

Introduction to Oracle VM (Xen) Networking

Dongli Zhang

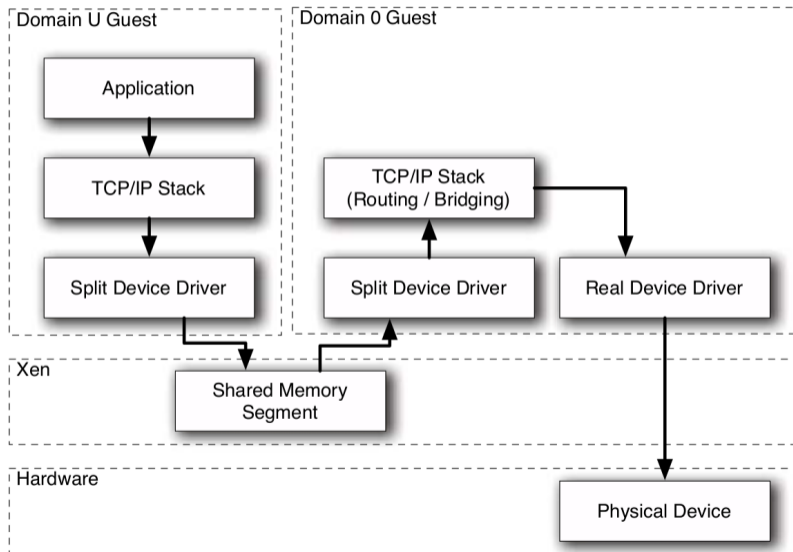
Oracle Asia Research and Development Centers (Beijing)

dongli.zhang@oracle.com

May 30, 2017

- Paravirtualized Networking
 - vif, bridge, bond
- Emulated Networking
- Environment:
 - xen: Oracle VM server 3.3.3 with xen-4.3.0-55.el6.47.33.x86_64
 - dom0: Unbreakable Enterprise Kernel v4.1.12-89
 - domU: Unbreakable Enterprise Kernel v4.1.12-89
- Prerequisite Knowledge: <http://finallyjustice.github.io/xen-arch.pdf>
 - xen framework
 - PVM vs. HVM vs. PVHVM
 - event channel, grant table
 - xen admin hands-on experience (preferred)

Paravirtual xen-netfront/xen-netback framework



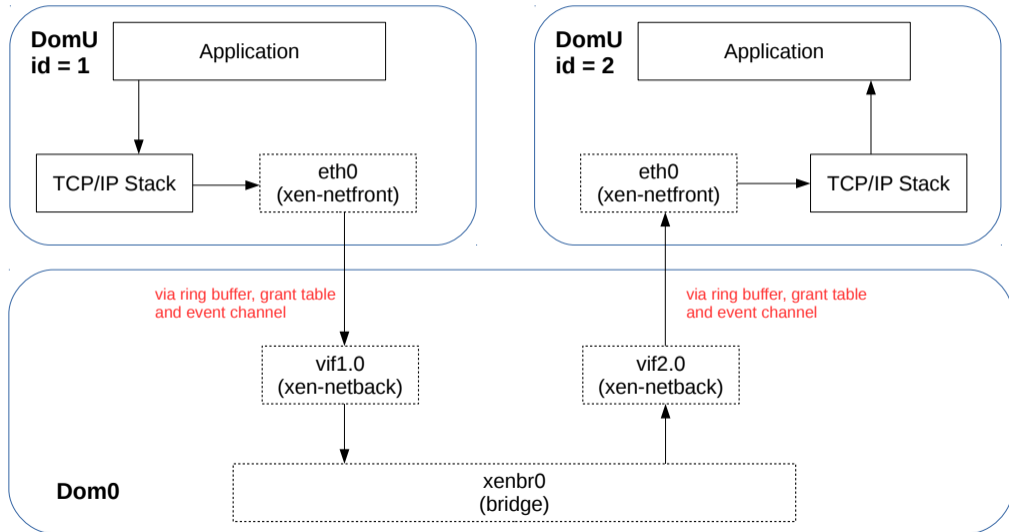
Unbreakable Enterprise Kernel v4.1.12-89

- `drivers/net/xen-netfront.c`
- `drivers/net/xen-netback/xenbus.c`
- `drivers/net/xen-netback/netback.c`
- `drivers/net/xen-netback/interface.c`

