



# Windows Server 2012 Network Virtualization Deep Dive!



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 )
- 三井情報株式会社で R&D 部門に所属しています。
- 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 )

# 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。
  - NVGRE や IP Rewrite の機能と実装方法が説明できる。
  - SC2012 VMM SP1 を利用する事のメリットが説明できる。

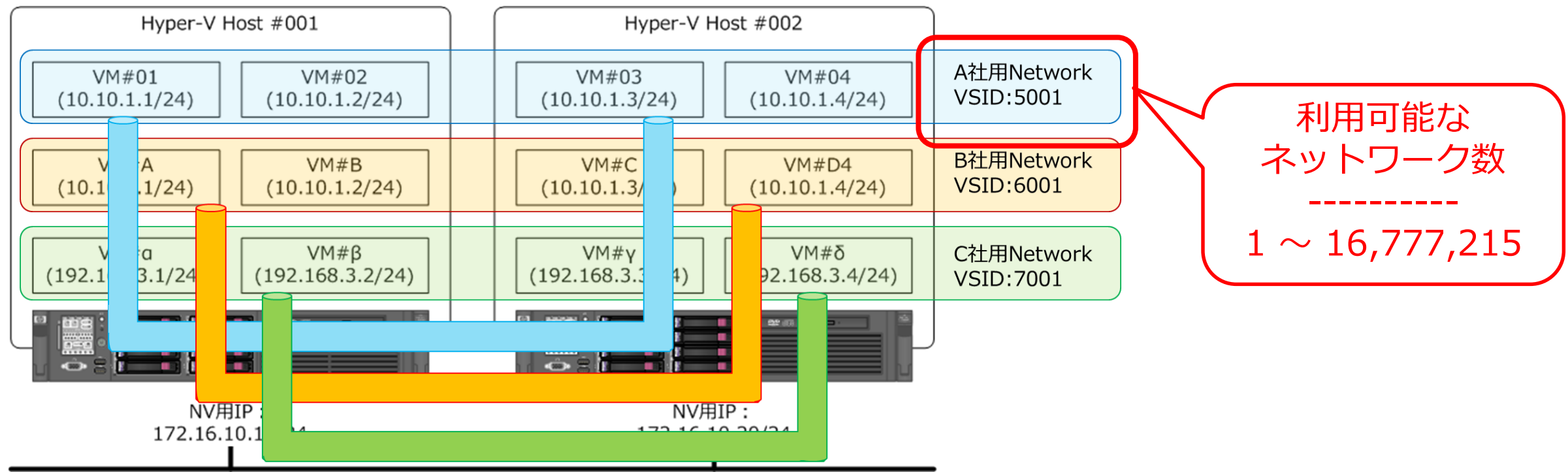
# アジェンダ

- NVGRE とは？
- Windows Server 2012 Network Virtualization Deep Dive
- System Center 2012 Virtual Machine Manager SP1
- Demo
- Network Virtualization Gateway
- まとめ
- Q & A
- Appendix A : IP Rewrite とは？（軽く）
- Appendix B : Network Virtualization の PowerShell での実装例
- Appendix C : Network Virtualization 処理オーバーヘッドの考察

# 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（Hyper-V 仮想スイッチ）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
  - 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 )	<b>Protocol 0x2F ( 8bit )</b>	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 )	<b>VSID ( 24bit )</b>	<b>FlowID ( 8bit )</b>
--------------------------------	--------------------------------------	---------------------------	----------------------------

0x2F = GRE

Inner Ethernet Header :

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



# 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 interface  
Ethernet II, Src: Intel\_ (68:05:ca:), Dst: Cisco\_ (08:00:0c:2c:54:00)  
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 interface  
Ethernet II, Src: Intel\_ (68:05:ca:), Dst: Cisco\_ (08:00:0c:2c:54:00)  
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]
```

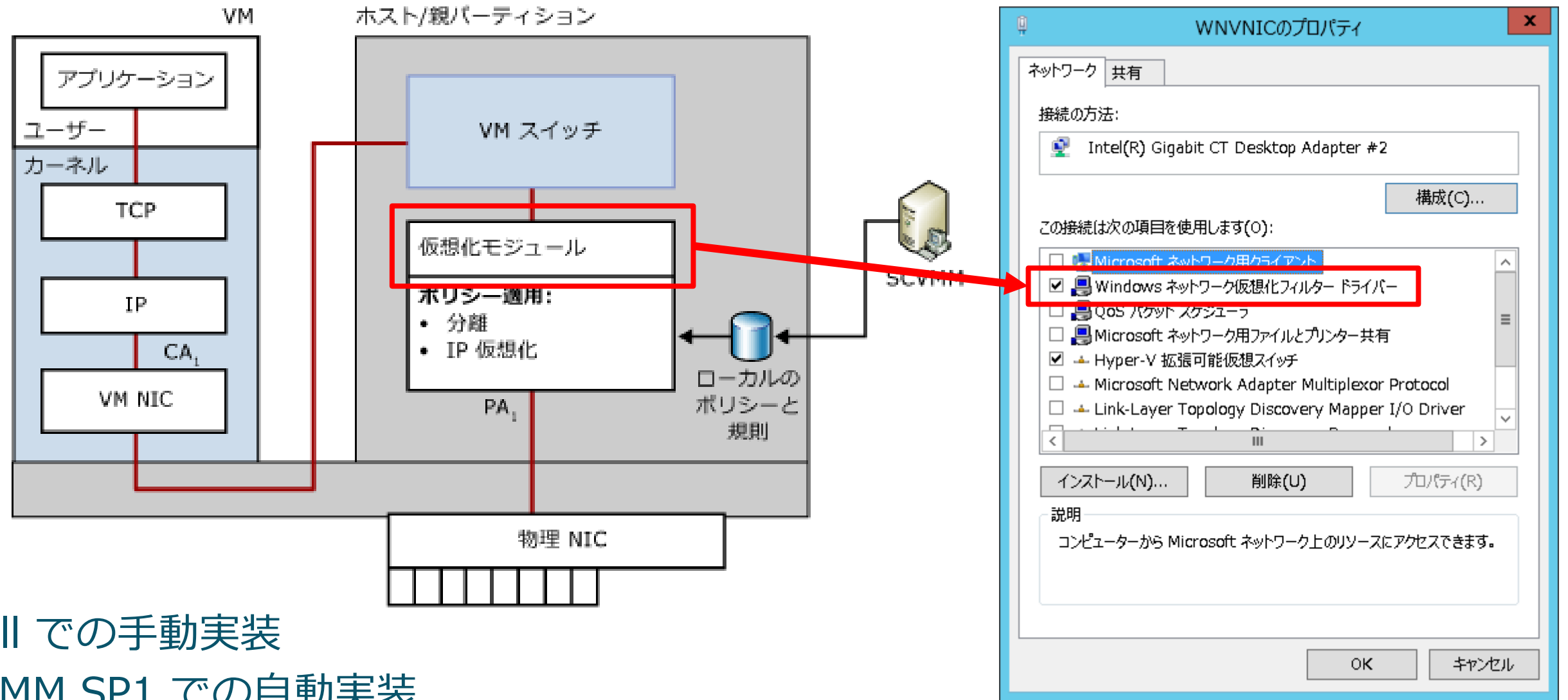
- これから検証を開始する場合、 3<sup>rd</sup> Party 実装の NVGRE 対応機器と接続試験をする場合、**必ず最新のパッチを適用してから開始してください**

# **Windows Server 2012 Network Virtualization Deep Dive**

# まずは用語の整理から

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	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**

# 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

# 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
```

# NVGREに関するいくつかの疑問

- NVGRE は GRE でオリジナルの Packet をカプセル化する方式である。
- いくつかの疑問……。
  - 実際に Network に流れるパケットサイズは？ Inner Frame が1518 byte packet だったら？
  - 仮想 Network 内で VLAN は 使用できるの？ Inner Frame に VLAN Tag は許容される？
  - VSID が異なる仮想 Network 間で通信したい場合はどうする？
  - 仮想 Network 内で Broadcast は使える？

# 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？



# 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。

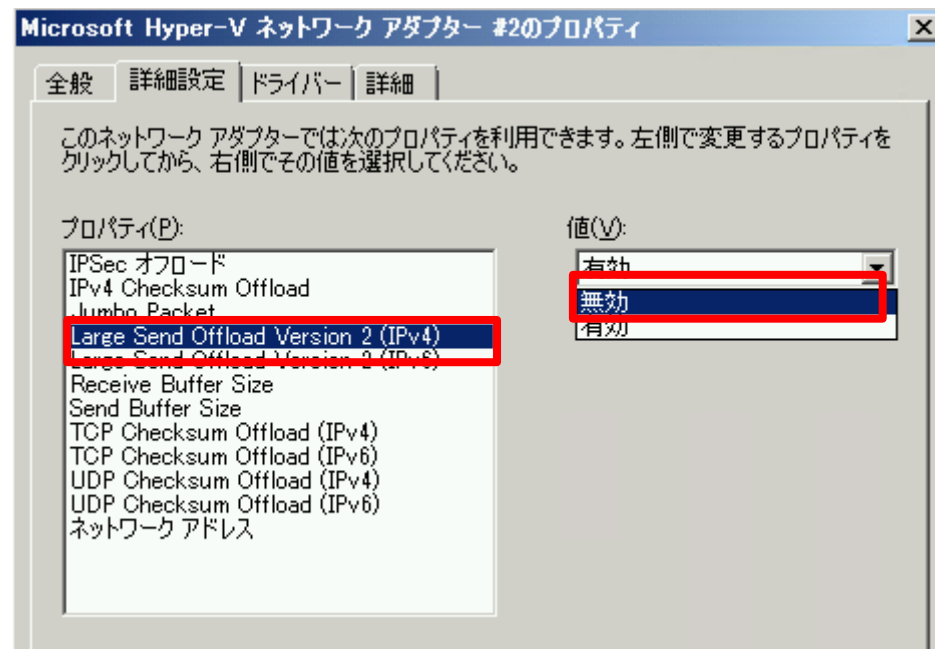
If a VLAN-tagged frame arrives encapsulated in NVGRE, then the decapsulating NVE SHOULD drop the frame.

