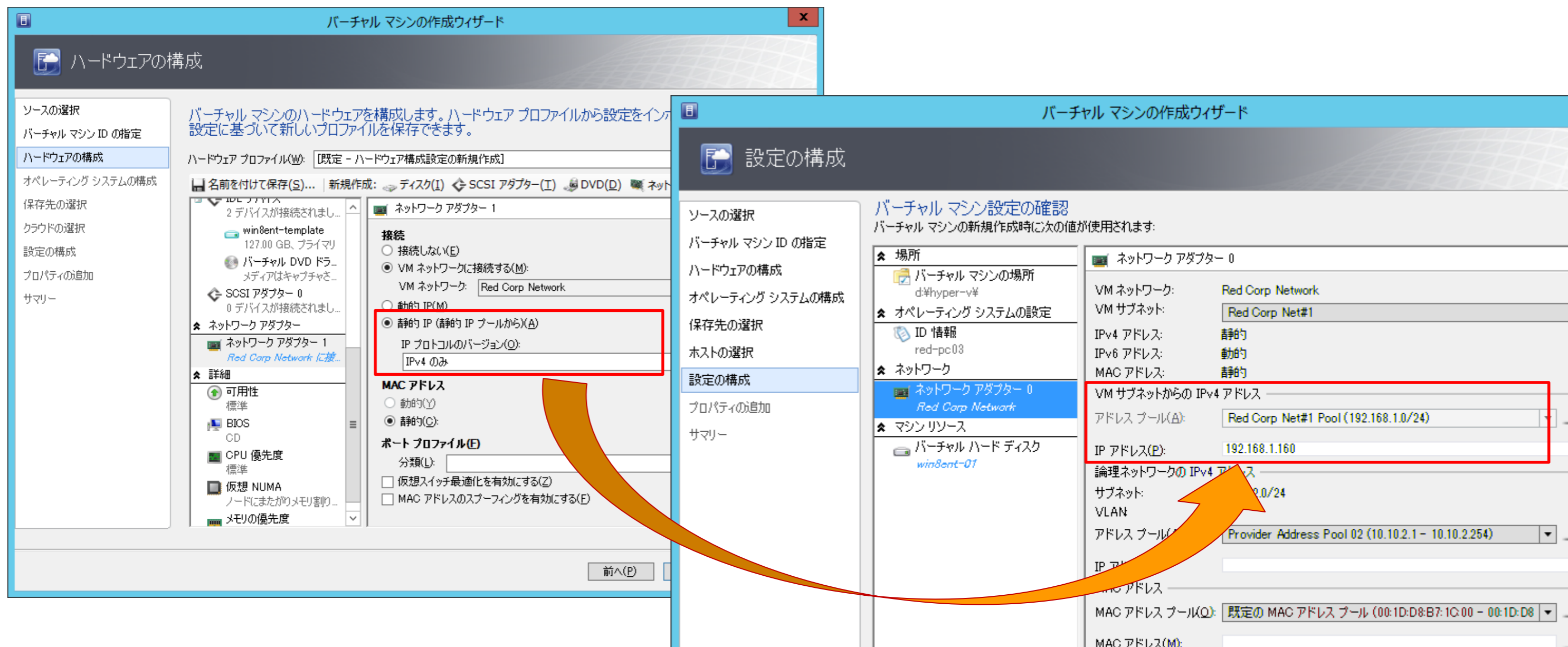
- Ethernet II, Src: 12:34:56:78:90:ab (12:34:56:78:90:ab), Dst: Microsof_b7:1c:08 (00:1d:d8:b7:1c:08)
 - Destination: Microsof_b7:1c:08 (00:1d:d8:b7:1c:08)
 - Source: 12:34:56:78:90:ab (12:34:56:78:90:ab)
Type: IP (0x0800)
- Internet Protocol Version 4, Src: 10.0.0.1 (10.0.0.1), Dst: 192.168.1.17 (192.168.1.17)
- User Datagram Protocol, Port: bootps (67), Dst Port: bootpc (68)
- Bootstrap Protocol

存在しえない MAC Address

Network 内に存在しない IP Address

テンプレート展開での静的割り当て

- テンプレート展開時に、静的設定及び IP アドレスの指定が可能



テンプレート展開での静的割り当て

- バーチャルマシン、プール割り当てとともに静的設定であることを確認

インターネット プロトコル バージョン 4 (TCP/IPv4) のプロパティ

全般

ネットワークでこの機能がサポートされている場合は、IP 設定を自動的に取得することができます。サポートされていない場合は、ネットワーク管理者に適切な IP 設定を問い合わせてください。

☐ IP アドレスを自動的に取得する(O)

☒ 次の IP アドレスを使う(S):

IP アドレス(I): 192 . 168 . 1 . 160

サブネット マスク(U): 255 . 255 . 255 . 0

デフォルト ゲートウェイ(D): 192 . 168 . 1 . 1

☐ DNS サーバーのアドレスを自動的に取得する(B)

☒ 次の DNS サーバーのアドレスを使う(E):

優先 DNS サーバー(P): 192 . 168 . 0 . 10

代替 DNS サーバー(A): . . .

☐ 終了時に設定を検証する(L)

詳細設定(Y)...

OK キャンセル

DHCP割り当て

```
管理者: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\M...
PS C:\Windows\system32> $pool = Get-SCStaticIPAddressPool -Name "Red Corp Net#1 Pool"
PS C:\Windows\system32> Get-SCIPAddress -StaticIPAddressPool $pool

Name                : 192.168.1.6
Address              : 192.168.1.6
AllocatingAddressPool : Red Corp Net#1 Pool (192.168.1.2 - 192.168.1.254)
AssignedToID         : a102e19b-9255-4a13-88c2-099023dcfbdf
AssignedToType        : VirtualNetworkAdapter
Type                 : DedicatedIP
State                : Assigned
Description           : WNV CA address for DHCP
ServerConnection     : Microsoft.SystemCenter.VirtualMachineManager.Remoting.S
                      : erverConnection
ID                   : 02e6b23e-f8c1-4f5f-8161-e5e537a298af
IsViewOnly           : False
ObjectType            : AllocatedIPAddress
MarkedForDeletion    : False
IsFullyCached        : True

Name                : 192.168.1.160
Address              : 192.168.1.160
AllocatingAddressPool : Red Corp Net#1 Pool (192.168.1.2 - 192.168.1.254)
AssignedToID         : f87df9f8-e097-467a-9f21-3dba202281fe
AssignedToType        : VirtualNetworkAdapter
Type                 : DedicatedIP
State                : Assigned
Description           : red-pc03
ServerConnection     : Microsoft.SystemCenter.VirtualMachineManager.Remoting.S
                      : erverConnection
ID                   : 3cffbb32-0564-4c8a-879a-381b9b186d28
IsViewOnly           : False
ObjectType            : AllocatedIPAddress
MarkedForDeletion    : False
IsFullyCached        : True
```

テンプレート展開での静的割り当て

- 静的 IP アドレスとして指定できるアドレスは、IP プールの範囲内のアドレス
 - 範囲外のアドレスを指定すると、ジョブが失敗します

The screenshot shows the SCVMM console with a table of jobs. The job '静的 IP アドレスプールから IP アドレスを割り当て' (Assign IP address from static IP address pool) is marked as failed. The details pane shows the command 'Grant-SCIPAddress' and an error message (Error 13666) stating that the specified IP address (192.168.101.99) is outside the range of the IP pool (192.168.101.101 to 192.168.101.200).

名前	状態	開始時刻	結果名	所有者
静的 IP アドレスプールから IP アドレスを割り当て	失敗	2013/05/22 17:00:01	失敗したジョブ	CONTOSO¥Administr...
バーチャル マシンの状態の...	完了	2013/05/22 16:52:41	blue-pctest99	CONTOSO¥vmadmin

静的 IP アドレスプールから IP アドレスを割り当て

状態: 失敗
コマンド: Grant-SCIPAddress
結果名:
開始: 2013/05/22 17:00:01
期間: 00:00:00
所有者: CONTOSO ¥Administrator

エラー (13666)
指定されたアドレスは、アドレス プールで管理されるアドレス範囲 の外にあります。
指定されたアドレス: 192.168.101.99
範囲の開始アドレス: 192.168.101.101
範囲の終了アドレス: 192.168.101.200
IP アドレス プール: Blue Corp TestNet Pool

推奨される操作
プールの管理範囲内のアドレスを指定してください。

The screenshot shows a PowerShell command prompt window with the following text:

```
PS C:\Windows\system32> Grant-SCIPAddress -StaticIPAddressPool $IPPool -GrantToOb...  
jectType VirtualNetworkAdapter -GrantToObjectID $vNICs[0].ID -Description $VM.N...  
ame -IPAddress 192.168.101.99  
Grant-SCIPAddress : 指定されたアドレスは、アドレス プールで管理されるアドレス範...  
囲 の外にあります。  
指定されたアドレス: 192.168.101.99  
範囲の開始アドレス: 192.168.101.101  
範囲の終了アドレス: 192.168.101.200  
IP アドレス プール: Blue Corp TestNet Pool (エラー ID: 13666)  
  
プールの管理範囲内のアドレスを指定してください。  
  
ジョブを再開するには、次のコマンドを実行してください:  
PS> Restart-Job -Job (Get-VMMServer localhost | Get-Job | where [ $_.ID -eq "[3...  
81ef3c0-c63e-49dc-9b15-fc59952d4ffd]" ] )  
発生場所 行:1 文字:1  
+ Grant-SCIPAddress -StaticIPAddressPool $IPPool -GrantToObjectType VirtualNetw...  
ork...  
+ ~~~~~  
+ CategoryInfo          : ReadError: (:) [Grant-SCIPAddress], CarmineExcep...  
tion  
+ FullyQualifiedErrorId : 13666,Microsoft.SystemCenter.VirtualMachineManag...  
er.Cmdlets.GrantSCIPAddress  
PS C:\Windows\system32>
```

- 当然のことながら、ライブマイグレーションを実施しても静的 IP アドレスは維持

既存バーチャルマシンでの静的割り当て

- 静的 IP アドレスの割り当ては、GUI 上ではテンプレート展開時のみ可能
 - 既存バーチャルマシンの設定を確認しても、静的 IP は選択不可



- PowerShell を利用することにより、既存バーチャルマシンでも静的 IP 設定が実施可能
 - 但し、割り当て可能な IP アドレスは、IP プールの範囲内のアドレス
 - 従って、ホストアドレス『1』は指定不可

既存バーチャルマシンでの静的割り当て

```
# "" 内で静的 IP アドレスを割り当てるバーチャルマシン名を指定
$VM_Name = "VMName"
# "" 内で割り当てる VM ネットワーク名を指定
$VMNetwork_Name = "VM Network"
# "" 内で割り当てる VM サブネット名を指定
$VMSubnet_Name = "VM Subnet"
# "" 内で割り当てる IP アドレスのプール名を指定
$IPPool_Name = "VM Network Pool"
# "" 内で割り当てる IP アドレスを指定
$VM_IPAddress = "192.168.1.10"
# "" 内で割り当てる MAC アドレスのプール名を指定
$MACPool_Name = "既定の MAC アドレス プール"
# "" 内で使用する仮想スイッチ名を指定
$vswitch_Name = "vswitch"

$VM = Get-SCVirtualMachine -Name $VM_Name
$vNICsMAC = Get-SCVirtualNetworkAdapter -VM $VM
$vNICs = $VM.VirtualNetworkAdapters
$MACPool = Get-SCMACAddressPool -Name $MACPool_Name
$IPPool = Get-SCStaticIPAddressPool -Name $IPPool_Name
$vNICsMAC = Get-SCVirtualNetworkAdapter -VM $VM
Grant-SCMACAddress -MACAddressPool $MACPool -VirtualNetworkAdapter $vNICsMAC
$MACAddr = Get-SCMACAddress | Where-Object {$_.AssignedToID -eq $vNICsMAC.ID}
Grant-SCIPAddress -StaticIPAddressPool $IPPool -GrantToObjectType VirtualNetworkAdapter -GrantToObjectID $vNICs[0].ID -Description $VM.Name -IPAddress $VM_IPAddress
```

既存バーチャルマシンでの静的割り当て

```
$VirtualNetworkAdapter = Get-SCVirtualNetworkAdapter -Name $VM_Name -ID $vNICs.ID
```

```
$VMNetwork = Get-SCVMNetwork -Name $VMNetwork_Name
```

```
$VMSubnet = Get-SCVMSubnet -Name $VMSubnet_Name | where {$_.VMNetwork.ID -eq $VMNetwork.ID}
```

