

NetKVM Presentation

Including Virtio / Virtio-net / NetKVM drivers

Wenkang Ji

wji@redhat.com



Agenda

- 1. Trap-And-Emulate && Hardware-Assisted Virtualization
- 2. Introduction of Virtio-net
 - Virtio paravirtualization framework introduction
 - Request transmission between virtio front-end and back-end
- 3. Netkvm feature cases
 - Parameter default values and ranges
 - Parameter introduction
- 4. Internal-kvm-guest-drivers-windows
 - VirtIO adapter Properties
- 5. Reference

Trap-And-Emulate && Hardware-Assisted Virtualization



“陷入模拟” (Trap-And-Emulate)

x86虚拟化厂商的解决方案:

- VMware - “Binary Translation”



提前发现敏感指令并通过插入断点来截获，
交由VMM来解释执行

- XenSource - “Paravirtualization”

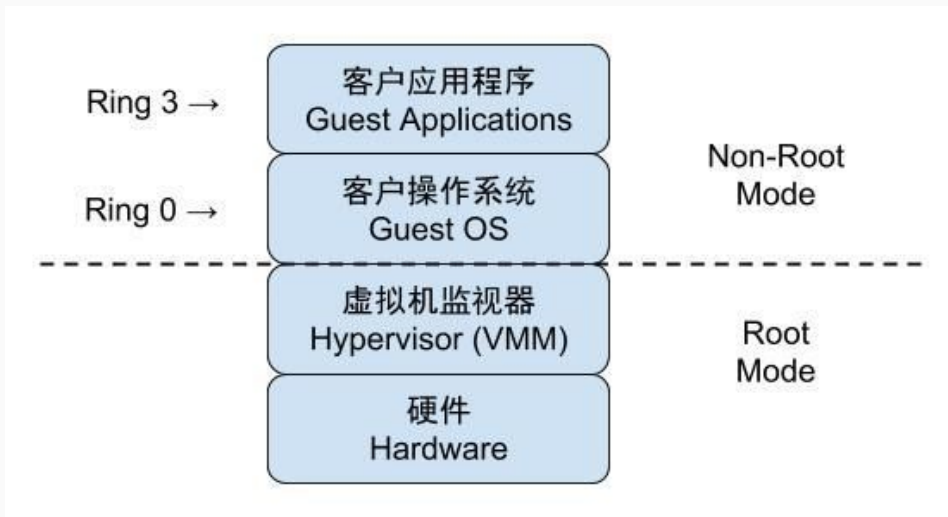


直接修改客户操作系统代码，将敏感指令改为Trap Call以通知VMM来处理

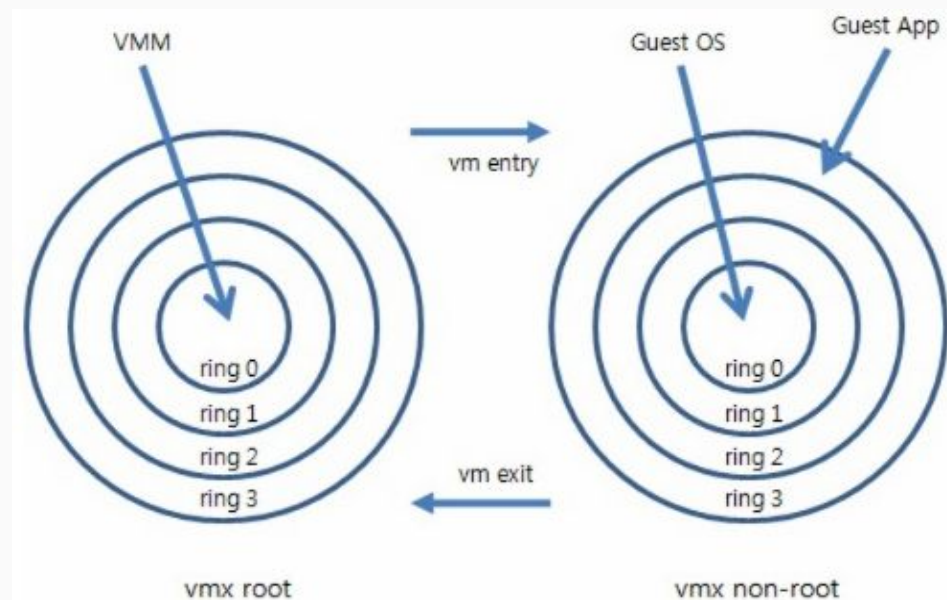
这种在软件上煞费苦心的做法解决了x86指令集的难题，但也影响了VMM的复杂度和性能，而“Paravirtualization”更是难以运用在Windows等闭源操作系统上。

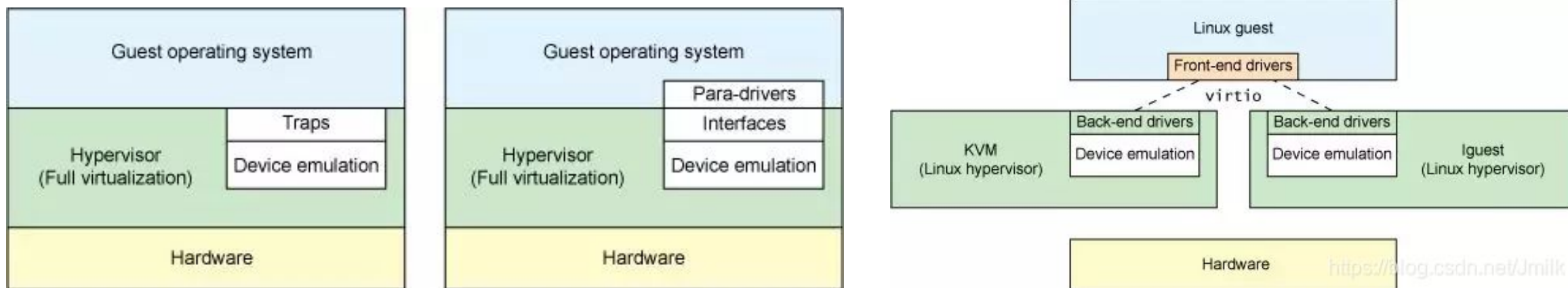


基于硬件辅助技术的虚拟化



以VT-x为例，Intel为处理器增加了“**根**”和“**非根**”两种模式；并且经过对相关指令的重新设计，原本不能通过先陷入后模拟方式执行的指令都可以顺利执行；最终，得益于硬件辅助技术，VMM不再需要对指令进行复杂的模拟。





High Performance: VirtIO eliminates the traps in full-virtualization mode, allowing the GuestOS to interact directly with the device emulation in the Hypervisor through VirtIO interfaces.

Low Overhead(低开销): VirtIO optimizes CPU performance by reducing frequent switches between kernel mode and user mode, as well as minimizing the performance overhead caused by frequent VM exits and entries.

Standardization: VirtIO provides a unified virtual device interface standard that can be applied across various virtualization solutions.

Virtio paravirtualization framework

Virtio 是 KVM 虚拟环境下针对 I/O 虚拟化的最主要的一个通用框架，KVM 使用半虚拟化框架 Virtio 前端后端驱动模块互相协作的方式，减少 I/O 操作时系统中断和权限转换所带来的开销，同时由于半虚拟化采用的数据描述符机制，用数据描述符来传递数据信息而不是进行跨主机数据拷贝，使得数据读写只发生在共享内存区域，减少冗余的数据拷贝。使用 Virtio 后的 I/O 性能有很大提升。

When talking about virtio-networking we can separate the discussion into two layers:

在讨论虚拟网络时，我们可以将讨论分为两个层面：

Control plane - Used for capability exchange negotiation between the host and guest both for establishing and terminating the data plane.

控制平面 - 用于主机和访客之间的能力交换协商，以建立和终止数据平面。

Data plane - Used for transferring the actual data (packets) between host and guest.

数据平面 - 用于在主机和访客之间传输实际数据(数据包)。

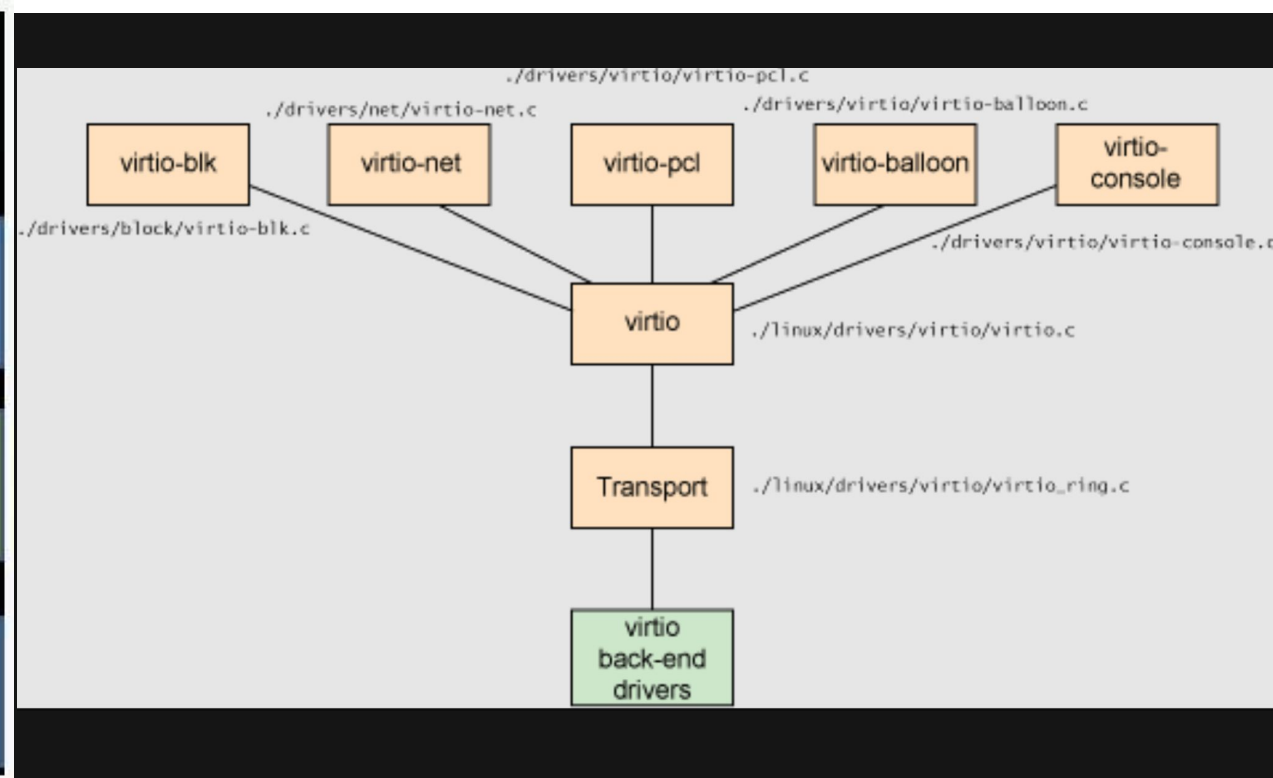
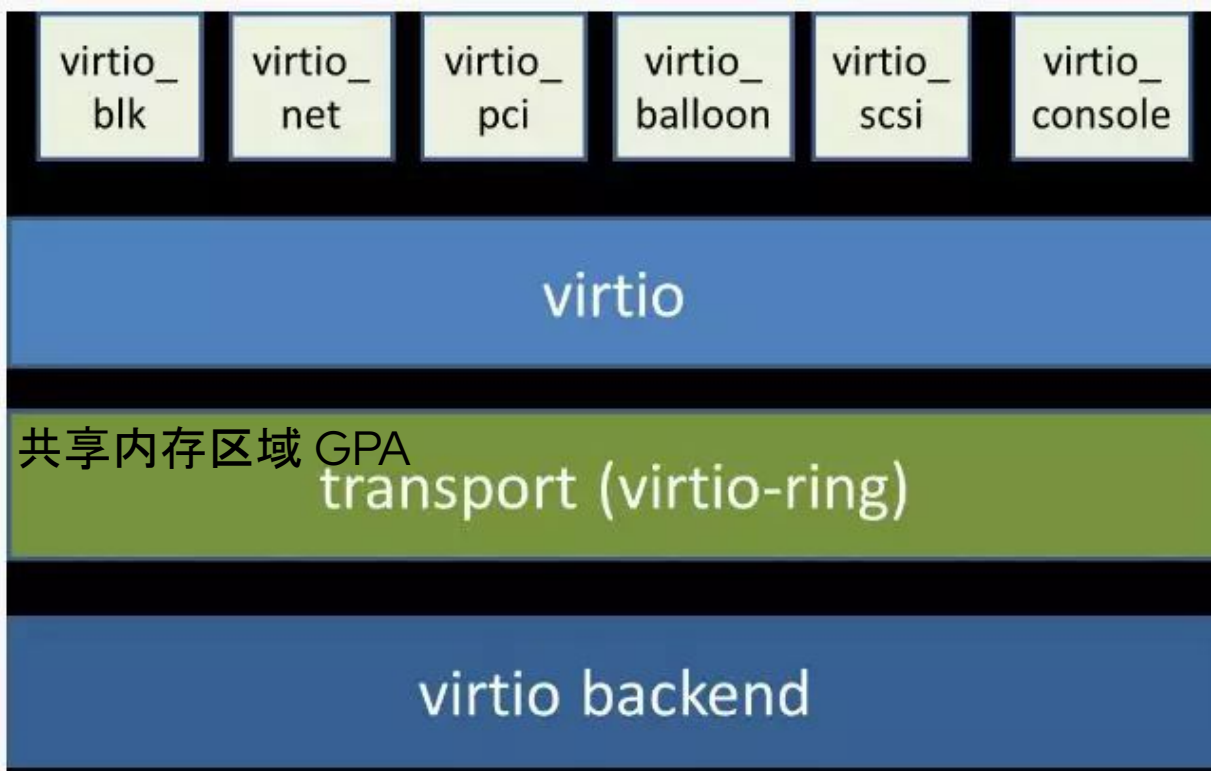
控制平面只需执行 virtio 规范，让 vhost-net 内核模块和 qemu 进程进行通信，然后转发到客户机，最后再转发到 virtio-net。

