Datacenter Fabric Workshop

Sockets Direct Protocol (SDP) for Windows - Motivation and Plans

Dror Goldenberg **Tzachi Dar**

Mellanox Technologies Inc.

{gdror, tzachid} at mellanox.co.il

22 August, 2005

Agenda

- SDP Protocol Overview
- SDP vs WSD
- SDP Stack Components
- Current Status

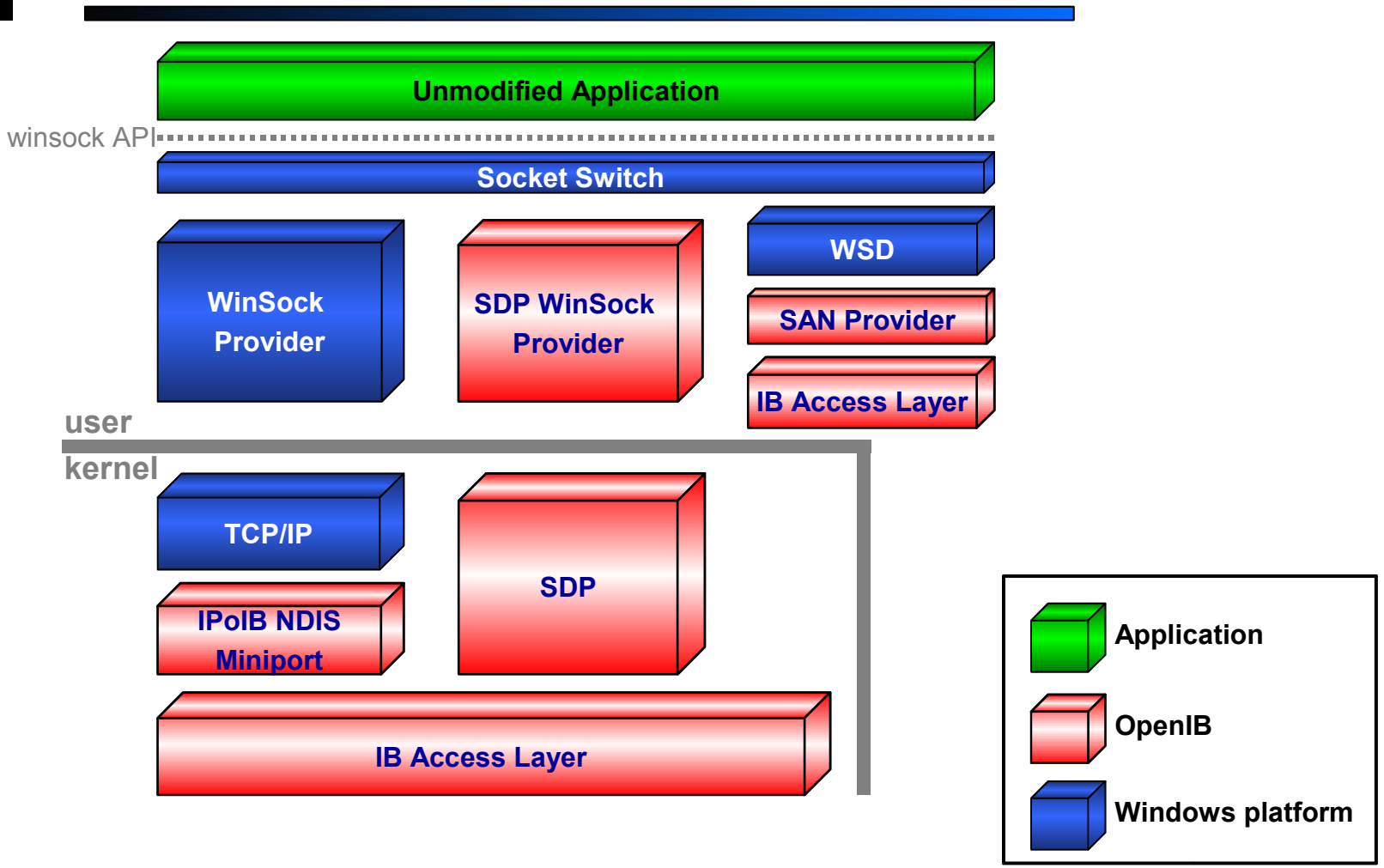
Key Message

With SDP, any traditional sockets-based applications can benefit from InfiniBand without any change.