```
Set-SCVirtualNetworkAdapter -VirtualNetworkAdapter $VirtualNetworkAdapter -VMNetwork $VMNetwork -VMSubnet $VMSubnet -VirtualNetwork $vswitch_Name -MACAddress $MACAddr.Address -MACAddressType Static -IPv4Address $VM_IPAddress -IPv4AddressType Static -IPv6AddressType Dynamic -NoPortClassification - EnableVMNetworkOptimization $false
```

既存バーチャルマシンでの静的割り当て

- 実行結果

The screenshot shows the 'ジョブ' (Job) window in Hyper-V Manager. The window title is 'ジョブ'. Below the title bar, there is a search bar and a list of recent jobs. The list is titled '最近のジョブ (107)'. The list has columns: 名前 (Name), 状態 (Status), 開始時刻 (Start Time), 結果名 (Result Name), and 所有者 (Owner). The list shows four jobs, all with a green checkmark icon and the status '完了' (Completed). The jobs are: 'バーチャル マシンの起動' (Virtual Machine Start), 'ネットワーク アダプターのプロ...' (Network Adapter Properties), '静的 IP アドレス プールか...' (Static IP Address Pool), and 'MAC アドレス プールから...' (MAC Address Pool). The job 'ネットワーク アダプターのプロ...' is highlighted with a red box. Below the list, there is a section for the selected job, 'ネットワーク アダプターのプロパティの変更' (Network Adapter Properties Change). This section shows the job status as '完了' (Completed), the command as 'Set-SCVirtualNetworkAdapter', the result name as 'red-pc05', the start time as '2013/05/30 14:19:17', the duration as '00:00:04', and the owner as 'CONTOSO ¥Administrator'. To the right of this information is a table showing the properties of the network adapter. The table has columns: プロパティ (Property), 以前の値 (Previous Value), and 新しい値 (New Value). The table lists the following properties: 'VirtualNetwork' (Vswitch1), 'EthernetAddressType' (静的 (Static)), 'EthernetAddress' (00:1D:D8:B7:1C:14), 'IPv4AddressType' (静的 (Static)), 'VMNetwork' (Red Corp Network), and 'VMSubnet' (Red Corp Net#2). At the bottom of the window, there are tabs for 'サマリー' (Summary), '詳細' (Details), and '変更履歴' (Change History). The 'サマリー' tab is selected. At the bottom right, there are checkboxes for '新規オブジェクトが作成されたときにこのウィンドウを表示する(S)' (Show this window when a new object is created) and buttons for '再開(R)' (Resume) and 'キャンセル(C)' (Cancel).

名前	状態	開始時刻	結果名	所有者
✓ バーチャル マシンの起動	完了	2013/05/30 14:19:27	red-pc05	CONTOSO¥Administra...
✓ ネットワーク アダプターのプロ...	完了	2013/05/30 14:19:17	red-pc05	CONTOSO¥Administra...
✓ 静的 IP アドレス プールか...	完了	2013/05/30 14:19:02	192.168.2.121	CONTOSO¥Administra...
✓ MAC アドレス プールから...	完了	2013/05/30 14:19:02	00:1D:D8:B7:1C:14	CONTOSO¥Administra...

プロパティ	以前の値	新しい値
仮想ネットワーク アダプター - red-pc05		
VirtualNetwork	(なし)	Vswitch1
EthernetAddressType	動的	静的
EthernetAddress	(なし)	00:1D:D8:B7:1C:14
IPv4AddressType	動的	静的
VMNetwork	(なし)	Red Corp Network
VMSubnet	(なし)	Red Corp Net#2

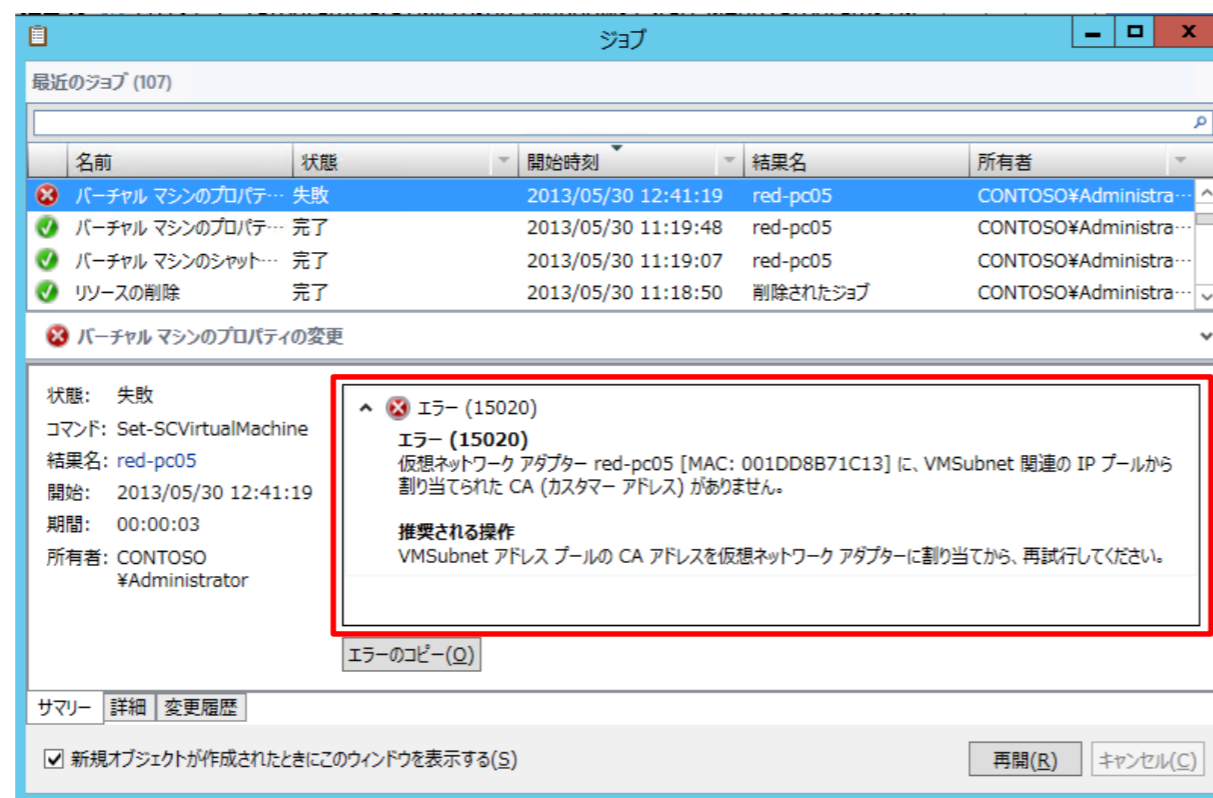


既存バーチャルマシン
での静的割り当て

DEMO

静的割り当ての注意点

- バーチャルマシンのNIC設定にて、接続先のVMネットワークやVMサブネットを変更、もしくは一度『接続なし』にした後に再度同じVMネットワークに接続した場合、以下のエラーが発生して構成変更が失敗します。



- 『 Grant-SCIPAddress 』 Cmdlet にて IP プールからアドレスを手動にて割り当てる必要がありますので、注意が必要です。

Broadcast over NVGRE

NVGRE でのブロードキャストの扱い

- Broadcast を利用するアプリケーションを使用しての検証を実施、以下の結果となりました。
 - 同一ホスト上の同一 仮想Networkに接続されている場合は、Broadcast 使用可能。
 - 異なるホスト上の場合は、同一仮想 Network でも Broadcast 使用不可。
- この結果から、同一物理ホスト上の同一仮想 Network 間は NVGRE によるカプセル化は行われていない模様です（仮想 Switch で折り返し通信？）
 - 同一物理ホスト上の仮想マシン間の通信で使用する L2 Frame Size を確認したところ、1518byte でした。

と、第 6 回 WS2012CD で
説明させていただきましたが……

NVGRE でのブロードキャストの扱い

- PowerShell、もしくは SC2012 VMM GUI からマルチキャストプールを設定することにより、仮想ネットワーク上で Broadcast が利用可能です。
 - PowerShell : 『 New-NetVirtualizationCustomerRoute 』 Cmdlet
 - SC2012 VMM : 論理ネットワーク → IP プールの作成
- 『 255.255.255.255/32 』 『ホストアドレス .255/32 (ex:192.168.1.255/32) 』 『 224.0.0.0/4 』 が VSID 単位で 一意のマルチキャストアドレスにマッピングされます。
- これにより、バーチャルマシン間のブロードキャスト通信が可能になります。
- DHCP のブロードキャストはフィルタードライバーで処理されてしまうため、仮想ネットワーク内で DHCP サーバーを運用する (BYO DHCP) ことはできません。

PowerShell での 実装

- PowerShell で ブロードキャスト通信を実装する場合、以下の Cmdlet を使用します。

- 使用コマンド : New-NetVirtualizationCustomerRoute
- コマンド使用例 :

```
New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}"  
-VirtualSubnetID "5001" -DestinationPrefix "224.0.0.0/4" -NextHop "239.0.1.1" -Metric 255
```

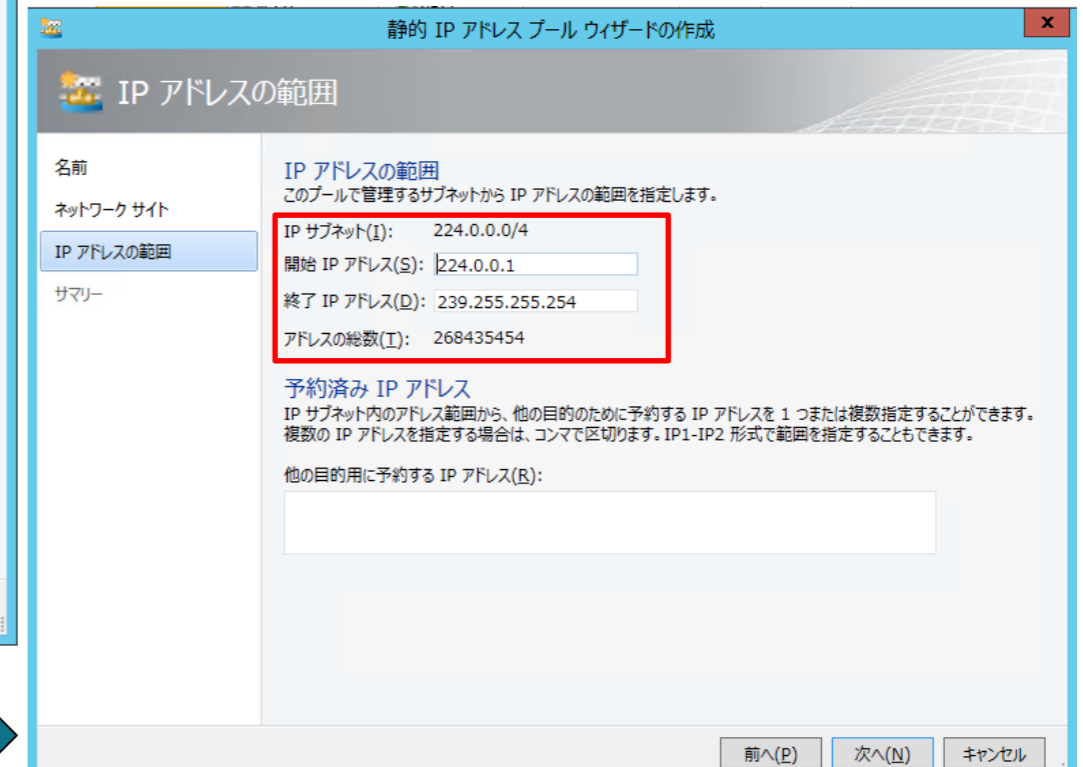
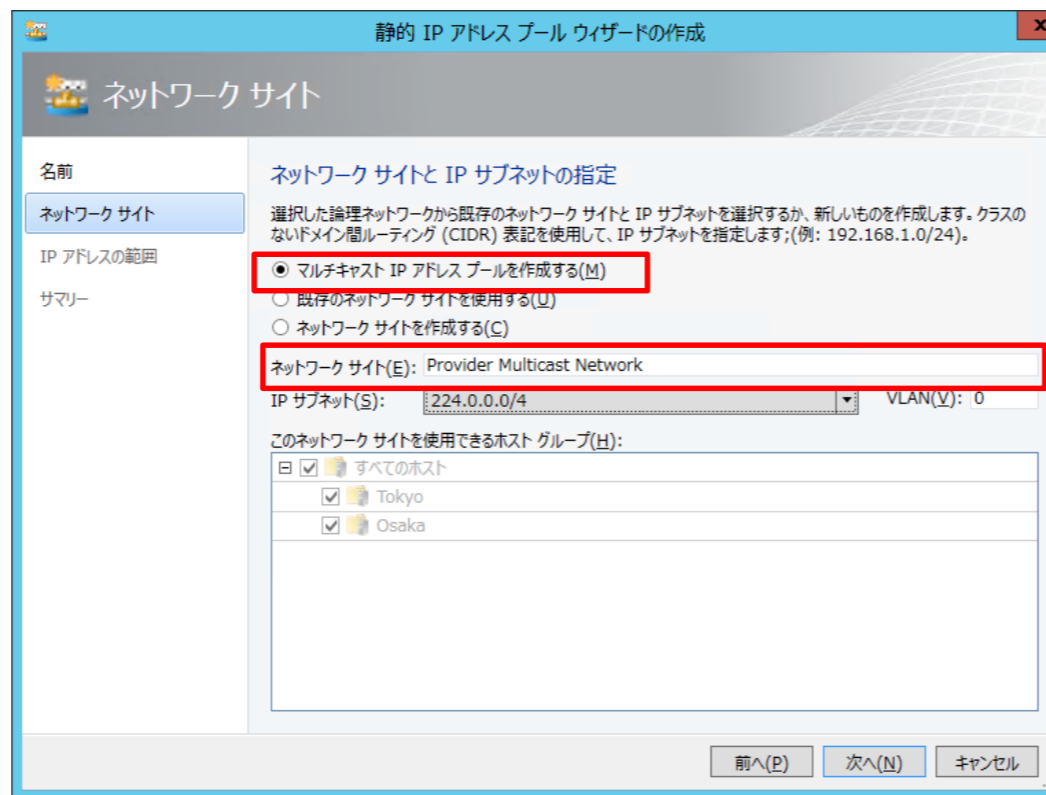
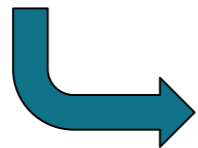
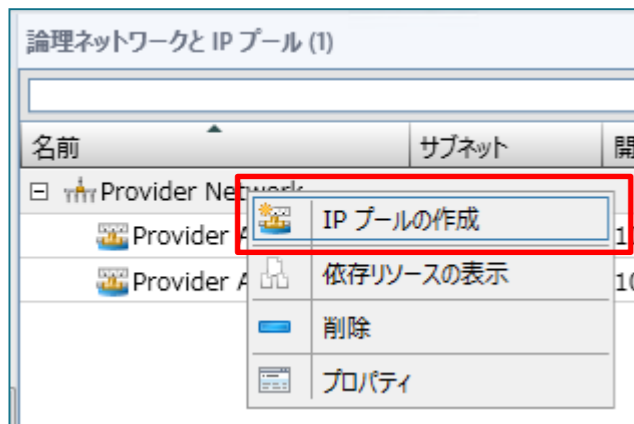
```
New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}"  
-VirtualSubnetID "5001" -DestinationPrefix "255.255.255.255/32" -NextHop "239.0.1.1" -Metric 255
```

```
New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}"  
-VirtualSubnetID "5001" -DestinationPrefix "192.168.1.255/32" -NextHop "239.0.1.1" -Metric 255
```

- ポイント : 『 -NextHop 』で割り当てたい マルチキャストアドレスを指定する。
指定可能な範囲は 224.0.0.0/4 (224.0.0.1 から 239.255.255.255) 。
MAC アドレスはマルチキャスト MAC アドレスとして自動生成されるため、設定不要

SC2012 VMM での実装

- 『論理ネットワーク』→『IP プールの作成』にて、マルチキャストアドレスプールを作成します。
 - 予めネットワークサイトを作成しておく必要はありません（作成しても選択できません）
 - デフォルトは『224.0.0.1 ~ 239.255.255.254』の 268,435,454 アドレスです。
→ 割り当て可能な VSID は16,773,118 です。



マルチキャスト割り当て確認

- 同一 VSID の複数のブロードキャスト/マルチキャストアドレスに対して、同一のプロバイダーマルチキャストアドレスが割り当てられていることが確認できます。
- 同一 RDID であっても VSID が異なる場合には、異なるプロバイダーマルチキャストアドレスが割り当てられています。

```
管理: Windows PowerShell
PS C:\Users\Administrator\CONTOSO> Get-NetVirtualizationCustomerRoute -RoutingDomainID "[FEED0D1C-C86A-4535-8F2F-495FF77B5A58]"

RoutingDomainID : {FEED0D1C-C86A-4535-8F2F-495FF77B5A58}
VirtualSubnetID : 12061486
DestinationPrefix : 255.255.255.255/32
NextHop          : 239.0.1.2
Metric           : 0

RoutingDomainID : {FEED0D1C-C86A-4535-8F2F-495FF77B5A58}
VirtualSubnetID : 2730398
DestinationPrefix : 255.255.255.255/32
NextHop          : 239.0.1.1
Metric           : 0

RoutingDomainID : {FEED0D1C-C86A-4535-8F2F-495FF77B5A58}
VirtualSubnetID : 12061486
DestinationPrefix : 224.0.0.0/4
NextHop          : 239.0.1.2
Metric           : 0

RoutingDomainID : {FEED0D1C-C86A-4535-8F2F-495FF77B5A58}
VirtualSubnetID : 2730398
DestinationPrefix : 224.0.0.0/4
NextHop          : 239.0.1.1
Metric           : 0

RoutingDomainID : {FEED0D1C-C86A-4535-8F2F-495FF77B5A58}
VirtualSubnetID : 2730398
DestinationPrefix : 192.168.2.255/32
NextHop          : 239.0.1.1
Metric           : 0

