Moving NewReno from Experimental to Proposed Standard?

∗

draft-ietf-tsvwg-newreno-00.txt
http://www.icir.org/floyd/talks/newreno-Jul03.pdf
http://www.icir.org/floyd/talks/newreno-Mar03.pdf

Sally Floyd and Tom Henderson

July, 2003
TSVWG, IETF

Moving NewReno from Experimental to Proposed Standard?

$\ast$

- The NewReno Modification to TCP's Fast Recovery Algorithm.

RFC 2582, Floyd, S., and Henderson, T., Experimental, April 1999.

- This is widely implemented.

(E.g., The TBIT web page, ”http://www.icir.org/tbit/”.)

Reno vs. NewReno:

∗

- NewReno performs *dramatically* better than Reno when multiple packets are dropped from a window of data.

  – Simulation-based Comparisons of Tahoe, Reno, and SACK TCP, K. Fall and S. Floyd, CCR, 1996.

- We would recommend NewReno over Reno, for TCP connections when the other end does not use SACK.

- We know of one scenario where Reno performs better than NewReno: with no loss but reordered packets.

  – See ”http://www.icir.org/floyd/talks/newreno-Mar03.pdf”.

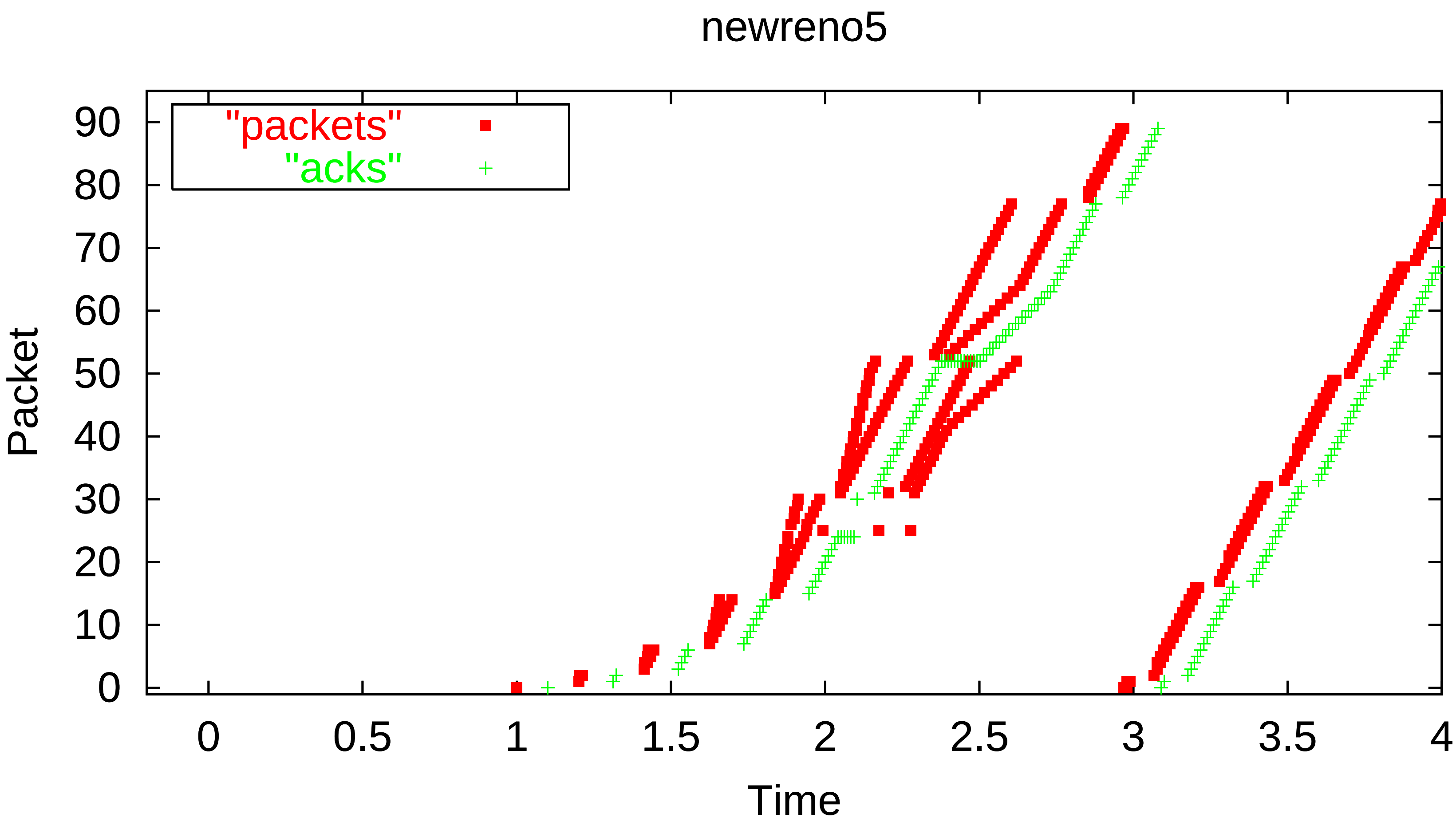
The main change to RFC 2582:

∗

- RFC 2582 describes a Careful and a Less Careful variant for avoiding multiple Fast Retransmits caused by the retransmission of packets already received by the receiver (bugfix), and recommends the Careful variant.

