Advancing RFC 4138
<draft-ietf-tcpm-rfc4138bis-01>
<draft-kojo-tcpm-frto-eval-01>

Pasi Sarolahti
Markku Kojo
Kazunori Yamamoto
Max Hata

IETF-70 / TCPM / Vancouver, BC, Canada / December 4th 2007
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>
• Changes from -00:
  • Added back the original SACK-algorithm from RFC 4138 after the common feedback to have the SACK-algorithm in the document
  • Clarified behavior on multiple timeouts
  • Clarified that ACKs that do not acknowledge new data but are not duplicate acknowledgements are ignored
  • Other small clarifications on both algorithms and general editing
  • Added one paragraph describing the basic idea of the SACK algorithm
Current Progress (cont’d)

• Wrote I-D "Evaluation of 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

• Changes from -eval-00:
  • Added a summary on experimentation with malicious receiver
    • receiver does not benefit from cheating when conservative response is used
    • receiver may benefit when aggressive response is used
  • General editing
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: straightforward to implement
    no issues with the spec
• Many implementations enable F-RTO by default
  • Windows Vista
    • Microsoft report at IETF-68 about their positive experiences
  • Linux
    • 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

• Basically ready for WGLC

• First need **green light** from WG for advancing to PS