**Last Update:** 7/3/2018

RS5 Validation Guide – RSC in the vSwitch

**Summary:** This document provides implementation steps to validate time Receive Segment Coalescing (RSC) in the vSwitch on Server 2019.

Table of Contents

[Description 3](#_Toc513043194)

[Required Configuration 3](#_Toc513043195)

[Troubleshooting and Feedback 5](#_Toc513043196)

[Test Activities 5](#_Toc513043197)

[Prerequisites 5](#_Toc513043198)

[Possible Architectures 6](#_Toc513043199)

[Activities 7](#_Toc513043200)

[Activity 1: Validate Default Configuration 7](#_Toc513043201)

[Task 1: Review the Global Offloads configuration 7](#_Toc513043202)

[Task 2: Review the default configuration for the vSwitch 7](#_Toc513043203)

[Task 3: Review the RSC Adapter Status 7](#_Toc513043204)

[Activity 2: Generate Traffic with RSC in the vSwitch Enabled 8](#_Toc513043205)

[Task 1: Download and Extract NTttcp 8](#_Toc513043206)

[Task 2: Configure Sender Firewall Rule 8](#_Toc513043207)

[Task 3: Configure Receiver Firewall Rule 8](#_Toc513043208)

[Task 4: Setup Receiving Session 9](#_Toc513043209)

[Task 5: Initiate Traffic Generation 9](#_Toc513043210)

[Task 6: Capture Measurements 10](#_Toc513043211)

[Activity 3: Generate Traffic with RSC in the vSwitch Disabled 10](#_Toc513043212)

[Task 1: Disable RSC in the vSwitch 10](#_Toc513043213)

[Task 2: Setup Receiving Session 11](#_Toc513043214)

[Task 3: Initiate Traffic Generation 11](#_Toc513043215)

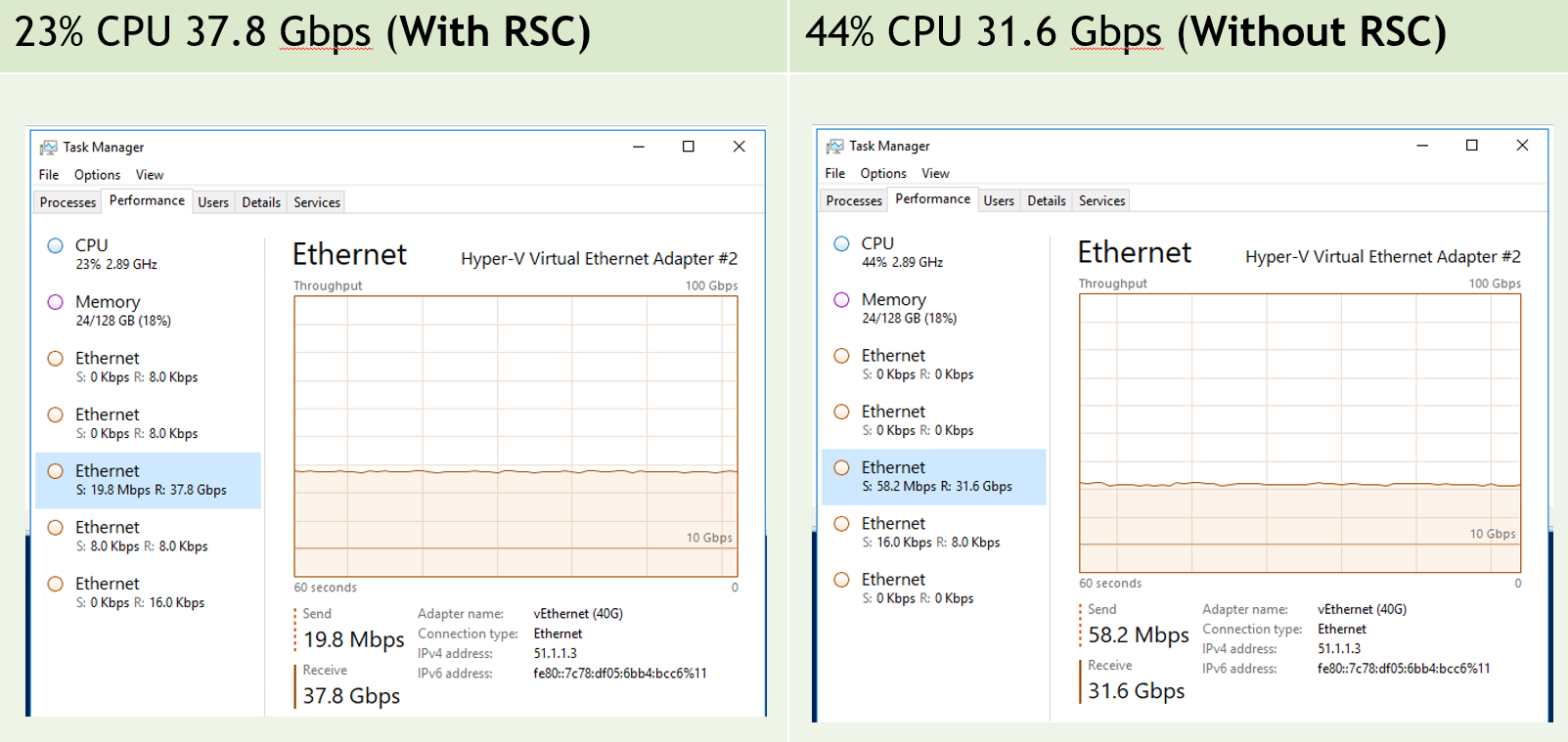
[Task 4: Capture Measurements 12](#_Toc513043216)

