

Deferred Segmentation For Wire-Speed Transmission of Large TCP Frames over Standard GbE Networks

Bilic Hrvoje (Billy)
Igor Chirashnya

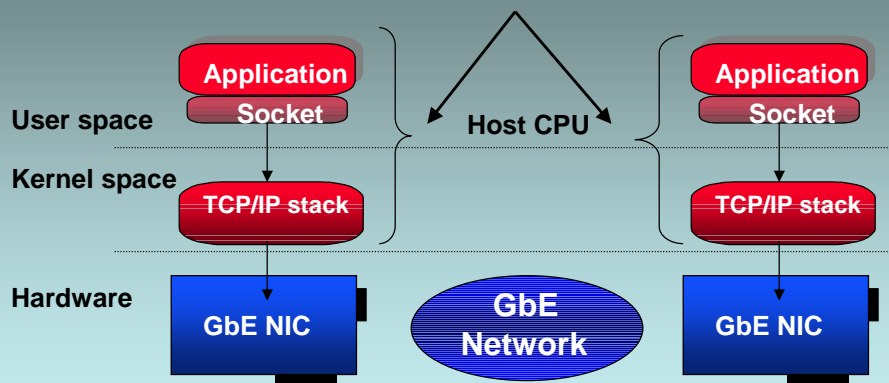
Yitzhak Birk
Zorik Machulsky

Technion - Israel Institute of technology
Department of Electrical Engineering



Motivation

- CPUs can not keep pace with network speed explosion
- TCP/IP stack processing by host CPU is performance bottleneck for TCP/IP traffic over Gigabit networks



More Intelligence is moved into networks !!!

Goals

- Solve the real problem - TCP/IP stack bottleneck for large TCP transfers over standard Gigabit Ethernet networks
- Reach the wire speed for large TCP/IP transfers
- Maximize host P/E ratio ($P/E = BW/CPU \text{ load}$)
- All this BUT
 - W/O modifications of existing (legacy) applications
 - W/O modifications of existing OS TCP/IP protocol stacks
 - With minimal Gigabit Ethernet NIC development efforts

End system TCP/IP overhead analysis

- Per connection
 - Decrease the number of packets exchanged on TCP connection establishment/termination
- Per byte
 - Zero-copy techniques
 - Checksum offload to HW
- Per packet overhead
 - Interrupt Coalescing
 - Large frames to carry data in TCP/IP stack

Previous approaches to TCP/IP bottleneck problem

- **TCP/IP splitting approach - split TCP/IP stack between applications, OS and NIC**
 - Hard to implement - requires modification NIC, OS and applications
- **TCP/IP offload - TCP/IP stack in HW**
 - Expensive HW and very hard to implement

No industry acceptance, mostly research works !!!

Our TCP/IP Pipelining approach

- **Offload to NIC only small subset of TCP/IP stack functionality and reach 1GbE wire speed**
 - BUT w/o any OS or applications modifications
 - **Pros**
 - No changes of existing OSs and applications
 - Small NIC implementation efforts & cheap HW
 - Ideal low cost solution for 1GbE networks
 - **Cons**
 - Does not impact small TCP/IP transfers
 - Not competitive with existing transport in H/W implementations FC and IB (higher CPU load)
 - Does not suit 10Gb/s transfer rates

Synergy with current industrial needs for 1GbE!!!

Previous Work on TCP/IP Acceleration

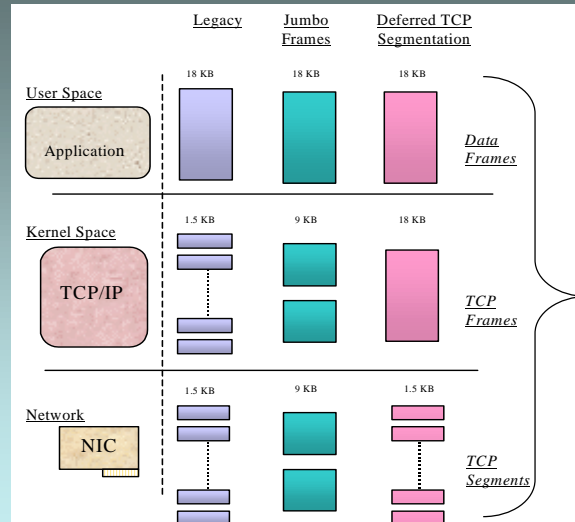
- **Duke university reasearch**
 - TCP/IP traffic over Myrinet Gigabit Networks
 - Performance increases with large MTUs(*)
 - MTU (1.5KB->32KB)
 - [BW](#)
 - [CPU Load highly decrease](#)
 - P/E ratio at least doubles on both Tx and Rx side
 - (*) All results w/o zero-copy and with checksum in HW
 - Myrinet Network has **UNLIMITED MTU** size (**up to 32KB**)
- **Alteon Inc. proposed Jumbo Frames (MTU = 9KB) as a solution for Gigabit Ethernet networks**

