

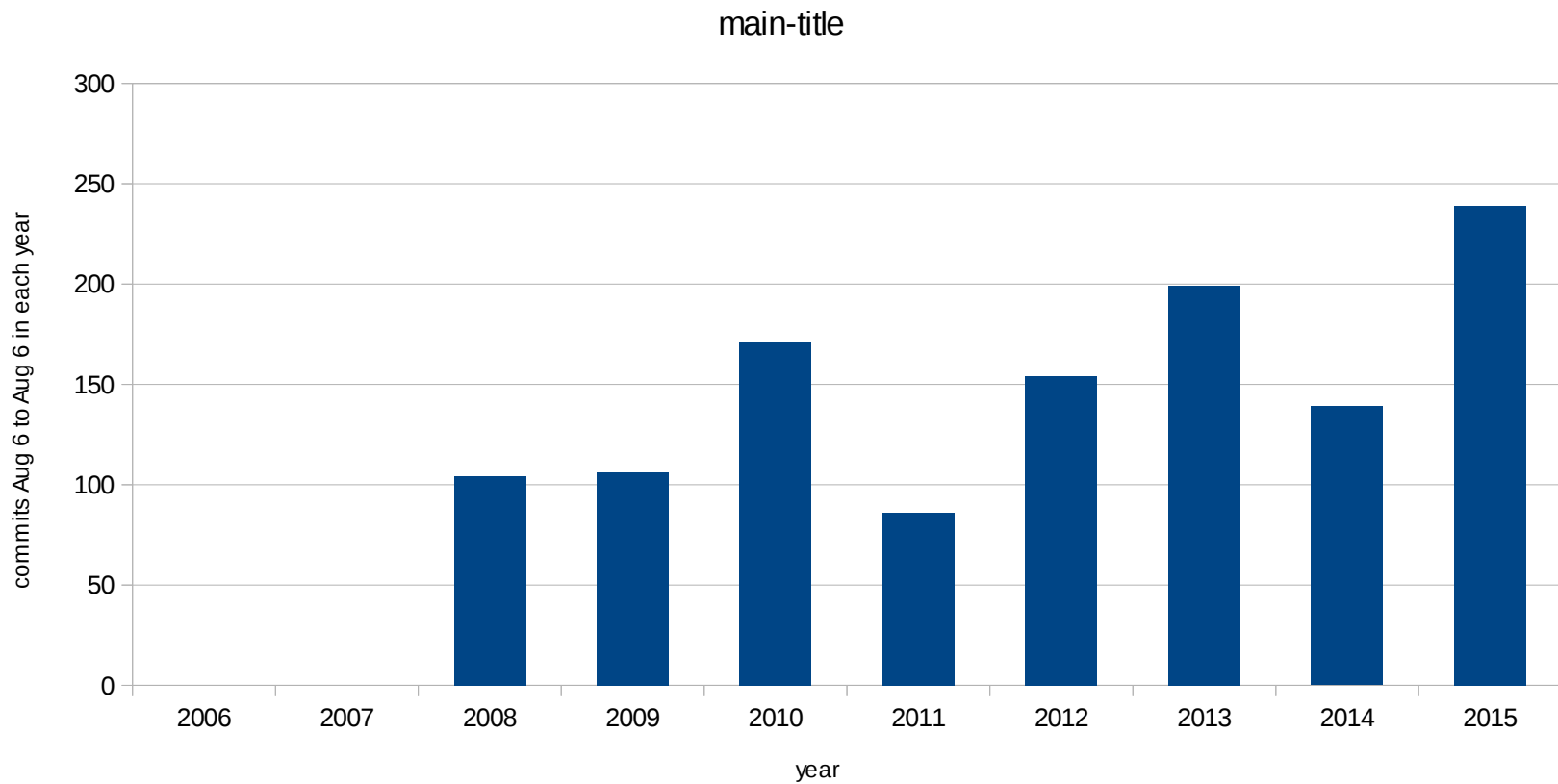
Virtio 1 - why do it? And - are we there yet?

2015

Michael S. Tsirkin
Red Hat

Uses material from
<https://lwn.net/Kernel/LDD3/>
Gcompris, tuxpaint
Distributed under the
Creative commons license.

Lots of work ...



Virtio 1: update

- Documented assumptions
- More Robust
- More Extendable



Conformance statements

Virtio 0.9

- DRIVER_OK status bit is set.
- The device can now be used.

```
drv→probe(dev);  
    netif_carrier_on(dev)
```



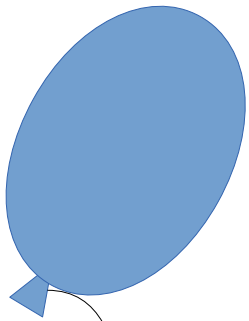
```
add_status(dev, DRIVER_OK);
```

Virtio 1.0

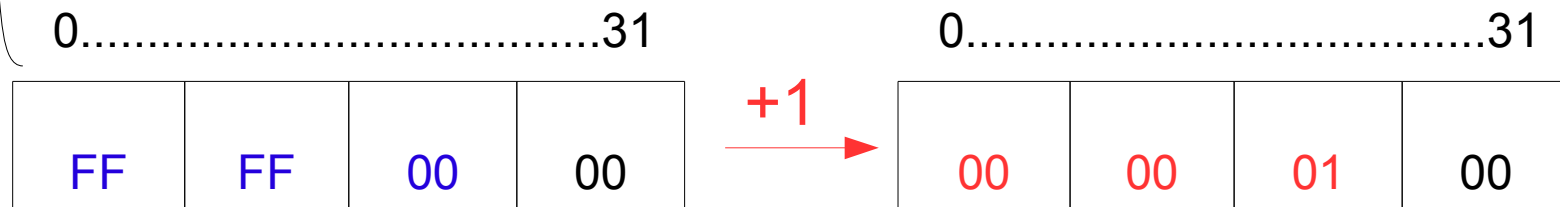
The driver **MUST NOT** notify the device before setting DRIVER_OK.

```
drv→probe(dev);  
    add_status(dev, DRIVER_OK);  
    netif_carrier_on(dev)
```



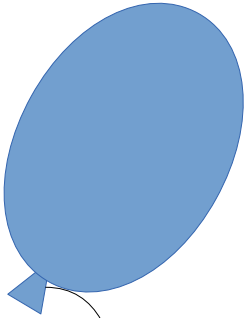


Virtio 0.9: inflate

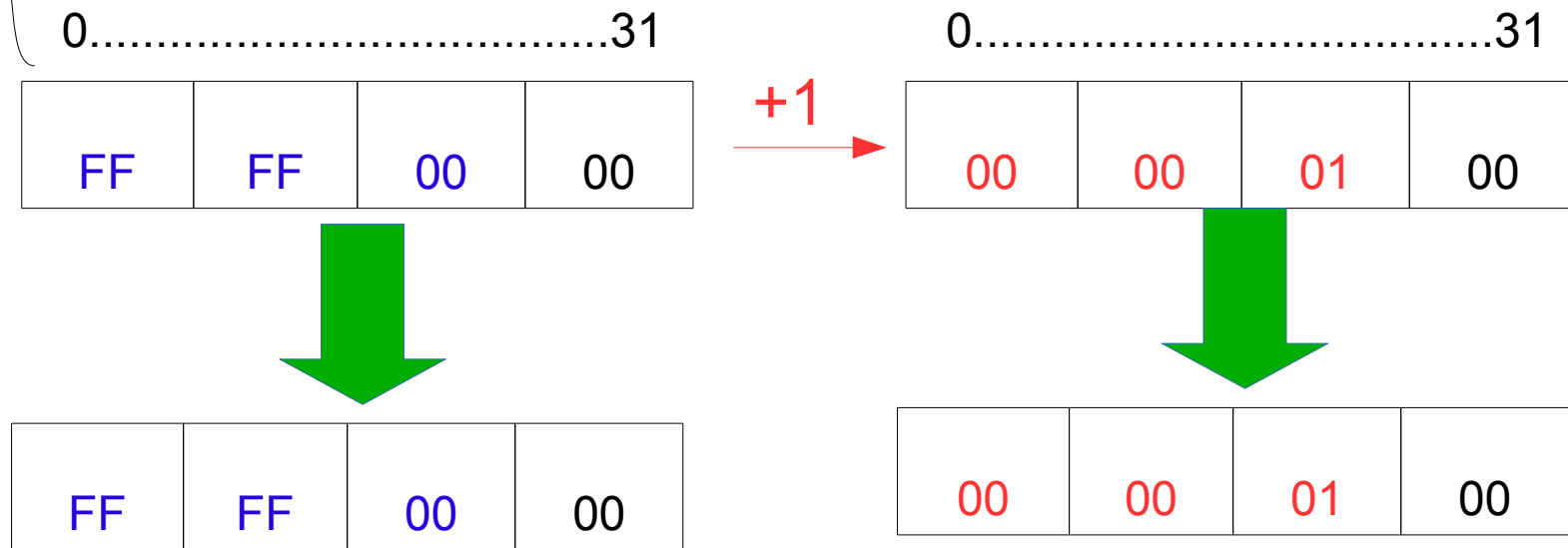


FF	FF	01	00
----	----	----	----

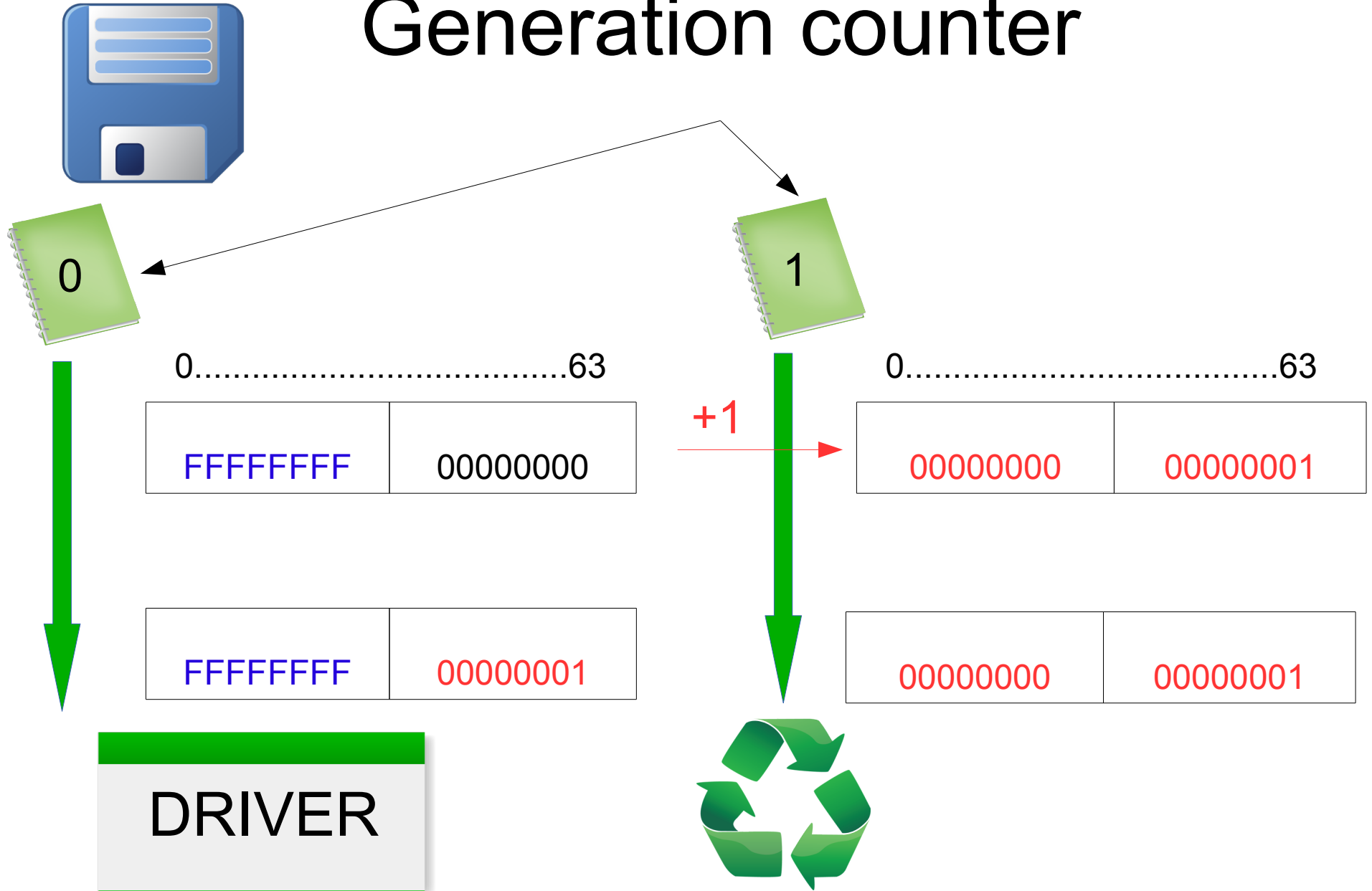




Virtio 1.0: inflate



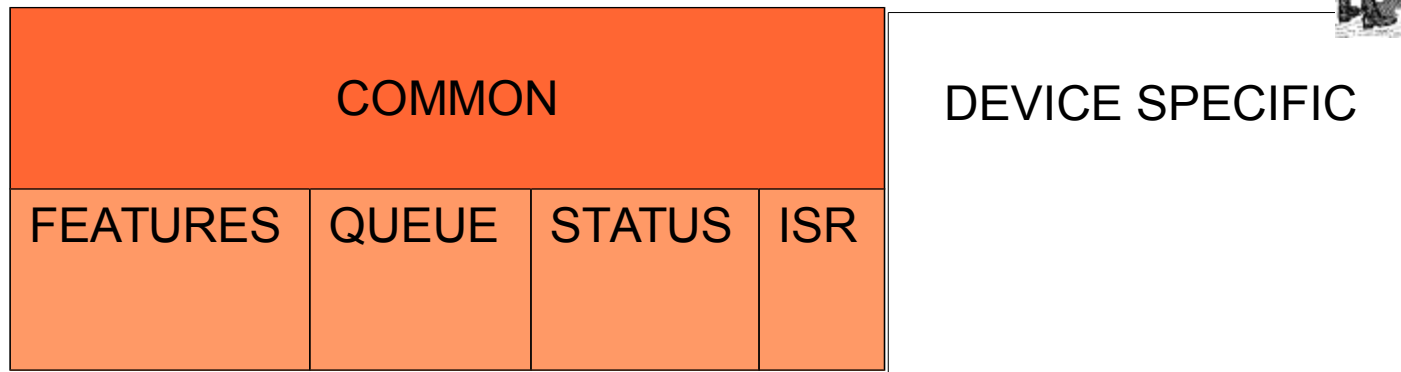
Generation counter



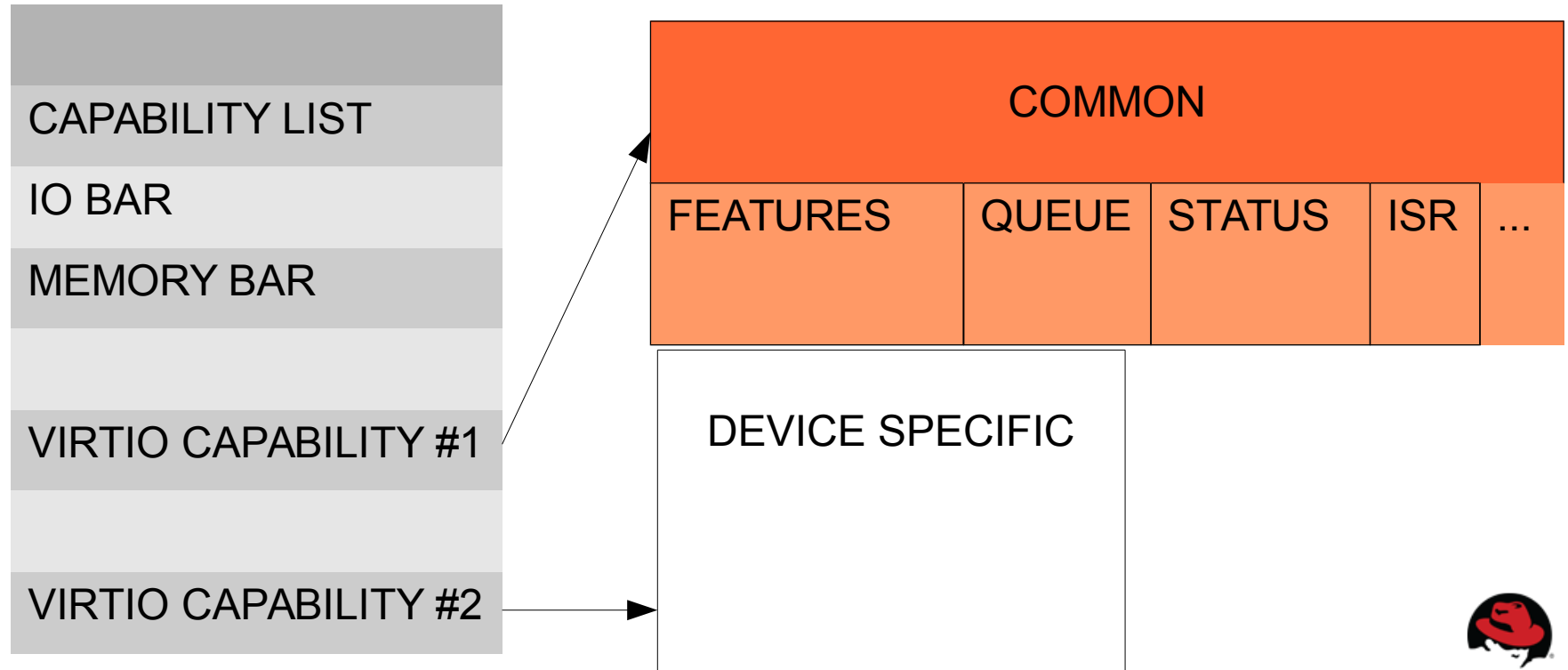
Memory map



0.9



1.0



Virtio 0.9: Port IO vs Memory

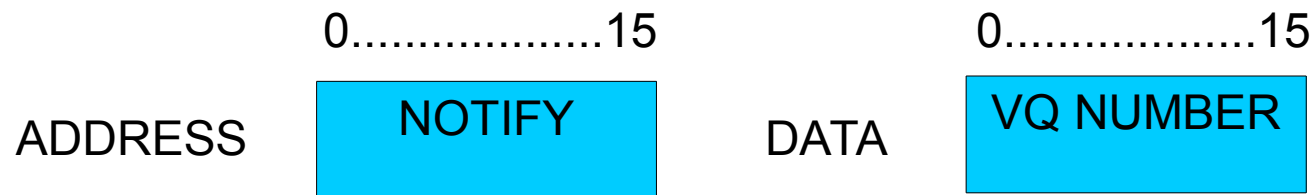
	Port IO	MM IO
x86 decode: address	✓	✓
x86 decode: data	✓	✗
Fast on x86	✓	✗
32/64 bit	✗	✓
Page tables	✗	✓
Required by PCI Express	✗	✓



