RACK: a new fast recovery

Yuchung Cheng <ycheng@google.com>

IETF 91 tcpm Nov 10, 2014

Fast recovery should use time not counters

RFC3517: after `dupthresh` DUPACKs

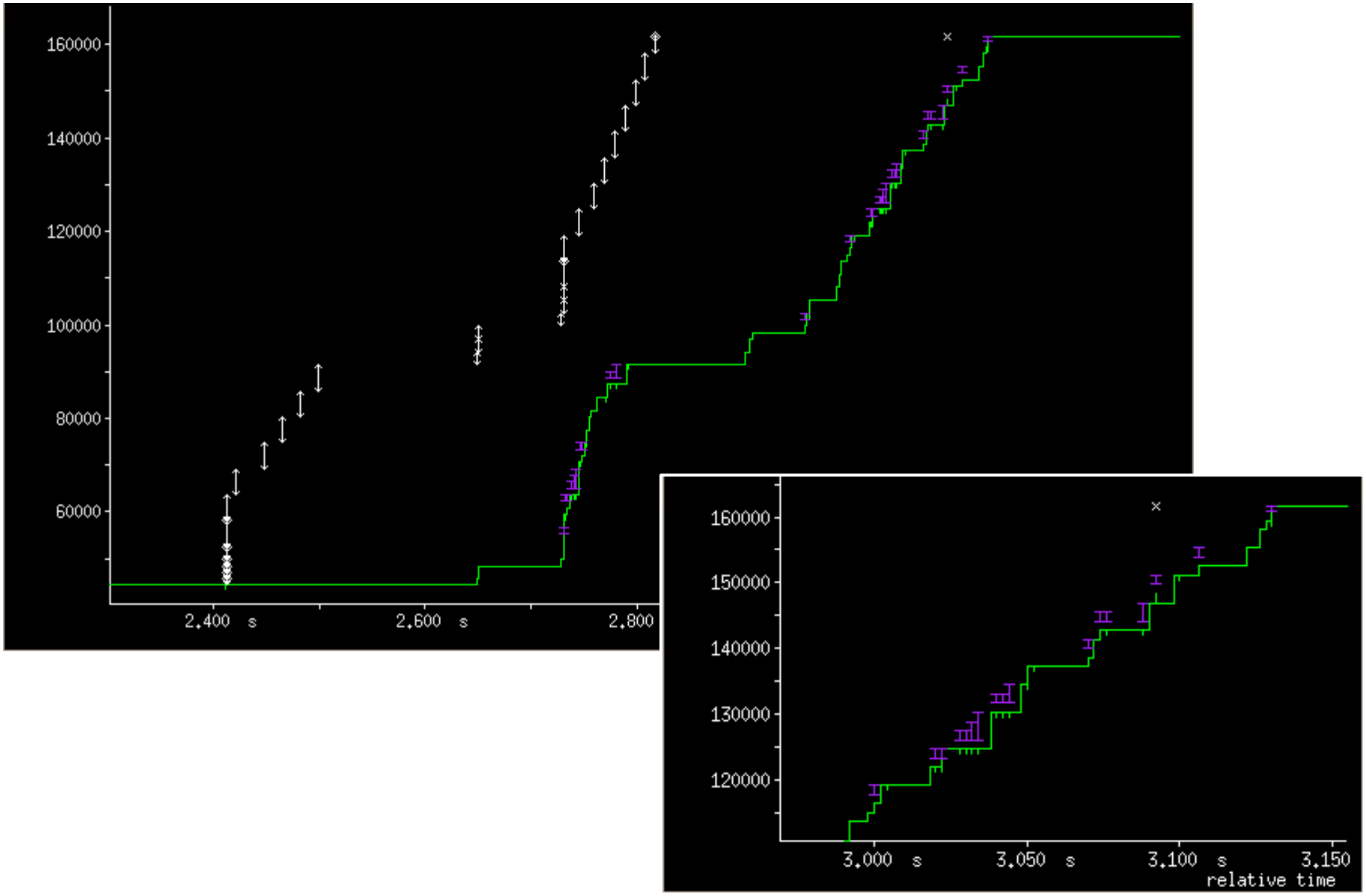
FACK: after `dupthresh`*MSS highest seq is sacked

RACK: after `reo_delay` past when some pkt sent later is delivered

Reordering is about delay

1. TE shifts entire flow to a shorter path
2. Network forwards packets on different paths or out-of-order

Low correlation to flight size


160000
140000
120000
100000
80000
60000
2.400 s
2.600 s
2.800
160000
150000
140000
130000
120000
3.000 s
3.050 s
3.100 s
3.150
relative time