# Description

In Windows Server 2019, we have continued driving performance improvements for Hyper-V customers. With Receive Segment Coalescing (RSC) in the vSwitch, we’ve taught an old feature a new trick enabling segment coalescing at the virtual switch level.

This allows you to coalesce segments even if your physical hardware does not support RSC and provides support for virtual machines to receive coalesced segments. Prior to Windows Server 2019, RSC was not supported for traffic traversing a virtual switch.

With Receive Segment Coalescing in the vSwitch you can decrease the CPU utilization as shown below and increase throughput:



Additional Notes:

* TCP Segments specifically; Other transports are not supported
* Can be enabled or disabled on the fly without disrupting existing connections

# Required Configuration

This scenario requires the following hardware and software features for successful completion:

|  |  |
| --- | --- |
| Hardware | Description |
| Windows Server 2019 (RS5) Hyper-V Host. This is the receiving system (Receiver). | This system must be able to run Hyper-V and must be configured with a vSwitch and a host vNIC (Switch Embedded Teaming (SET) is supported).  The host vNIC is the receiving NIC where RSC in the vSwitch will be toggled. |
| Windows Server or Windows Client. This system is the traffic generation system (Sender) | This additional system will be used to send traffic to the host vNIC  This system is the sending NIC. |
| Testing Tools | This guide leverages NTttcp, a free network traffic generation and measurement tool. You are free to use your own tools if you prefer. |

# Troubleshooting and Feedback

If you encounter any challenges during your validation, please use one of the contact methods identified on the title page of this document.

Also, please submit your feedback! Our most actionable feedback regularly comes through the TAP and MVP communities and increases the quality of each feature that you provide feedback on!

We have attempted to capture general questions that we would like feedback on, however all constructive feedback is good feedback 😊 Please let us know if you have other recommendations/feedback outside of the questions included.

# Test Activities

Below are the activities that will support our validation:

1. Validate the default configuration of Windows Server 2019; that the feature reports as on and data flows through the virtual switch to a Hyper-V virtual machine.
2. Use a network traffic generation tool (in our case NTttcp) to take initial measurements.
3. Turn off RSC in the vSwitch and re-capture and compare measurements.

