Microsoft

Optimizing and Troubleshooting

Hyper-V Networking

Mitch Tulloch with
the Windows Server Team

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Control Number (PCN): 2013938862
ISBN: 978-0-7356-7900-9

Printed and bound in the United States of America.

First Printing

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Book Support at mspinput@microsoft.com. Please tell us what you think of this book at *http://www.microsoft.com/learning/booksurvey*.

Acquisitions Editor: Anne Hamilton
Developmental Editor: Karen Szall
Project Editor:
Editorial Production: Jean Trenary
Technical Reviewer:
Copyeditor: Megan Smith-Creed
Indexer:
Cover:

Contents

Introduction

Troubleshooting is a difficult art to learn because it requires deep knowledge of the subject of study, familiarity with a wide variety of tools, and thinking that can be both sequentially logical and inspirationally outside the box. Perhaps the best way of learning such arts is by watching experts demonstrate their skills as they are exhibited in different situations.

Optimizing how something performs can also be quite difficult to master. If you've ever used an old-fashioned radio where you had to find your station using a dial, you'll realize that a certain degree of fiddling is required to tune things just right. Now imagine a device that has dozens of dials, each tuning a different variable, with all the variables related to one another so that tuning one affects the settings of the others. Tuning an information technology system can often be just like that…or worse!

Optimizing and Troubleshooting Hyper-V Networking is all about watching the experts as they configure, maintain, and troubleshoot different aspects of physical and virtual networking for Hyper-V hosts and the virtual machines running on these hosts. And when I use the word "expert" here, I really mean it, because the contributors to this book all work at Microsoft and have first-hand knowledge and experience with the topics they cover. The different sections in this book range from how to automate the network configuration of Hyper-V hosts using Windows PowerShell to get it right the first time so you won't have to troubleshoot, to step-by-step examples of how different networking problems were identified, investigated, and resolved.

Of course there's no way to exhaustively or even systematically cover the subject of optimizing and troubleshooting Hyper-V networking in a short book like this. But I hope that by reading this book (or by referring to certain topics when the need arises) your own troubleshooting skills will become more finely honed so you will be able to apply them more effectively even in scenarios that are not described in this text.

This book assumes that you are a moderately experienced administrator of the Windows Server virtualization platform. You should also have at least a basic understanding of Windows PowerShell and familiarity with tools and utilities for managing Windows servers, Hyper-V hosts, virtual machines, and the various components of an enterprise networking infrastructure.

The main focus of this book is on the Windows Server 2012 version of Hyper-V and associated networking capabilities. Some content in this book may also be applicable for earlier versions of Hyper-V and Windows Server, and we've tried to indicate this wherever applicable.

Good luck in mastering this arcane art!

—Mitch Tulloch, Series Editor

About the contributors

Cristian Edwards Sabathe is the EMEA Regional Workload Lead for Server Virtualization based in Barcelona, Spain. Cristian has over five years of support and virtualization experience and has a deep technical hands-on experience with Hyper-V and SCVMM since Windows 2008. He is a Subject Matter Expert in the WW Microsoft Virtualization team and content creator of Workshops for Premier and MCS customers. Together with the SCOM PFE Diego Martinez Rellan, he is also the author of the *Hyper-V Management Pack Extensions* available from http://hypervmpe.codeplex.com. Cristian's contributions to the community can be found on his personal blog at http://blogs.technet.com/cedward and in the World Wide PFE virtualization blog at http://blogs.technet.com/virtualpfe.

Jason Dinwiddie is a Senior Consultant with Microsoft Consulting Services. Jason is an eight-year veteran at Microsoft as a Senior Consultant for State and Local Government. With 16 years of overall IT experience, Jason is focused on virtualization, management, and private cloud, specializing in Hyper-V.

