# Packet Reordering Metric for IPPM

**http://www.ietf.org/internet-drafts/draft-ietf-ippm-reordering-06.txt**

*****

**Al Morton**

**Len Ciavattone**

**Gomathi Ramachandran**

**Stanislav Shalunov**

**Jerry Perser**

**August 2, 2004**

# *Definition:* Type-P-Reordered

- Source applies a Message Number, (or Payload Number, or Time Stamp) as the basis for determining order.
- Destination knows the "Next Expected"

A reordered packet outcome occurs when :

The packet has a Source sequence number lower than the Next Expected, and therefore the packet is reordered. The Next Expected value does not change on the arrival of this packet.

On successful arrival of a packet with sequence number n:

```
if s >= NextExp, then    /* packet in-order */
        NextExp = s + payload_size + 1;
else                             /* when s < NextExp  */
        designate packet s as reordered;
```

# *Changes in draft 06*

- Appendix on Fragment Reordering is *Informative*
- Ran I-D nits tool, fixed some minor issues
- Added Table of Contents (over page threshold)
- Many New Readers and Comments (no major changes)
- Comments on the List: Fabien Michaut
  - 1. On page 13, "Definition 2 : The degree of n-reordering of the sample is m/l" . It seams that m is not previously defined.
  - 2. On page 14, What do you mean by adjacent? "1. n is a count of *adjacent* early packets." Is it always the case? Consider the following example : In the arrival sequence s={1, 2, 3, 7, 9, 4, 5, 6, 8}, packet 4 is n-reordered with n=2, but packets 7 and 9 are not adjacent.

  ➔ Defined “m” in Section 5.1.3 and

  ➔ clarified adjacent to mean “consecutive arrival positions” in 5.1.4

# *Changes in draft 06*

- **Comments from David Newman (page 1 of 2)**
  - **Section 2: define minimal level of "orderedness" expected in a given sample**
    - **this a general measurement Issue, mentioned in the expanded steady-state and transient reordering paragraph.**
  - **Section 3: it would be useful to define a state for packets that are received in-order, but greater than NextExp.**
    - **New section 3.4 defining Sequence Discontinuities and SeqDiscontinuitySize**
    - **Added a pointer in section 4.4.3 on Reordering Discontinuities**
  - **Section 4.2.2: "packet emission" is ambiguous as to ingress or egress.**
    - **changed to "packet transmission"**
  - **Section 4.5.3: variable "P" was undefined**
    - **should have been lower case "p", as in p++**

# *Changes in draft 06*

- **Comments from Henk Uijterwaal:**
  - **Clarify Goal and Objectives statements (Section 2.2)**
    - **MUST have one or more relevant applications, such as receiver design or network characterization.**
    - **Metrics SHOULD became "It is desirable for Reordering Metrics to have one or more of the following attributes:"**
  - **Clarified Definition in Section 3.3: when packet is reordered Type-P-Reordered = TRUE**
  - **More consistent sub-section outlines in Sec 4 and 5**
    - 4.x.1. Metric Name
    - 4.x.2. Parameters
    - 4.x.3. Definition
    - 4.x.4. Discussion
    - **Required separate sub-sections for Late-Time Offset and Byte Offset metrics.**
  - **All Discussion sub-sections give the application of the metric (was missing from "Gaps" and "Runs" metrics).**

# *Changes in draft 06*

- **Comments from Michal Przybylski (on the list):**
  - ➔ **Section 6, noted that minimum packet spacing (at wire speed) may detect the greatest extent of reordering.**
  - ➔ **Also, certain patterns of packet lengths (bursts of long, then short) may reveal parallel paths.**
  - ➔ **Agreed that a "derived metric" for Receiver Buffer occupation would be useful -- using packet loss, reordering, and ?(IPDV?) metrics as a foundation. (This is beyond the scope of the current ippm-reordering draft.)**
- **Bartek and Przybylski successfully implemented much of ippm-reordering-05**

# *Changes in draft 06 and positive feedback*

- **More Comments from David Newman (for Sec. 6)**
  - ➔ **Duplicate packet detection and exclusion -- added the "low storage" suggestion to measurement issues.**
  - ➔ **Use of a sliding history window helps to reduce storage requirements. Setting an upper bound on the "useful" reordering extent determines the storage size necessary.**
  - ➔ **Issue to note: determining reordering extents and gaps is tricky when there are overlapped or nested events - test instrument and reordering complexity are directly correlated.**
  - ➔ **Recognized several issues for implementation, so re-titled section `6 Measurement and Implementation Issues`**
- **Comment from Nathan Kube, at University of Victoria in British Columbia, Canada.**
  - ➔ **I read ... (draft-ieif-ippm-reordering-05.txt). I have found many of your metrics to be quite useful in my research.**

# *What about Reordering Density (RD)?*

- **Brief list discussion compared the metrics, and concluded (based on version 02):**
  - ➔ **both are useful**
  - ➔ **RD includes <u>both</u> Loss and Reordering => badly named**
  - ➔ **RD is really a "Derived Metric", based on singletons of Loss and Reordering.**
  - ➔ **Treat this as a separate work item?**
    - ✦ **consider re-naming the metric "Receiver Buffer Density" or something similar**
    - ✦ **re-write the Intro with Receiver Emphasis**

# *Summary*

- **Seven versions of the WG draft in 2 years.**
  - ➔ **Lots of time for readers to find it**
  - ➔ **Productive discussion at ~every meeting back to IETF-53**
- **Lots of good comments and feedback on 05, in 06**
- **Several independent implementations (early)**
- **Any more Comments?**
  - ➔ **(or are we ready for IPPM WG Last Call?)**