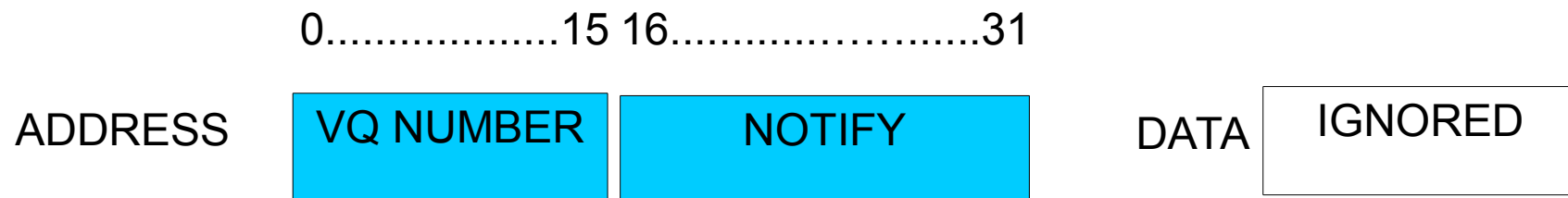
Fast MMIO

avoid need to decode data

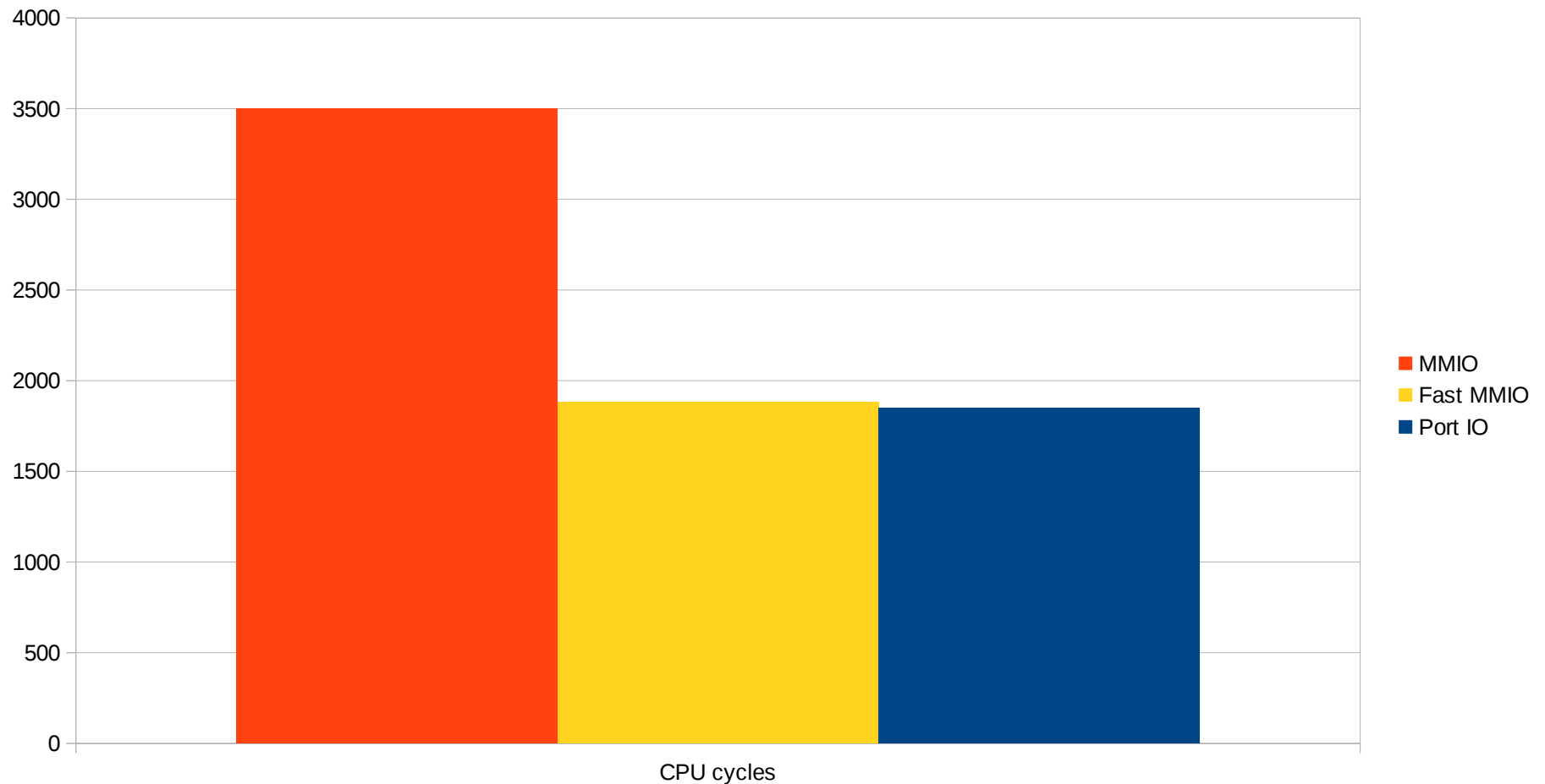
0.9



1.0



Virtio 1: Access times on KVM x86: Cycles per access (lower is better)

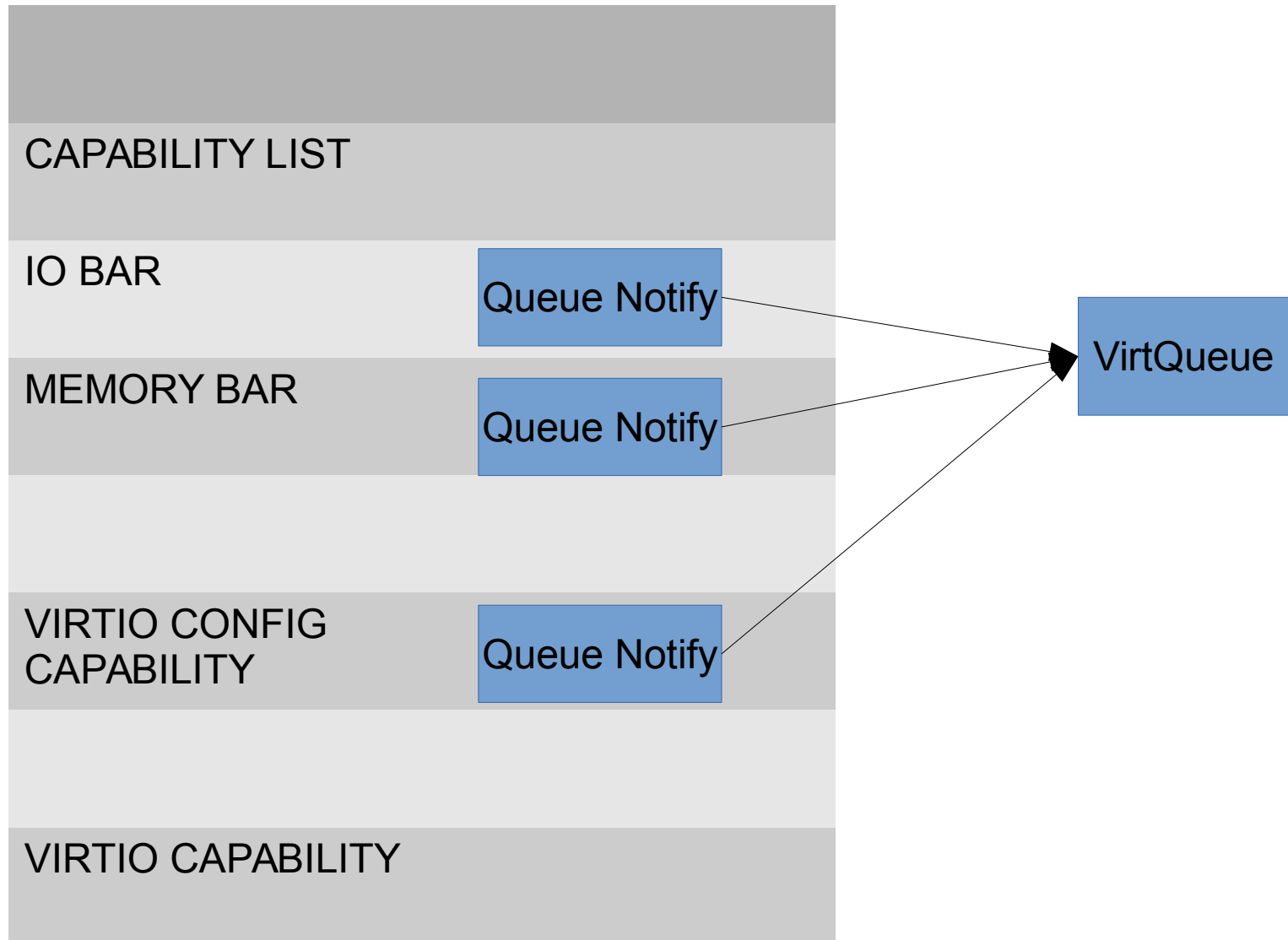


Virtio 1: Port IO vs Memory

	Port IO	MM IO
x86 decode: address	✓	✓
Fast on x86	✓	✓
32/64 bit	✗	✓
Page tables	✗	✓
Required by PCI Express	✗	✓



Memory Region Aliases





Ethernet MAC

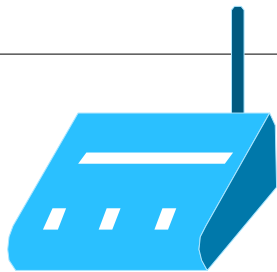
soft mac

52	54	00	12	34	56
----	----	----	----	----	----



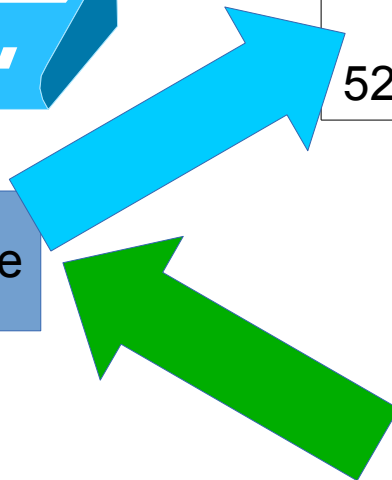
0.9

1.0



VirtQueue

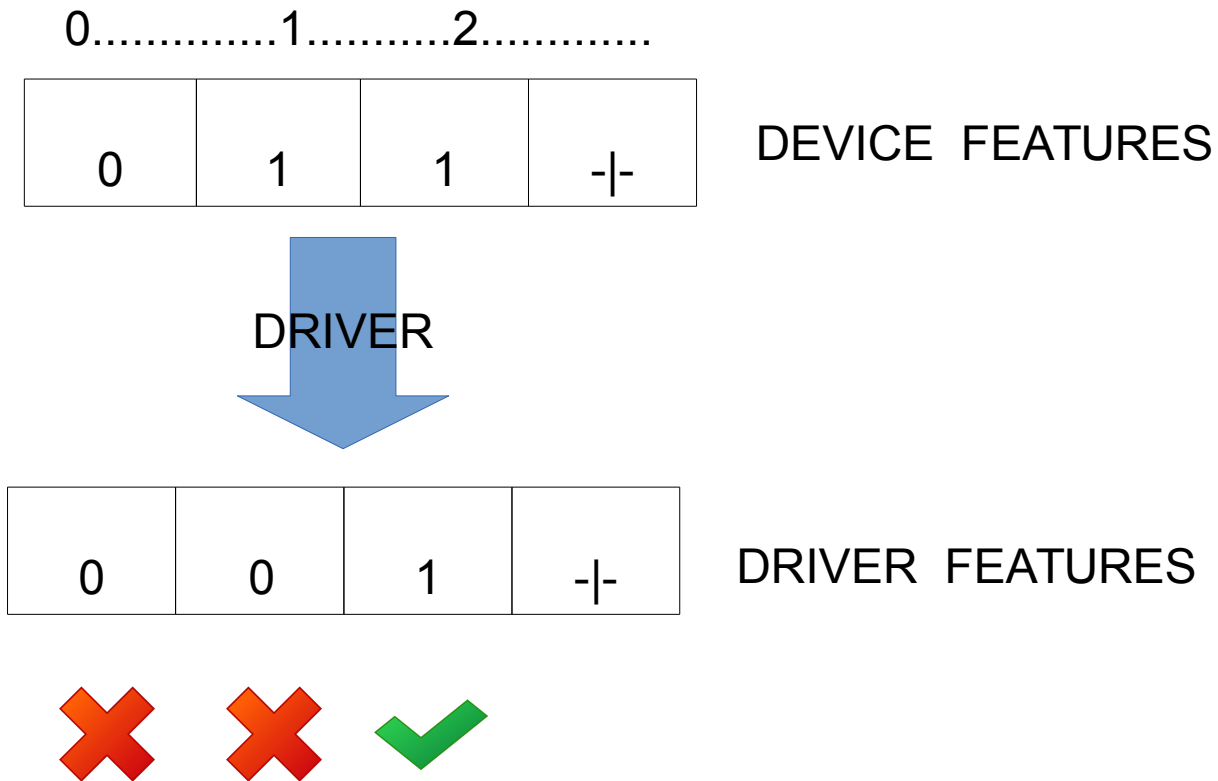
52	54	00	12	34	56
----	----	----	----	----	----



DRIVER



Virtio feature negotiation



Defaults must be maintained forever!



Virtio 1: Error handling

- DRIVER: set features
- DRIVER: set FEATURES_OK bit
- DEVICE: check features
- DEVICE: clear FEATURES_OK on error
- DRIVER: check FEATURES_OK bit
- DRIVER: fail gracefully if not set

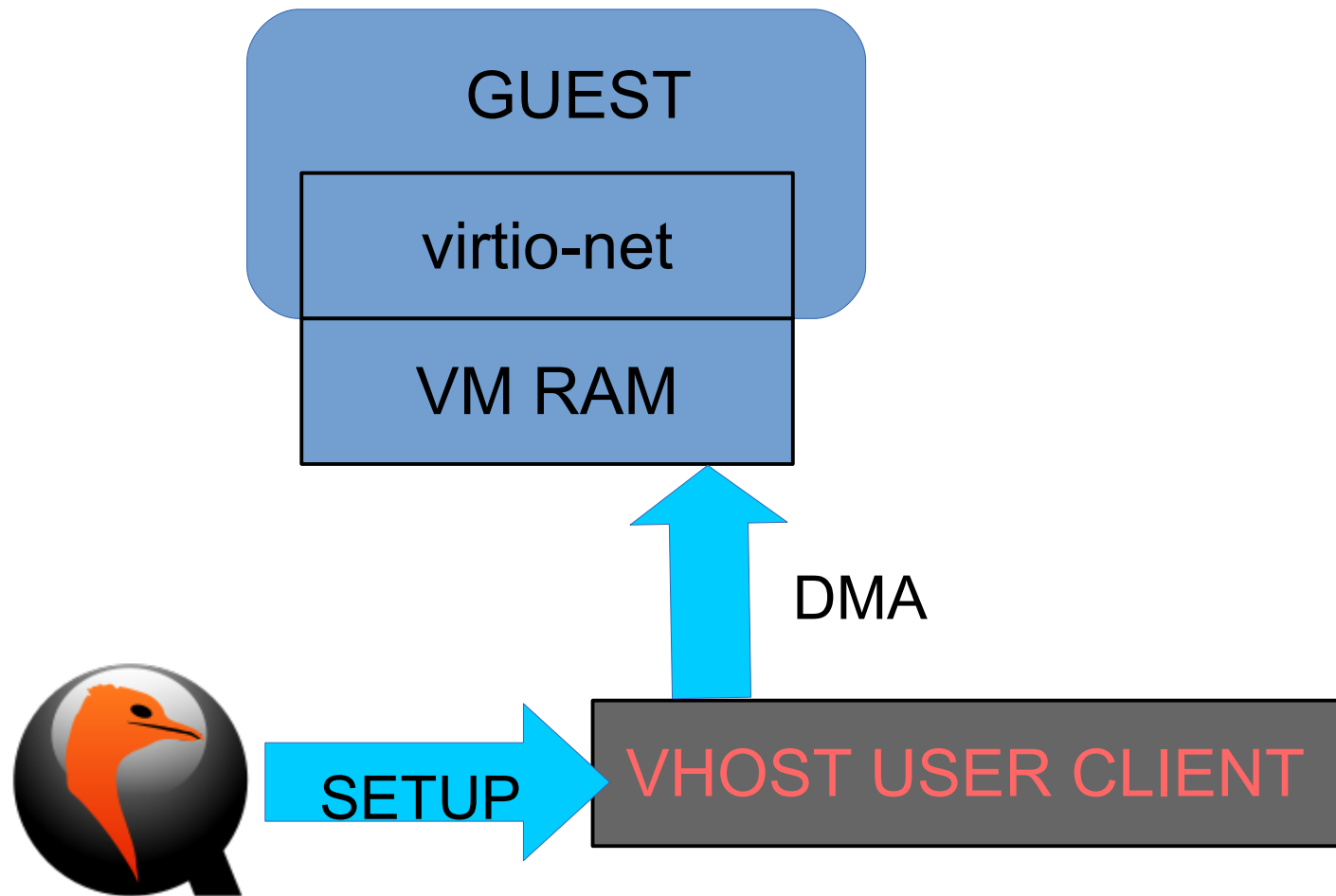


Error handling: Virtio 0.9

- Can't recover from device errors
- Not very useful?
- Just stop guest.



