Direct Data Placement (DDP) over Reliable Transports

55th IETF – Atlanta

20th November 2002

draft-shah-iwarp-ddp-01

Hemal V. Shah, Intel Corporation
hemal.shah@intel.com

DDP Overview

- A Message oriented protocol that supports two data transfer models
  - Tagged Buffer data transfer model
    - Data Sink advertises an identifier (STag) for the ULP buffer
    - Data Source specifies STag and Tagged Offset (TO) to transfer data to a portion of the Tagged Buffer (sender-based ULP buffer management)
    - Allows multiple DDP Messages targeted to a Tagged Buffer with a single buffer advertisement
  - Untagged Buffer data transfer model
    - Enables data transfer without requiring buffer advertisement
    - Receiver can queue up a series of ULP buffers to specify the order in which the buffers will be consumed (receiver-based ULP buffer management)
    - Each Untagged DDP Message from the Data Source consumes an Untagged Buffer at the Data Sink
    - Requires associating a receive ULP buffer (based on MSN and QN fields) for each DDP Message
      - If an Untagged DDP Message arrives without an associated Untagged Buffer, the DDP Message is dropped and DDP Stream is terminated

DDP Overview (Continued)

- DDP segments each DDP message into DDP Segments
- Each DDP Segment is self-describing for data placement
  - DDP enables reassembly of ULP Payload contained in DDP Segments of a DDP Message to occur within associated ULP Buffer
- DDP provides in-order delivery to ULP
  - A Message Payload is delivered when
    - All DDP Segments of a DDP Message has been received
    - Payload of the DDP Message has been Placed into the associated ULP buffer
    - All prior DDP Messages have been Placed
    - All prior DDP Messages have been Delivered
- Each DDP Stream is mapped over an LLP Stream that provides in-order, reliable delivery

Comments on DDP draft -01

- STag validation semantics Clarification
  - STag validation on a DDP Stream
- DDP Segmentation clarification
  - MO or TO ordering, Overlap in DDP Segment, and message Interleaving
- Should DDP draft define requirements for unreliable transports?
- Local interface requirements for buffers
- Usage of RsvdULP field in DDP Header

STag Validation Semantics

- Different Models for STag Validity
  - Unspecified (unacceptable)
    - Pro: Minimal specification
    - Con: Vulnerable to attacks
  - STag is associated with exactly one DDP Stream by the ULP
    - Pro: Prevents all accidental over-exposure of STags on multiple DDP Streams
    - Con: Restricts STag usage per DDP Stream
  - STag is associated with an access group of DDP Streams by the ULP
    - Pro: STag is valid on all DDP Streams that are part of the same the access group
    - Con: Access group management
  - Explicit:
    - Pro: ULP specifies the DDP Streams on which the STag is valid (simple specification)
    - Con: Adds complexity of managing and searching the list of DDP Streams per STag
- Exactly one DDP Stream and an access group of DDP Streams seem to be reasonable models (based on the discussion on the reflector)

Clarifications on DDP Segmentation

- Issue: Order of DDP segments in a DDP Msg
  - Increasing MO order for Untagged DDP Message
  - Increasing TO order for Tagged DDP Message
- No overlapping of payload is allowed among DDP Segments of a DDP Message
- In a DDP Stream, interleaving of DDP Segments of different DDP Messages is not allowed at the Data Source
  - Data Sink is not required to verify

Requirements for Unreliable Transports

- Should DDP draft define requirements for Unreliable Transports?
  - No, the draft is “DDP over Reliable Transports”

Local Interface Requirement for Buffers

- Do we need to specify requirements for the local interface to DDP for Untagged and Tagged buffers?
- Yes: from the perspective of getting access control and protection behavior for buffers right
- No: from the perspective of full API specification and API rules

RsvdULP field in DDP Header

- Usage:
  - To allow ULP to pass control fields in the DDP Header
  - To avoid making space for ULP control fields between DDP header and payload to be placed
- 1 octet for Tagged DDP Messages
- 5 octets for Untagged DDP Messages