RoutingDomainID : {FEED0D1C-C86A-4535-8F2F-495FF77B5A58}
VirtualSubnetID : 12061486
DestinationPrefix : 192.168.1.255/32
NextHop          : 239.0.1.2
Metric           : 0
```

実際のブロードキャストパケット

The image shows a Wireshark packet capture of a NetBIOS broadcast packet. The packet list on the left shows five packets. Packet 13 is selected, and its details are shown on the right. The packet is an NBNS Name query response. The details pane shows the following structure:

- Frame 13: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits) on interface 0
- Ethernet II, Src: Cisco_62:23:2e (44:03:a7:62:23:2e), Dst: IPv4mcast_00:01:01 (01:00:5e:00:01:01)
- Internet Protocol Version 4, Src: 10.10.1.19 (10.10.1.19), Dst: 239.0.1.1 (239.0.1.1)
- Generic Routing Encapsulation (transparent Ethernet bridging)
 - Flags and version: 0x2000
 - Protocol Type: Transparent Ethernet bridging (0x6558)
 - Key: 0x6b30f107
- Ethernet II, Src: Microsof_b7:1c:04 (00:1d:d8:b7:1c:04), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
- Destination: Broadcast (ff:ff:ff:ff:ff:ff)
- Source: Microsof_b7:1c:04 (00:1d:d8:b7:1c:04)
- Type: IP (0x0800)
- Internet Protocol Version 4, Src: 192.168.1.8 (192.168.1.8), Dst: 192.168.1.255 (192.168.1.255)
- User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137)
- NetBIOS Name Service

The packet bytes are shown at the bottom, with the first 20 bytes highlighted in blue. The packet is a NetBIOS Name Service (NBNS) query response.

Outer Frame は
マルチキャストアドレス

GRE Header

Inner Frame は
ブロードキャスト (NBNS)

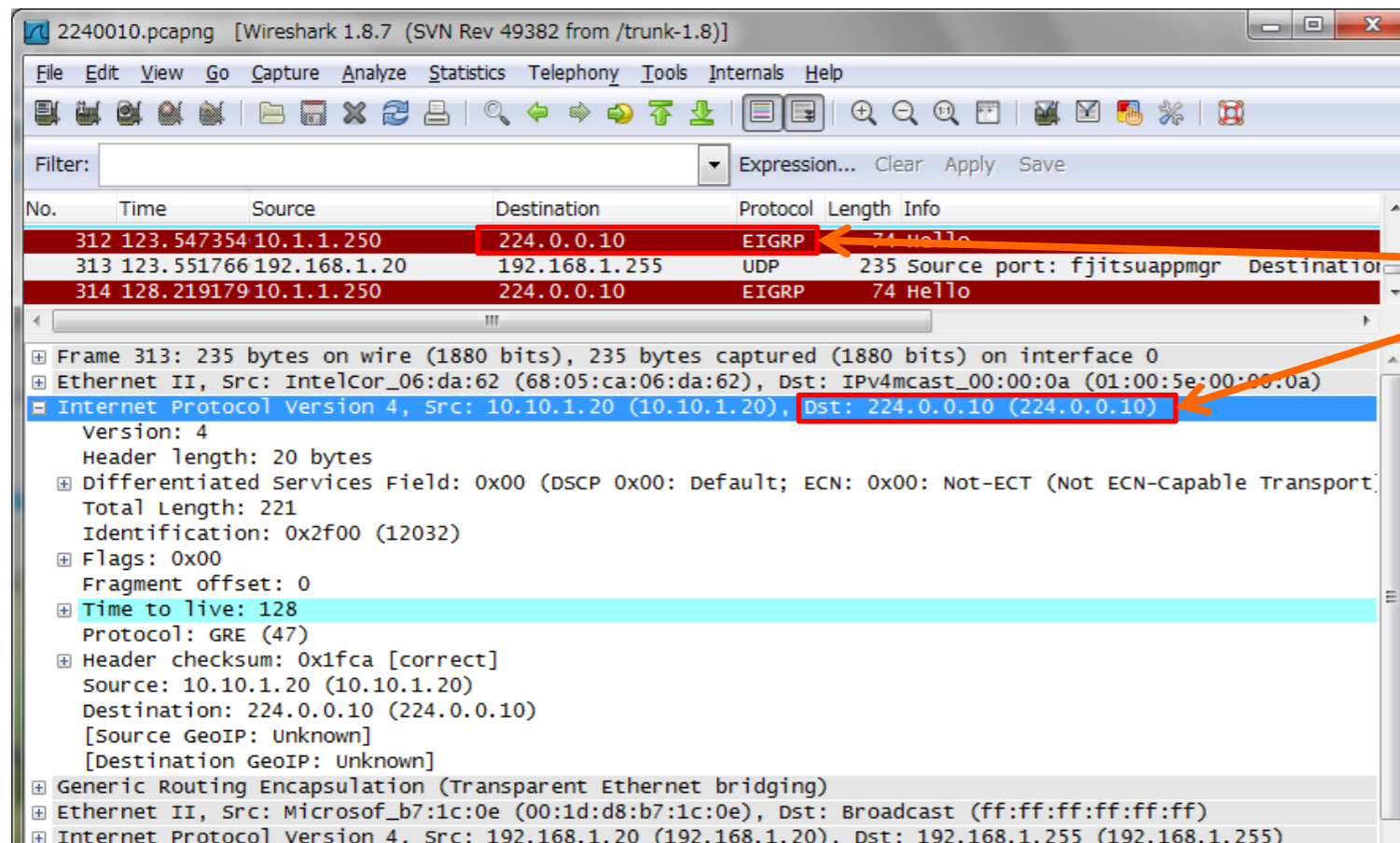


Broadcast over NVGRE

DEMO

ブロードキャスト通信実装時の注意点

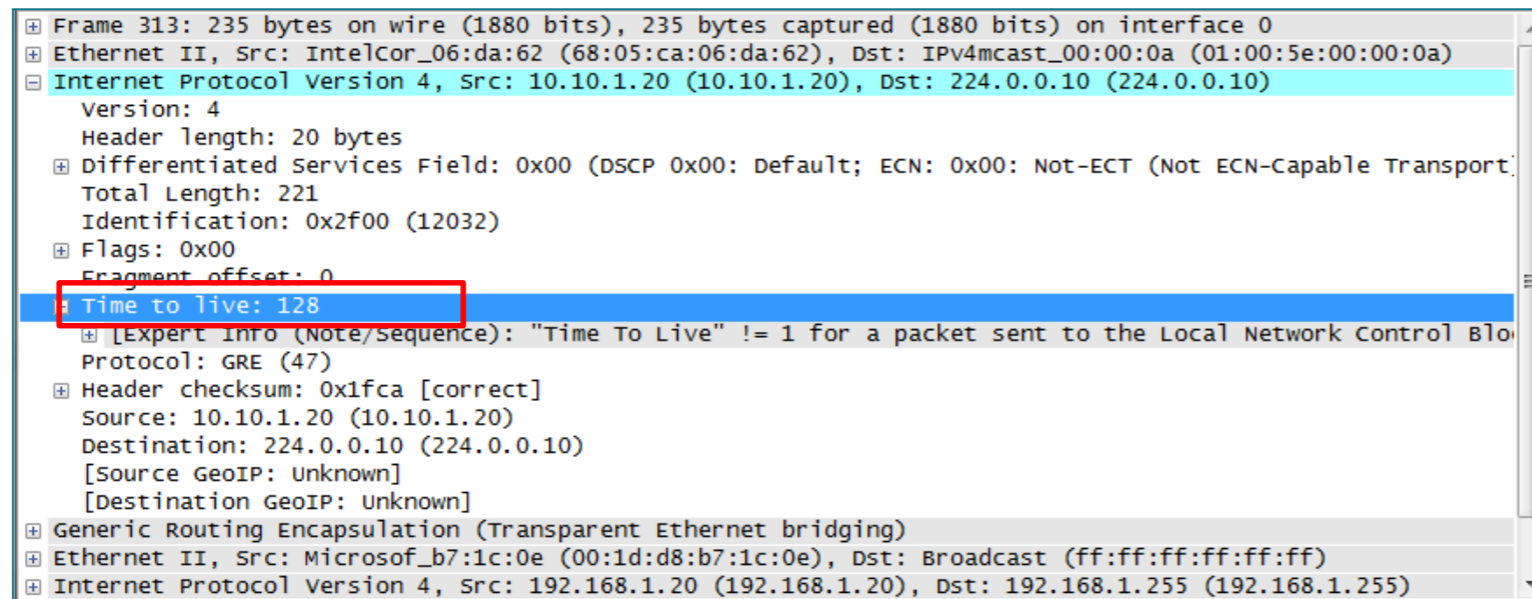
- SC2012 VMM からデフォルト設定でマルチキャスト IP プールを作成すると、『224.0.0.1』から割り当てられることになります。
- 224.0.0.0/24 は『予約済みリンクローカルアドレス』として IANA によって予約されている（RFC 1112）ため、利用しないことをお勧めします。
 - 以下のように既存マルチキャスト通信と同じアドレスを使用してしまいます。



ルーティングプロトコルが使用している
マルチキャストアドレスと、
同じアドレスを使用してしまっている

ブロードキャスト通信実装時の注意点

- また、224.0.0.0/24 を使用した場合、NVGRE 用に生成されるマルチキャストの Time-To-Live(TTL) 値が『 128 』となっています。

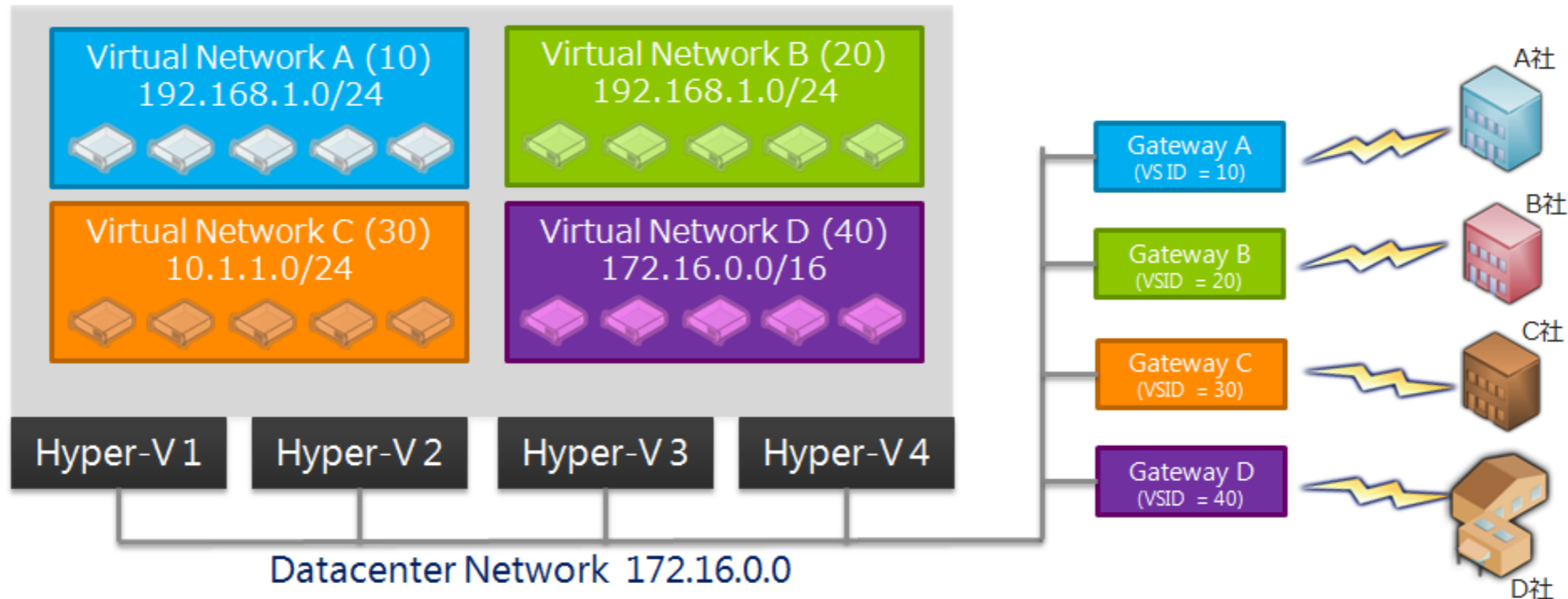


- 『予約済みリンクローカルアドレス』の TTL 値は、通常『 1 』がセットされています。
- リンクローカルであるため、通常ルーティングは行われません。
- マルチキャストがルーティングされない、などの事態の発生も予想される為、239.0.1.0/24 ~ 239.255.255.255/24 （除く 239.128.0.0/24 ）（限定スコープアドレス）の利用をお勧めします。

Network Virtualization Gateway

Network Virtualization Gateway

- 仮想 Network と物理 Network の接続点
- NVGRE のカプセリング処理と、物理 Network への Routing を実施
 - VPN Gateway や NAT Router として動作
- Gateway がいないと、仮想 Network は独立した Network として動かざるを得ないので、Network Virtualization を考える上で Gateway は非常に重要なコンポーネント

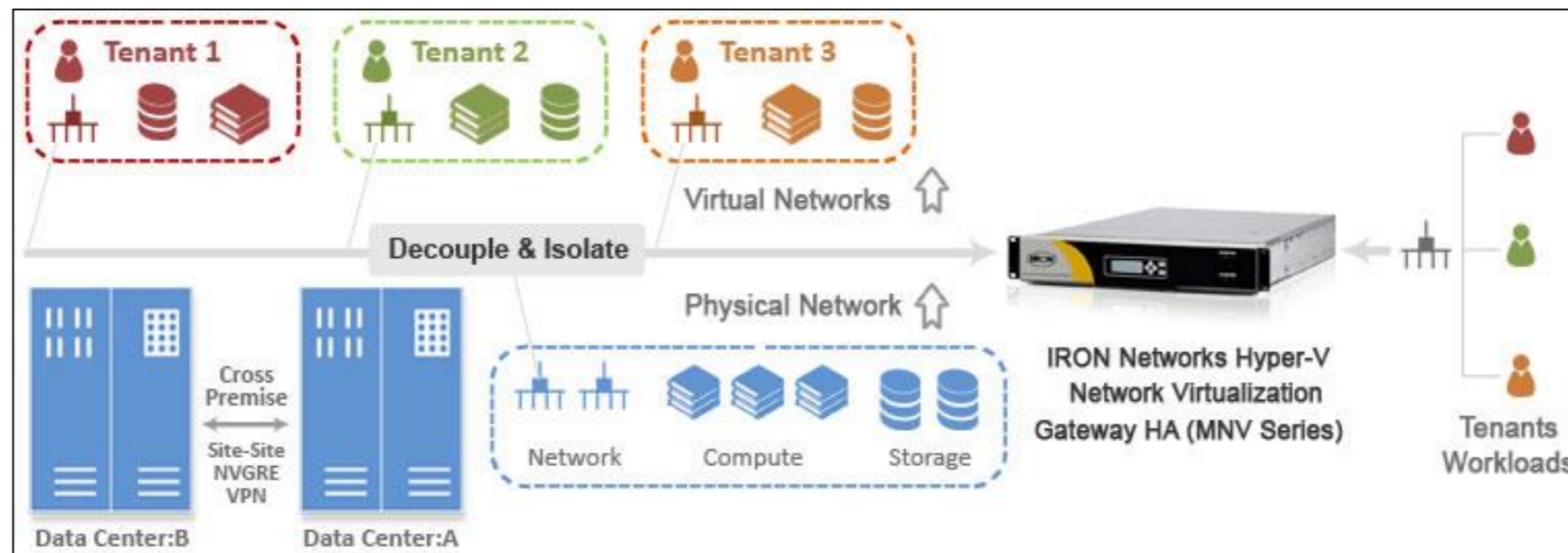


Network Virtualization Gateway と SC2012 VMM

- SC2012 VMM での Network Virtualization では、Gateway は『 Gateway Provider 』とのセットで実装される。
- 『 Gateway Provider 』は SC2012 VMM Serverに導入され、SC2012 VMM と連携して、Gateway に対して必要な設定（ VSID や Customer Address / Provider Address 、 VM Network の Routing Table 等）を送信／設定を実施
 - Provider は、Gateway のベンダーから提供
 - Provider は SC2012 VMM に導入し、 VM Subnet のプロパティ内で設定
- Gateway 用として、単純に 2 Ethernet な仮想マシンを準備／接続しても、SC2012 VMM からはその仮想マシンが『 Gateway 用の仮想マシン』として認識できない為、使用不可
 - Gateway （ Software 実装／ Hardware 実装を問わず）を SC2012 VMM に認識させる為に、『 Gateway Provider 』が必要
- 3rd Party から提供予定。

3rd Party 実装例

- IRON Networks (旧 nAppliance Networks)
Gateway MNV Appliance - Microsoft Hyper-V Network Virtualization Gateway
URL : <http://www.ironnetworks.com/products/NetGateway-MNV>



- F5 Networks BIG-IP LTM VE (予定)
URL : <http://download.microsoft.com/download/C/F/2/CF2F9D51-5D9E-45FE-B134-D0783220DCE8/20130315-F5.pdf>

WNV Gateway の接続方式

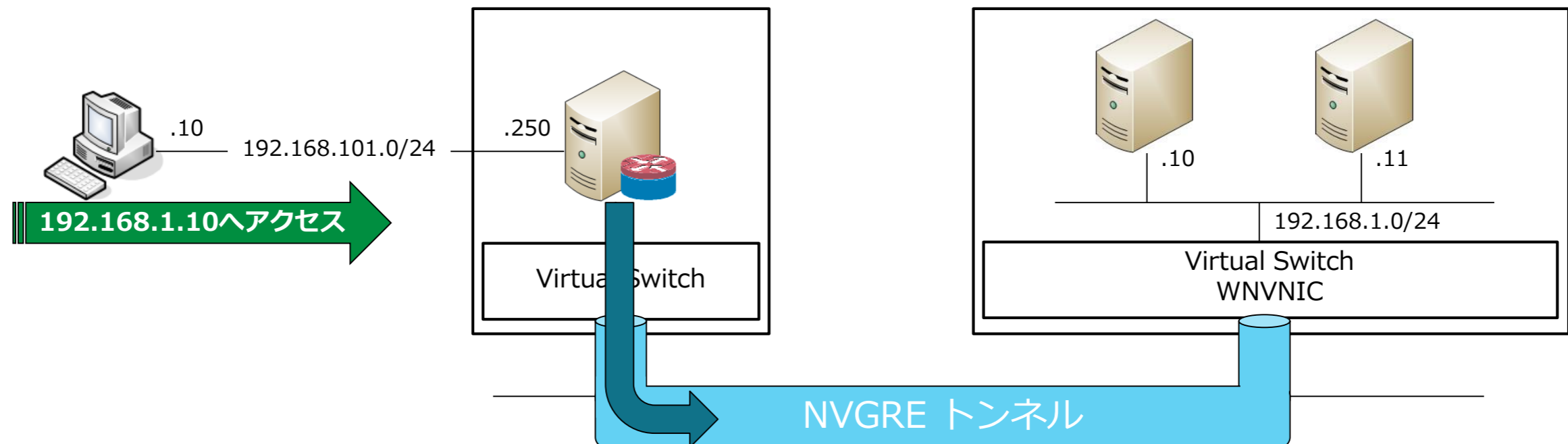
- WNV Gateway で提供される接続方式は『ローカル接続』と『リモート接続』
 - ローカルネットワーク：ローカルルーティングで仮想ネットワークを接続
 - リモートネットワーク：VPN（IKEv2 / PPTP / L2TP）経由でリモートネットワークと仮想ネットワークを接続



- Gateway は、VM ネットワーク単位で設定可能です。
 - 個別設定が可能になっています
- Gateway に対する諸設定（ルーティング／VPN 設定）は、すべて SC2012 VMMから実施
 - そのための PowerShell Cmdlet も準備されています
 - 『*-SCNetworkGateway』 『*-SCVPNConnection』 etc...

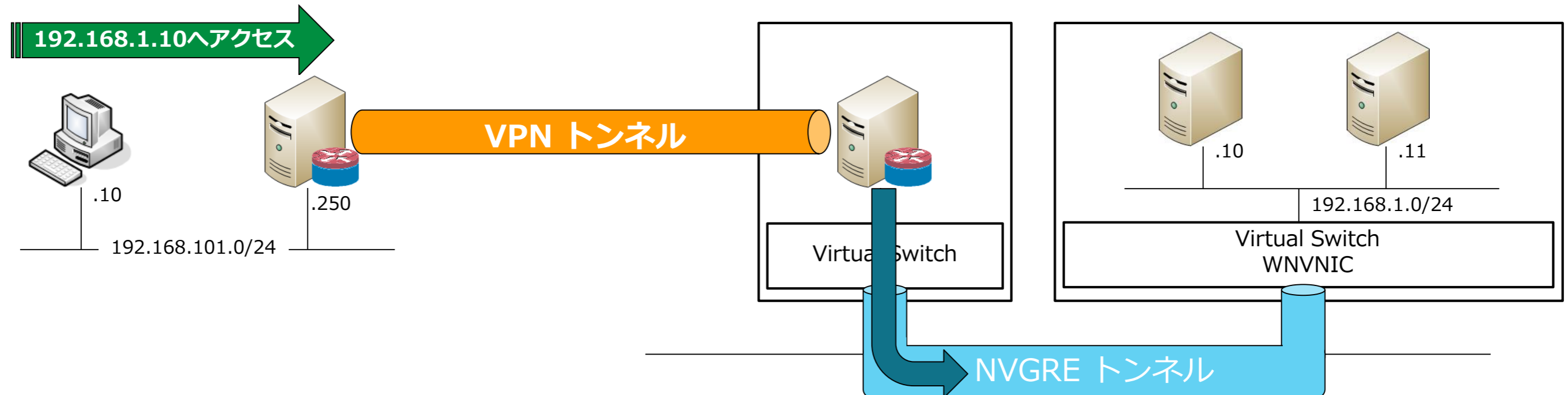
WNV Gateway の接続方式：ローカルネットワーク

- Gateway がローカルルーターとして稼働し、直接ルーティング接続を行う方式。
 - WNV Gateway がローカル接続可能なネットワークで実装可能
 - プライベートクラウドでの実装を想定
- 仮想ネットワークへのルーティングを WNV Gateway に対して設定することにより、通常のネットワークノードとして接続することが可能。
 - ルーティングプロトコルによるダイナミックルーティングの可否は、使用する WNV Gateway に依存します。



WNV Gateway の接続方式：リモートネットワーク

- Gateway が VPN の終端装置として稼働し、対向の VPN 機器と接続、VPN経由で Site-to-Site （ S2S ） 接続を行う方式。
 - WNV Gateway とVPN接続が可能であることが接続条件なため、公共ネットワーク（インターネット）経由でも安全にアクセスが可能
 - ハイブリッドクラウドでの実装を想定
- VPN の認証方式／暗号化方式は、使用する WNV Gateway と対向の VPN 機器に依存します。



WNV Gateway の接続方式：リモートネットワーク

- リモートネットワーク接続では、認証方法（事前共有キー／証明書）、暗号化方式、VPN プロトコルがSC2012 VMM コンソールから指定可能です。

Red Corp Network のプロパティ

名前

VM サブネット

ゲートウェイ

VPN 接続

VPN 設定

アクセス

VPN エンドポイントとの接続の詳細情報を入力してください

リモート エンドポイント(D): 192.168.0.210

☒ 次の資格情報を使用して認証する(C)

実行アカウント(B): VPN 参照(B)...

☐ VPN サーバーで認証に証明書の自動選択を使用する(U)

☐ 証明書を使用して認証(A)

証明書(E): 参照(B)...

Red Corp Network のプロパティ

名前

VM サブネット

ゲートウェイ

VPN 接続

VPN 設定

アクセス

VPN 接続設定

☐ 既定の接続設定を使用する(U)

認証方法(M): PSKOnly

暗号化方法(E): AES256

整合性チェック方法(H): SHA256

暗号化変換定数(C): DES3

認証変換定数(A): SHA256128

PFS グループ(F): PFS2048

Diffie-Hellman グループ(D): Group2

VPN プロトコル(O): IKEv2

IKEv2

L2TP

PPTP

WNV Gateway の接続方式：リモートネットワーク

- WNV Gateway がサポートする機能（認証方法、暗号化方式、VPN プロトコル等々）は『Get-SCNetworkGateway』Cmdlet にて取得が可能です。
- 対向の VPN 装置の選定や、設定確認時に利用すると便利です。