“**BUT** Standard GbEthernet networks MTU = 1.5KB”

Main Idea

- **Gain the perfomance of large MTU networks for GbE**
- **Cheat TCP/IP stack, by causing large TCP frames transfers to NIC => Emulate Myrinet network**
 - [Emulate network with large MTU size](#)
 - [MSS spoofing - Intecept SYN packets and modify MSS value](#)
- **Break the large frames down to Ethernet packets with standard MTU (1.5KB) size**
 - [Deferred TCP segmentation - ACK coalescing](#)
 - [Targeted Systems](#)

Advantages over Jumbo Frames



- Inter-operates with standard GbE networks
- Performance - Frames > 9K (up to 64KB)
- Smaller frames are more network friendly

Demonstration of Completeness

- **Ethernet Emulation Environment**
 - Two IBM Evaluation boards inter-connected by Ethernet network
 - IBM Evaluation board
 - Integrated PPC405 running our TCP segm - ACK coalescing firmware
 - Connected over PCI to Pentium 450MHz host running Linux R.Hat 6.0
 - **Demonstrate TCP segmentation - ACK coalescing mechanism with unmodified TCP/IP stacks**
- **Gigabit Ethernet Environment**
 - Two Alteon GbE AceNICs interconnected by GbE network
 - Alteon AceNIC
 - Device Driver performs TCP segm -ACK coalescing mechanism
 - Connected over PCI to Pentium 450MHz host running Linux R.Hat 6.0
 - **Demonstrate completeness while inter-operates with an unmodified off-the-shelf receiver**
- “Netperf” used for TCP traffic generation; “Tcpdump” for monitoring

