

- Remote Direct Memory Access

- RDMA over IP:

- Direct Data Placement

- random access data transfer

- RDMA

- memory behavior (read/write/atomic...)

- over IP

- economies of scale

- CPU and network throughput vs. memory bandwidth

- The copy problem solutions

- page flipping

- scatter/gather API

- direct data placement/access

- Hardware vs. Software, Rx vs. Tx side

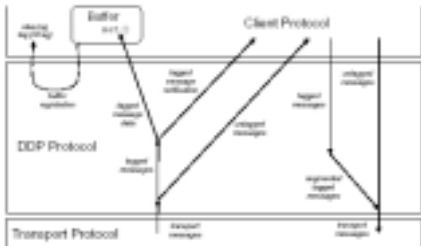
## main idea

- receivers register buffers (with some range of addresses) and create “steering tags”
- sender sends stag + data (+ offset)
- NIC/protocol/network steers data to appropriate buffer

## Remote Direct Data Placement RDDP

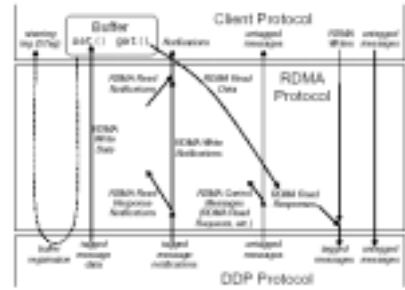
- message oriented protocol – transport + message delimiting (ULP framing)
- direct data placement
  - message segmentation and random access buffers
- RDMA
  - memory semantics: read, write, flow control, atomic ops
- UL client protocol

## DDP



tagged and untagged messages  
 buffer: set(address, data), get(address), stag  
 notification

## RDMA layer



notifications:  
 delivery of untagged msg (the msg itself),  
 delivery of tagged msg  
 rdma read completion  
 errors