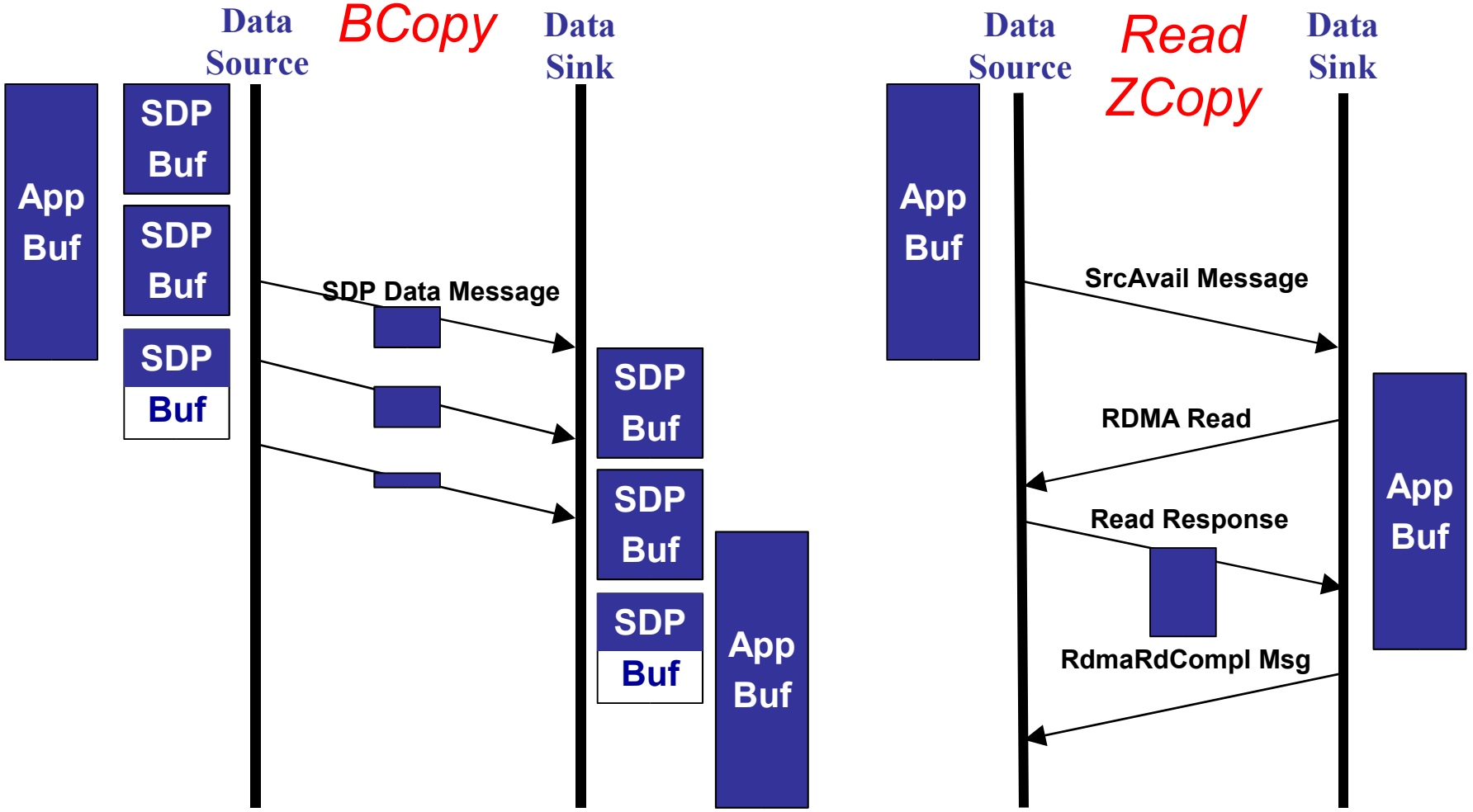
Sockets Direct Protocol (SDP) Overview

- Transparent to the application
- Maintains SOCK_STREAM Semantics
- Leverages InfiniBand Capabilities
  - Transport Offload – Reliable Connection
  - Zero Copy – Using RDMA
  - Kernel bypass (implementation dependent)
- Standardized wire protocol

SDP Stack Overview



Data Transfer Modes



SDP vs WSD

	Sockets Direct Protocol (SDP)	Windows Sockets Direct (WSD)
API	Winsock 2.x, POSIX/BSD like API	
WHQL	None	SAN / Winsock Direct
Wire Protocol Specification	IB spec 1.2	Microsoft Proprietary
OS Support	Win XP, WS 2K3	WS 2K3
Interoperability	Windows, Linux and any OS that conforms to IB specification	Windows Server only
Code	Open-source	Protocol - Windows proprietary SAN Provider - open-source
IHV Module	Socket Provider Library SDP kernel module	SAN Provider Library
Implementation Domain	Mostly kernel (similar to Linux)	Mostly user

ULP Comparison

	IB Verbs	SDP/WSD	IPoIB
API	Low level	Socket (TCP only)	Socket
Latency	lowest	lower	low
Bandwidth	highest	high	medium
CPU Utilization	lowest (if not polling)	BCopy high ZCopy low	highest
Kernel bypass		SDP WSD	
Stack Overhead	Light	Medium	High
Memory Registration	Explicitly by application/ middleware	Heuristics by SDP	None
Application Adaptation	Porting/ Development Required	Supports Unmodified Application	

SDP Socket Provider

- User-mode library
- Implements Socket Provider Interface (SPI)
  - Supports TCP protocol
  - WSPxxx function for each socket call
- Socket switch implemented in the library
  - Policy based selection of SDP vs TCP
- SDP calls are redirected to SDP module

SDP Module

- Kernel module
  - Implemented as a high level driver
- Connection establishment
  - Routing
  - ARP through IPoIB
  - Path Record
  - CM
- Data transfer mechanism
  - BCopy for first release
  - Using physical memory region for local SDP buffers

Current Status

- Maintainer
  - Tzachi Dar
- Code supports
  - Socket, Connect, Bind, Listen, Accept, Close
  - Send/Recv
  - BCopy mode only
- Planned BCopy release Sep/05
  - Will be posted to openib.org svn

Thank You !