User Guide for NIC Teaming (LBFO)

A Guide to Windows Server 2016 Technical Review NIC Teaming for the novice and the expert.

*Version: May 20, 2016 update*

# NIC Teaming

NIC teaming, also known as Load Balancing/Failover (LBFO), allows multiple network adapters to be placed into a team for the purposes of

* bandwidth aggregation, and/or
* traffic failover to maintain connectivity in the event of a network component failure.

This feature has long been available from NIC vendors but until Windows Server 2012 NIC teaming was not included with Windows Server.

The following sections address:

* NIC teaming architecture
* Bandwidth aggregation (also known as load balancing) mechanisms
* Failover algorithms
* NIC feature support – stateless task offloads and more complex NIC functionality
* A detailed walkthrough how to use the NIC Teaming management tools

There are two versions of NIC teaming in Windows Server 2016:

1. The stand-alone NIC Teaming solution, i.e., the NIC Teaming solution that was available in Windows Server 2012 and Windows Server 2012 R2, is available in Windows Server 2016 in ServerCore and full Server versions.
2. A new teaming solution integrated with the Hyper-V switch and known as Switch-Embedded Teaming (SET), is available in all SKUs of Windows Server 2016 including Nano.

NIC teaming is not available in any client operating systems such as Windows 10, however the NIC teaming User Interface in the Remote Server Administration Tools (RSAT) and the NIC Teaming Windows PowerShell Cmdlets can both be run on Windows 10 so that a personal computer running Windows 10 can be used to manage the stand-alone NIC teaming solution on one or more servers running Windows Server 2012, Windows Server 2012 R2, or Windows Server 2016.

Bluetooth®, Infiniband®, and other trademarks throughout this document are the property of their respective owners. Hyper-V® and Windows® are trademarks of Microsoft Corporation.

# Contents, Figures, Tables, and Glossary

## Table of Contents

[1 NIC Teaming 1](#_Toc451511043)

[2 Contents, Figures, Tables, and Glossary 3](#_Toc451511044)

[2.1 Table of Contents 3](#_Toc451511045)

[2.2 List of Tables 5](#_Toc451511046)

[2.3 List of Figures 5](#_Toc451511047)

[2.4 Glossary 6](#_Toc451511048)

[3 Technical Overview 8](#_Toc451511049)

[3.1 Traditional architectures for NIC teaming 8](#_Toc451511050)

[3.2 Architecture for NIC Teaming in Windows Server 2016 9](#_Toc451511051)

[3.3 Configurations for NIC Teaming 10](#_Toc451511052)

[3.4 Algorithms for load distribution 12](#_Toc451511053)

[3.5 Interactions between Configurations and Load distribution algorithms 13](#_Toc451511054)

[3.5.1 Switch Independent configuration / Address Hash distribution 13](#_Toc451511055)

[3.5.2 Switch Independent configuration / Hyper-V Port distribution 14](#_Toc451511056)

[3.5.3 Switch Independent configuration / Dynamic distribution 14](#_Toc451511057)

[3.5.4 Switch Dependent configuration / Address Hash distribution 15](#_Toc451511058)

[3.5.5 Switch Dependent configuration / Hyper-V Port distribution 15](#_Toc451511059)

[3.5.6 Switch Dependent configuration / Dynamic distribution 15](#_Toc451511060)

[3.6 NIC teaming inside of Virtual Machines (VMs) 16](#_Toc451511061)

[3.7 Hyper-V ports in the Host Partition 18](#_Toc451511062)

[3.8 Feature compatibilities 18](#_Toc451511063)

[3.8.1 Stand-alone NIC Teaming and Virtual Machine Queues (VMQs) 20](#_Toc451511064)

[3.8.2 Hyper-V Network Virtualization (HNVv1) / NV-GRE compatibility 22](#_Toc451511065)

[3.8.3 Hyper-V Network Virtualization – SDN (HNVv2) 22](#_Toc451511066)

[3.9 NIC Requirements and limitations 22](#_Toc451511067)

[3.9.1 Number of NICs in a team in a native host 22](#_Toc451511068)

[3.9.2 Number of NICs in a team in a Hyper-V VM 23](#_Toc451511069)

[3.9.3 Types of NICs in a team 23](#_Toc451511070)

[3.9.4 Number of team interfaces for a team 23](#_Toc451511071)

[3.10 Teaming of different speed NICs 23](#_Toc451511072)

[3.10.1 Stand-alone NIC Teaming 23](#_Toc451511073)

[3.10.2 Switch-embedded teaming 24](#_Toc451511074)

[3.11 Teams of teams 24](#_Toc451511075)

[3.12 MAC address use and MAC address management 24](#_Toc451511076)

[3.12.1 The MAC address of the team 24](#_Toc451511077)

[3.12.2 MAC address use on transmitted packets 25](#_Toc451511078)

[3.13 Industry terms for NIC Teaming 26](#_Toc451511079)

[3.14 Troubleshooting (The dangers of using a powerful tool) 26](#_Toc451511080)

[3.14.1 Using VLANs 26](#_Toc451511081)

[3.14.2 Interactions with other teaming solutions 28](#_Toc451511082)

[3.14.3 MAC address conflicts 29](#_Toc451511083)

[3.14.4 Physical network segmentation 29](#_Toc451511084)

[3.14.5 Hardware that doesn’t conform to specification 29](#_Toc451511085)

[3.14.6 Physical switch security features 30](#_Toc451511086)

[3.14.7 Disabling and Enabling with Windows PowerShell 30](#_Toc451511087)

[4 Managing NIC Teaming 30](#_Toc451511088)

[4.1 Stand-alone NIC Teaming 30](#_Toc451511089)

[4.1.1 Invoking the Management UI for NIC Teaming 31](#_Toc451511090)

[4.1.2 The components of the NIC Teaming Management UI 33](#_Toc451511091)

[4.1.3 Adding a server to be managed 36](#_Toc451511092)

[4.1.4 Removing a server from the managed servers list 37](#_Toc451511093)

[4.1.5 Creating a team 37](#_Toc451511094)

[4.1.6 Checking the status of a team 39](#_Toc451511095)

[4.1.7 Modifying a team 40](#_Toc451511096)

[4.1.8 Deleting a team 49](#_Toc451511097)

[4.1.9 Viewing statistics for a team or team member 50](#_Toc451511098)

[4.2 Managing a switch-embedded team 52](#_Toc451511099)

[4.2.1 Creating a switch-embedded team 52](#_Toc451511100)

[4.2.2 Adding or removing a member of a switch-embedded team 53](#_Toc451511101)

[4.2.3 Removing a switch-embedded team 53](#_Toc451511102)

[4.2.4 Changing the load distribution algorithm of a switch-embedded team 53](#_Toc451511103)

[4.2.5 Forcing the affinity of a VM or vNIC to a physical NIC 53](#_Toc451511104)

[5 Evolution of Windows Server NIC Teaming 54](#_Toc451511105)

[5.1 NIC Teaming difference between Windows Server 2016 and Windows Server 2012 R2 54](#_Toc451511106)

[5.1.1 Stand-alone NIC Teaming 54](#_Toc451511107)

[5.1.2 Switch Embedded Teaming (SET) 54](#_Toc451511108)

[5.1.3 Comparison of features and feature compatibility between stand-along NIC Teaming and SET 55](#_Toc451511109)

[5.2 NIC Teaming differences between Windows Server 2012 R2 and Windows Server 2012 55](#_Toc451511110)

[6 Frequently asked questions (FAQs) 56](#_Toc451511111)

[7 Power User tips for the NIC Teaming User Interface (stand-alone NIC Teaming) 59](#_Toc451511112)

## List of Tables

[Table 1 - Feature interactions with NIC teaming 18](#_Toc416424748)

[Table 2 - VMQ mode by Teaming mode and Load distribution mode 20](#_Toc416424749)

## List of Figures

[Figure 1 - Standard NIC teaming solution architecture and Microsoft vocabulary 7](#_Toc416424750)

[Figure 2 - Stand-alone NIC Teaming supporting Hyper-V and also supporting non-Hyper-V applications 8](https://microsoft-my.sharepoint.com/personal/donst_microsoft_com/Documents/LBFO/User's%20Guide/LBFO%20User's%20Guide%20WS-TR2015.docx#_Toc416424751)

[Figure 3 - Switch-embedded Teaming supporting host applications and VMs 8](#_Toc416424752)

[Figure 4 - NIC Teaming in a VM with SR-IOV with two VFs 16](https://microsoft-my.sharepoint.com/personal/donst_microsoft_com/Documents/LBFO/User's%20Guide/LBFO%20User's%20Guide%20WS-TR2015.docx#_Toc416424753)

[Figure 5 - Enabling VM NIC Teaming in Hyper-V Manager 17](https://microsoft-my.sharepoint.com/personal/donst_microsoft_com/Documents/LBFO/User's%20Guide/LBFO%20User's%20Guide%20WS-TR2015.docx#_Toc416424754)

[Figure 6 - VLAN misconfiguration (stand-alone NIC Teaming) 26](#_Toc416424755)

[Figure 7 – NIC Teaming Windows PowerShell Cmdlets 30](#_Toc416424756)

[Figure 8 - PowerShell Get-Help 30](https://microsoft-my.sharepoint.com/personal/donst_microsoft_com/Documents/LBFO/User's%20Guide/LBFO%20User's%20Guide%20WS-TR2015.docx#_Toc416424757)

[Figure 9 - Invoking the UI from Server Manager Local Server screen 31](#_Toc416424758)

[Figure 10- Invoking the UI from Server Manager All Servers screen 31](#_Toc416424759)

[Figure 11- Invoking the UI from a Windows PowerShell prompt 32](#_Toc416424760)

[Figure 12- Invoking the UI from a Command Prompt 32](#_Toc416424761)

[Figure 13 - the NIC Teaming Management UI tiles 33](#_Toc416424762)

[Figure 14 - Column Chooser menus 34](#_Toc416424763)

[Figure 15 – Tasks menus and Right-click action menus 35](#_Toc416424764)

[Figure 16 - Team Interfaces Tasks and Right-click Action Menu 35](#_Toc416424765)

[Figure 17 - New Team dialog box 36](#_Toc416424766)

[Figure 18 - New Team dialog box with Additional Properties expanded 37](#_Toc416424767)

[Figure 19 - Team with a faulted member 39](#_Toc416424768)

[Figure 20 - Modifying Team Properties 40](#_Toc416424769)

[Figure 21 - Modifying a team's Teaming mode, Load distribution mode, and Standby Adapter 41](#_Toc416424770)

[Figure 22 - Selecting Add Interface 44](#_Toc416424771)

[Figure 23 - New team interface dialog box 44](#_Toc416424772)

[Figure 24 - Team Interface tab after creating new team interface 45](#_Toc416424773)

[Figure 25 - Selecting a team interface to change the VLAN ID 46](#_Toc416424774)

[Figure 26- Network Adapter Properties dialog box for team interfaces 47](#_Toc416424775)

[Figure 27 - Deleting a team 48](#_Toc416424776)

[Figure 28- Statistics information for teams and team members 49](#_Toc416424777)

[Figure 29- Statistics information for teams and team interfaces 50](#_Toc416424778)

[Figure 30 - General settings dialog box 51](#_Toc416424779)

## Glossary

Some terms and acronyms used in this document may not be familiar to the reader. The following table should assist the reader’s understanding. See also Section 3.13.

|  |  |
| --- | --- |
| **Term/Acronym** | **Definition or Expansion** |
| LACP | Link Aggregation Control Protocol. See section 3.3 |
| NIC | Network Interface Card. In the current world of multi-port cards often used to mean Network Interface (i.e., an Ethernet physical connection). By extension any software emulation of a network interface structure in Windows, hence vNIC, tNIC, vmNIC, etc. |
| SET | Switch Embedded Teaming. In Windows Server 2016 an alternative NIC Teaming technology was released where the NIC Teaming logic is integrated into the Hyper-V switch. |
| RSS | Receive Side Scaling. A feature in Windows that spreads incoming packet processing across multiple processors. |
| tNIC | A network interface exposed by the NIC Teaming module. Also known as Team Interface or Team NIC. |
| VLAN | Virtual Local Area Network. A method for carrying traffic for different sets of traffic (users) in a way that makes it inaccessible to other traffic (users). VLANs are indicated by a number between 0 and 4094 in a field in the Ethernet MAC header. To be more precise, IEEE 802.1Q defines a 12-bit field specifying the VLAN to which the frame belongs. The hexadecimal values of 0 (0x000) and 4095 (0xFFF) are reserved. All other values may be used as VLAN identifiers, allowing up to 4,094 VLANs. The reserved value VLAN=0 indicates that the frame does not belong to any VLAN and is the equivalent of traffic without the 802.1Q header present (untagged packets). |
| VM | Virtual Machine |
| vmNIC | A port on the Hyper-V switch exposed as a networking interface in a VM |
| VMQ | Virtual Machine Queue(s). A feature in Windows that aggregates all the incoming packet traffic destined for a particular Hyper-V switch port into a single queue and associates the processing of that queue to a specific processor. This provides workload distribution across processors in a Hyper-V host. |
| vNIC | A port on the Hyper-V switch exposed as a networking interface in the host partition |

# Technical Overview

## Traditional architectures for NIC teaming

Virtually all NIC teaming solutions on the market have an architecture similar to that shown in Figure 1.

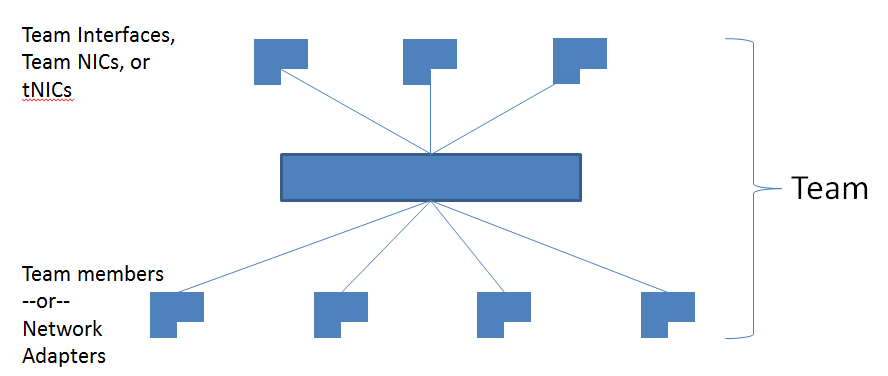


Figure 1 - Standard NIC teaming solution architecture and Microsoft vocabulary

One or more physical NICs are connected into the NIC teaming solution common core, which then presents one or more virtual adapters (*team NICs* [tNICs] or *team interfaces*) to the operating system. There are a variety of algorithms that distribute outbound traffic (load) between the NICs.