Performance Estimations

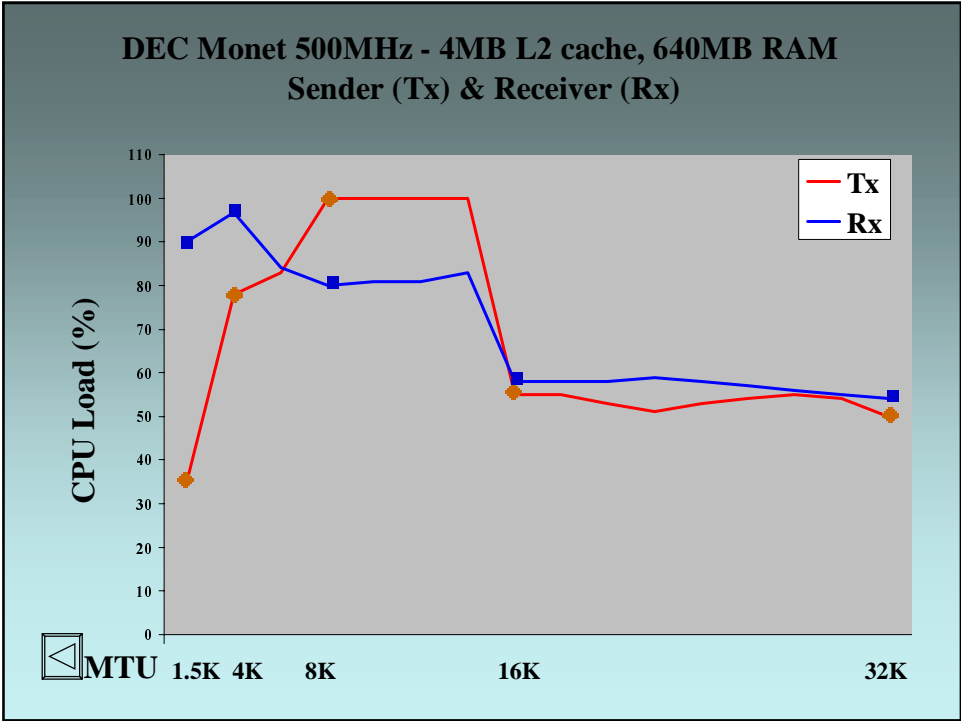
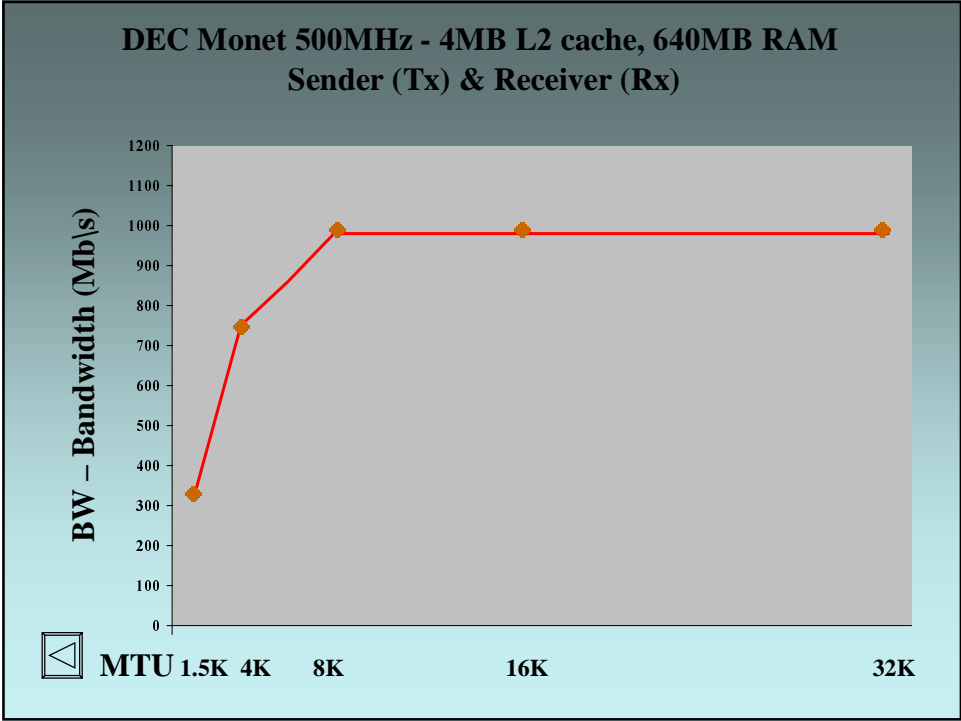
- **TCP conn. receiver side is assumed not to be bottleneck**
 - Multiple clients compose connections Receiver side
- **Estimated performance on the Sender side**
 - Host CPU process large TCP frames (>8KB) at wire speed - Duke
 - GbE NIC performs its standard functions at wire speed
 - Additional processing by NIC related to TCP seg - ACK coalescing
 - Tx path
 - TCP segmentation
 - » Original TCP and IP headers used as a templates
 - » Start processing already when headers are in Tx FIFO
 - » Assembly TCP seg. code running on embedded upprocessor < 200 cycles (100MHz clock => 2us) << 12 us (1.5K\1Gb/s)
 - Rx path
 - Calculate checksum on-the-fly while receiving data from network
 - ACK coalescing - overall ~2.5us
 - » Flow Lookup - < 2us (100MHz; 10K entries)
 - » ACK coal. Assembly code for embedded processor ~0.5us
 - » ACK rate in average ~ TCP segments trans. rate ~ 12.us

Summary

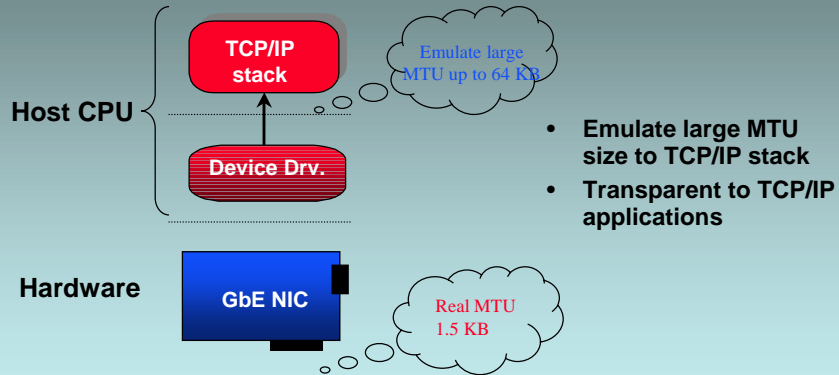
- **CPUs can not keep pace with network speed explosion**
 - TCP/IP intelligence should be moved into the network
- **Smart simple TCP/IP acceleration by ASIC (NIC) solves TCP/IP stack bottleneck for 1GbE networks**
- **Multiple Gigabit (10G) networks will require different solutions**
 - FC and IB - Implement transport protocols in HW
 - 10GbE -TCP/IP networks will require different solution to compete



