Evaluating F-RTO (RFC 4138)

Markku Kojo, Kazunori Yamamoto, Max Hata, Pasi Sarolahti

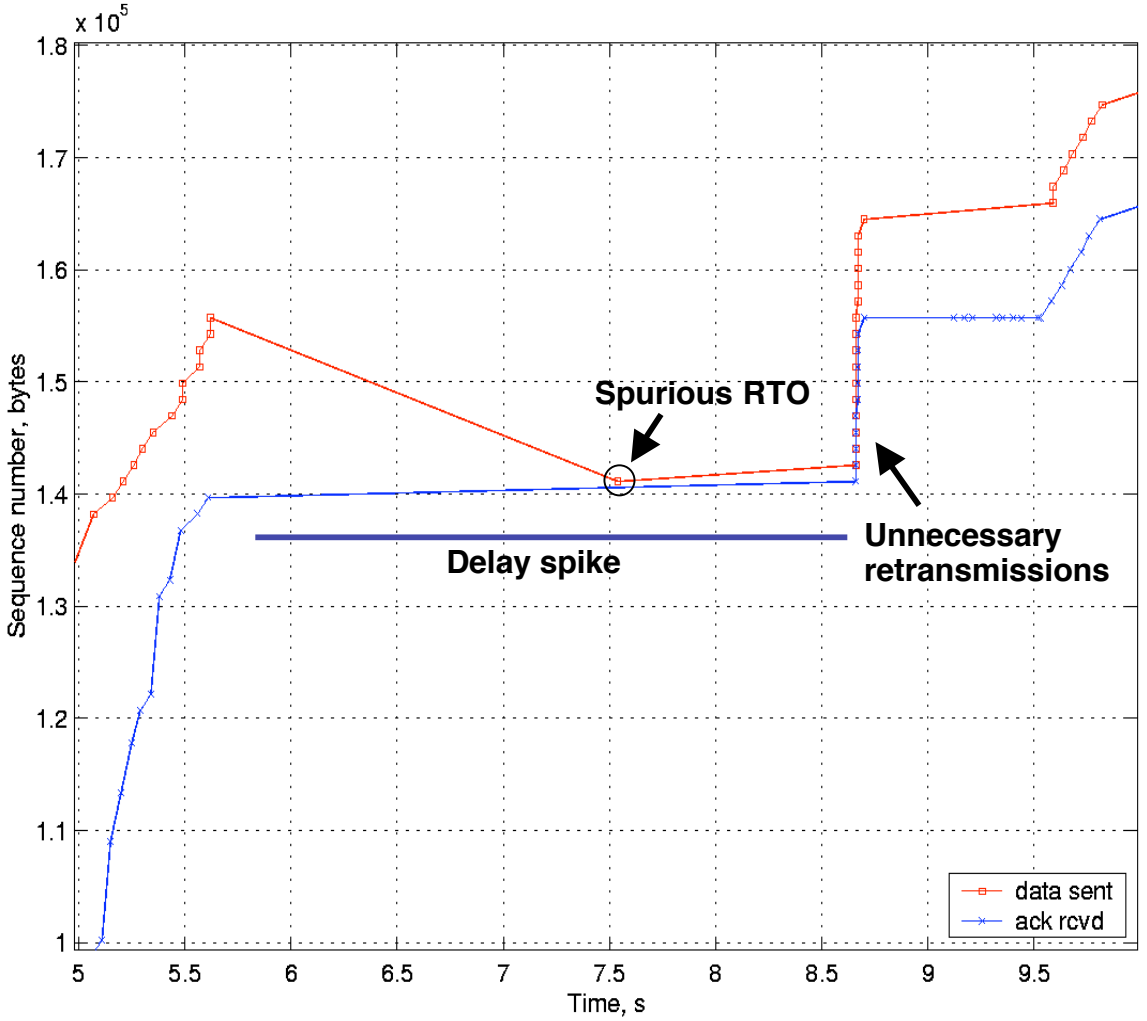
Draft available at: http://www.cs.helsinki.fi/u/sarolaht/frto/