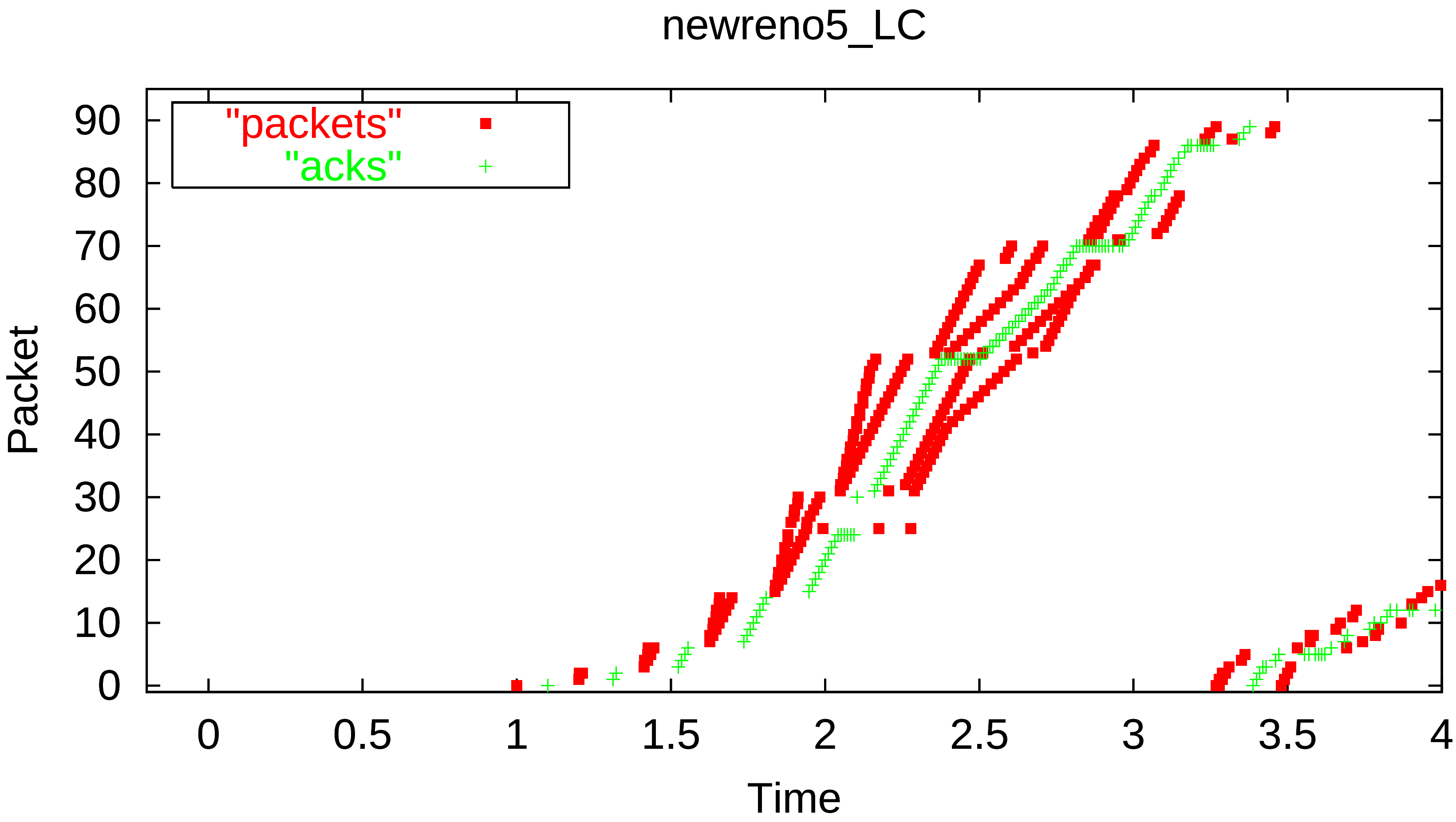
- The new draft *requires* instead of *recommend* the Careful variant.

The Careful NewReno and Reordered Packets:



```
./test-all-newreno newreno5
```

The Less Careful NewReno and Reordered Packets:



./test-all-newreno newreno5_LC

Other changes to RFC 2582

∗

- New section: ”Implementation issues for the data sender”.
  - Discusses methods for limiting bursts when exiting Fast Recovery.

- New section: ”Comparisons between Reno and NewReno TCP”.
  - Includes a discussion of the response to reordering, where NewReno performs worse than Reno or SACK.

- New section: ”Changes relative to RFC 2582”.

- Added a paragraph about differences between RFC 2582 and [FF96].

- RFC 2582 used two separate variables, ”send_high” and ”recover”, and this document has merged them into a single variable ”recover”.

Changes to make:

∗

- Mention, in section on ”Implementation issues for the data sender”, that the sender might want a separate flag to record whether it is in the Fast Recovery procedure.
  – For robustness with window updates and out-of-order acks.

- Add an implementation note about taking care about sequence wrap.

Questions:

*

- Is this ready for WG last call, to advance to Proposed Standard?
- Any other changes that should be made at this time?