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 capture of an ICMP packet. The packet list pane shows a packet of length 590 bytes at time 21:59:35.733921000. 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 following hex data:

```
0020 01 66 03 04 46 dc 00 00 05 b2 45 00 05 dc 00 4b .f.  
0030 40 00 80 06 70 b5 c0 a8 01 66 c0 a8 01 65 00 50 @..  
0040 c0 05 af c7 72 17 eb 5c 4e be 50 10 02 01 2d 16 ..  
0050 00 00 48 54 54 50 2f 31 2e 31 20 32 30 30 20 4f ..H  
0060 4b 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a K..  
0070 20 74 65 78 74 2f 70 6c 61 69 6e 0d 0a 4c 61 73 text/p|ain..Las
```

The status bar at the bottom indicates: MTU of next hop (icmp.mtu), 2 bytes. Packets: 283 Displayed: 283 Marked: 0 Load time: 0:00.060 Profile: Default

※ 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。

The image shows a Wireshark capture of NVGRE traffic. The main pane displays a list of packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. A red box highlights a gap in the time column between packets 34 and 35, with a callout box stating '時間差（空白の時間）の発生'. Another red box highlights the 'Length' column, showing values of 96 and 1514 bytes, with a callout box stating 'ICMP Packet は観測されず、Packet Size は 1514byte ※'. The bottom pane shows the details of a selected packet, including source and destination ports and sequence numbers.

No.	Time	Source	Destination	Protocol	Length	Info
32	21:59:34.465622000	192.168.1.101	192.168.1.102	HTTP	358	GET /testdata.txt HTTP/1.1
33	21:59:34.678571000	192.168.1.102	192.168.1.101	TCP	96	http > 49157 [ACK] Seq=1 Ack=263 win=131328 Len=0
34	21:59:36.422047000	192.168.1.102	192.168.1.101	TCP	1514	TCP segment of a reassembled PDU
35	21:59:36.631710000	192.168.1.101	192.168.1.102	TCP	96	49157 > http [ACK] Seq=263 Ack=1419 win=129980 Len=0

時間差（空白の時間）の発生

ICMP Packet は観測されず、  
Packet Size は 1514byte ※

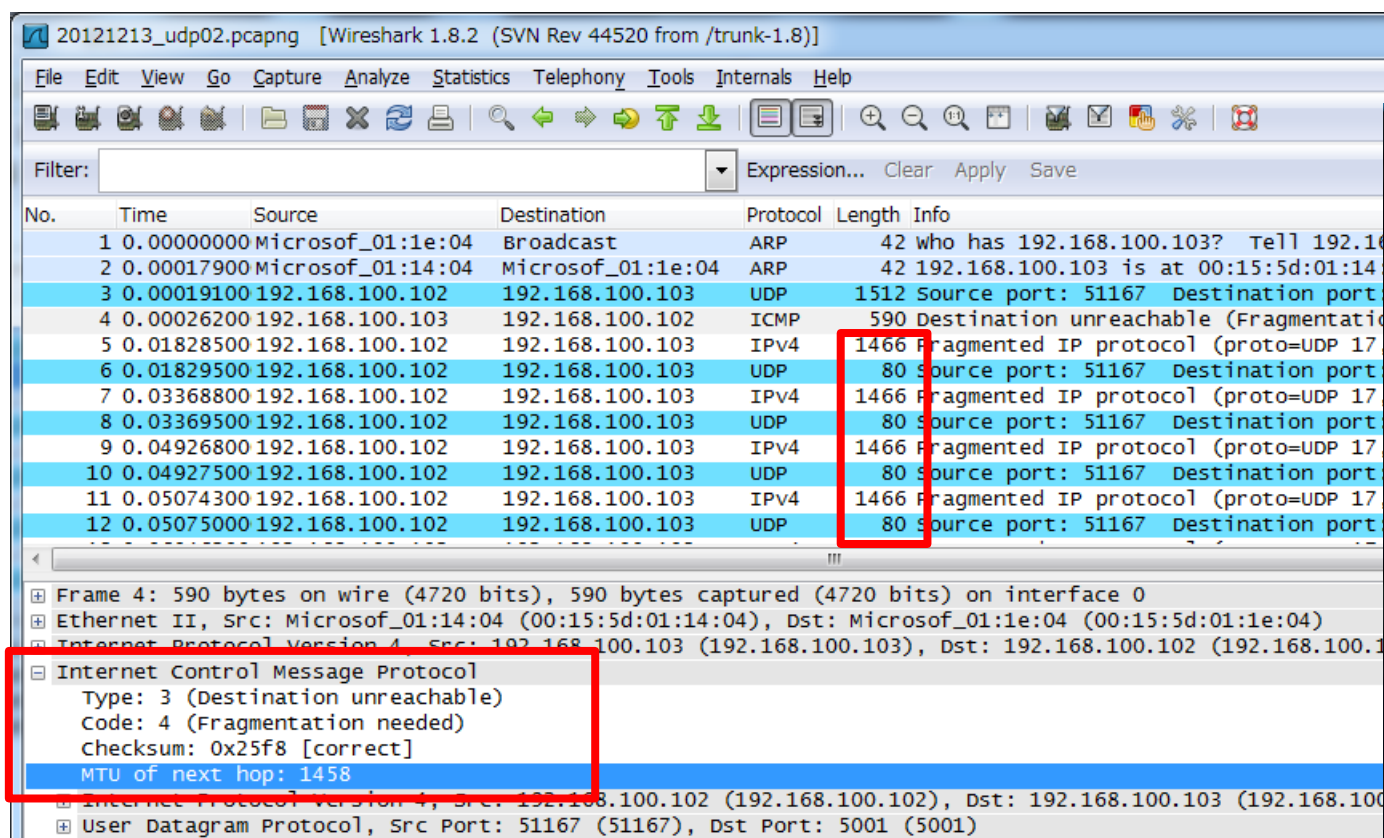
**結論としては、物理 Network 上での Packet 分割は発生していない模様**

# 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

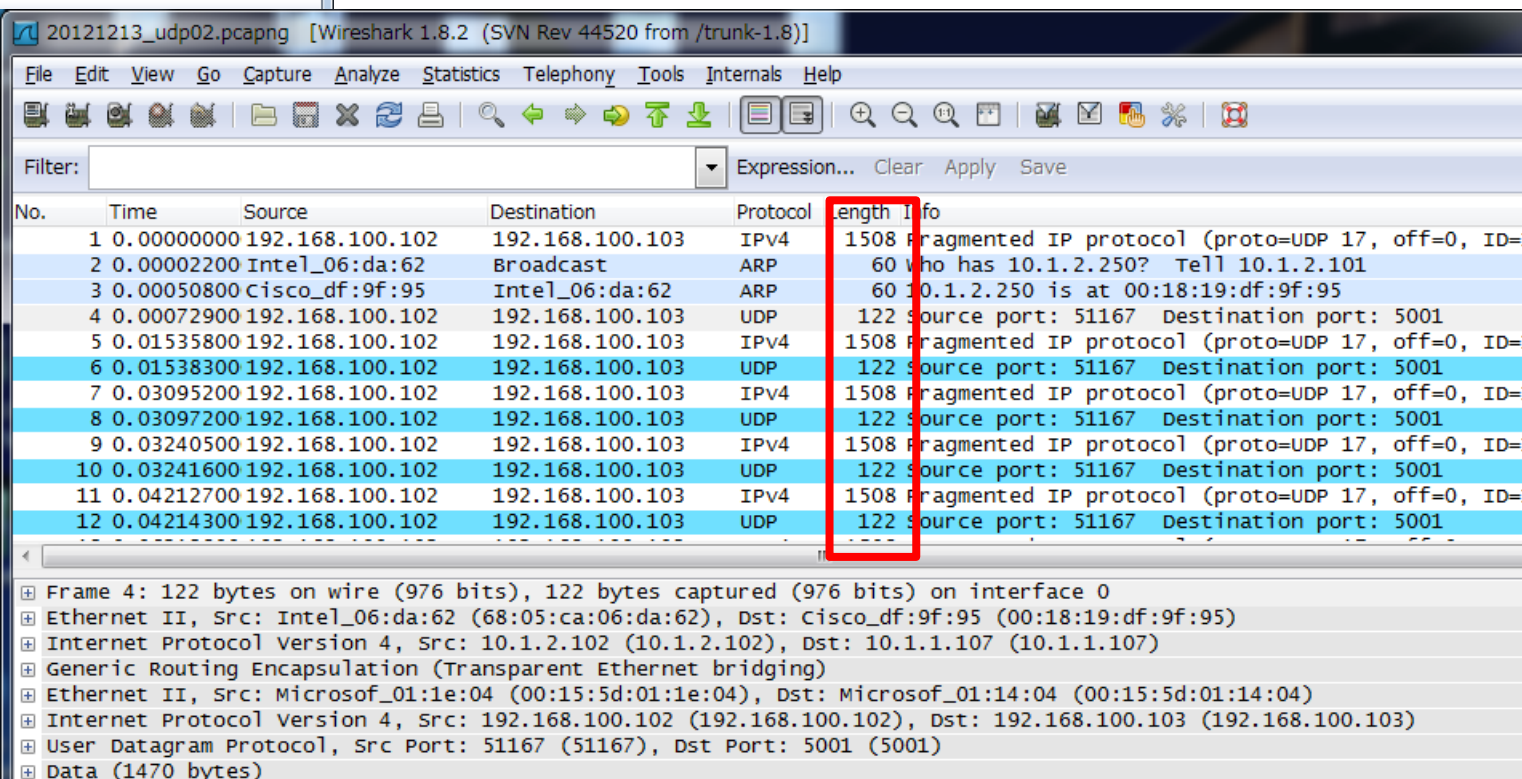
- 仮想マシン上の Packet で、icmp ( Path MTU Discovery ) を確認。次の Packet から MTU サイズを調整 / 分割 ( 1466byte + 80byte ※) して送信している事も確認しました。
- 物理 Network 上でも 1508byte + 122byte ( NVGRE オーバーヘッド 42byte ) Packet ※ で通信している事を確認しました。



Wireshark capture showing an ICMP Path MTU Discovery packet. The packet list table has a red box around the length '1466' in row 4. The packet details pane also has a red box around the 'MTU of next hop: 1458' field.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	Microsof_01:1e:04	Broadcast	ARP	42	who has 192.168.100.103? Tell 192.168.100.103
2	0.00017900	Microsof_01:14:04	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) [RST] Seq=0 Len=0
5	0.01828500	192.168.100.102	192.168.100.103	IPv4	1466	Fragmented IP protocol (proto=UDP 17, off=0, ID=122) [RST] Seq=0 Len=0
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) [RST] Seq=0 Len=0
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) [RST] Seq=0 Len=0
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) [RST] Seq=0 Len=0
12	0.05075000	192.168.100.102	192.168.100.103	UDP	80	Source port: 51167 Destination port: 5001

Frame 4: 590 bytes on wire (4720 bits), 590 bytes captured (4720 bits) on interface 0  
Ethernet II, Src: Microsof\_01:14:04 (00:15:5d:01:14:04), Dst: Microsof\_01:1e:04 (00:15:5d:01:1e:04)  
Internet Control Message Protocol  
Type: 3 (Destination unreachable)  
Code: 4 (Fragmentation needed)  
Checksum: 0x25f8 [correct]  
MTU of next hop: 1458



Wireshark capture showing an NVGRE encapsulated packet. The packet list table has a red box around the length '1508' in row 1. The packet details pane also has a red box around the 'Data (1470 bytes)' field.

