Datacenter Fabric Workshop

Sockets Direct Protocol (SDP)

Dror Goldenberg **Michael S. Tsirkin**

Mellanox Technologies Inc.

{gdror,mst} at mellanox.co.il

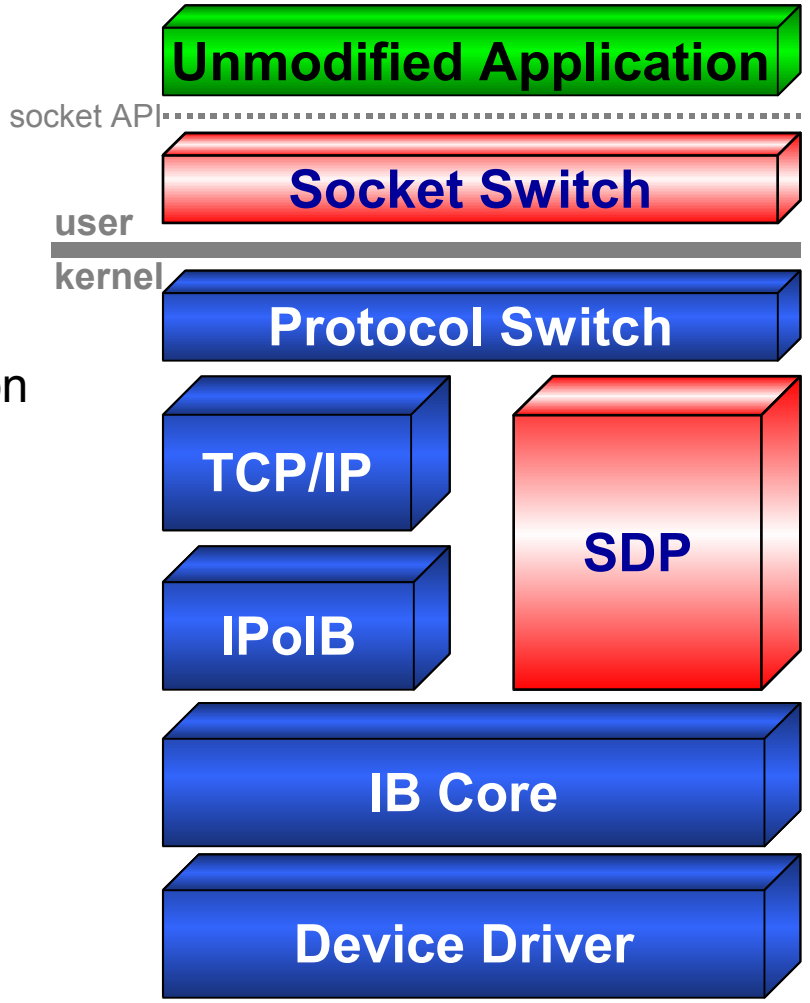
22 August, 2005

Agenda

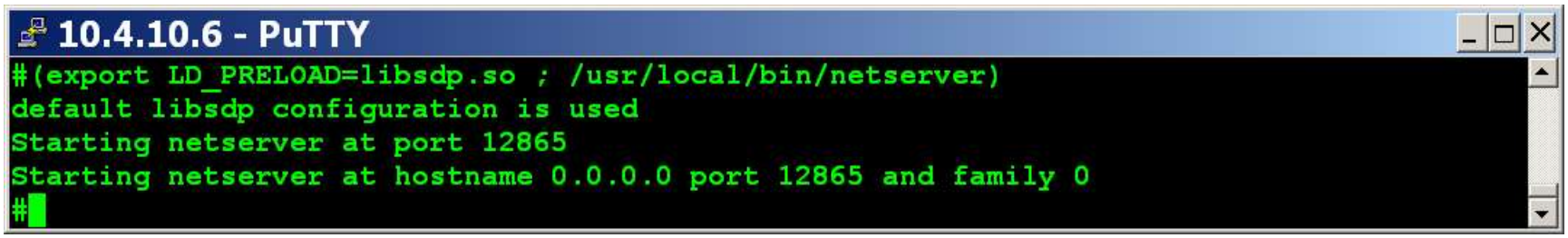
- SDP Protocol Overview
- SDP Stack Components
- Status and Plans

