Network Interface Cards

ECE/CS598HPN

Radhika Mittal
Network Interface Card (NIC)

- Physical layer processing
- Link layer processing
- Direct Memory Access (DMA) for copying data.
- Mechanism to trigger interrupts.
Modern NICs do much more than this
NIC Features: Protocol Offload

• TCP Segmentation offload
  • Split a large outgoing packet into MTU-sized packets and assign appropriate headers.

• Checksum Offload
  • TCP / UDP / IPv4 checksum computation.

• Large Receive Offload
  • Combine multiple MTU-sized packets for the same connection into a single large packet.
NIC Features: Packet Steering

• Receive Side Scaling
  • Load balance incoming packets across different queues.
  • Hash of packet header fields mapped to queue index.
  • Can pick which queue corresponds to which index.

• Flow Director
  • Maintain explicit mapping between packet header fields and queue.
  • Other actions including dropping and incrementing counters.
NIC Features: Virtualization

Diagram showing a Virtual Machine Manager (VMM) managing three virtual machines (VM) with guest operating systems and applications.
NIC Features: Virtualization
NIC Features: Virtualization
NIC Features: Virtualization

• SR-IOV (Single Root I/O Virtualization)
  
  • Provides the hardware abstraction of a ‘virtual function’ (VF).
  
  • Multiple ‘virtual functions’ mapped to a single physical function.
  
  • VMM maps a virtual function space to a specific VM.
NIC Features: Virtualization

• SR-IOV (Single Root I/O Virtualization)
  • Share a single physical port across multiple VMs.

• VMDq (Virtual Machine Device Queues)
  • Sort packets across VM specific queues based on MAC address and VLAN tags.

• Round-robin across VM queues.
NIC Features: Tunneling

- Examples:
  - VXLAN:
    - Offload encapsulation/decapsulation.
    - Ability to parse tunneled information.
  - NVGRE:
Limitations

• Lack of flexibility and fine-grained control.
  • E.g. TSO offload can be useless without VXLAN support.
  • Even minor fixes can take years.

• Resource constraints.
  • Limited memory (packet buffers, flow table size, etc).
  • E.g. Flow Director allows only 8K flow entries.
SoftNIC: A Software NIC to Augment Hardware

Sangjin Han, Keon Jang, Aurojit Panda, Shoumik Palkar, Dongsu Han, Sylvia Ratnasamy
SoftNIC Design Goals

• Programmability and extensibility
• Application performance isolation
• Backwards Compatibility
Architecture
Packet Processing Example
Resource Scheduling

• Allocate both processor and bandwidth resources.

C1, C3: high priority, 1 Gbps
C2, C4: low priority, no limit
Per VM: 5 Gbps limit
Implementation

• Over DPDK.

• Dedicate a small number of cores to SoftNIC.
  • Multi-core scaling achieved by associating each SoftNIC core with different set of queues.
  • Requires peers to ensure packets from same flow go to the same queue.

• Polling to check for packets from vport and pport.

• Batching to amortize software processing overheads.
Evaluation

The graph shows the roundtrip latency (us) for different scenarios:

- **T1**: Hosts (raw packet)
- **T2**: Hosts (kernel UDP)
- **T3**: VM-to-VM (inter-machine)
- **T4**: VM-to-VM (intra-machine)

The y-axis represents the roundtrip latency in microseconds (us), ranging from 0 to 100. The x-axis lists the different scenarios. The bars indicate the performance of DPDK, SoftNIC, ixgbe, SR-IOV, and SoftNIC, with error bars showing variability.
Case Studies

• If NICS do not understand tunneling format, cannot support TSO for “inner” TCP frames.

• SoftNIC can be used to augment the TSO/LRO feature in these cases.
Case Studies

• NIC supports a limited number of “rate-limiters” – few hundreds.
  • There may be thousands of flows.

• SoftNIC can be used to implement a scalable rate limiter.
Case Studies

• Flow Director directs packets with specific header fields to specific queues.
  • Can only support 8K entries.

• SoftNIC can support almost unlimited flow entries using system memory.
Case Studies

- Scaling legacy applications: send packets to different cores based on hash of packet header fields.

- RSS (NIC feature) is too limiting.

- SoftNIC can be used to provide such scaling.
Your Opinions

• Pros:
  • End-to-end latency and throughput is very close to bare metal.
  • Scheduling framework to support a wide range of policies.
  • Better cache utilization and low context-switching overhead by dedicating a small number of cores.
  • Case studies to show how SoftNIC can implement a variety of NIC features.
  • Backwards-compatible.
  • Flexible
Your Opinions

• **Cons:**
  • What happens as network speed increases further?
  • High CPU usage due to constant polling.
  • Small latency overhead.
  • Users need to define modules and workflow.
  • Comparison with FPGA-based or SoC-based NICs?
Your Opinions

• Ideas:
  • Telemetry in SoftNIC.
  • Use SoftNIC to provide SLOs to VMs/Apps in multi-tenant datacenters.
  • More features/usecases based on hardware-software codesign.
  • Congestion control for pipeline / scheduling.
  • Explore control plane interaction.
  • Comparison with programmable hardware NIC.