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) [RST] Seq=0 Len=0
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) [RST] Seq=0 Len=0
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) [RST] Seq=0 Len=0
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) [RST] Seq=0 Len=0
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) [RST] Seq=0 Len=0
12	0.04214300	192.168.100.102	192.168.100.103	UDP	122	Source port: 51167 Destination port: 5001

Frame 4: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0  
Ethernet II, Src: Intel\_06:da:62 (68:05:ca:06:da:62), Dst: Cisco\_df:9f:95 (00:18:19:df:9f:95)  
Internet Protocol Version 4, Src: 10.1.2.102 (10.1.2.102), Dst: 10.1.1.107 (10.1.1.107)  
Generic Routing Encapsulation (Transparent Ethernet bridging)  
Ethernet II, Src: Microsof\_01:1e:04 (00:15:5d:01:1e:04), Dst: Microsof\_01:14:04 (00:15:5d:01:14:04)  
Internet Protocol Version 4, Src: 192.168.100.102 (192.168.100.102), Dst: 192.168.100.103 (192.168.100.103)  
User Datagram Protocol, Src Port: 51167 (51167), Dst Port: 5001 (5001)  
Data (1470 bytes)

※ FCS 含まず

# 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 Switch 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
```

**Network の Default Gateway**

**Subnet の Gateway Address**

**Routing Domain ID が異なる為疎通不可**



# NVGRE での Broadcast 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される L2 Frame Size を確認したところ、1518byte でした。
- 異なる物理ホスト上の仮想マシン間の Broadcast 通信に関しては、RFC ドラフトでは、Multicast を利用して通信可能と記載されています。
