NAME

ptnet - Ethernet driver for passed-through netmap ports

SYNOPSIS

This network driver is included in netmap(4), and it can be compiled into the kernel by adding the following line in your kernel configuration file:

device netmap

DESCRIPTION

The **ptnet** device driver provides direct access to host netmap ports, from within a Virtual Machine (VM). Applications running inside the VM can access the TX/RX rings and buffers of a netmap port that the hypervisor has passed-through to the VM. Hypervisor support for **ptnet** is currently available for QEMU/KVM. Any netmap(4) port can be passed-through, including physical NICs, vale(4) ports, netmap pipes, etc.

The main use-case for netmap passthrough is Network Function Virtualization (NFV), where middlebox applications running within VMs may want to process very high packet rates (e.g., 1-10 millions packets per second or more). Note, however, that those applications must use the device in netmap mode in order to achieve such rates. In addition to the general advantages of netmap, the improved performance of **ptnet** when compared to hypervisor device emulation or paravirtualization (e.g., vtnet(4), vmx(4)) comes from the hypervisor being completely bypassed in the data-path. For example, when using vtnet(4) the VM has to convert each mbuf(9) to a VirtIO-specific packet representation and publish that to a VirtIO queue; on the hypervisor side, the packet is extracted from the VirtIO queue and converted to a hypervisor-specific packet representation. The overhead of format conversions (and packet copies, in same cases) is not incured by **ptnet** in netmap mode, because mbufs are not used at all, and the packet format is the one defined by netmap (e.g., *struct netmap_slot*) along the whole data-path. No format conversions or copies happen.

It is also possible to use a **ptnet** device like a regular network interface, which interacts with the FreeBSD network stack (i.e., not in netmap mode). However, in that case it is necessary to pay the cost of data copies between mbufs and netmap buffers, which generally results in lower TCP/UDP performance than vtnet(4) or other paravirtualized network devices. If the passed-through netmap port supports the VirtIO network header, **ptnet** is able to use it, and support TCP/UDP checksum offload (for both transmit and receive), TCP segmentation offload (TSO) and TCP large receive offload (LRO). Currently, vale(4) ports support the header. Note that the VirtIO network header is generally not used in NFV use-cases, because middleboxes are not endpoints of TCP/UDP connections.

TUNABLES

Tunables can be set at the loader(8) prompt before booting the kernel or stored in loader.conf(5).

dev.netmap.ptnet_vnet_hdr

This tunable enables (1) or disables (0) the VirtIO network header. If enabled, **ptnet** uses the same header used by vtnet(4) to exchange offload metadata with the hypervisor. If disabled, no header is prepended to transmitted and received packets. The metadata is necessary to support TCP/UDP checksum offloads, TSO, and LRO. The default value is 1.

SEE ALSO

netintro(4), netmap(4), vale(4), virtio(4), vmx(4), ifconfig(8)

HISTORY

The **ptnet** driver was written by Vincenzo Maffione <*vmaffione@FreeBSD.org*>. It first appeared in FreeBSD 12.0.