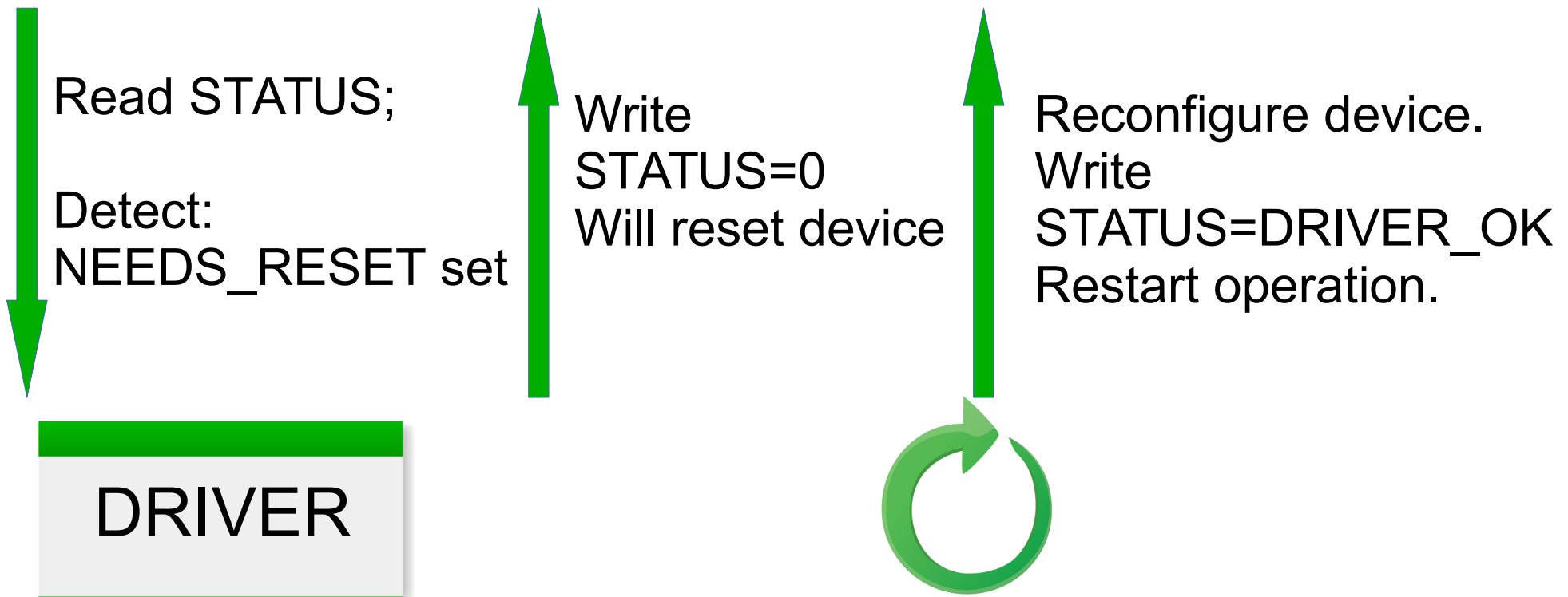
Vhost-user



Client crash or restart need not cause guest crash!

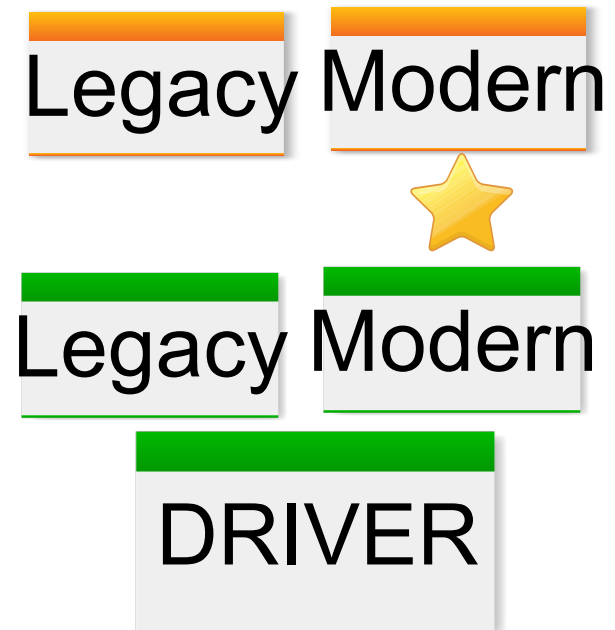


DEVICE_NEEDS_RESET

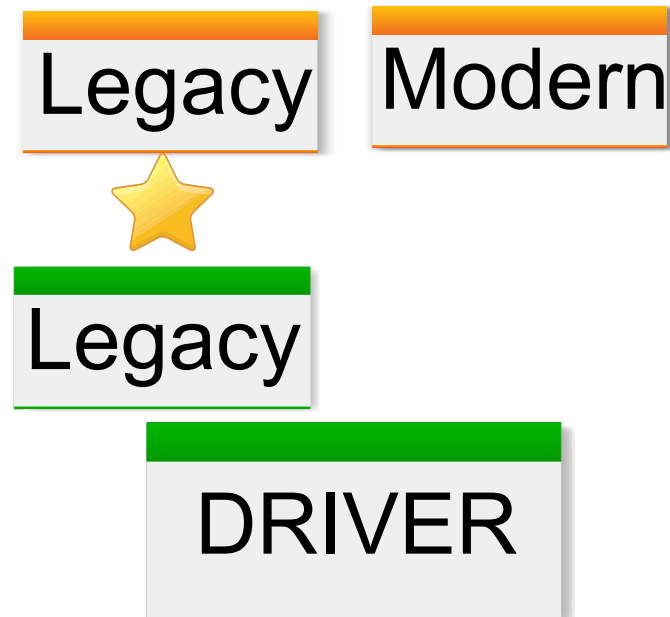


Compatibility

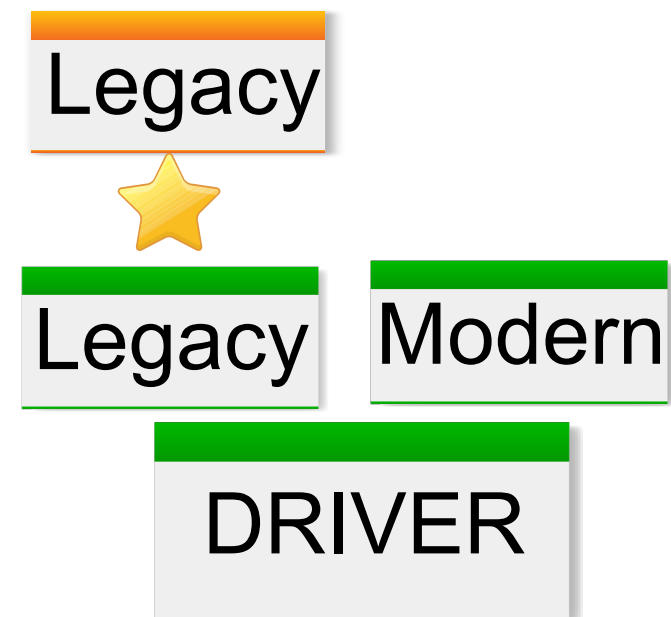
Transitional
Device & Driver



Legacy Driver



Legacy Device



Are we there yet?

GUEST



BIOS



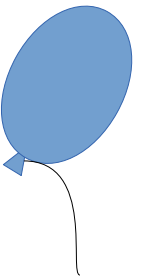
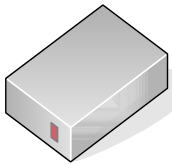
VHOST USER



KVM

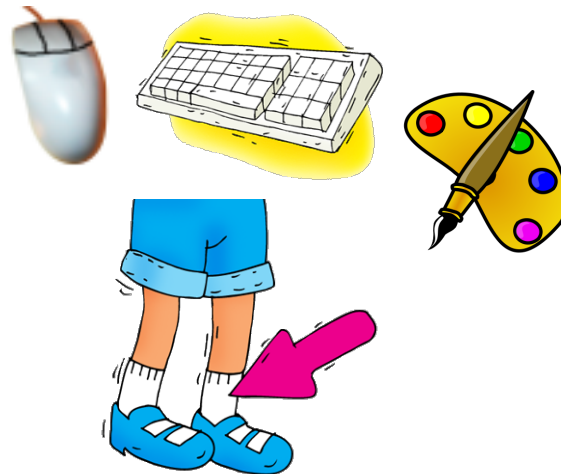


VHOST

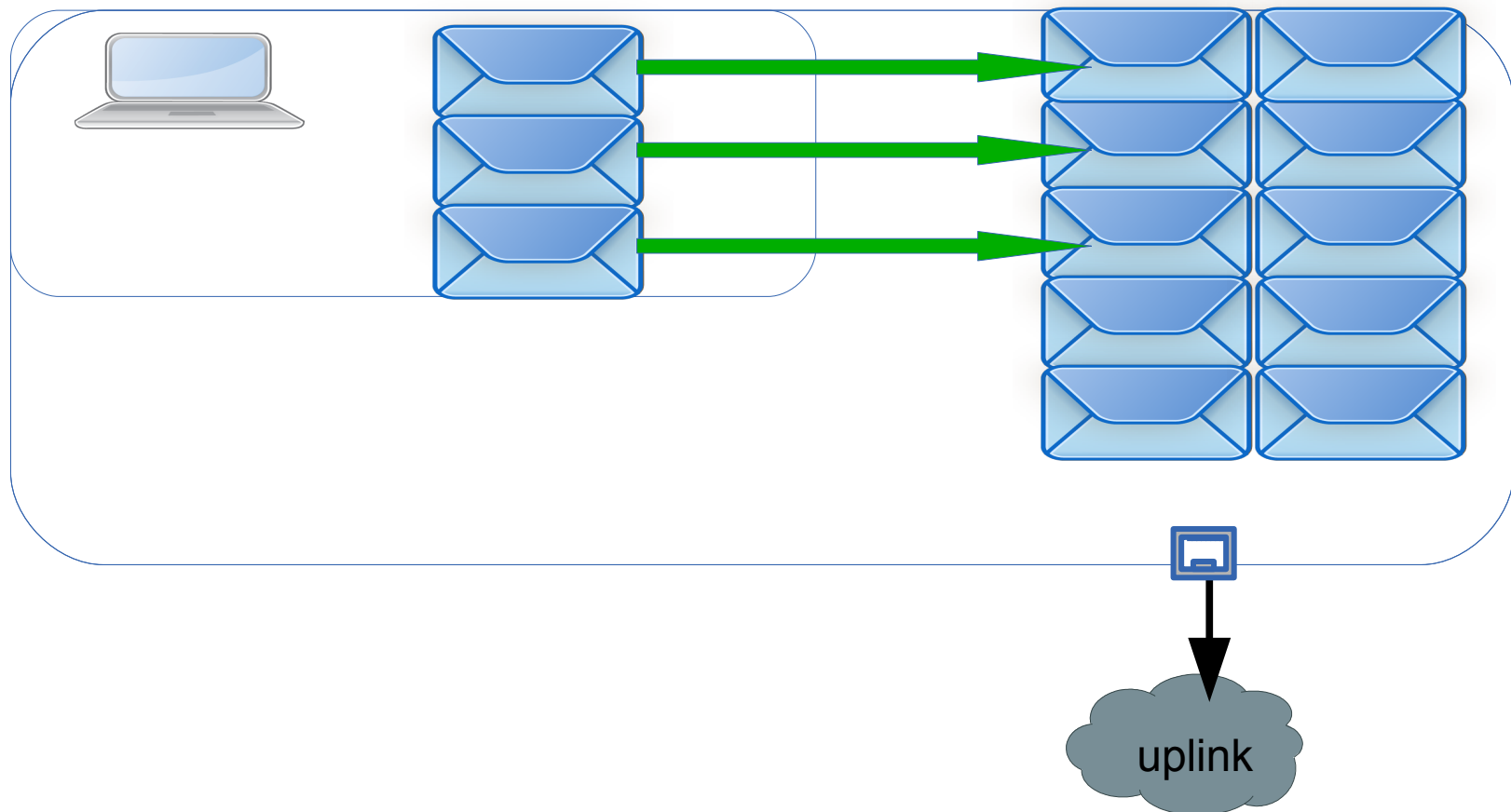


What to expect?

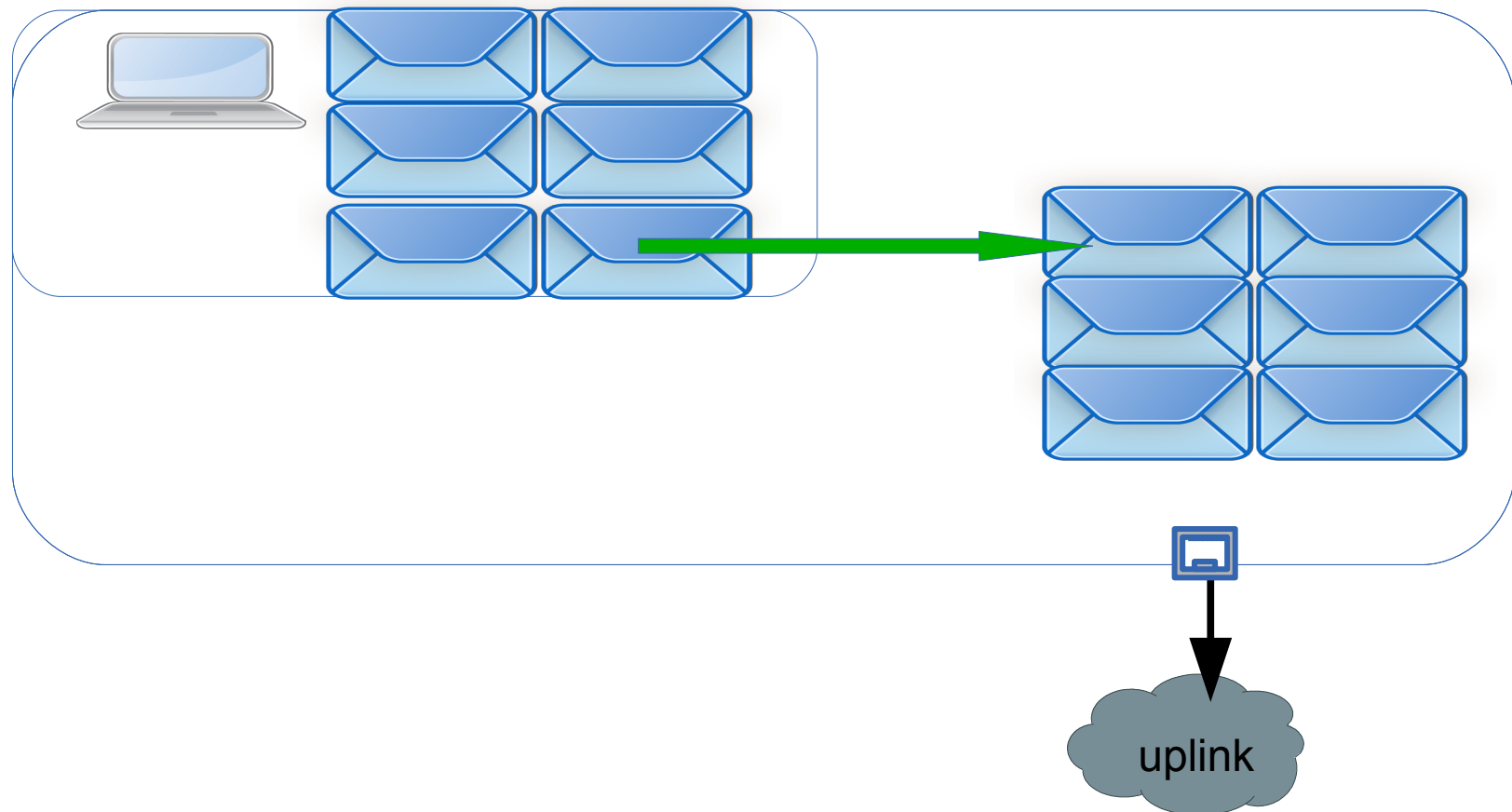
- Current: Virtio-v1.0-cs03
- Next bugfix: Virtio-v1.0-cs04
 - Virtio-blk: writeback / writethrough control
 - More update guidance
- Next feature: Virtio-v1.1-cs01
 - Virtio-input
 - Virtio-gpu
 - Virtio-vsock