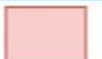
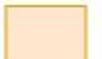
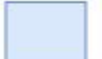


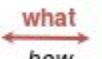

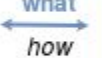
vhost-net 使用 vhost 协议建立框架，然后用于数据平面，使用共享内存区域在主机和客户机内核之间直接转发数据包。包括了 Multi-Queues 的概念！

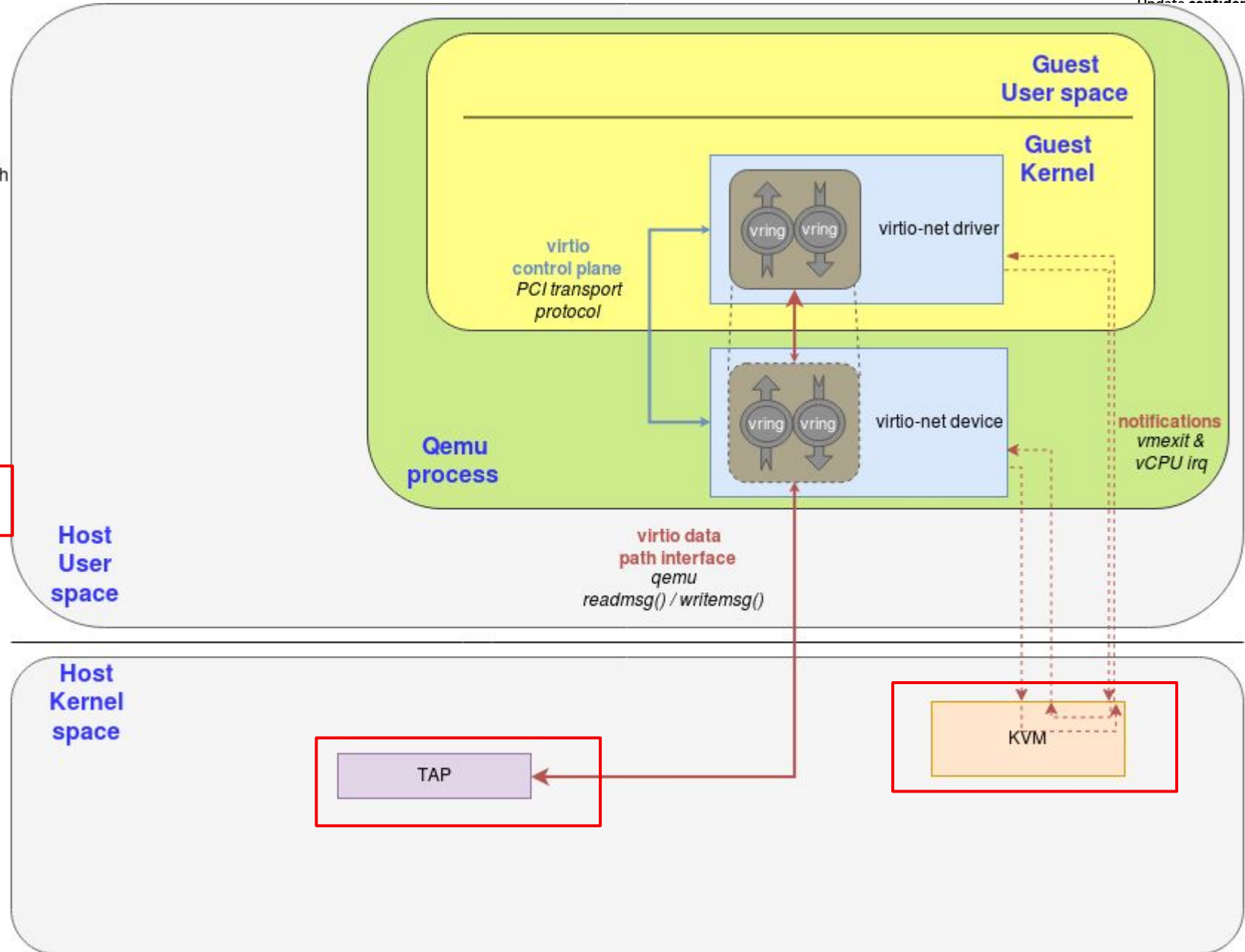
Introduction of Virtio-net

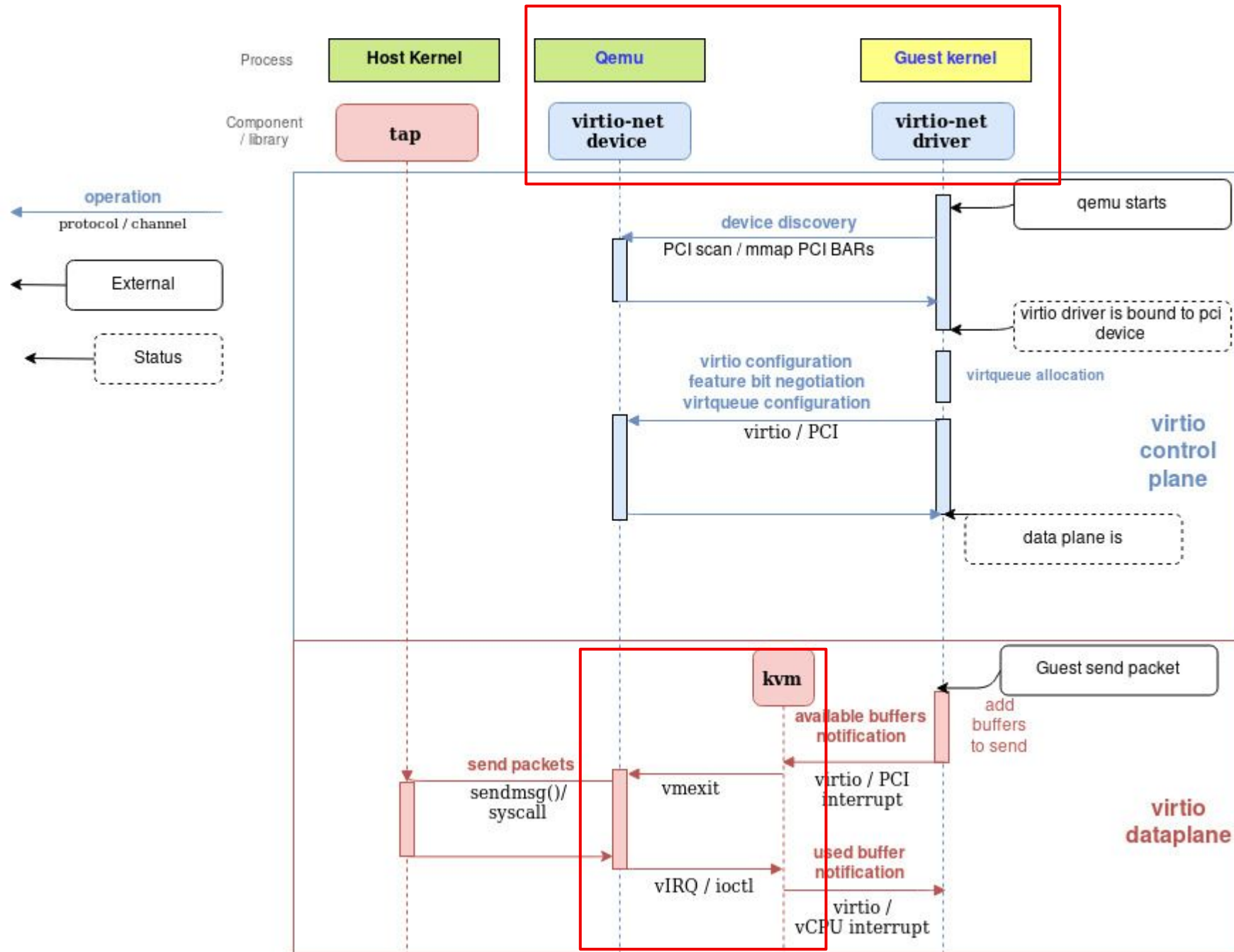


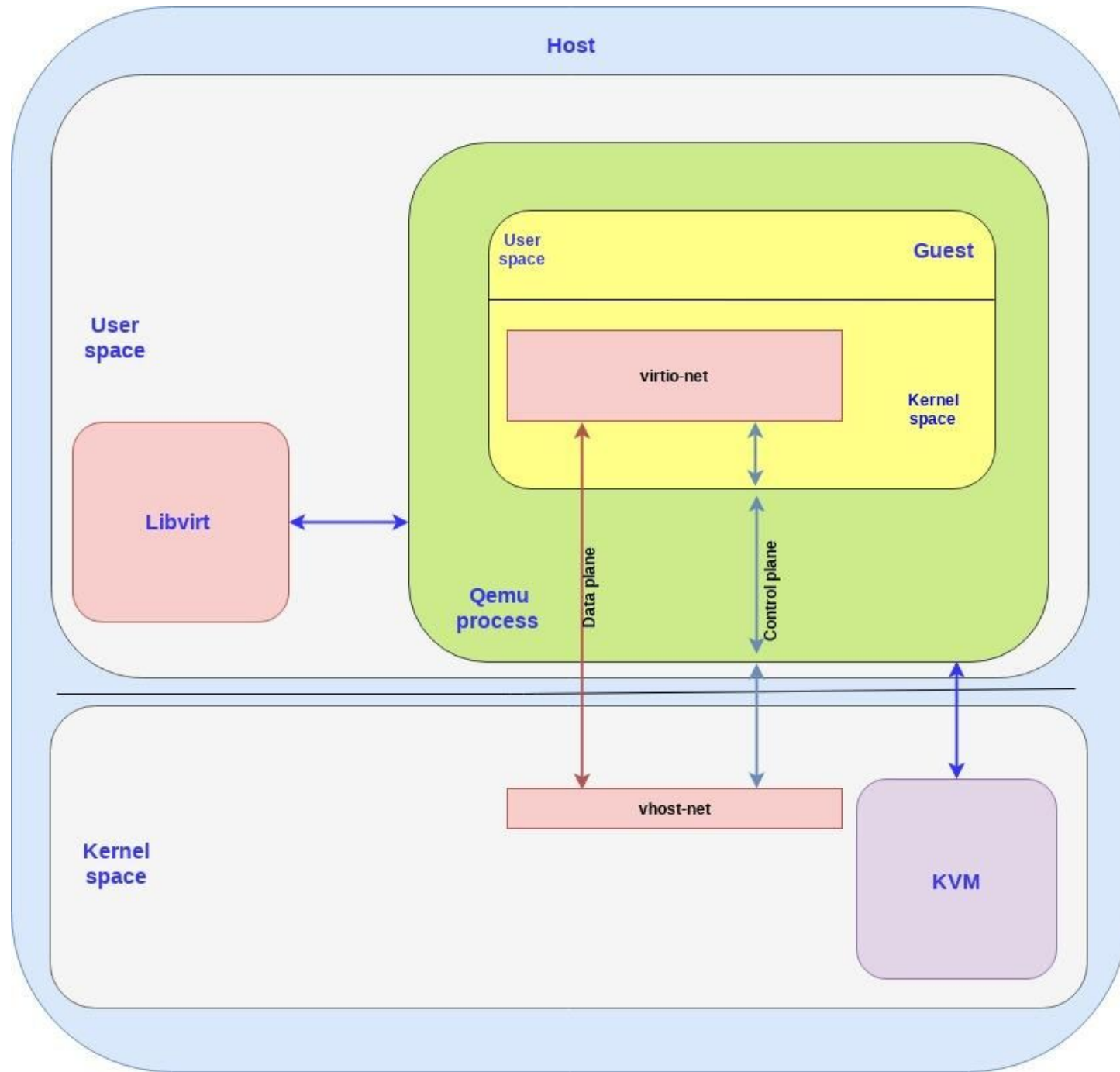
Virtio paravirtualization framework introduction



Symbol	Meaning
	virtio data path element
	non-virtio data path element
	virtio control path element
	port
	virtio shared memory
	data path
	interrupts / notifications
	control path