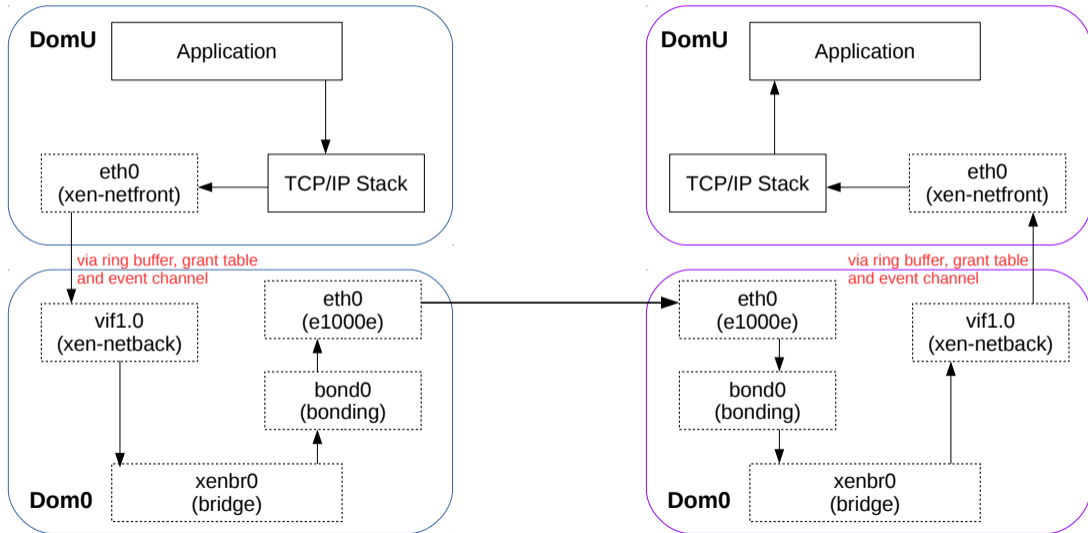
kernel upstream v4.9-rc8

- `drivers/net/xen-netfront.c`
- `drivers/net/xen-netback/xenbus.c`
- `drivers/net/xen-netback/netback.c`
- `drivers/net/xen-netback/interface.c`
- `drivers/net/xen-netback/rx.c`
- `drivers/net/xen-netback/hash.c`

Paravirtual networking scenario 1/2



Paravirtual networking scenario 2/2



PV driver vs. PCI driver

	PCI driver	PV driver
device abstraction	pci_device, pci_driver	
device discovery	PCI Tree	
device configuration	PCI Config Space (IO/MMIO)	
data flow	DMA Ring Buffer	
shared memory	N/A or IOMMU	
interrupt	IOAPIC, MSI, MSI-X	



PV driver vs. PCI driver

	PCI driver	PV driver
device abstraction	pci_device, pci_driver	xenbus_device, xenbus_driver
device discovery	PCI Tree	
device configuration	PCI Config Space (IO/MMIO)	
data flow	DMA Ring Buffer	
shared memory	N/A or IOMMU	
interrupt	IOAPIC, MSI, MSI-X	



PV driver vs. PCI driver

	PCI driver	PV driver
device abstraction	pci_device, pci_driver	xenbus_device, xenbus_driver
device discovery	PCI Tree	Xenstore
device configuration	PCI Config Space (IO/MMIO)	
data flow	DMA Ring Buffer	
shared memory	N/A or IOMMU	
interrupt	IOAPIC, MSI, MSI-X	



PV driver vs. PCI driver

	PCI driver	PV driver
device abstraction	pci_device, pci_driver	xenbus_device, xenbus_driver
device discovery	PCI Tree	Xenstore
device configuration	PCI Config Space (IO/MMIO)	Xenstore
data flow	DMA Ring Buffer	
shared memory	N/A or IOMMU	
interrupt	IOAPIC, MSI, MSI-X	



PV driver vs. PCI driver

	PCI driver	PV driver
device abstraction	pci_device, pci_driver	xenbus_device, xenbus_driver
device discovery	PCI Tree	Xenstore
device configuration	PCI Config Space (IO/MMIO)	Xenstore
data flow	DMA Ring Buffer	Memory Ring Buffer
shared memory	N/A or IOMMU	
interrupt	IOAPIC, MSI, MSI-X	



PV driver vs. PCI driver

	PCI driver	PV driver
device abstraction	pci_device, pci_driver	xenbus_device, xenbus_driver
device discovery	PCI Tree	Xenstore
device configuration	PCI Config Space (IO/MMIO)	Xenstore
data flow	DMA Ring Buffer	Memory Ring Buffer
shared memory	N/A or IOMMU	Grant Table
interrupt	IOAPIC, MSI, MSI-X	