```
管理者: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Microsoft Management Console
PS C:\Windows\system32> Get-SCNetworkGateway

Name                           : WNV Gateway
Description                     :
ConnectionString                : 192.168.0.200
RunAsAccount                    : WNV Admin
NVGRESupported                  : True
NVGREMultiHopSupported          : False
MaxRoutingDomainSupported       : 2
MaxVSIIDSupported               : 200
MaxWNPoliciesSupported          : 1000
MaxVPNConnectionSupported       : 2
MaxVPNConnectionPerRoutingDomainSupported : 1
VPNEncryptionMethodsSupported   : DES, DES3, AES128, AES192, AES256
VPNIntegrityCheckMethodsSupported : MD5, SHA1, SHA256, SHA384
VPNCipherTransformsSupported    : DES, DES3, AES128, AES192, AES256, GCMAES128, GCMAES192, GCMAES256
VPNAuthenticationTransformsSupported : SHA256128, MD596, SHA196, GCMAES128, GCMAES192, GCMAES256
VPNPFSGroupsSupported           : None, PFS1, PFS2, PFS2048, ECP256, ECP384, PFSMM, PFS24
VPNDHGroupsSupported            : None, Group1, Group2, Group14, ECP256, ECP384, Group24
VPNProtocolSupported             : IKEv2
VPNAuthenticationSupported       : PSKOnly, MachineCertificates
DeviceId                        : GATEWAY
Manufacturer                     : Microsoft
Model                           : Win8VMHost
VMNetworkGateways                : [Red Corp Network_WNV Gateway, Blue Corp Network_WNV Gateway]
GatewayConnections               : [[17290f66-6723-4823-919c-aa65edc547dc]]
ConfigurationProvider             : Microsoft Software Gateway Provider
ServerConnection                 : Microsoft.SystemCenter.VirtualMachineManager.Remoting.ServerConnection
ID                               : c1563b42-79f3-4640-a3db-64293db8d88a
IsViewOnly                       : False
ObjectType                       : NetworkGateway
MarkedForDeletion                 : False
IsFullyCached                     : True
```



WNV Gateway

DEMO

NVGRE ホスト側負荷評価

NVGRE ホスト負荷試験

- 2 台の物理ホスト上に配置された、計 15 × 2 台のバーチャルマシンを使用
- バーチャルマシンは 2 台ずつ異なる VSID に所属（計 15 Virtual Network を使用）