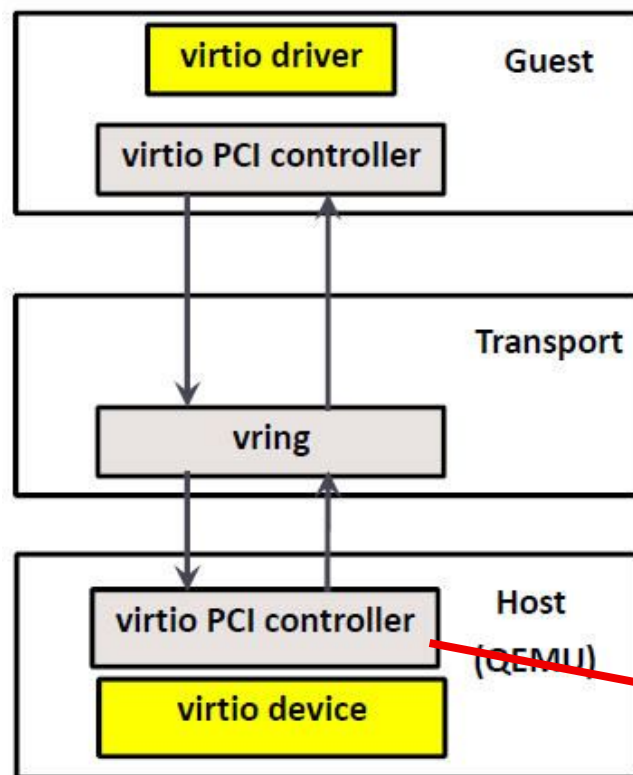


Request transmission between virtio front-end and back-end

V

M

T



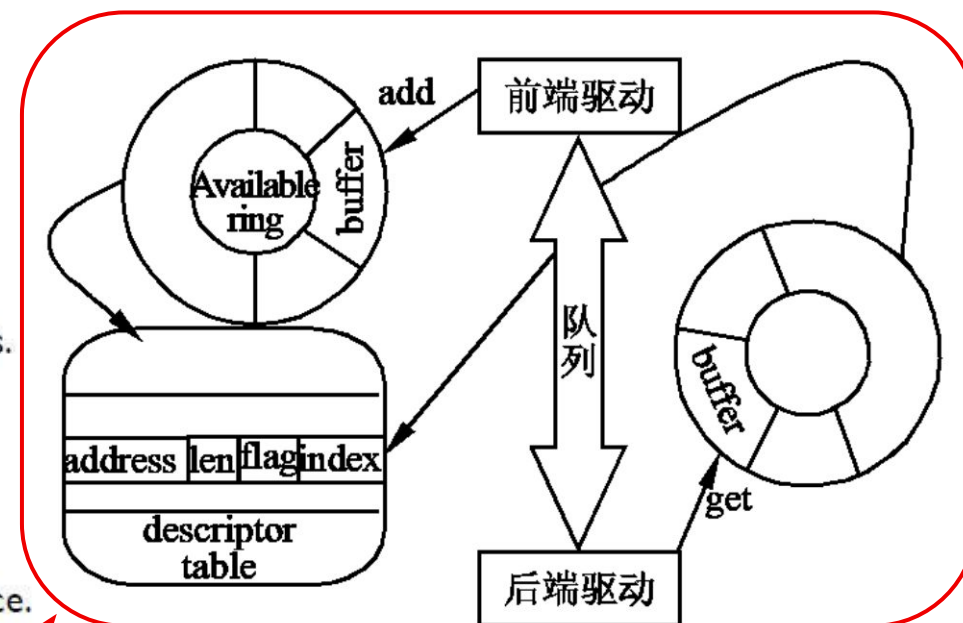
Front-end

A kernel module in guest OS.
Accepts I/O requests from user process.
Transfer I/O requests to back-end.

Back-end

A device in QEMU.
Accepts I/O requests from front-end.
Perform I/O operation via physical device.

- Virtqueues (per device)
- Vring (per virtqueue)
- Queue requests



The virtio interface consist of the following mandatory parts ([virtio1.1 spec](#)):

virtio 界面由以下必备部分组成 (virtio1.1 规范):

- Device status field 设备状态栏
- Feature bits 功能位
- Notifications 通知
- One or more virtqueues 一个或多个虚拟队列

这些部分共同协作, 使得 vring 传输和设备配置能够作为 PCI 设备呈现。

Netkvm feature cases

the vendor implementing HW NICs based on virtio spec

lshw -c network -businfo

```
pci@0000:87:00.0 ens7f0 network Ethernet
```

```
Controller XL710 for 40GbE QSFP+
```

```
pci@0000:87:00.1 ens7f1 network Ethernet
```

```
Controller XL710 for 40GbE QSFP+
```



Qemu level

Backend Device

-netdev **tap**,script=/etc/qemu-ifup,downscript=no,id=hostnet1,**vhost=on**

-netdev **tap**,script=/etc/qemu-ifup,downscript=no,id=hostnet1,**vhost=on**

-netdev **tap**,id=dev1,**vhost=on**,script=/etc/ifup_script,downscript=/etc/ifdown_script

Frontend Device

-device **virtio-net-pci**,netdev=hostnet1,id=net1,mac=00:52:11:36:3f:00,bus=pci.0

-device **virtio-net-pci**,netdev=hostnet1,id=net1,mac=00:52:11:36:3f:00,bus=pci.0

-device **virtio-net-pci**,netdev=dev1,mac=9a:e8:e9:ea:eb:ec,id=net1

参数名	意义	取值范围	默认值	需要与什么参数一起使用
id	标识此 <code>-netdev</code> 配置的唯一 ID	任意字符串	无	<code>-device</code> 中的 <code>netdev</code>
vhost	启用或禁用 vhost-net	on 或 off	off	无
script	指定启动 tap 设备时执行的脚本路径	有效的文件路径	无	type=tap
downscript	指定关闭 tap 设备时执行的脚本路径	有效的文件路径	无	type=tap
vhostforce	强制启用或禁用 vhost-net	on 或 off	off	vhost
queues	指定多队列的数量	正整数	1	mq
poll-us	设置 tap 设备轮询新数据间隔, 以微秒为单位	正整数	无	type=tap
host_mtu	设置虚拟网络设备的最大传输单元 (MTU)	正整数	无	无
type	指定 <code>-netdev</code> 类型	tap, user, bridge, socket, vde, l2tpv3, netmap, vhost-user, hubport	无	无

参数名	意义	取值范围	默认值	类别 Update confidential designator here
netdev	引用 <code>-netdev</code> 配置的 ID, 连接该网络设备	<code>-netdev</code> 配置的有效 ID	无	网络设备相关参数
mac	指定网络设备的 MAC 地址	有效的 MAC 地址	随机生成	网络设备相关参数
bus	指定设备连接到哪个 PCI 总线	有效的总线名称	<code>pci.0</code>	网络设备相关参数
id	标识此 <code>-device</code> 配置的唯一 ID	任意字符串	无	网络设备相关参数
type	指定 <code>-device</code> 类型 (见详细类型示例)	<code>virtio-net-pci, e1000, rtl8139, ne2k_pci, vmxnet3, pcnet</code> , 等	无	网络设备相关参数
mq	启用或禁用多队列支持	<code>on</code> 或 <code>off</code>	<code>off</code>	缓冲区和队列相关参数
queues	指定多队列的数量	正整数	1	缓冲区和队列相关参数
vectors	指定中断向量的数量	正整数	3	缓冲区和队列相关参数
indirect_desc	启用或禁用间接描述符	<code>on</code> 或 <code>off</code>	<code>off</code>	缓冲区和队列相关参数

status	启用或禁用设备状态报告	on 或 off	off	网卡状态相关参数
speed	指定网络设备的速度	正整数	无	网卡状态相关参数
duplex	指定全双工或半双工模式	full 或 half	full	网卡状态相关参数
iommu_platform	启用或禁用 IOMMU 平台支持	on 或 off	off	IOMMU 相关参数
ats	启用或禁用地址转换服务 (ATS)	on 或 off	off	IOMMU 相关参数
disable-modern	禁用现代设备接口	on 或 off	off	其他不常用或辅助功能参数
disable-legacy	禁用传统设备接口	on 或 off	on	其他不常用或辅助功能参数
ctrl_vq	启用或禁用控制虚拟队列	on 或 off	off	其他不常用或辅助功能参数
ctrl_vlan	启用或禁用 VLAN 控制	on 或 off	off	其他不常用或辅助功能参数

Test cases

- 网络配置与测试
- 驱动安装与验证
- 性能测试
- 文件传输测试
- 热插拔测试
- 虚拟机迁移测试
- 功能性测试
- 杂项测试

分类	测试名称	工具/任务	说明
网络配置与测试	网络地址配置和测试	Change the network address & ifconfig	Update confidential designator here 修改网络地址, 配置网络接口的 IP 地址和子网掩码
网络配置与测试	网络连通性测试	Ping	测试本地和远程主机之间的连通性及延迟
驱动安装与验证	Virtio 协议驱动程序安装和配置测试	netcfg -v -l vioprot.inf -c p -i VIOPROT	使用 netcfg 命令安装和配置 Virtio 协议驱动程序
驱动安装与验证	驱动程序安装验证测试	netkvm device driver can be installed correctly	验证 netkvm 驱动程序是否正确安装
驱动安装与验证	驱动程序签名验证测试	驱动签名验证	验证驱动程序签名
驱动安装与验证	驱动程序升级降级测试	驱动升级降级	测试驱动程序的升级和降级
性能测试	网络性能测试	NTttcp	测试和测量网络吞吐量和性能
性能测试	网络性能测试	netperf / iperf	测试网络带宽、延迟和其他性能指标
性能测试	多队列性能测试	vhost 和多队列相关参数	测试多队列和 vhost 相关的网络性能
文件传输测试	文件传输测试	SCP	通过 SSH 协议在本地和远程主机之间安全地传输文件
热插拔测试	热插拔测试	Plug / Unplug	测试网络设备的插拔和功能
迁移测试	虚拟机迁移测试	Migration	将虚拟机从一个主机迁移到另一个主机, 保持虚拟机运行状态
功能性测试	功能性测试	legacy / transitional / modern 模式	测试网卡的三种模式: legacy / transitional / modern
功能性测试	网络流量回放测试	tcpreplay	将捕获的网络流量回放放到网络中, 用于测试和调试
杂项测试	杂项测试	vlan, pxe, MTU, MSI, multiqueues, mq, vhost	测试各种杂项功能, 包括 VLAN、PXE、MTU、MSI、多队列等



Netkvm feature cases

Parameter default values and ranges

Parameter introduction

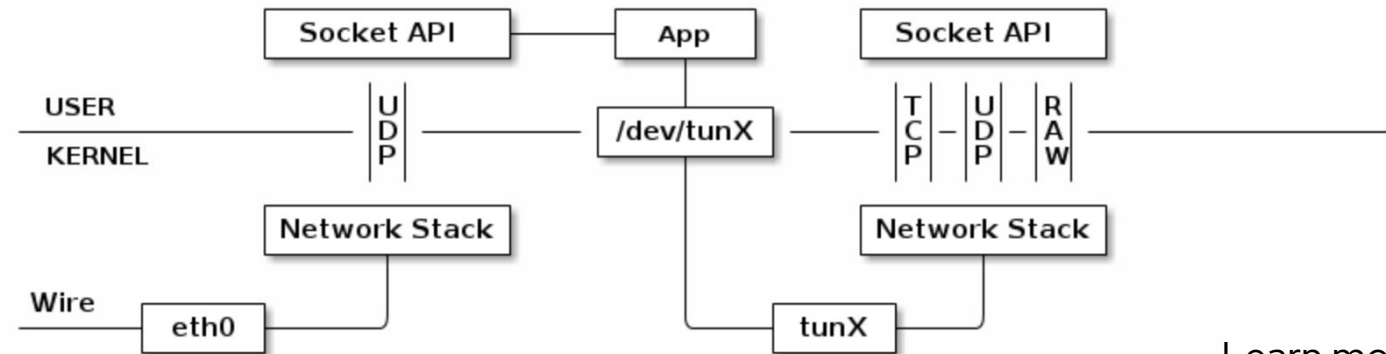
Driver module

- **virtio-net-pci**
- e1000e
- e1000
- rtl8139

TUN device / TAP device

TUN device

Simulate an UDP VPN process.



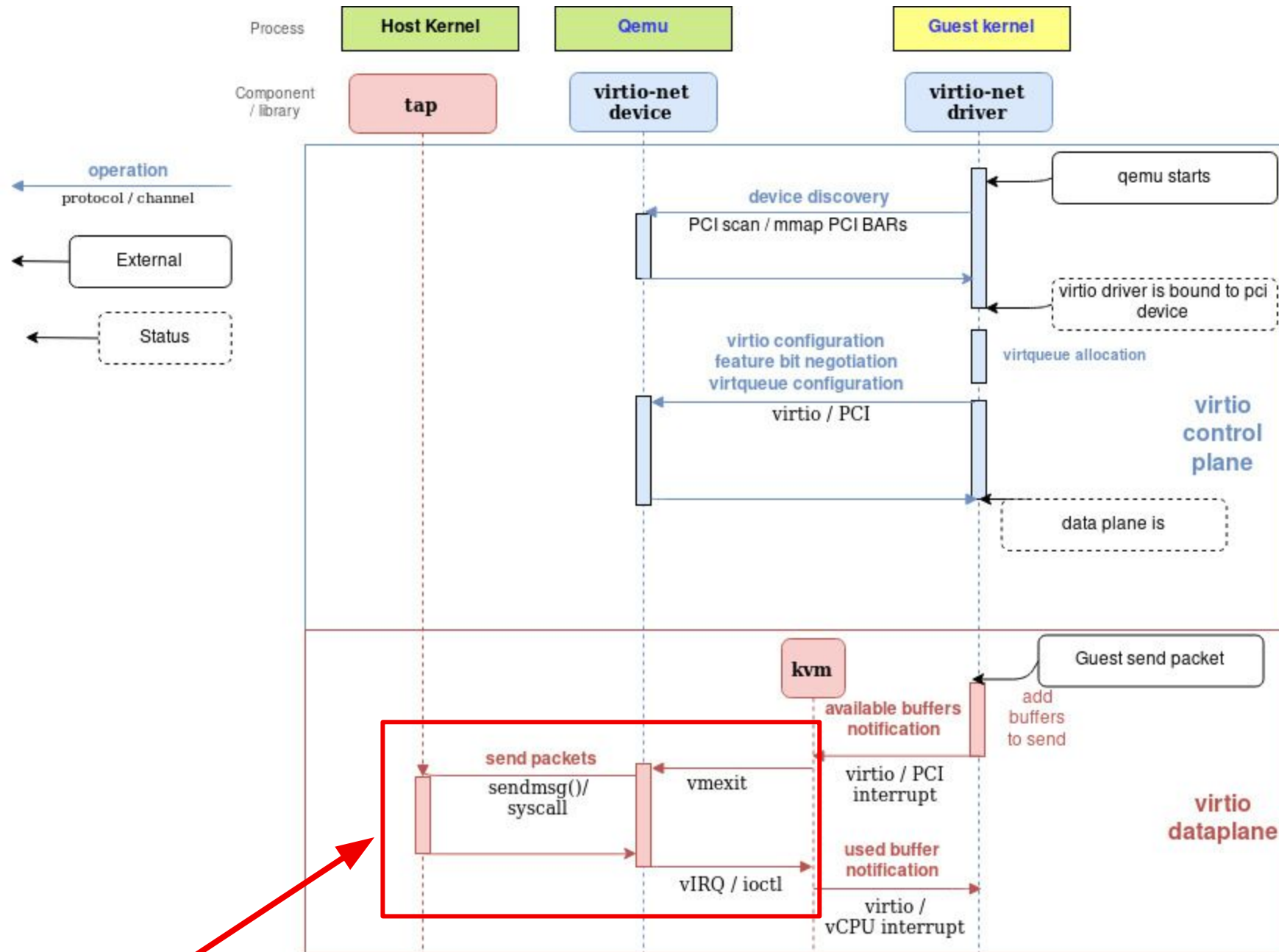
TAP device

Like TUN device, here is a list of the main differences between tun and tap.