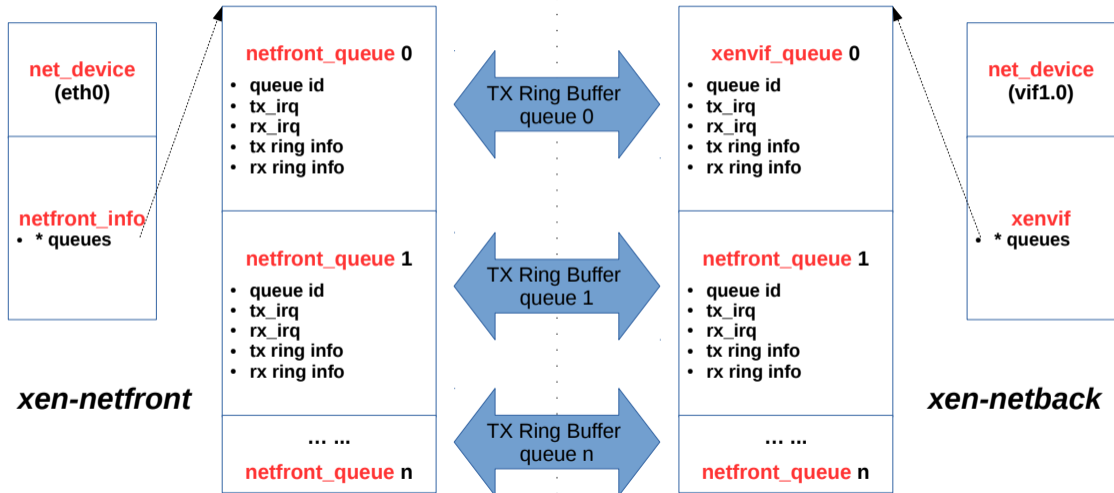


PV driver vs. PCI driver

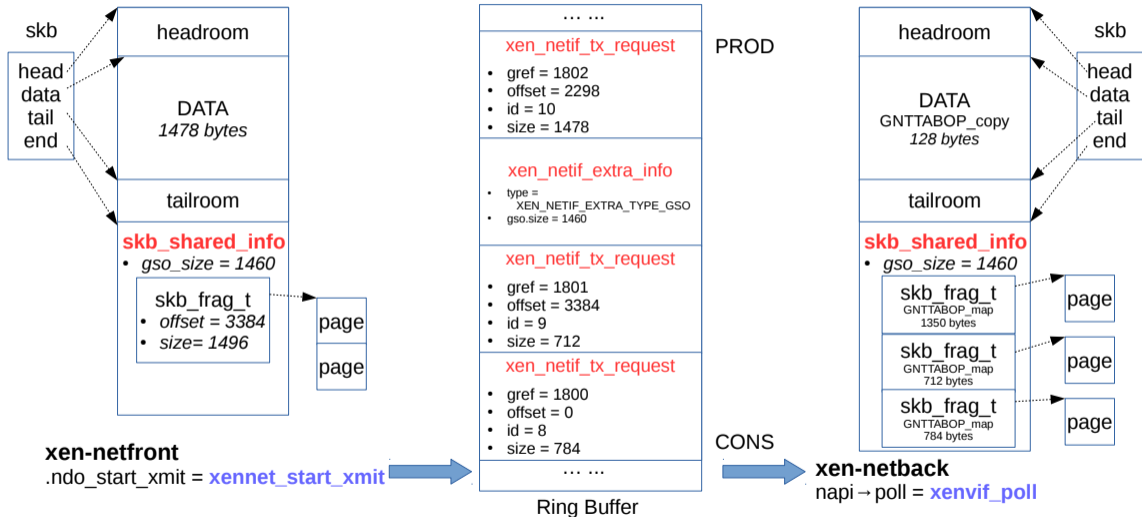
	PCI driver	PV driver
device abstraction	pci_device, pci_driver	xenbus_device, xenbus_driver
device discovery	PCI Tree	Xenstore
device configuration	PCI Config Space (IO/MMIO)	Xenstore
data flow	DMA Ring Buffer	Memory Ring Buffer
shared memory	N/A or IOMMU	Grant Table
interrupt	IOAPIC, MSI, MSI-X	Event Channel