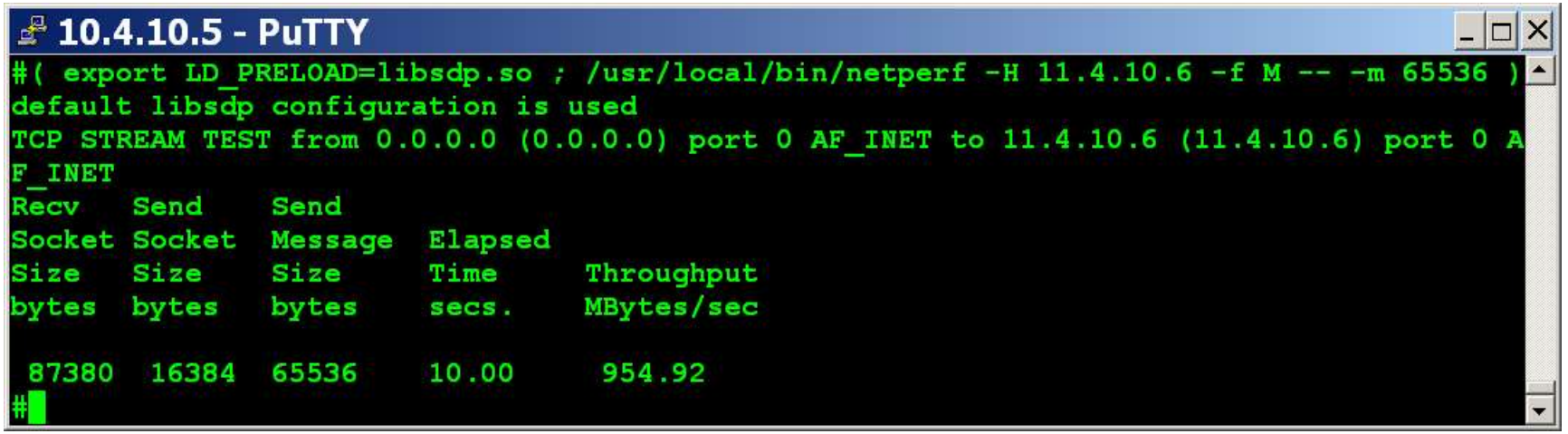
Sockets Direct Protocol (SDP) Overview

- Transparent to the application
- Maintains socket semantics
- Leverages InfiniBand capabilities
  - Transport Offload – Reliable Connection
  - Zero Copy – Using RDMA
- Standardized wire protocol



SDP is Easy





Or, explicitly use SDP socket:

socket(**AF_INET_SDP**, SOCK_STREAM, 0);

ULP Comparison

	IB Verbs	SDP	IPoIB
API	Low level	Socket (TCP only)	Socket
Latency	2.6us (polling) 11.8us (event)	15us	18.6us
Bandwidth	1411 MB/s	BCopy 960MB/s ZCopy 1387MB/s (AIO)	319MB/s
CPU Utilization	protocol dependent	BCopy 100% out of 400% ZCopy 76% out of 400% (AIO)	124% out of 400%
Kernel bypass			
Stack Overhead	Light	Medium	High
Memory Registration	Explicitly by application/ middleware	Heuristics by SDP	None
Application Adaptation	Porting/ Development Required	Supports Unmodified Application	

Socket Switch

- Socket switch transparently changes AF_INET to AF_INET_SDP

- Userland library – libsdp.so
  - Preloaded (LD_PRELOAD)
  - Intercepts socket control calls
  - Data-path is unaffected

- Configurable policy
  - SDP, TCP or both
  - Selection is based on
    - Destination address/port (client side)
    - Local port (passive side)
    - Application name