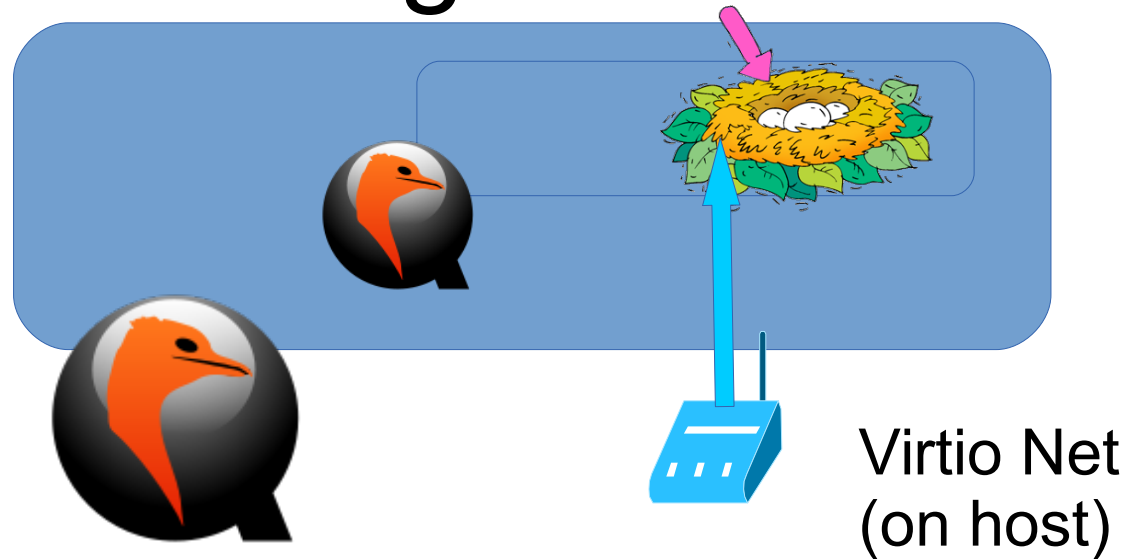
TX: Interrupt avoidance



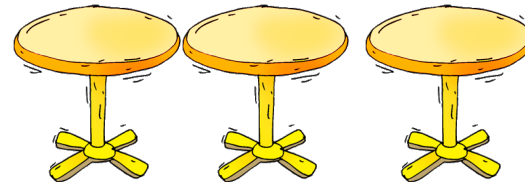
TX: Interrupt coalescing



Pass-through for nested virt



- Memory mapped: use page tables



- IOMMU: translate and protect guest memory



Virtio as PCI Express device

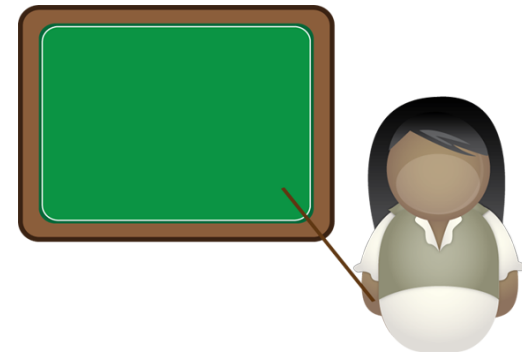


- Uses memory mapped IO support
- Multi-root for NUMA
- Native hotplug
- Advanced Error Reporting



Summary

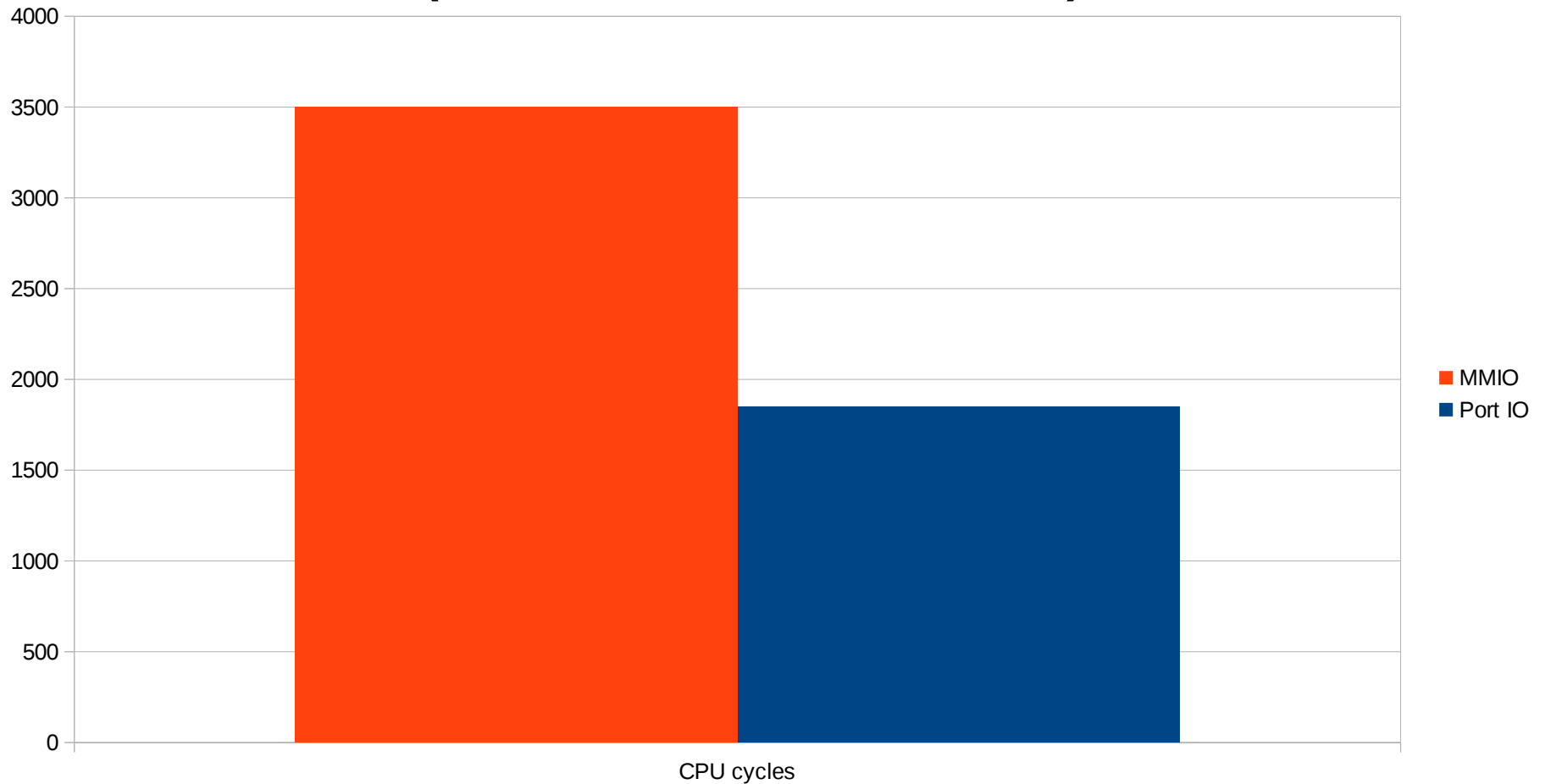
- Why do it?
 - Improved robustness for virtual devices
- Are we there yet?
 - Yes!
 - And there's more to come.



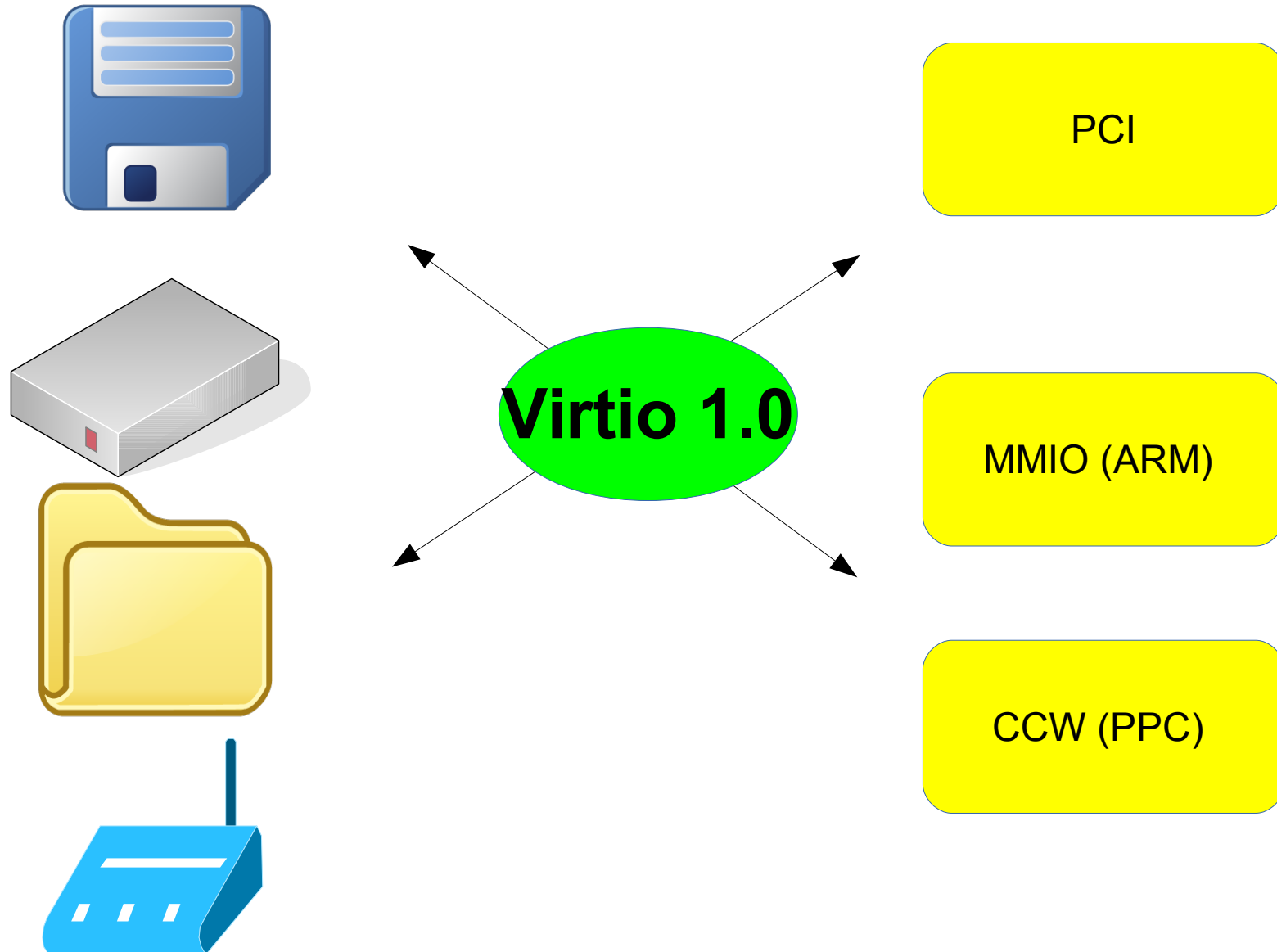
Thank you!



Virtio 0.9: Port IO versus memory on KVM x86: cycles per access (lower is better)



OASIS Virtio TC

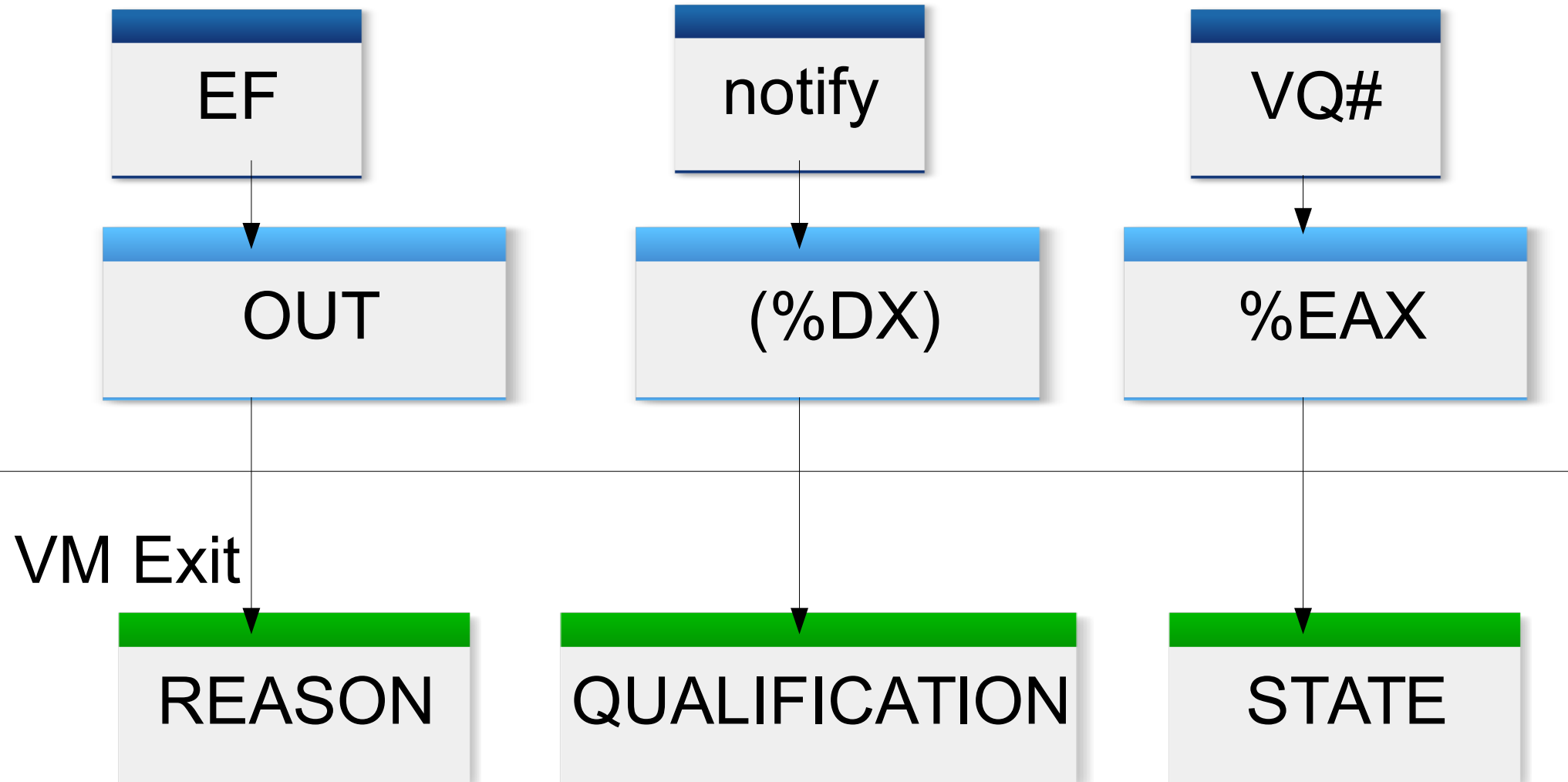


Virtio 1.0

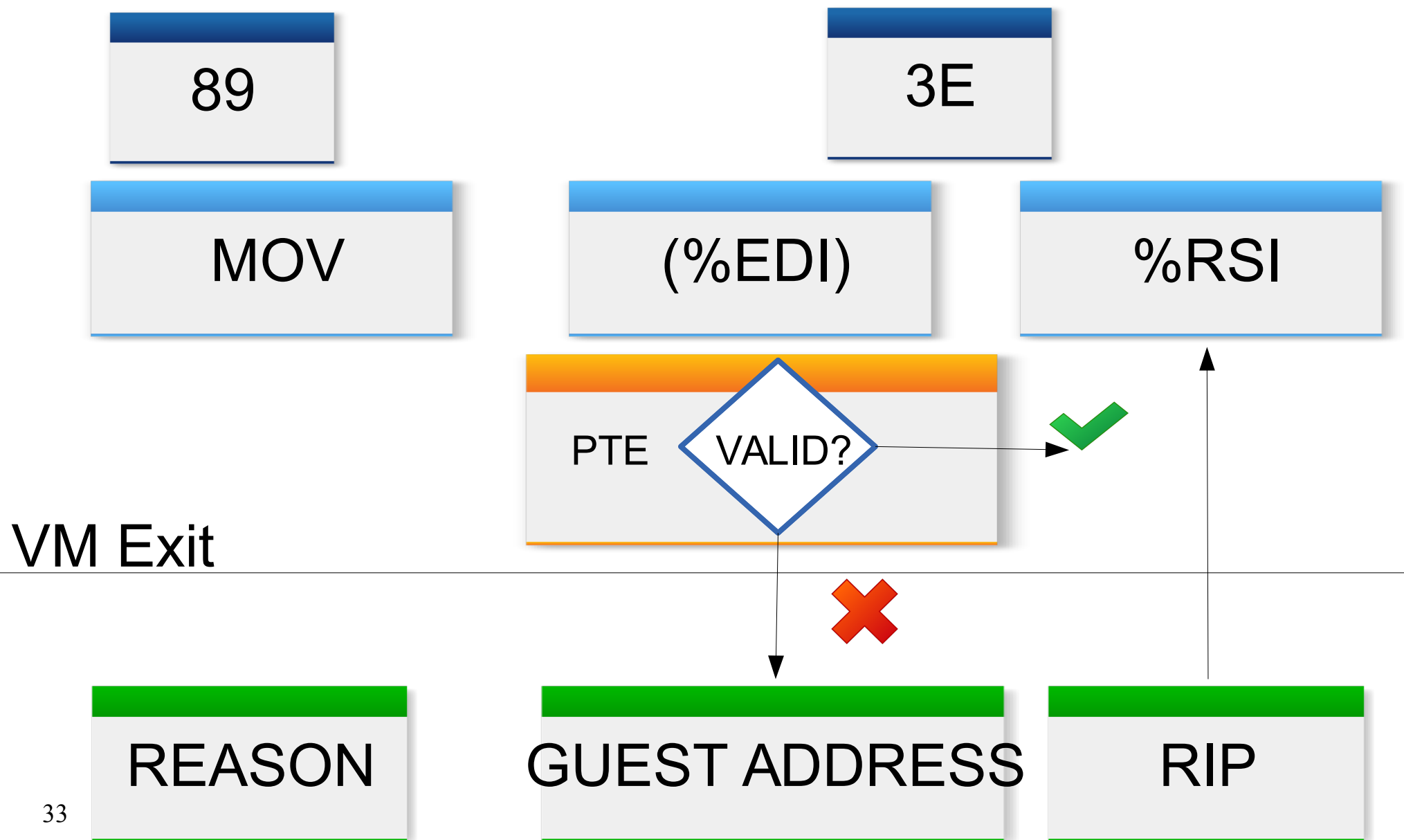
- Virtio PCI:
 - Replace Port IO with Memory mapped IO
 - PCI Express (hotplug, AER, multi-root, SRIOV)
 - Infinite features
- Reduced memory requirements
- Fixed endianness
- Compatibility