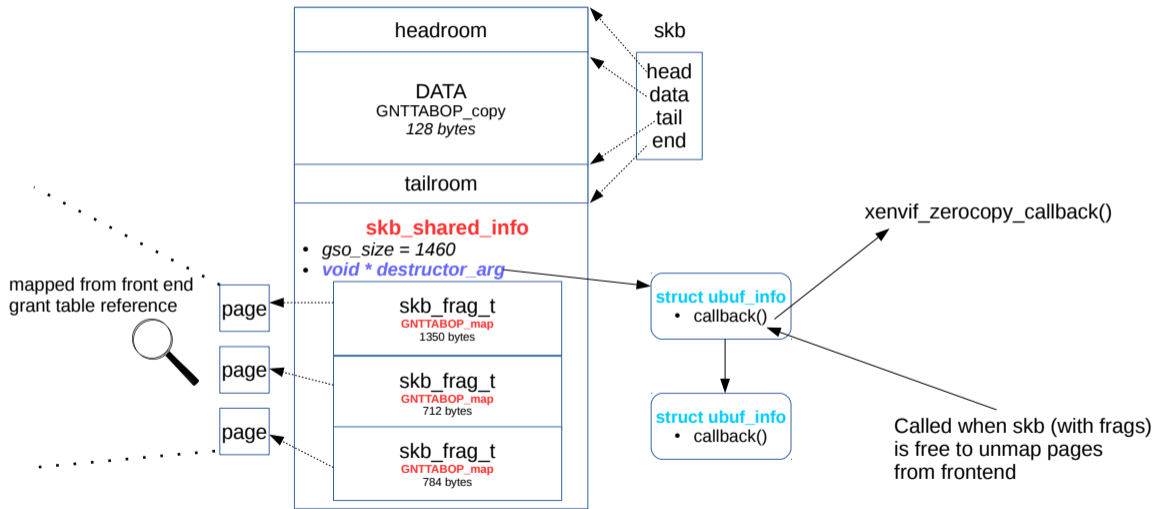
pv xmit: front \rightarrow backend 1/3



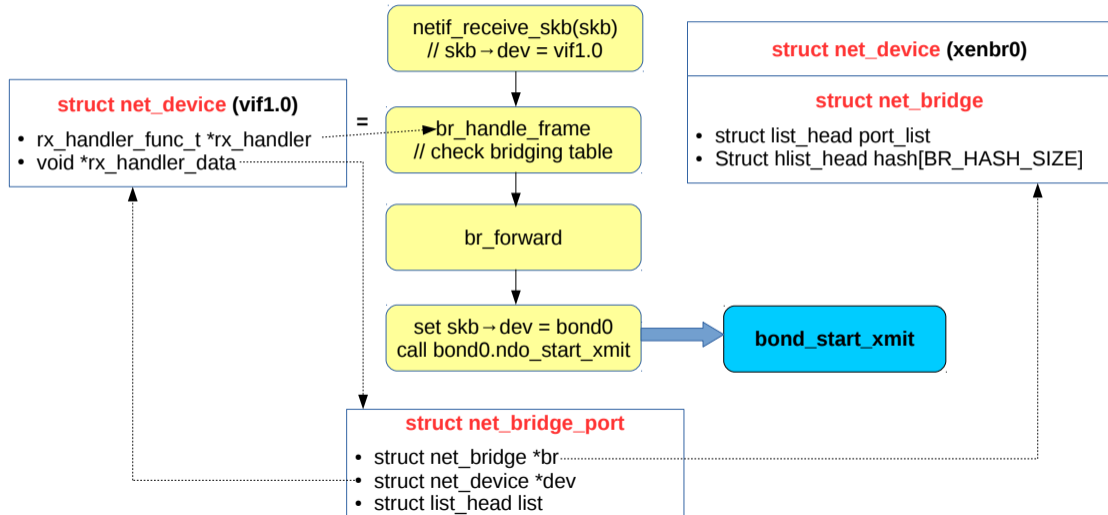
pv xmit: front —> backend 2/3



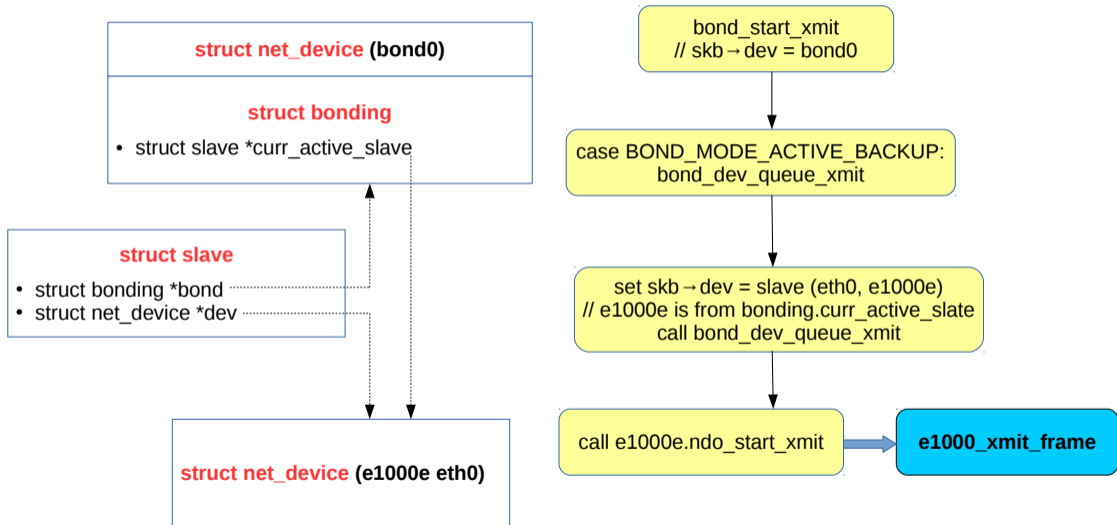
pv xmit: front —> backend 3/3



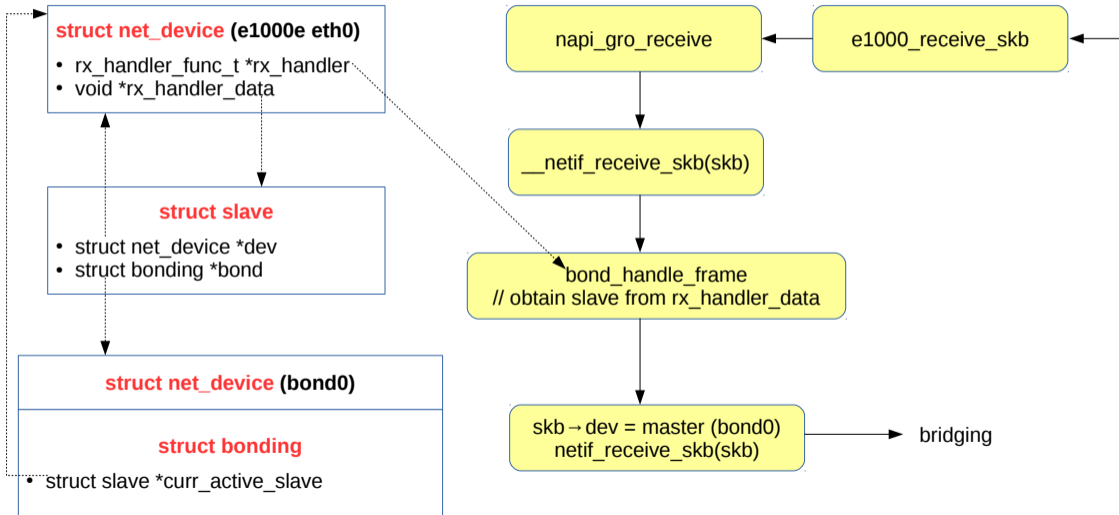
pv xmit: backend \rightarrow bridge \rightarrow bond