History

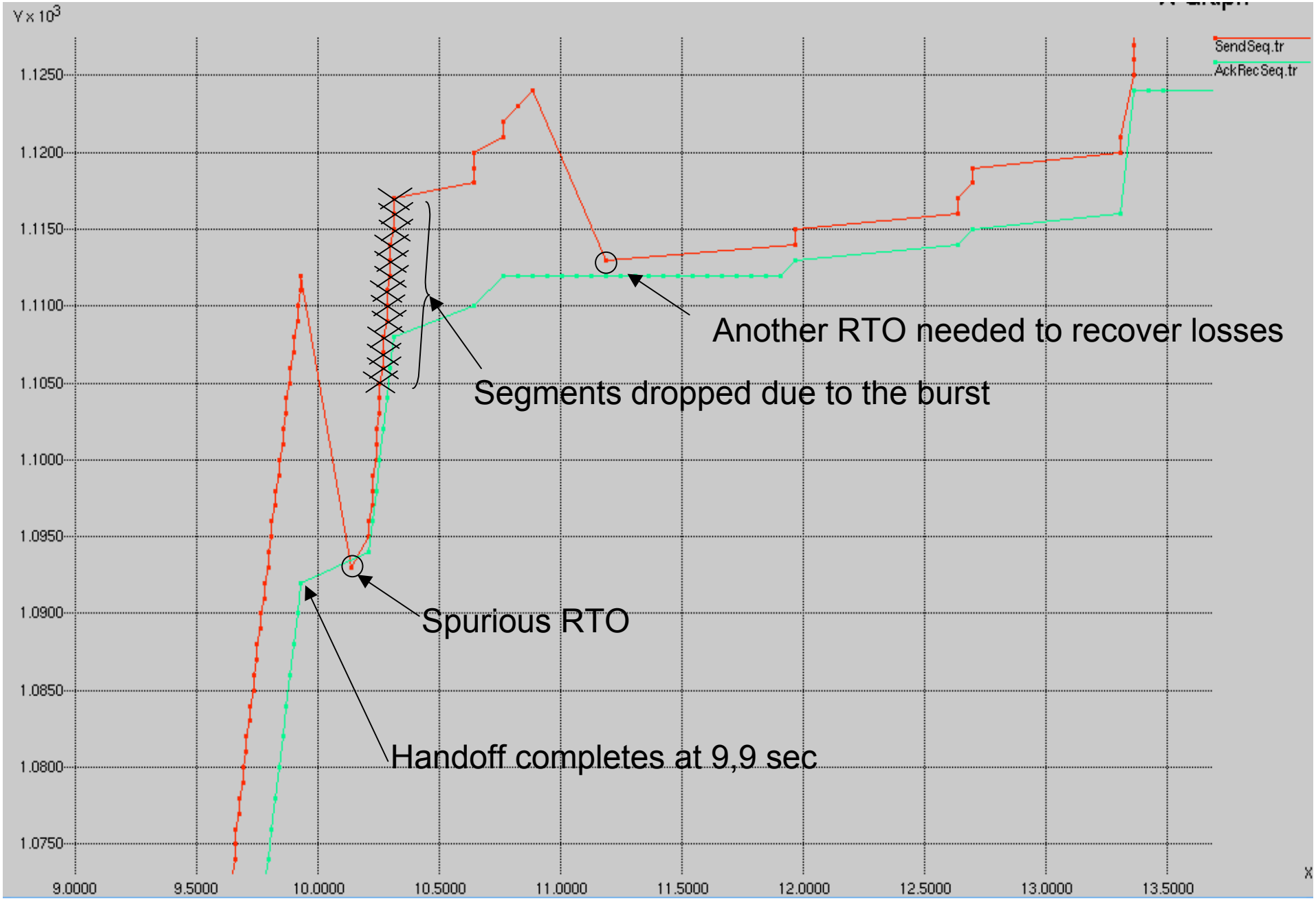
- **Experimental RFC 4138, Aug 2005**
- **A number of known F-RTO implementations are out there**
- **Experimentations have been carried with several implementations showing positive results**
- **Proposals to advance to PS have been expressed earlier**
- **Advancing to PS was discussed in IETF-67**
  - **We were asked to write a document that**
    - **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**
- **As a first step:**
  - **We wrote Internet-Draft *"Evaluation of RFC 4138"***
    - **<draft-kojo-tcpm-frto-eval-00b.txt> (not yet in repositories)**
    - **Available at: http://www.cs.helsinki.fi/u/sarolaht/frto/**

Spurious RTOs on Regular TCP

- 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
  - Network resources are wasted
  - Dishonors packet conservation principle
  - In many cases severe performance penalty to the TCP flow