  - 但し、以下のようにも記載されています  
For interoperability reasons, future version of this draft will specify a standard way to map VSID to IP multicast address.
- 異なる物理ホスト上の仮想マシンの ARP は、仮想化フィルタードライバーで代理応答します。（『 Get-NetVirtualizationLookupRecord 』で得られるテーブルに基づいて処理されます）

# 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。

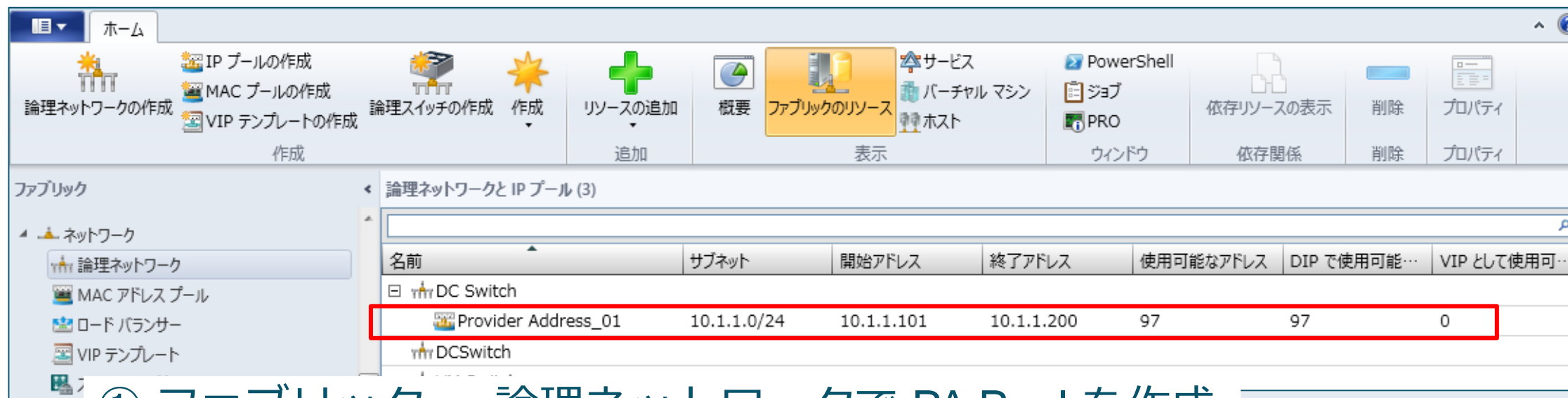
# **System Center 2012 Virtual Machine Manager SP1**

**Network Virtualization を中心に**

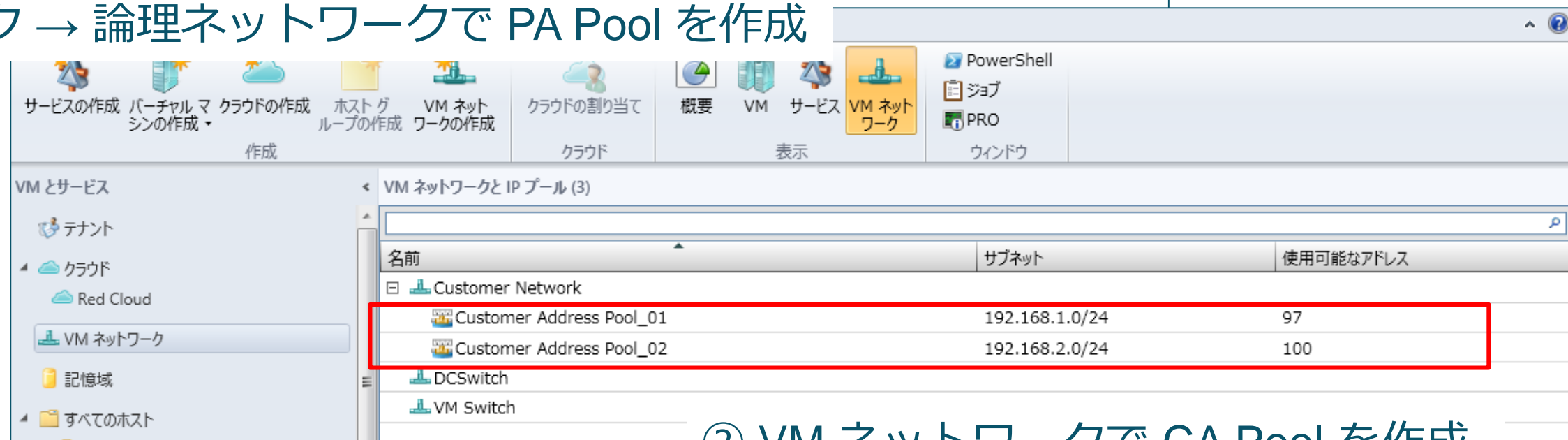
# SC2012 VMM SP1 での Network Virtualization

- 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
- 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
  - 既存 VM を Cloud に参加させ、Network Virtualization に追加した場合は、DHCP のみ使用可能

# 具体的な SC2012 VMM SP1 ネットワーク設定

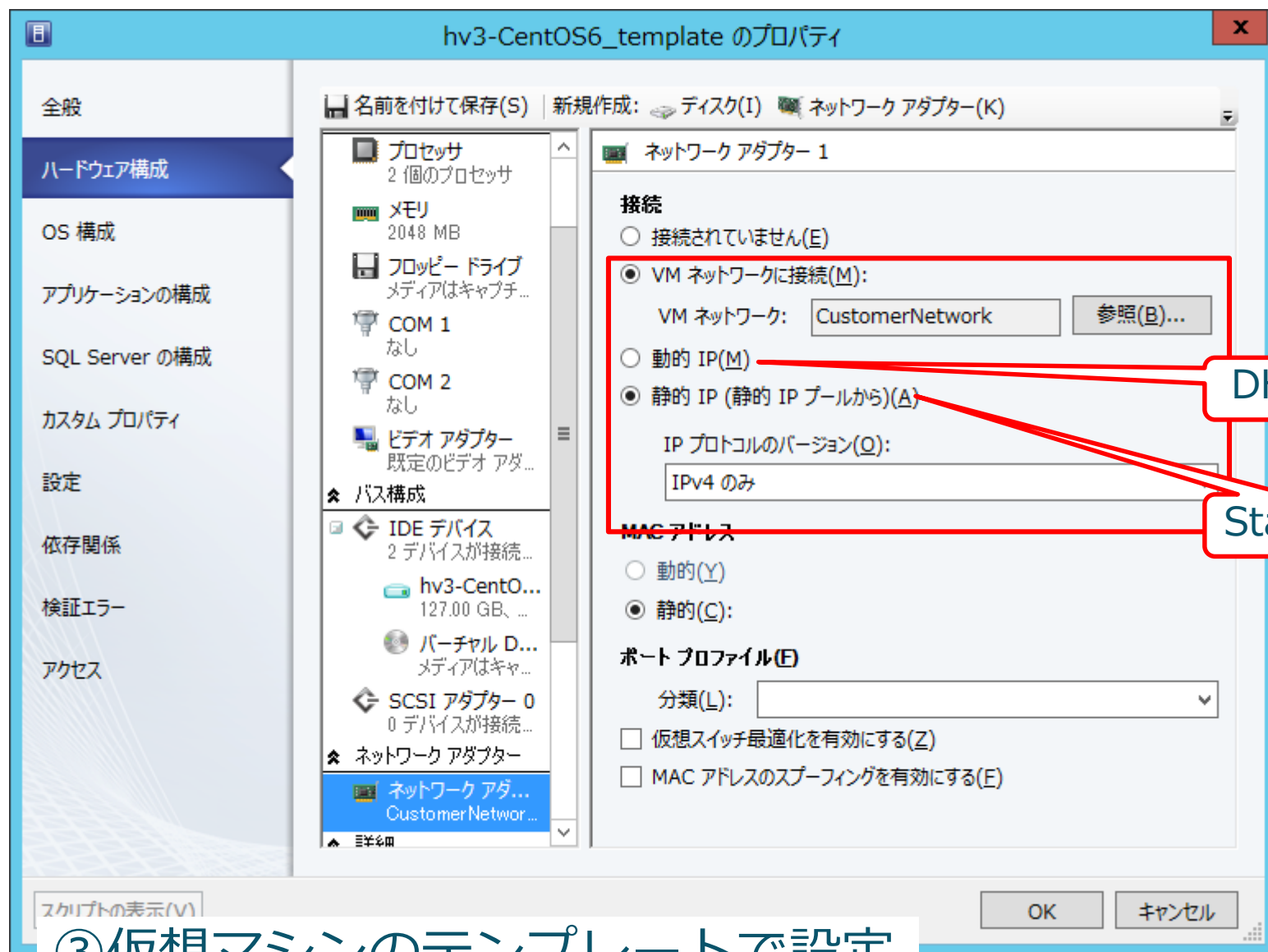


① ファブリック → 論理ネットワークで PA Pool を作成

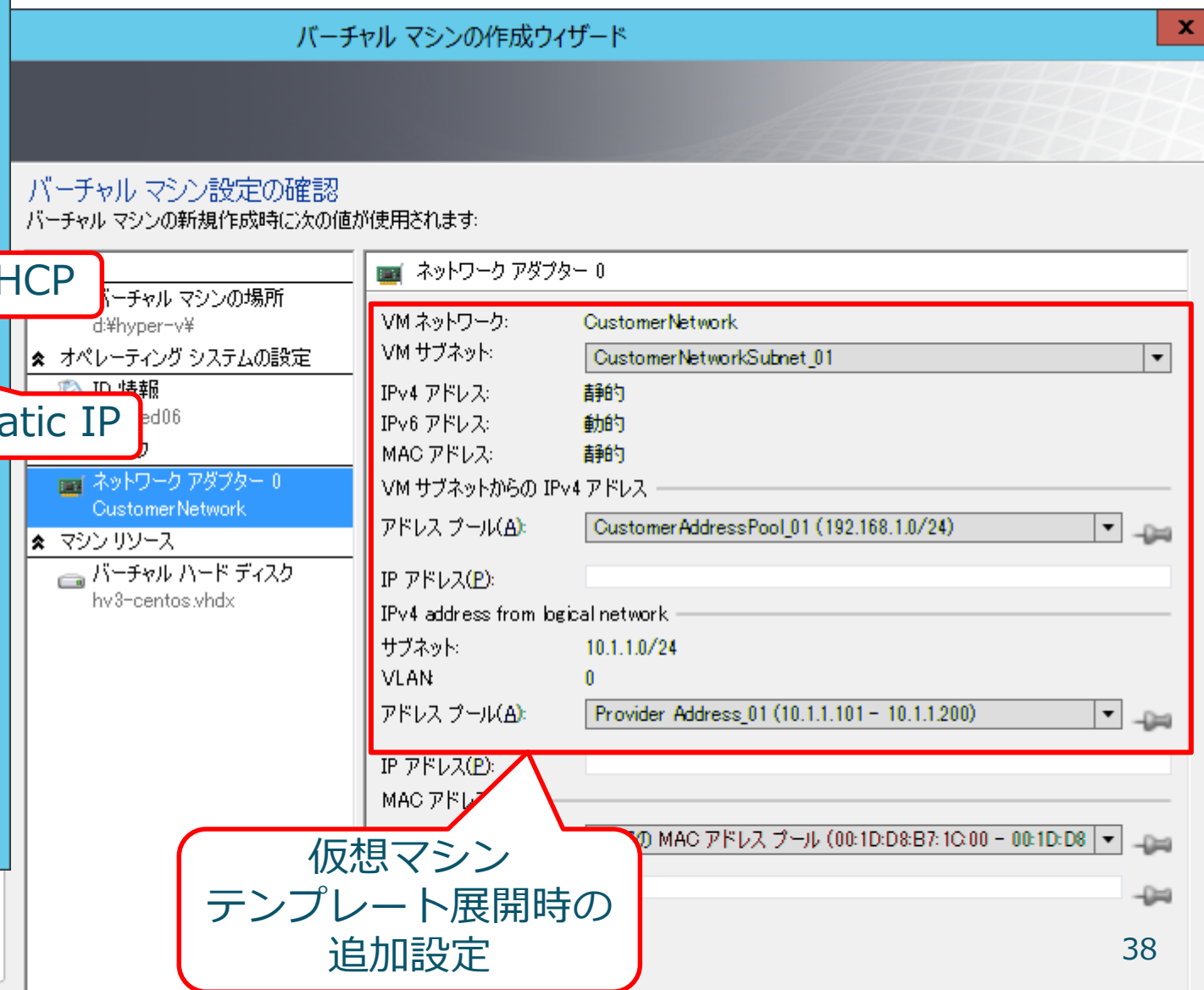


② VM ネットワークで CA Pool を作成

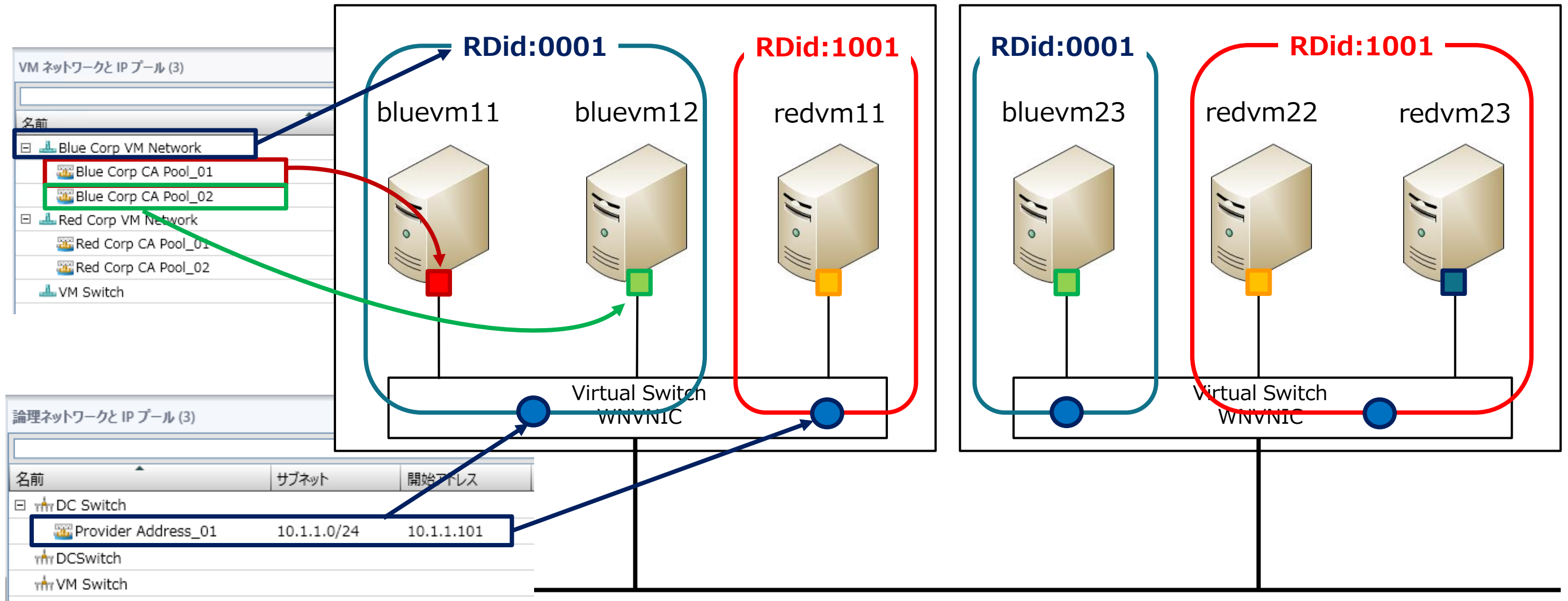
# 具体的な SC2012 VMM SP1 ネットワーク設定



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

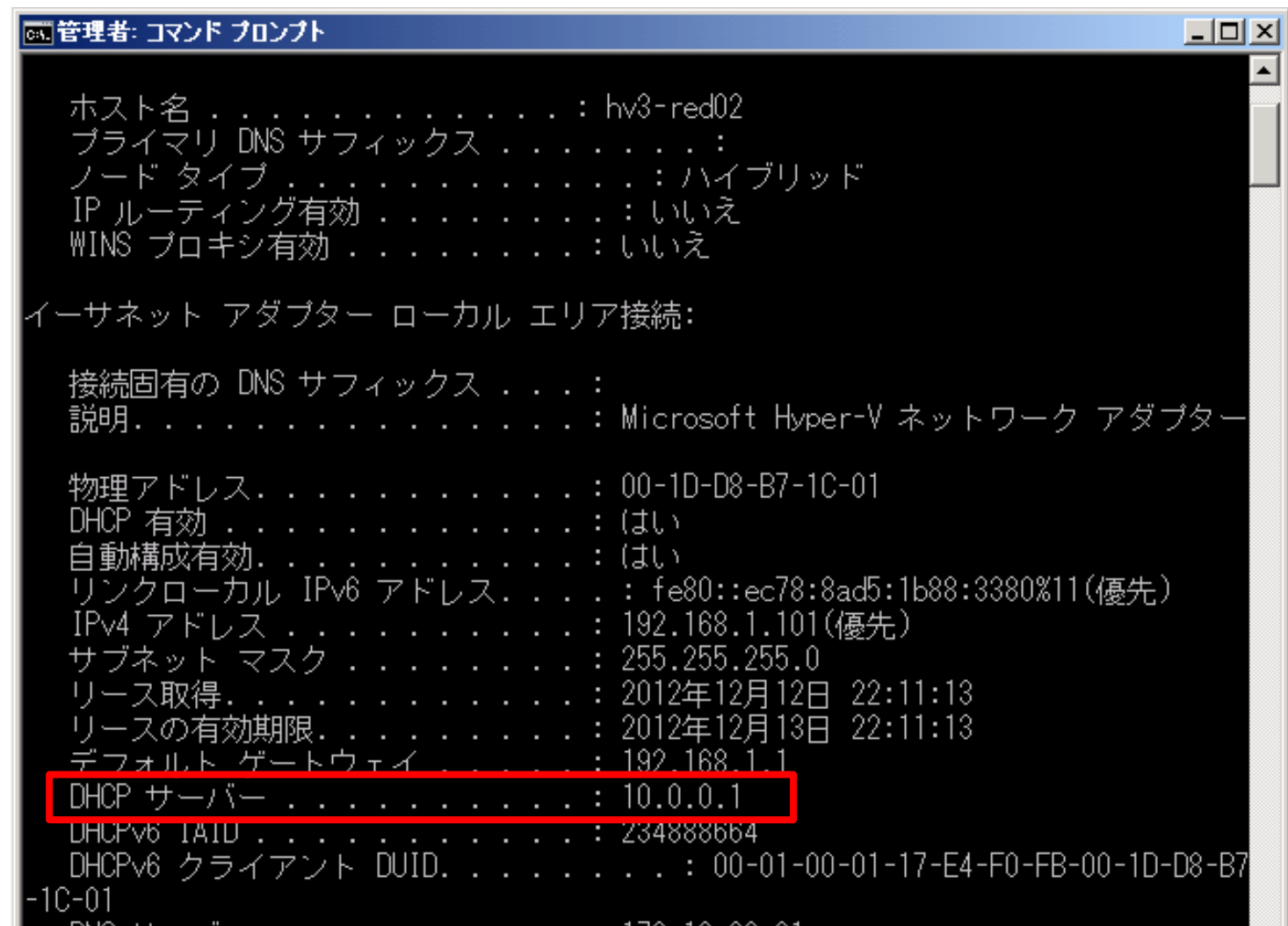


PAは、同一ホスト内であっても、Routing Domain ID単位で個別にアサインされる。



# 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
- 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、条件を精査する必要あり



