



Smart NICs

Ian Pratt

Smart L2 NIC features



- Privileged/unprivileged NIC driver model
- Free/rx/tx descriptor queues into guest
- Packet demux and tx enforcement
- Validation of frag descriptors
- TX QoS
- CSUM offload / TSO / LRO / intr coalesce

Smart L2 NIC features



- Packet demux to queues
 - MAC address (possibly multiple)
 - VLAN ttag
 - L3/L4 useful in some environments
- Filtering
 - Source MAC address and VLAN enforcement
 - More advanced filtering
- TX rate limiting: x KB every y ms

Design decisions



- Inter-VM communication
 - Bounce via bridge on NIC
 - Bounce via switch
 - Short circuit via netfront
- Broadcast/multicast
- Running out of contexts
 - Fallback to netfront
- Multiple PCI devs vs. single
- Card IOMMU vs. architectural

Memory registration



- Pre-registering RX buffers is easy as they are recycled
- TX buffers can come from anywhere
 - Register all guest memory
 - Copy in guest to pre-registered buffer
 - Batch, register and cache mappings
- Pinning can be done in Xen for architectural IOMMUs, dom0 driver for NIC IOMMUs

VM Relocation



- Privileged state relocated via xend
 - Tx rate settings, firewall rules, credentials etc.
- Guest can carries state and can push down unpriv state on the new device
 - Promiscuous mode etc
- Heterogeneous devices
 - Need to change driver
 - Device independent way of representing state
 - (more of a challenge for RDMA / TOE)

Design options



- Proxy device driver
 - Simplest
 - Requires guest OS to have a driver
- Driver in stub domain, communicated to via netchannel like interface
 - Overhead of accessing driver
- Driver supplied by hypervisor in guest address space
 - Highest performance
- “Architectural” definition of netchannel rings
 - Way of kicking devices via Xen