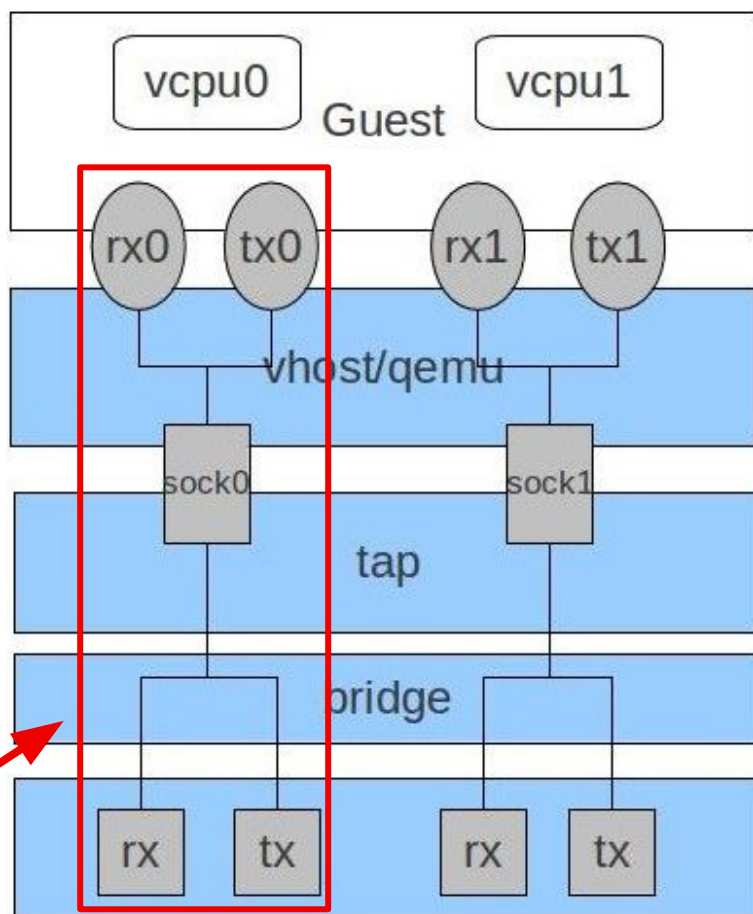
1. /dev/tunX works on IP layer (ip_forward)
2. /dev/tapX work on MAC layer (bridge, MAC broadcast)

Learn more:

- MacVLan - bridge / host-only
- MacVTap
- socket
- hubport
- user
- l2tpv3
- bridge
- vhost-user



Multi-Queues



vectors=queues*2+2

queues: 表示接收和发送队列的数量。

Queues: 每个 vCPU 有一个接收队列和一个发送队列, 总共 2 个队列

Vectors: queues*2 每个 CPU 需要两个 MSI vector, 一个用于接收, 一个用于发送。Vectors +2: 额外的两个 MSI vector 用于控制操作。包括中断管理、错误处理等。

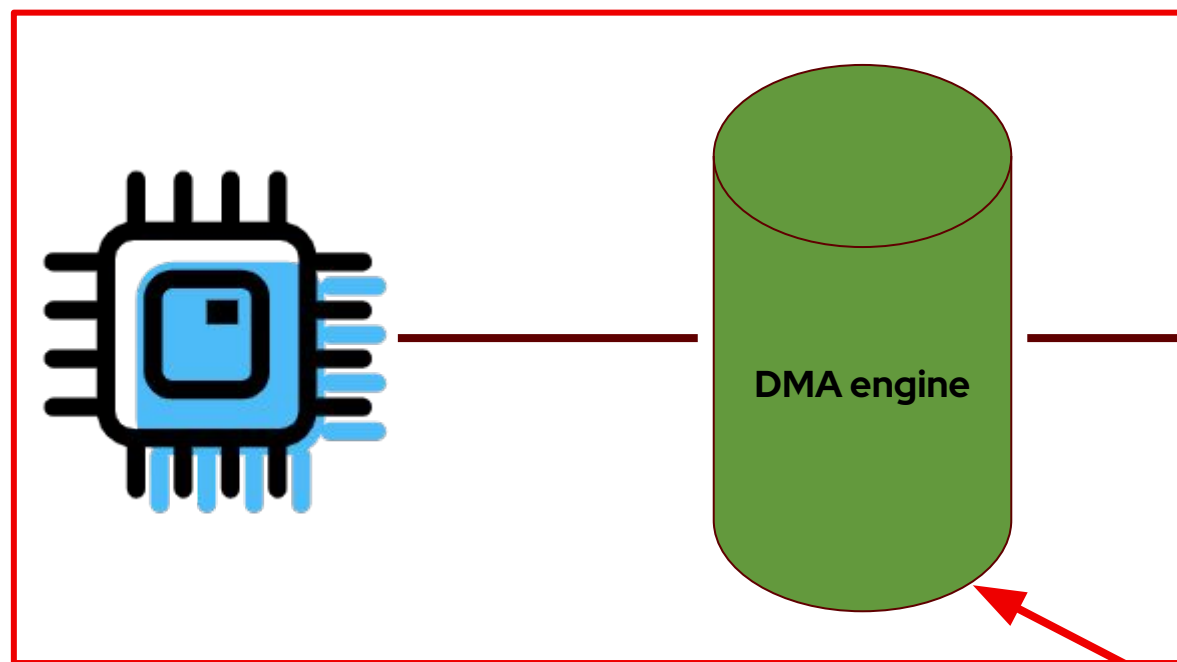
performance benefit:

- 队列数等于 vCPU 数。这是因为多队列支持优化了 RX 中断亲和性和 TX 队列选择, 以使特定队列对特定 vCPU 专用。

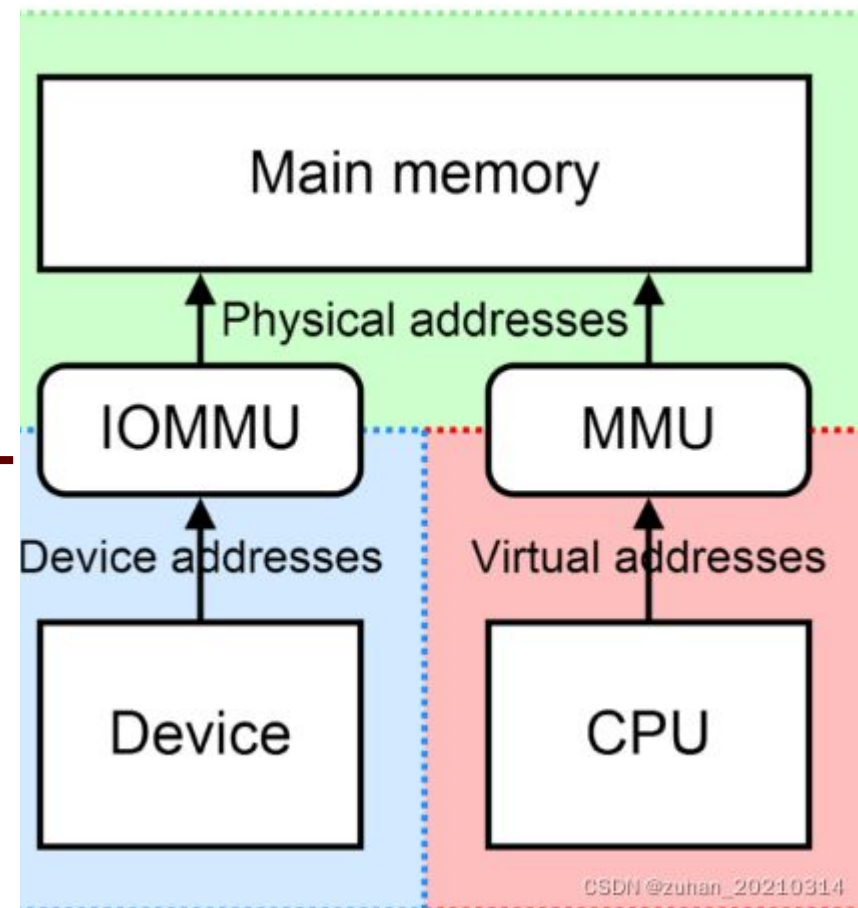
Limitations:

- 客户操作系统只能使用约 200 个 MSI 向量。每个网卡队列都需要一个 MSI 向量, 任何 virtio 设备或分配的 PCI 设备也是如此。使用多个 virtio 网卡和 vCPU 定义实例可能会导致触及客户 MSI 限制。
- 启用 virtio-net 多队列会增加网络总吞吐量, 但同时也会增加 CPU 消耗。

