Implementation and Analysis of Large Receive Offload in a Virtualized System

Takayuki Hatori and Hitoshi Oi*
Computer Architecture and Operating Systems Group,
The University of Aizu
(*presenter)

VPACT08, Austin, TX
Apr 20, 2008

Outline

- System level virtualization
- Large Receive Offload
- Xen internal network architecture
- Large Receive Offload Implementation
- Experimental Results & Analysis
- Conclusions

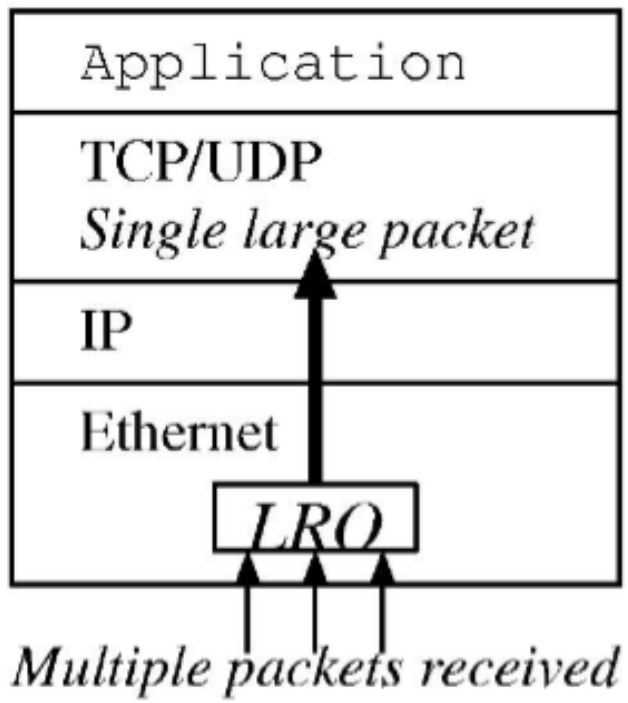
System Level Virtualization

- Multiple independent “machines” on top of single hardware platform
- Utilization and consolidation of hardware resources
- Isolation and protection against software malfunction and attack to VMs.
- These advantages come with overhead: especially in I/O components

Objectives

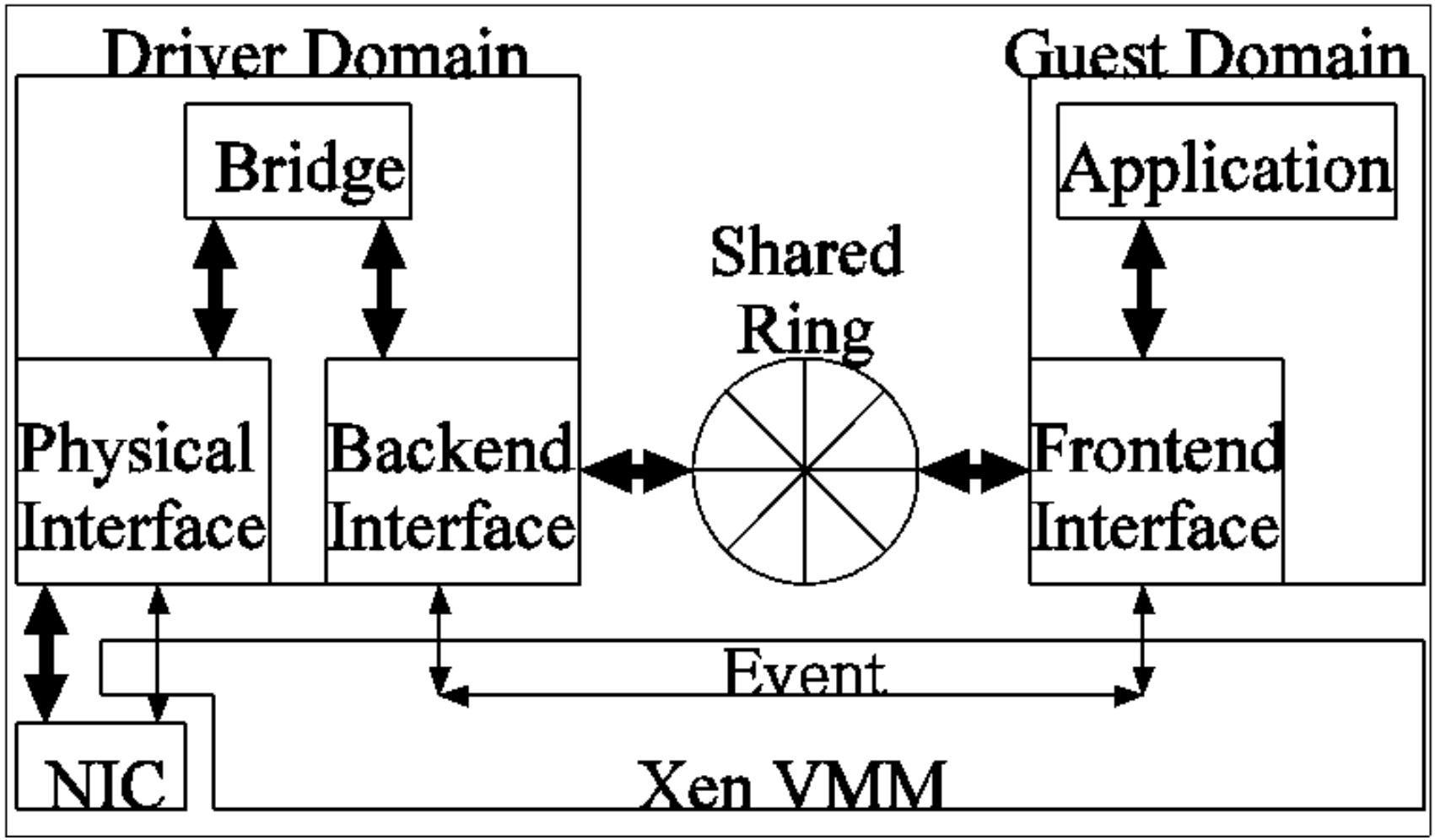
- Porting LRO to a Xen virtualized system and see how it improves network performance.
- Modifications to internal network architecture (interface between domains).
- Experiments and evaluations
  - Sender-receiver programs transfer 10GB data
  - Measured throughput, CPU utilization

