Faster Restart for TCP Friendly Rate Control (TFRC)

draft-ietf-dccp-tfrc-faster-restart-05.txt

E. Kohler, S. Floyd, and A. Sathiaseelan

March 2008,

DCCP Working Group

Faster Restart for TFRC:

- After an idle period of at least NFT (no feedback):
  - The allowed sending rate is not reduced below *twice* the initial sending rate;
  - Quadruple sending rate each RTT up to old rate (decayed over time);

There have been no changes since last IETF!

- There will be a report on new simulation results before the next IETF.

Old slides:

-

Changes from draft-ietf-dccp-tfrc-faster-restart-03.txt:

- Removed Section 4.1 on receive rate, after it is made into a Errata for RFC 4342. Feedback from Gerrit Renker.
- Additional reporting on simulations.
- Added a section on Interoperability Issues.
- Specified CCID 3 and 4 impact in the introduction.
- Nits from Gorry Fairhurst and Arjuna.
- Changed targeted decay time to configurable DelayTime. Feedback from Gerrit Renker.

Performance after long idle periods:

- RFC 3448:
  - Allowed sending rate is halved when NoFeedback Timer expires, down towards initial sending rate.
  - First feedback packet after idle period reports receive rate of one packet per RTT.
    - Allowed sending rate is at most twice receive rate.
- RFC3448bis after a long idle period:
  - First feedback packet after idle period reports receive rate of one packet per RTT.
    - Receive rate is NOT based only on this feedback packet.
- RFC3448bis with Faster Restart:
  - Allowed sending rate is halved when NFT expires,
    down towards *twice* initial sending rate.
  - Then each RTT quadruple allowed sending rate towards X_fast_max.

(X_fast_max: interpolated highest receive rate since last loss)

Performance in long data-limited periods:

- RFC 3448:
  - Allowed sending rate is at most twice:
    receive rate.
- RFC3448bis:
  - Allowed sending rate is at most twice:
    max (recent receive rate,
    receive rate before data-limited period).
- RFC3448bis with Faster Restart:
  - Allowed sending rate is at most:
    max (value from RFC3448bis,
    X_fast_max).

(X_fast_max: interpolated highest receive rate since last loss)

Faster Restart Interoperability Issues with RFC 3448:

- Faster Restart:
  - a sender-only change.
  - built upon RFC3448bis (not RFC 3448).
- How does Faster Restart interact with a receiver using RFC 3448?
  - Performance is NOT higher than with a receiver using RFC3448bis.
  - *No backwards interoperability issue*.

RFC 4342 Errata:

- Section 6 says:
  - 2. A Receive Rate option, defined in Section 8.3, specifying the rate at which data was received since the last DCCP-Ack was sent.
- It should say:
  - 2. A Receive Rate option, defined in Section 8.3, specifying the rate at which data was received over the last round-trip time.
- Makes CCID-3 consistent with RFC 3448 and RFC3448bis.

Faster Restart Interoperability Issues in DCCP's CCID 3:

- Faster Restart builds on RFC3348bis, not RFC 3448.
- New CCID-3:
  - CCID-3 with Faster Restart and RFC 4342 Errata.
- Old CCID-3:
  - CCID-3 without Faster Restart and RFC 4342 Errata.
- New CCID-3 improves performance after idle and data-limited periods.
- Performance with a new CCID-3 sender and an old CCID-3 receiver is similar to performance with new CCID-3 for both end-nodes.
  - Partial-deployment is NOT an problem.

Future simulations:

- Can Faster Restart negatively impact others?
  - Simulation work to consider reverse traffic.
  - Simulations for wireless.
  - Experiments to assess incentive for padding.
- Simulations will focus on packet drop rates during the Faster Restart period.
- Assess if it is safe for use in Internet.
  - If not, what needs to be evaluated?

End Date?

- Some simulations already done.
  - More are planned for January 2008.
- Expect to have answers for next IETF.
  - Also depends on maturity of RFC3448bis.