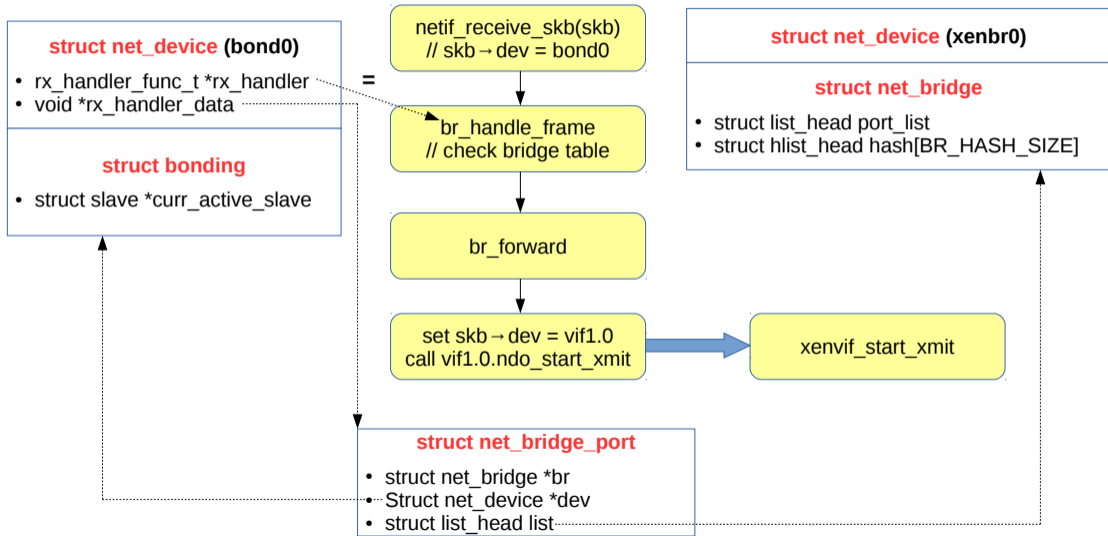
pv xmit: bond —> physical NIC



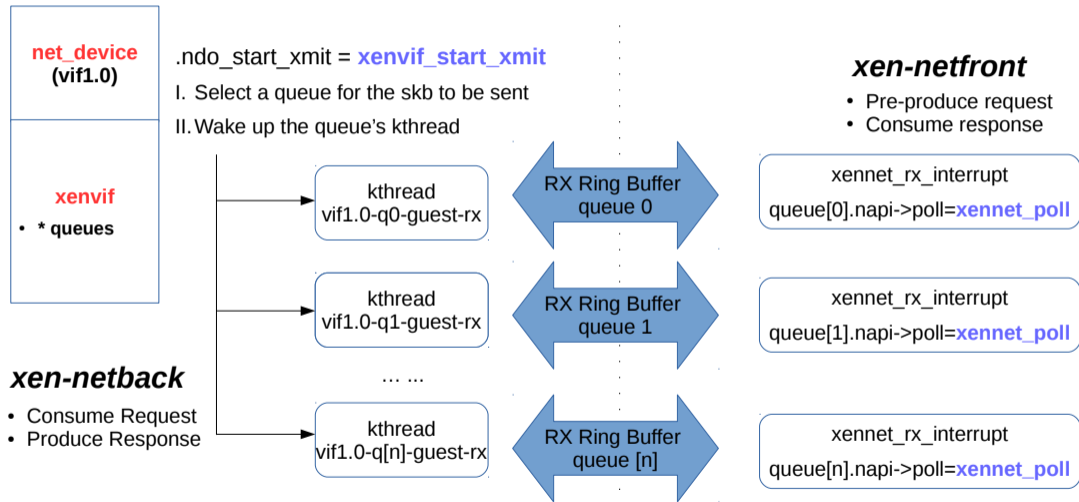
pv recv: physical NIC —> bond —> bridge



pv recv: bridge —> backend



pv recv: backend —> frontend



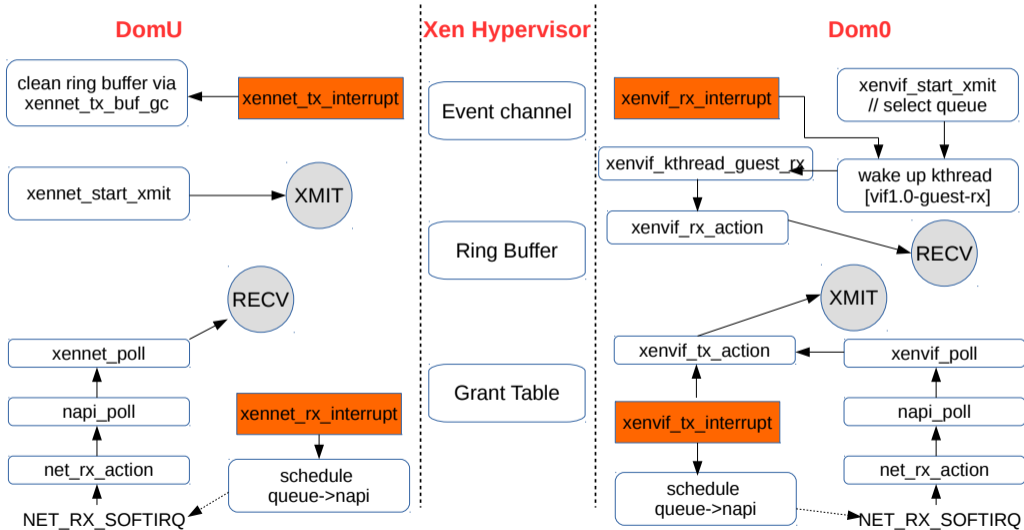
netfront to netback (produce req)

- 1 1st page of linear data (skb->data)
- 2 extra info (xen_netif_extra_info)
- 3 the rest of linear data (skb->data)
- 4 all skb fragments (skb_shinfo(skb)->frags)