- Disk I/O を伴わない、ネットワーク帯域計測ツールである『iperf.exe』を、ネットワーク負荷として使用
- iperf.exe を実行するバッチファイルを作成、タスクスケジューラにてすべてのバーチャルマシンで同時刻に実行するように設定
- 物理ホストを跨ぐように組み合わせた、15 組のバーチャルマシン間で iperf.exe による負荷を発生させ、その際のホスト側 CPU 利用率をパフォーマンスモニターにて計測
- 物理ホスト間のネットワーク帯域は 1Gbps （Gigabit Ethernet × 1）
- 同条件で iperf.exe を実行するバーチャルマシンの稼働数を 30 台／20 台／10 台と変更し、有意な差が発生するかを確認する

NVGRE ホスト負荷試験：試験環境

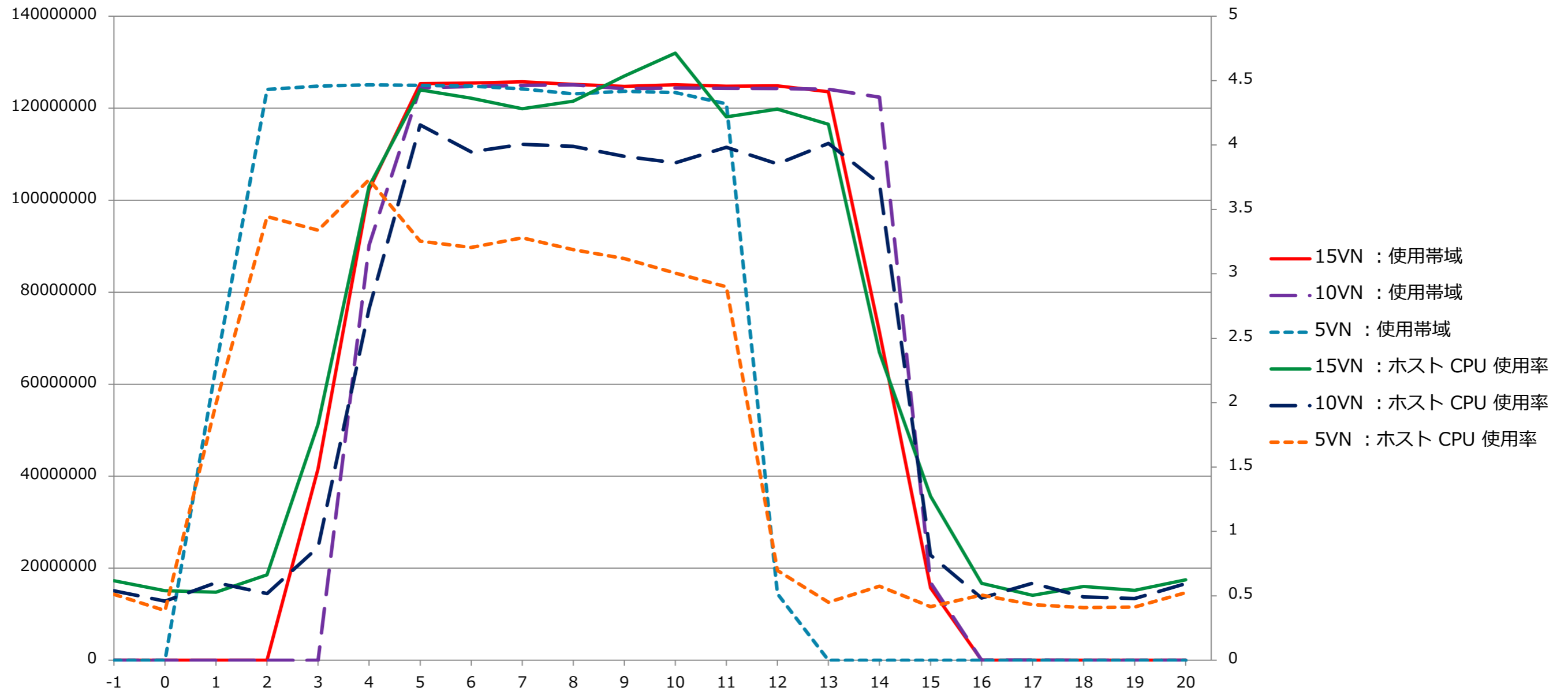
- 物理ホスト

- CPU : インテル Xeon プロセッサ X6550 (2GHz : 8 コア / 16 スレッド) × 2
- メモリ : 256GB (8GB × 32 : 1066MHz)
- HDD : 146GB SAS × 2 (RAID 0)

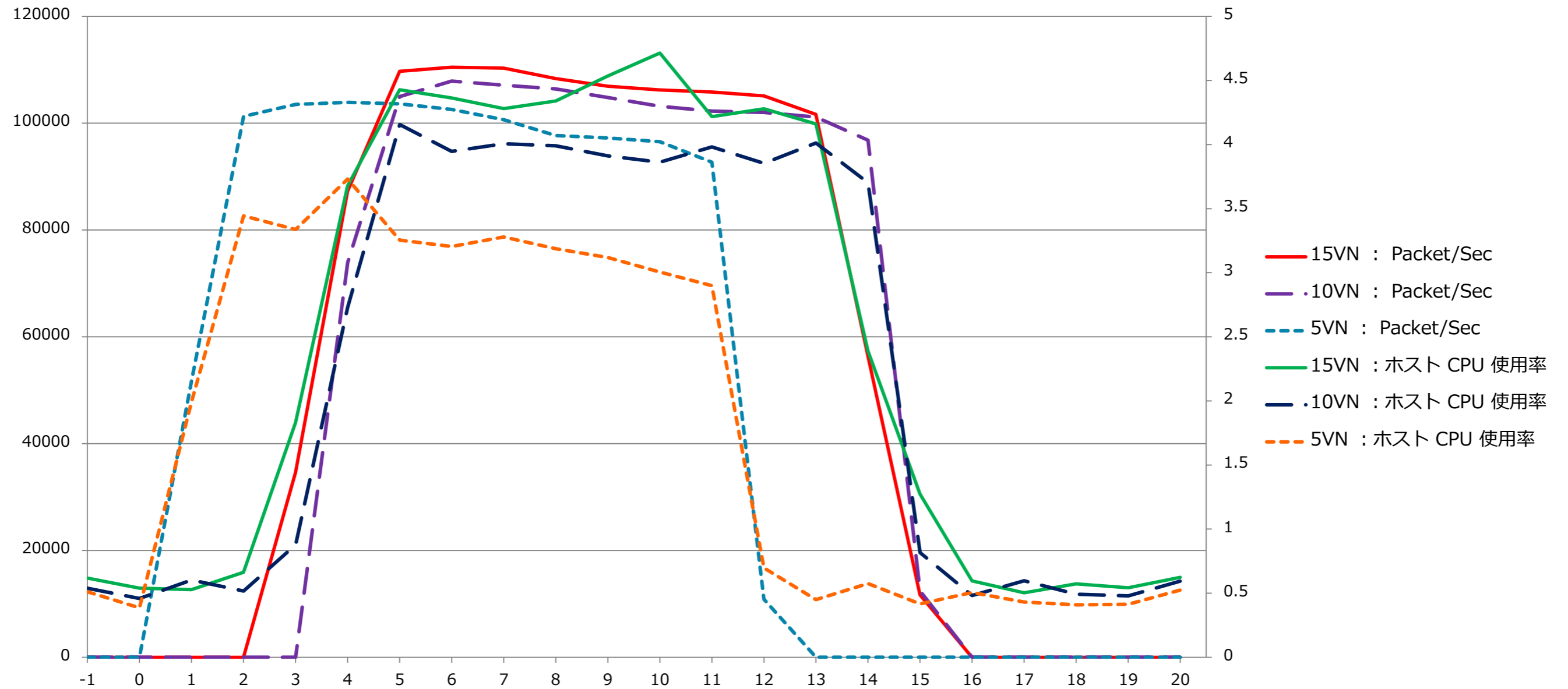
- バーチャルマシン

- vCPU : 2 vCPU
- メモリ : 4GB
- HDD : IDE 64GB × 1 (差分ディスク)

NVGRE ホスト負荷試験：結果（１）



NVGRE ホスト負荷試験：結果（2）

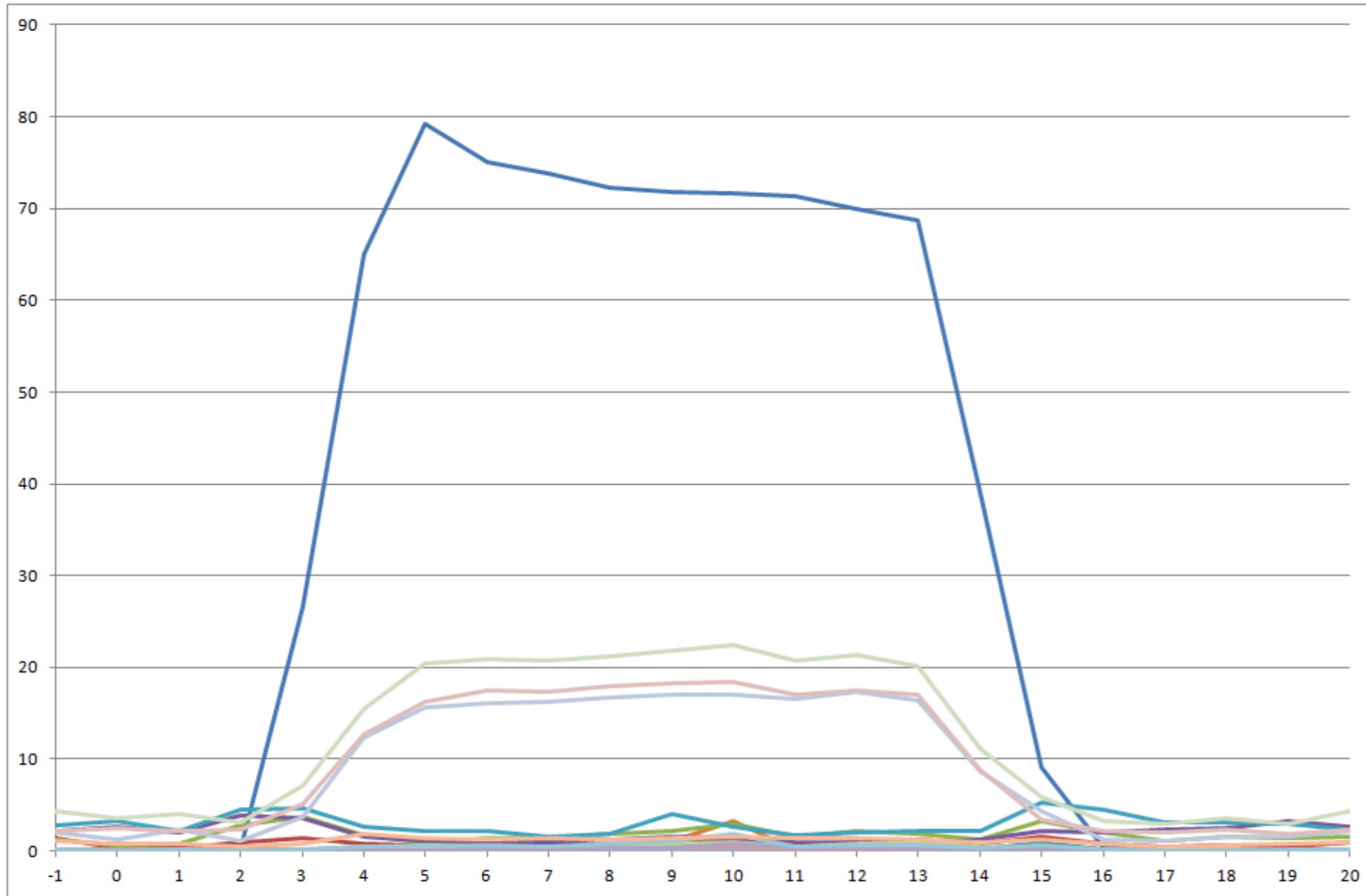


NVGRE ホスト負荷試験：考察

- 使用帯域を使いきっている状況下でも CPU 利用率が変化（グラフ 1）
- カプセル化を行っているパケット数と同調して変化（グラフ 2）
- 従って、トラフィックを発生させているバーチャルマシンの台数には依存せず、 NVGRE でカプセル化を行うパケット数（Packets/Sec）によってホスト側の CPU 負荷が変化すると考えられる
- CPU 負荷は最大時で 4.71%

⇒ CPU 負荷の実態はどうなっている？ 綺麗に並列処理されている？

NVGRE ホスト負荷試験：追加確認

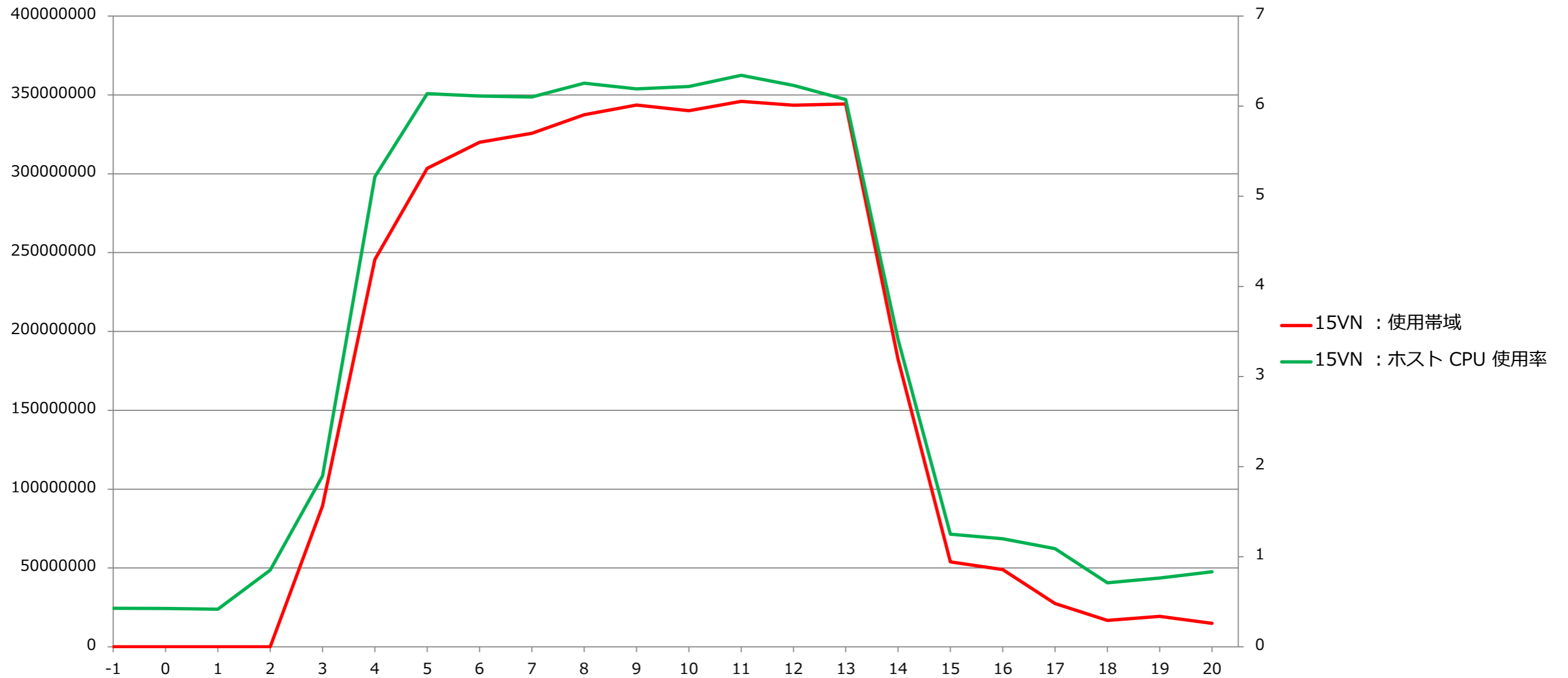


- 15VN テスト時に 全 VP (32 VP) の使用率を取得
- 32 VP 中、 1VP の負荷上昇を確認
- パケット処理数の負荷傾向と同じ傾向を示している

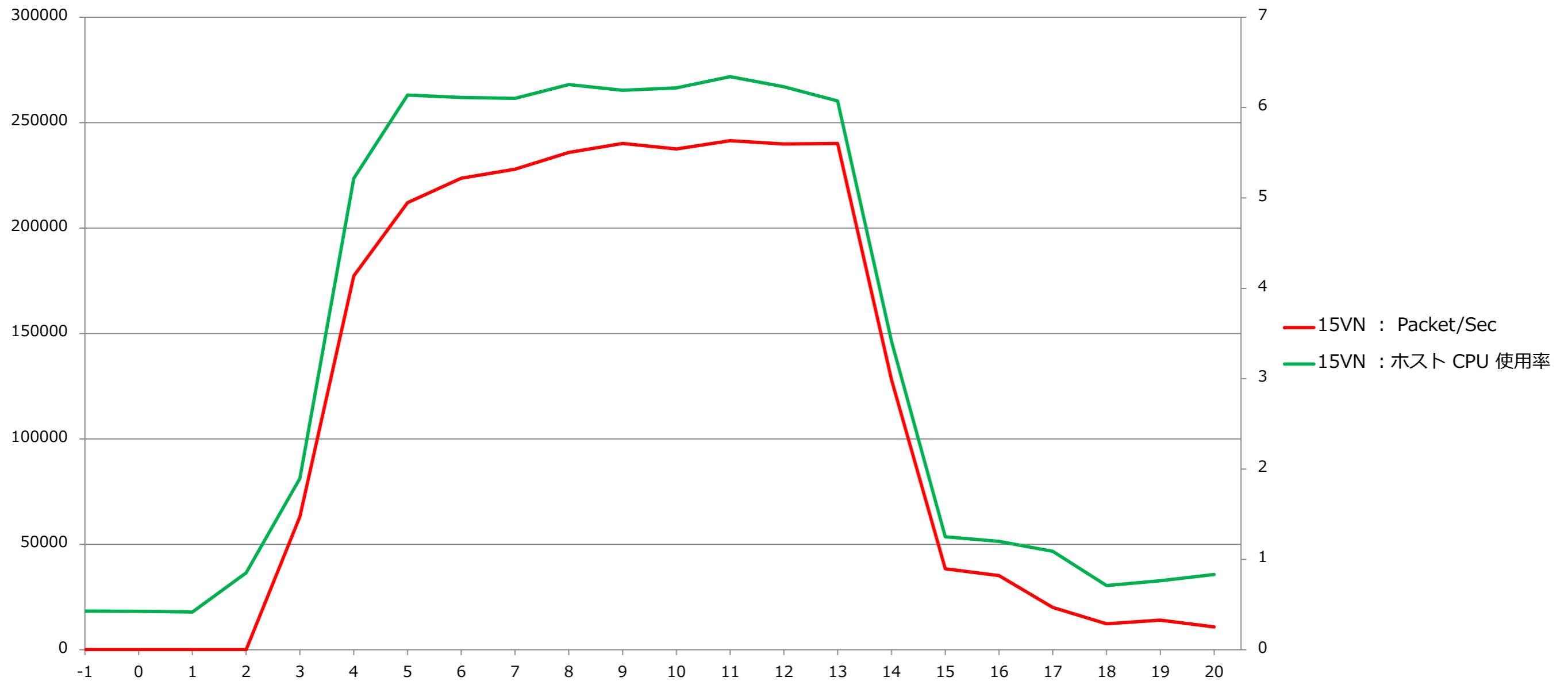
NVGRE ホスト負荷試験： 10Gb 試験

- 2 台の物理ホスト上に配置された、計 15 × 2 台のバーチャルマシンを使用
- バーチャルマシンは 2 台ずつ異なる VSID に所属（計 15 Virtual Network を使用）
- Disk I/O を伴わない、ネットワーク帯域計測ツールである『iperf.exe』を、ネットワーク負荷として使用
- 物理ホストを跨ぐように組み合わせた、15 組のバーチャルマシン間で iperf.exe による負荷を発生させ、その際のホスト側 CPU 利用率をパフォーマンスモニターにて計測
- 物理ホスト間のネットワーク帯域は **10Gbps** （ **10 Gigabit Ethernet × 1** ）
- 全 VP のCPU 使用率も同時に取得し、特定 VP の処理集中傾向がみられるかを確認する

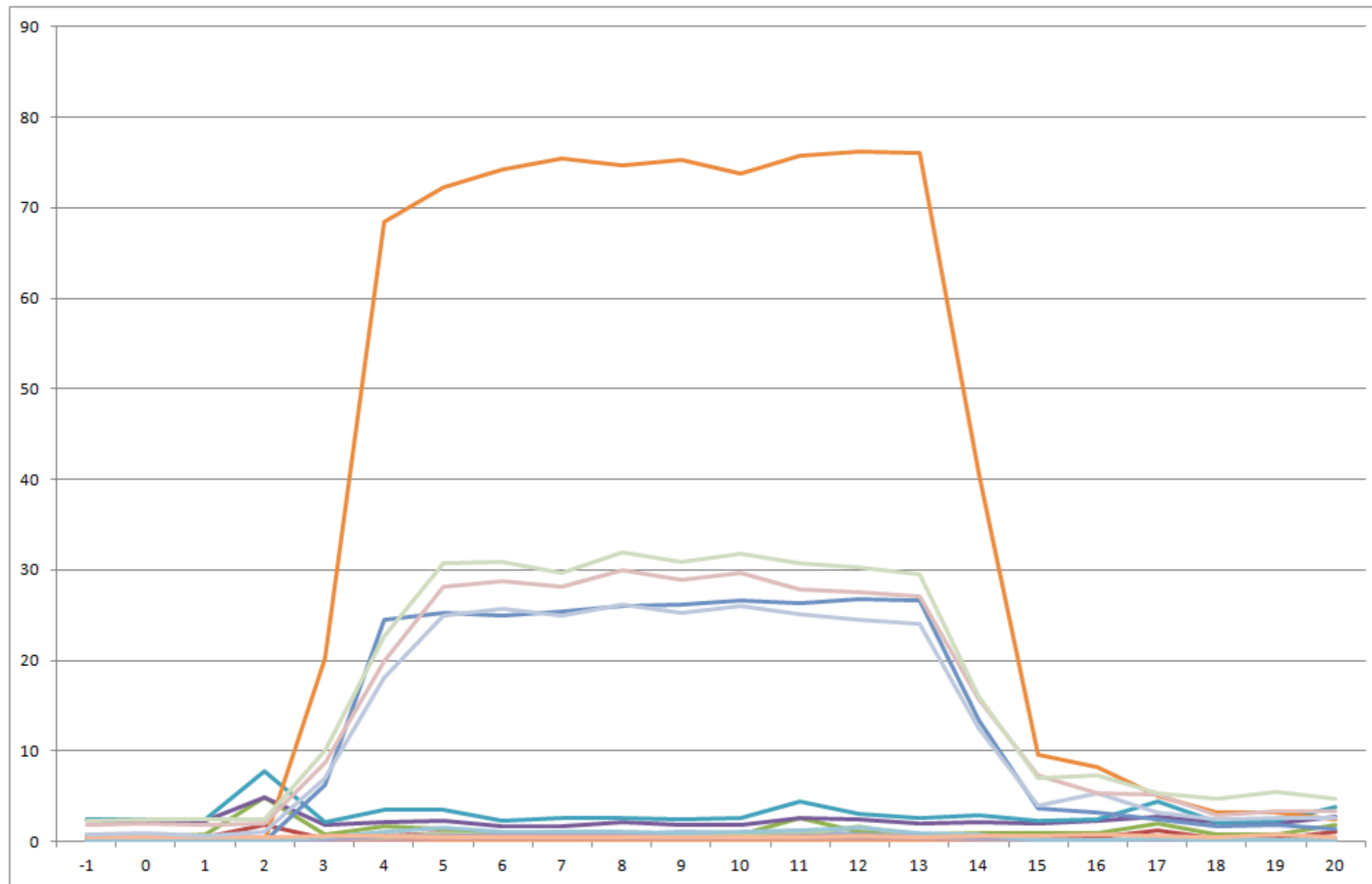
NVGRE ホスト負荷試験： 10Gb 試験結果（ 1 ）



NVGRE ホスト負荷試験： 10Gb 試験結果（ 2 ）



NVGRE ホスト負荷試験： 10Gb 試験結果（ 3 ）



NVGRE ホスト負荷試験： 10Gb 試験考察

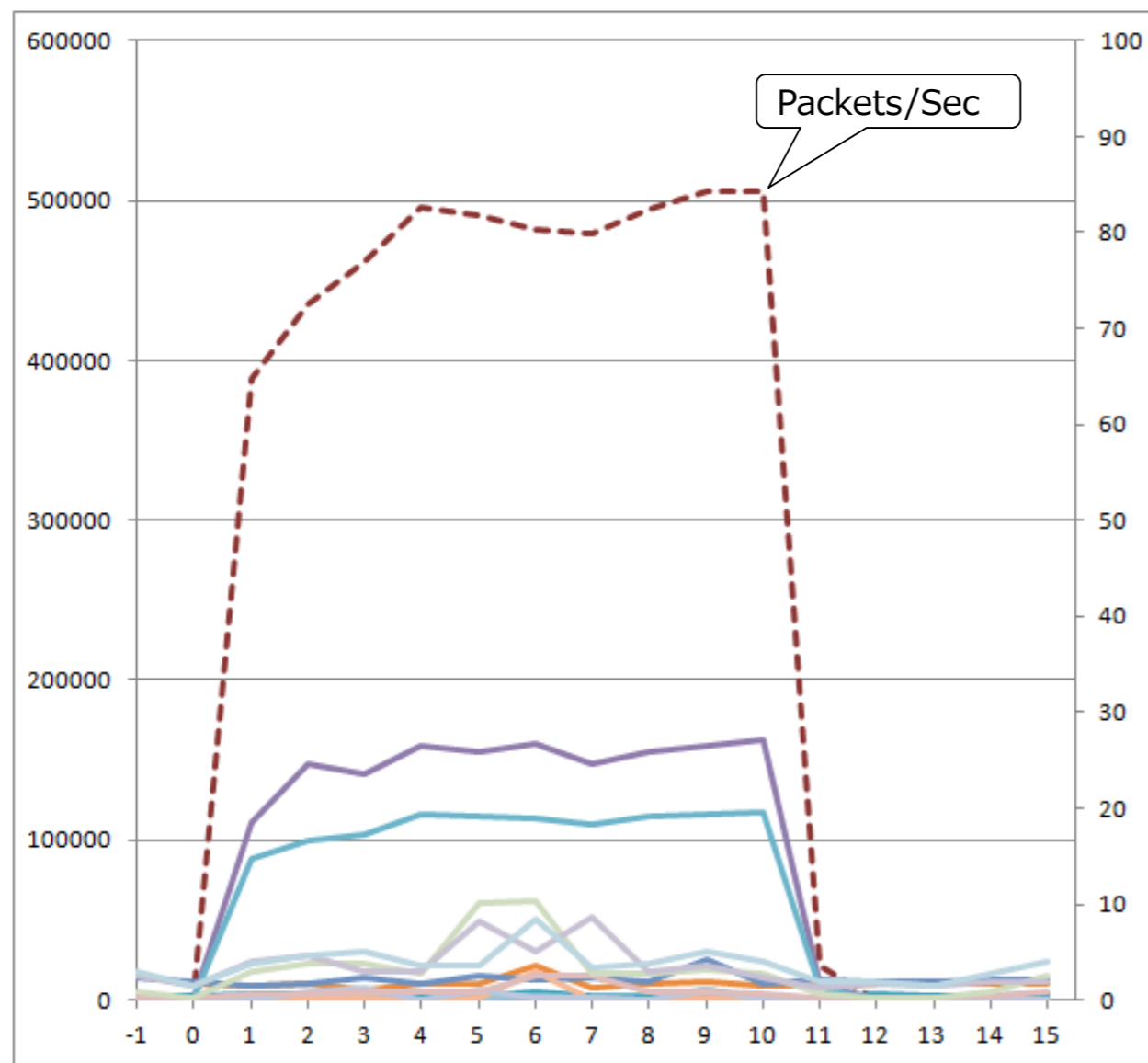
- 使用帯域は 2.58 Gbps 程度であり、帯域には余裕がある状況。
- PPS の観点からもネットワーク負荷は十分に余裕があり、 10G 環境下ではネットワーク性能を使いきれていない状況を確認。
- CPU 負荷（全体）では最大時で 6.3% 程度
- Root VP の状況を確認すると、特定の VP に処理が集中している状況を確認（最大時： 76.2% ）

⇒ この負荷は、本当に全部 NVGRE の処理負荷？

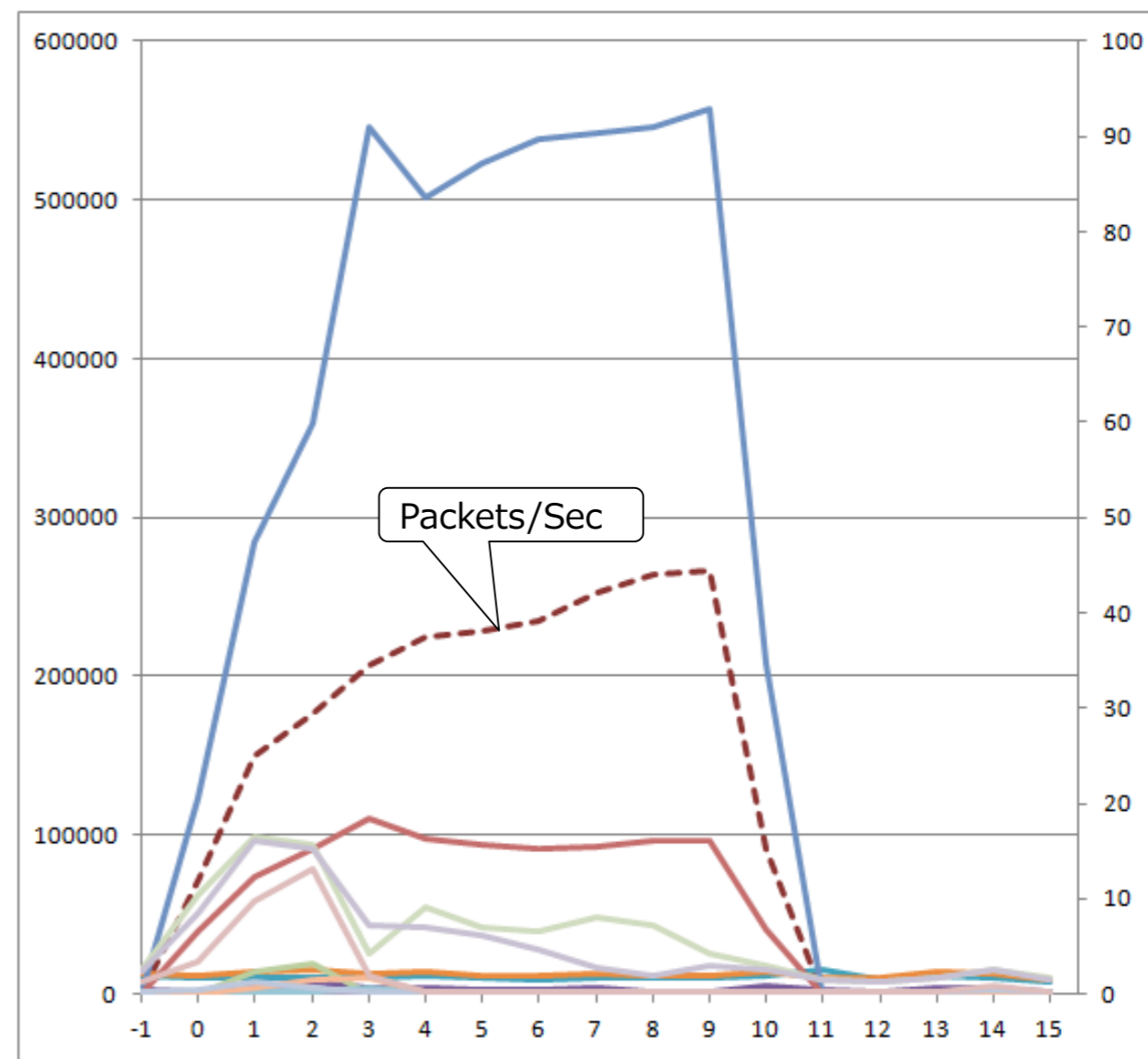
比較試験： non-NVGRE vs NVGRE

- 2 台の物理ホスト上に配置された、2 台のバーチャルマシンを使用