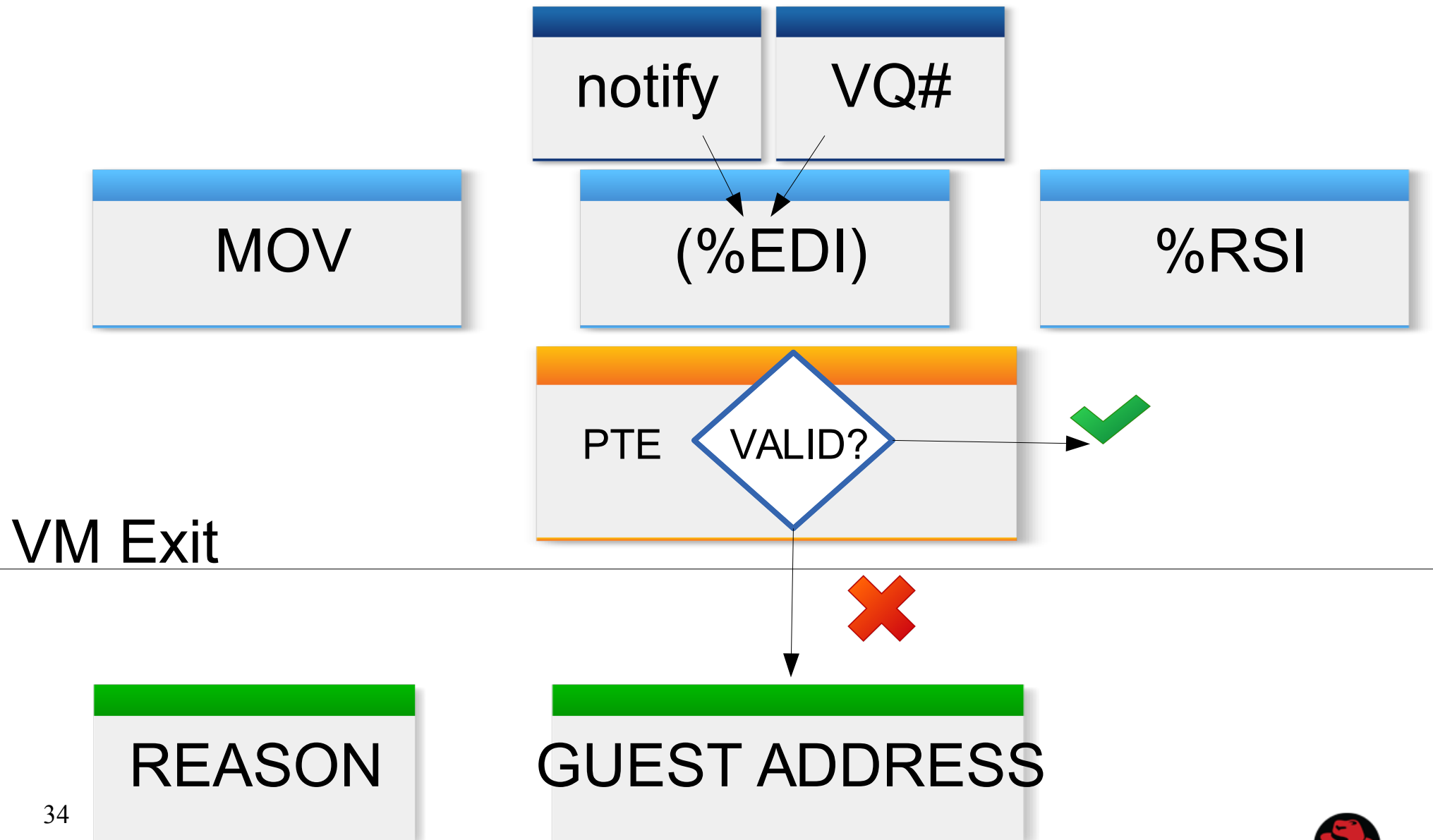
Port IO: outl



Memory mapped IO: writel



Fast MMIO

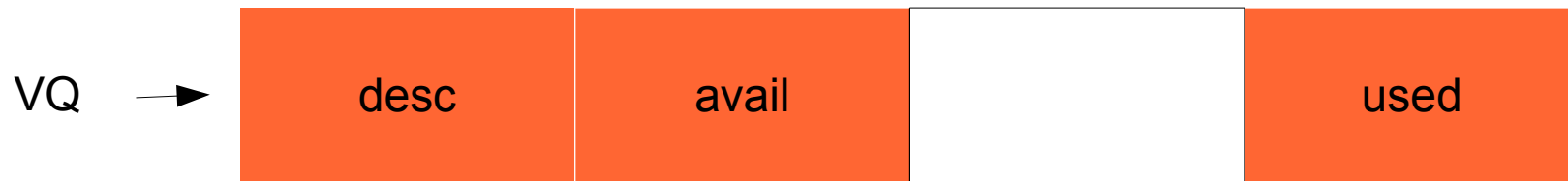


Multiple interfaces

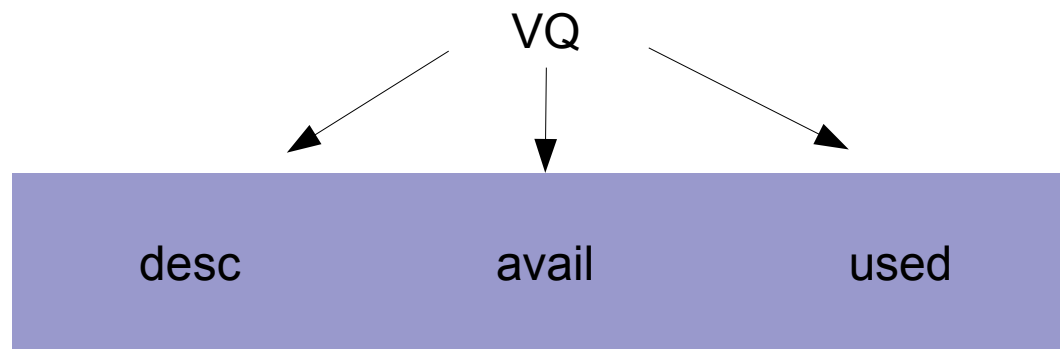


Memory requirements

0.9



1.0



features

0	1	1	- -
---	---	---	-----

DRIVER

0	1 v 0	1	- -
---	-------------	---	-----

1.0

SEL 1 2 3 4

0...

.....

DRIVER

A sequence of five boxes, each containing a different number of dots (4, 3, 2, 1, and an ellipsis), illustrating a decreasing pattern.

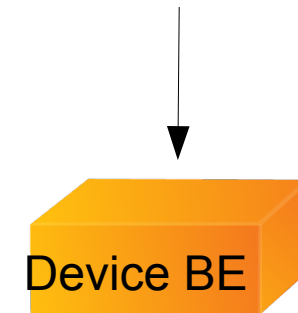
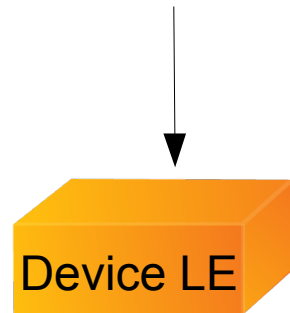
STATUS = FEATURES_OK