# Prerequisites

This section outlines the prerequisites needed to complete the activities.

* Configure Windows Server 2019 (RS5) Hyper-V Host as the Receiver
  + Redstone 5 server builds will be the latest Long-Term Servicing Branch (LTSC) and is now known as Windows Server 2019 – For purposes of this document, this terminology will be used interchangeably.
  + The Receiver must have a Hyper-V switch configured.
    - RSC in the vSwitch works well with an IOV enabled vSwitch and/or Switch Enabled Teaming (SET). You may use the same vSwitch to validate other High Performance Networking features such as Guest RDMA or Dynamic Virtual Machine Multi-Queue (d.VMMQ).
    - To create a vSwitch on the Receiver:
    - You may use this PowerShell command as an example to create the vSwitch

**New-VMSwitch -Name 'VMSTest' -AllowManagementOS $true**

* + - Make certain to **AllowManagementOS $true** to create a host v NIC
    - Create an IP Address on the host vNIC created for the vSwitch
* Additional Windows Server (Sender)
  + Configure at least one additional system
  + This system will be used to send traffic to the Reciver and subsequently does not require any special features or configuration.
* Tools
  + This guide leverages a network traffic generation tools called NTttcp. This tool will be needed on both the sending and receiving system.
  + <https://gallery.technet.microsoft.com/NTttcp-Version-528-Now-f8b12769/file/159655/1/NTttcp-v5.33.zip>

# Possible Architectures

To test this feature, two physical systems are required:

**Sender** is a separate physical system. The **Receiver** is a Windows Server 2019 Hyper-V Host with a vNIC in the parent partition. Since the host vNIC sits above the virtual switch, this scenario is acceptable.

Sender

192.168.24.101

Receiver

Hyper-V Host vNIC

192.168.24.102

Altneratively, you may also use a virtual machine as the receiver. **Sender** is still a separate physical system. The **Receiver** is a Windows Server 2019 virtual machine with a vmNIC.

Sender

192.168.24.100

Hyper-V Host

192.168.24.101

Receiver

Virtual Machine

192.168.24.102

# Activities

## Activity 1: Validate Default Configuration

This activity will validate the default configuration of RSC in the vSwitch.

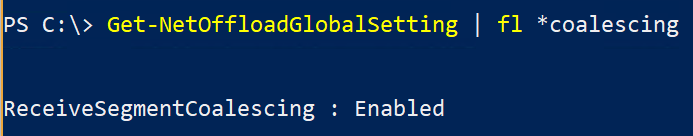
* Task 1 – Review the Global Configuration
* Task 2 – Review the default configuration for the vSwitch
* Task 3 – Review the RSC Adapter Status

### Task 1: Review the Global Offloads configuration

In this task, we will validate the default system-wide configuration. System-wide **ReceiveSegmentCoalescing** should report enabled (this is the default).

At an elevated PowerShell prompt, run the following command:

**Get-NetOffloadGlobalSetting | FL \*coalescing**

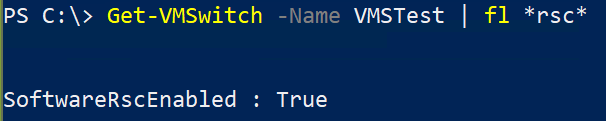


### Task 2: Review the default configuration for the vSwitch

In this task, we will validate the default vSwitch configuration. The vSwitch’s default configuration of **SoftwareRscEnabled** should be True.