Jean-Pierre R M de Tiege is a Senior Technologist for Charteris (http://www.charteris.com) currently working at Microsoft on the Government Gateway team as a build manager. Jean-Pierre has worked in a variety of fields over the last 14 years, from e-learning to e-commerce, and has worked with Microsoft technology since the first .NET version came out, initially in the Netherlands but now full time in the United Kingdom.

Jeff Stokes is a Senior Premier Field Engineer (PFE) at Microsoft. Jeff has been in the IT industry for 19 years, initially cutting his teeth at DEC and climbing the system administrator ladder from there. He regularly posts to his popular TechNet blog "Dude Where's My PFE?" which can be found at http://blogs.technet.com/b/jeff_stokes/.

Keith Hill is a Senior Support Escalation Engineer with the Windows Server Core High Availability Team. Keith started his Microsoft journey in 1999 on the afterhours support team. He moved to the cluster team about seven years later, and two year ago became the Support Topic Owner for Hyper-V within Commercial Technical Support (CTS). Keith would like to thank **John Howard**, Program Manager for Hyper-V, for his assistance in writing the SR-IOV section of this book. Keith would also like to thank **Tina Chapman**, a Lab Engineer with the US-CSS CC lab group, for her assistance in writing the NPIV section of this book.

Madhan Sivakumar is a Software Development Engineer II (SDE II) in Windows Core Networking at Microsoft. Madhan graduated from the University of Florida in 2008 and joined Microsoft as a developer in the Windows Core Networking team. In Windows 7, he worked on implementing network Quality of Service in the Windows networking stack. In Windows 8, he was part of the Hyper-V networking team and was responsible for improving network diagnostics in the Hyper-V environment. He also implemented features like VM QoS and IPSec task offload support for virtual machines in Windows Server 2012. His LinkedIn profile can be found at http://www.linkedin.com/in/madhansivakumar.

Mark Ghazai is a Data Center Specialist with Microsoft U.S. State and Local Government (SLG) team. His goal is to address challenging issues within SLG customer datacenters and their journey toward private and public cloud adoption. Assisting customers to get a deeper understanding of managed and consolidated datacenters powered by Windows Server 2012, Windows Server 2012 Hyper-V, Remote Desktop, VDI, and System Center 2012 suite, along with Microsoft Identity Management Solutions (FIM, UAG, TMG) is his main area of focus. Before this role, he was a Senior Premier Field Engineer (PFE) and Senior Support Escalation Engineer for several years. His TechNet blog can be found at http://blogs.technet.com/mghazai.

Nick Eales is a Senior Premier Field Engineer at Microsoft, based in Sydney, Australia. Nick has 17 years of industry experience, with the last eight of those years at Microsoft. Within Microsoft, Nick has worked on multiple teams focusing on Core Platforms support, Failover Clustering and Hyper-V, and currently is the architect for the Hyper-V Risk Assessment Program and one of the leads for the Failover Clustering Risk Assessment Program.

Shabbir Ahmed is a Partner Enterprise Architect (Infrastructure) with the Partner Enterprise Architect Team (PEAT). Shabbir helps Microsoft partners build hosting solutions. He is best in working with partners/customers to link and apply complex technologies to their business strategies and continues to be a creative thinker with high energy and enthusiasm. Apart from Microsoft Certifications he was Microsoft MVP from 2011 to 2013 and holds multiple certifications including CCIE, CEH, and ISO 27001 LA. His LinkedIn profile can be found at http://in.linkedin.com/pub/shabbir-ahmed/58/575/209.

Subhasish Bhattacharya is a Program Manager for Clustering and High Availability at Microsoft. He has worked at Microsoft at for seven years in multiple teams including High Availability and Clustering and Core Networking (DNS). His LinkedIn profile can be found at http://www.linkedin.com/pub/subhasish-bhattacharya/1/a75/b0.