- Disk I/O を伴わない、ネットワーク帯域計測ツールである『iperf.exe』を、ネットワーク負荷として使用
- NVGRE を設定した場合、設定しない場合でそれぞれパフォーマンスを測定
- 接続する仮想スイッチは、両パターンともに『標準スイッチ』
- 物理ホスト間のネットワーク帯域は **10Gbps** （ **10 Gigabit Ethernet × 1** ）
- 全 VP のCPU 使用率も同時に取得し、特定 VP の処理集中傾向がみられるかを確認する

比較試験： non-NVGRE vs NVGRE 結果



non-NVGRE 環境
(Packets/Sec / Root-VP CPU 使用率)



NVGRE 環境
(Packets/Sec / Root-VP CPU 使用率)

比較試験： non-NVGRE vs NVGRE ： 考察

- NVGRE を使用しない環境では、 5.5Gbps 程度のスループットに対して、 NVGRE 環境では 2.8Gbps 程度のスループット。
- NVGRE を使用しない環境と比較して、 1 Root VP （ Root VP 15 ） に対して負荷が発生していることが確認できた。
- この負荷がボトルネックになって、スループットが上がらない？

⇒ この負荷を H/W オフロードできれば.....



ConnectX-3
PRO

Benefits	Key Features
	<ul style="list-style-type: none">• 1us MPI ping latency• Up to 40/56GbE per port• Single- and Dual-Port options available• PCI Express 3.0 (up to 8GT/s)• CPU offload of transport operations• Application offload• Precision Clock Synchronization• HW Offloads for NVGRE and VXLAN encapsulated traffic• End-to-end QoS and congestion control• Hardware-based I/O virtualization• RoHS-R6

まとめ

まとめ

- Network Virtualization は非常に便利な機能です
- Private Cloud 等、 multi-tenant を意識した設計をする場合には、お勧め機能の一つです
→ 事業部単位や子会社単位で基盤を提供し、論理的には異なる Network としたい、等々
- SC2012 VMM SP1 と組み合わせることにより、Windows Network Virtualization の能力を最大限引き出すことが可能です。
- GUI で実現できない場合は、 PowerShell で。
- 欠けたピースも揃いつつあり、目の前には vNext が見え始めています。
- 是非現バージョンで検証を行って、 vNext に備えてください。

Q & A

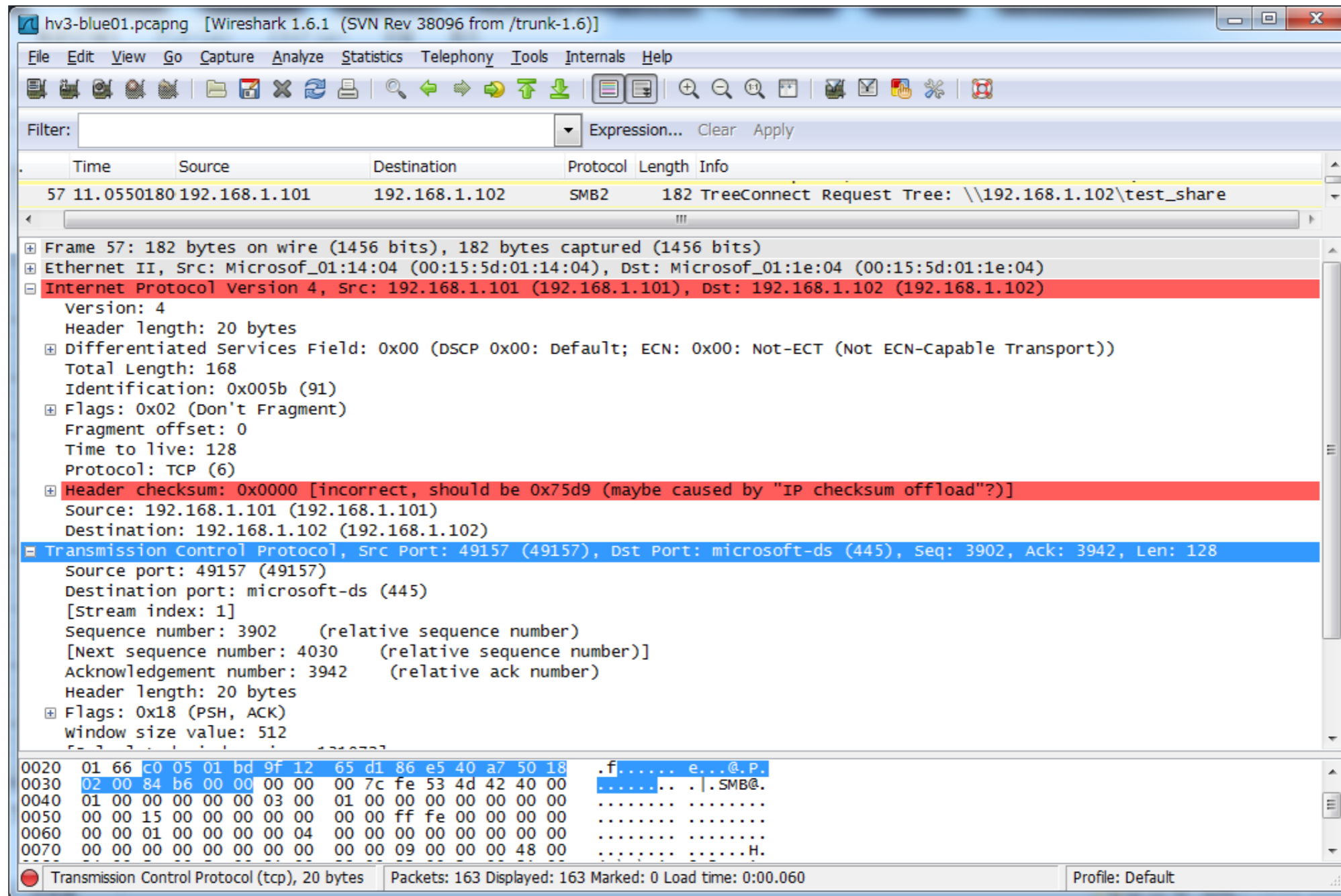


Appendix A : IP Rewrite とは ? (軽く)

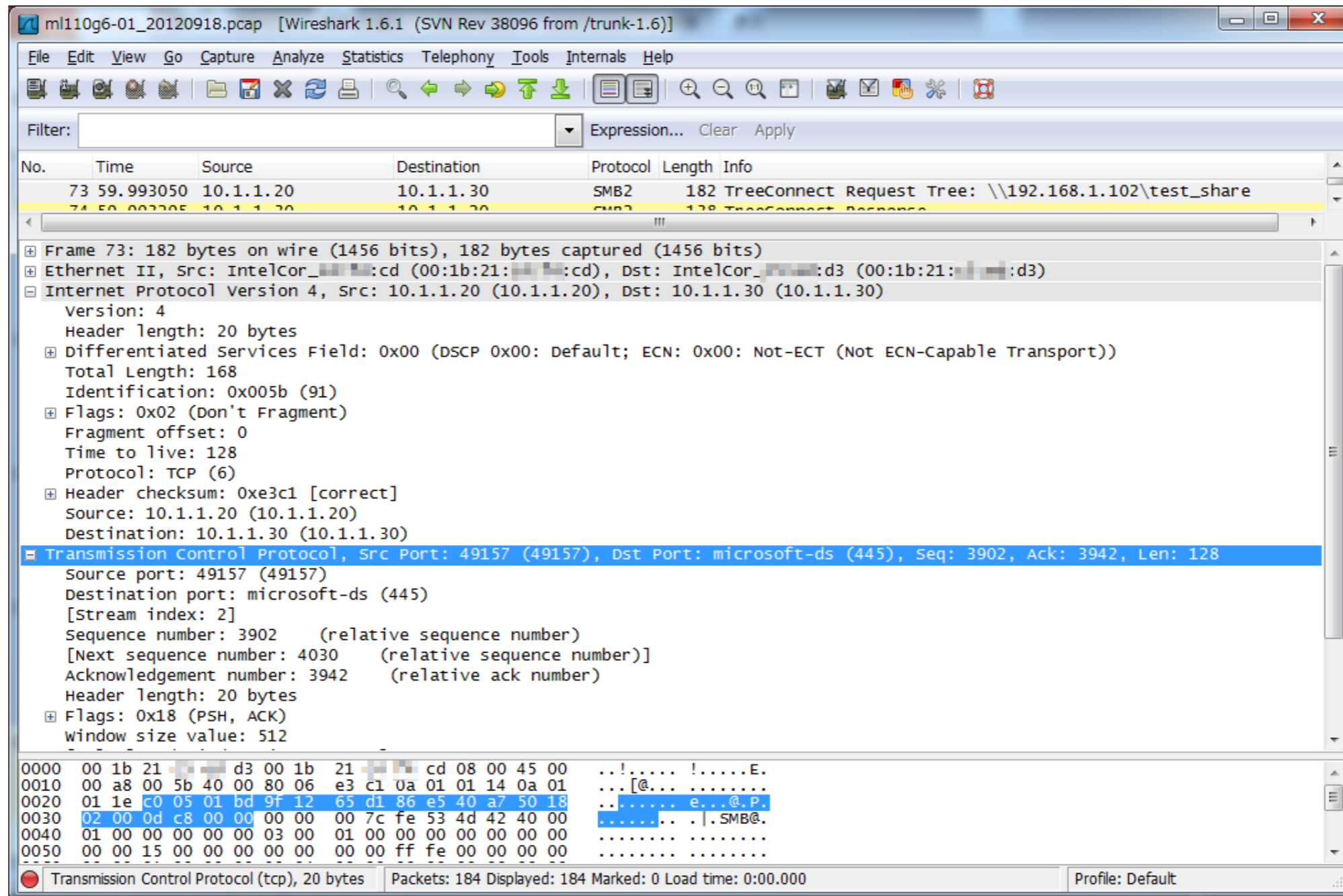
IP Rewrite のポイント

- データセンター内 IP Address と仮想マシン IP Address の 1 対 1 NAT
 - ペイロード含め、一切の変更を行わずに、MAC Address / IP Address を書き換え
 - カプセル化を行わない為、パケットオーバーヘッドは一切なし
 - TCP オフロード等の H/W 支援機能がフル活用可能
- Network 経路上での等コストマルチパス（ ECMP ）バランシングも、ネットワーク機器の設定を変更する事なく動作可能
- アクセススイッチ（仮想化モジュール）で NAT 処理を行う為、仮想マシンは仮想ネットワークを全く意識しない

IP Rewrite パケットキャプチャ : Guest OS



IP Rewrite パケットキャプチャ : Network

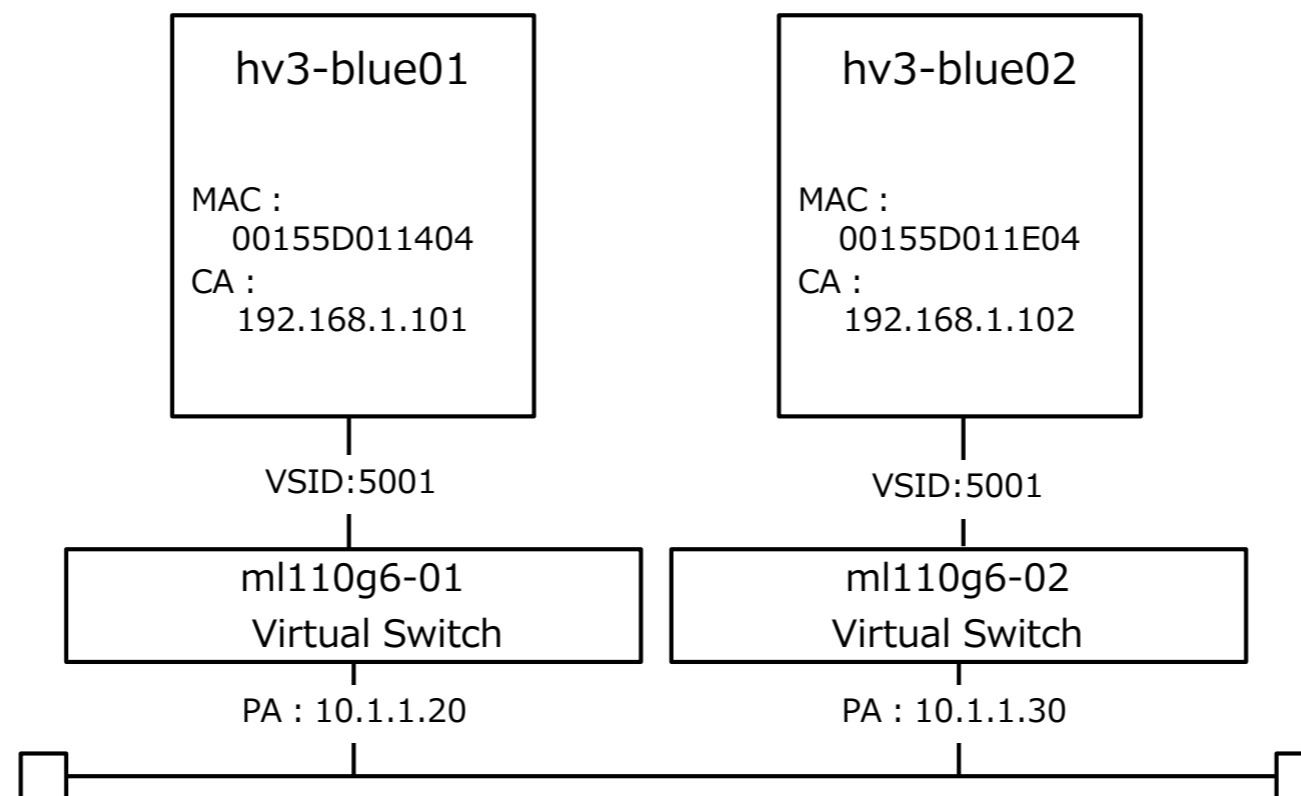


Appendix B : Network Virtualization の PowerShell での実装例

実装例（1）基本形

2 台の物理ホスト上に配置された、2 台の仮想マシンで Network Virtualization を実装。
トンネル方式は NVGRE 。

実装例（1）基本形・構成図



実装例 (1) 基本形・PowerShell