Large Receive Offload (LRO)

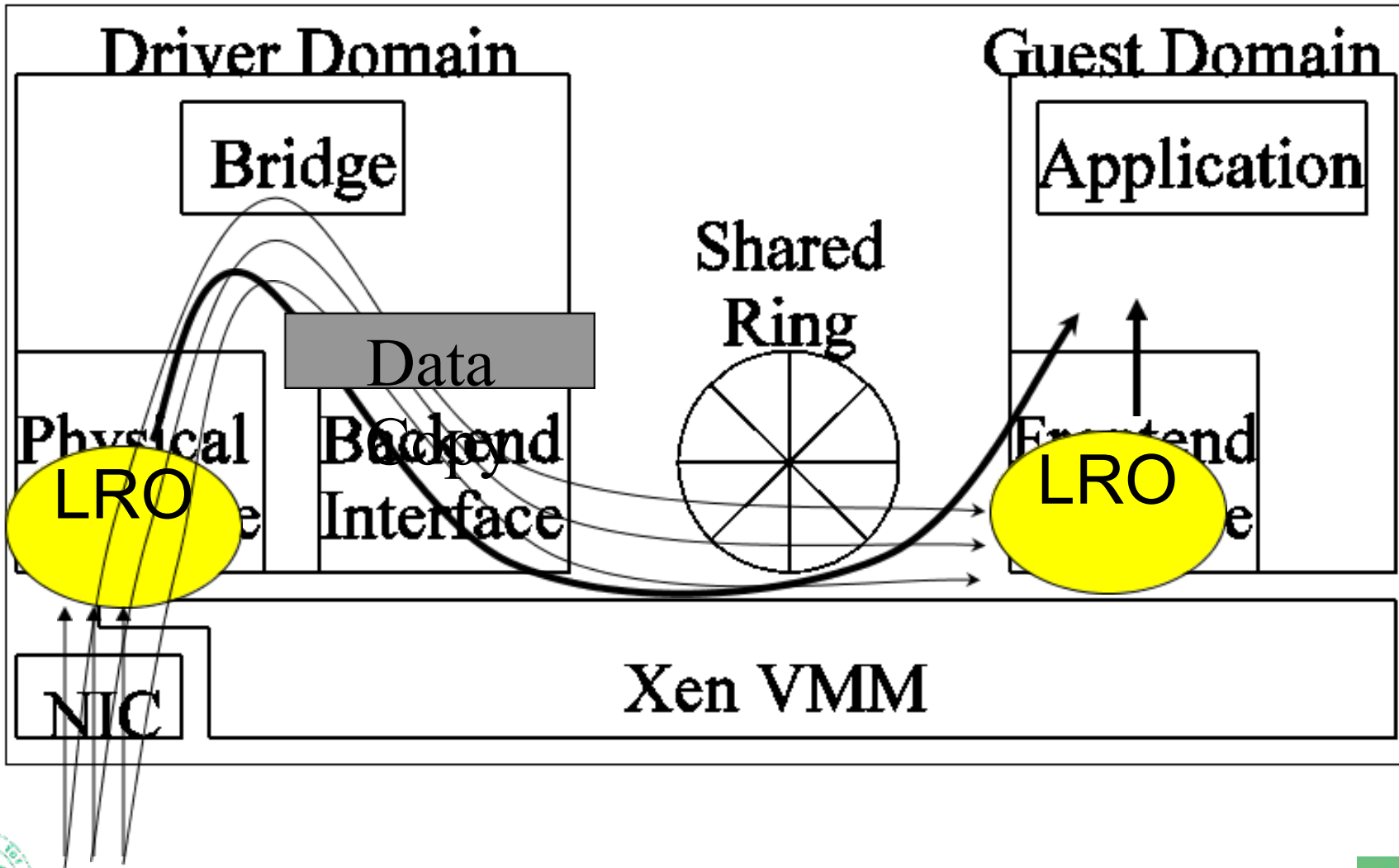


- Receive multiple packets in a single receive operation
- Aggregate packets into groups and pass them to the upper layers
- Reduce overhead in packet-related data structures (e.g. skb).