Thomas Roettinger is a Program Manager in the Partner and Customer Ecosystem Team at Microsoft and works with technologies like Hyper-V and System Center Virtual Machine Manager. His team runs the Windows Server TAP Program and collects very early technology best practices. Before he joined the Product Group he was the EMEA Virtualization Lead in Microsoft Premier Field Engineering. During this time he was responsible for various services such as the Hyper-V Risk Assessment Program and the Implementing Hyper-V Workshop. He has rich experience in cloud implementations across various business segments such as hosters and enterprises. Thomas maintains a personal blog at http://blogs.technet.com/b/cloudytom and also contributes to his team blog at http://blogs.technet.com/b/wincat.

Tim Quinn is a Support Escalation Engineer on the Windows Platform Distributed Systems Networking team. He delivers reactive support for Microsoft Networking technologies such as DNS, DHCP, Remote Access, and core network connectivity, including troubleshooting of Hyper-V Network Virtualization.

Trevor Cooper-Chadwick is a Principle Consultant with Microsoft Consulting Services UK. A Subject Matter Expert in the WW Microsoft Virtualization team, he is passionate about helping customers architect and deploy highly effective infrastructure solutions leveraging both private and public cloud technologies and services. An IT veteran with many years of experience spanning Internet, Grid, and High Performance Computing, he has spent the last five years defining and building leading-edge solutions using Hyper-V, System Center Virtual Machine Manager and Azure.

About the companion content

The companion content for this book consists of a zip file containing the Windows PowerShell scripts found in certain sections of this title. This companion content can be downloaded from the following page:

http://aka.ms/TroubleshootHyper-VNetworking/files

Acknowledgments

Thanks to Anne Hamilton and Karen Szall at Microsoft Press, to Megan Smith-Creed our copy editor, and to Jean Trenary for production services.

Errata & book support

We've made every effort to ensure the accuracy of this content and its companion content. Any errors that have been reported since this content was published are listed on our Microsoft Press site:

http://aka.ms/TroubleshootHyper-VNetworking/errata

If you find an error that is not already listed, you can report it to us through the same page.

If you need additional support, email Microsoft Press Book Support at mailto:mspinput@microsoft.com.

Please note that product support for Microsoft software is not offered through the addresses above.

We want to hear from you

At Microsoft Press, your satisfaction is our top priority, and your feedback our most valuable asset. Please tell us what you think of this book at:

http://aka.ms/tellpress

The survey is short, and we read every one of your comments and ideas. Thanks in advance for your input!

Stay in touch

Let's keep the conversation going! We're on Twitter: http://twitter.com/MicrosoftPress.

Virtual switch

The Hyper-V virtual switch in Windows Server 2012 has new capabilities that can provide for tenant isolation, traffic shaping, protection against malicious virtual machines, and easier troubleshooting of issues. The virtual switch is also extensible and is built on an open platform that enables independent software vendors to add or extend the capabilities provided natively in the virtual switch. Non-Microsoft extensions can be developed that can emulate the full capabilities of hardware-based switches to allow for implementing more complex virtual environments and solutions.

The virtual switch is implemented as a layer 2 virtual network that you can use to connect virtual machines to the physical network. The virtual switch also provides policy enforcement for security, isolation, and service levels and supports Network Device Interface Specification (NDIS) filter drivers and Windows Filtering Platform (WFP) callout drivers to support non-Microsoft extensible plug-ins that can provide enhanced networking and security capabilities.

In this section, Madhan Sivakumar, a Software Development Engineer II on the Windows Core Networking team at Microsoft, explains how you can reduce network downtime using the rich diagnostics available for the Windows Server 2012 Hyper-V virtual switch.

Reducing network downtime with rich diagnostics in Hyper-V virtual switch

Imagine a situation where you have just deployed hundreds of virtual machines across different hosts and now you are getting reports that some virtual machines have lost network connectivity. This situation is not hard to imagine for most IT/network administrators since most have had to deal with this issue at some point in the past.

