Advancing RFC 4138

<draft-ietf-tcpm-rfc4138bis-01>
<draft-kojo-tcpm-frto-eval-01>

Pasi Sarolahti

Markku Kojo

Kazunori Yamamoto

Max Hata

IETF-71 / TCPM / Philadelphia, PA, USA / March 11th, 2008

Problems with regular TCP

- On Spurious Timeouts:
  - Regular TCP sender retransmits whole window unnecessarily in slow start
  - Network resources are wasted
  - In many cases severe performance penalty to the TCP flow
  - Dishonors packet conservation principle

F-RTO: Detecting Spurious RTOs

- F-RTO slightly modifies TCP sender behavior
  - After RTO retransmission try to send a couple of new segments
  - If new acknowledgements for non-retransmitted segments flow in, assume RTO was spurious
  - Otherwise new segments trigger DupACKs, and sender should assume genuine RTO
- No TCP options required
- Compatible with existing TCP implementations
- Does not cause network congestion
- Might not detect spurious timeout in some cases
  - If F-RTO does not detect spurious RTO, it reverts back to traditional RTO recovery

Current Progress

- Revised RFC 4138 targeting at PS <draft-ietf-tcpm-rfc4138bis-01>
- No changes since last meeting
    - We consider draft ready

Evaluation of RFC 4138

- Accompanying draft evaluating RFC 4138
  - <draft-kojo-tcpm-frto-eval-01.txt>
  - Points out the problems with regular RTO recovery and usefulness of F-RTO
  - Evaluates F-RTO to show it is not harmful to the network, corner cases included
  - Summarizes experimentation results

Ready to advance?

- A number of known F-RTO implementations are out there
- Proposals and support to advance to PS have been expressed several times by implementors
- Experimentations have been carried with several implementations showing positive results
- All feedback has been positive
    - Implementors: no issues with the spec
    - Many implementations enable F-RTO by default
        - Windows Vista
            - Microsoft report at IETF-68 about their positive experiences
        - Linux:
            - basic F-RTO implemented since the early days of F-RTO algorithm
            - SACK enhanced F-RTO enabled by default from up-coming release of 2.6.24 and onward, and falls back to basic variant if SACK not negotiated

Next Steps

- Ready for WGLC?