iommu_platform & ats

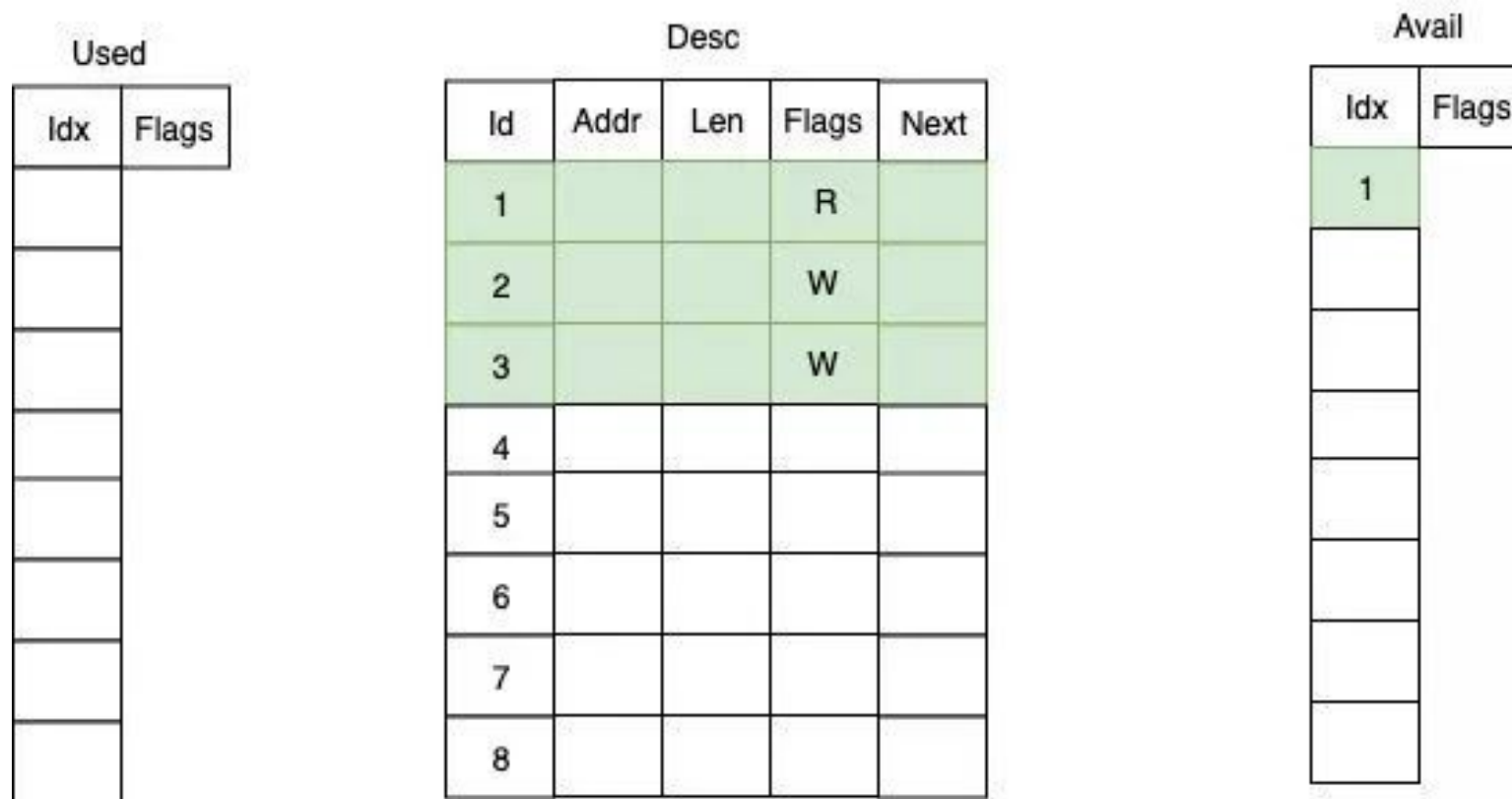


设备需要执行地址转换

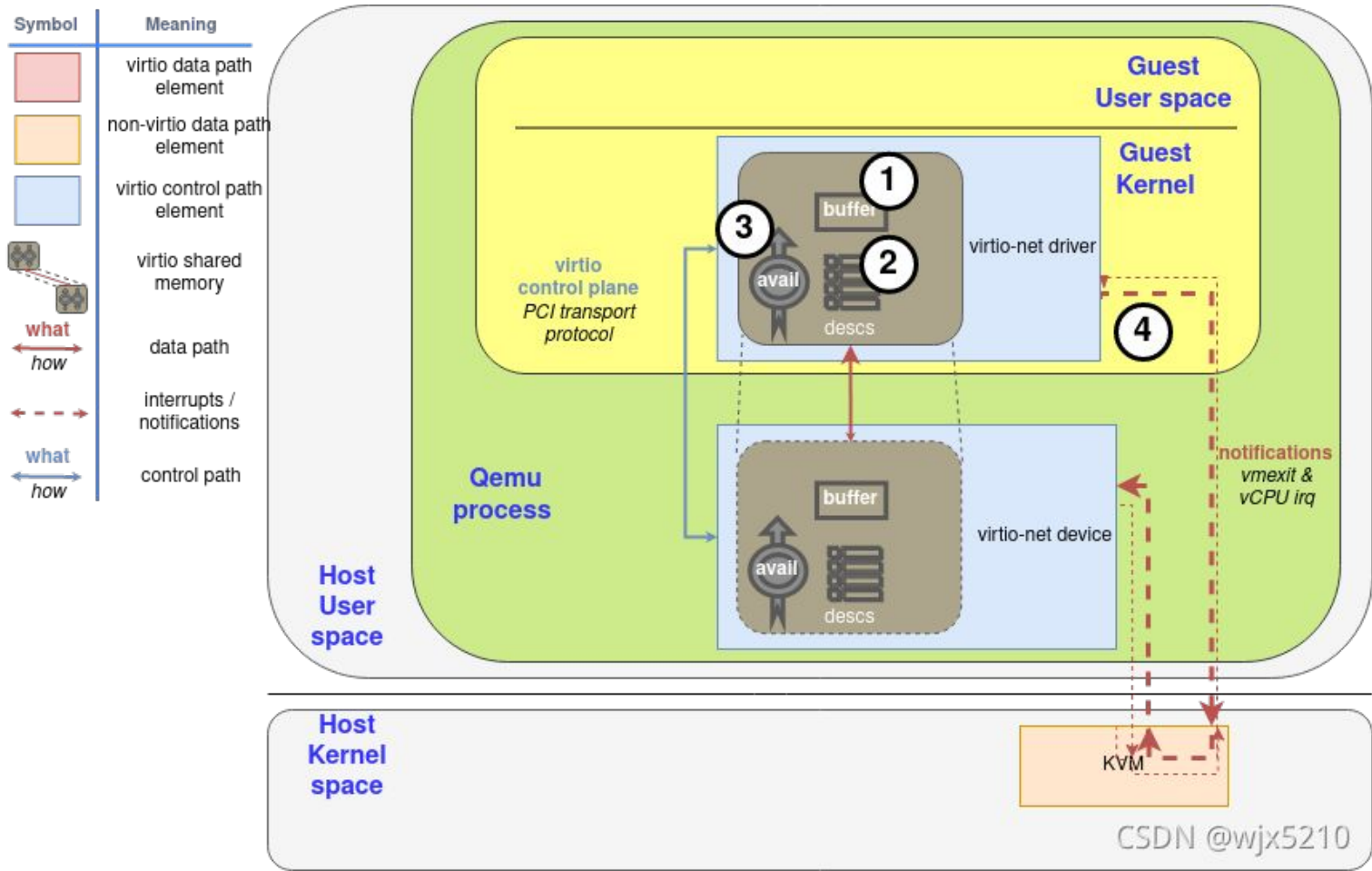


CSDN @zuhan_20210314








indirect_desc (间接描述符: 向设备提供大量数据)

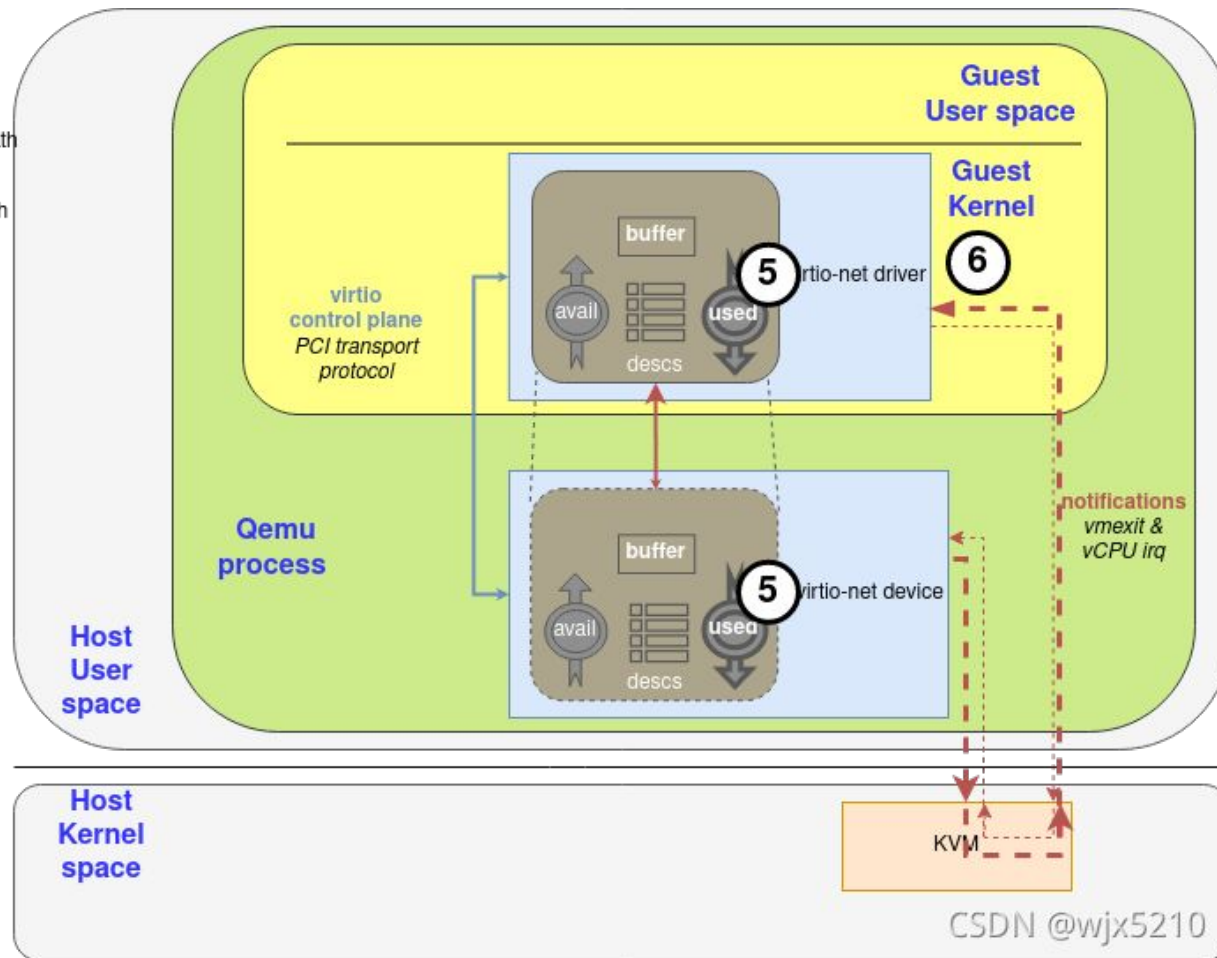


indirect_desc

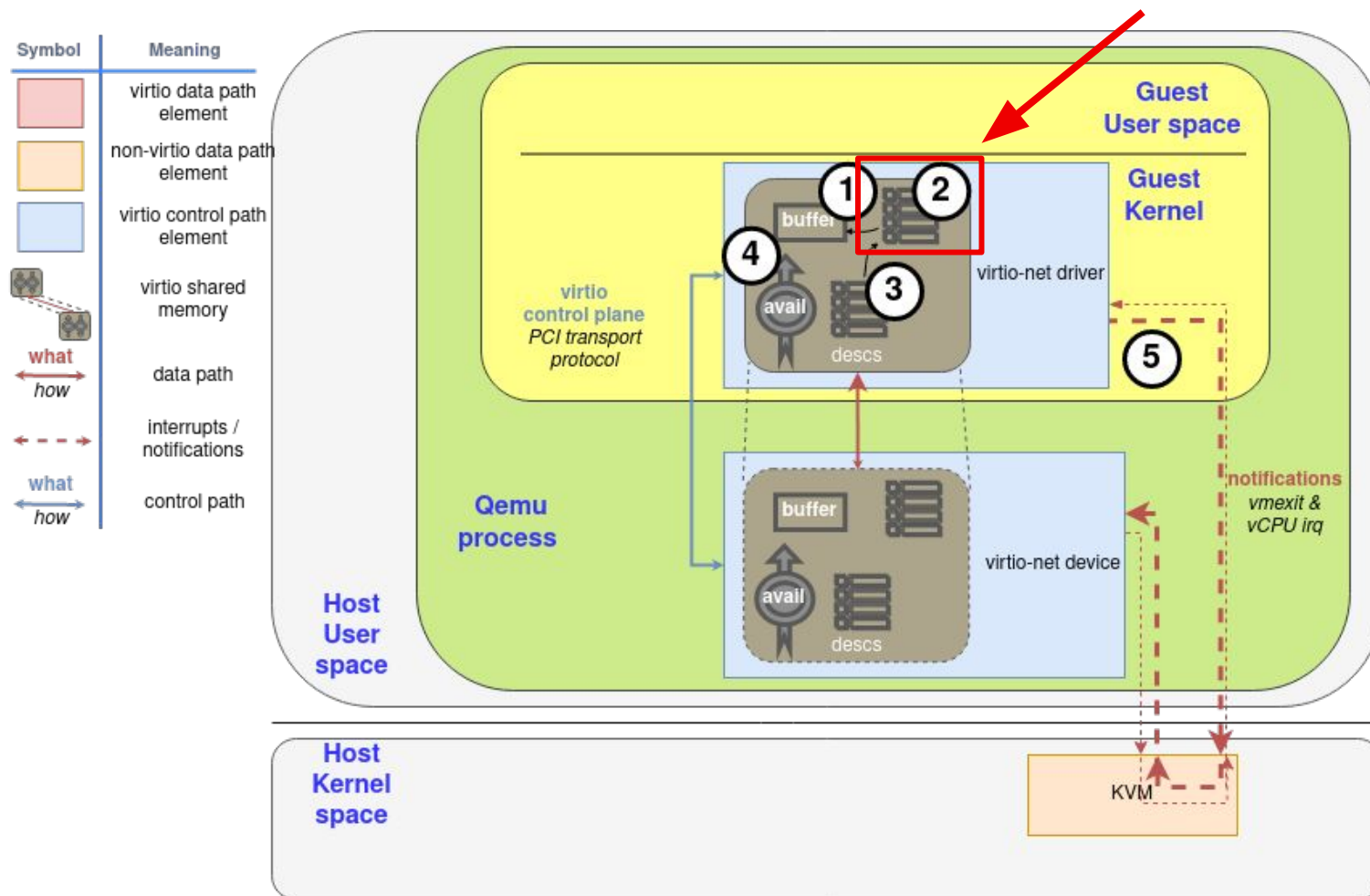


indirect_desc

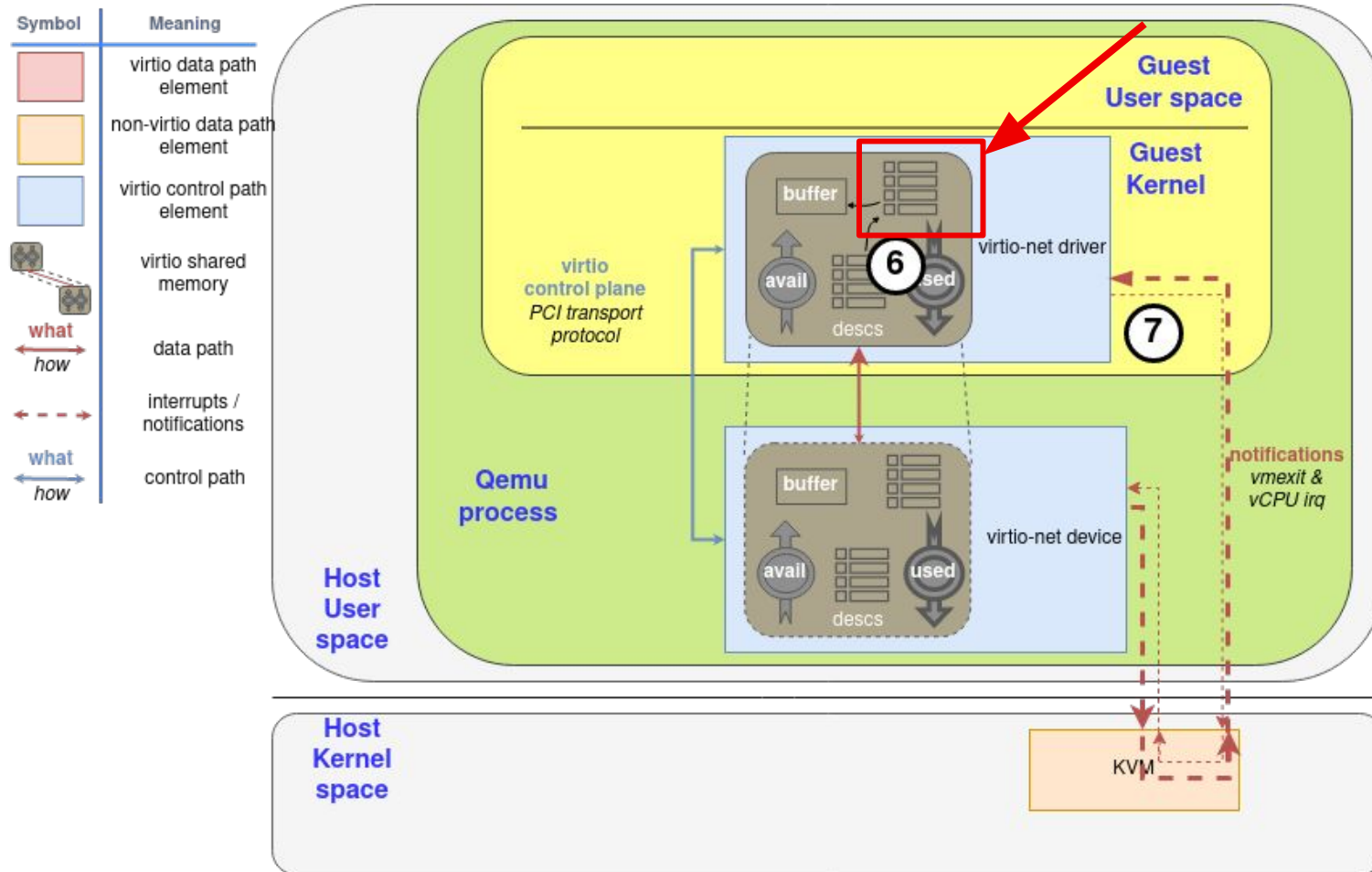
Symbol	Meaning
	virtio data path element
	non-virtio data path element
	virtio control path element
	virtio shared memory
	data path
	interrupts / notifications
	control path