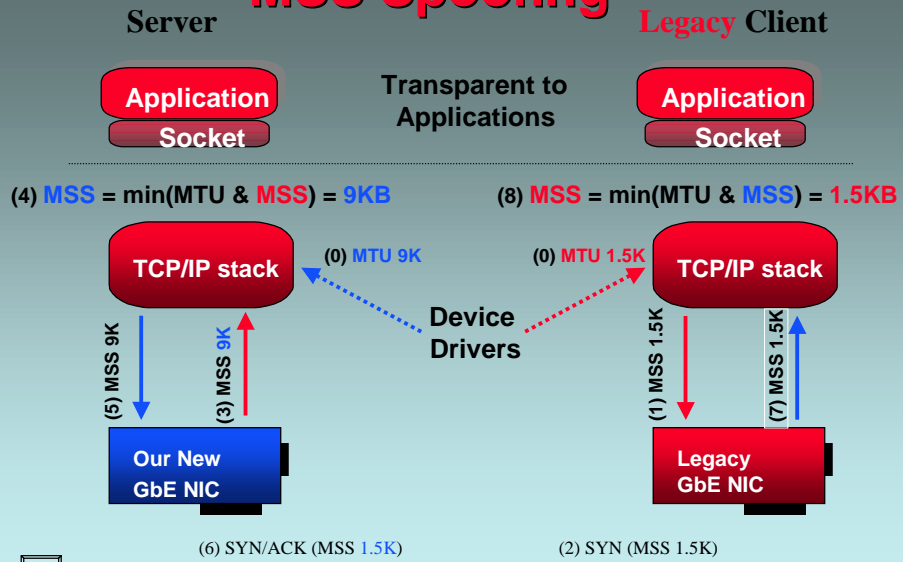
TCP/IP stack in HW !!!



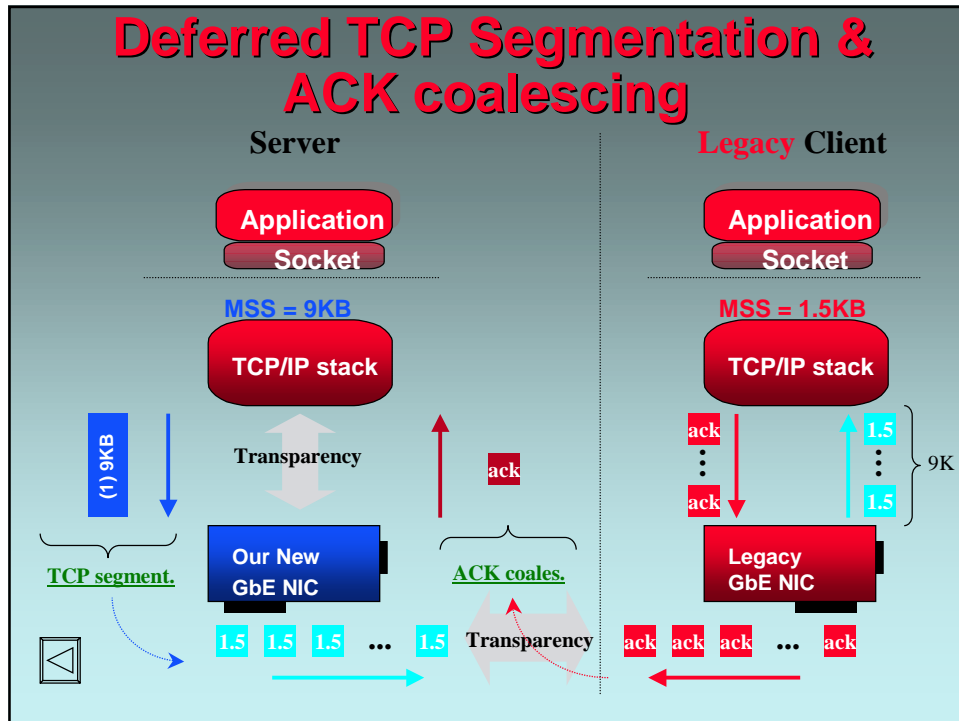
MTU Emulation



MSS Spoofing



Deferred TCP Segmentation & ACK coalescing



TCP segmentation - ACK coalescing Targeted Systems

- **Systems with Assymetric TCP traffic**
 - Accelerates mainly Tx path (outbound traffic)
 - Servers serving many clients with large transmit outbound traffic
 - Enhanced ASIC/NIC required only on Server side
 - Completely Transparent to clients
 - Re-segmentation can be implemented but rise cost of NIC
- **NAS boxes, File, Video & Application Servers**
 - Enhanced Server GbE NIC
- **Data Cache Systems**
 - NIC or ASIC depend on internal architecture
- **ISCSI HBA or RAID controller ASIC**
- **Smart ISCSI/IB/FC/SCSI boxes**
 - NIC or ASIC for Storage Routers/Gateways, Proxies etc.