```
管理者: コマンド プロンプト

ホスト名 . . . . . : hv3-red02
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 ローカル エリア接続:

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-01
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::ec78:8ad5:1b88:3380%11(優先)
IPv4 アドレス . . . . . : 192.168.1.101(優先)
サブネット マスク . . . . . : 255.255.255.0
リース取得 . . . . . : 2012年12月12日 22:11:13
リースの有効期限 . . . . . : 2012年12月13日 22:11:13
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 IAU . . . . . : 234888664
DHCPv6 クライアント DUID . . . . . : 00-01-00-01-17-E4-F0-FB-00-1D-D8-B7-1C-01
DNS . . . . . : 170.10.00.01
```



# SC2012 VMM SP1 での DHCP 実装

20121213\_dhcp.pcapng [Wireshark 1.8.2 (SVN Rev 44520 from /trunk-1.8)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
17	264.704673	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xecb0e2aa
18	264.713182	10.0.0.1	192.168.100.102	DHCP	328	DHCP Offer - Transaction ID 0xecb0e2aa
19	264.713607	0.0.0.0	255.255.255.255	DHCP	358	DHCP Request - Transaction ID 0xecb0e2aa
20	264.714044	10.0.0.1	192.168.100.102	DHCP	328	DHCP ACK - Transaction ID 0xecb0e2aa

Frame 18: 328 bytes on wire (2624 bits), 328 bytes captured (2624 bits) on interface 0

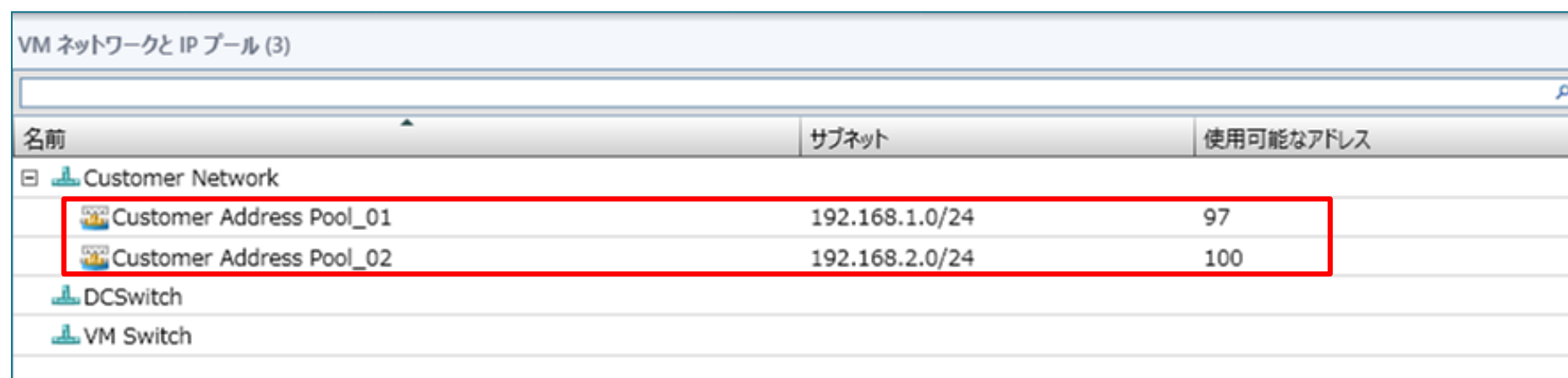
- Ethernet II, Src: 12:34:56:78:90:ab (12:34:56:78:90:ab), Dst: Microsof\_01:1e:04 (00:15:5d:01:1e:04)
  - Destination: Microsof\_01:1e:04 (00:15:5d:01:1e:04)
  - Source: 12:34:56:78:90:ab (12:34:56:78:90:ab)
  - Type: IP (0x0800)
- Internet Protocol Version 4, Src: 10.0.0.1 (10.0.0.1), Dst: 192.168.100.102 (192.168.100.102)
- User Datagram Protocol, Src 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

# 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。



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

```
管理: 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: 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

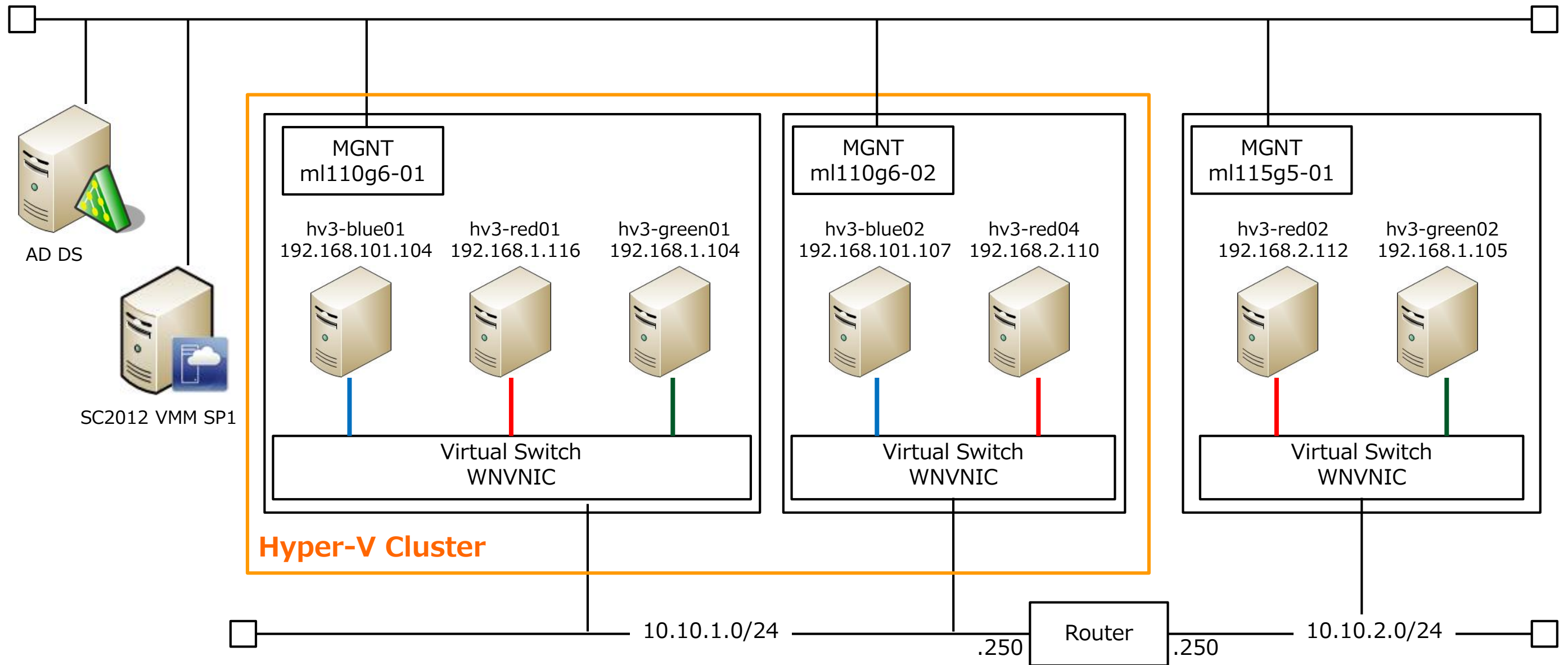
CustomerAddress : 192.168.1.54
VirtualSubnetID : 7696957
MACAddress       : 001dd8b71c00
ProviderAddress  : 10.1.1.55
CustomerID       : {D0CFFFE5-3A24-48DE-BC19-D99E071082FA}
```



Network Virtualization  
with SC2012 VMM SP1

DEMO

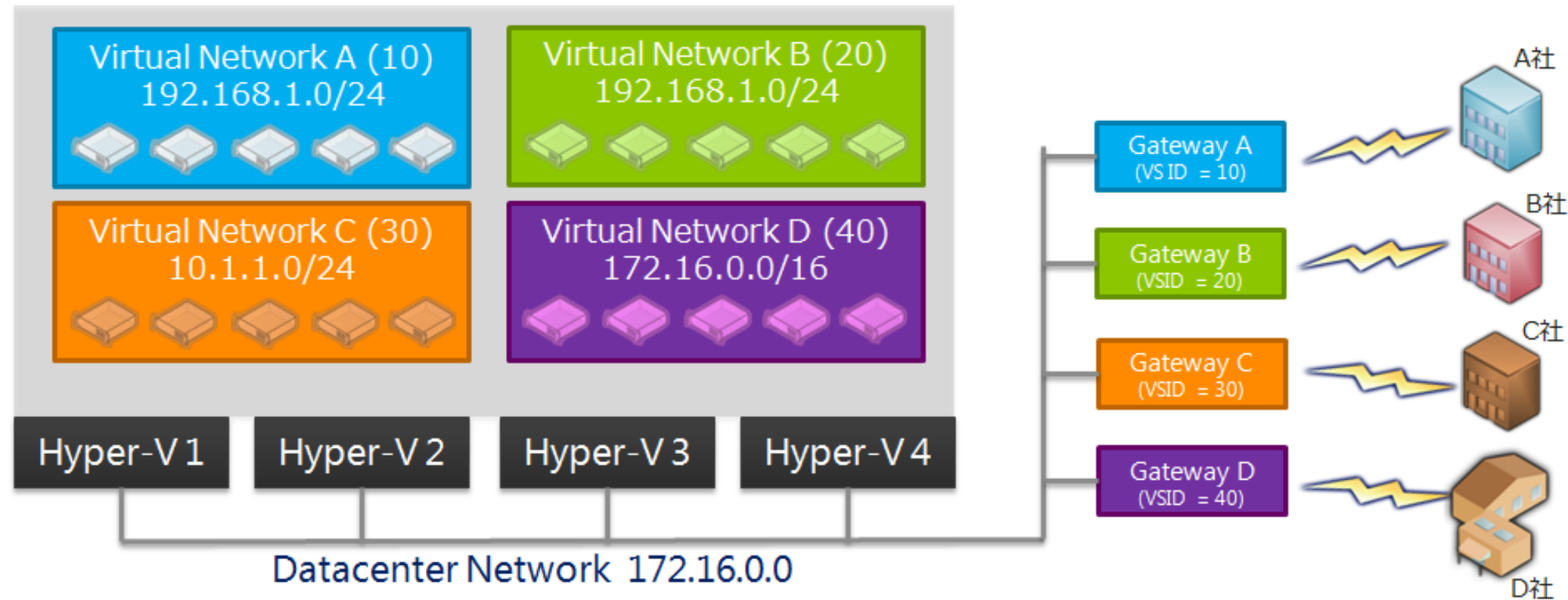
# 本日の Demo 環境



# 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
- 3<sup>rd</sup> Party から提供予定。



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
- NVGREでは Packetの Fragment が発生しますが、特定条件だと発生しません。  
アプリケーションの動作確認の際は、発生条件に注意して確認してください。  
また、アプリケーションでUDPを使用している場合には、特に注意が必要です。
- Provider Network ( Address ) 設計は、若干のコツがいる模様です。
- 対応製品の情報は、この後！

# 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>

Microsoft System Center Virtual Machine Manager 2012 ファーストルック ステップバイステップ ガイド

<http://technet.microsoft.com/ja-jp/virtualization/hh529164>

# 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（Hyper-V 仮想スイッチ）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

The image shows a Wireshark packet capture window for a file named 'hv3-blue01.pcapng'. The main display area shows a list of captured packets, with packet 57 selected. The packet list table is as follows:

No.	Time	Source	Destination	Protocol	Length	Info
57	11.0550180	192.168.1.101	192.168.1.102	SMB2	182	TreeConnect Request Tree: \\192.168.1.102\test_share

The packet details pane for packet 57 shows the following structure:

- Frame 57: 182 bytes on wire (1456 bits), 182 bytes captured (1456 bits)
- Ethernet II, Src: Microsof\_01:14:04 (00:15:5d:01:14:04), Dst: Microsof\_01:1e:04 (00:15:5d:01:1e:04)
- Internet Protocol Version 4, Src: 192.168.1.101 (192.168.1.101), Dst: 192.168.1.102 (192.168.1.102)
  - Version: 4
  - Header length: 20 bytes
  - Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
  - Total Length: 168
  - Identification: 0x005b (91)
  - Flags: 0x02 (Don't Fragment)
  - Fragment offset: 0
  - Time to live: 128
  - Protocol: TCP (6)
  - Header checksum: 0x0000 [incorrect, should be 0x75d9 (maybe caused by "IP checksum offload"?)]
  - Source: 192.168.1.101 (192.168.1.101)
  - Destination: 192.168.1.102 (192.168.1.102)
- Transmission Control Protocol, Src Port: 49157 (49157), Dst Port: microsoft-ds (445), Seq: 3902, Ack: 3942, Len: 128
  - Source port: 49157 (49157)
  - Destination port: microsoft-ds (445)
  - [Stream index: 1]
  - Sequence number: 3902 (relative sequence number)
  - [Next sequence number: 4030 (relative sequence number)]
  - Acknowledgement number: 3942 (relative ack number)
  - Header length: 20 bytes
  - Flags: 0x18 (PSH, ACK)
  - Window size value: 512

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```
0020 01 66 c0 05 01 bd 9f 12 65 d1 86 e5 40 a7 50 18 .f.....e...@.P.  
0030 02 00 84 b6 00 00 00 00 7c fe 53 4d 42 40 00 .....|.SMB@.  
0040 01 00 00 00 00 00 03 00 01 00 00 00 00 00 00 .....  
0050 00 00 15 00 00 00 00 00 00 00 ff fe 00 00 00 00 .....  
0060 00 00 01 00 00 00 00 00 04 00 00 00 00 00 00 .....  
0070 00 00 00 00 00 00 00 00 00 09 00 00 00 48 00 .....H.
```