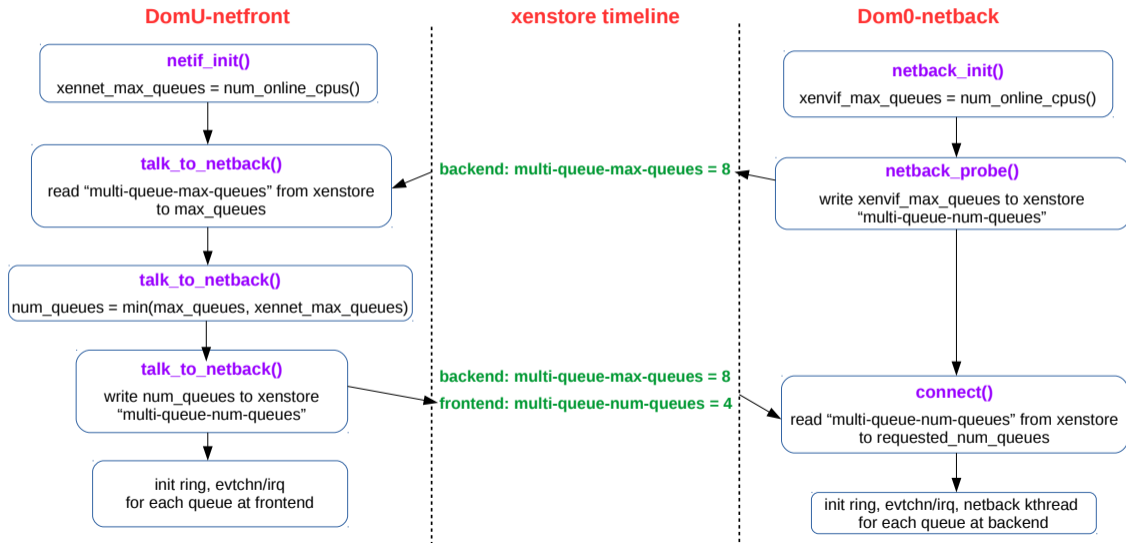
netback to netfront (produce rsq)

- 1 1st page of linear data (skb->data)
- 2 extra info (xen_netif_extra_info)
- 3 the rest of linear data (skb->data)
- 4 all skb fragments (skb_shinfo(skb)->frags)

xen-netfront/xen-netback summary: irq and napi



features: multiqueue (default)



- Segmentation Offload
 - GSO (Generic Segmentation Offload): software segmentation
 - TSO (TCP Segmentation Offload): hardware segmentation