Xen Internal Network Architecture



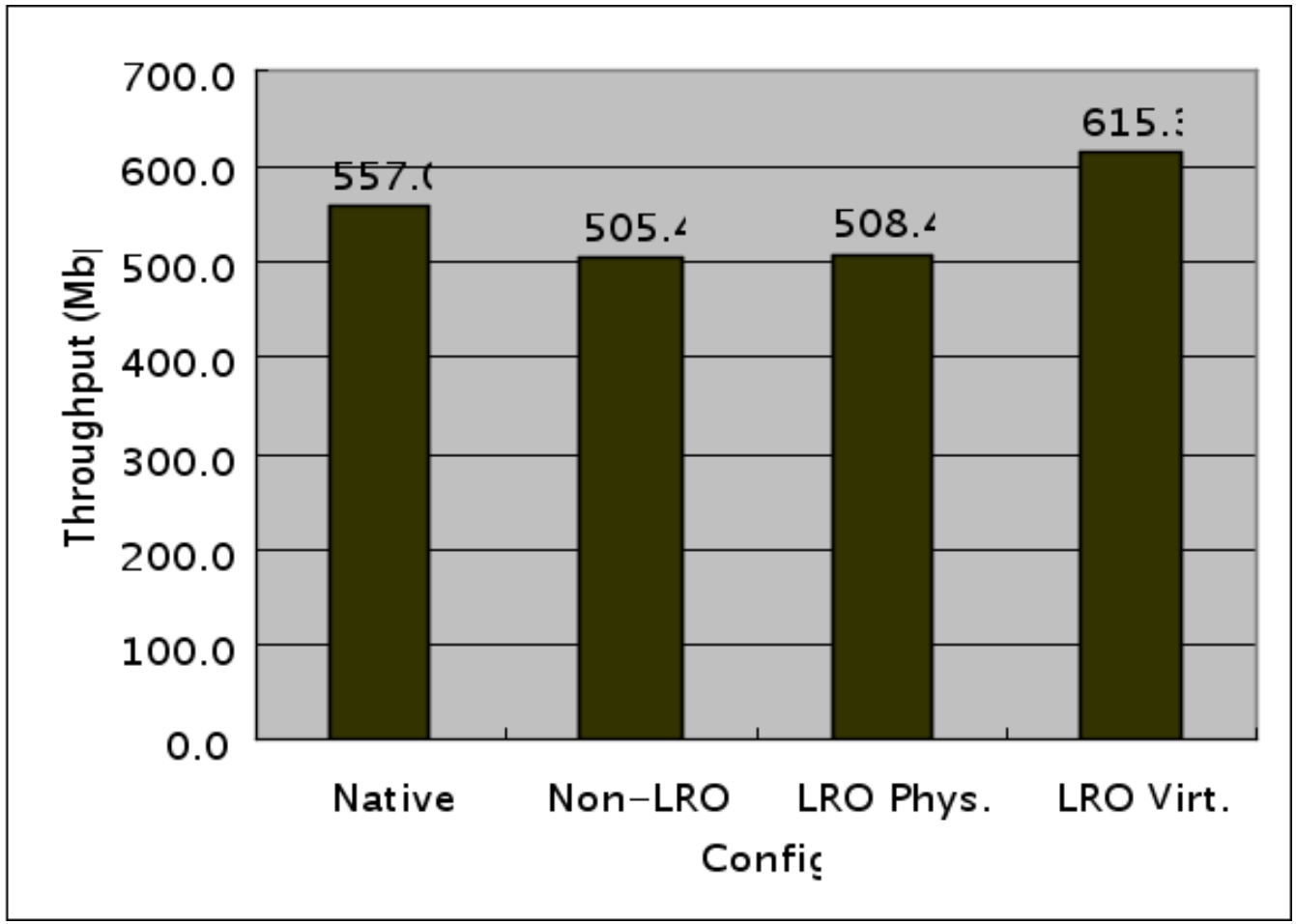
LRO Implementation



Experimental Setup

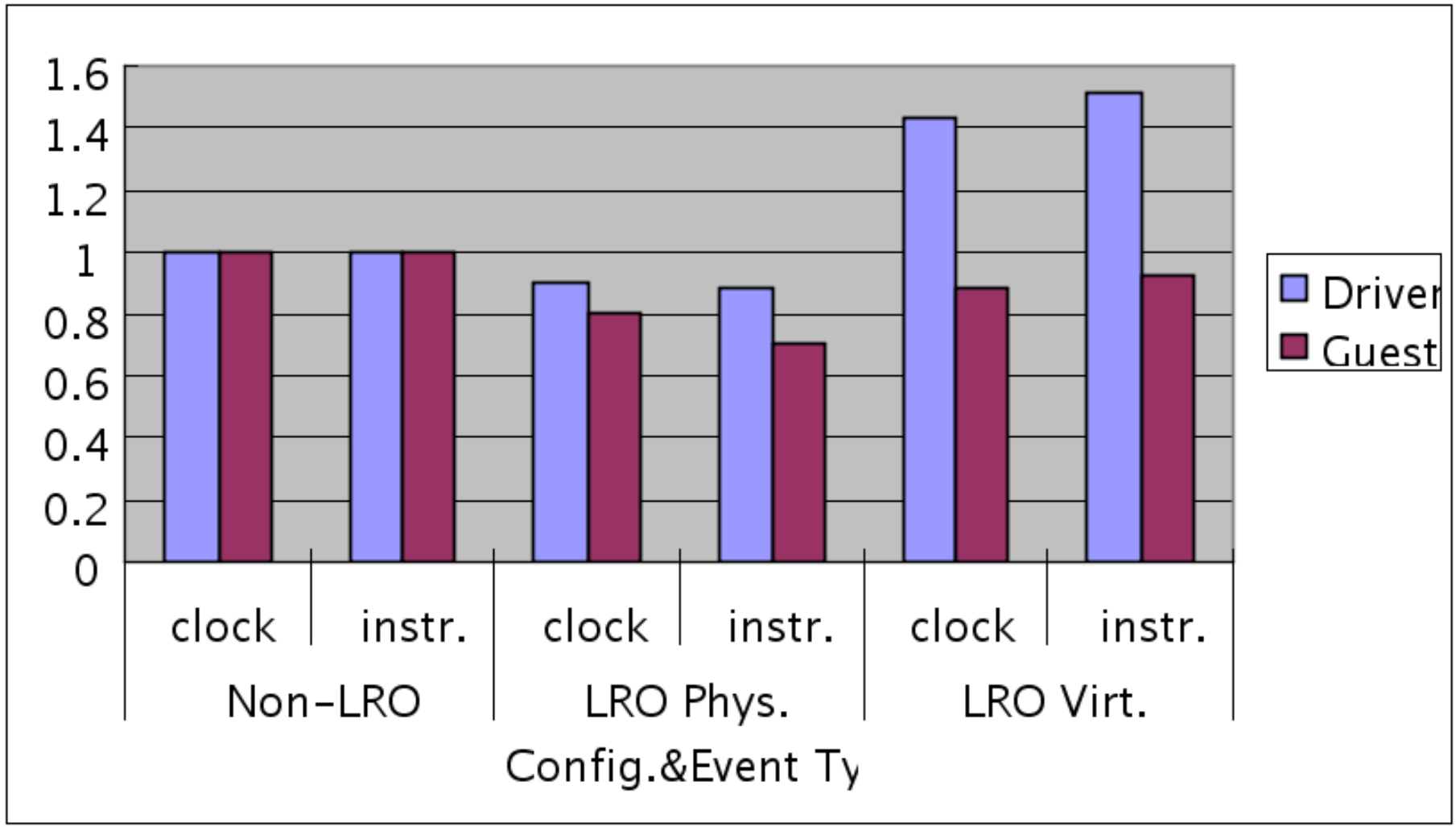
- Operating System: CentOS (Linux 2.6.18)
- Xen: 3.1.0
- Receiver: Xeon 1.86GHz, 2GB Mem
- Sender: AMD Athlon 64x2 2GHz, 2GB Mem
- NIC: On-board 1Gbps
- Workload: Simple sender-receiver programs that transfer 10GB data with MTU=1500B.