Endianness

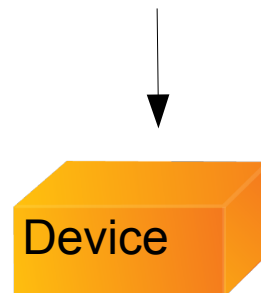
intel

PPC

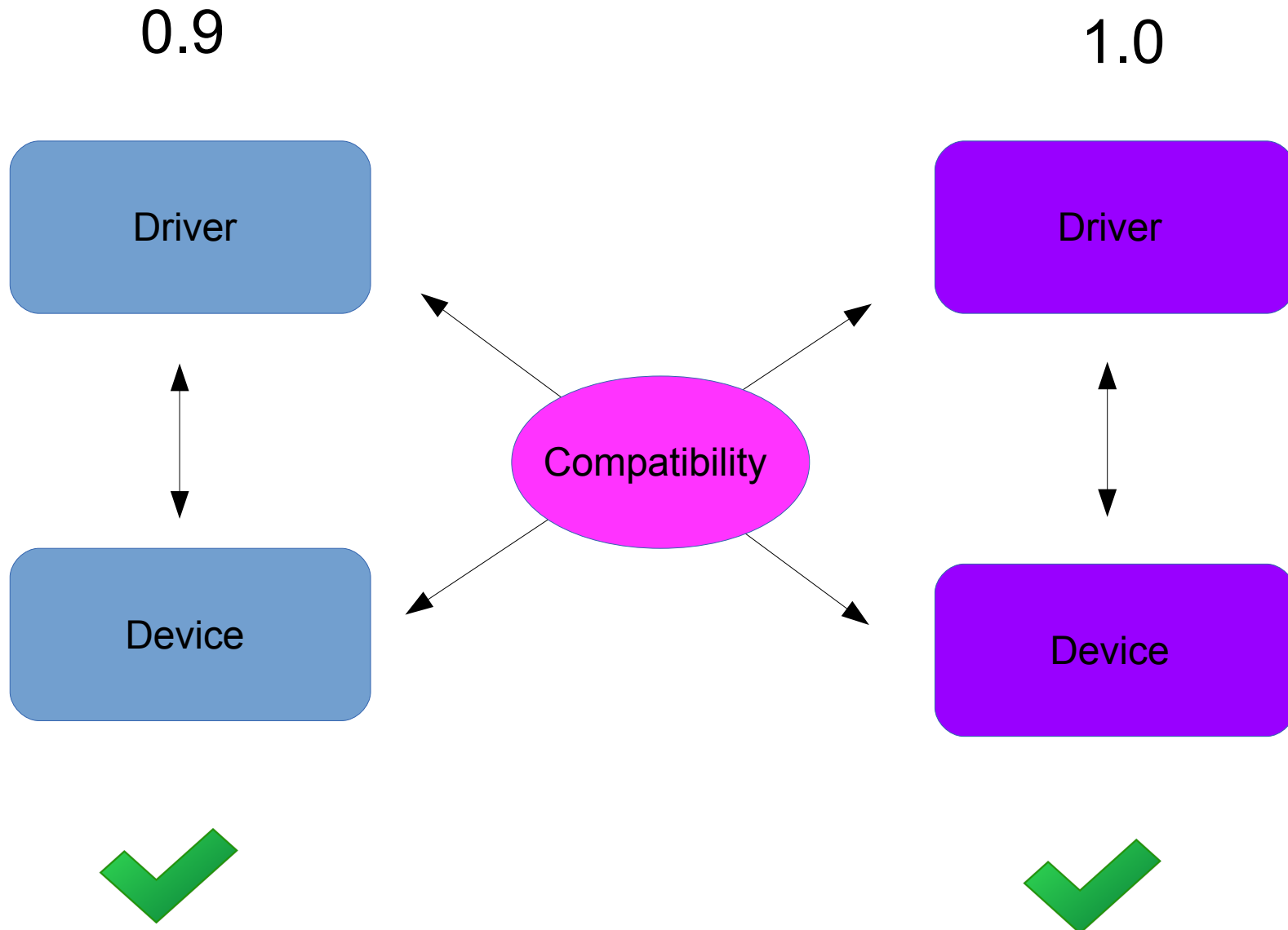
Virtio 0.9



Virtio 1.0

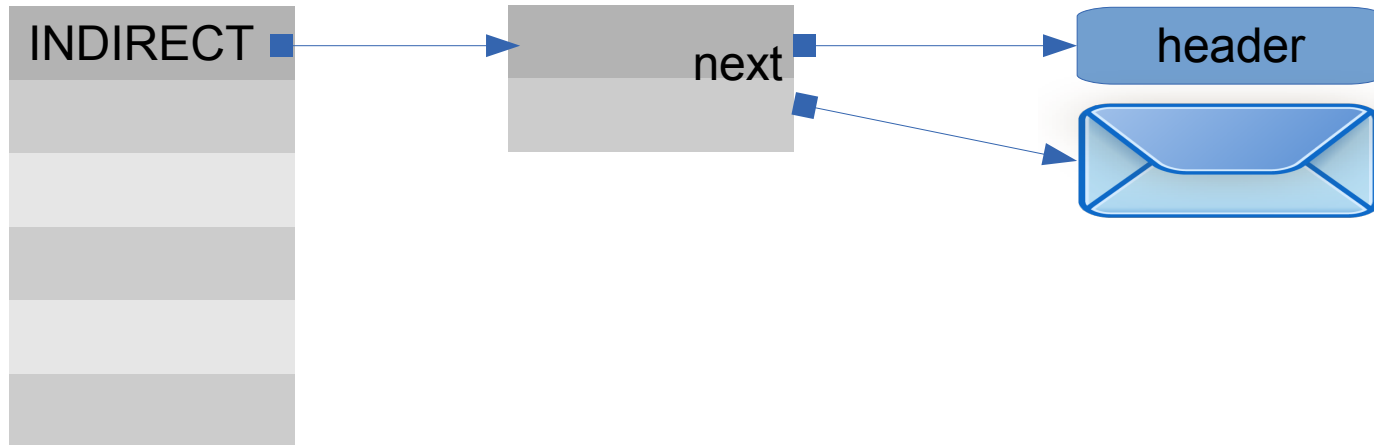


compatibility

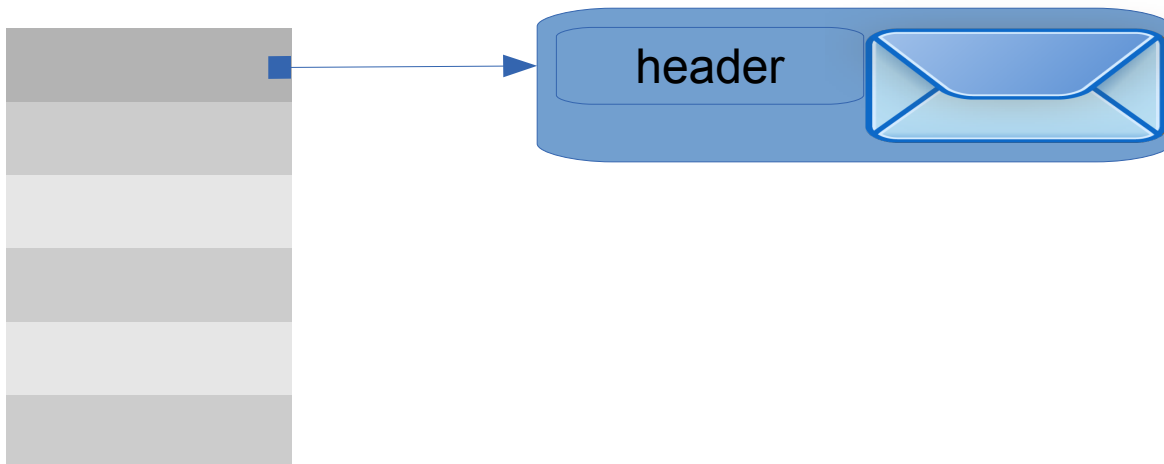


Packet layout

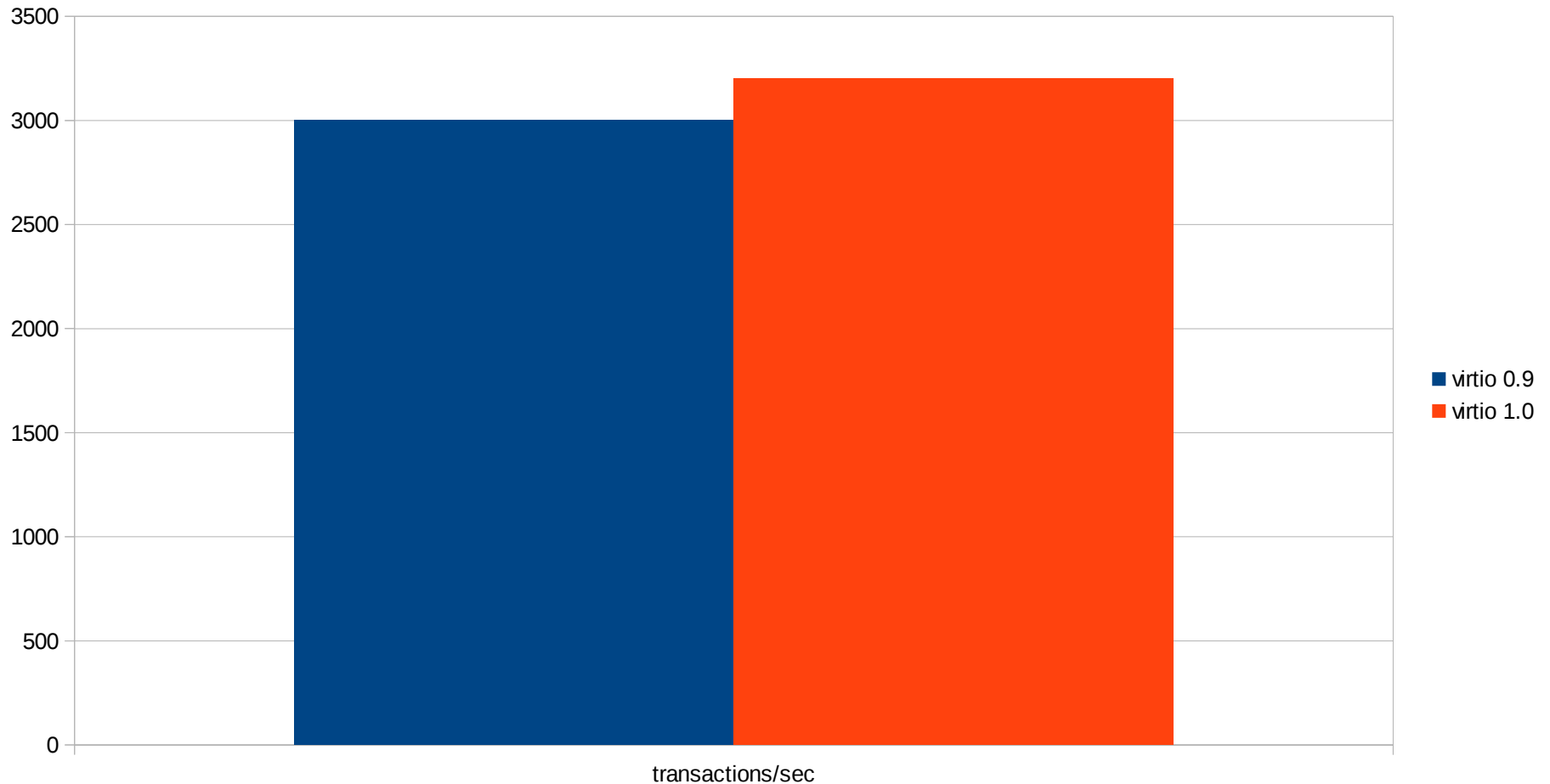
Virtio 0.9



Virtio 1.0



Packet layout: transactions per sec (higher is better)



More: virtio 1.0 versus 0.9.5

- Virtio 9p 
- Virtio blk: WCE 
- Virtio-net Multiqueue 
- Virtio-net dynamic offloads
- Already upstream (based on spec draft)