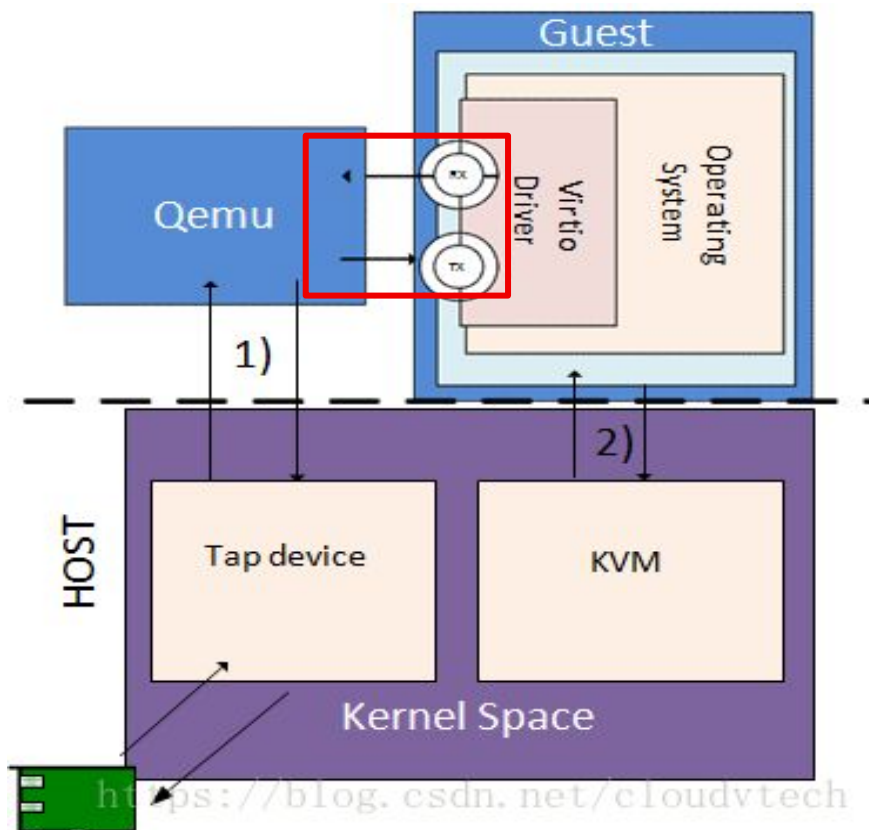
indirect_desc



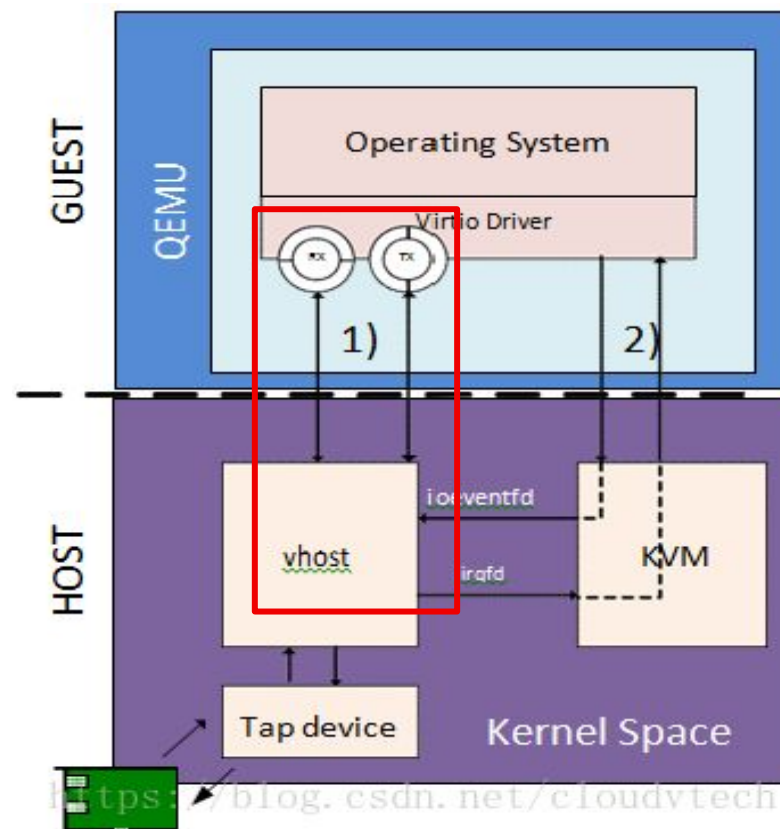
indirect_desc



VHOST 对比











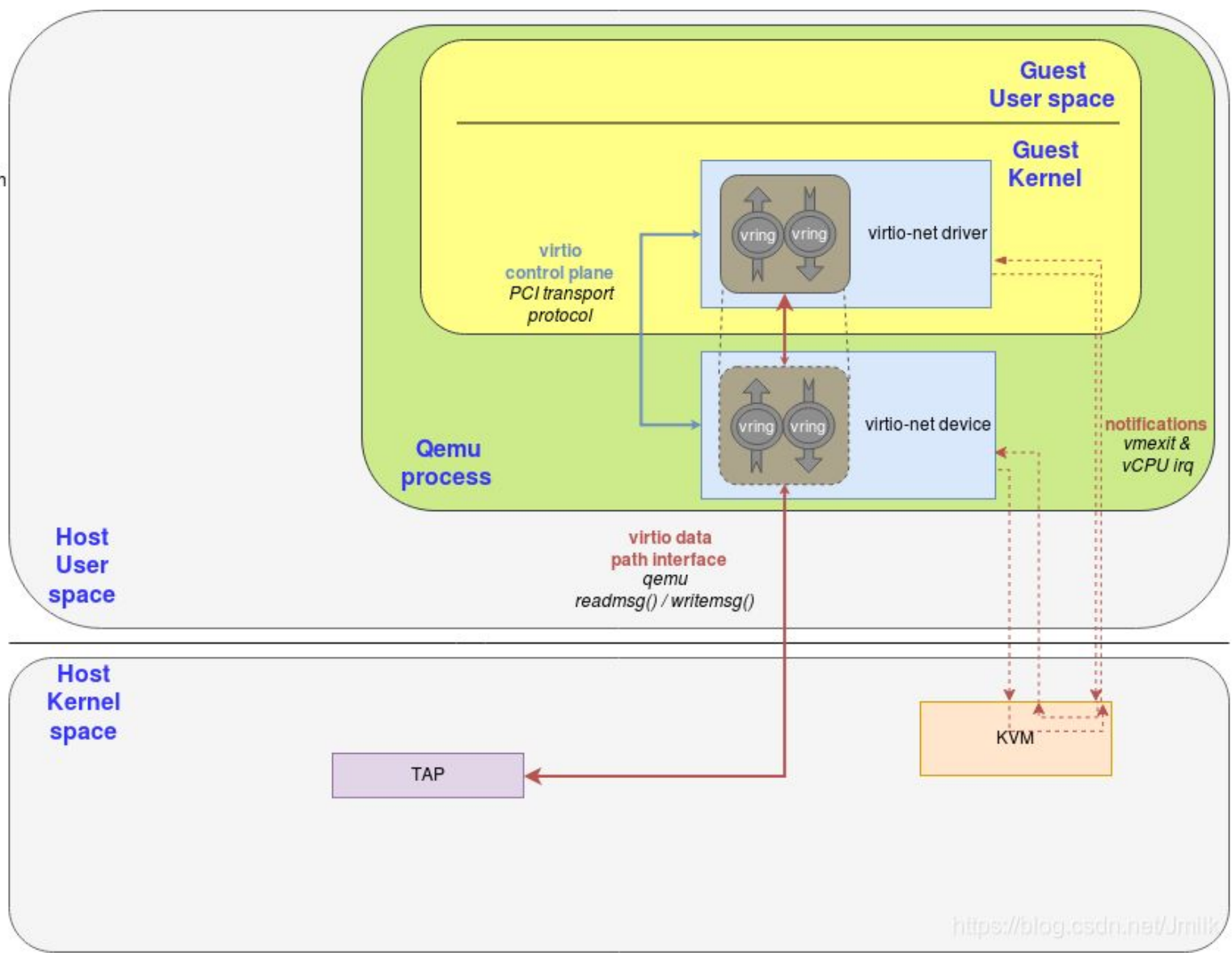
VM-entry / exit 产生了昂贵的上下文切换

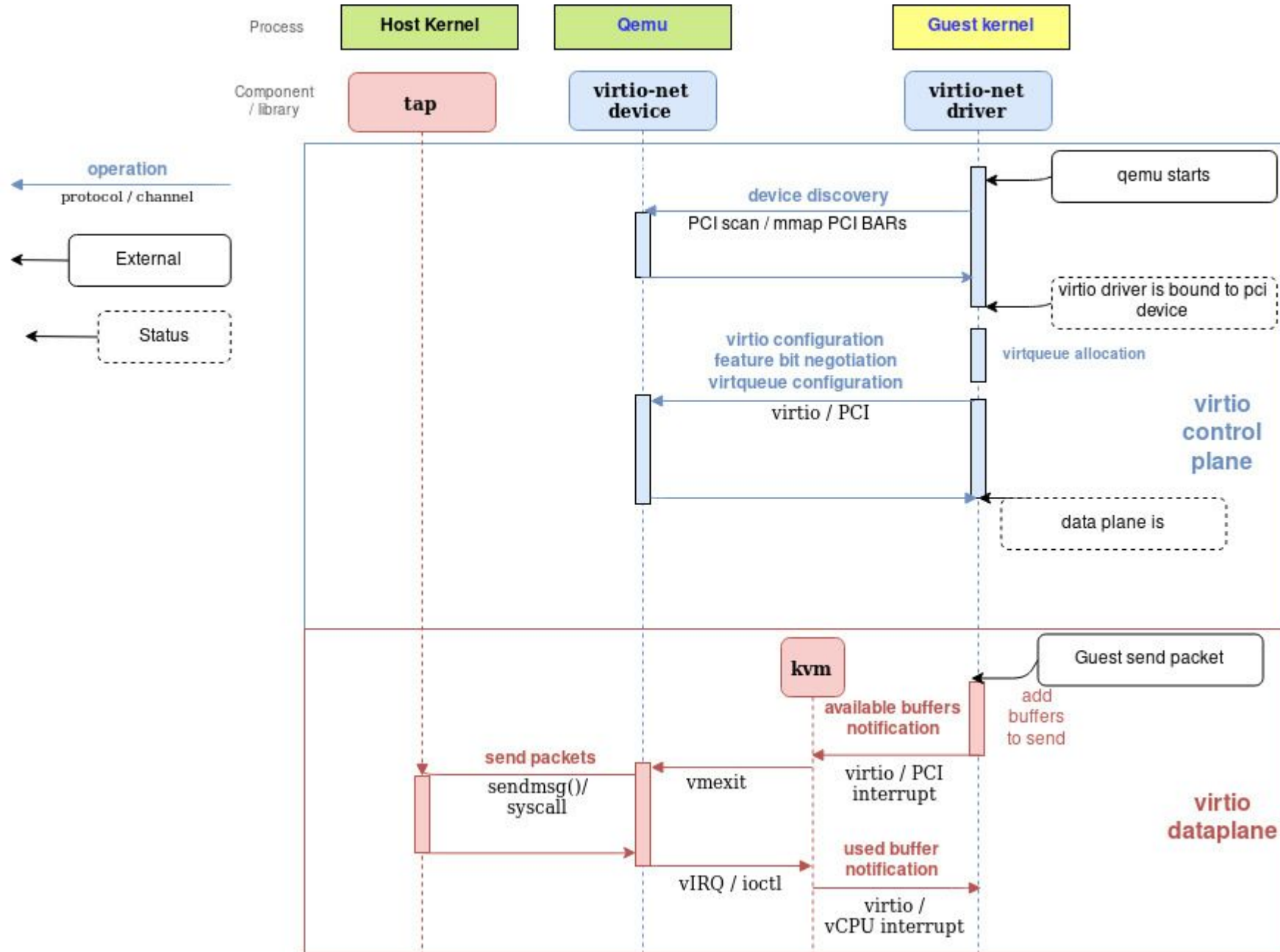


将数据平面卸载给 vhost 协议执行数据转发








Vhost

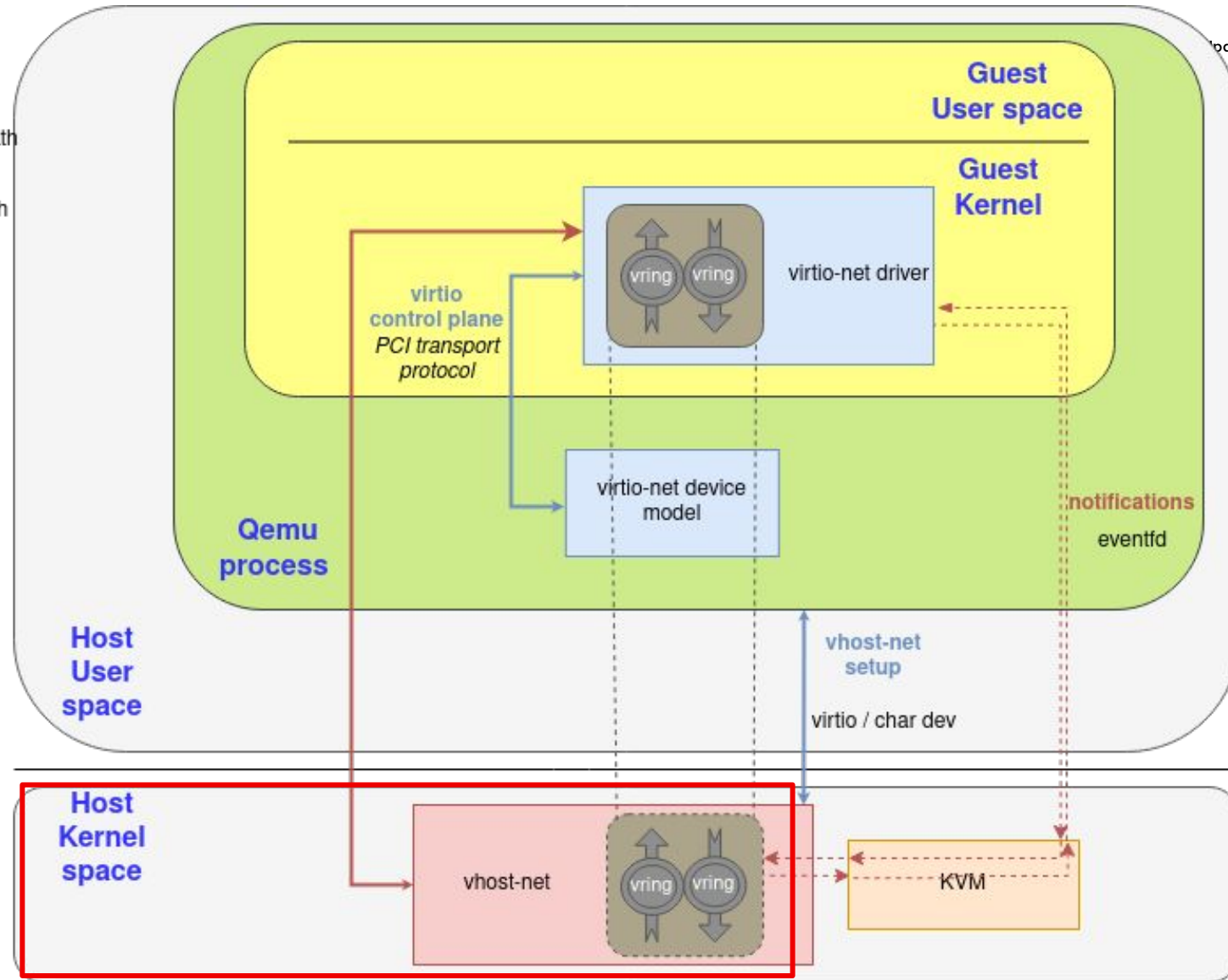
Symbol	Meaning
	virtio data path element
	non-virtio data path element
	virtio control path element
	port
	virtio shared memory
	data path
	interrupts / notifications
	control path





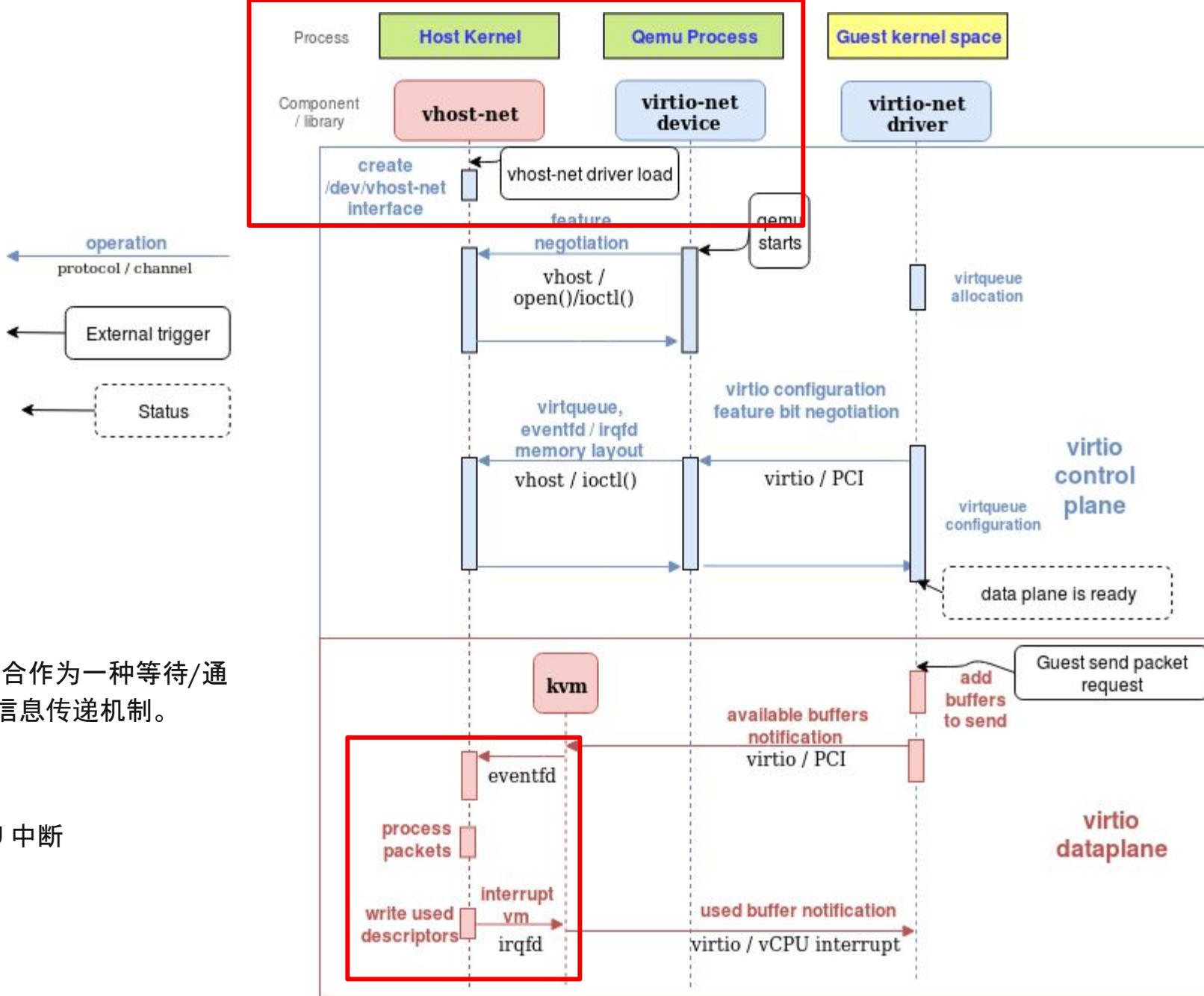
Vhost-net

Symbol	Meaning
	virtio data path element
	non-virtio data path element
	virtio control path element
	virtio shared memory
	data path