The status bar at the bottom indicates: Transmission Control Protocol (tcp), 20 bytes | Packets: 163 Displayed: 163 Marked: 0 Load time: 0:00.060 | Profile: Default

# IP Rewrite パケットキャプチャ : Network

The image shows a Wireshark packet capture window for a file named 'ml110g6-01\_20120918.pcap'. The interface includes a menu bar, a toolbar, and a filter field. The packet list pane shows two packets: packet 73 (SMB2 TreeConnect Request) and packet 74 (SMB2 TreeConnect Response). Packet 73 is selected, and its details pane shows the following information:

- Frame 73: 182 bytes on wire (1456 bits), 182 bytes captured (1456 bits)
- Ethernet II, Src: IntelCor\_...:cd (00:1b:21:...:cd), Dst: IntelCor\_...:d3 (00:1b:21:...:d3)
- Internet Protocol Version 4, Src: 10.1.1.20 (10.1.1.20), Dst: 10.1.1.30 (10.1.1.30)
  - Version: 4
  - Header length: 20 bytes
  - Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
  - Total Length: 168
  - Identification: 0x005b (91)
  - Flags: 0x02 (Don't Fragment)
  - Fragment offset: 0
  - Time to live: 128
  - Protocol: TCP (6)
  - Header checksum: 0xe3c1 [correct]
  - Source: 10.1.1.20 (10.1.1.20)
  - Destination: 10.1.1.30 (10.1.1.30)
- Transmission Control Protocol, Src Port: 49157 (49157), Dst Port: microsoft-ds (445), Seq: 3902, Ack: 3942, Len: 128
  - Source port: 49157 (49157)
  - Destination port: microsoft-ds (445)
  - [Stream index: 2]
  - Sequence number: 3902 (relative sequence number)
  - [Next sequence number: 4030 (relative sequence number)]
  - Acknowledgement number: 3942 (relative ack number)
  - Header length: 20 bytes
  - Flags: 0x18 (PSH, ACK)
  - window size value: 512

The packet bytes pane shows the raw data in hexadecimal and ASCII. The ASCII portion is: `...!.....!.....E.  
...[@.....  
.....e...@.P.  
.....|.SMB@.  
.....  
.....`

The status bar at the bottom indicates: Transmission Control Protocol (tcp), 20 bytes | Packets: 184 Displayed: 184 Marked: 0 Load time: 0:00.000 | Profile: Default