vhost updates

- Vhost scsi  
- Vhost-net zero copy transmit  
- No need for driver changes



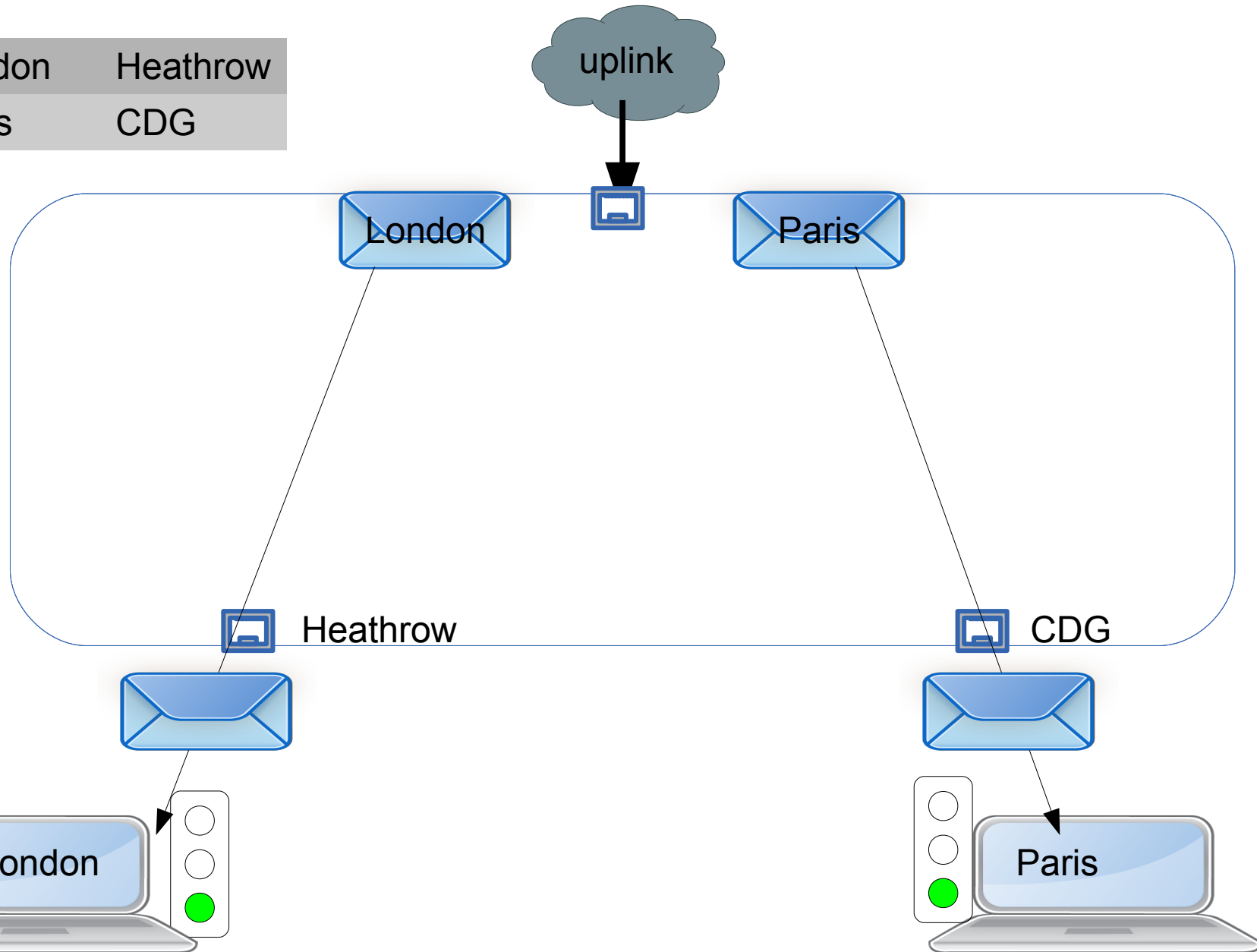
Kvm networking

- Openvswitch – if time allows
- Ethernet bridge

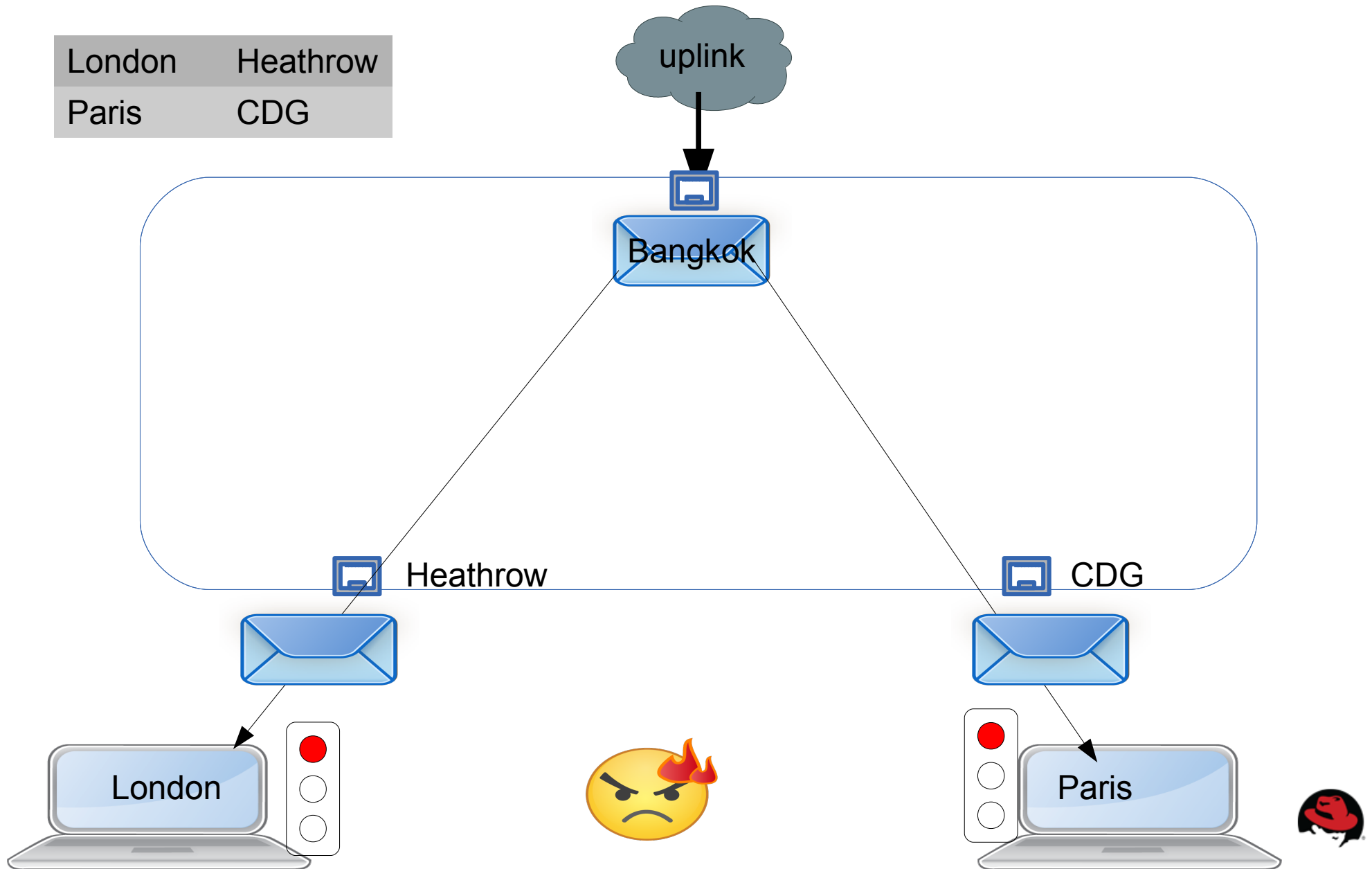


Bridge FDB

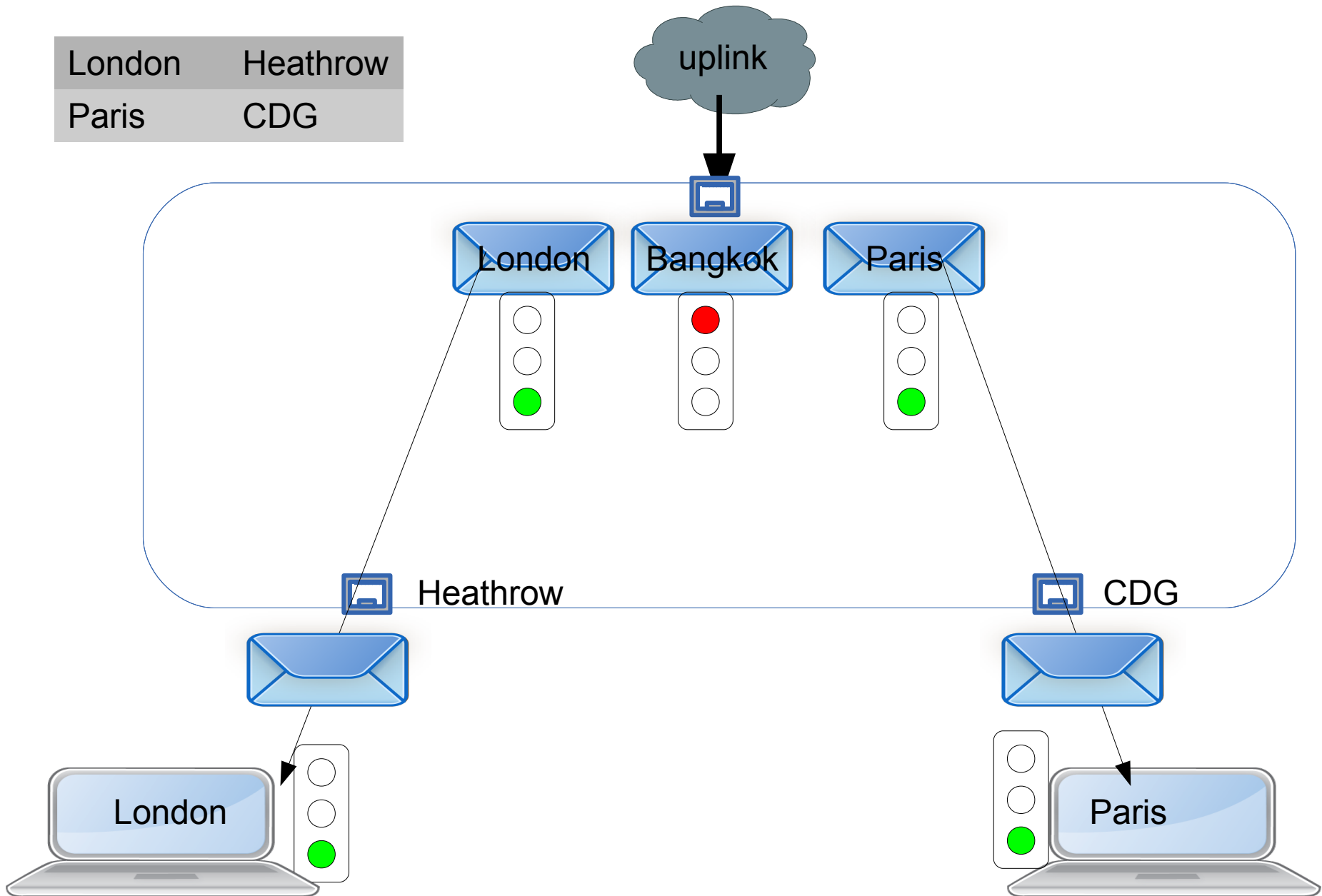
London	Heathrow
Paris	CDG



Flood: DOS potential

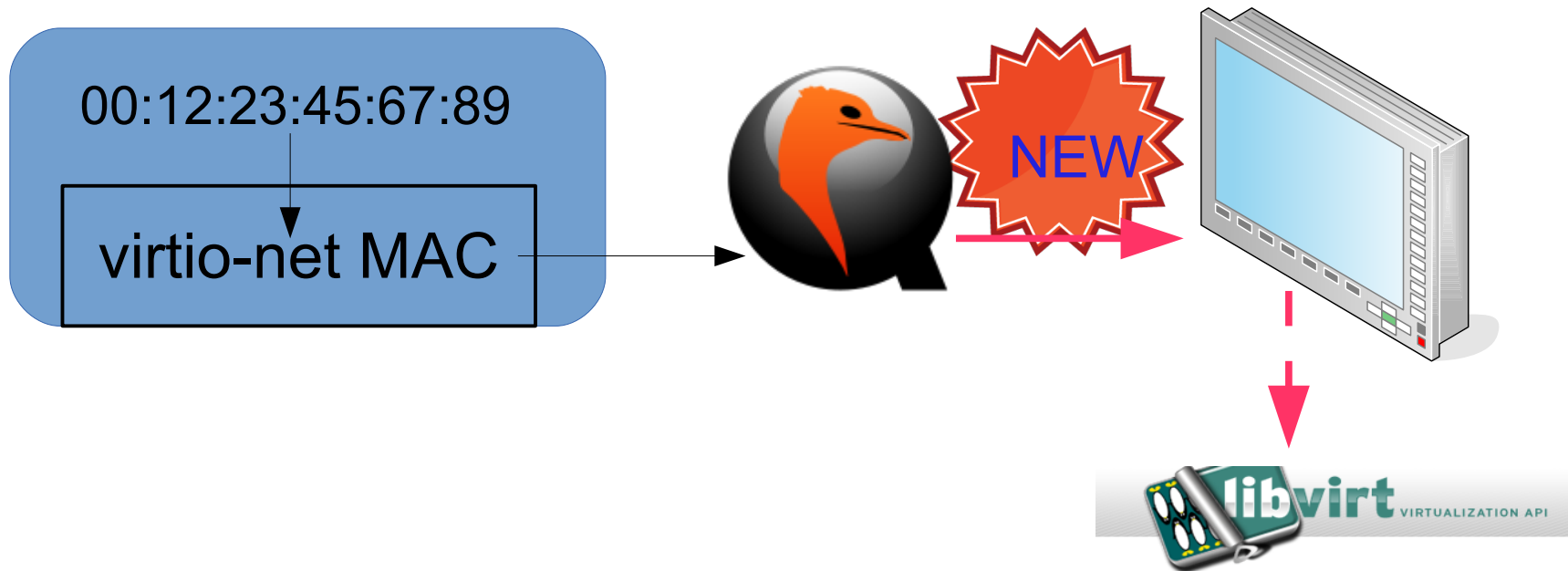


Disable flood



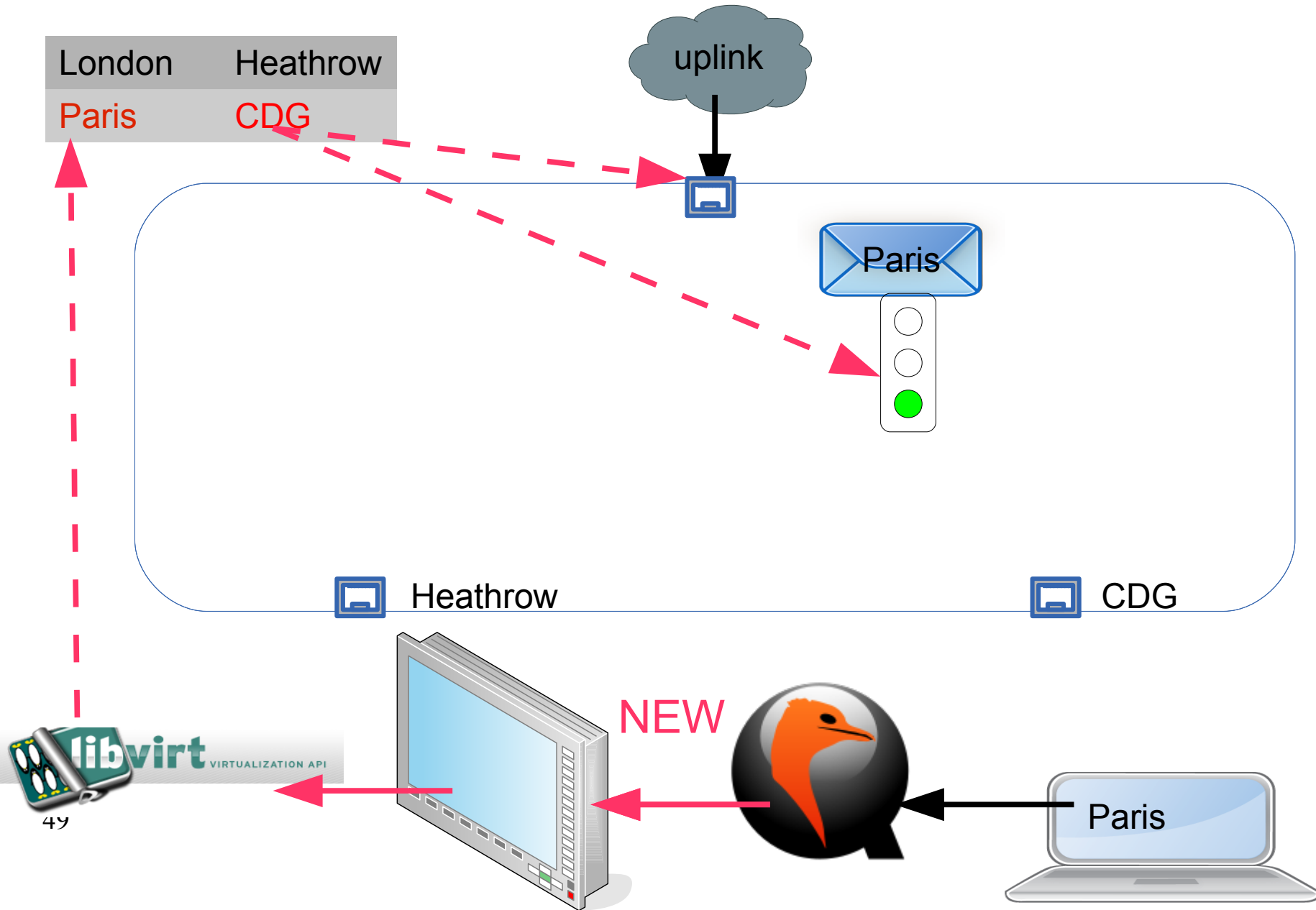
softmac

- `Ifconfig eth0 hw ether 00:12:23:45:67:89`

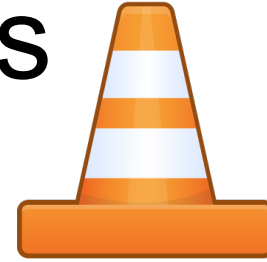


The diagram illustrates a virtualization architecture with the following components and connections:

- Hosts:** Two hosts are shown at the bottom: a server labeled "virt VIRTUALIZATION API" and a laptop labeled "Paris".
- Virtual Machines (VMs):** Two VMs are shown within a large rounded rectangle: "Heathrow" (represented by a server icon) and "CDG" (represented by a server icon).
- Network:** A cloud labeled "uplink" is connected to the "Heathrow" VM via a black arrow. Dashed red arrows show network paths from the "Heathrow" VM to the "CDG" VM and from the "CDG" VM to the "Paris" VM.
- Migration:** A black arrow points from the "Paris" laptop to a black circle containing an orange bird icon (representing a VM image). A red arrow labeled "NEW" points from this circle to the "virt" server, indicating a new VM creation or migration.



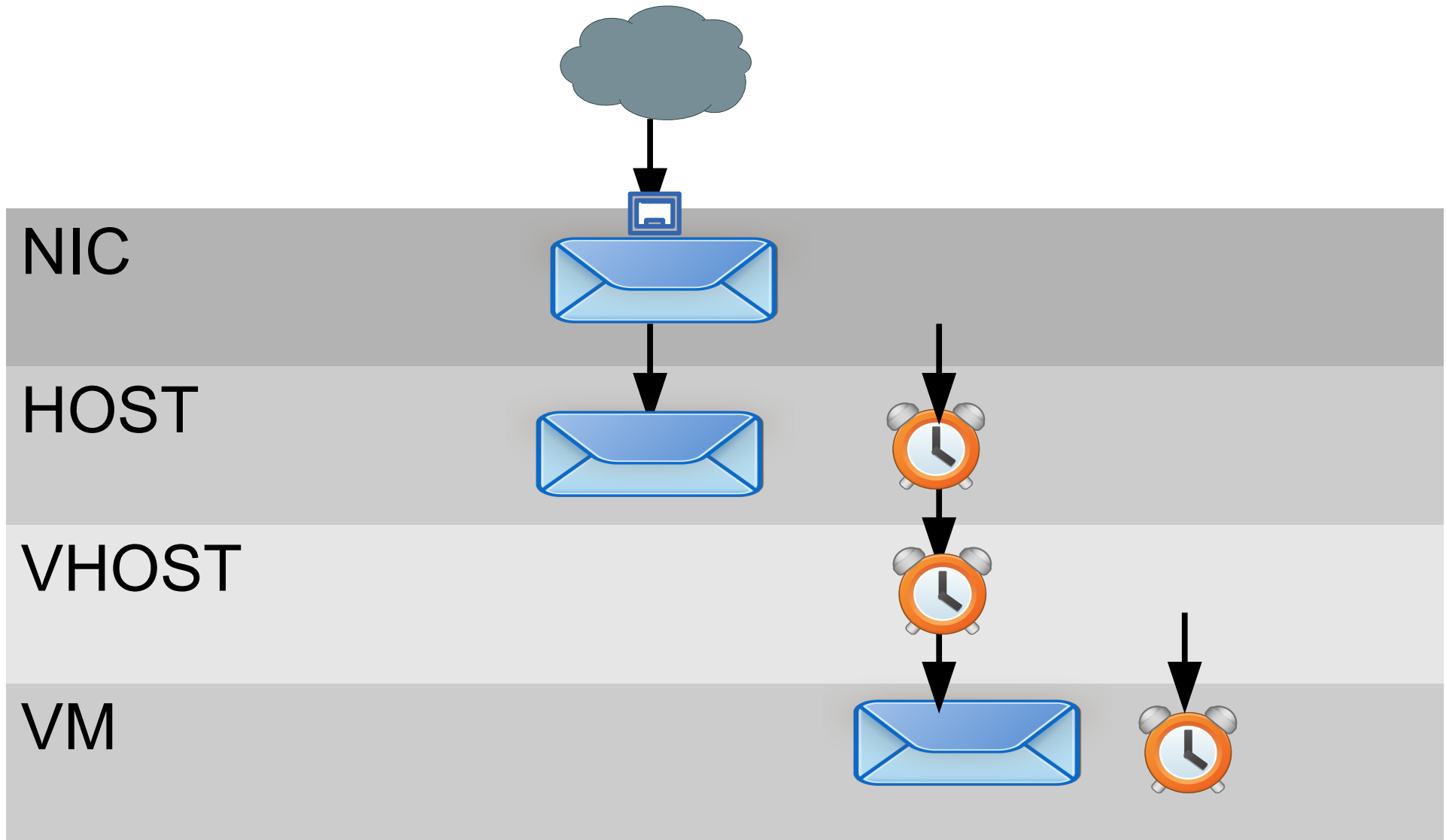
Work in progress



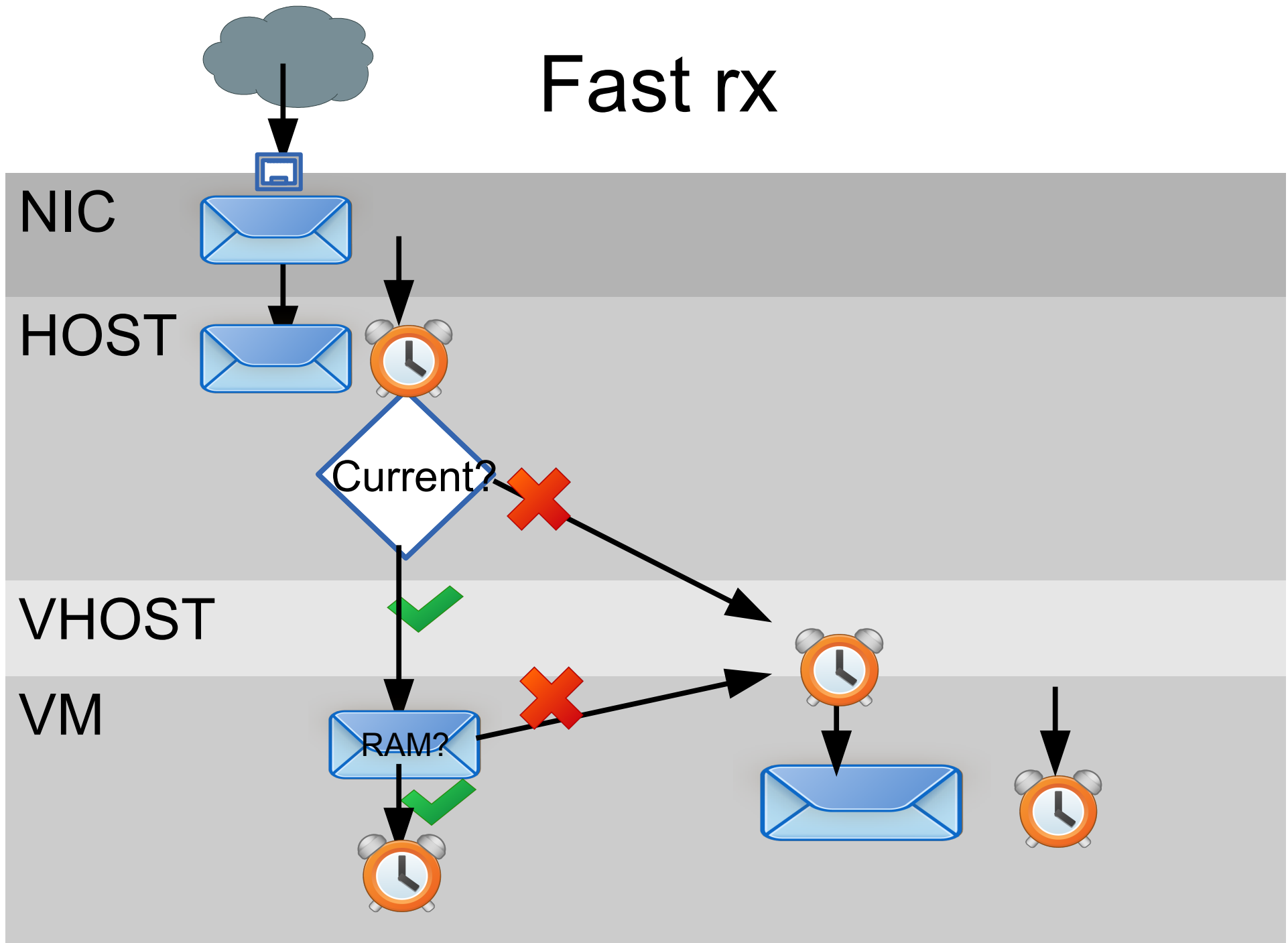
- ELVIS (vhost blk/vhost net)
- Virgl
- Vhost-net performance



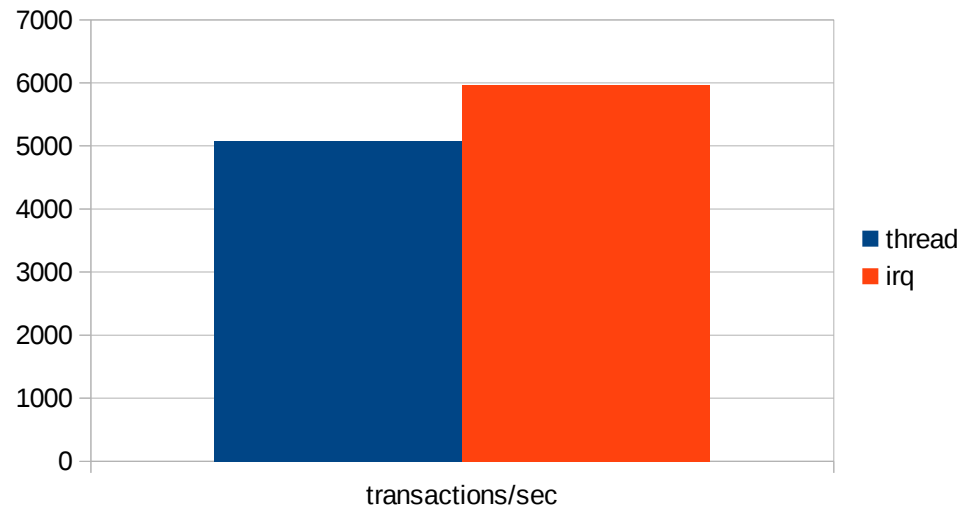
RX latency



Fast rx



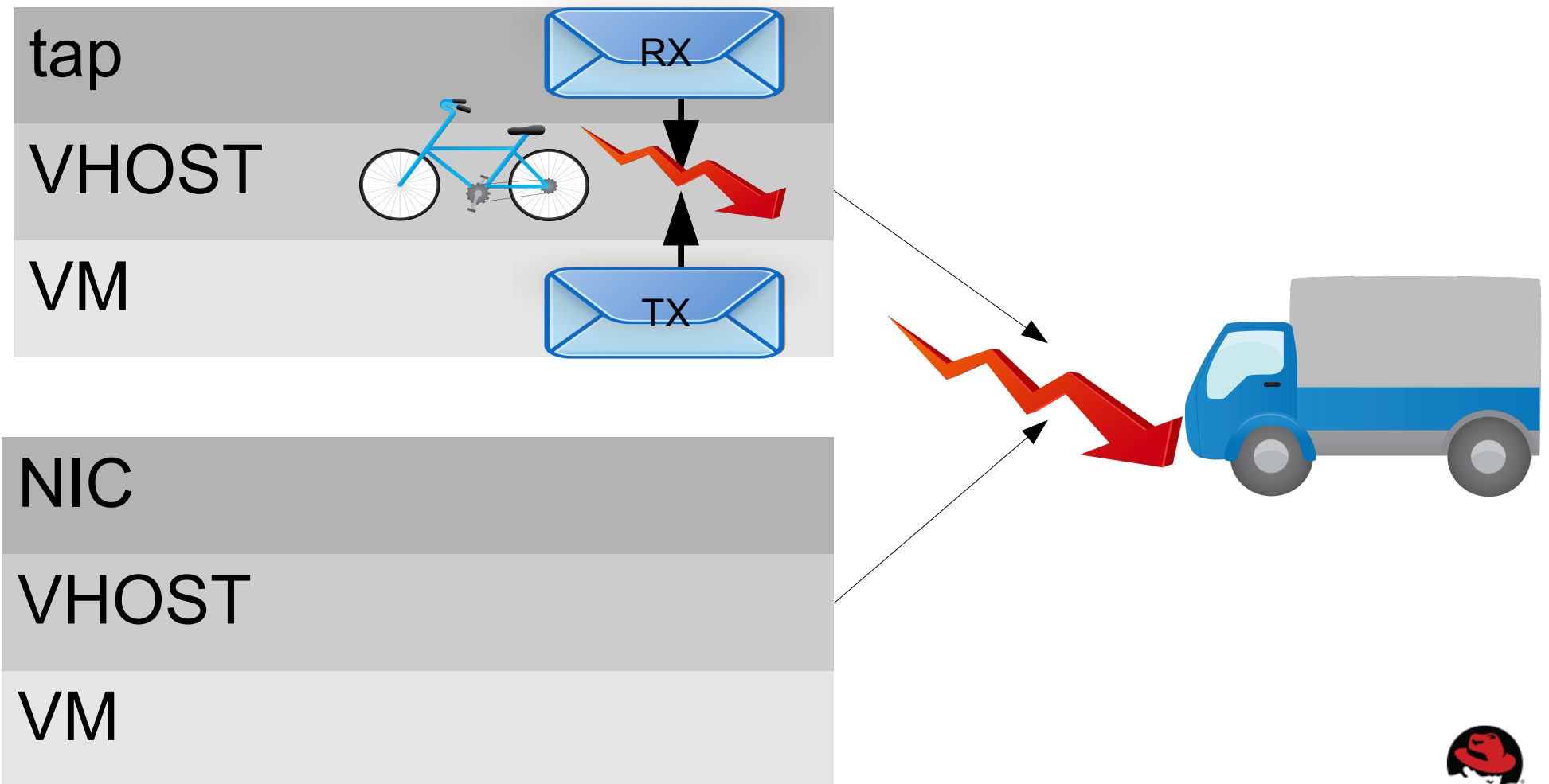
Fast rx: transactions per sec (higher is better)