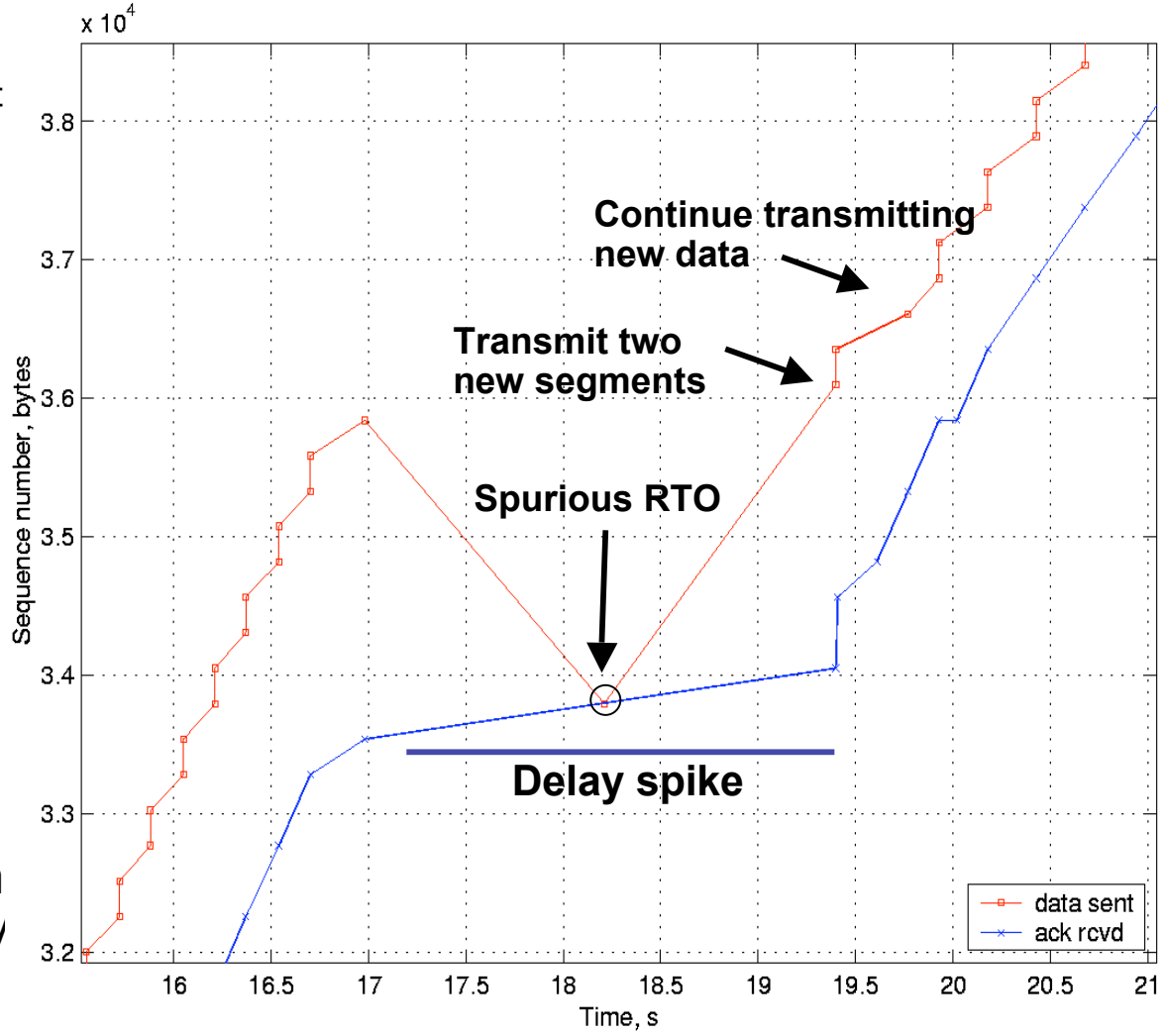
Spurious RTO due to vertical handoff from a low-latency to high-latency access link



Spurious RTO and F-RTO

- When delay spike causes RTO to expire, retransmit 1st unacknowledged segment
- 1st ACK acknowledges the retransmission: send 2 new segments
- 2nd ACK acknowledges data that was not retransmitted: RTO is declared spurious

- Benefits of F-RTO:
  - Avoids unnecessary retransmissions
  - Allows adhering to packet conservation principle
  - Prevents the TCP flow from severe performance penalty
  - Enables RTT samples from delayed segments



Can F-RTO be harmful? NO!

- If RTO is not spurious (or F-RTO fails to detect)
  - F-RTO reverts back to traditional RTO recovery
  - Exactly same amount of segments get transmitted
- Hidden losses when F-RTO declares RTO spurious
  - A few known scenarios
    1. Loss of the unnecessary RTO retransmission
    2. Severe reordering
       - retransmitted segment bypasses the full window of original segments
    3. Malicious receiver
       - Delays ACKs until RTO expires and retransmitted segment arrives
       - ACKs data it has not received
  - 1 & 2 considered as rare corner cases; won't harm TCP flow
  - With 3 benefit is questonable; concealing losses harms TCP flow
  - None of these can harm the network, if conservative response is taken
    - F-RTO sender is recommended to take the spurious RTO as a congestion signal

Next Steps

- Revise 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?
    - do not specify any response in the new draft, or
    - recommend implementing conservative response, i.e., take spurious RTO as a congestion signal
      - possibly include guidelines for a conservative response
  - Maybe specify a conservative response in a separate document?