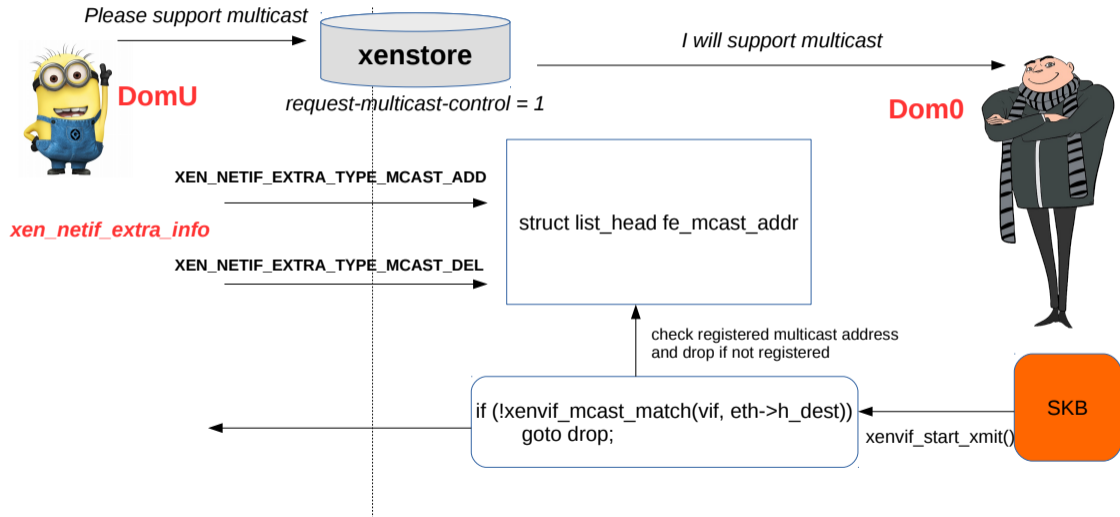
- Segmentation Offload
 - GSO (Generic Segmentation Offload): software segmentation
 - TSO (TCP Segmentation Offload): hardware segmentation
- TSO would postpone segmentation to as late (low level) as possible

- Segmentation Offload
 - GSO (Generic Segmentation Offload): software segmentation
 - TSO (TCP Segmentation Offload): hardware segmentation
- TSO would postpone segmentation to as late (low level) as possible
- TSO info is shared via "struct xen_netif_extra_info gso" in ring buffer
 - `gso.gso->u.gso.size = skb_shinfo(skb)->gso_size;`
 - `gso->u.gso.type = XEN_NETIF_GSO_TYPE_TCPV4;`

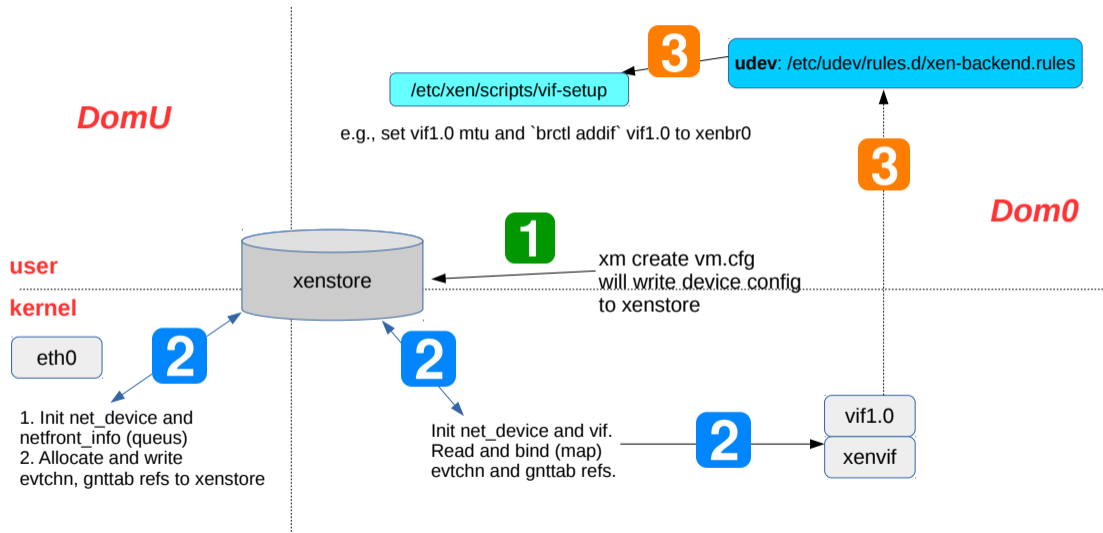
- Segmentation Offload
 - GSO (Generic Segmentation Offload): software segmentation
 - TSO (TCP Segmentation Offload): hardware segmentation
- TSO would postpone segmentation to as late (low level) as possible
- TSO info is shared via "struct xen_netif_extra_info gso" in ring buffer
 - `gso.gso->u.gso.size = skb_shinfo(skb)->gso_size;`
 - `gso->u.gso.type = XEN_NETIF_GSO_TYPE_TCPV4;`
- TSO and other offload features are stored in xenstore (e.g., feature-gso-tcpv4)
 - `.ndo_fix_features = xennet_fix_features`
 - `.ndo_set_features = xennet_set_features`