The only reason to create multiple team interfaces is to logically divide inbound traffic by virtual LAN (VLAN). This allows a host to be connected to different VLANs at the same time. *When a team is connected to a Hyper-V switch all VLAN segregation should be done in the Hyper-V switch instead of in the NIC Teaming software.*

This traditional architecture provides a solution that can operate on all versions of Windows Server 2016 whether Hyper-V services are deployed or a bare-metal server is being used. This NIC Teaming technology will be referred to as stand-alone NIC Teaming for the rest of this guide.

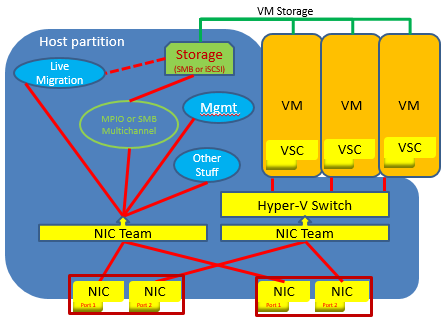


Figure - Stand-alone NIC Teaming supporting Hyper-V and also supporting non-Hyper-V applications

## Architecture for NIC Teaming in Windows Server 2016

With this release of Windows Server an alternative NIC Teaming solution has been provided for environments where Hyper-V is installed and the Software Defined Networking stack (SDN-stack) is being used. This solution integrates the teaming logic into the Hyper-V switch. This technology will be referred to as Switch-Embedded Teaming (SET) for the rest of this guide.

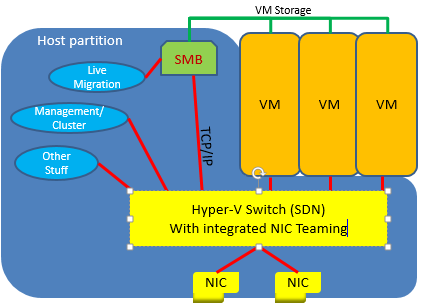


Figure 3 - Switch-embedded Teaming supporting host applications and VMs

## Configurations for NIC Teaming

There are two basic configurations for NIC Teaming.

* **Switch-independent teaming**. This configuration does not require the switch to participate in the teaming. Since in switch-independent mode the switch does not know that the network adapter is part of a team in the host, the adapters may be connected to different switches. Switch independent modes of operation do not require that the team members connect to different switches; they merely make it possible.
  + **Active/Standby Teaming[[1]](#footnote-2)**: Some administrators prefer not to take advantage of the bandwidth aggregation capabilities of NIC Teaming. These administrators choose to use one or more team members for traffic (active) and one team member to be held in reserve (standby) to come into action if an active team member fails. To use this mode set the team to Switch-independent teaming mode and then select a standby team member through the management tool you are using. Active/Standby is not required to get fault tolerance; **fault tolerance is always present** anytime there are at least two network adapters in a team. Furthermore, in any Switch Independent team with at least two members, Windows NIC Teaming allows one adapter to be marked as a standby adapter. That adapter will not be used for outbound traffic unless one of the active adapters fails. Inbound traffic (e.g., broadcast packets) received on the standby adapter will be delivered up the stack. At the point that all team members are restored to service the standby team member will be returned to standby status.  
     Once a standby member of a team is connected to the network all network resources required to service traffic on the member are in place and active. Customers will see better network utilization and lower latency by operating their teams with all team members active. Failover, i.e., redistribution of traffic across the remaining healthy team members, will occur anytime one or more of the team members reports an error state exists.
* **Switch-dependent teaming.** This configuration requires the switch to participate in the teaming. Switch dependent teaming requires all the members of the team to be connected to the same physical switch.[[2]](#footnote-3) There are two modes of operation for switch-dependent teaming:
  + **Generic or static teaming (IEEE 802.3ad draft v1).** This mode requires configuration on both the switch and the host to identify which links form the team. Since this is a statically configured solution there is no additional protocol to assist the switch and the host to identify incorrectly plugged cables or other errors that could cause the team to fail to perform. This mode is typically supported by server-class switches.
  + **Link Aggregation Control Protocol teaming (IEEE 802.1ax, LACP).** This mode is also commonly referred to as IEEE 802.3ad as it was developed in the IEEE 802.3ad committee before being published as IEEE 802.1ax.[[3]](#footnote-4) IEEE 802.1ax works by using the Link Aggregation Control Protocol (LACP) to dynamically identify links that are connected between the host and a given switch. This enables the automatic creation of a team and, in theory but rarely in practice, the expansion and reduction of a team simply by the transmission or receipt of LACP packets from the peer entity. Typical server-class switches support IEEE 802.1ax but most require the network operator to administratively enable LACP on the port.[[4]](#footnote-5) Windows NIC Teaming always operates in LACP’s Active mode with a short timer. No option is presently available to modify the timer or change the LACP mode.

Both of these modes allow both inbound and outbound traffic to approach the practical limits of the aggregated bandwidth because the pool of team members is seen as a single pipe.

Because the switch determines the inbound load distribution it is important to research what options may be available for inbound load distribution management. Many switches only use destination IP address to team member mapping which may result in a less granular distribution than is needed to get good inbound load distribution. Since this guide can’t cover all the settings on all switches it remains an exercise for the reader to understand the capabilities of the adjacent network switches.

In Windows Server 2016

* Stand-alone NIC Teaming supports all these modes;
* Switch-embedded NIC Teaming supports Switch Independent mode with no standby team members.

## Algorithms for load distribution

Outbound traffic can be distributed among the available links in many ways. One rule that guides any distribution algorithm is to try to keep all packets associated with a single flow (TCP-stream) on a single network adapter. This rule minimizes performance degradation caused by reassembling out-of-order TCP segments.

Stand-alone NIC teaming supports the following traffic load distribution algorithms:

* **Hyper-V switch port.** Since VMs have independent MAC addresses, the VM’s MAC address or the port it’s connected to on the Hyper-V switch can be the basis for dividing traffic. There is an advantage in using this scheme in virtualization. Because the adjacent switch always sees a particular MAC address on one and only one connected port, the switch will distribute the ingress load (the traffic from the switch to the host) on multiple links based on the destination MAC (VM MAC) address. This is particularly useful when Virtual Machine Queues (VMQs) are used as a queue can be placed on the specific NIC where the traffic is expected to arrive. However, if the host has only a few VMs, this mode may not be granular enough to get a well-balanced distribution. This mode will also always limit a single VM (i.e., the traffic from a single switch port) to the bandwidth available on a single interface. *Windows Server* 2012 R2 *uses the Hyper-V Switch Port as the identifier rather than the source MAC address as, in some instances, a VM may be using more than one MAC address on a switch port.*
* **Address Hashing.** This algorithm creates a hash based on address components of the packet and then assigns packets that have that hash value to one of the available adapters. Usually this mechanism alone is sufficient to create a reasonable balance across the available adapters.

The components that can be specified, using PowerShell, as inputs to the hashing function include the following:

* + Source and destination TCP ports and source and destination IP addresses (this is used by the user interface when “Address Hash” is selected)
  + Source and destination IP addresses only
  + Source and destination MAC addresses only

The TCP ports hash creates the most granular distribution of traffic streams resulting in smaller streams that can be independently moved between members. However, it cannot be used for traffic that is not TCP or UDP-based or where the TCP and UDP ports are hidden from the stack, such as IPsec-protected traffic. In these cases, the hash automatically falls back to the IP address hash or, if the traffic is not IP traffic, to the MAC address hash.

See Section 4.1.7.2.3 for the PowerShell commands that can switch a team between load distribution modes and a deeper explanation of each hashing mode.

* **Dynamic.** This algorithm takes the best aspects of each of the other two modes and combines them into a single mode.
  + Outbound loads are distributed based on a hash of the TCP Ports and IP addresses. Dynamic mode also rebalances loads in real time so that a given outbound flow may move back and forth between team members.
  + Inbound loads are distributed as though the Hyper-V port mode was in use. See Section 3.5 for more details.

The outbound loads in this mode are dynamically balanced based on the concept of **flowlets**. Just as human speech has natural breaks at the ends of words and sentences, TCP flows (TCP communication streams) also have naturally occurring breaks. The portion of a TCP flow between two such breaks is referred to as a flowlet. When the dynamic mode algorithm detects that a flowlet boundary has been encountered, i.e., a break of sufficient length has occurred in the TCP flow, the algorithm will opportunistically rebalance the flow to another team member if appropriate. The algorithm may also periodically rebalance flows that do not contain any flowlets if circumstances require it. As a result the affinity between TCP flow and team member can change at any time as the dynamic balancing algorithm works to balance the workload of the team members.

Switch-embedded Teaming supports only the Hyper-V switch port and Dynamic load distribution algorithms.

## Interactions between Configurations and Load distribution algorithms

### Switch Independent configuration / Address Hash distribution

This configuration will distribute the load through the use of the selected level of address hashing. It defaults to using TCP ports and IP addresses to seed the hash function.

Because a given IP address can only be associated with a single MAC address for routing purposes, this mode receives inbound traffic on only one team member (the primary member). This means that the inbound traffic cannot exceed the bandwidth of one team member no matter how much is getting sent.

This mode is best used for teaming in a VM. (See Section 3.6.)

In all other cases where this configuration was recommended in previous versions of Windows Server the configuration, Switch Independent/Dynamic, described in Section 3.5.3 should provide better performance.

### Switch Independent configuration / Hyper-V Port distribution

This configuration will send packets using all active team members distributing the load based on the Hyper-V switch port number. Each Hyper-V port will be bandwidth limited to not more than one team member’s bandwidth because the port is affinitized to exactly one team member at any point in time.

Because each Hyper-V port is associated with a single team member, this mode receives inbound traffic for the VM’s switch port on the same team member the switch port’s outbound traffic uses. This also allows maximum use of Virtual Machine Queues (VMQs) for better performance over all.

In all cases where this configuration was recommended in previous versions of Windows Server the configuration, Switch Independent/Dynamic, described in 3.5.3 , will provide better performance except when teaming NICs that operate at or above 10Gbps. When teaming these high-bitrate NICs, HyperVPort distribution mode may provide better system performance than Dynamic distribution mode.

### Switch Independent configuration / Dynamic distribution

This configuration will distribute the load based on the TCP Ports address hash as modified by the Dynamic load balancing algorithm. The Dynamic load balancing algorithm will redistribute flows to optimize team member bandwidth utilization so individual flow transmissions may move from one active team member to another. The algorithm takes into account the small possibility that redistributing traffic could cause out-of-order delivery of packets so it takes steps to minimize that possibility.

The receive side, however, will look identical to Hyper-V Port distribution. Each Hyper-V switch port’s traffic, whether bound for a virtual NIC in a VM (vmNIC) or a virtual NIC in the host (vNIC), will see all its inbound traffic arriving on a single NIC.

**This mode is best used for teaming in both native and Hyper-V environments** except when:

1. Teaming is being performed in a VM,
2. Switch dependent teaming (e.g., LACP) is required by policy, or
3. Operation of an Active/Standby team is required by policy.

Note: When teaming NICs that operate at or above 10Gbps, HyperVPort distribution mode may provide better system performance than Dynamic distribution mode.

### Switch Dependent configuration / Address Hash distribution

This configuration will distribute the load through the use of the selected level of address hashing. It defaults to using TCP ports and IP addresses to seed the hash function.

Like in all switch dependent configurations, the switch determines how to distribute the inbound traffic among the team members. The switch is expected to do a reasonable job of distributing the traffic across the team members but it has complete independence to determine how it does so.

In all cases where this configuration was recommended in Windows Server 2012 the new configuration, Switch Dependent/Dynamic, described in 3.5.6, will provide better performance. As a result this mode is not a first-choice recommendation for any workload.

SET does not support Switch Dependent configuration in this release.

### Switch Dependent configuration / Hyper-V Port distribution

This configuration will distribute the load based on the Hyper-V switch port number. Each Hyper-V port will be bandwidth limited to not more than one team member’s bandwidth because the port is affinitized to exactly one team member at any point in time.

Like in all switch dependent configurations, the switch determines how to distribute the inbound traffic among the team members. The switch is expected to do a reasonable job of distributing the traffic across the team members but it has complete independence to determine how it does so.

In all cases where this configuration was recommended in Windows Server 2012 the new configuration, Switch Dependent/Dynamic, described in 3.5.6, will provide better performance.

SET does not support Switch Dependent configuration in this release.

### Switch Dependent configuration / Dynamic distribution

This configuration will distribute the load based on the TransportPorts address hash as modified by the dynamic load balancing algorithm. The Dynamic load balancing algorithm will redistribute flows to optimize team member bandwidth utilization so individual flow transmissions may move from one active team member to another. The algorithm takes into account the small possibility that redistributing traffic could cause out-of-order delivery of packets so it takes steps to minimize that possibility.

Like in all switch dependent configurations, the switch determines how to distribute the inbound traffic among the team members. The switch is expected to do a reasonable job of distributing the traffic across the team members but it has complete independence to determine how it does so.

This mode is best used for teaming in all cases where Switch Dependent configurations are required by policy. Depending on the capabilities of the adjacent switch it may also do very well, and perhaps even best, when supporting teaming of native workloads.

SET does not support Switch Dependent configuration in this release.

## NIC teaming inside of Virtual Machines (VMs)

Only stand-alone NIC Teaming may be used to provide NIC Teaming in a VM.

NIC Teaming in a VM only applies to vmNICs connected to external switches; vmNICs connected to internal or private switches will show as disconnected when they are in a team.

NIC teaming in Windows Server 2012 R2 may also be deployed in a VM. This allows a VM to have virtual NICs connected to more than one Hyper-V switch and still maintain connectivity even if the physical NIC under one switch gets disconnected. This is particularly important when working with Single Root I/O Virtualization (SR-IOV) because SR-IOV traffic doesn’t go through the Hyper-V switch and thus cannot be protected by a team in or under the Hyper-V host. With the VM-teaming option an administrator can set up two Hyper-V switches, each connected to its own SR-IOV-capable NIC.

* Each VM can have a virtual function (VF) from one or both SR-IOV NICs and, in the event of a NIC disconnect, fail-over from the primary VF to the back-up adapter (VF).
* Alternately, the VM may have a VF from one NIC and a non-VF vmNIC connected to another switch. If the NIC associated with the VF gets disconnected, the traffic can fail-over to the other switch without loss of connectivity.