At an elevated PowerShell prompt, run the following command (note, the vmswitch name may be different in your scenario):

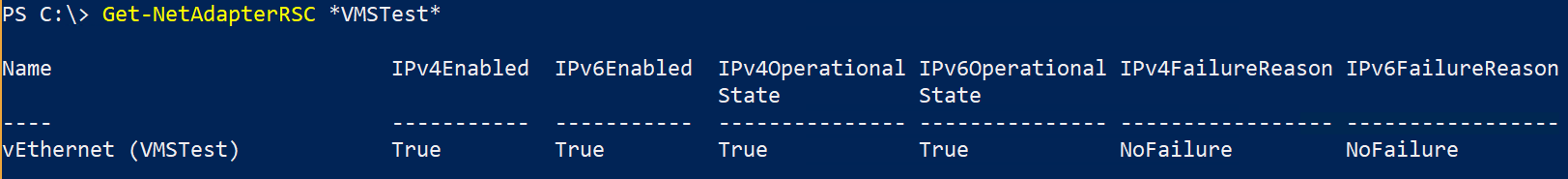
**Get-VMSwitch -Name VMSTest | FL \*RSC\***



### Task 3: Review the RSC Adapter Status

In this task, we will validate the adapter status for RSC. If any adapter will not function with RSC, it will be reflected here. It is important to understand that only adapters with TCP/IP bound will report an appropriate status.

At an elevated PowerShell prompt, run the following command (note, the vmswitch name may be different in your scenario):



**Questions:**

* Did the **IPv4FailureReason** or **IPv6FailureReason** report a failure?
* Were the messages clear?

## Activity 2: Generate Traffic with RSC in the vSwitch Enabled

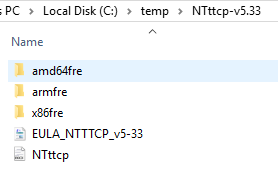
This activity will use [NTttcp](https://gallery.technet.microsoft.com/NTttcp-Version-528-Now-f8b12769/file/159655/1/NTttcp-v5.33.zip) to generate traffic from the sending system, to the receiving system and capture measurements with RSC in the vSwitch **enabled**.

* Task 1 – Download and Extract NTttcp
* Task 2 – Configure Sender Firewall Rules
* Task 3 – Configure Receiver Firewall Rules
* Task 4 – Setup Receiving Session
* Task 5 – Initiate Traffic Generation
* Task 6 – Capture Measurements

### Task 1: Download and Extract NTttcp

**Note:** This task should be completed on both the sending and receiving systems.

Download [NTttcp](https://gallery.technet.microsoft.com/NTttcp-Version-528-Now-f8b12769/file/159655/1/NTttcp-v5.33.zip) and extract the contents in a directory on the sending and receiving systems. Open an elevated PowerShell prompt and navigate to your system’s architecture folder. For example, the extracted contents of the NTttcp should look like this:



And you should navigate the PowerShell prompt to this:



### Task 2: Configure Sender Firewall Rule

NTttcp will be blocked by the Windows (or other) firewall if a rule is not in-place to allow the traffic. Since Ntttcp can use different ports, you should create a firewall rule that allows the **program** to run. This must be done on the **both** the sending and receiving systems.

For example, if NTttcp was extracted to c:\temp, you could use this command to create an exception:

New-NetFirewallRule -DisplayName 'NTTCP' -Name 'NTTCP' `

-Program 'c:\temp\ntttcp-v5.33\amd64fre\ntttcp.exe' `

-Direction Outbound -Action Allow `

**Note:** The command above is a single command that stretches to multiple lines due to the size of the page.

### Task 3: Configure Receiver Firewall Rule

NTttcp will be blocked by the Windows (or other) firewall if a rule is not in-place to allow the traffic. Since Ntttcp can use different ports, you should create a firewall rule that allows the **program** to run. This must be done on the **both** the sending and receiving systems.

For example, if NTttcp was extracted to c:\temp, you could use this command to create an exception:

New-NetFirewallRule -DisplayName 'NTTCP' -Name 'NTTCP' `

-Program 'c:\temp\ntttcp-v5.33\amd64fre\ntttcp.exe' `

-Direction Inbound -Action Allow

**Note:** The command above is a single command that stretches to multiple lines due to the size of the page.

### Task 4: Setup Receiving Session

Run the following command to allow the receiver to receive traffic from the sender.