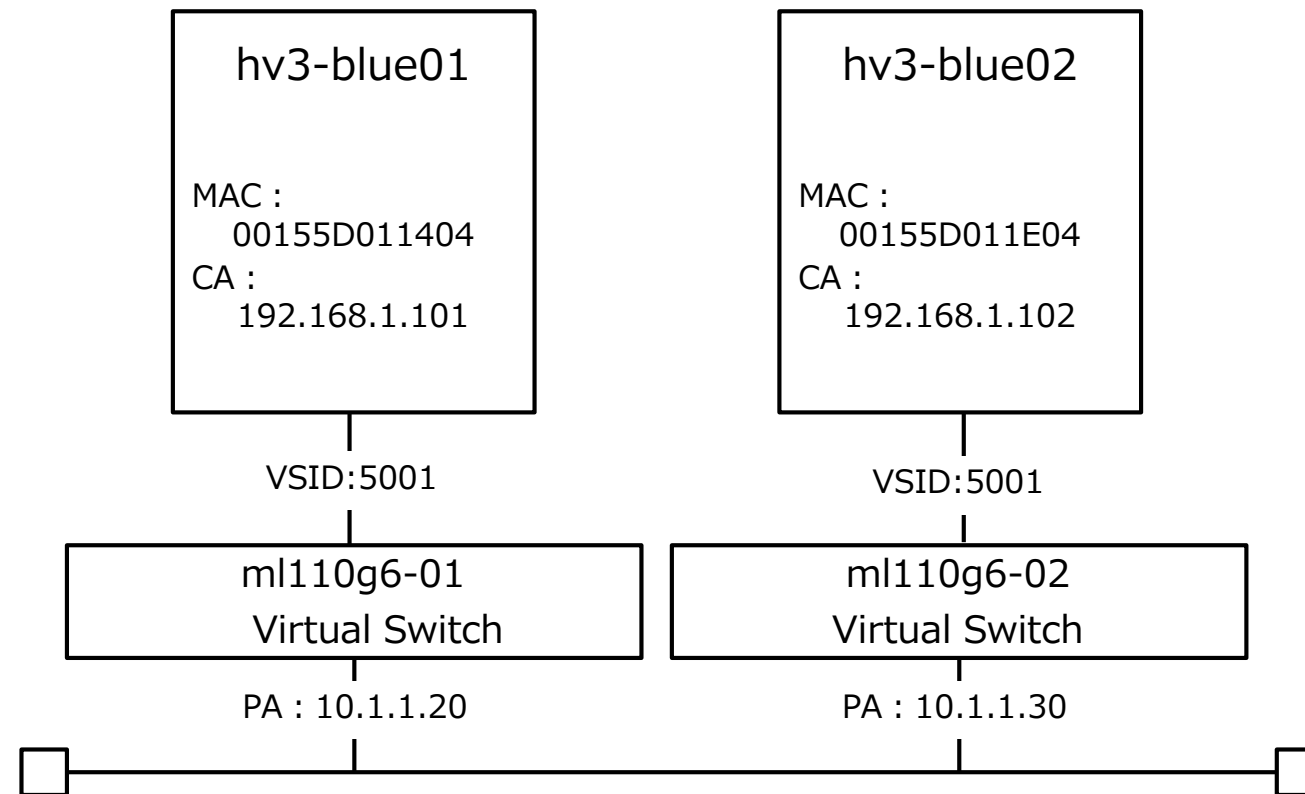


# Appendix B : Network Virtualization の PowerShell での実装例

## 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;
}

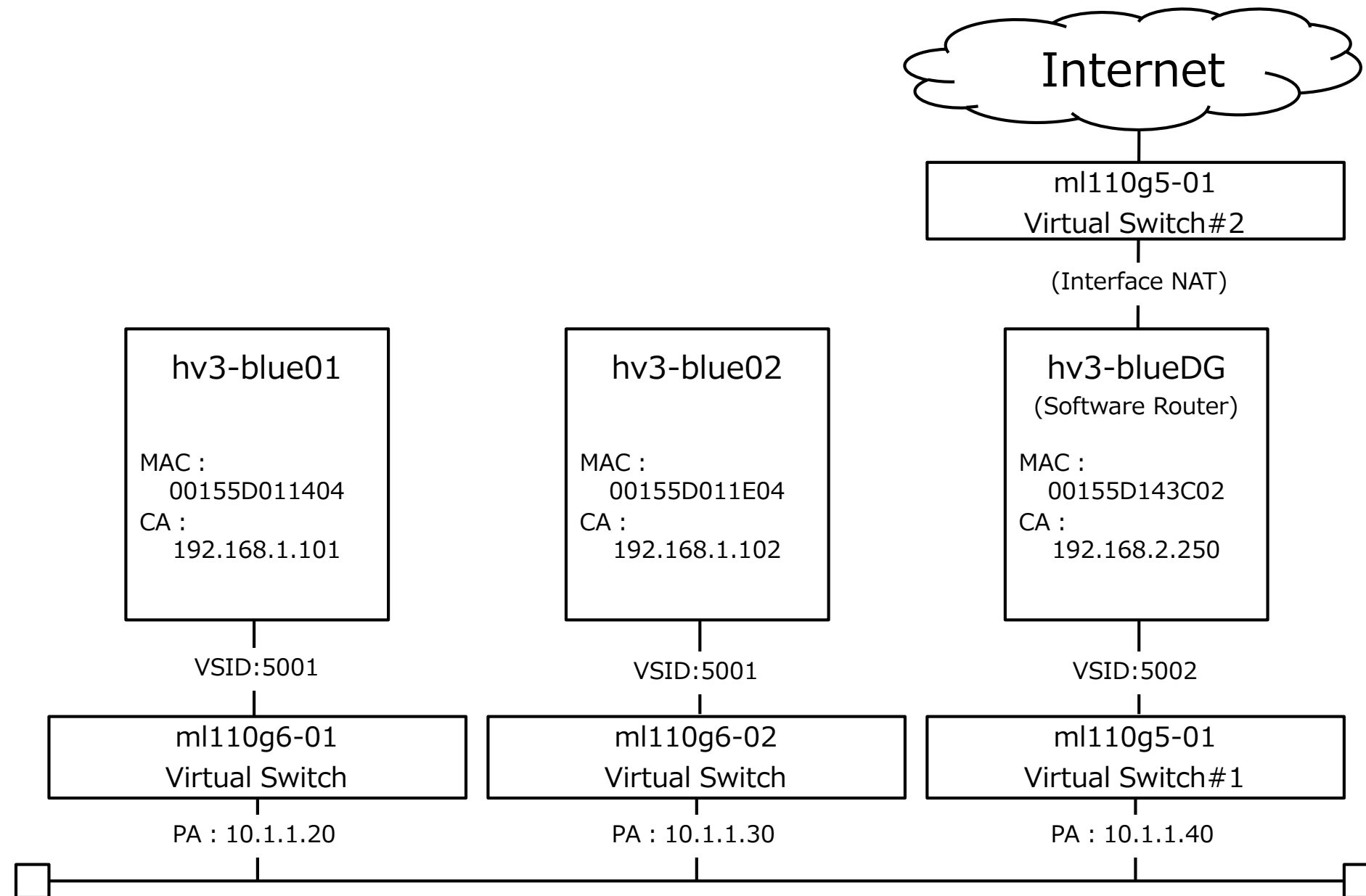
$iface = Get-NetAdapter $WNVNIC -CimSession "ml110g6-02"
New-NetVirtualizationProviderAddress -InterfaceIndex $iface.InterfaceIndex -ProviderAddress "10.1.1.30" -PrefixLength 24 -CimSession "ml110g6-02"