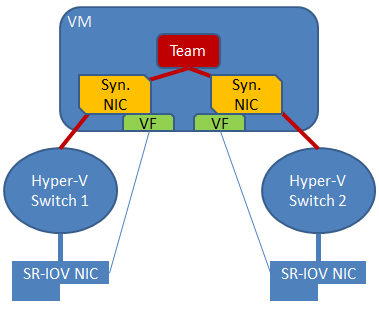


Figure - NIC Teaming in a VM with SR-IOV with two VFs

**Note:**  Because fail-over between NICs in a VM might result in traffic being sent with the MAC address of the other vmNIC, each Hyper-V switch port associated with a VM that is using NIC Teaming must be set to allow teaming There are two ways to enable NIC Teaming in the VM:

1. In the Hyper-V Manager, in the settings for the VM, select the VM’s NIC and the **Advanced Settings** item, then enable the checkbox for NIC Teaming in the VM. See Figure 5.

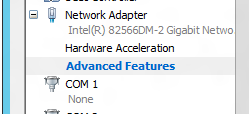
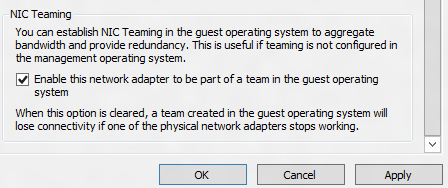


Figure - Enabling VM NIC Teaming in Hyper-V Manager

1. Run the following Windows PowerShell cmdlet in the host with elevated (Administrator) privileges.

Set-VMNetworkAdapter -VMName <VMname> -AllowTeaming On

Teams created in a VM can only run in Switch Independent configuration using one of the Address Hash distribution modes. Only teams where each of the team members is connected to a different external Hyper-V switch are supported.

Teaming in the VM does not affect Live Migration. The same rules exist for Live Migration whether or not NIC teaming is present in the VM.

## Hyper-V ports in the Host Partition

Hyper-V virtual NICs exposed in the host partition (vNICs) **must not be placed in a team**. Teaming of vNICs inside of the host partition is not supported in any configuration or combination. Attempts to team vNICs may result in a complete loss of communication in the event that network failures occur.

## Feature compatibilities

Stand-alone NIC teaming is compatible with all networking capabilities in Windows Server 2012 R2 with six exceptions: SR-IOV, RDMA, Native host Quality of Service, TCP Chimney, and 802.1X Authentication.

* For SR-IOV and RDMA, data is delivered directly to the NIC without passing it through the networking stack (in the host OS in the case of virtualization). Therefore, it is not possible for the team to look at or redirect the data to another path in the team.
* When QoS policies are set on a native or host system and those policies invoke minimum bandwidth limitations, the overall throughput through a NIC team will be less than it would be without the bandwidth policies in place.
* TCP Chimney is not supported with NIC teaming in Windows Server 2012 R2 since TCP Chimney has the entire networking stack offloaded to the NIC.
* 802.1X Authentication should not be used with NIC Teaming and some switches will not permit configuration of both 802.1X Authentication and NIC Teaming on the same port.
* When DCB is used with stand-alone NIC Teaming and software QoS applied above the teaming is unaware of the DCB settings. This can, in some circumstances, lead to challenges with bandwidth management.

Table 1 - Feature interactions with NIC teaming (See also section 5.1.3)

|  |  |  |
| --- | --- | --- |
| Feature | Stand-alone NIC Teaming | SET |
| Datacenter bridging (DCB) | Works independent of NIC Teaming so is supported if the team members support it. May lead to bandwidth management challenges. | Works cooperatively with SDN-QoS and SET. |
| Hyper-V Network Virtualization (HNV) / NV-GRE (HNVv1) | Supported when teaming is installed under the Hyper-V switch (see more in Section 3.8.2). | Not supported. |
| Hyper-V Network Virtualization (HNV) SDN version (HNVv2) | Not supported | Supported. NV-GRE and VxLAN support in HNVv2 are both supported. |
| IPsec Task Offload (IPsecTO) | Supported if all team members support it. | Not supported |
| Large Send Offload (LSO) | Supported if all team members support it. | |
| Receive side coalescing (RSC) | Supported in hosts if any of the team members support it. Not supported through Hyper-V switches. | Not supported |
| Receive side scaling (RSS) | NIC teaming supports RSS in the host. The Windows Server 2012 R2 TCP/IP stack programs the RSS information directly to the Team members. | Not supported |
| Virtual RSS | Supported | |
| Receive-side Checksum offloads (IPv4, IPv6, TCP) | Supported if any of the team members support it. | |
| Remote Direct Memory Access (RDMA) | Since RDMA data bypasses the Windows Server 2012 R2 protocol stack, team members will not also support RDMA. | Supported |
| Single root I/O virtualization (SR-IOV) | Since SR-IOV data bypasses the host OS stack, NICs exposing the SR-IOV feature will no longer expose the feature while a member of a team. Teams can be created in VMs to team SR-IOV virtual functions (VFs). See also Section 3.6. | Will be supported in the final release of Windows Server 2016 |
| TCP Chimney Offload | Not supported | |
| Transmit-side Checksum offloads (IPv4, IPv6, TCP) | Supported if all team members support it. | |
| Virtual Machine Queues (VMQ) | Supported when teaming is installed under the Hyper-V switch (see more in Section 3.8.1). | Supported |
| QoS in host/native OSs | Use of minimum bandwidth policies will degrade throughput through a team. | Not supported |
| Virtual Machine QoS (VM-QoS) | VM-QoS is affected by the load distribution algorithm used by NIC Teaming. For best results use Dynamic load distribution mode. | Not supported |
| SDN-QoS | Not supported | Supported. |
| 802.1X authentication | Not compatible with many switches. Should not be used with NIC Teaming. | |

### Stand-alone NIC Teaming and Virtual Machine Queues (VMQs)

VMQ and NIC Teaming work well together; VMQ should be enabled anytime Hyper-V is enabled. Depending on the switch configuration mode and the load distribution algorithm, NIC teaming will either present VMQ capabilities to the Hyper-V switch that show the number of queues available to be the smallest number of queues supported by any adapter in the team (Min-queues mode) or the total number of queues available across all team members (Sum-of-Queues mode). Specifically,

* if the team is in Switch-Independent teaming mode and the Load Distribution is set to Hyper-V Port mode or Dynamic mode, then the number of queues reported is the sum of all the queues available from the team members (**Sum-of-Queues** mode);
* otherwise the number of queues reported is the smallest number of queues supported by any member of the team (**Min-Queues** mode).

Table 2 - VMQ mode by Teaming mode and Load distribution mode

|  |  |  |  |
| --- | --- | --- | --- |
| Distribution mode→  Teaming mode↓ | Address Hash modes | HyperVPort | Dynamic |
| Switch independent | **Min-Queues** | **Sum-of-Queues** | **Sum-of-Queues** |
| Switch dependent | **Min-Queues** | **Min-Queues** | **Min-Queues** |

Here’s why:

* When the switch independent team is in Hyper-V Port mode or Dynamic mode the inbound traffic for a Hyper-V switch port (VM) will always arrive on the same team member. The host can predict/control which member will receive the traffic for a particular VM so NIC Teaming can be more thoughtful about which VMQ Queues to allocate on a particular team member. NIC Teaming, working with the Hyper-V switch, will set the VMQ for a VM on exactly one team member and know that inbound traffic will hit that queue.
* When the team is in any switch dependent mode (static teaming or LACP teaming), the switch that the team is connected to controls the inbound traffic distribution. The host’s NIC Teaming software can’t predict which team member will get the inbound traffic for a VM and it may be that the switch distributes the traffic for a VM across all team members. As a result the NIC Teaming software, working with the Hyper-V switch, programs a queue for the VM on every team member, not just one team member.
* When the team is in switch-independent mode and is using an address hash load distribution algorithm, the inbound traffic will always come in on one NIC (the primary team member) – all of it on just one team member. Since other team members aren’t dealing with inbound traffic they get programmed with the same queues as the primary member so that if the primary member fails any other team member can be used to pick up the inbound traffic and the queues are already in place.

There are a few VMQ settings that will help the system perform even better. For context keep in mind that most NICs have queues that can be used for either RSS or VMQ but not both at the same time. Hence some VMQ settings appear to be settings for RSS queues but are really settings on the generic queues that both RSS and VMQ use depending on which feature is presently in use.

Each NIC has, in its advanced properties, values for \*RssBaseProcNumber and \*MaxRssProcessors.

* Ideally each NIC should have the \*RssBaseProcNumber set to an even number greater than or equal to two (2). This is because the first physical processor, Core 0 (logical processors 0 and 1), typically does most of the system processing so the network processing should be steered away from this physical processor. (Some machine architectures don’t have two logical processors per physical processor so for such machines the base processor should be greater than or equal to 1. If in doubt assume your host is using a 2 logical processor per physical processor architecture.)
* If the team is in **Sum-of-Queues** mode the team members’ processors should be, to the extent practical, non-overlapping. For example, in a 4-core host (8 logical processors) with a team of 2 10Gbps NICs, you could set the first one to use base processor of 2 and to use 4 cores; the second would be set to use base processor 6 and use 2 cores.
* If the team is in **Min-Queues** mode the processor sets used by the team members must be identical.

### Hyper-V Network Virtualization (HNVv1) / NV-GRE compatibility

Windows Server stand-alone NIC Teaming is fully compatible with Hyper-V Network Virtualization (HNVv1). The HNV management system provides information to the NIC Teaming driver that allows NIC Teaming to distribute the load in a way that is optimized for the HNV traffic. For further information on this topic see the documentation for HNV.

SET does not support operation with HNVv1.

### Hyper-V Network Virtualization – SDN (HNVv2)

SET fully supports all the capabilities of HNVv2 including both NV-GRE and VxLAN operation.

Stand-alone NIC Teaming does not support operation with HNVv2.

## NIC Requirements and limitations

For various reasons not all maximum configuration limits may be achievable at the same time.

SET requires all NICs in a team to be identical (i.e., same manufacturer, same model, same firmware and driver).

### Number of NICs in a team in a native host

NIC teaming requires the presence of at least one Ethernet NIC. A team of one NIC may be used for separation of traffic using VLANs. Obviously a team with only one team member has no failure protection. Fault protection (failover) requires a minimum of two Ethernet NICs in the team.

Stand-alone NIC Teaming supports up to 32 NICs in a team.

SET supports up to 8 NICs in a team.

### Number of NICs in a team in a Hyper-V VM

Stand-alone NIC Teaming supports teams with two members in VMs. Larger teams can be created but such teams are not supported. Every team member must be connected to a different external Hyper-V switch and the VM’s networking interfaces must be set to allow teaming.

SET may not be used to provide teaming in a VM.

### Types of NICs in a team

Any **Ethernet** NIC that has passed the Windows Hardware Qualification and Logo test (WHQL tests) may be used in a team.

NICs representing technologies other than Ethernet are not supported in teams (e.g., WWAN, WLAN/WiFi, Bluetooth, Infiniband including IPoIB NICs).

Hyper-V vNICs, i.e., Hyper-V switch ports exposed as NICs in the host partition, may not be placed in a team.

The kernel debug network adapter (KDNIC) may not be placed in a team.

NICs that are being used for network boot may not be placed in a team.

### Number of team interfaces for a team

Stand-alone NIC Teaming supports up to 32 team interfaces per team.

SET architecture does not expose team interfaces. See Hyper-V switch documentation for switch-port information.

## Teaming of different speed NICs

### Stand-alone NIC Teaming

Teaming of NICs capable of operating at different speeds but presently operating at the same speed is supported, e.g., a 10 Gbps NIC operating as a 1Gbps NIC may be teamed with another NIC operating at 1Gbps.

Teaming of NICs with different speed connections is not supported. The teaming software will allow you to build such a team; however the traffic distribution algorithms in this release do not base the distribution on the speeds of the connections. A team consisting of a 10Gbps NIC and a 100Mbps NIC will send approximately half of the traffic to each NIC.

If you are creating a team where one team member will be active and another will be available to help out if the active team member fails (active/standby mode, see Section 3.3), a lower speed NIC may be used as the standby NIC so that connectivity is maintained.

### Switch-embedded teaming

Only NICs of the same make, model, and firmware load may be teamed. Teaming of NICs that are not identical is not supported. If the NICs are capable of operating at more than one bitrate (e.g., 10Gbps/40Gbps) all team members must be set to operate at the same bitrate.

## Teams of teams

A Team of teams is a team where the team members are actually team interfaces from other teams. This is not supported in the stand-alone NIC Teaming solution. Furthermore, attempting to place the team interface from a 3rd-party teaming solution into a Microsoft team may result in an unstable system that may lose connectivity to the network. ***DO NOT mix elements of 3rd-party teaming solutions with elements of any of Microsoft’s NIC Teaming solutions***.

## MAC address use and MAC address management

### The MAC address of the team

This section applies only to NIC Teaming. It does not apply to SET as a SET team has no MAC address.

In switch independent mode with address hash or dynamic load distribution the team will use the MAC address of the *primary* team member (one selected from the initial set of team members) on outbound traffic. The *primary* team member is the first team member to bind to the team after team creation or host reboot. Since the *primary* team member may change in a non-deterministic manner at each boot, NIC disable/enable action, or other reconfiguration activities, the MAC address of the team may vary from time to time. Normally this won’t cause problems but there are a few cases where it may.

If the *primary* team member is removed from the team and then placed into operation there may be a MAC address conflict. To resolve this conflict disable and enable the team interface. The process of doing a disable and enable operation on the team interface will cause it to select a new MAC address from the remaining team members.

If MAC address stability is desired for the team the administrator can set the MAC address of the team to any MAC address the administrator wants to use by setting it in the primary team interface the same way that an administrator can set the MAC address of any physical NIC.

MAC address changes are not supported in VMs.

### MAC address use on transmitted packets

This section applies to both NIC Teaming and SET.

In Switch Independent mode with Address Hash or Dynamic distribution the packets from a single source (e.g., a single VM) may be distributed across multiple team members at the same time. In order to prevent the switches from getting confused and to prevent *MAC flapping* alarms the frames transmitted on team members other than the primary team member will have the source MAC address replaced. This will result in any given MAC address being always used on only one interface unless and until failure occurs.