**.\NTttcp.exe -R -M 1,\*,192.168.24.102 -l 64k -a 6 -t 30**

Note the following from the above command:

* NTttcp.exe exists in the current location of the prompt
* **-R** identifies this as the receiving system
* **-M** indicates the number of threads to be used – in this example, 1 thread will be used
* **\*** indicates the processors to use – In this example, all processors can be used
* **192.168.24.102** indicates the receiving systems IP address
* **-t 30** indicates the (length in seconds) of the test

Additional, information on the commands are available in the NTttcp documentation. Please feel free to alter the exact commands for the test based on your environment.

### Task 5: Initiate Traffic Generation

Run the following command to initiate traffic from the sender.

**.\NTttcp.exe -S -M 1,\*,192.168.24.102 -l 64k -a 6 -t 30**

Note the following from the above command:

* NTttcp.exe exists in the current location of the prompt
* **-S** identifies this as the sending system
* **-M** indicates the number of threads to be used – in this example, 1 thread will be used
* **\*** indicates the processors to use – In this example, all processors can be used
* **192.168.24.102** indicates the receiving systems IP address– This should be the same on the sending and receiving systems
* **-t 30** indicates the (length in seconds) of the test

Additional, information on the commands are available in the NTttcp documentation. Please feel free to alter the exact commands for the test based on your environment.

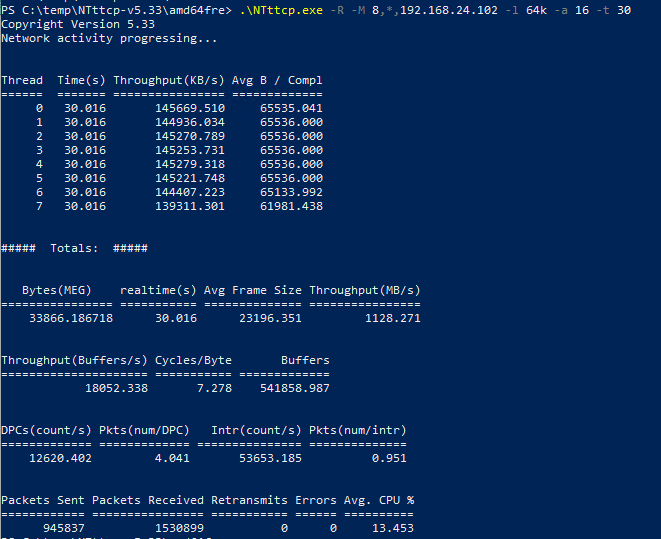
Once this command has been executed traffic should begin to flow. This can be identified by the sending and receiving systems showing the following output:



### Task 6: Capture Measurements

After 30 seconds (**-t 30**) the command will return and display results. RSC functions on the receive side and coalesces data prior to delivery to the operating system. As such, we are only interested in the metrics from the receiver.

The entire output is included here:



Review the **Avg. CPU%**, **Throughput (MB/s)**, and **Cycles/Byte** from the receiver’s output

## Activity 3: Generate Traffic with RSC in the vSwitch Disabled

This activity will generate traffic from the sending system, to the receiving system and capture measurements with RSC in the vSwitch **disabled**.

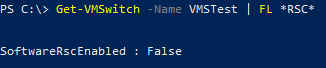
* Task 1 – Disable RSC in the vSwitch
* Task 2 – Setup Receiving Session
* Task 3 – Initiate Traffic Generation
* Task 4 – Capture Measurements

### Task 1: Disable RSC in the vSwitch

To disable RSC in the vSwitch, on the receiving system, run the following PowerShell command:



Verify this has been disabled with the following command:



### Task 2: Setup Receiving Session

In this task, you’ll setup the receiving session as you did in Activity 2. The commands are included here for simplicity. Run the following command to allow the receiver to receive traffic from the sender.