```
New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.101" -ProviderAddress "10.1.1.20" -MACAddress "00155D011404" -Rule "TranslationMethodEncap" -CimSession "ml110g6-01"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.102" -ProviderAddress "10.1.1.30" -MACAddress "00155D011E04" -Rule "TranslationMethodEncap" -CimSession "ml110g6-01"

New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.101" -ProviderAddress "10.1.1.20" -MACAddress "00155D011404" -Rule "TranslationMethodEncap" -CimSession "ml110g6-02"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.102" -ProviderAddress "10.1.1.30" -MACAddress "00155D011E04" -Rule "TranslationMethodEncap" -CimSession "ml110g6-02"

New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}" -VirtualSubnetID "5001" -DestinationPrefix "192.168.1.0/24" -NextHop "0.0.0.0" -Metric 255 -CimSession "ml110g6-01"

New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}" -VirtualSubnetID "5001" -DestinationPrefix "192.168.1.0/24" -NextHop "0.0.0.0" -Metric 255 -CimSession "ml110g6-02"

$cred = Get-Credential "dob1¥administrator"
$WNVNIC = "WNVNIC"

$iface = Get-NetAdapter $WNVNIC -CimSession "ml110g6-01"
New-NetVirtualizationProviderAddress -InterfaceIndex $iface.InterfaceIndex -ProviderAddress "10.1.1.20" -PrefixLength 24 -CimSession "ml110g6-01"

Invoke-Command -ComputerName "ml110g6-01" -Credential $cred {
Get-VMNetworkAdapter "hv3-blue01" | where {$_.MacAddress -eq "00155D011404"} | Set-VMNetworkAdapter -VirtualSubnetID 5001;
}

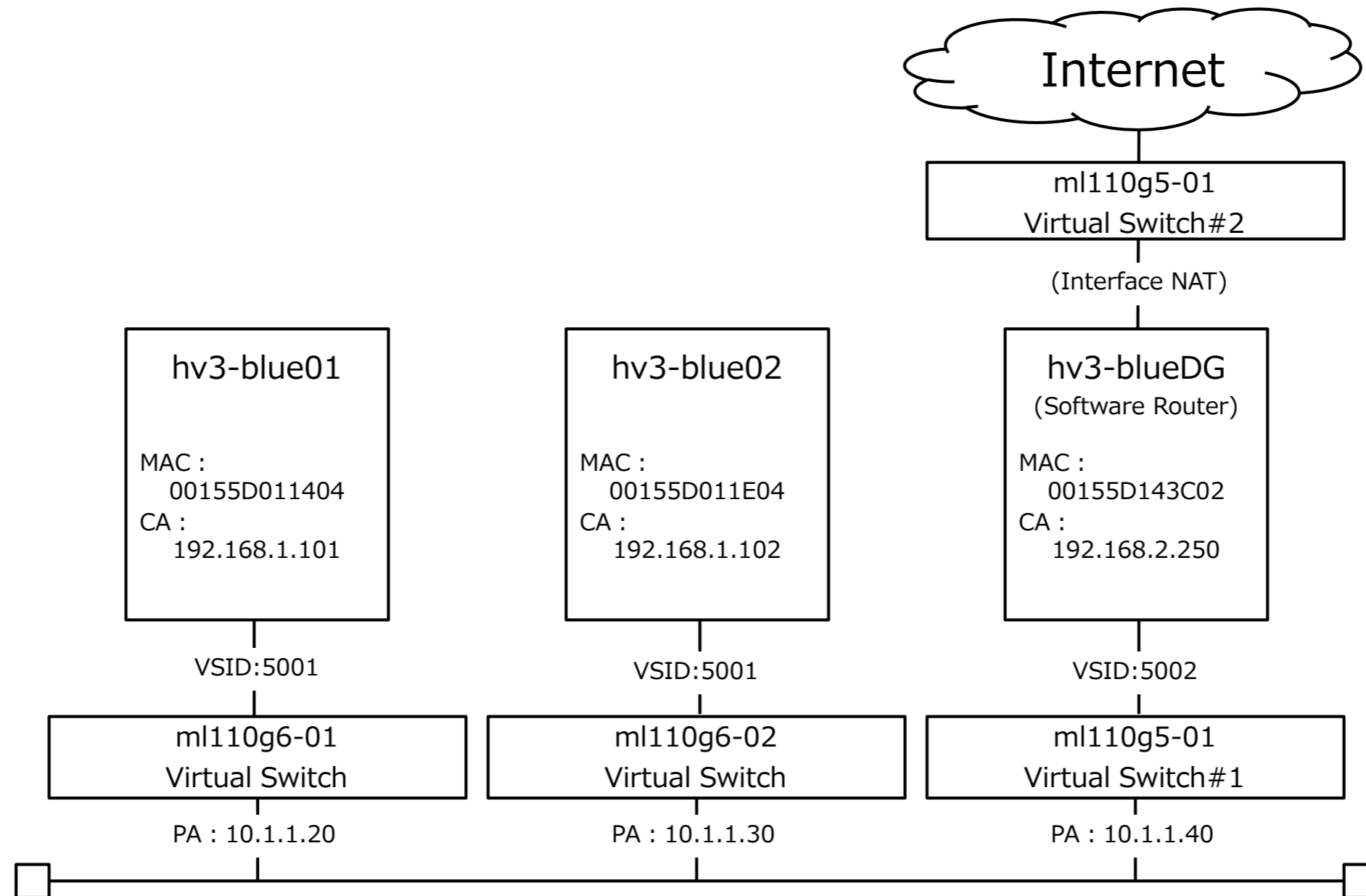
$iface = Get-NetAdapter $WNVNIC -CimSession "ml110g6-02"
New-NetVirtualizationProviderAddress -InterfaceIndex $iface.InterfaceIndex -ProviderAddress "10.1.1.30" -PrefixLength 24 -CimSession "ml110g6-02"

Invoke-Command -ComputerName "ml110g6-12" -Credential $cred {
Get-VMNetworkAdapter "hv3-blue02" | where {$_.MacAddress -eq "00155D011E04"} | Set-VMNetworkAdapter -VirtualSubnetID 5001;
}
```

実装例（2）応用形

3 台の物理ホスト上に配置された、3 台の仮想マシンで Network Virtualization を実装。
仮想マシン 2 台は Windows Server 、もう 1 台は Software Router。
Software Router 経由で Internet と通信可能。
Software Router は異なるセグメント（異なる VSID ）に設定、 VSID 間で Routing を実施。
トンネル方式は NVGRE 。

実装例（2）応用形・構成図



実装例 (2) 応用形・PowerShell (1)

```
New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.101" -ProviderAddress "10.1.1.20" -MACAddress "00155D011404" -Rule "TranslationMethodEncap" -VMName "hv3-blue01" -CimSession "ml110g6-01"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.102" -ProviderAddress "10.1.1.30" -MACAddress "00155D011E04" -Rule "TranslationMethodEncap" -VMName "hv3-blue02" -CimSession "ml110g6-01"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.1" -ProviderAddress "169.254.254.254" -MACAddress "101010101001" -Rule "TranslationMethodEncap" -VMName "hv3-blue-GW" -CimSession "ml110g6-01"

New-NetVirtualizationLookupRecord -VirtualSubnetID "5002" -CustomerAddress "192.168.2.250" -ProviderAddress "10.1.1.40" -MACAddress "00155D143C02" -Rule "TranslationMethodEncap" -VMName "hv3-blueDG" -CimSession "ml110g6-01"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5002" -CustomerAddress "0.0.0.0" -ProviderAddress "10.1.1.40" -MACAddress "00155D143C02" -Rule "TranslationMethodEncap" -VMName "BlueWildcard" -CimSession "ml110g6-01"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5002" -CustomerAddress "192.168.2.1" -ProviderAddress "169.254.254.254" -MACAddress "101010101011" -Rule "TranslationMethodEncap" -VMName "hv3-blueDGW" -CimSession "ml110g6-01"

New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.101" -ProviderAddress "10.1.1.20" -MACAddress "00155D011404" -Rule "TranslationMethodEncap" -VMName "hv3-blue01" -CimSession "ml110g6-02"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.102" -ProviderAddress "10.1.1.30" -MACAddress "00155D011E04" -Rule "TranslationMethodEncap" -VMName "hv3-blue02" -CimSession "ml110g6-02"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.1" -ProviderAddress "169.254.254.254" -MACAddress "101010101001" -Rule "TranslationMethodEncap" -VMName "hv3-blue-GW" -CimSession "ml110g6-02"

New-NetVirtualizationLookupRecord -VirtualSubnetID "5002" -CustomerAddress "192.168.2.250" -ProviderAddress "10.1.1.40" -MACAddress "00155D143C02" -Rule "TranslationMethodEncap" -VMName "hv3-blueDG" -CimSession "ml110g6-02"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5002" -CustomerAddress "0.0.0.0" -ProviderAddress "10.1.1.40" -MACAddress "00155D143C02" -Rule "TranslationMethodEncap" -VMName "BlueWildcard" -CimSession "ml110g6-02"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5002" -CustomerAddress "192.168.2.1" -ProviderAddress "169.254.254.254" -MACAddress "101010101011" -Rule "TranslationMethodEncap" -VMName "hv3-blueDGW" -CimSession "ml110g6-02"

New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.101" -ProviderAddress "10.1.1.20" -MACAddress "00155D011404" -Rule "TranslationMethodEncap" -VMName "hv3-blue01" -CimSession "ml110g5-01"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.102" -ProviderAddress "10.1.1.30" -MACAddress "00155D011E04" -Rule "TranslationMethodEncap" -VMName "hv3-blue02" -CimSession "ml110g5-01"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.1" -ProviderAddress "169.254.254.254" -MACAddress "101010101001" -Rule "TranslationMethodEncap" -VMName "hv3-blue-GW" -CimSession "ml110g5-01"

New-NetVirtualizationLookupRecord -VirtualSubnetID "5002" -CustomerAddress "192.168.2.250" -ProviderAddress "10.1.1.40" -MACAddress "00155D143C02" -Rule "TranslationMethodEncap" -VMName "hv3-blueDG" -CimSession "ml110g5-01"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5002" -CustomerAddress "0.0.0.0" -ProviderAddress "10.1.1.40" -MACAddress "00155D143C02" -Rule "TranslationMethodEncap" -VMName "BlueWildcard" -CimSession "ml110g5-01"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5002" -CustomerAddress "192.168.2.1" -ProviderAddress "169.254.254.254" -MACAddress "101010101011" -Rule "TranslationMethodEncap" -VMName "hv3-blueDGW" -CimSession "ml110g5-01"
```

実装例（2）応用形・PowerShell（2）

```
New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}" -VirtualSubnetID "5001" -DestinationPrefix "192.168.1.0/24" -NextHop "0.0.0.0" -Metric 255 -CimSession "ml110g6-01"
New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}" -VirtualSubnetID "5002" -DestinationPrefix "192.168.2.0/24" -NextHop "0.0.0.0" -Metric 255 -CimSession "ml110g6-01"
New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}" -VirtualSubnetID "5002" -DestinationPrefix "0.0.0.0/0" -NextHop "192.168.2.250" -Metric 255 -CimSession "ml110g6-01"

New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}" -VirtualSubnetID "5001" -DestinationPrefix "192.168.1.0/24" -NextHop "0.0.0.0" -Metric 255 -CimSession "ml110g6-02"
New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}" -VirtualSubnetID "5002" -DestinationPrefix "192.168.2.0/24" -NextHop "0.0.0.0" -Metric 255 -CimSession "ml110g6-02"
New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}" -VirtualSubnetID "5002" -DestinationPrefix "0.0.0.0/0" -NextHop "192.168.2.250" -Metric 255 -CimSession "ml110g6-02"