When a failure is detected on the primary NIC the NIC Teaming software will start using the primary team member’s MAC address on the team member chosen to serve as the temporary primary team member (i.e., the one that will now appear to the switch as the primary team member). This change will only apply to traffic that was going to be sent on the primary team member with the primary team member’s MAC address as its source MAC address. Other traffic will continue to be sent with whatever source MAC address it would have used prior to the failure.

To be specific:

* In Switch Independent mode with Address Hash distribution
  + All special packets including ARP, NS, and ICMP packets are sent on the primary team member (this is not an exhaustive list)
  + All traffic sent on NICs other than the primary team member are sent with the source MAC address modified to match the NIC on which they are sent
  + All traffic sent on the primary team member is sent with the original source MAC address (which may be the team’s source MAC address)
* In Switch Independent mode with HyperVPort distribution
  + Every vmSwitch port is affinitized to a team member[[5]](#footnote-6)
  + Every packet is sent on the team member to which the port is affinitized
  + No source MAC replacement is done
* In Switch Independent mode with Dynamic distribution
  + Every vmSwitch port is affinitized to a team member5
  + All ARP/NS packets are sent on the team member to which the port is affinitized
  + Packets sent on the team member that is the affinitized team member have no source MAC address replacement done
  + Packets sent on a team member other than the affinitized team member will have source MAC address replacement done
* In Switch Dependent mode (all distributions)
  + No source MAC replacement is done[[6]](#footnote-7)

## Industry terms for NIC Teaming

Here are some other terms that are used in the context of NIC Teaming and what they are called in Windows Server:

|  |  |
| --- | --- |
| **Term** | **What we call it** |
| IEEE 802.3ad | LACP (or IEEE 802.1ax LACP) |
| Link Aggregation Group (LAG) | Team (often refers to a team that uses LACP) |
| Load Balancing and Failover (LBFO) | NIC Teaming |
| NIC Bonding | NIC Teaming |

## Troubleshooting (The dangers of using a powerful tool)

NIC Teaming and the powerful administration tools in Windows Server are very powerful tools that can be misused, misconfigured, and may cause loss of connectivity if the administrator isn’t careful. Here are some common issues:

### Using VLANs

VLANs are a powerful tool that solves many problems for administrators. There are a few rules for using VLANs that will help to make the combination of VLANs and NIC Teaming a very positive experience.

1. Anytime you have NIC Teaming enabled, the physical switch ports the host is connected to should be set to trunk (promiscuous) mode. The physical switch should pass all traffic to the host for filtering without modification.[[7]](#footnote-8)
2. Anytime you have NIC Teaming enabled, you must not set VLAN filters on the NICs using the NICs advanced properties settings. Let the teaming software or the Hyper-V switch (if present) do the filtering.

When using SET all VLAN settings must be configured on the VM’s switch port.

#### VLANs in a Hyper-V host

This section applies only to NIC Teaming. It does not apply to SET as a SET team has no team interfaces on which a VLAN may be enabled.

In a Hyper-V host VLANs should be configured only in the Hyper-V switch, not in the stand-alone NIC Teaming software. Configuring team interfaces with VLANs can easily lead to VMs that are unable to communicate on the network due to collisions with VLANs assigned in the Hyper-V switch. Consider the following NIC Teaming example:

|  |  |  |
| --- | --- | --- |
| Example of WRONG configuration |  | Example of WRONG configuration |

Figure 6 - VLAN misconfiguration (stand-alone NIC Teaming)

Figure 6 shows a common misconfiguration that occurs when administrators try to use team interfaces for VLAN support and also bind the team to a Hyper-V switch. In this case VM C will never receive any inbound traffic because all the traffic destined for VLAN 17 is taken out at the teaming module. All traffic *except traffic tagged with VLAN 17* will be forwarded to the Hyper-V switch, but VM C’s inbound traffic never arrives. This kind of misconfiguration has been seen often enough for Microsoft to declare this kind of configuration, i.e., VLANs exposed at the teaming layer while the team is bound to the Hyper-V switch, unsupported. Repeat: If a team is bound to a Hyper-V switch the team MUST NOT have any VLAN-specific team interfaces exposed. This is an unsupported configuration.

#### VLANs in a Hyper-V VM

1. The preferred method of supporting multiple VLANs in a VM is to provide the VM multiple ports on the Hyper-V switch and associate each port with a VLAN. Never team these ports in the VM as it will certainly cause communication problems.
2. If the VM has multiple SR-IOV VFs make sure they are on the same VLAN before teaming them in the VM. It’s easily possible to configure the different VFs to be on different VLANs and, like in the previous case, it will certainly cause communication problems.
3. The only safe way to use VLANs with NIC Teaming in a guest is to team Hyper-V ports that are
   1. Each connected to a different external Hyper-V switch, and
   2. Each configured to be associated with the same VLAN (or all associated with untagged traffic only).
   3. TIP: If you must have more than one VLAN exposed into a guest OS consider renaming the ports in the guest to indicate what the VLAN is. E.g., if the first port is associated with VLAN 12 and the second port is associated with VLAN 48, rename the interface Ethernet to be EthernetVLAN12 and the other to be EthernetVLAN48. Renaming interfaces is easy using the Windows PowerShell Rename-NetAdapter cmdlet or by going to the Network Connections panel in the guest and renaming the interfaces.

### Interactions with other teaming solutions

Some users will want to use other NIC teaming solutions for a variety of reasons. This can be done but there are some risks that the system administrator should be aware of.

1. If the system administrator attempts to put a NIC into a 3rd party team that is presently part of a Microsoft NIC Teaming team, the system will become unstable and communications may be lost completely.
2. If the system administrator attempts to put a NIC into a Microsoft NIC Teaming team that is presently part of a 3rd party teaming solution team the system will become unstable and communications may be lost completely.

As a result it is **STRONGLY RECOMMENDED** that no system administrator ever run two teaming solutions at the same time on the same server. The teaming solutions are unaware of each other’s existence resulting in potentially serious problems.

In the event that an administrator violates these guidelines and gets into the situation described above the following steps may solve the problem.

1. Reboot the server. Forcibly power-off the server if necessary to get it to reboot.
2. When the server has rebooted run this Windows PowerShell cmdlet:

Get-NetLbfoTeam | Remove-NetLbfoTeam

1. Use the 3rd party teaming solution’s administration tools and remove all instances of the 3rd party teams.
2. Reboot the server again.

Microsoft continues its longstanding policy of not supporting 3rd party teaming solutions. If a user chooses to run a 3rd party teaming solution and then encounters networking problems, the customer should call their teaming solution provider for support. If the issue is reproducible without the 3rd party teaming solution in place, please report the problem to Microsoft.

### MAC address conflicts

See section 3.12.

### Physical network segmentation

NIC teaming is a layer 2 feature, meaning that all NICs in the team must be connected to the same subnet.  In other words, there must always be a path to get from any NIC in the team to any other NIC in the team going only through layer 2 switches and not any layer 3 entities (e.g. routers). Administrators should check the physical layout and ensure that all NICs in the team are on the same subnet to avoid connectivity problems.

### Hardware that doesn’t conform to specification

NIC teaming may send packets from the same IP address with multiple different source MAC addresses. Receivers of those packets must resolve the IP address of the host or VM to a single particular MAC instead of responding to the MAC address from which the packet was received. Clients that correctly implement the address resolution protocols, IPv4’s ARP or IPv6’s neighbor discovery protocol (NDP), will send packets with the correct destination MAC address (the MAC address of the VM or host that owns that IP address).

Microsoft has seen cases where certain embedded hardware, such as SAN controllers, do not correctly implement the address resolution protocols and/or do not explicitly resolve an IP address to a MAC address using ARP or NDP. Instead these non-conforming devices copy the source MAC contained in a received packet and use that as the destination MAC in the corresponding outgoing packets. This results in packets being sent to the wrong destination MAC address and thus getting dropped by the virtual switch because they don’t match any known destination.

Administrators that are having trouble connecting to SAN controllers or other embedded hardware should take packet captures and determine whether or not their hardware is implementing ARP or NDP and contact their hardware vendor for support.

### Physical switch security features

Depending on configuration, NIC teaming may send packets from the same IP address with multiple different source MAC address. This can trip up security features on the physical switch such as dynamic ARP inspection or IP source guard especially if the physical switch is not aware that the ports are part of a team (switch-independent mode). Administrators should check the switch logs to determine whether switch security features are causing connectivity problems with NIC teaming.

### Disabling and Enabling with Windows PowerShell

A common reason for a team to not be passing traffic is that the team interface is disabled. We’ve seen a number of cases where attempts to use the power of Windows PowerShell have resulted in unintended consequences. For example, the sequence:

Disable-NetAdapter \*

Enable-NetAdapter \*

does not enable all the NetAdapters that it disabled. This is because disabling all the underlying physical member NICs will cause the team interface to be removed and no longer show up in Get-NetAdapter. Thus the Enable-NetAdapter \* will not enable the team NIC since that adapter has been removed. It will however enable the member NICs, which will then (after a short time) cause the team interface to be recreated. The team interface will still be in a “disabled” state since it has not been re-enabled. Enabling the team interface after it is recreated will cause traffic to begin to flow again.

# Managing NIC Teaming

## Stand-alone NIC Teaming

Windows Server stand-alone NIC Teaming has two inbox management tools. Other Microsoft management products may also provide the ability to manage Windows Server stand-alone NIC teaming. Windows client SKUs that allow the installation of the Remote Server Administration Tools (RSAT)[[8]](#footnote-9) are also able to manage NIC teaming on both Windows Server 2012 R2 hosts and Windows Server 2012 hosts.

This section provides a detailed presentation of the stand-alone NIC Teaming management UI. For each operation the equivalent Windows PowerShell cmdlets will be provided. The NIC Teaming management UI operates exclusively through the use of Windows PowerShell underneath. Everything the UI does can be done with Windows PowerShell. The NIC Teaming Windows PowerShell module is NetLbfo and all the cmdlet names start with NetLbfo.

For a full understanding of the Windows PowerShell commands for NIC Teaming,

Get-Command -Module NetLbfo

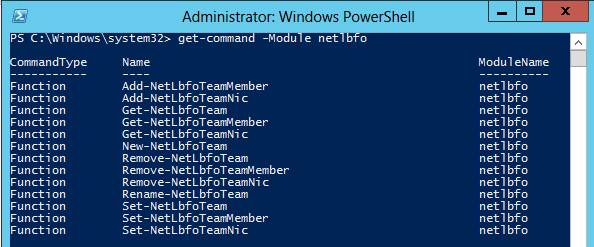
will return a list of all the NIC Teaming Windows PowerShell commands as shown in Figure 7 .

Figure 7 – NIC Teaming Windows PowerShell Cmdlets

Executing the Windows PowerShell command Get-Help <command> will provide a full syntax and all the options available.

Get-Help New-NetLbfoTeam

is shown in Figure 8.

### Invoking the Management UI for NIC Teaming

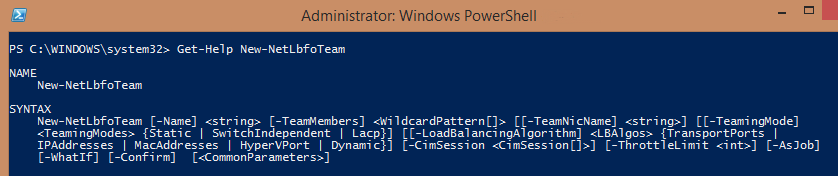


Figure - PowerShell Get-Help

There are three common ways to start the NIC Teaming Management UI tool.

* From the Server Manager **Local Server** window (In the NIC Teaming item click on **Disabled** or **Enabled**)

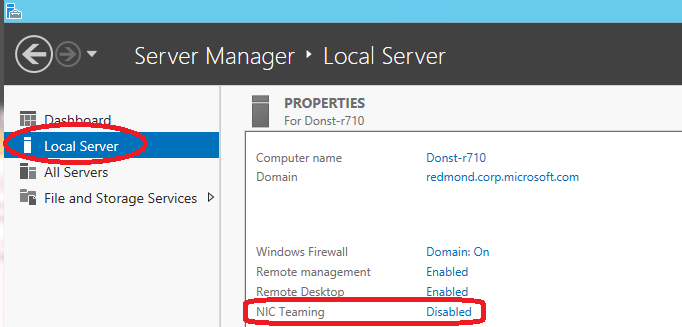


Figure 9 - Invoking the UI from Server Manager Local Server screen

* From the Server Manager **All Servers** window right-click on the server to be managed and select the **Configure Network Adapter Teaming** action.