**.\NTttcp.exe -R -M 1,\*,192.168.24.102 -l 64k -a 6 -t 30**

Note the following from the above command:

* NTttcp.exe exists in the current location of the prompt
* **-R** identifies this as the receiving system
* **-M** indicates the number of threads to be used – in this example, 1 thread will be used
* **\*** indicates the processors to use – In this example, all processors can be used
* **192.168.24.102** indicates the receiving systems IP address
* **-t 30** indicates the (length in seconds) of the test

Additional, information on the commands are available in the NTttcp documentation. Please feel free to alter the exact commands for the test based on your environment.

### Task 3: Initiate Traffic Generation

In this task, you’ll setup the receiving session as you did in Activity 2. The commands are included here for simplicity. Run the following command to initiate traffic from the sender.

**.\NTttcp.exe -S -M 1,\*,192.168.24.102 -l 64k -a 6 -t 30**

Note the following from the above command:

* NTttcp.exe exists in the current location of the prompt
* **-S** identifies this as the sending system
* **-M** indicates the number of threads to be used – in this example, 1 thread will be used
* **\*** indicates the processors to use – In this example, all processors can be used
* **192.168.24.102** indicates the receiving systems IP address– This should be the same on the sending and receiving systems
* **-t 30** indicates the (length in seconds) of the test

Additional, information on the commands are available in the NTttcp documentation. Please feel free to alter the exact commands for the test based on your environment.

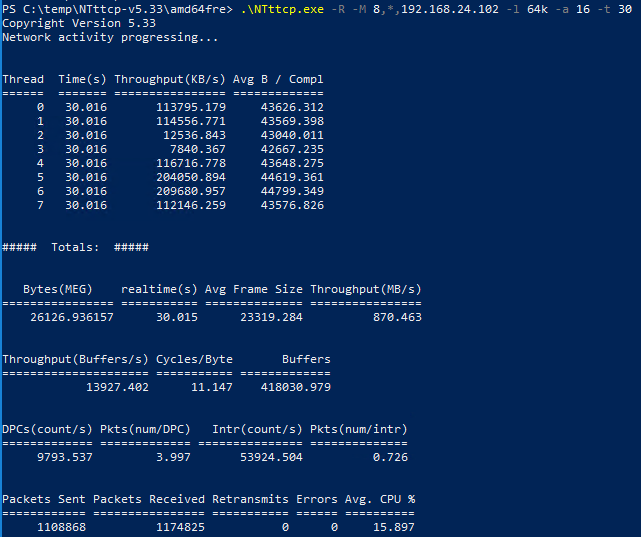
Once this command has been executed traffic should begin to flow. This can be identified by the sending and receiving systems showing the following output:



### Task 4: Capture Measurements

After 30 seconds (**-t 30**) the command will return and display results. RSC functions on the receive side and coalesces data prior to delivery to the operating system. As such, we are only interested in the metrics from the receiver.

The entire output is included here:



Review the **Avg. CPU%**, **Throughput (MB/s)**, and **Cycles/Byte** from the receiver’s output and compare against the receiver’s output while RSC in the vSwitch was enabled. The information from the example in this document is capture in the table below for comparison. This example was with a 10GbE Adapters. As you increase to faster adapters (25/40/100GbE) you see greater and greater gains.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Avg. CPU %** | **Cycles/Byte** | **Throughput (MB/s)** |
| **RSC in the vSwitch Enabled** | 13.453 | 7.278 | 1128.271 |
| **RSC in the vSwitch Disabled** | 15.897 | 11.147 | 870.463 |
| Improvement with RSC in the vSwitch Enabled | | | |
|  | CPU Utilization lowered by ~15% | ~35% less cycles/byte | ~23% greater throughput (at lower CPU cost) |

**Questions**:

* Were you able to observe lower CPU utilization for the same workloads when RSC in the vSwitch was enabled?
* Overall, did your performance improve?

Note: As previously mentioned, performance gains increase at higher bandwidths. For example, a 40GbE NIC will see greater advantages than a 10GbE NIC.

**Summary:**

Thank you for validating Receive Segment Coalescing in the vSwitch. Please make sure to submit your feedback!