New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}" -VirtualSubnetID "5001" -DestinationPrefix "192.168.1.0/24" -NextHop "0.0.0.0" -Metric 255 -CimSession "ml110g5-01"
New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}" -VirtualSubnetID "5002" -DestinationPrefix "192.168.2.0/24" -NextHop "0.0.0.0" -Metric 255 -CimSession "ml110g5-01"
New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}" -VirtualSubnetID "5002" -DestinationPrefix "0.0.0.0/0" -NextHop "192.168.2.250" -Metric 255 -CimSession "ml110g5-01"

$cred = Get-Credential "dob1¥administrator"
$WNVNIC = "WNVNIC"

$iface = Get-NetAdapter $WNVNIC -CimSession "ml110g6-01"
New-NetVirtualizationProviderAddress -InterfaceIndex $iface.InterfaceIndex -ProviderAddress "10.1.1.20" -PrefixLength 24 -CimSession "ml110g6-01"
Invoke-Command -ComputerName "ml110g6-01" -Credential $cred {
    Get-VMNetworkAdapter "hv3-blue01" | where {$_.MacAddress -eq "00155D011404"} | Set-VMNetworkAdapter -VirtualSubnetID 5001;
}

$iface = Get-NetAdapter $WNVNIC -CimSession "ml110g6-02"
New-NetVirtualizationProviderAddress -InterfaceIndex $iface.InterfaceIndex -ProviderAddress "10.1.1.30" -PrefixLength 24 -CimSession "ml110g6-02"
Invoke-Command -ComputerName "ml110g6-02" -Credential $cred {
    Get-VMNetworkAdapter "hv3-blue02" | where {$_.MacAddress -eq "00155D011E04"} | Set-VMNetworkAdapter -VirtualSubnetID 5001;
}

$iface = Get-NetAdapter $WNVNIC -CimSession "ml110g5-01"
New-NetVirtualizationProviderAddress -InterfaceIndex $iface.InterfaceIndex -ProviderAddress "10.1.1.40" -PrefixLength 24 -CimSession "ml110g5-01"
Invoke-Command -ComputerName "ml110g5-01" -Credential $cred {
    Get-VMNetworkAdapter "hv3-blueDG" | where {$_.MacAddress -eq "00155D143C02"} | Set-VMNetworkAdapter -VirtualSubnetID 5002;
}
```

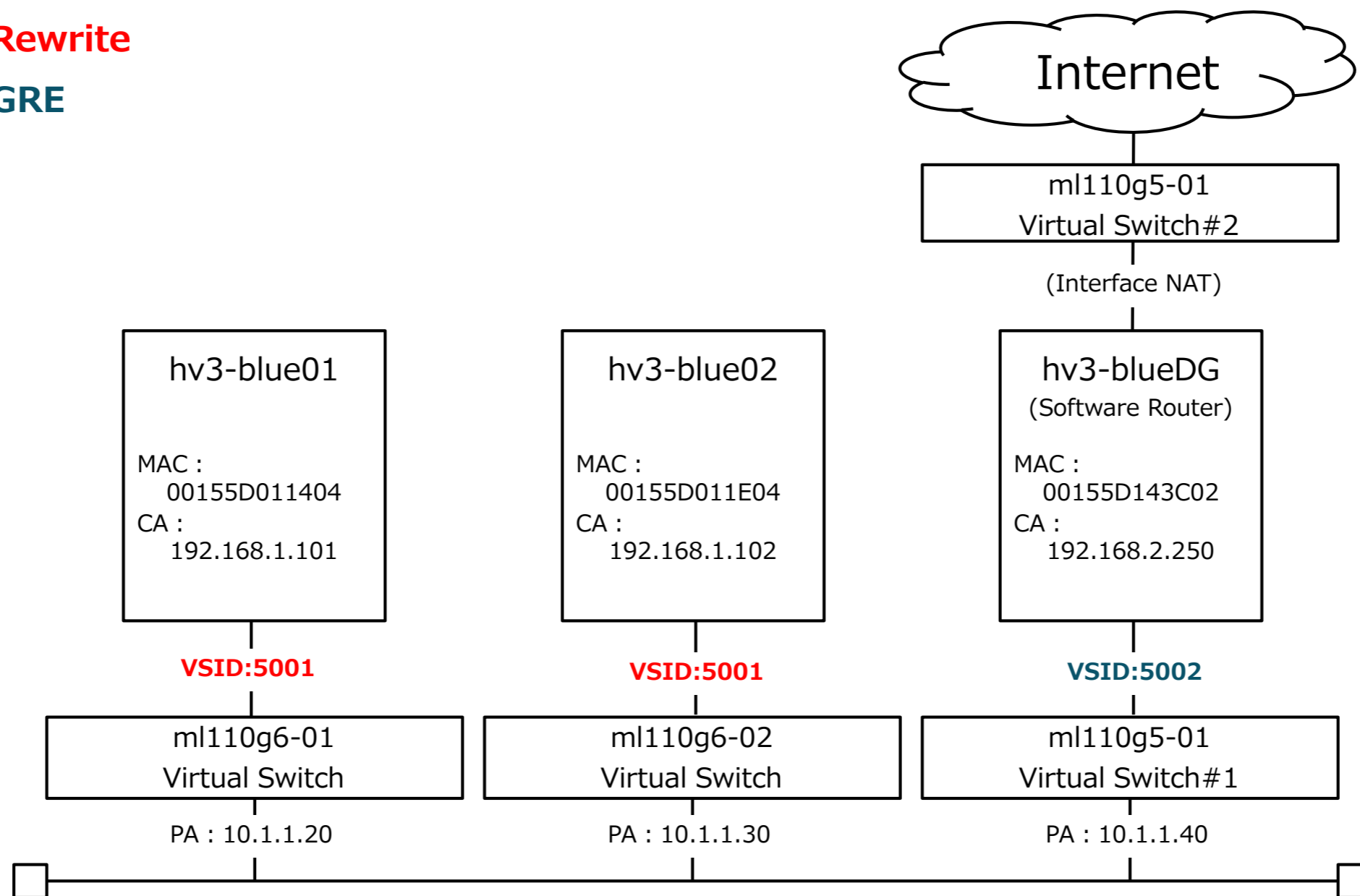
実装例（3）超応用形

3 台の物理ホスト上に配置された、3 台の仮想マシンで Network Virtualization を実装。
仮想マシン 2 台は Windows Server 、もう1台は Software Router 。
Software Router 経由で Internet と通信可能。
Software Router は異なるセグメント（異なる VSID ）に設定、 VSID 間で Routing を実施。
Windows Server 間のトンネル方式は IP Rewrite 。
Windows Server と Software Router 間のトンネル方式は NVGRE 。

実装例 (3) 超応用形・構成図

VSID:5001 → IP Rewrite

VSID:5002 → NVGRE



実装例 (3) 超応用形・PowerShell (1)

```
New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.101" -ProviderAddress "10.1.1.20" -MACAddress "00155D011404" -Rule "TranslationMethodNAT" -VMName "hv3-blue01" -CimSession "ml110g6-01"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.102" -ProviderAddress "10.1.1.30" -MACAddress "00155D011E04" -Rule "TranslationMethodNAT" -VMName "hv3-blue02" -CimSession "ml110g6-01"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.1" -ProviderAddress "169.254.254.254" -MACAddress "101010101001" -Rule "TranslationMethodNAT" -VMName "hv3-blue-GW" -CimSession "ml110g6-01"

New-NetVirtualizationLookupRecord -VirtualSubnetID "5002" -CustomerAddress "192.168.2.250" -ProviderAddress "10.1.1.40" -MACAddress "00155D143C02" -Rule "TranslationMethodEncap" -VMName "hv3-blueDG" -CimSession "ml110g6-01"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5002" -CustomerAddress "0.0.0.0" -ProviderAddress "10.1.1.40" -MACAddress "00155D143C02" -Rule "TranslationMethodEncap" -VMName "BlueWildcard" -CimSession "ml110g6-01"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5002" -CustomerAddress "192.168.2.1" -ProviderAddress "169.254.254.253" -MACAddress "101010101011" -Rule "TranslationMethodEncap" -VMName "hv3-blueDGW" -CimSession "ml110g6-01"

New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.101" -ProviderAddress "10.1.1.20" -MACAddress "00155D011404" -Rule "TranslationMethodNAT" -VMName "hv3-blue01" -CimSession "ml110g6-02"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.102" -ProviderAddress "10.1.1.30" -MACAddress "00155D011E04" -Rule "TranslationMethodNAT" -VMName "hv3-blue02" -CimSession "ml110g6-02"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.1" -ProviderAddress "169.254.254.254" -MACAddress "101010101001" -Rule "TranslationMethodNAT" -VMName "hv3-blue-GW" -CimSession "ml110g6-02"

New-NetVirtualizationLookupRecord -VirtualSubnetID "5002" -CustomerAddress "192.168.2.250" -ProviderAddress "10.1.1.40" -MACAddress "00155D143C02" -Rule "TranslationMethodEncap" -VMName "hv3-blueDG" -CimSession "ml110g6-02"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5002" -CustomerAddress "0.0.0.0" -ProviderAddress "10.1.1.40" -MACAddress "00155D143C02" -Rule "TranslationMethodEncap" -VMName "BlueWildcard" -CimSession "ml110g6-02"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5002" -CustomerAddress "192.168.2.1" -ProviderAddress "169.254.254.253" -MACAddress "101010101011" -Rule "TranslationMethodEncap" -VMName "hv3-blueDGW" -CimSession "ml110g6-02"

New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.101" -ProviderAddress "10.1.1.20" -MACAddress "00155D011404" -Rule "TranslationMethodNAT" -VMName "hv3-blue01" -CimSession "ml110g5-01"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.102" -ProviderAddress "10.1.1.30" -MACAddress "00155D011E04" -Rule "TranslationMethodNAT" -VMName "hv3-blue02" -CimSession "ml110g5-01"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "192.168.1.1" -ProviderAddress "169.254.254.254" -MACAddress "101010101001" -Rule "TranslationMethodNAT" -VMName "hv3-blue-GW" -CimSession "ml110g5-01"

New-NetVirtualizationLookupRecord -VirtualSubnetID "5002" -CustomerAddress "192.168.2.250" -ProviderAddress "10.1.1.40" -MACAddress "00155D143C02" -Rule "TranslationMethodEncap" -VMName "hv3-blueDG" -CimSession "ml110g5-01"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5002" -CustomerAddress "0.0.0.0" -ProviderAddress "10.1.1.40" -MACAddress "00155D143C02" -Rule "TranslationMethodEncap" -VMName "BlueWildcard" -CimSession "ml110g5-01"
New-NetVirtualizationLookupRecord -VirtualSubnetID "5002" -CustomerAddress "192.168.2.1" -ProviderAddress "169.254.254.253" -MACAddress "101010101011" -Rule "TranslationMethodEncap" -VMName "hv3-blueDGW" -CimSession "ml110g5-01"
```

実装例 (3) 超応用形・PowerShell (2)

```
New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}" -VirtualSubnetID "5001" -DestinationPrefix "192.168.1.0/24" -NextHop "0.0.0.0" -Metric 255 -CimSession "ml110g6-01"
New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}" -VirtualSubnetID "5002" -DestinationPrefix "192.168.2.0/24" -NextHop "0.0.0.0" -Metric 255 -CimSession "ml110g6-01"
New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}" -VirtualSubnetID "5002" -DestinationPrefix "0.0.0.0/0" -NextHop "192.168.2.250" -Metric 255 -CimSession "ml110g6-01"

New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}" -VirtualSubnetID "5001" -DestinationPrefix "192.168.1.0/24" -NextHop "0.0.0.0" -Metric 255 -CimSession "ml110g6-02"
New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}" -VirtualSubnetID "5002" -DestinationPrefix "192.168.2.0/24" -NextHop "0.0.0.0" -Metric 255 -CimSession "ml110g6-02"
New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}" -VirtualSubnetID "5002" -DestinationPrefix "0.0.0.0/0" -NextHop "192.168.2.250" -Metric 255 -CimSession "ml110g6-02"

New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}" -VirtualSubnetID "5001" -DestinationPrefix "192.168.1.0/24" -NextHop "0.0.0.0" -Metric 255 -CimSession "ml110g5-01"
New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}" -VirtualSubnetID "5002" -DestinationPrefix "192.168.2.0/24" -NextHop "0.0.0.0" -Metric 255 -CimSession "ml110g5-01"
New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}" -VirtualSubnetID "5002" -DestinationPrefix "0.0.0.0/0" -NextHop "192.168.2.250" -Metric 255 -CimSession "ml110g5-01"

$cred = Get-Credential "dob1¥administrator"
$WNVNIC = "WNVNIC"

$iface = Get-NetAdapter $WNVNIC -CimSession "ml110g6-01"
New-NetVirtualizationProviderAddress -InterfaceIndex $iface.InterfaceIndex -ProviderAddress "10.1.1.20" -PrefixLength 24 -CimSession "ml110g6-01"
Invoke-Command -ComputerName "ml110g6-01" -Credential $cred {
Get-VMNetworkAdapter "hv3-blue01" | where {$_.MacAddress -eq "00155D011404"} | Set-VMNetworkAdapter -VirtualSubnetID 5001;
}

$iface = Get-NetAdapter $WNVNIC -CimSession "ml110g6-02"
New-NetVirtualizationProviderAddress -InterfaceIndex $iface.InterfaceIndex -ProviderAddress "10.1.1.30" -PrefixLength 24 -CimSession "ml110g6-02"
Invoke-Command -ComputerName "ml110g6-02" -Credential $cred {
Get-VMNetworkAdapter "hv3-blue02" | where {$_.MacAddress -eq "00155D011E04"} | Set-VMNetworkAdapter -VirtualSubnetID 5001;
}

$iface = Get-NetAdapter $WNVNIC -CimSession "ml110g5-01"
New-NetVirtualizationProviderAddress -InterfaceIndex $iface.InterfaceIndex -ProviderAddress "10.1.1.40" -PrefixLength 24 -CimSession "ml110g5-01"
Invoke-Command -ComputerName "ml110g5-01" -Credential $cred {
Get-VMNetworkAdapter "hv3-blueDG" | where {$_.MacAddress -eq "00155D143C02"} | Set-VMNetworkAdapter -VirtualSubnetID 5002;
}
```

Appendix C : サンプルスクリプト

静的 IP アドレス割り当てデモで使った PowerShell

```
# "" 内で静的 IP アドレスを割り当てるバーチャルマシン名を指定
$VM_Name = "VM Name"
# "" 内で割り当てる VM ネットワーク名を指定
$VMNetwork_Name = "VM Network Name"
# "" 内で割り当てる VM サブネット名を指定
$VMSubnet_Name = "VM Subnet Name"
# "" 内で割り当てる IP アドレスのプール名を指定
$IPPool_Name = " VM Subnet IP Pool"
# "" 内で割り当てる IP アドレスを指定
$VM_IPAddress = "xxx.xxx.xxx.xxx"
# "" 内で割り当て済み MAC アドレスを指定
$MACAddr = "xx:xx:xx:xx:xx:xx"
# "" 内で使用する仮想スイッチ名を指定
$vswitch_Name = "Vswitch Name"

$VM = Get-SCVirtualMachine -Name $VM_Name
$vNICsMAC = Get-SCVirtualNetworkAdapter -VM $VM
$vNICs = $VM.VirtualNetworkAdapters
$IPPool = Get-SCStaticIPAddressPool -Name $IPPool_Name
Grant-SCIPAddress -StaticIPAddressPool $IPPool -GrantToObjectType VirtualNetworkAdapter -GrantToObjectID $vNICs[0].ID -Description $VM.Name -IPAddress
$VM_IPAddress
$VirtualNetworkAdapter = Get-SCVirtualNetworkAdapter -Name $VM_Name -ID $vNICs.ID

$VMNetwork = Get-SCVMNetwork -Name $VMNetwork_Name
$VMSubnet = Get-SCVMSubnet -Name $VMSubnet_Name | where {$_.VMNetwork.ID -eq $VMNetwork.ID}

Set-SCVirtualNetworkAdapter -VirtualNetworkAdapter $VirtualNetworkAdapter -VMNetwork $VMNetwork -VMSubnet $VMSubnet -VirtualNetwork $vswitch_Name -
MACAddress $MACAddr -MACAddressType Static -IPv4Address $VM_IPAddress -IPv4AddressType Static -IPv6AddressType Dynamic -NoPortClassification -
EnableVMNetworkOptimization $false
```