	interrupts / notifications
	control path



update confidential designator here








an in-kernel virtio-net device (called vhost-net) to offload the data plane directly to the kernel

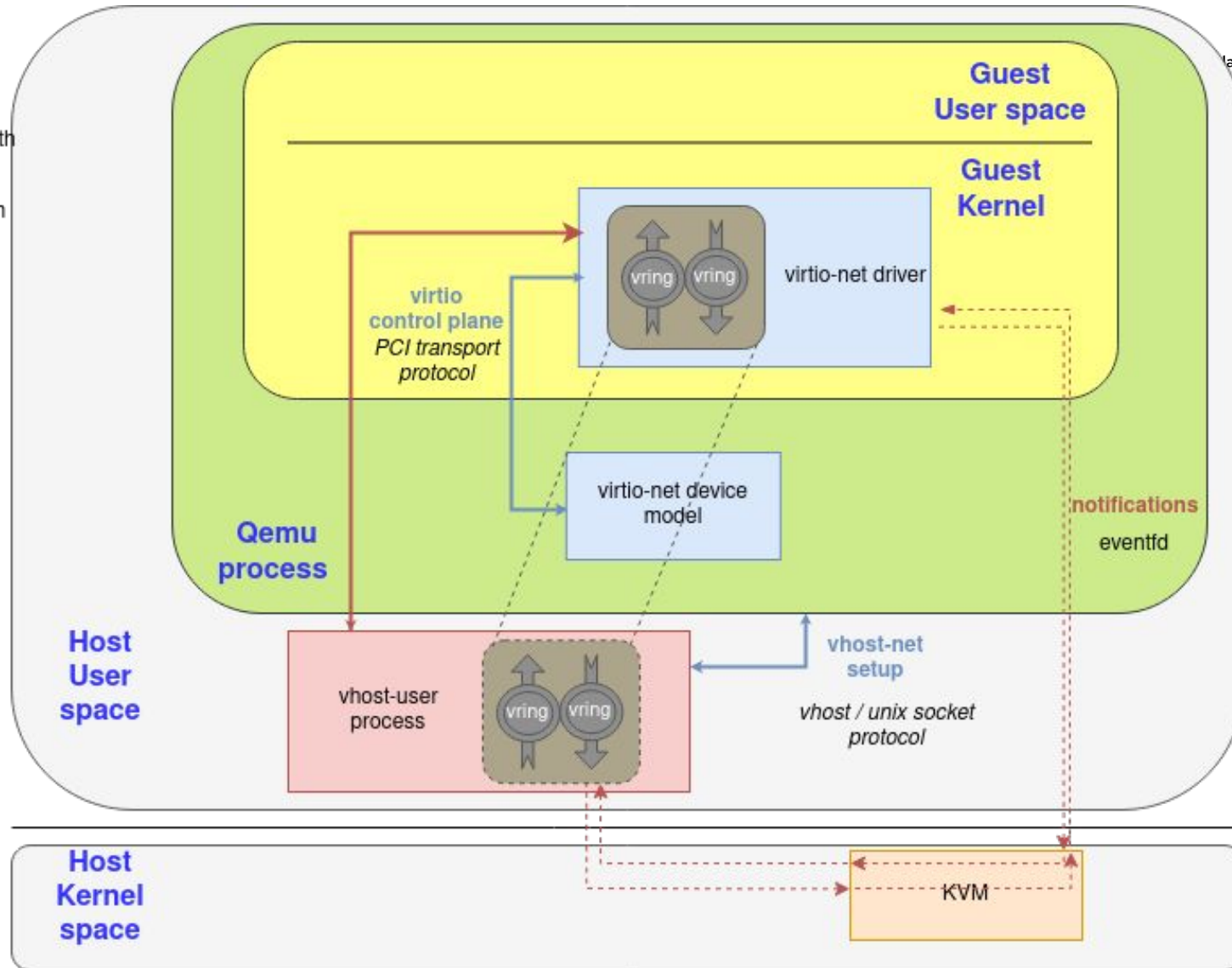


这使得它们更适合作为一种等待/通知机制, 而不是信息传递机制。

irqfd 注入 vCPU 中断

Vhost-user

Symbol	Meaning
	virtio data path element
	non-virtio data path element
	virtio control path element
	virtio shared memory
	data path
	interrupts / notifications
	control path



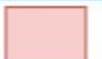
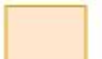
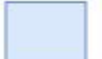







the virtio device from the kernel to an userspace process, that can run a packet forwarding framework like DPDK