Throughput & LRO Rate Comparison

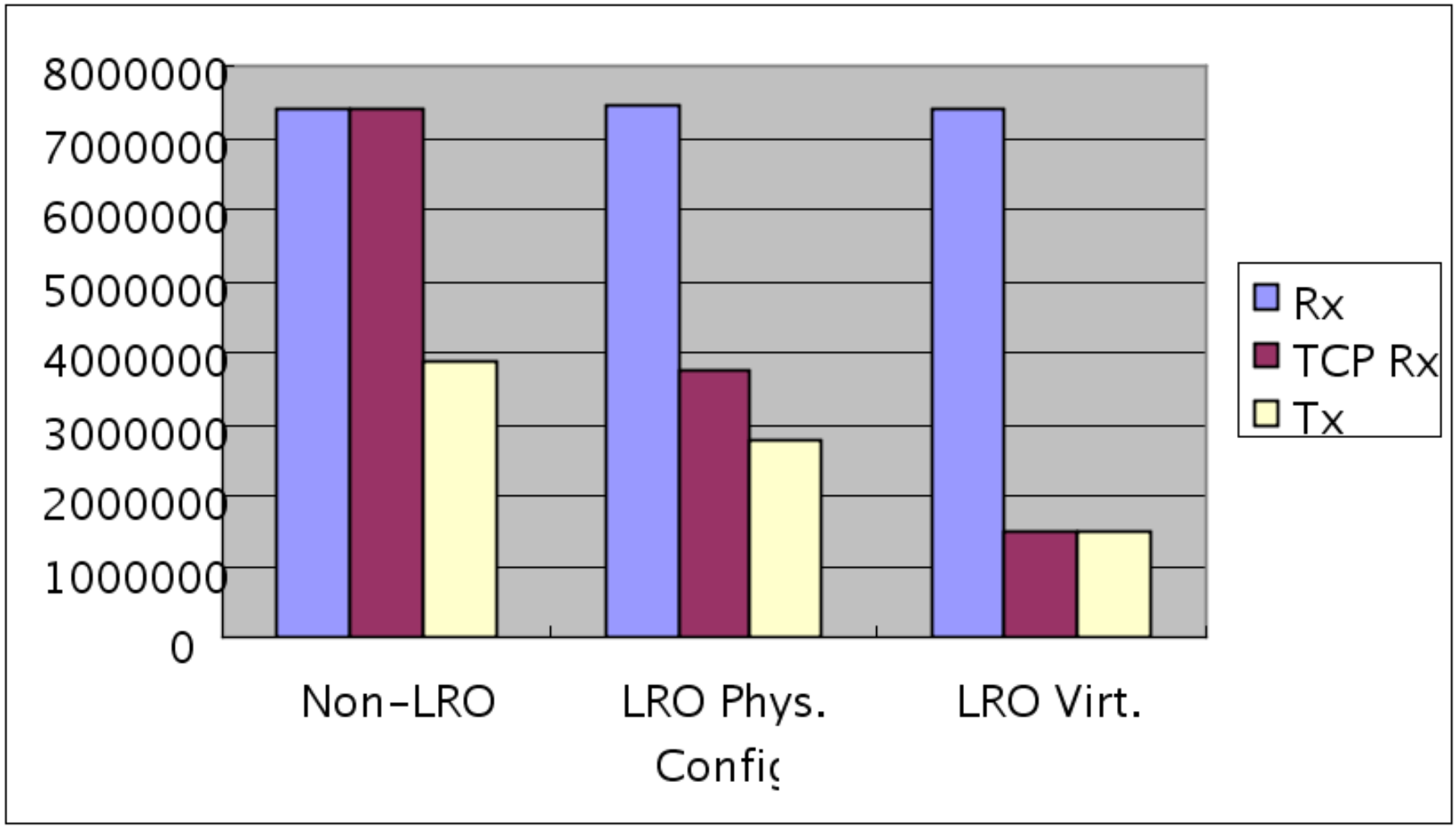


Config.	LRO rate
Native	1.00
Non-LRO	1.00
LRO Phys	1.98
LRO Virt.	4.91

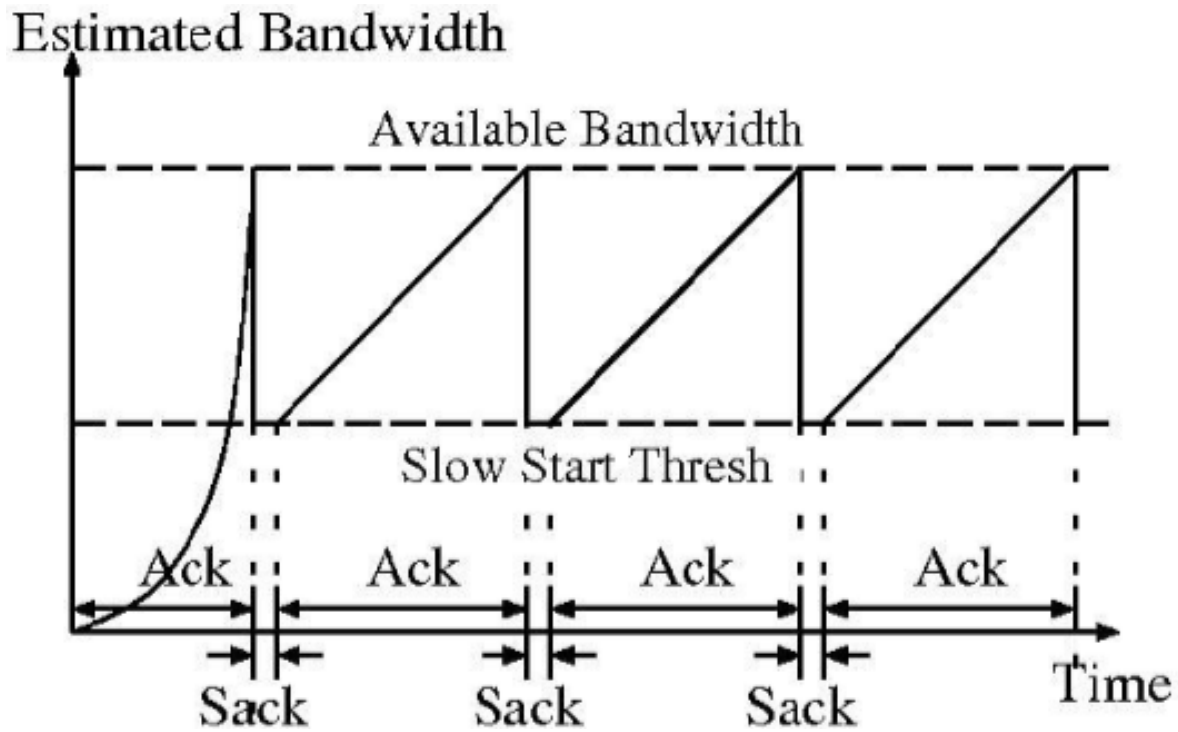
Clock Cycles & Instruction Count



Network Traffic at Receiver

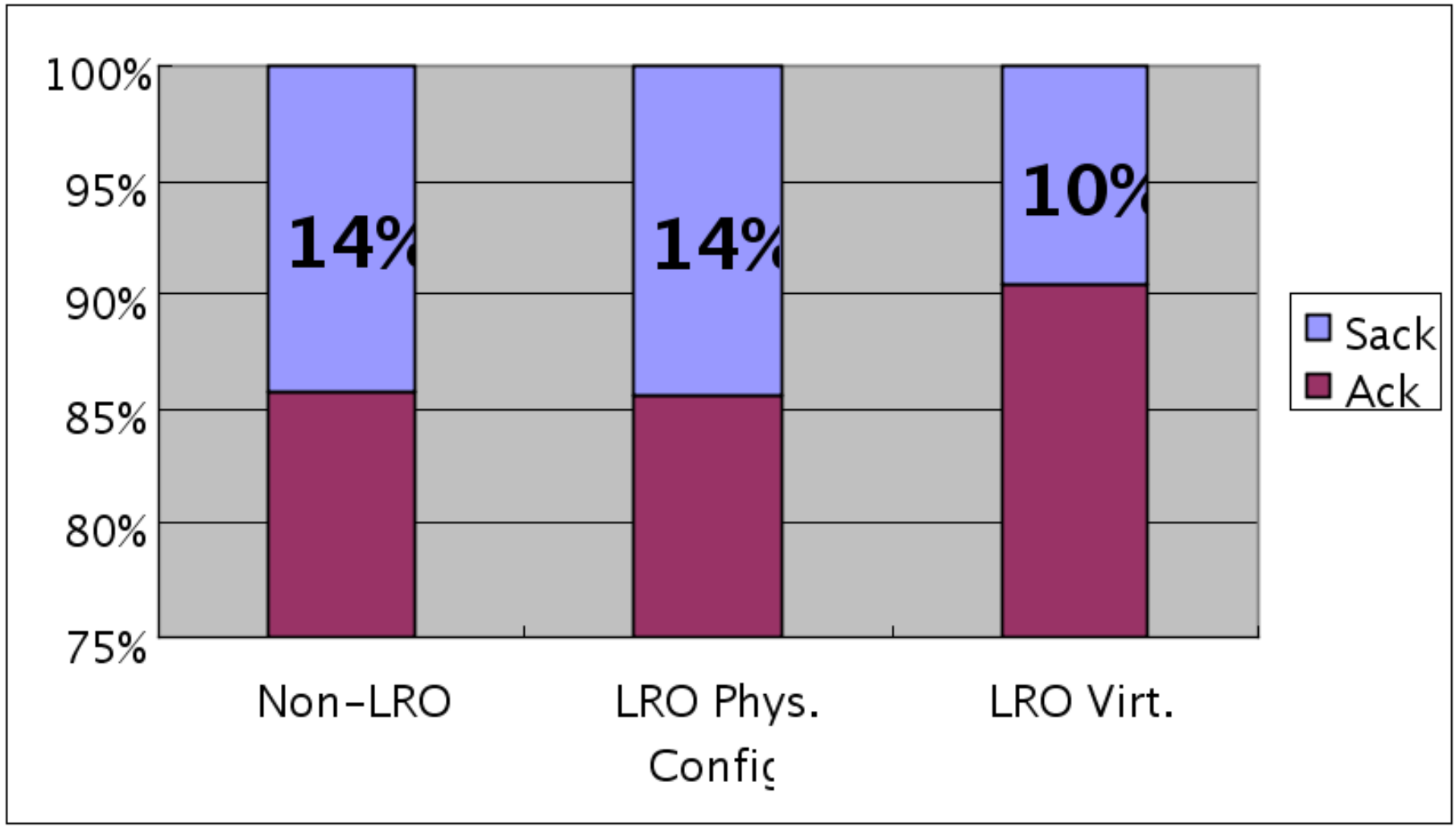


Bandwidth Estimation at Sender



Ack received at Sender increases
estimated bandwidth Till Sack received

Breakdown of Acknowledgment



Analysis of Performance Improvement

- LRO reduced acknowledgment
- Ack. increased
  - By unsuccessful delayed acknowledgment
- Selective acknowledgment rate dropped
  - Sacks were not aggregated by delayed acknowledgment
- High estimated bandwidth in TCP layer
  - Uses selective acknowledgment rate for estimation
- High throughput achieved

Conclusions

- LRO implementation in Xen and experimental results presented.
- LRO in the physical interface improved the CPU utilization across the system
- LRO in the virtual interface achieved high throughput, which is the result of LRO and delayed acknowledgment combination.
- Further Optimization
  - LRO aware Network Bridge
  - Option to disable delayed acknowledgment

Thank you for your attention

Any questions ?