There could be many reasons for broken network connectivity; for example, misconfiguration, wrong placement of virtual machines, miscommunication between the network administrator and virtual machine administrator. What the administrator dreads the most is the downtime caused by broken connectivity as they wait for the support team to diagnose the issue and restore connectivity. Reducing network downtime was one of the highest priorities in developing Windows Server 2012, which gives administrators a rich set of diagnostics tools and features to quickly identify issues and fix them. This section goes over these new features and some improvements made to existing features.

System event log

When you receive an issue report, the first thing you do is look at the system event log. There are a number of error/warning events in the system event log that are logged by the provider Hyper-V vmswitch, which captures the configuration/setup errors with enough detail to help you understand the issue. Let's say you start with the system event log to diagnose virtual machine network connectivity issues and notice the following error event being logged. You'll know that virtual machine connectivity has been blocked because one of the required extensions is missing:

```
Connectivity has been blocked for NIC 32FC2EED-6AA4-4F03-8926-3C5AF80EF5A6--A610DE2F-
0B59-40B1-91C1-AB513E0F5F6E (Friendly Name: Network Adapter) on port 83805C62-C57F-4EC1-
B000-433D1914A16C (Friendly Name: ). Extension {5cbf81be-5055-47cd-9055-a76b2b4e369e} is
required on the port, but it is not active on switch EF4EE212-5D11-477C-BE86-
B131ECA4E397 (Friendly Name: ext).
```

You can make use of the new PowerShell cmdlets to get the list of switch extensions currently installed:

```
PS C:\test> Get-VMSwitchExtension ext

Id                   : 5CBF81BE-5055-47CD-9055-A76B2B4E369E
Name                 : Switch Extensibility Test Extension 2
Vendor               : Microsoft
Version              : 6.0.5019.0
ExtensionType        : Filter
ParentExtensionId    :
ParentExtensionName  :
SwitchId             : EF4EE212-5D11-477C-BE86-B131ECA4E397
SwitchName           : ext
Enabled              : False
Running              : False
ComputerName         : 27-3145J0630
IsDeleted            : False
```

Here you notice that the extension with ID 5CBF81BE-5055-47CD-9055-A76B2B4E369E from the event log is not enabled on this switch even though it is marked as mandatory for the virtual machine. Enabling this extension from PowerShell will restore network connectivity to the virtual machine.

As you can see, Windows Server 2012 logs these events with as much detail as possible so that it is easy for administrators to figure out what is going on. From this particular event log, the administrator knows which virtual machine (from the NIC and port names/friendly names) connected to which switch has connectivity issues, along with the reason for broken connectivity. This is just one example of more than 50 events that are logged to the system log by vmswitch for easy diagnosis.

Performance counters

In the above scenario, connectivity is restored to the virtual machine after the required extension network is installed. However, imagine you discover that two virtual machines connected to the same virtual switch are unable to connect to each other. If you are unable to find sufficient information in the system event log to diagnose this issue, the next step would be to launch Performance Monitor and take a look at the following counter providers:

- Hyper-V Virtual Switch
- Hyper-V Virtual Switch Port
- Hyper-V Virtual Network Adapter

For diagnosing network connectivity issues, the following counters would be of interest:

- Dropped Packets Incoming/sec
- Dropped Packets Outgoing/sec
- Extensions Dropped Packets Incoming/sec
- Extension Dropped Packets Outgoing/sec

Separate counters clearly identify where the packets are being dropped: switch or switch extensions. When you see that the Dropped Packets Incoming/sec is high, you know that there has been some misconfiguration in the switch:

\\MADHANS-TESTHP		
Hyper-V Virtual Network Adapter	TestLogicalSwitch_CCF4C0A2-B213-4A35-80B2-4D97F4A66A6F	VM1_Legacy Network Adapter_ABE31850-AE81-4DD7-BB48-7F7D51A04053--0
Dropped Packets Incoming/sec	0.000	0.000
Dropped Packets Outgoing/sec	0.000	0.995
Extensions Dropped Packets Incoming/sec	0.000	0.000
Extensions Dropped Packets Outgoing/sec	0.000	0.000

In the above example, the parent partition is unable to communicate with the virtual machine named VM1. The dropped counters of the parent partition virtual NIC is zero. However, the outgoing dropped counter of the virtual machine virtual network adapter is greater than zero. If all of the virtual NIC and switch dropped counters show zero dropped packets, it would be a good idea to examine whether the packet is getting dropped in the virtual machine itself by checking the firewall and other settings in the virtual machine OS.

