Advancing RFC 4138
<draft-ietf-tcpm-rfc4138bis-00>
<draft-kojo-tcpm-frto-eval-00>

Pasi Sarolahti
Markku Kojo
Kazunori Yamamoto
Max Hata

IETF-69 / Chicago, IL, USA / July 26th 2007
Spurious Retransmission Timeouts

- Delay spikes occur on wireless networks due to
  - handoffs
  - link-layer error recovery
  - bandwidth variation
- Delay spike may trigger TCP retransmission timer
- Problems:
  - Regular TCP sender retransmits whole window unnecessarily in slow start
  - Wasted network resources
  - Dishonors packet conservation principle
  - In many cases severe performance penalty to the TCP flow
F-RTO History

- Experimental RFC 4138, Aug 2005
- A number of known F-RTO implementations are out there
- Experimentations have been carried with several implementations showing positive results
- Proposals to advance to PS have been expressed earlier several times
- Advancing to PS was discussed in IETF-67 & IETF-68
  - Internet-Draft "Evaluation of RFC 4138" <draft-kojo-tcpm-frto-eval-00.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
F-RTO: Detecting Spurious RTO

• 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 performs as standard TCP
Evaluation Report

• Test runs in emulated wireless network
  • Linux implementation
  • Different delay & loss scenarios to verify that F-RTO works as expected
• Test runs in real W-CDMA network
  • HP-UX server at fixed end
  • Different terminal mobility patterns
  • Amount of unnecessarily retransmitted data reduced by 82%
  • F-RTO detected 71% of the spurious timeouts
    • In 28% of cases F-RTO could not be applied because there were no new data to send
    • In 0.7 % of cases advertised window limited sending of new data
    • In 0.3 % of cases duplicate ACKs prevented F-RTO
• Microsoft report at IETF-68 about their positive experiences
  • Based on expirements -> F-RTO enabled by default in Vista
Current Progress and Next Steps

• Revised RFC 4138 targeting at PS
  • Specify basic algorithm and TCP only
  • Leave the following as experimental and do not include in the Standards Track specification
    • F-RTO with SCTP
    • SACK-Enhanced variant of F-RTO

• Response
  • Proposal: simple response will be outlined in this document
    • Do not retransmit outstanding segments after detecting spurious RTO
    • Follow RFC 2581 on congestion control
  • Possible to apply other proposed responses documented in separate RFCs