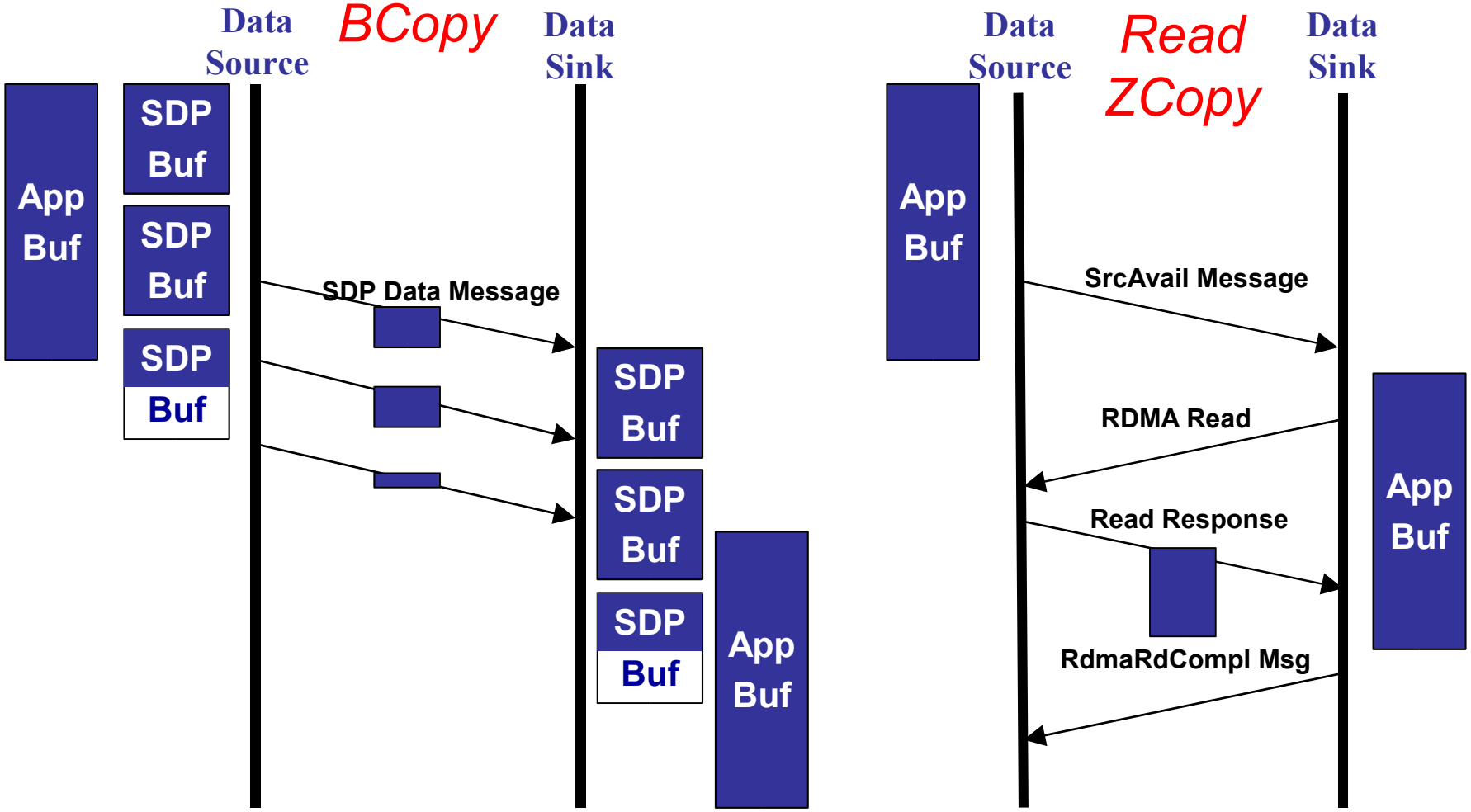
Addressing and Connection Management

- IPv4 supported
- IP based routing
- IPoIB ARP to locate GID
- SA query for PathRecord
- CM used to establish/teardown connection
- IBTA assigned service IDs for servers listening on “TCP” ports

API Supported

- Socket synchronous calls
  - connect(), send(), recv(), etc.
  - Out of band data
  - BCopy only supported on trunk
- Asynchronous I/O
  - Linux specific: io_submit(), io_getevents(), etc.
  - BCopy and ZCopy

Data Transfer Modes



Memory Management

- SDP Private buffers
  - Control and buffer-copy messages
  - Allocated: get_free_page()
  - Managed as a shared pool of resources
  - Mapped for DMA upon posting to HCA, unmapped upon completion
  - HCA access through DMA Memory Region (local R/W)

- RDMA buffers
  - Used for sending and receiving user buffers through zero-copy
  - Fast Memory Region (FMR) pool for HCA access
  - Pinning through get_user_pages() / put_page()

SDP Connection

- IB connection utilizes:
  - QP
  - CQ for send
  - CQ for receive
  - PD, DMA MR, FMR pool are shared at the SDP level
  - Has upper bounds on shared local buffers usage

Progress Since Feb 2005

- Stack maintainer
  - We thank Libor Michalek for the great work he did !
  - Michael S. Tsirkin and Tom Duffy

- Work done so far (svn log…)
  - Initial check-in 2/11/2005
  - Enabled AIO Zero-Copy using FMRs
  - Simplified connection management state transitions and reference counting
  - Coding style cleanups
  - Portability
  - Race conditions eliminated / Bugs Fixed
  - Memory pinning using get_user_pages() / put_page()

Status/Short Term Plans

- Prepare for kernel inclusion
  - Coding style is clean
  - Clean out remaining portability, code duplication issues
- Zero-Copy
  - Only AIO from user-space supported on trunk
  - Send/Recv being developed on a branch
- Verification
  - Automated testing
  - Stress testing
- Latency reduction

Longer Term Plans

- Socket options
  - SO_SNDBUF / SO_RCVBUF
  - Keepalive
- IPv6 addressing
- High availability
  - APM
  - Socket duplication
- Scalability
  - Posted receive slow start
  - SRQ
- Connection rate
- ES-API

Thank You !

Backup

Benchmarks

- UVerbs
  - Bandwidth: perftest/rdma_bw
  - Latency (polling): perftest/rdma_lat
  - Latency (event): ibv_rc_pingpong
- SDP
  - Bandwidth (BCopy): netperf (TCP_STREAM)
  - Bandwidth (ZCopy): ttcp.aio.x (20 outstanding requests)
  - Latency: netperf (TCP_RR)
- IPoIB
  - Bandwidth: netperf (TCP_STREAM)
  - Latency: netperf (TCP_RR)
- Latency measured for 1B transactions
- Bandwidth measured for 64KB transactions

Benchmarking Platform

- Hardware
  - Dell 2850
  - x86_64 Dual Xeon 3.2GHz, 1MB Cache, HT Enabled, 4GB RAM
  - InfiniHost III Ex PCI Express adapter
    - IB DDR, MemFree, Firmware 5.1.0
- Software
  - Linux 2.6.12
  - Latest OpenIB stack (svn version 3056)