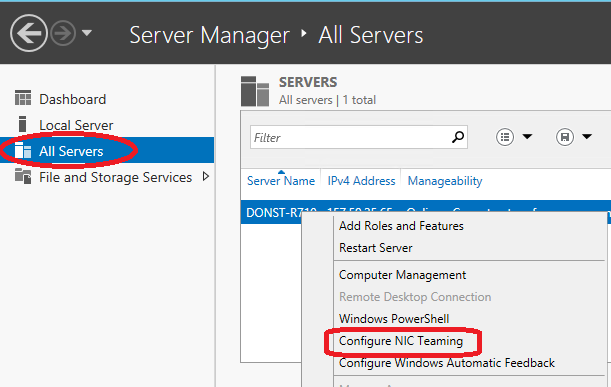


Figure 10- Invoking the UI from Server Manager All Servers screen

* From a command prompt or Windows PowerShell prompt: execute LbfoAdmin.exe.
  + The first time the NIC Teaming UI (LbfoAdmin.exe) is run, if it is invoked with no arguments it will attempt to manage NIC teaming on the local host. If the /servers flag is present then all the servers in the list of servers that follows will be included in the servers to be managed.
    - If the system detects that a server is already in the list of managed servers it will not add it to the list.
    - It is possible for the same server to appear more than once in the list especially where the name is presented differently (e.g., once with just the machine name and once with a fully-qualified machine name).
  + After the first time the NIC Teaming UI has been run it will remember the set of servers you last managed and will attempt to continue to manage that set of servers.
  + To cause the NIC Teaming UI to forget the previous settings, restore initial settings, and run like it was the first time invocation, run

LbfoAdmin.exe /ResetConfig

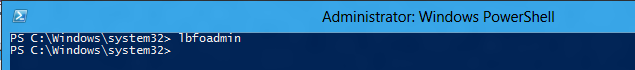


Figure 11- Invoking the UI from a Windows PowerShell prompt

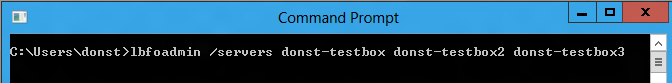


Figure 12- Invoking the UI from a Command Prompt

### The components of the NIC Teaming Management UI

The NIC Teaming management UI consists of 3 primary windows (tiles):

* The Servers tile
* The Teams tile, and
* The Adapters and Interfaces tile

and two secondary tiles that are present or not present depending on the size of the window and the selections within the window:

* The Team Statistics tile, and
* The Adapter Statistics tile

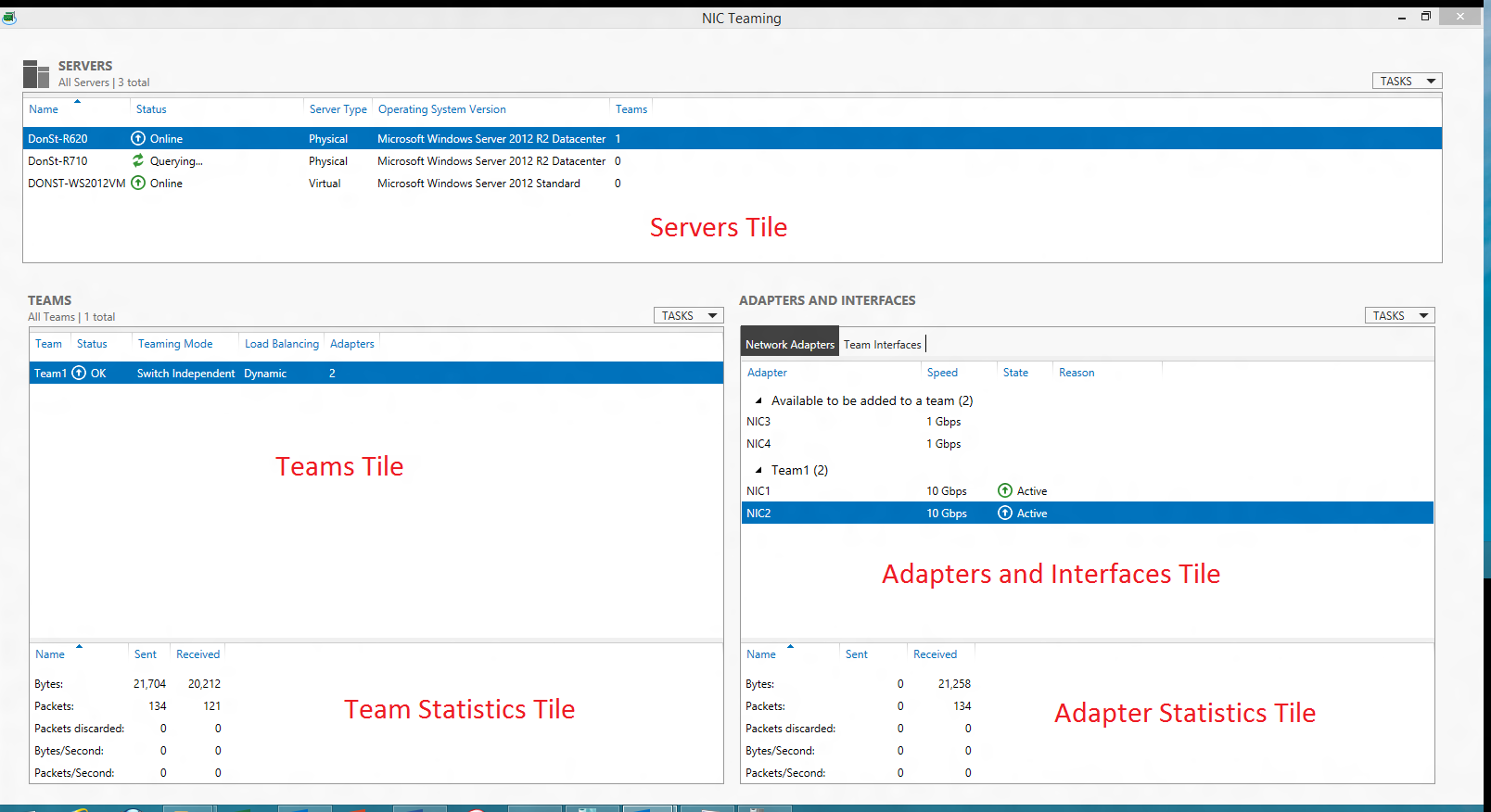


Figure 13 - the NIC Teaming Management UI tiles

The **Adapters and Interfaces tile** is shared by two tabs:

* The **Network Adapters tab**
* The **Team Interfaces tab**

Each tile or tab has a set of columns that can be shown or hidden. The column chooser menus are made visible by right-clicking on any column header. (For illustrative purposes the screen shot in Figure 14 shows a column chooser in every tile and both the Network Adapters and Team Interfaces tabs active. Only one column chooser can be active at a time.)

Contents of any tile may be sorted by any column. To sort by a particular column left click on the column title. In Figure 14 the **Servers** tile is sorted by Server Name; the indication is the little triangle in the **Name** column title in the **Servers** tile.

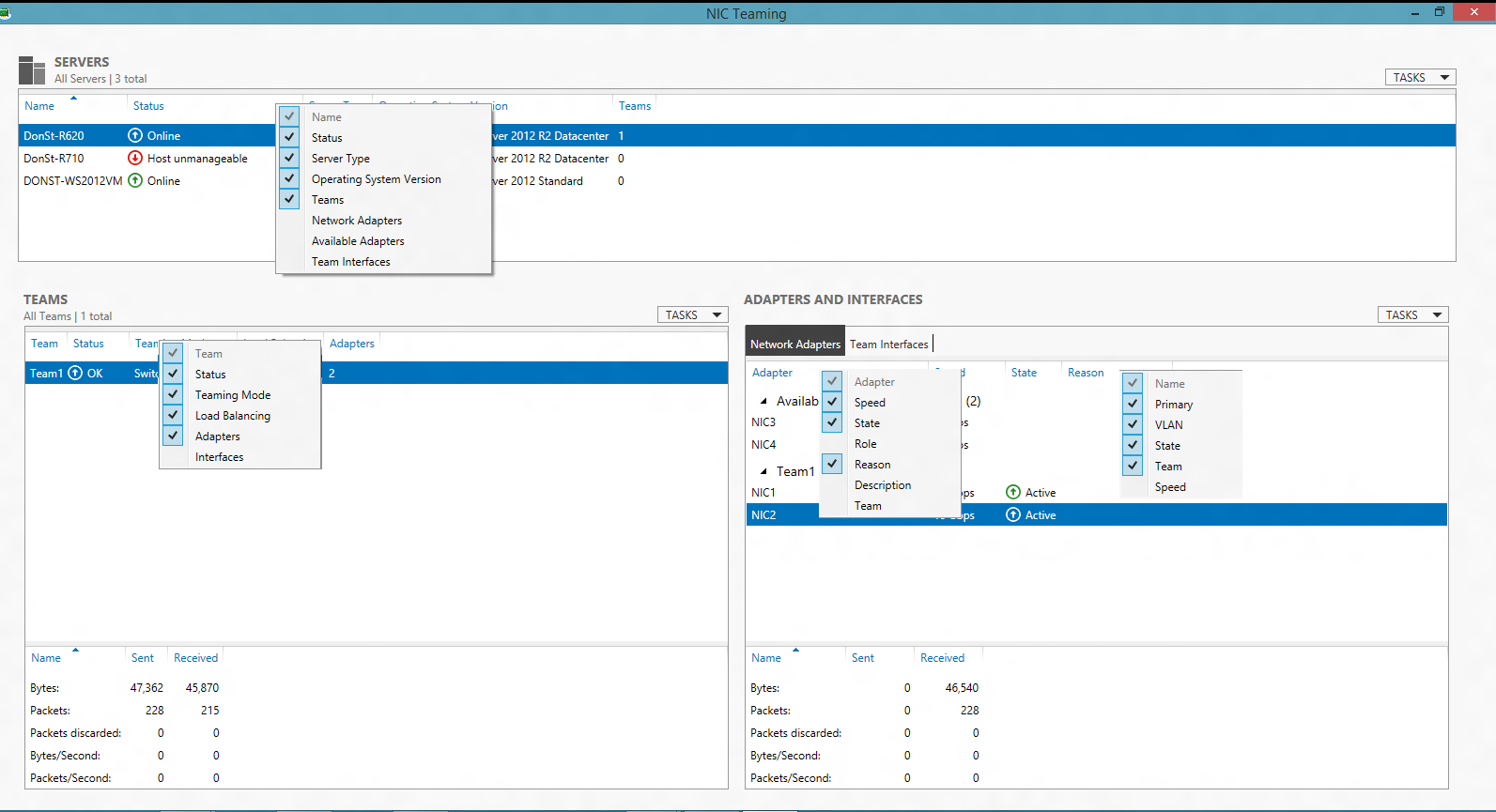


Figure 14 - Column Chooser menus

Each tile also has a **Tasks** dropdown menu and a right-click context menu. The **Tasks** menus can be opened by clicking on the **Tasks** box at the top right corner of the tile and then any available task in the list can be selected. The right-click context menus are activated by right-clicking in the tile. The menu options will vary based on context. (For illustrative purposes the screen shot in Figure 15 shows all the **Tasks** menus and a right-click menu in every tile. Only one right-click menu or **Tasks** menu can be active at any time.) Figure 16 shows the **Tasks** menus and right-click menu for the Team Interfaces tab.

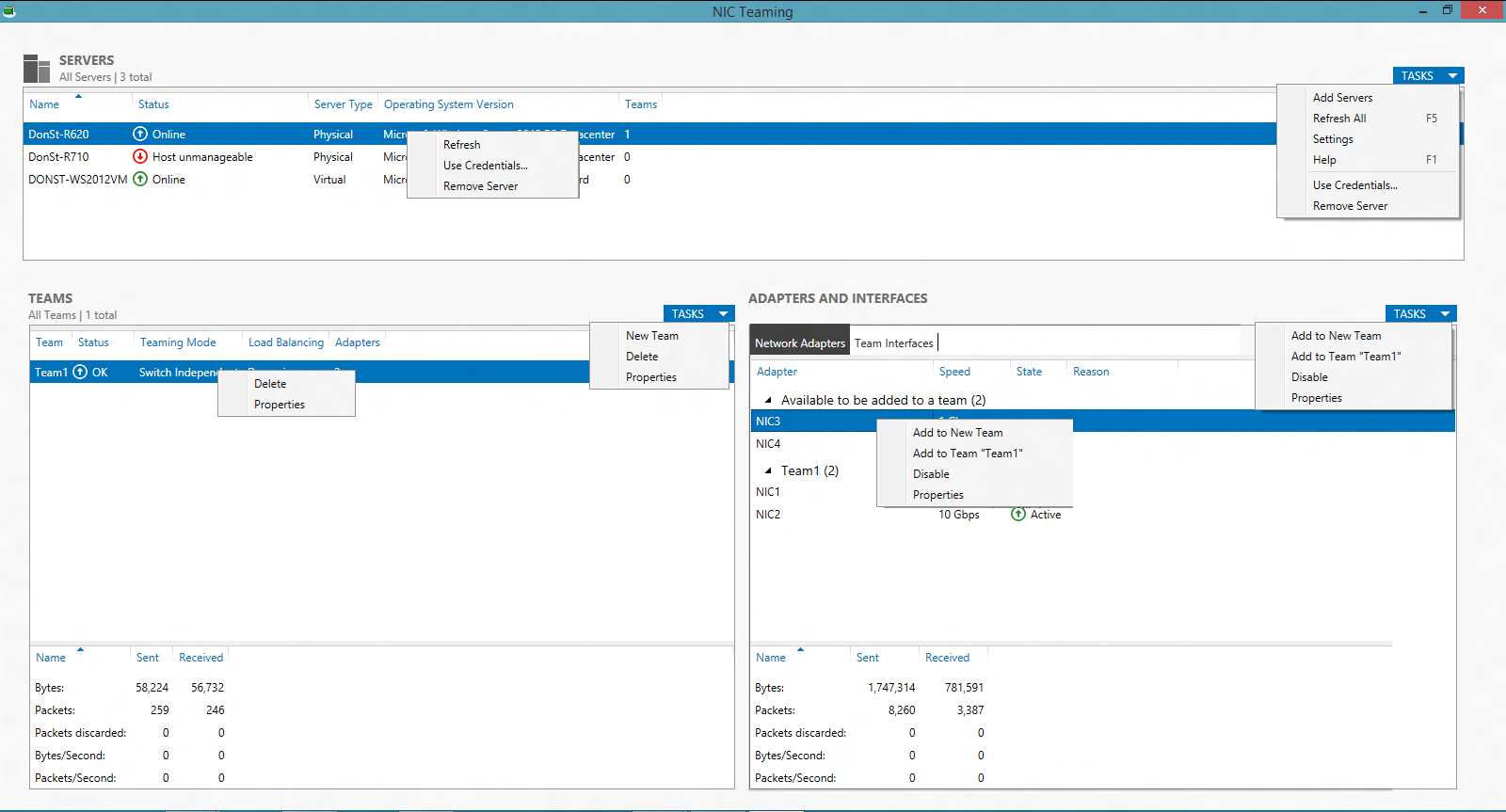


Figure 15 – Tasks menus and Right-click action menus

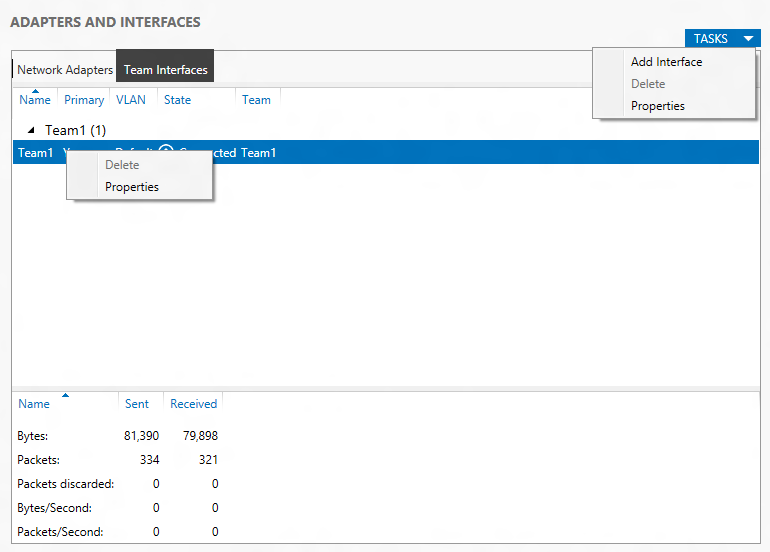


Figure 16 - Team Interfaces Tasks and Right-click Action Menu

### Adding a server to be managed

To add a server to the list of servers to be managed, i.e., the list of servers in the **Servers** tile, select the **Servers** tile **Tasks** menu. Selecting the **Add servers** action will pop-up the **Add server** dialog box. The **Add server** dialog box for the NIC Teaming UI is the same as the **Add server** dialog box for Server Manager.