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;
}
```

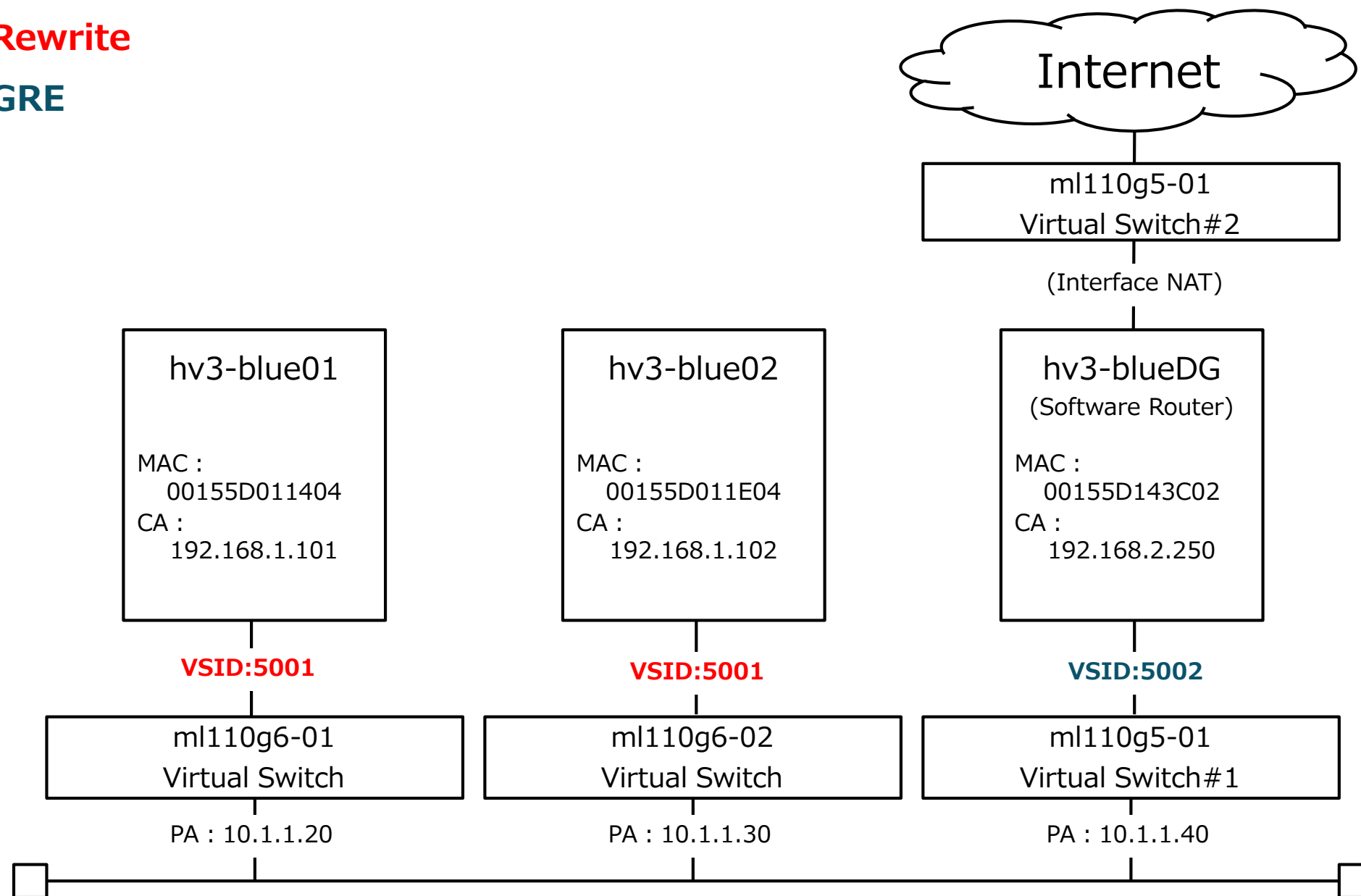
## 実装例（3）超応用形

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;
}

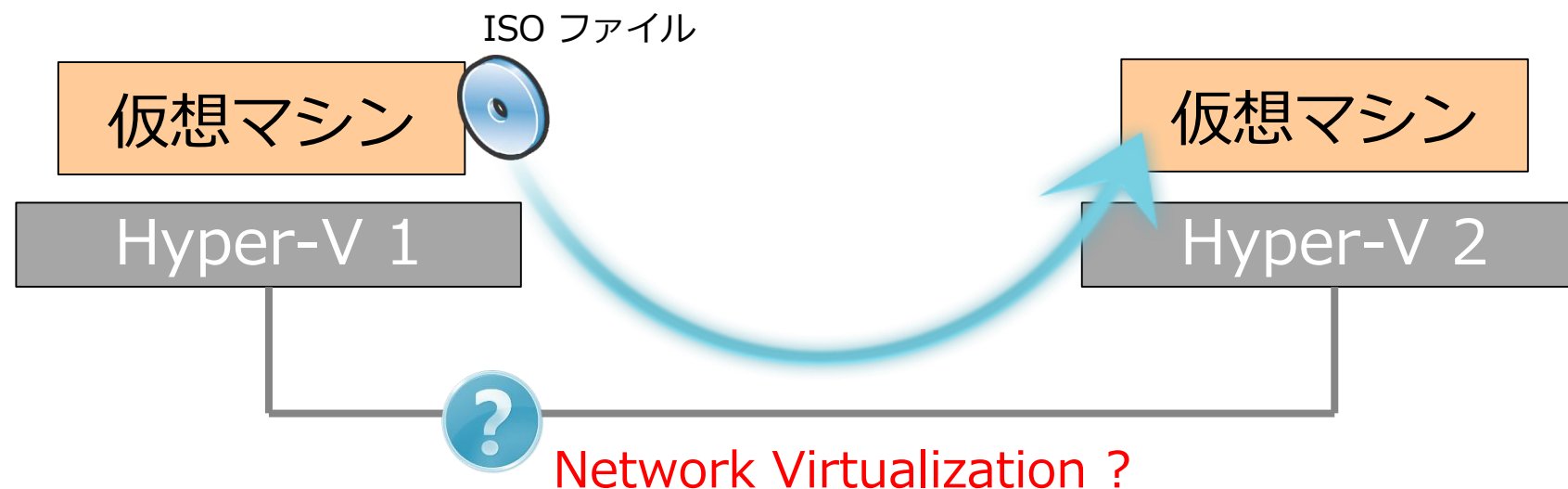
$iface = Get-NetAdapter $WNVNIC -CimSession "ml110g6-02"
New-NetVirtualizationProviderAddress -InterfaceIndex $iface.InterfaceIndex -ProviderAddress "10.1.1.30" -PrefixLength 24 -CimSession "ml110g6-02"
Invoke-Command -ComputerName "ml110g6-02" -Credential $cred {
Get-VMNetworkAdapter "hv3-blue02" | where {$_.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 : Network Virtualization 処理オーバーヘッドの考察

# 比較テスト（1）

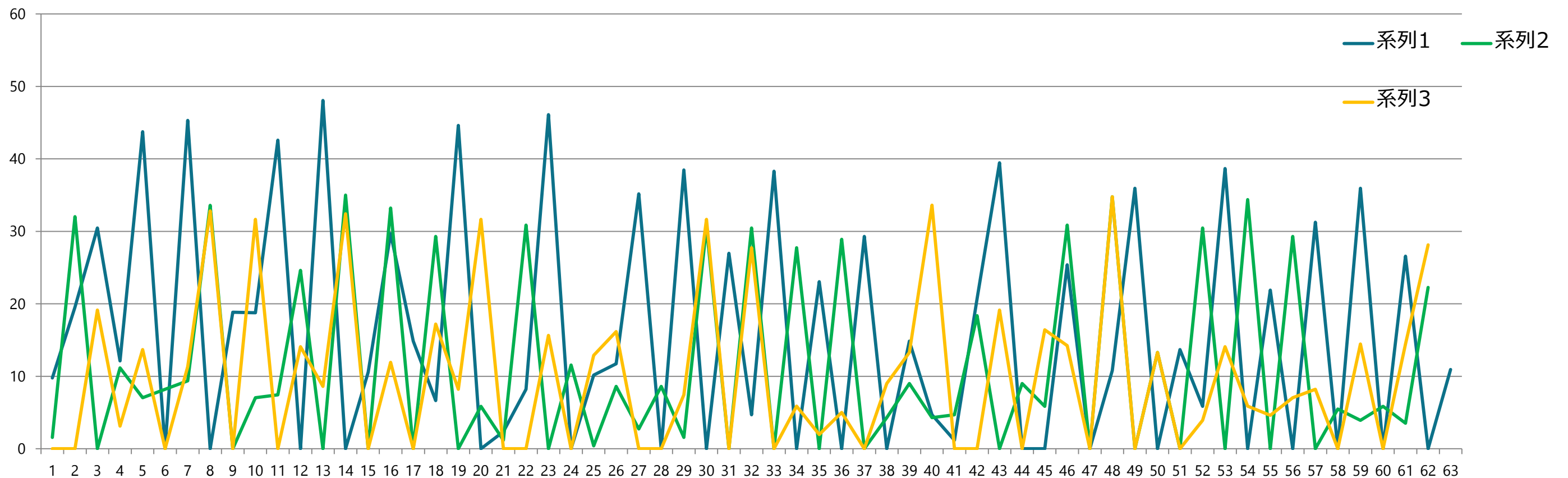
- 2 台の物理ホスト上に配置された、2 台の仮想マシンを使用
- 極力条件を同一にする為に、テスト前にスナップショットを取得し、テスト後にスナップショットの破棄を実施
- Network Virtualization 未実施、NVGRE、IP Rewrite の各方式で 3.89GB の ISO ファイルを仮想マシン間でコピーして、コピー時間を計測
- 試行回数 5 回での平均値を結果として採用
- 同時に、コピー中の物理ホストの CPU 利用率をパフォーマンスモニターにて計測





# 比較テスト（1）：結果

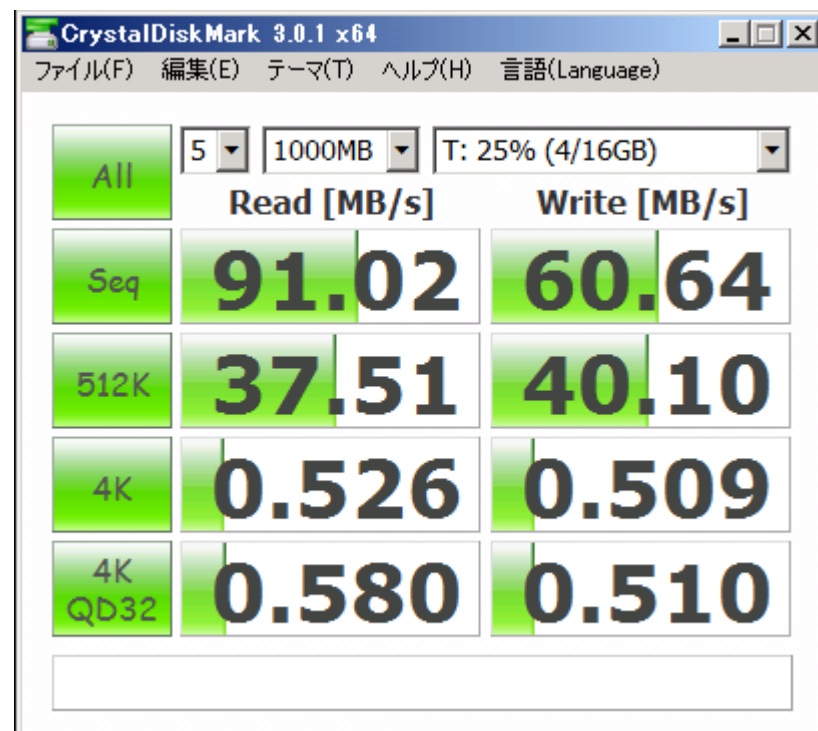
NV実施方法	平均コピー時間	平均スループット	平均CPU利用率
Network Virtualization なし	1分3秒49	503.9Mbps	11.46%
NVGRE	1分2秒09	513.8Mbps	18.71%
IP Rewrite	1分1秒88	516.0Mbps	14.34%



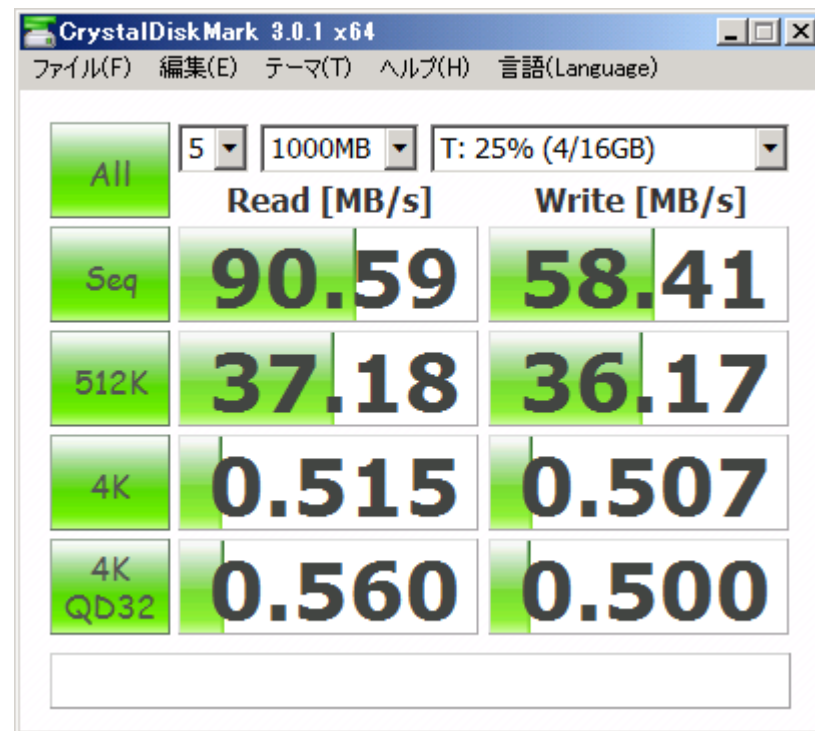
## 比較テスト（２）

- 2 台の物理ホスト上に配置された、2 台の仮想マシンを使用
- 極力条件を同一にする為に、テスト前にスナップショットを取得し、テスト後にスナップショットの破棄を実施
- Network Virtualization 未実施、NVGRE、IP Rewrite の各方式で、共有フォルダの Disk I/O Benchmark テストを実施
- 試行回数 5 回での平均値を結果として採用

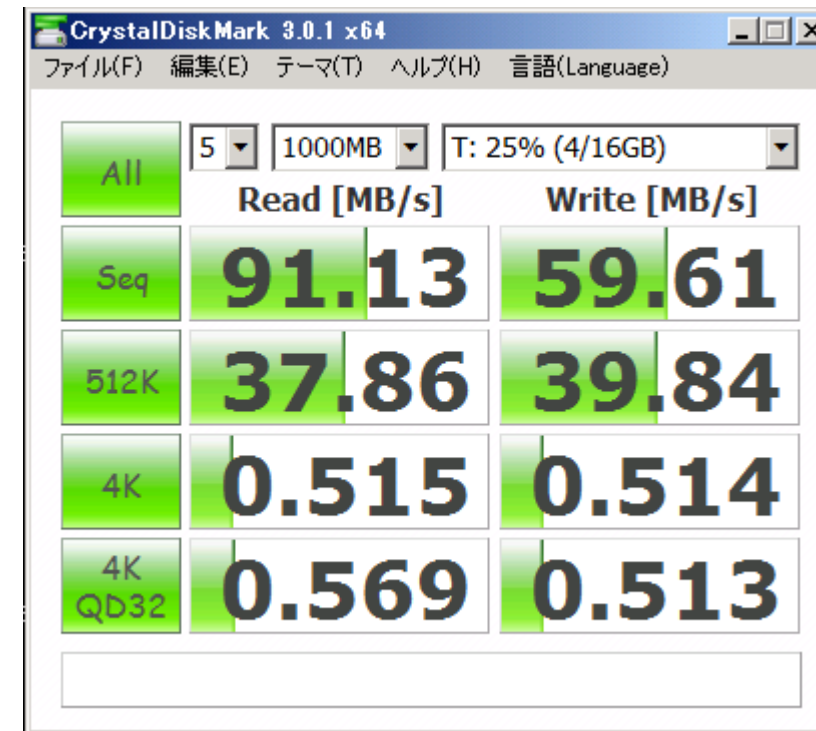
# 比較テスト（2）：結果



Network Virtualization なし



NVGRE



IP Rewrite

※試行回数 5 回の中で、平均値に最も近いベンチマーク結果を掲載