Data Center Networking and Protocol Processing

CSE 291E / ECE 260C
Spring 2002

University of California
San Diego
Overview

• Applications for TCP Offload
• The TCP Offload problem
Why we have a need for TCP offload processing?

- Storage networking
  - iSCSI
- Clustering
  - VI – Virtual Interface
  - RDMA – Remote Direct Memory Access
- These protocols run over TCP/IP/GE
  - TCP/IP/GE possibly used as unified fabric for LAN, SAN, and clustering
The Road to Storage Networking

- Storage growing exponentially
  - More new electronic data generated in next few years than total in human history
  - IDC forecast (nearly 70% network attached by 2004)
Problems with Local Storage

- What do you do when you run out?
- What to do with all that unused (wasted) disk space on local drives
  - Today, 160GB ATA drives = $200, but you may only need 10GB
  - 320GB ATA drives by end 2002
Fibre Channel

- Developed to enable storage area networking (SAN)
  - Move disks to the network
  - Can aggregate them, virtualize them, and dynamically re-allocate them
  - Provides latency and throughput comparable to local SCSI bus
  - Enables centralized storage management and backups
Why TCP/IP/Ethernet not used for SAN?

• Only 10/100Mbps
  • **Too slow**: need gigabit speeds for storage
• TCP/IP designed as “internetworking” protocol across different physical media
  • e.g. multi-hop across WAN could be 56k modem – metro SONET – core ATM network – metro Ethernet – wireless GPRS
  • Need end-to-end congestion control, need acknowledgement ...
• TCP/IP in OS stacks involve 70K instructions and many memory copies managed by OS
  • **Latency too long** as a result
Fibre Channel

- 1 Gbps, 1st fiber then copper
- Assumes dedicated network, so assume
  - Largely error-free transmission
    - Provide link layer checksums and retransmission
  - Limited congestion problems
    - Provide link layer credit-based flow control
  - No out-of-order packet arrival
- Fibre channel as implemented is for “within data center” storage networking
- No real fibre channel specs defined over WAN
  - Currently requires tunneling over TCP/IP (FCIP, iFCP)
Then Came Gigabit Ethernet

• Borrowed gigabit Fibre Channel PHY and link layer concepts
  • Similar link layer checksum and flow control
  • Looks like 10G Ethernet PHY and 10G Fibre Channel PHY will be same as well
• SCSI over UDP/IP/GE should have comparable performance and latency as SCSI over FC
• Proprietary implementations from 3ware, DotHill, LeftHand Networks ...
• IETF won’t standardized on UDP b/c it wants the standard to work over WAN
iSCSI

• IETF standardized on iSCSI, which is SCSI over TCP/IP
• Works “within data center” and **over WAN**
• iSCSI proponents argue gigabit Ethernet infrastructure much less expensive than FC infrastructure b/c of economy of scale
Need for TCP Offload Engine

- High-latency and poor CPU utilization
  - 1MHz CPU per 1Mb/s of storage traffic = 1GHz CPU fully utilized to handle 1Gb/s of storage traffic
  - 10Gb/s traffic simply not feasible
- With TCP offload engine, current tests suggest comparable latency and performance as Fibre Channel
- Intel gigabit iSCSI adapter uses 1GHz XScale to implement TCP in microcode
- 10G TCP offload engine remains open problem
The Road to Clustering

• A lot of work in scientific computing community to use a cluster of servers to execute scientific applications
  • e.g. message-passing based primitives like PVM and MPI
• Needed a low-latency high-speed interconnect
• Proprietary IPC (interprocess communication) interconnects
  • Myrinet
  • Giganet (now part of Emulex)
  • ServerNet
• Commercial API’s
  • VI
  • RDMA (e.g. iWARP)
Why TCP/IP/Ethernet not used for clustering?

- Same reason why TCP/IP/Ethernet was not used for storage networking
  - Only 100Mbps
  - TCP too much latency
  - TCP too much CPU overhead
Why TCP/IP and Gigabit Ethernet is being considered for IPC?

- No difference in PHY
- TCP offload engines enable OS bypass
- Applications can use VI, SDP, RDMA APIs (various IETF standardization efforts)
- TCP offload engines + gigabit Ethernet NICs handle protocol handshakes
The Road to a Unified Fabric

• **Infiniband**
  • Developed precisely to unify storage networking and clustering (IPC) via single fabric

• **Problem**
  • Infiniband, TCP/IP/Ethernet, Fibre Channel all have their own protocol stacks and required IT skills
  • World not likely to adopt Infiniband across the WAN
The Road to a Unified Fabric

- Fibre Channel can be a candidate for IPC and single fabric unification, but the FC camp didn’t want to fight the Infiniband vs. FC battle for Server I/O
- FC camp simply wanted to continue to secure a stronger foot-hold in SAN
- TCP/IP/Ethernet with full protocol offload is a candidate for single fabric unification
  - The jury is still out
- Or we’ll simply continue to have multiple fabrics
Overview

• Applications for TCP Offload
• The TCP Offload problem
Traditional TCP/IP Stack

User-level Transport API

Application

Transport Library

Kernel Mode

Transport Driver

Kernel Applications

Kernel-level Transport API

TCP

IP (IPSEC)

MAC Driver

Host Software

Network Interface Hardware

MAC

PHY

User Mode

70K Instructions
Host TCP/IP Overhead

- Operating system related
  - Mode transitions, context switches, interrupt processing, synchronization

- Protocol specific
  - Header processing, state maintenance, fragmentation, reassembly, reliability, flow control, congestion control

- Data touching & manipulation
  - Data integrity check, encryption/decryption
Host TCP/IP Overhead (cont.)

- Data placement & movement
  - Memory copies & DMA transfers
- Buffer management
- Other
  - Data structure manipulations, locks, timers
TCP/IP Offload (TOE)

- Offload full or partial TCP/IP processing
- Bypass host based TCP/IP processing
- Lower host CPU utilization
- Near wire-speed TCP throughput
Copy on Receive

- Generic TOE NICs do not perform direct data placement
- Generic TOE
  - Keeps data received from the network into temp buffers
  - And later on copies it into application buffers
- Copies can be expensive at speeds 10Gbps+
  - Requires expensive high-speed memories on TOE
  - NIC buffers (not the app. Buffers) dictate TCP window size
Direct Data Placement (DDP) and Remote Direct Memory Access (RDMA)

• To avoid copy on receive & reduce NIC buffering
• DDP requires more complexity & intelligence
  • Parse ULP (upper layer protocol) headers & separate ULP payloads
  • Demux multiple incoming streams
  • Knowledge of application buffers
• Remote Direct Memory Access (RDMA)
  • A method for directly accessing memory on a remote system without interrupting host CPUs
• But need framing over TCP/IP
TOE with Framing, DDP & RDMA

User-level Transport API

User Mode

Transport Library

Kernel Mode

Transport Driver

Kernel Applications

Transport Stack with offload support

TOE Driver

Host Software

Network Interface Hardware

RDMA

DDP

TCP ULP Framing

TCP

IP (IPSEC)

MAC

PHY
Questions?