Hyper-V network driver

Compatibility

This driver is compatible with Windows Server 2012 R2, 2016 and Windows 10.

Features

Checksum offload

The netvsc driver supports checksum offload as long as the Hyper-V host version does. Windows Server 2016 and Azure support checksum offload for TCP and UDP for both IPv4 and IPv6. Windows Server 2012 only supports checksum offload for TCP.

Receive Side Scaling

Hyper-V supports receive side scaling. For TCP & UDP, packets can be distributed among available queues based on IP address and port number.

For TCP & UDP, we can switch hash level between L3 and L4 by ethtool command. TCP/UDP over IPv4 and v6 can be set differently. The default hash level is L4. We currently only allow switching TX hash level from within the guests.

On Azure, fragmented UDP packets have high loss rate with L4 hashing. Using L3 hashing is recommended in this case.

For example, for UDP over IPv4 on eth0:

To include UDP port numbers in hashing:

```
ethtool -N eth0 rx-flow-hash udp4 sdfn
```

To exclude UDP port numbers in hashing:

```
ethtool -N eth0 rx-flow-hash udp4 sd
```

To show UDP hash level:

```
ethtool -n eth0 rx-flow-hash udp4
```

Generic Receive Offload, aka GRO

The driver supports GRO and it is enabled by default. GRO coalesces like packets and significantly reduces CPU usage under heavy Rx load.

Large Receive Offload (LRO), or Receive Side Coalescing (RSC)

The driver supports LRO/RSC in the vSwitch feature. It reduces the per packet processing overhead by coalescing multiple TCP segments when possible. The feature is enabled by default on VMs running on Windows Server 2019 and later. It may be changed by ethtool command:

```
ethtool -K eth0 lro on
ethtool -K eth0 lro off
```

SR-IOV support

Hyper-V supports SR-IOV as a hardware acceleration option. If SR-IOV is enabled in both the vSwitch and the guest configuration, then the Virtual Function (VF) device is passed to the guest as a PCI device. In this case, both a synthetic (netvsc) and VF device are visible in the guest OS and both NIC's have the same MAC address.

The VF is enslaved by netvsc device. The netvsc driver will transparently switch the data path to the VF when it is available and up. Network state (addresses, firewall, etc) should be applied only to the netvsc device; the slave device should not be accessed directly in most cases. The exceptions are if some special queue discipline or flow direction is desired, these should be applied directly to the VF slave device.

Receive Buffer

Packets are received into a receive area which is created when device is probed. The receive area is broken into MTU sized chunks and each may contain one or more packets. The number of receive sections may be changed via ethtool Rx ring parameters.

There is a similar send buffer which is used to aggregate packets for sending. The send area is broken into chunks, typically of 6144 bytes, each of section may contain one or more packets. Small packets are usually transmitted via copy to the send buffer. However, if the buffer is temporarily exhausted, or the packet to be transmitted is an LSO packet, the driver will provide the host with pointers to the data from the SKB. This attempts to achieve a balance between the overhead of data copy and the impact of remapping VM memory to be accessible by the host.

XDP support

XDP (eXpress Data Path) is a feature that runs eBPF bytecode at the early stage when packets arrive at a NIC card. The goal is to increase performance for packet processing, reducing the overhead of SKB allocation and other upper network layers.

hv_netvsc supports XDP in native mode, and transparently sets the XDP program on the associated VF NIC as well.

Setting / unsetting XDP program on synthetic NIC (netvsc) propagates to VF NIC automatically. Setting / unsetting XDP program on VF NIC directly is not recommended, also not propagated to synthetic NIC, and may be overwritten by setting of synthetic NIC.

XDP program cannot run with LRO (RSC) enabled, so you need to disable LRO before running XDP:

```
ethtool -K eth0 lro off
```

XDP_REDIRECT action is not yet supported.