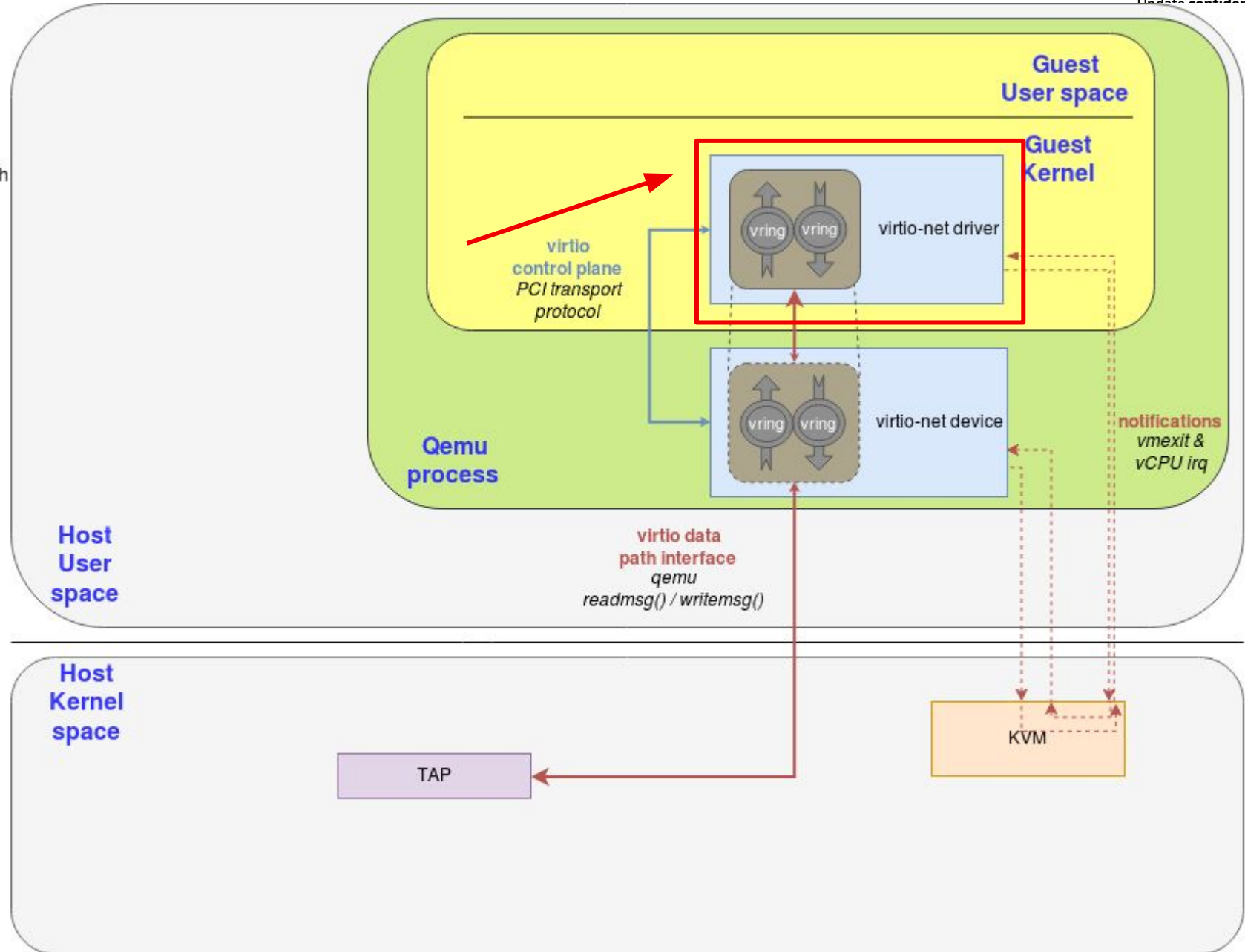
legacy / transitional / modern

特性	旧版模式 (Legacy Mode)	过渡模式 (Transitional Mode)	现代模式 (Modern Mode)
描述	最早的 Virtio 实现方式	兼容旧版和现代模式的过渡方案	改进后的 Virtio 实现方式
接口方式	I/O 端口 (I/O port)	I/O 端口和内存映射 I/O (MMIO)	内存映射 I/O (MMIO) 或 PCI 配置空间
驱动支持	支持早期的 Virtio 驱动程序	兼容旧版和现代驱动	支持新的 Virtio 驱动程序
性能	性能较低	性能中等	性能较高, 支持更多高级特性
相关文件	<code>virtio_pci_legacy.c</code> , <code>virtio_pci_legacy_dev.c</code> , <code>virtio_pci_admin_legacy_io.c</code>	兼容旧版和现代文件	<code>virtio_pci_modern.c</code> , <code>virtio_pci_modern_dev.c</code>
用途	用于需要向后兼容的虚拟化环境	确保新旧驱动和设备之间的兼容性	适用于需要高性能和最新特性的虚拟化环境

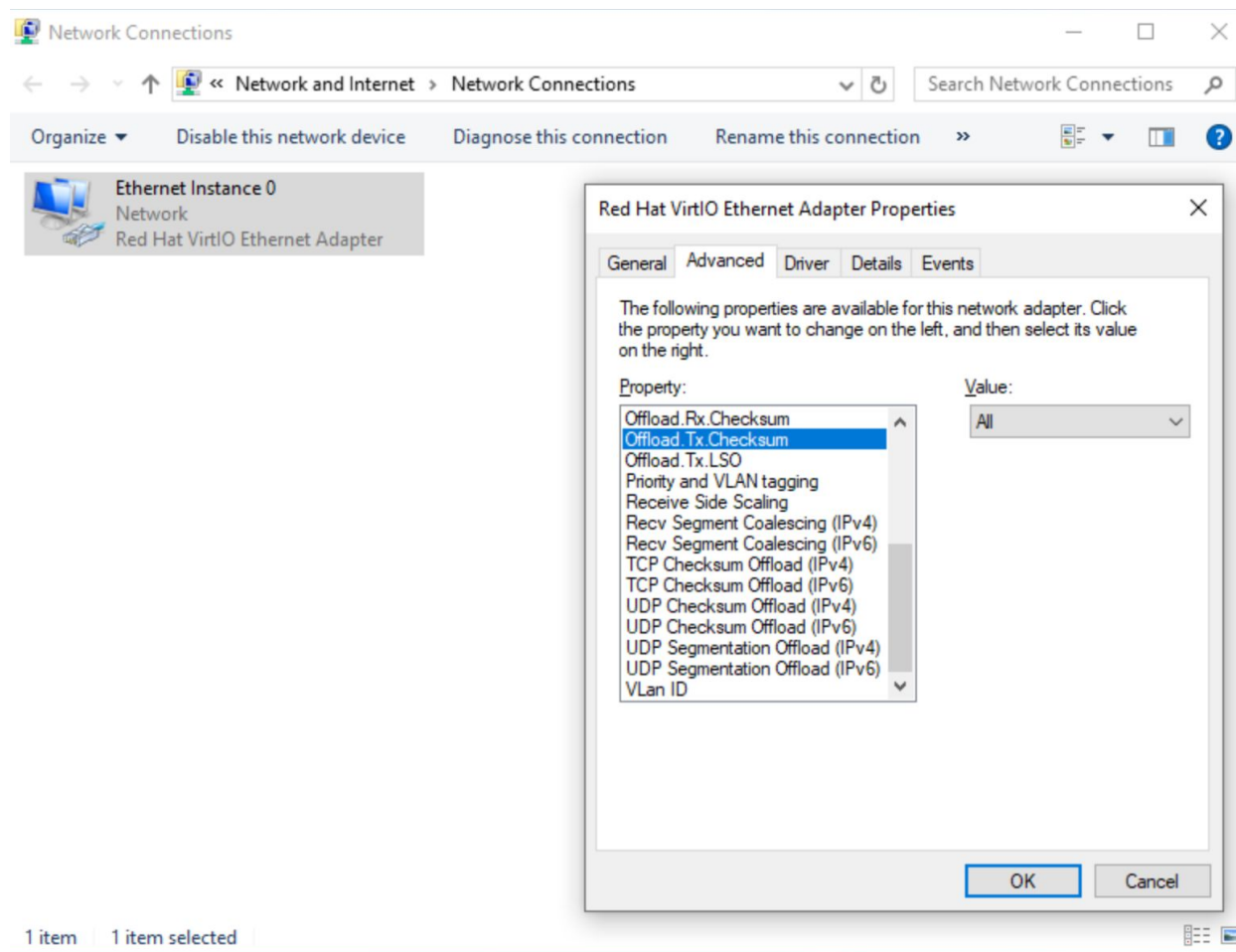
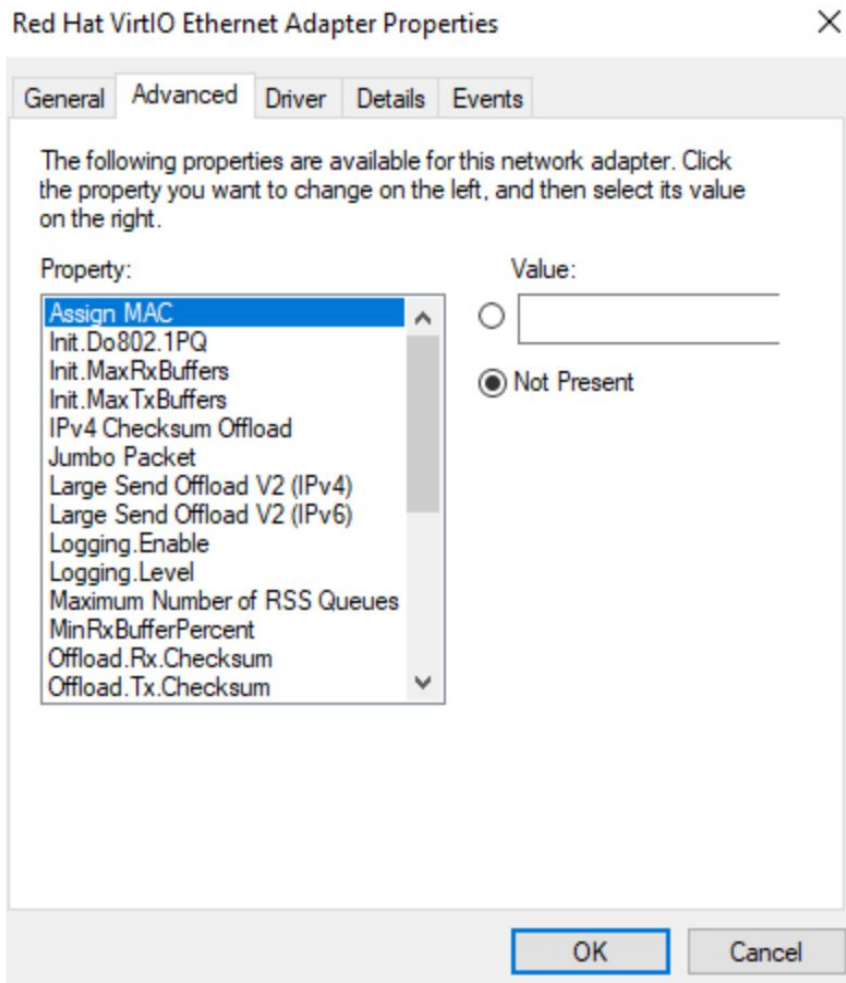
Internal-kvm-guest-driver s-windows



Symbol	Meaning
	virtio data path element
	non-virtio data path element
	virtio control path element
	port
	virtio shared memory
	what
	how
	interrupts / notifications
	what
	how



VirtIO adapter Properties



VirtIO adapter Properties - Log Part

Friendly name	Old name	Type	Default	Description
Logging.Enable	Logging enabled	Boolean	1	Setting to 0 suppresses all the debug printouts
Logging.Level	Logging level	Integer	0	Debug level, increment increases verbosity. 0: Errors only. 1-2 Configuration. 3-4: Packet flow. 5-6: Interrupt and DPC level trace. Warning! Using high debug level will slow down the VM.
Logging.Statistics(sec)	Log statistics period	Integer	0	Setting to N provides periodical statistics printout each N seconds

VirtIO adapter Properties - Initial parameters Part

Friendly name	Old name	Type	Default	Description
Assign MAC	Mac Address	String	NA	Set locally administered MAC address for the paravirtualized NIC. Keep in mind that this address must have locally administered group bit set: http://standards.ieee.org/develop/regauth/tut/macgrp.pdf . If invalid MAC address provided, the driver will reject its assignment but the GUI might still reflect invalid MAC address.
Init.Do802.1PQ	802.1PQ	Boolean	1	Enable support of Priority/VLAN tags population and removal
Init.MTUSize	MTU size	Integer	1500	MTU size can be set between 500 and 65500 in steps of one.
Init.MaxTxBuffers	Initial Tx buffers	Integer	1024	Indicate how much TX ring descriptors the driver will allocate. Possible values: 16, 32, 64, 128, 256, 512, 1024.
Init.MaxRxBuffers	Initial Rx buffers	Integer	256	Indicate how much RX ring descriptors the driver will allocate. Possible values: 16, 32, 64, 128, 256, 512, 1024.
Offload.Tx.Checksum1	Offload Tx TCP checksum	List box	TCP/UDP	Enable TX checksum offloading. TCP/UDP - TCP and UDP checksum offload. TCP - TCP only. Disable - TX checksum offload is disabled.
Offload.Tx.LSO	Offload Tx LSO	Boolean	1	Enable TX TCP Large Segment Offload
Offload.Rx.Checksum	NA	List box	Disabled	Enable RX checksum offloading. Disable - disabled. All - TCP\UDP and IP. TCP\UDP - TCP and UDP checksum offload. TCP - TCP only checksum offload.


- ▶ Linus paper : [virtio: Towards a De-Facto Standard For Virtual I/O Devices](#)
- ▶ Yan suggested: [Home · virtio-win/kvm-guest-drivers-windows Wiki · GitHub](#)
 - [Windows guest debugging presentation from KVM Forum 2012 | PPT](#)
 - [QEMU Development and Testing Automation Using MS HCK - Anton Nayshtut and Yan Vugenfirer, Daynix](#)
 - [Internals of NDIS driver for VirtIO based network adapter - KVM](#)
- ▶ Vhost: [DPDK系列之十六: Virtio技术分析之二, vhost技术对于virtio的增强 原创](#)
- ▶ Virtio Gitbook: [virtio设备驱动程序](#)
- ▶ Summary: [virtio学习](#)
 - <https://www.oasis-open.org/?s=virtio>
- ▶ SDN lab: [详解: VirtIO Networking 虚拟网络设备实现架构](#)
- ▶ 论文: [半虚拟化框架Virtio的网络请求性能优化 - 刘禹燕, 牛保宁](#)
- ▶ Virtio-net: [Virtio-Net 技术分析](#)
- ▶ Red Hat: <https://www.redhat.com/en/virtio-networking-series>
 - <https://www.redhat.com/en/blog/virtqueues-and-virtio-ring-how-data-travels>

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

 [linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)

 [facebook.com/redhatinc](https://www.facebook.com/redhatinc)

 [youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)

 twitter.com/RedHat

