Tag and End of Message Bits

Brian Batchelder

1394PWG

January 22, 1999

Tag Bit - Definition

- From the PWG Requirements Documents:
  - 1.2.3 Data tagging
    - Data can be tagged as "special data" by the sending endpoint. The transport will indicate to the receiving endpoint that the data is tagged. This is also known as out-of-band data. The special data is synchronous with the rest of the data.
- The tag bit is the mechanism the sending transport uses to notify the receiving transport that the data is tagged.

Tag Bit - Examples

- Stream Job-related messages
  - Spooler can send data-synchronous “StartJob”, “EndJob”, etc. messages WITHOUT understanding nor modifying the data.

Tag Bit - Other Transports (1284.4)

- Definition: **Out-of-band packet:** A packet of exceptional data in the data stream. This type of packet may contain special information relating to the data at this position in the data stream, for example, “End of Job.”
- Implemented as bit in 1284.4 header

Tag Bit - Other Transports (TCP)

- Urgent Data
  - “TCP also provides a means to communicate to the receiver of data that at some point further along in the data stream than the receiver is currently reading there is urgent data. TCP does not attempt to define what the user specifically does upon being notified of pending urgent data, but the general notion is that the receiving process will take action to process the urgent data quickly.”
- Implemented as bit in TCP header

Tag Bit - Implementation

- The tag bit is written to memory by the transport sending the data.
- I2T Data
  - Set by initiator.
  - Add Tag bit to CDB.
- T2I Data
  - Set by target
  - Add Tag bit to the status block.

End of Message Indicator

- From the PWG Requirements Documents:
  - 1.1.2 Provide in-order, byte-stream and in-order, datagram services
    - Datagram: Data is guaranteed to be delivered to the receiving endpoint in the same form as it was presented by the sending endpoint. For example, if data is presented by the sending endpoint in a buffer of 30 bytes, it must be delivered to the receiving endpoint in a buffer of 30 bytes. The transport stack may limit the size of datagrams. It does not have to support segmentation and reassembly of client buffers that do not fit in a single transport packet.
    - Issue: *Can the transport really limit this to a single transport packet, or do the clients require larger datagrams?*

EOM Indicator - Examples

- Target w/ limited buffer space
  - A client datagram might not fit within a single transport buffer. It can be split across multiple buffers.

EOM Indicator - Examples

- Start sending message before fully created
  - Allows performance optimization ala HTTP 1.1 and print spoolers, which can start sending the message before they have finished creating it. The receiver can then begin processing the message earlier.

EOM Indicator - Other Transports (HTTP)

- HTTP uses “chunking” to send messages of indeterminate sizes.
- End of Message is indicated using a zero-length packet.
- Doesn’t feel “clean”

EOM Indicator - Implementation (Option 1)

- Client Implementation
- The EOM indicator is set by the client sending the data on last or only message buffer.
- I2T Data/T2I Data
  - Client sends tag bit and packet indicating “End of Message”

EOM Indicator - Implementation (Option 2)

- Transport Implementation
- The EOM indicator is set by the transport sending the data on last or only message buffer.
- I2T Data/T2I Data
  - Transport sends zero-length packet after end of message.

EOM Indicator - Implementation (Option 3)

- Transport Implementation
- The EOM indicator is set by the transport sending the data on last or only message buffer.
- I2T Data
  - Initiator sets EOM bit in CDB.
- T2I Data
  - Target sets EOM bit in the status block.