Algorithm

`rack_snt`: last xmit time of the most recently (s)acked pkt

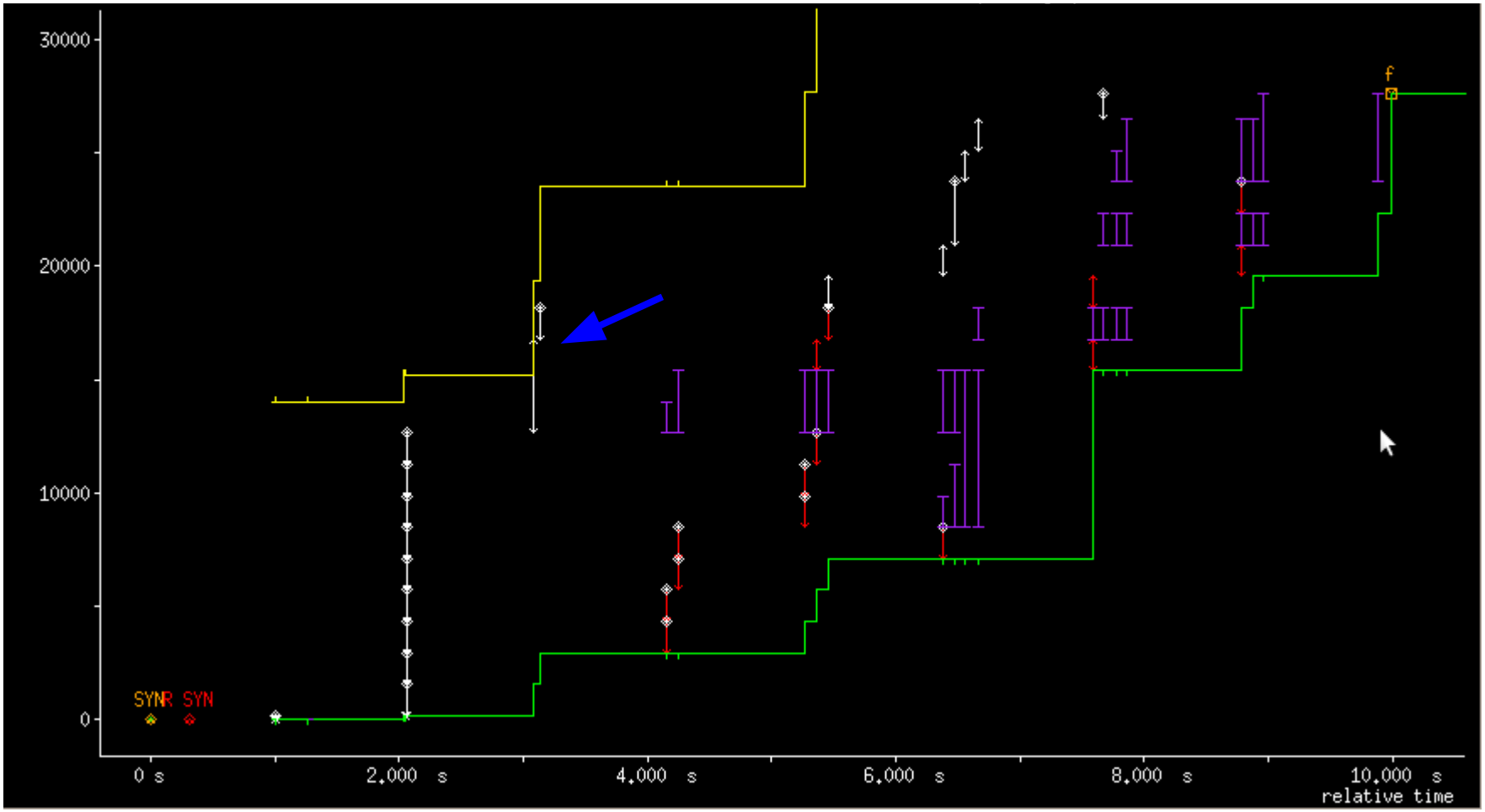
On receiving an ACK, for each pkt p:

- `p_snt`: last xmit time of p
- Mark P lost if `rack_snt - p_snt > reo_delay`

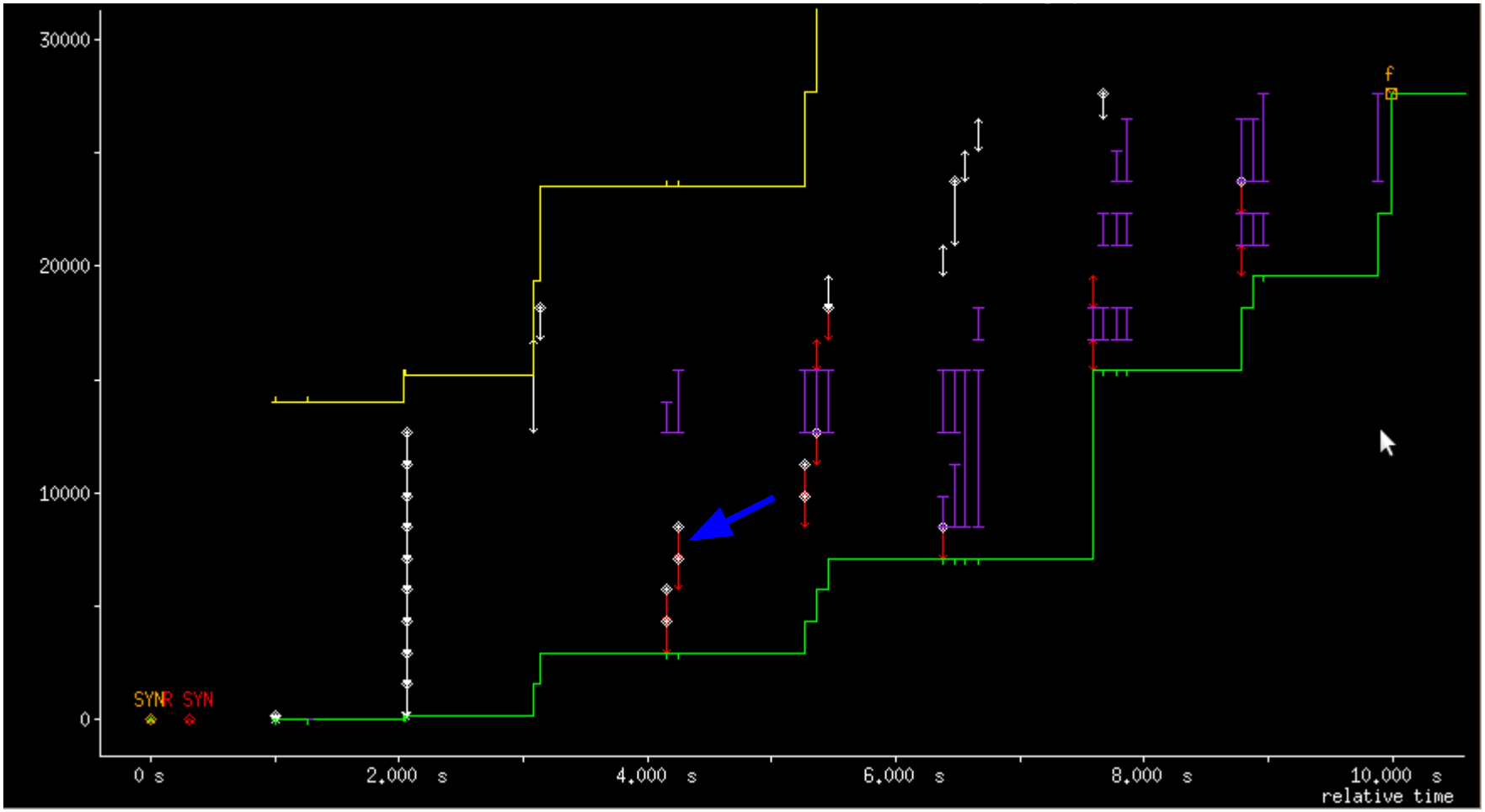
`reo_delay` starts with 0 and dynamically increase based on reordering interval

RACK works for original *and* retransmitted data

Recover tail drops (blue arrow)



Recover lost retransmits (blue arrow)



Replace all counter-based heuristics

Classic/dynamic dupthresh

RFC 3517 - SACK recovery

RFC 5827 - Early retransmit

FACK

Lost retransmit

Thin-stream dupack

Work in progress

Dynamic reo_delay

Integrate TLP draft-dukkipati-tcpm-tcp-loss-probe-01

Linux patches and IETF draft under way