Diagnostic event log and packet capture

Now, you have identified the switch is dropping outgoing packets from the virtual machine, but you don't yet know the reason. One way to determine the root cause would be to go over all the switch port configurations manually to check if you have missed something. However, this is tedious and time consuming. Since the goal is to minimize network downtime, a new Windows Server 2012 feature makes this process fast. You can use the diagnostic event log to capture Vmswitch debug events. Here is the command to start the debug channel:

```
Netsh trace start provider=Microsoft-Windows-Hyper-V-Vmswitch
```

After reproduciing the connectivity issue, stop the tracing session:

```
Netsh trace stop
```

You can open the generated ETL file using Event Viewer or Netmon (more on opening these files using Netmon later). As the packet flows through vmswitch, a number of events are being generated to trace the flow:

- When vmswitch receives the packet from the source NIC:

  ```
  NBL received from Nic CCF4C0A2-B213-4A35-80B2-4D97F4A6A46F (Friendly Name:
  TestLogicalSwitch) in switch 1C3F4C4C-47B9-4BE2-A563-F2800468D9B9 (Friendly Name:
  TestLogicalSwitch)
  ```

- When the packet is routed from the source NIC to the destination NIC(s):

  ```
  NBL routed from Nic CCF4C0A2-B213-4A35-80B2-4D97F4A6A46F (Friendly Name:
  TestLogicalSwitch) to Nic ABE31850-AE81-4DD7-BB48-7F7D51A04053--0 (Friendly Name:
  Legacy Network Adapter) on switch 1C3F4C4C-47B9-4BE2-A563-F28004
  ```

- When the packet is delivered to the destination NIC:

  ```
  NBL delivered to Nic ABE31850-AE81-4DD7-BB48-7F7D51A04053--0 (Friendly Name:
  Legacy Network Adapter) in switch 1C3F4C4C-47B9-4BE2-A563-F2800468D9B9 (Friendly
  Name: TestLogicalSwitch)
  ```

When packets are dropped in vmswitch for any reason, you'll usually see a corresponding dropped event log entry:

```
NBL originating from Nic ABE31850-AE81-4DD7-BB48-7F7D51A04053--0 (Friendly Name: Legacy
Network Adapter) was dropped in switch 1C3F4C4C-47B9-4BE2-A563-F2800468D9B9 (Friendly
Name: TestLogicalSwitch), Reason Failed Security Policy
```

For some dropped event logs, there would be another event log with more details. In the previous example, the packet was dropped because of a failed security policy, but it is unclear which security policy actually caused the drop. This event is followed by another event giving more details:

```
A packet was dropped on port 72542DDC-A517-4E70-8BB6-B33B7C409C1F (Friendly Name:
Dynamic Ethernet Switch Port) on switch 1C3F4C4C-47B9-4BE2-A563-F2800468D9B9 (Friendly
Name: TestLogicalSwitch) because the packet is filtered by Port ACL.
```

With this event, you can immediately identify why the virtual machines were unable to ping each other. These inter-virtual machine packets were dropped due to a Port ACL configured on one of the switch ports. You can identify the port where the packets were dropped by looking at the NIC/port dropped counters. At this point you just need to review the port ACLs that are set on this switch port to either fix this issue or verify that the packet was correctly dropped according to the rules.