- Segmentation Offload
 - GSO (Generic Segmentation Offload): software segmentation
 - TSO (TCP Segmentation Offload): hardware segmentation
- TSO would postpone segmentation to as late (low level) as possible
- TSO info is shared via "struct xen_netif_extra_info gso" in ring buffer
 - `gso.gso->u.gso.size = skb_shinfo(skb)->gso_size;`
 - `gso->u.gso.type = XEN_NETIF_GSO_TYPE_TCPV4;`
- TSO and other offload features are stored in xenstore (e.g., feature-gso-tcpv4)
 - `.ndo_fix_features = xennet_fix_features`
 - `.ndo_set_features = xennet_set_features`
- checksum offload
 - `XEN_NETTXF_csum_blank`: Protocol checksum field is blank in the packet (hardware offload)
 - `XEN_NETTXF_data_validated`: Packet data has been validated against protocol checksum

features: multicast



xen-netfront/xen-netback init



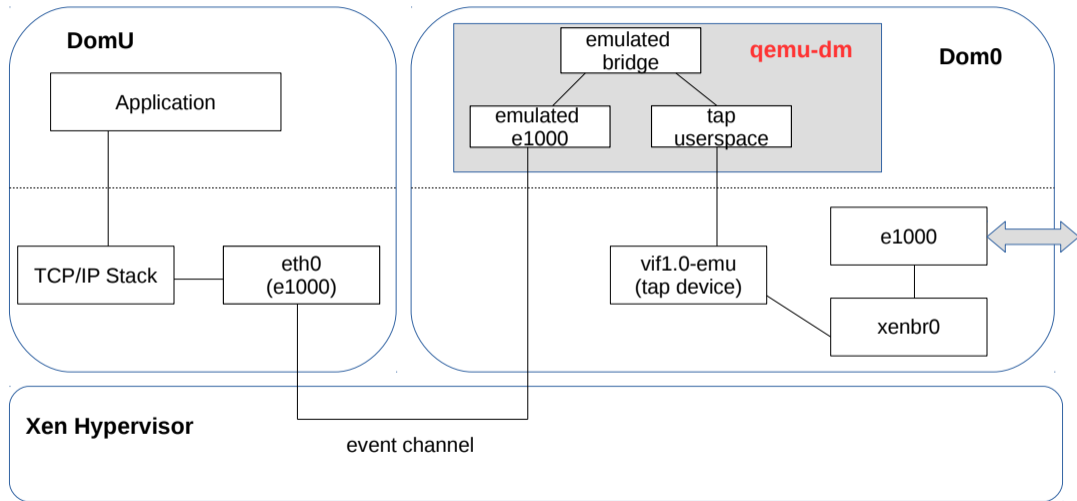
- netfront/netback multiqueue
- Limit and pin dom0 CPUs to first NUMA socket
- Interrupt affinity to reduce CPU 0 workload
- domU vcpu affinity to improve memory access performance
- Jumbo frame
- NIC offload
- TCP Parameter Settings



Interesting works related to paravirtual I/O

- Achieving 10 Gb/s Using Safe and Transparent Network Interface Virtualization. VEE 2009
- Efficient and Scalable Paravirtual I/O System. USENIX ATC 2013
- rIOMMU: Efficient IOMMU for I/O Devices that Employ Ring Buffers. ASPLOS 2015
- vRIO: Paravirtual remote I/O. ASPLOS 2016

Networking Emulation with QEMU



qemu arguments

pvm

```
/usr/lib/xen/bin/qemu-dm -d 4 -serial pty -domain-name testpv -videoram 4 -k en-us -vnc 0.0.0.0:0 -vncunused -M xenpv
```

pvhvm

```
/usr/lib/xen/bin/qemu-dm -d 5 -domain-name oel65.xm -videoram 4 -k en-us -vnc 0.0.0.0:0 -vncunused -vcpus 2 -vcpu_avail 0x3 -boot dc -serial pty -acpi -net none -M xenfv
```

hvm

```
/usr/lib/xen/bin/qemu-dm -d 3 -domain-name oel65.xm -videoram 4 -k en-us -vnc 0.0.0.0:0 -vncunused -vcpus 2 -vcpu_avail 0x3 -boot dc -serial pty -acpi  
-net nic,vlan=1,macaddr=00:16:e3:cc:64:a9,model=e1000  
-net tap,vlan=1,ifname=vif3.0-emu,bridge=xenbr0,script=no,downscript=no  
-M xenfv
```

- xen paravirtual networking workflow



Take-Home Message

- xen paravirtual networking workflow
- xen paravirtual networking framework



Take-Home Message

- xen paravirtual networking workflow
- xen paravirtual networking framework
- xen paravirtual networking init, protocol, features



Take-Home Message

- xen paravirtual networking workflow
- xen paravirtual networking framework
- xen paravirtual networking init, protocol, features
- xen paravirtual networking performance



Take-Home Message

- xen paravirtual networking workflow
- xen paravirtual networking framework
- xen paravirtual networking init, protocol, features
- xen paravirtual networking performance
- xen emulated networking