### Removing a server from the managed servers list

To remove a server from the list of managed servers right click on the server to be deleted and then select the **Remove server** action.

### Creating a team

There are two ways to invoke the **New Team** dialog box:

* Select the **Tasks** menu in the **Teams** tile and then select **New Team**, or
* Right click on an available adapter in the **Network Adapters** tab and select the **Add to new team** item. Multi-select works for this: you can select multiple adapters, right-click on one, select **Add to new team**, and they will all be pre-marked in the **New Team** dialog box.

Both of these will cause the **New Team** dialog box to pop-up.

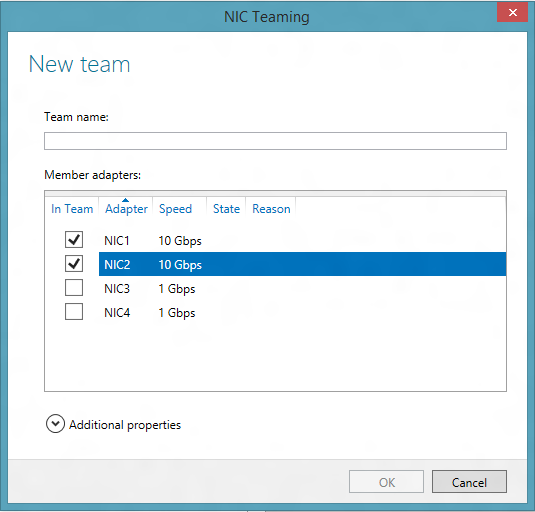


Figure 17 - New Team dialog box

When the **New Team** dialog box pops-up there are two actions that MUST be taken before the team can be created:

* A **Team name** must be provided, and
* One or more adapters must be selected to be members of the team

Optionally, the administrator may select the **Additional properties** item and configure the teaming mode, load distribution mode, and the name of the first (primary) team interface.

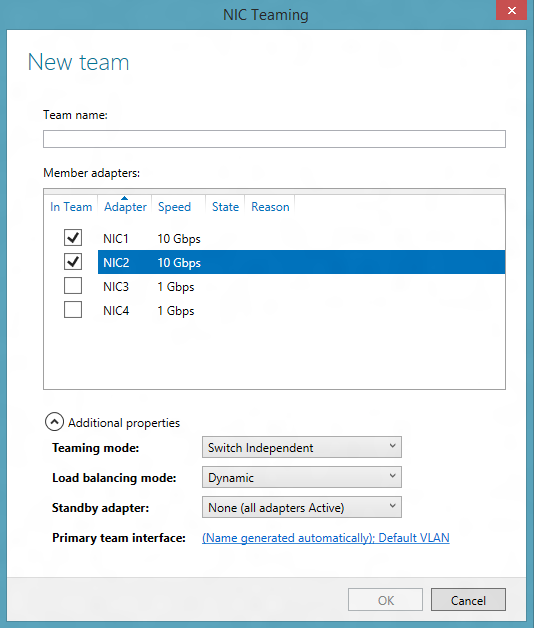


Figure 18 - New Team dialog box with Additional Properties expanded

In Additional properties the Load distribution mode drop-down provides three options: Address Hash, Hyper-V Port, and Dynamic. The Address Hash option in the UI is the equivalent of the TransportPorts option in Windows PowerShell. To select additional Address hashing algorithms use Windows PowerShell as described below and in Section 4.1.7.2.3.

This is also the place where those who want to have a Standby adapter in the team (See Section 3.3) to set the Standby adapter. Selecting the Standby adapter drop-down will give a list of the team members. The administrator can set one of them to be the Standby Adapter. A Standby adapter is not used by the team unless and until another member of the team fails. *Standby adapters are only permitted in Switch Independent mode. Changing the team to any Switch Dependent mode will cause all members to be made active members.*

When the team name, the team members, and optionally any additional properties (including the Primary team interface name or standby adapter) have been set to the administrator’s choices, the administrator will click on the OK button and the team will be created. Team creation may take several seconds and the NICs that are becoming team members will lose communication for a very short time.

Teams can also be created through Windows PowerShell. The Windows PowerShell to do exactly what these figures have shown is

New-NetLbfoTeam Team1 NIC1,NIC2

Teams can be created with custom advanced properties. See Sections 4.1.7.2.2 and 4.1.7.2.3 for more information about these flags.

New-NetLbfoTeam Team1 NIC1,NIC2 -TeamingMode LACP

‑LoadBalancingAlgorithm HyperVPort

If the team is being created in a VM, you **MUST** follow the instructions to allow guest teaming as described in Section 3.5.2.

### Checking the status of a team

Whenever the NIC Teaming UI is active the current status of all NICs in the team, the status of the team, and the status of the server will be shown. In Figure 19, in the **Network Adapters** tab of the **Adapters and Interfaces** tile, NIC 3 shows as faulted. The reason given is **Media Disconnected** (i.e., the cable is unplugged). This causes the team, Team1, to show a **Warning** as it is still operational but degraded. If all the NICs in the team were faulted it would show **Fault** instead of **Warning**. The server, DONST-R710, now shows **Warning**. If the team was not operational the server indication would be **Fault**. This makes it easy to scan the list of servers to see if there are any problems.

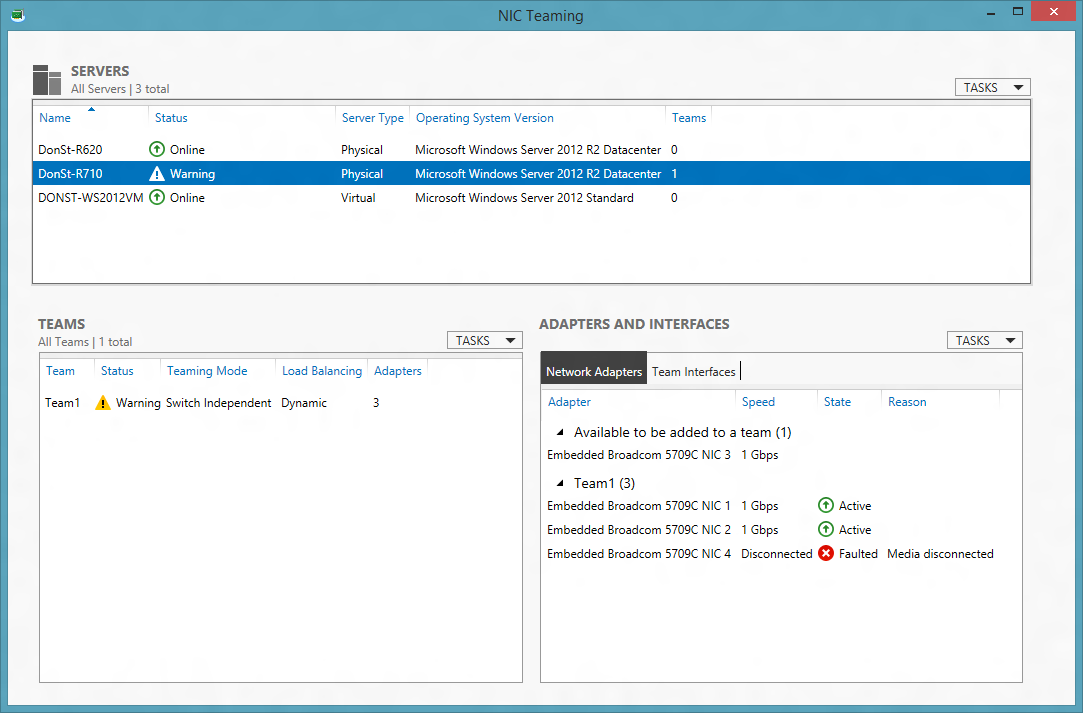


Figure 19 - Team with a faulted member

### Modifying a team

#### Modifying a team through the UI

Within the UI, modifications to the team can be done by selecting a team in the Team tile, right-clicking on the team, and selecting the Modify Team action. Selecting Modify Team will pop-up the Team properties dialog box. This dialog box is very similar to the New Team dialog box described in Section 4.1.5.

In the Team properties dialog box the following actions can be accomplished:

* Rename the team: Select the team name and edit it.
* Add team members: Select additional adapters from the Member Adapters tile
* Remove team members: De-select adapters from the Member Adapters tile. At least one adapter must be selected.

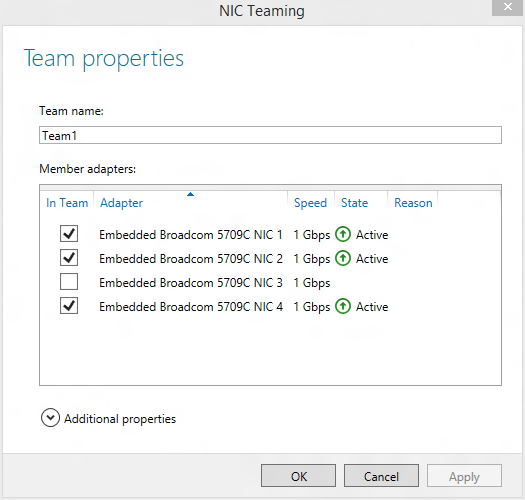


Figure 20 - Modifying Team Properties

If the Additional properties drop-down item is selected then the Teaming mode and Load distribution mode may also be modified. This Additional properties drop-down also allows the administrator to select a standby adapter when active-standby mode is desired.

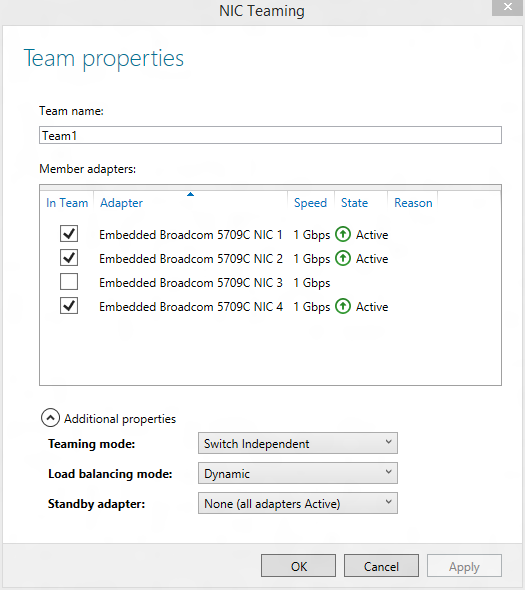


Figure 21 - Modifying a team's Teaming mode, Load distribution mode, and Standby Adapter

#### Modifying a team through Windows PowerShell

##### Renaming a team

To rename Team1 and give it the name TeamA, the Windows PowerShell is:

Rename-NetLbfoTeam Team1 TeamA

##### Changing the teaming mode

The Windows PowerShell options for teaming mode are:

* SwitchIndependent
* Static
* LACP

These options are described further in Section 3.3.

To change Team1 to an 802.1ax LACP team, the Windows PowerShell is:

Set-NetLbfoTeam Team1 ‑TeamingMode LACP

The “-TeamingMode” flag can be abbreviated “-TM”, as in

Set-NetLbfoTeam Team1 –TM LACP

**Note**: For security reasons teams created in VMs may only operate in SwitchIndependent mode.

##### Changing the load distribution algorithm

The Windows PowerShell options for load distribution algorithm are:

* TransportPorts
* IPAddresses
* MacAddresses
* HyperVPort
* Dynamic

The first three of these options represent the address hashing alternatives presented in Section 3.4.

* TransportPorts, the default mode, is the same as what the UI calls **Address Hash** mode. In this mode the hash will be created using the TCP or UDP ports and the source and destination IP addresses as parameters to the hash creation algorithm. This creates the most granular traffic streams so it will normally result in the most even load distributions.
* IPAddresses only uses the source and destination IP addresses as inputs to the hash creation function. As a result fewer streams will traffic will be identified. This will cause the streams to be larger and may result is a less well distributed traffic load than would be seen with TransportPorts. There is one situation where this mode is particularly useful: the traffic is either not TCP/UDP or the TCP/UDP headers are hidden. This would include, for example, a case where all traffic flowing across the team is IPsec traffic. *Note, however, that the TransportPorts mode will fall back to this mode whenever the Transport Ports are not visible in the packets so there is really no need to set this mode.*
* MacAddresses seeds the hash creation algorithm with the source and destination MAC addresses. Functionally this is very similar in behavior to the IPAddresses option as there is usually a one-to-one correspondence between an IP address and a MAC address. This mode is really only useful if the traffic flowing across the team is not IP traffic. *Note, however, that the IPAddresses mode will fall back to this mode whenever the IP addresses are not visible in the packets so there is really no need to set this mode.*

The hash algorithm will always drop down to the level of the information available. For example, if the mode is set to TransportPorts but the traffic is all IPsec secured traffic, the teaming algorithm will automatically use IPAddresses mode for the IPsec traffic since the TCP/UDP ports are not visible to the team. NIC Teaming will continue to use the better modes for traffic that has the required information visible in the packets.

To change Team1’s Load balancing algorithm to Hyper-V Port, the Windows PowerShell is:

Set-NetLbfoTeam Team1 ‑LoadBalancingAlgorithm HyperVPort

The “-LoadBalancingAlgorithm” flag can be abbreviated “-LBA”, as in

Set-NetLbfoTeam Team1 ‑LBA HyperVPort

To change the Teaming mode and Load balancing algorithm at the same time,

Set-NetLbfoTeam Team1 ‑TM LACP ‑LBA HyperVPort

**Note**: Teams created in VMs may not use the HyperVPort load distribution algorithm.

##### Adding new members to the team

To add NIC1 to Team1 the Windows PowerShell command is:

Add-NetLbfoTeamMember NIC1 Team1

##### Removing members from the team

To remove NIC1 from Team1 the Windows PowerShell command is:

Remove-NetLbfoTeamMember NIC1 Team1

##### Setting a team member to be the Standby Adapter

A team member can be set as the Standby Adapter through Windows PowerShell:

Set-NetLbfoTeamMember NIC4 -AdministrativeMode Standby

At most one team member may be in standby mode at any point in time. If a different team member is already in standby mode that team member must be returned to active mode before this Windows PowerShell cmdlet will succeed.