Packet capture within vmswitch

One of the most common tools used for diagnosis is packet capture. Until the current release of Windows Server, you could not capture packets flowing within vmswitch. With the extensible virtual switch in Windows Server 2012, you can capture packets at both ingress

(when the packet enters the switch) and egress (when the packet leaves the switch). This is done through the unified tracing packet capture driver, which in Windows 8 has been updated to a switch extension. To turn on capture within vmswitch, use the following command:

```
Netsh trace start provider=Microsoft-Windows-Hyper-V-Vmswitch capture=yes,
capturetype=vmswitch
```

This will capture all packets flowing through all switches on the host. To include packet capture in the host NDIS stack, use the following:

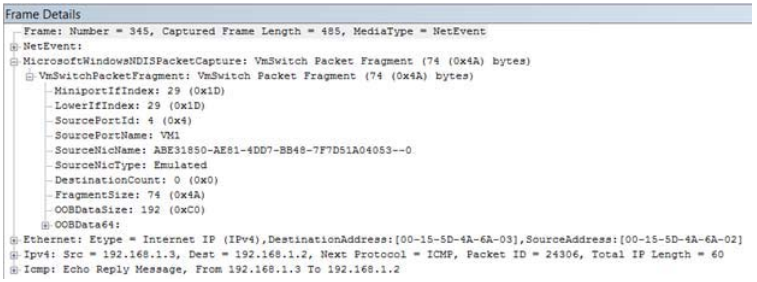
```
capturetype=both
```

To stop the tracing session and generate an ETL file, use the following command:

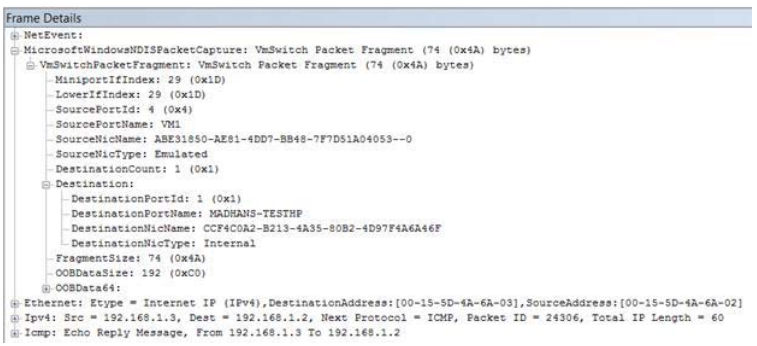
```
Netsh trace stop
```

This ETL file can be opened using Netmon. You need the parsers to view this capture (and the vmswitch events mentioned in the earlier section) using Netmon. The parsers can be downloaded from the CodePlex site at http://nmparsers.codeplex.com/releases.

The following screenshot shows capture at ingress:



This looks like any other Netmon capture. This is an ICMP Reply packet. However, this capture has additional information that helps in quicker analysis. This also captures the VM Name, Port ID, Source NIC name, and so on. The capture at egress also includes these fields along with the destination information:



In the above egress capture, the packet is being routed from VM1 to an internal virtual NIC on the host. In the case of broadcast/multicast packets, the capture will show a destination array with information about each destination in the array.

I hope that these new features will help you diagnose issues faster and more easily, thereby reducing the network downtime for virtual machines and the host.

—Madhan Sivakumar, Software Development Engineer II, Windows Core Networking

Additional resources

Here are a few additional resources concerning this topic:

- Hyper-V Virtual Switch Overview (TechNet Library) at: http://technet.microsoft.com/en-us/library/hh831823.aspx
- Hyper-V: Virtual Networking Survival Guide (TechNet Wiki) at: http://social.technet.microsoft.com/wiki/contents/articles/151.hyper-v-virtual-networking-survival-guide.aspx
- Hyper-V Access Control Lists (ACLs) (TechNet Library) at: http://technet.microsoft.com/en-us/library/jj679878.aspx#bkmk_portacls