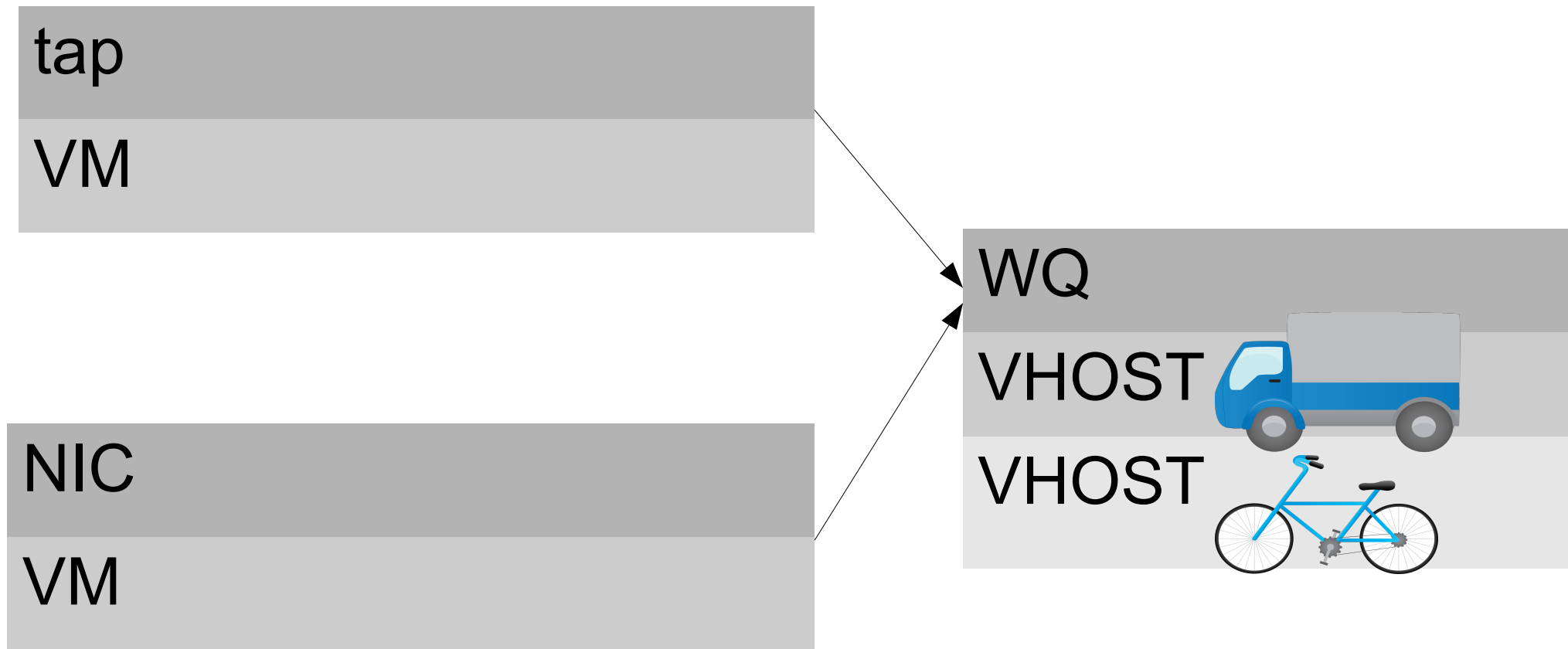
Hit	331668
Miss	79



Vhost-net threading

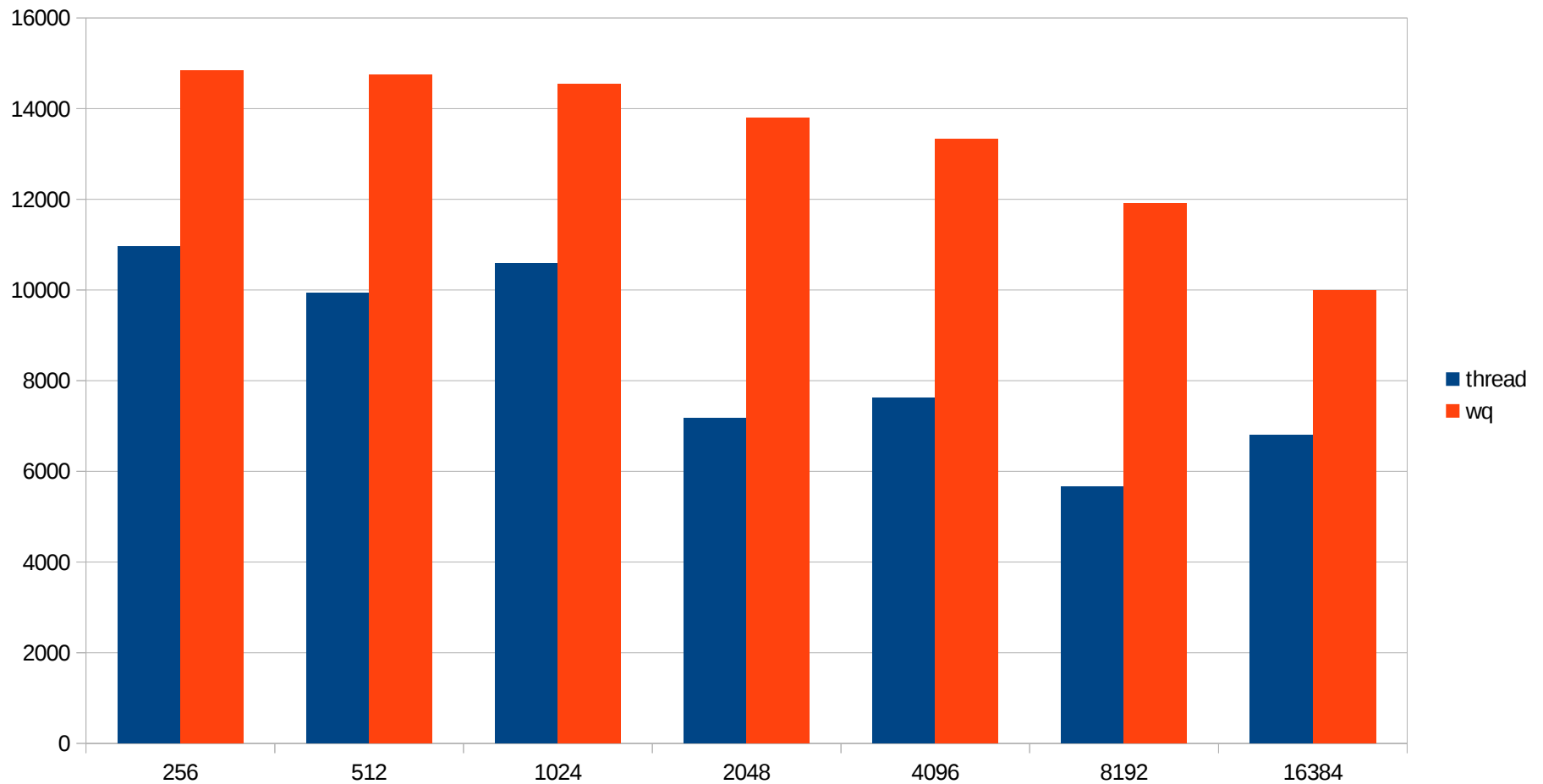


Vhost-net thread pool



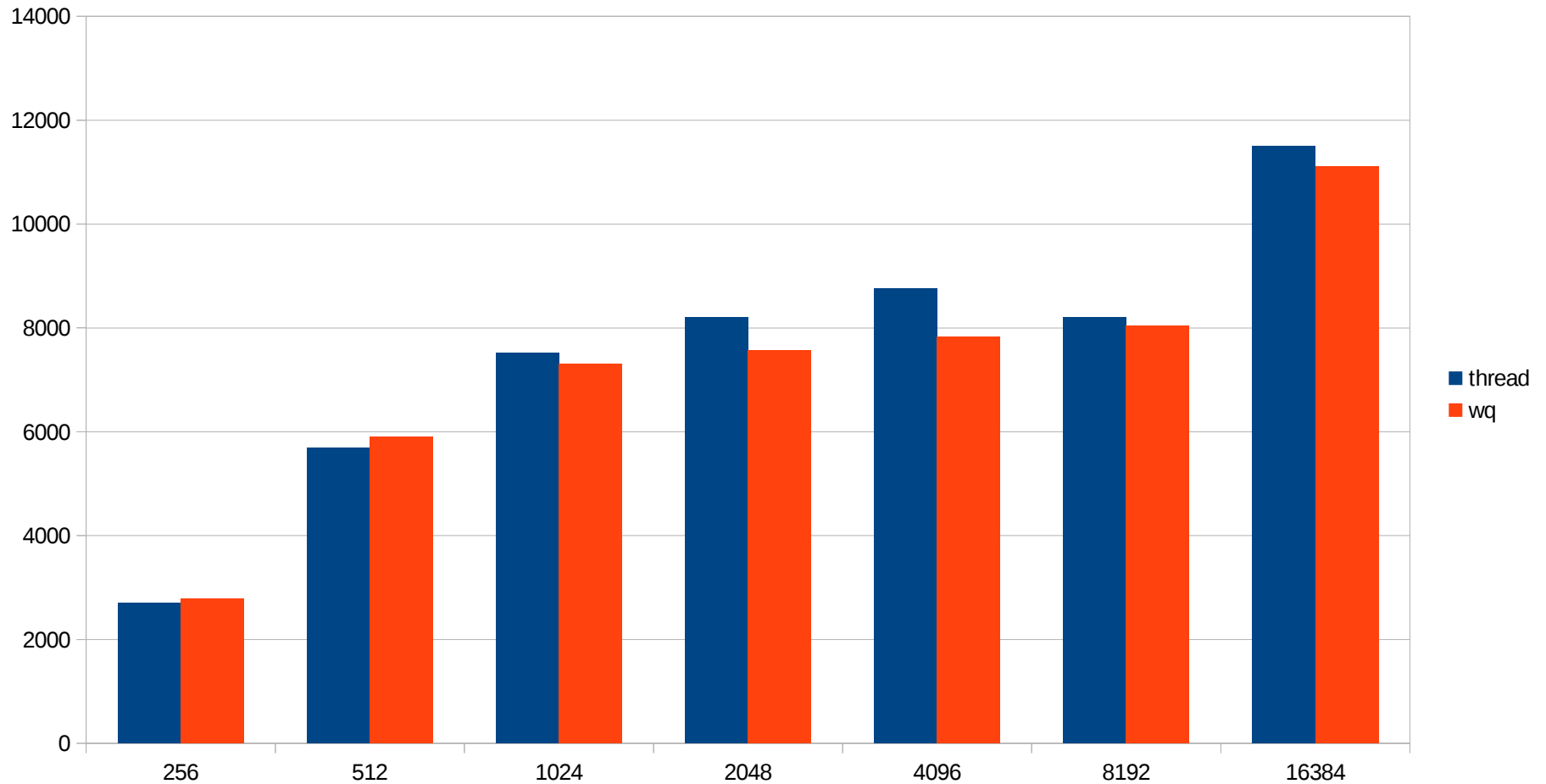
threading: UDP RR

transactions/sec (higher is better)



threading: TCP STREAM

transactions/sec (higher is better)



summary

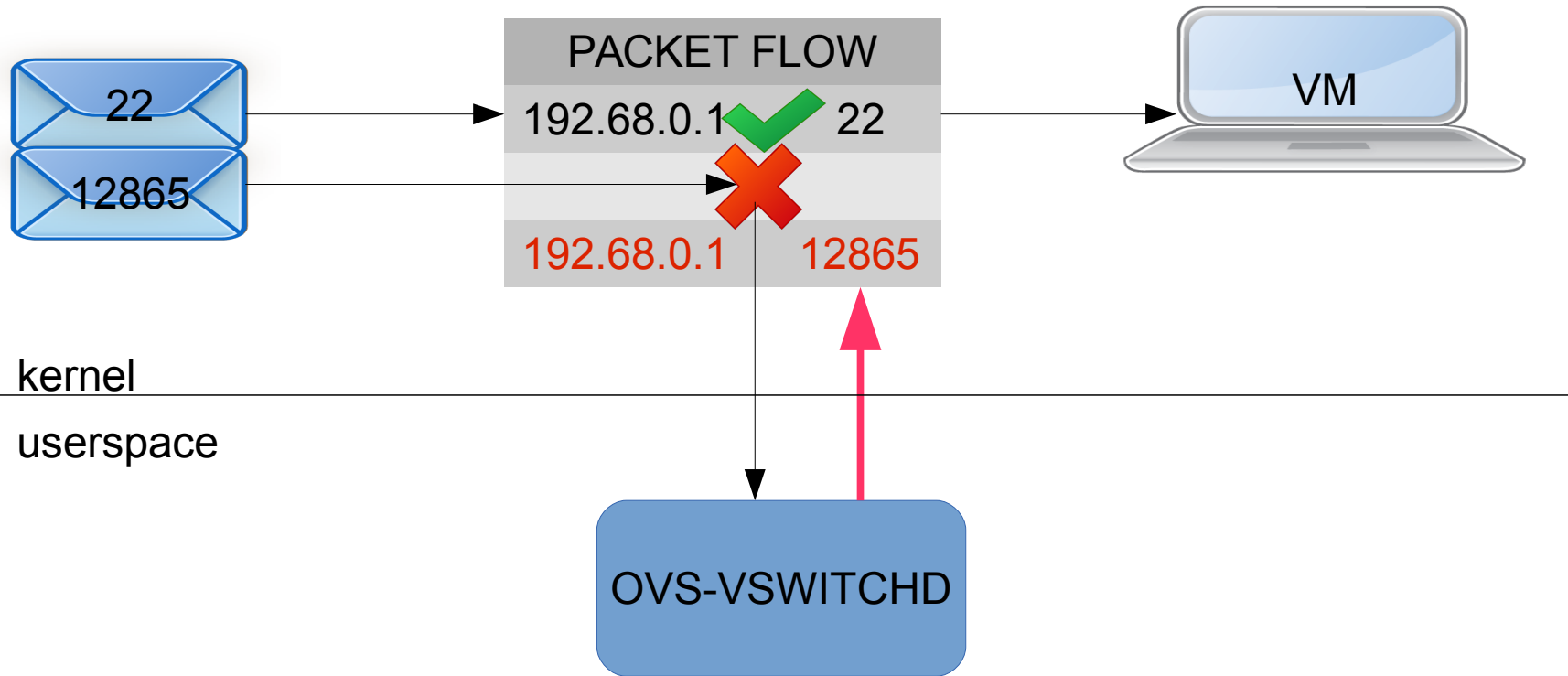
- Performance
- Manageability
- Security



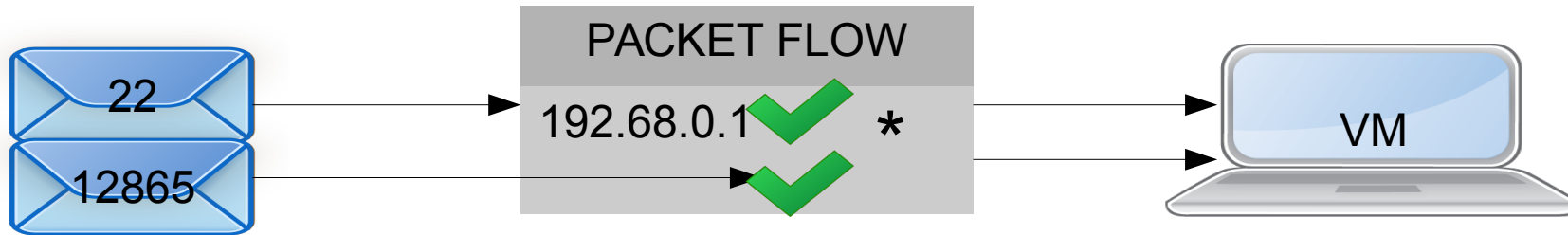
Questions?



OVS: flow match



OVS: wildcard match



kernel

userspace

OVS-VSWITCHD



Wilcard: netperf CRR (higher is better)