#### Adding new interfaces to the team

To add a new interface to the team select the Team in the Teams Tile and the Team Interfaces tab in the Adapters and Interfaces tile. Select the Tasks menu in the Adapters and Interfaces tile, then select Add Interface.

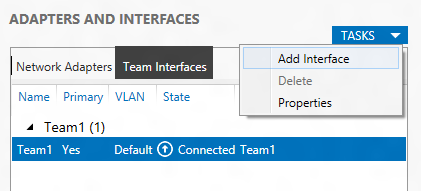


Figure 22 - Selecting Add Interface

Selecting the Add Interface action item pops-up the New team interface dialog box.

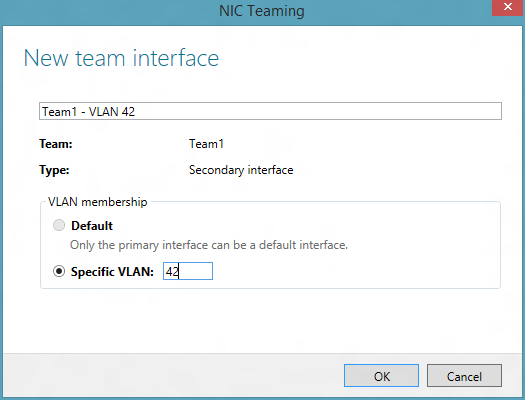


Figure 23 - New team interface dialog box

Since only one team interface, the primary team interface, can be in Default mode, the new team interface **must** have a specific VLAN value. As the specific VLAN value is entered the name of the interface will be modified to be the team name followed by the VLAN value of this team interface. The interface name can be modified to any other name (duplicates are not allowed) if the administrator chooses to do so.

Selecting OK will create the new team interface.

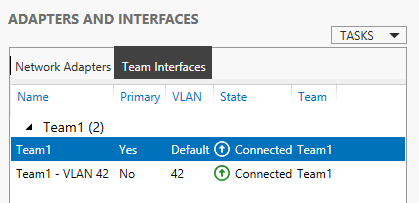


Figure 24 - Team Interface tab after creating new team interface

The Windows PowerShell to add a team interface with VLAN 42 to Team1 is

Add-NetLbfoTeamNIC Team1 -VlanID 42

#### Modifying team interfaces

There are only two modifications that can be done to a team interface:

* change the team interface name and/or
* change the VLAN ID.

To modify the team interface VLAN ID select and then right-click the team interface in the Team Interfaces tab. Select the Properties action item.

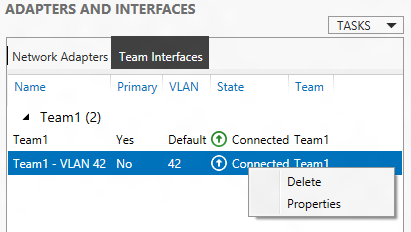


Figure 25 - Selecting a team interface to change the VLAN ID

This pops-up the Network Adapter Properties dialog box. This dialog box has some useful information about the team interface. It also has the box where the new VLAN ID can be entered. If a new VLAN ID is entered and the team name is the one the system provided when the team interface was created the team interface name will be changed to reflect the new VLAN ID. If the team interface name has been previously changed then the team name will not be changed when the new VLAN ID is entered.

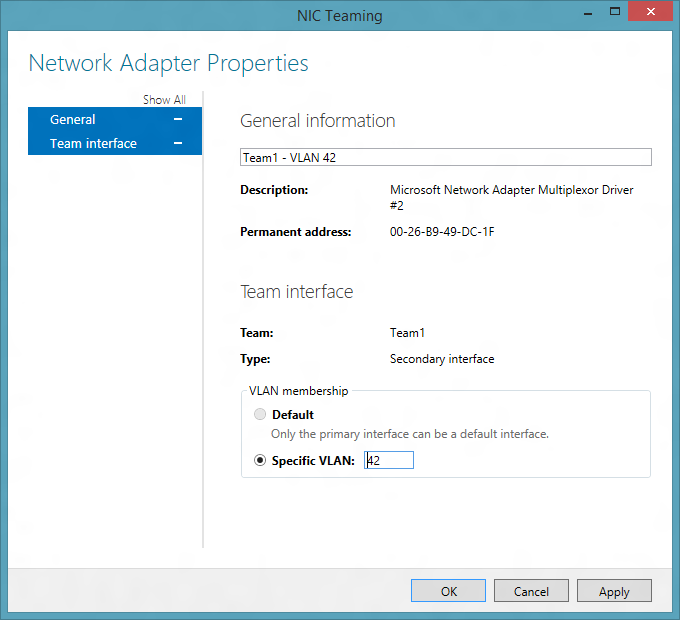


Figure 26- Network Adapter Properties dialog box for team interfaces

To modify a team interface’s VLAN ID in Windows PowerShell

Set-NetLbfoTeamNIC “Team1 ‑ VLAN 42” -VlanID 15

Just as in the UI, changing the VLAN ID will cause the team interface name to change if the team interface name is still the same as the one the system created when the team interface was created. I.e., if the team interface name is <teamName ‑ VLAN xx> where xx is the VLAN ID of the team interface, then the VLAN ID portion of the team interface name will be modified to reflect the new VLAN ID.

If the primary interface was assigned a VLAN and the administrator wants to set it to Default mode the Windows PowerShell command is:

Set-NetLbfoTeamNIC “Team1 ‑ VLAN 12” -Default

Since only the primary team interface can be in Default mode this command will fail for any other team interface.

#### Removing interfaces from the team

To delete a team interface, select and then right-click the team interface in the Team Interfaces tab. Select the Delete team interface action item. (See Figure 25.) A confirmation dialog box will pop-up. Once confirmed the team interface is deleted.

The Primary team interface (i.e., the one that was created when the team was created) can’t be deleted except by deleting the team.

To delete a team interface in Windows PowerShell

Remove-NetLbfoTeamNIC “Team1 ‑ VLAN 42”

### Deleting a team

To delete a team from the server select the team in the Teams tile. Right-click the team and select the Delete team action item.

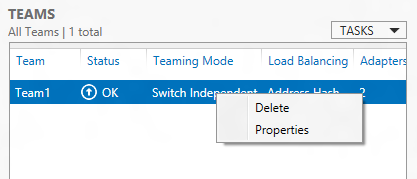


Figure 27 - Deleting a team

A confirmation dialog box will be displayed. Once confirmed the team will be deleted.

To delete a team in Windows PowerShell

Remove-NetLbfoTeam Team1

To remove all teams from the server in Windows PowerShell (i.e., to clean up the server),

Get-NetLbfoTeam | Remove-NetLbfoTeam

### Viewing statistics for a team or team member

If the UI window is sufficiently tall a statistics tile appears at the bottom of the Team tile and the Adapters and Interfaces tile. These statistics windows reflect the traffic of the selected team and selected team member. If you don’t see the statistics try making the UI window a little taller.

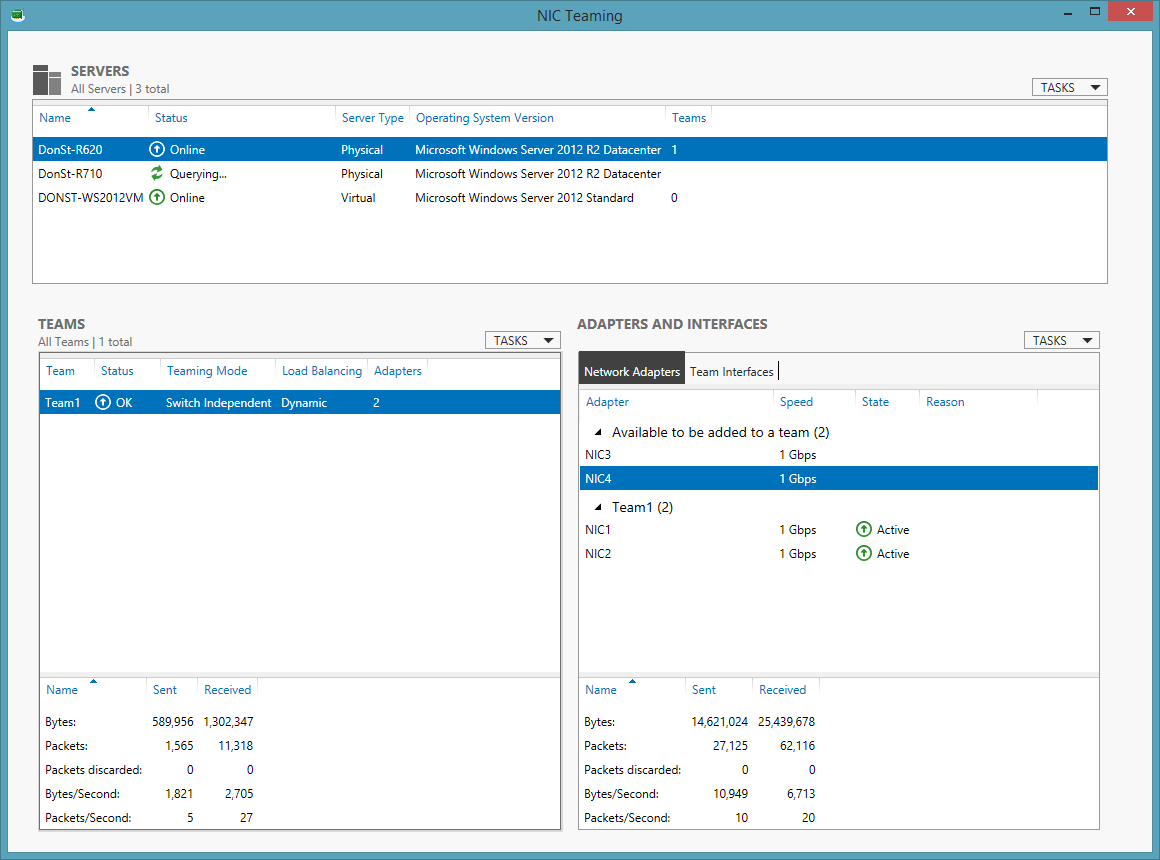


Figure 28- Statistics information for teams and team members

#### Viewing statistics for a team interface

If the Team Interfaces tab is selected in the Adapters and Interfaces tile the statistics at the bottom of the Adapters and Interfaces tile will be those of the selected team interface.

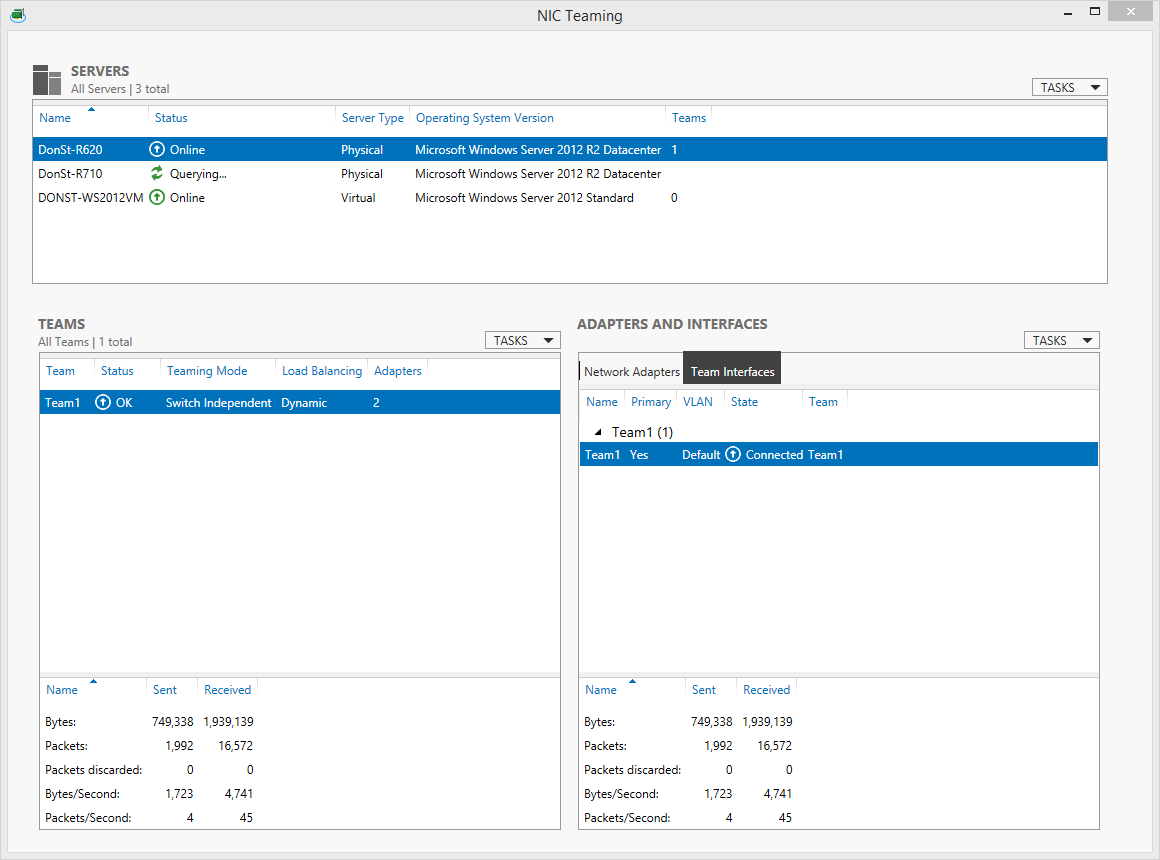


Figure 29- Statistics information for teams and team interfaces

#### Setting frequency of Statistics updates

The frequency of statistics updates and other updates can be set by selection Settings in the Servers tile Tasks menu. Selecting this item brings up the General Settings dialog box.

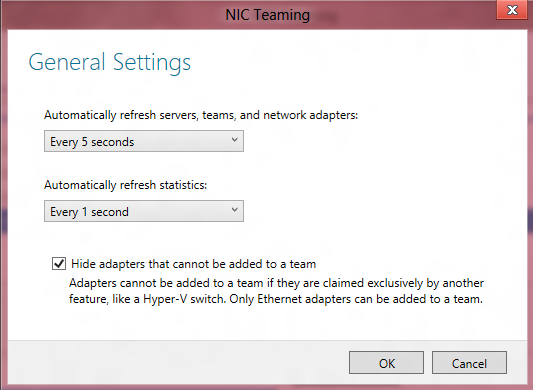


Figure 30 - General settings dialog box

The two drop-down lists in this dialog box allow the user to change how often the UI is refreshed. The settings apply equally to all servers in the servers list.

