Packet Reordering Metric for IPPM - Comments from WG Last Call

`http://www.ietf.org/internet-drafts/draft-ietf-ippm-reordering-09.txt`

*

Al Morton

Len Ciavattone

Gomathi Ramachandran

Stanislav Shalunov

Jerry Perser

March 7, 2005

What is Packet Reordering?

- **Packets arrive at Dst, but not in send order.**
  - → 1, 2, 3, 7, 8, 9, 10, 11,... Loss,no reordering
  - → 1, 2, 3, 7, 8, 9, 4, 5, 6, 10, 11,...reordering

- **In the “world of order” all these packets are of interest.**
  - → 1, 2, 3, 7, 8, 9, 4, 5, 6, 10, 11,...
  - → | Early | Late |
  - → No reordering until Late Packets Arrive
  - → # of Early Packets => Reordering Extent

Comments on draft 08 in Last Call

- Comments Vern Paxson and Phil Chimento (off-list); Mark Allman and Michal Przybylski (on-list)
- Overall - critical to underline that inter-packet spacing, variations in packet sizes, and flow identifiers all have potential for *major* impact
  - ➔ New Section 2.3 “Required Context for All Reordering Metrics”
  - ➔ Whenever a metric is reported, it MUST include the parameters above to provide context.
- Section 2: several small edits in first paragraph
- Section 2.2: Clarify goals of “Quantification methodS” and “concatenation” to estimate E2E
- Section 3: Mention generalized req. for order identification from a mathematical POV in the Reordered Definition
  - ➔ BUT we quickly narrow down to message numbers only

Comments on draft 08 in Last Call

- Section 3: Duplicate Packets are reordered?
  - ➔ The Metric Definition is valid, keeps Dups orthogonal to reordering
- Section 3.2: NextExp parameter still had time&bytes
- Section 3.2: SrcByte and PayloadSize Parameters Optional
  - ➔ but, left SrcTime Mandatory, as with T for Loss, 1-way Delay
  - ➔ Consistency: OWAMP packet format has a Seq. Num. &TS
- Section 3.4: Loss and Reordering cannot be completely untangled, a reordered packet could be subsequently lost. (added a para. noting this).
- Section 3.4: a Sequence Discontinuity is only local; there may be other instances of discontinuities.
  - ➔ Packet arrival order can influence the number of Discon.

Comments on draft 08 in Last Call

- **Section 4 : Harmonize terminology - make it uniform to help the reader.**
  - ➔ **All the parameters, even the ones about to be defined, are called out in the Metric Parameters Section**
  - ➔ **Several changes: LateTime(i) ⇒ LateTime(s[i]) so we relate a metric to a packet by sequence number**
  - ➔ **No parameters are re-used, all have unique definitions & some params moved earlier to formalized definition in 4.1**
- **Section 4.2.4: Toned down "A receiver must possess storage to restore order..."**
- **Section 4.3: For the arrival sequence 1, 10, 5 (where packets 2, 3, 4, and 6 through 9 are lost)**
  - ➔ **LateTime would not indicate exactly how "late" packet 5 is from its intended arrival position.**
  - ➔ **Interpolation would be like a Single-Point Delay Variation Metric (see Recs I.356 and Y.1540)**

Comments on draft 08 in Last Call

- **Section 4.4: Byte Stream Offset Def. is confusing, difference between definition and equation?**
  - ➔ **New Definition, earlier packets with higher # are buffered**
  - ➔ **works like a real buffer intending to restore order**
  - ➔ **Tends to Obviate specification of Reordering Buffer Density**
- **Section 4.6.2: (Reordering Free Runs) Parameter "q" was mislabeled**
  - ➔ **q is the sum of the squares of run lengths**
  - ➔ **no need to store each run length to compute q**
  - ➔ **Edited for clarity and integration with other metrics**
- **Section 4.6.4: (Discussion) More like an example**
  - ➔ **Revised the definitions to match the example! (min run=0)**
  - ➔ **More discussion, e.g., how q is useful**
  - ➔ **Also revised Section 7.4**

Comments on draft 08 in Last Call

- Section 5 : No Comment - most reviewers have not commented on this section (?)
- Section 5 : *Still Bogus*, but less so since requiring BTC. Now there are Issues with the two main justifications for this metric:
  - “...useful for determining the portion of reordered packets that can or cannot be restored to order in a typical TCP receiver buffer based on their arrival order alone.”
  - “- For n=3, a NewReno TCP sender would retransmit 1 packet in response to an instance of 3-reordering and therefore consider this packet lost for the purposes of congestion control (the sender will half its congestion window)”
  - BTC sending is window-based, and it is possible to
    - see if reordering can be sorted out within the window, but this metric does not “know/use” the window size
    - the BTC stream will exhibit retransmission if needed, so why do we need n-reordering to tell us this?

Comments on draft 08 in Last Call

- **Section 5.4 Revisions:**
  - ➔ Rip-out the explanations that n-reordering can predict “packets that are as good as lost” on its own.
  - ➔ Add that n-reordering is helpful for matching the duplicate ACK threshold setting to a given path. For example, if a path exhibits no more than 5-reordering, a threshold of 6 may avoid unnecessary retransmissions.
  - ➔ Now that we’ve required BTC sending, remove the “less complicated than TCP” statement.
  - ➔ Added References to RFC 2581 TCP Congestion Control and RFC 2960 SCTP in the discussion of n=3.

Comments on draft 08 in Last Call

- **Section 6: Measurement and Implementation Issues**
  - ➔ **Add a clear statement that to gauge reordering for an application, it is RECOMMENDED to use the same sending pattern as the application of interest.**
  - ➔ **Poisson Streams: can’t make inferences to app. perf.**
  - ➔ **Suggest TCP Timestamp option (RFC 1323) as a way to disambiguate TCP Retransmits (agreed)**
  - ➔ **Removed “...the closest possible spacing should reveal the greatest extent of steady-state reordering”**
  - ➔ **Inserted Cautions associated with testing at link-speed, applicable to payload pattern testing, too.**
    - ✦ **“...streams sent at the link speed serialization limit MUST have limited duration and MUST consider packet loss as an indication that the stream has caused congestion, and suspend further testing.”**
  - ➔ **Removed “Some in-order packets may not be useful to TCP...”**

Comments on draft 08 in Last Call

- **Section 7: (Examples)**
  - → **“add pkt length”, but they don’t fit in the margins, size stated in the text**
  - → **“they’re great”, we’ll keep’em**
- **Section 8: (Security) Why store user payloads?**
  - → **Clarified that user payloads are only temporarily stored for size computation, and that a hashing function should be suitable for comparison purposes**
- **Appendix A: Example 2 code only computes # packets reordered for one method at a time**
  - → **Modified to compute both and compare**

Summary

- Work accepted by IPPM in March 2002
- 1st Last Call on version 07, Oct 2004
  - ➔ most comments addressed, but a few more showed up in...
- 2nd Last Call on version 08, Dec 04/Jan 05
  - ➔ Comments Addressed in version 09
- Here's a snapshot of the changes:

  http://home.comcast.net/~acmacm/
- Any more Comments?
  - ➔ (or are we ready for *another* Last Call?)