This menu also allows the administrator to decide whether or not adapters that are not able to be part of a team should be shown in the UI. By default these adapters that can’t be part of a team are not shown in the adapters list.

## Managing a switch-embedded team

Switch-embedded teams are best managed using the Virtual Machine Manager (VMM).

In the event an administrator prefers to use PowerShell to manage a switch-embedded team, here are the cmdlets to use.

### Creating a switch-embedded team

A switch-embedded team must be created at the time the Hyper-V switch is created.

When creating the Hyper-V switch using the New-VMSwitch PowerShell cmdlet, the “‑EnableEmbeddedTeaming” option must be selected. For example, the PowerShell cmdlet shown here

New-VMSwitch -Name TeamedvSwitch -NetAdapterName "NIC 1","NIC 2"

-EnableEmbeddedTeaming $true

will create a Hyper-V switch named TeamedvSwitch with embedded teaming and two initial team members.

### Adding or removing a member of a switch-embedded team

The Set-VMSwitchTeam cmdlet has a -NetAdapterName <string[]> option. To change the team members in a switch embedded team enter the desired list of team members after the ‑NetAdapterName option. Using the team created in section 4.2.1, the cmdlet

Set-VMSwitch -Name TeamedvSwitch -NetAdapterName "NIC 1","NIC 3"

will delete team member “NIC 2” and add new team member “NIC 3”.

Alternatively, the Add-VMSwitchTeamMember and Remove-VMSwitchTeamMember cmdlets may be used.

### Removing a switch-embedded team

A switch-embedded team can only be removed by removing the Hyper-V switch that contains it. Use the Remove-VMSwitch cmdlet to remove the Hyper-V switch.

### Changing the load distribution algorithm of a switch-embedded team

The Set-VMSwitchTeam cmdlet has a -LoadBalancingAlgorithm <VMSwitchLoadBalancingAlgorithm> option. This option takes one of two possible values: HyperVPort, or Dynamic. To set or change the load distribution algorithm for a switch-embedded team use this option. For example,

Set-VMSwitch -Name TeamedvSwitch -VMSwitchLoadBalancingAlgorithm Dynamic

The load distribution mode of switch-embedded teams defaults to “Dynamic.”

### Forcing the affinity of a VM or vNIC to a physical NIC

Windows Server 2016 allows the host administrator to force a host vNIC or a vmNIC to be affinitized to a particular team member. While there are few good reasons to do this, a number of customers have requested this feature in order, for example, to try to ensure that traffic from a given vNIC, e.g., a storage vNIC, uses a particular NIC to send traffic so that it passes through a shorter path to the storage server.

Setting an affinity will not prevent failover to another physical NIC if the selected NIC encounters failures. The affinity will be restored when the selected NIC is restored to operation.

#### Setting up an affinity between a virtual NIC and a physical NIC

The PowerShell cmdlet Set-VMNetworkAdapterTeamMapping will establish an affinity between a virtual NIC (vNIC or vmNIC) and a team member. For example:

Set-VMNetworkAdapterTeamMapping -VMNetworkAdapterName SMB1 –ManagementOS –PhysicalNetAdapterName Ethernet2

will force the traffic from vNIC *SMB1* to be sent and received on physical adapter *Ethernet2*. Similarly,

Set-VMNetworkAdapterTeamMapping -VMName Foo –PhysicalNetAdapterName Ethernet2

will force the traffic from the VM named *Foo* to be sent and received on physical adapter *Ethernet2*.

Run Get-Help Set-VMNetworkAdapterTeamMapping for complete details.

#### Checking the affinity between a virtual NIC and a physical NIC

Of course if an affinity can be established there must be a way to check and see what affinities have been established. Checking the current affinities can be accomplished using the Get‑VMNetworkAdapterTeamMapping PowerShell cmdlet.

Run Get-Help Get-VMNetworkAdapterTeamMapping for complete details.

#### Removing the affinity between a virtual NIC and a physical NIC

If an affinity is no longer desired it can be removed using the Remove‑VMNetworkAdapterTeamMapping PowerShell cmdlet.

Run Get-Help Remove-VMNetworkAdapterTeamMapping for complete details.

# Evolution of Windows Server NIC Teaming

NIC Teaming was introduced to Windows Server in Windows Server 2012. The sections below address the differences release by release.

## NIC Teaming difference between Windows Server 2016 and Windows Server 2012 R2

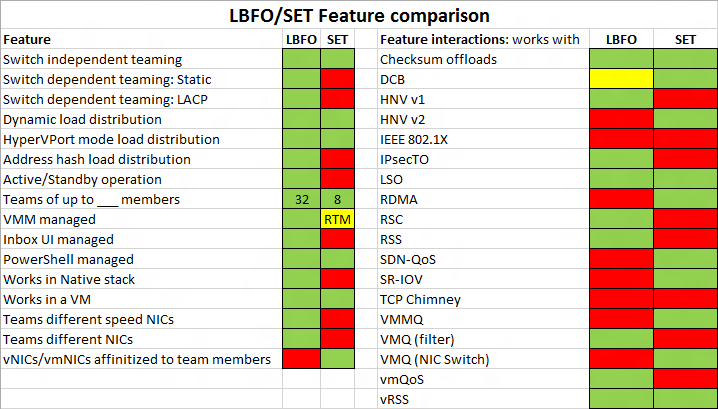
### Stand-alone NIC Teaming

There are no changes of substance to the stand-alone NIC Teaming between Windows Server 2012 R2 and Windows Server 2016.

### Switch Embedded Teaming (SET)

Switch-embedded teaming is new to Windows Server 2016. This feature of the SDN stack provides advanced capabilities such as teaming with RDMA and teaming with SR-IOV. It is a stripped down solution intended to specifically support features of the SDN stack.

### Comparison of features and feature compatibility between stand-along NIC Teaming and SET



## NIC Teaming differences between Windows Server 2012 R2 and Windows Server 2012

The primary user visible changes to NIC Teaming in the Windows Server 2012 R2 release include:

* The addition of load balancing mode “Dynamic” for better performance and better balancing of NIC bandwidth utilization. See Section 3.4.
* Improved interoperability and performance when used with Hyper-V Network Virtualization (NVGRE).
* The addition of an Operating System Version column in the Servers tab of the NIC Teaming administration GUI tool so that the user can determine what level OS is being managed.

# Frequently asked questions (FAQs)

|  |  |
| --- | --- |
| Q1: | Is Active/Standby (Active/Passive) mode supported?  Yes, in stand-alone NIC Teaming. Create a team with two members (Section 4.1.5). Set one as the Standby adapter (see advanced properties drop-down in Section 4.1.5). Teams with more than two members may be created, but only 1 member may be selected as the standby member. The standby member will be activated for transmission any time that one of the active members is faulted. (Packets received on standby members are forwarded up the stack.) |
| Q2: | Do I have to select a standby member to get fault tolerance (failover)?  No. In any team with two or more network adapters if a network adapter fails in an Active/Active configuration, the traffic on that network adapter will gracefully failover to the other network adapters in the team even if none of the other adapters are in standby mode. Selecting a standby adapter does not improve failover latency or gracefulness. |
| Q3: | What is the recommended configuration for Hyper-V?  See section 3.5. |
| Q4: | Why does a NIC failure cause my VM to lose connectivity?  If the VM is running NIC teaming you MUST make sure you’ve followed the steps shown in Section 3.6. |
| Q5: | How does KDNIC interact with NIC Teaming?  The kernel debug network adapter (KDNIC) should not be placed in a team (Rationale: this NIC isn’t meant for production use in servers).  If you have an existing team built over a physical NIC and you enable network debugging over that NIC then the physical NIC is disabled (it shows up as “banged out” in device manager) and the KDNIC virtual adapter carries your network traffic instead. But the team is still attached over the physical NIC (which is now disabled), so the team will treat the NIC as failed. (The purpose of NIC teaming is to gracefully handle NIC failures so your teamed traffic will still flow uninterrupted through the other team members that have network connectivity).  If you need to use network debugging on a machine with NIC teaming you should set up debugging on a NIC that doesn’t participate in teaming.  Summary: network debugging does not coexist with NIC teaming. (This is by design.) |
| Q6: | I can’t see the statistics for my team or for a NIC.  Just make the window a little taller. Statistics disappear if the UI window gets too short. |
|  |  |
| Q7: | How can I tune my Hyper-V host for better CPU utilization by the NIC Teaming software? |
|  | See Section 3.8.1 to see how to select appropriate settings for VMQs. |
| Q8: | Why does the NIC in my VM always show as disconnected when I put it in a team? |
|  | See Section 3.6. Only vmNICs connected to external switches are permitted to be teamed. vmNICs connected to internal or private switches will show as disconnected with error code VirtualSwitchLacksExternalConnectivity. |
| Q9: | My firewall appliances or other core network infrastructure can’t seem to reach my VMs sometimes. Why not? |
|  | Standard practice requires equipment sending IP packets to do a lookup of the IP to MAC address mapping. Some equipment, however, in an effort to optimize their performance, may cache the source MAC address of the last packet in which a particular source IP address arrived. Since MAC address replacement is a common practice in NIC teaming solutions the cached MAC address may not be useable. This can lead to networking issues. We’ve seen this with one popular load balancer vendor in particular but it may occur with others as well. |
| Q10 | I added a vSwitch on top of my team and now I can’t seem to get 10G worth of traffic to a single vNic. Why not? |
|  |  |
|  | RSS is not supported on Hyper-V virtual switch ports exposed to the host operating system and thus all the network processing for that vNic will happen on a single core. The amount of throughput a single core can handle will vary depending on hardware configuration, but 10G is generally much more than a single core can handle. |
|  |  |
| Q11 | What is the difference between a team interface with VLAN 0 and the default team interface? |
|  |  |
|  | The team interface with VLAN 0 passes through only untagged traffic (i.e. traffic that has no VLAN), while the default interface passes through all traffic that is not claimed by any other interface and may handle both tagged and untagged traffic. |
|  |  |
| Q12 | I upgraded my OS version and my teams disappeared. Why? |
|  | In Windows Server 2012 and Windows Server 2012 R2 upgrades do not maintain the team in the new OS. The administrator should delete all teams before upgrading and recreate them after the upgrade completes. |

# Power User tips for the NIC Teaming User Interface (stand-alone NIC Teaming)

Here are some power-user tips for the NIC Teaming UI:

**Quickly return to your most commonly-used servers**. When you add a server to the NIC Teaming UI, you can quickly bring up that server again with the Windows taskbar’s Jump List. Right-click on the taskbar icon for NIC Teaming, and select the server name from the list. That’s it!

**Customizing the UI.** As mentioned in Section 4.1.2, columns may be added or removed from any tile. What that section didn’t mention is that tile contents may be sorted by any column. So, for example, the **Servers** tile could be sorted by the **Server Type** column so that the Physical Servers showed up above (or below) the Virtual Servers (VMs). Network adapters could be sorted by **Speed** or **State**, for example, if the administrator wanted that ordering instead of the default ordering.

**Reset the GUI and start from scratch**. The GUI will remember the servers you’ve added, the position of the main window, which columns you’ve hidden and shown, and any search filters you’ve saved. Most of the time, this is great. But if you ever want to start over with a clean slate, you can run

LbfoAdmin.exe /ResetConfig

to wipe away your old GUI settings, and load up the defaults. Of course, this will **not** affect the teams on any of your servers; it only resets GUI state.

**Navigate the GUI from your keyboard.** Keyboard lovers will appreciate these accelerators. You may have already guessed that F5 refreshes server status. But you can also hit ALT+1 to set focus to the Servers tile, ALT+2 to move to the Teams tile, ALT+3 to activate the Adapters tile, and ALT+4 to focus the Team Interfaces tile.

**Authenticate to a non-domain-joined computer**. If you would like to use different credentials to manage a remote server, you can right-click on the server name and select “Use Credentials…”. This feature is built on the same WinRM management technology as Windows PowerShell, and it can be configured in the same way. By default, non-domain-joined servers are not trusted for authentication. If you would like to allow a server to be trusted, you can use the Windows PowerShell command:

Set-Item WSMan:\localhost\Client\TrustedHosts *myserver* -Concatenate

This will allow you to authenticate against a server named “myserver”, whether you are using the Windows PowerShell Get-Credential cmdlet, or whether you are managing the server in the NIC Teaming UI. Caution: this lowers the default security settings so you should only use this command in a trusted environment.

**NIC Teaming speaks your language.** If you manage servers in a multi-lingual environment, you might notice NetLbfo Windows PowerShell cmdlets or the NIC Teaming UI giving you messages in the wrong language. Some messages (most commonly, error messages) come from the target server, and are selected from the target server’s set of installed languages. You can install your favorite language pack on the target server to change the language of the server-generated messages.

1. Some vendors refer to this configuration as Active/Passive teaming. [↑](#footnote-ref-2)
2. Some vendors have multi-chassis switches that report all the aggregate ports as being from the same switch. This configuration is supported as long as the switch vendor’s management allows all the ports to be placed in the same team. These multi-chassis systems form a Multi-Chassis Link Aggregation Group (MC-LAG) for the purposes of LACP operation. [↑](#footnote-ref-3)
3. See <http://en.wikipedia.org/wiki/Link_aggregation> for the history behind this change in identification of this standard. [↑](#footnote-ref-4)
4. There are security challenges to allowing completely dynamic IEEE 802.1ax operation on a switch. As a result, switches today still require the switch administrator to configure the switch ports that are allowed to be members of such a team. [↑](#footnote-ref-5)
5. If this mode is used for native host teaming it is the same as if there were exactly one vmSwitch port in use. [↑](#footnote-ref-6)
6. There is a known situation in Windows Server 2012 where source MAC replacement is done in LACP teaming mode. This should not affect any operations. This was fixed in the Windows Server 2012 R2 release. [↑](#footnote-ref-7)
7. Advanced users may choose to restrict the switch ports to only passing the VLANs present on the host. While this may slightly improve performance in networks with many VLANs that the local host doesn’t access, it risks creating difficult to diagnose problems when, for example, a VM is migrated to a host and it uses a VLAN not previously present on the destination host. [↑](#footnote-ref-8)
8. RSAT tools for Windows 8.1 are available at http://microsoft.com [↑](#footnote-ref-9)