draft-dukkipati-tcpm-tcp-loss-probe-00

N. Dukkipati, N. Cardwell, Y. Cheng, M. Mathis

TCPM WG @IETF 85, 6 Nov 2012.

Tail drops

TCP recovers tail drops in two ways

1. Fast: send more new data to trigger FR (limited-transmit)
2. Slow: timeout

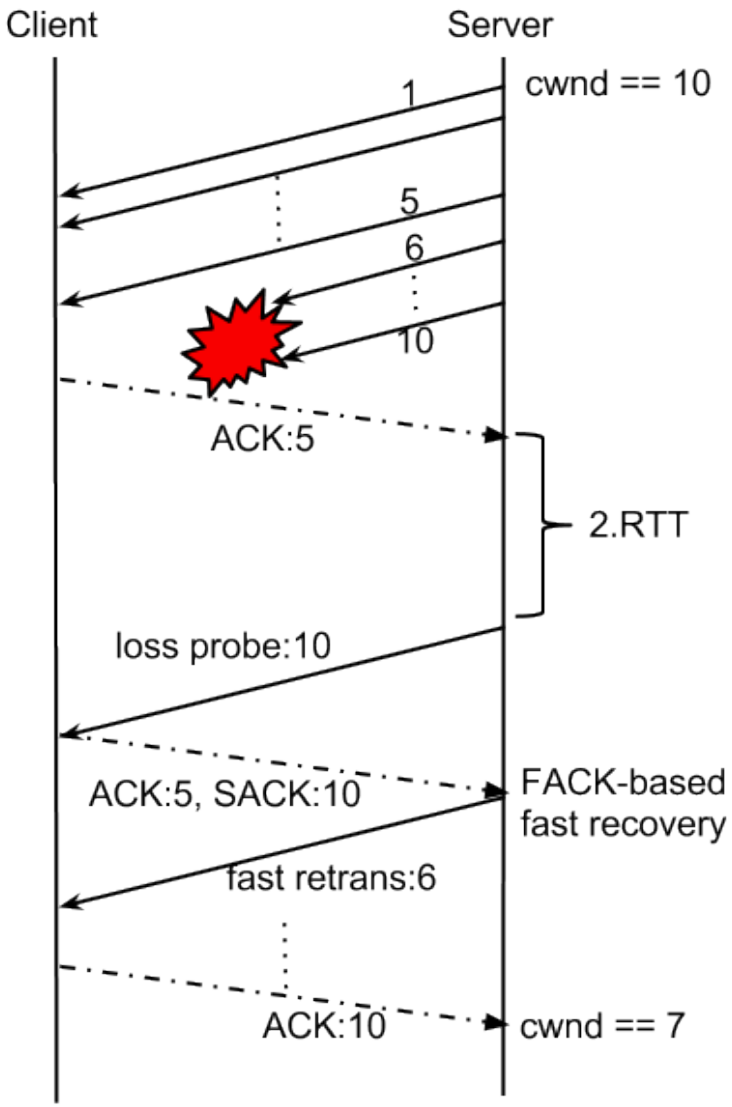
For Web traffic the situation is terrible

1. Often no new data to "probe"
2. Timeout is slow and has collateral damage
    a. RTO is not seasoned yet
    b. Retransmit & slow-start from cwnd of 1
3. Tail drops are very common
    a. 70% losses on Google.com are recovered by timeout

Idea: within 1-2 RTTs, retransmit the last packet to trigger FR

Google

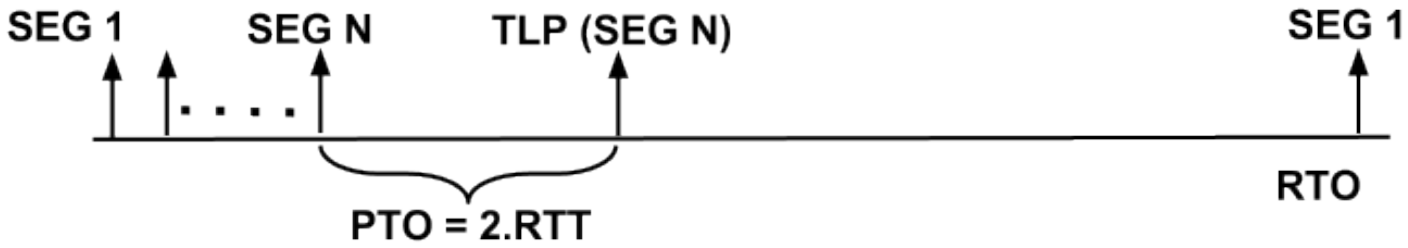
TLP example



Google

When to send TLP?

- TLP is scheduled only if PTO < RTO.



- `PTO = max(2*SRTT, 10ms);  FlightSize > 1`
  `= max(2*SRTT, 1.5*SRTT + WCDelAckT); FlightSize == 1`

- Experimenting with
  - Extend RTO to alway send TLP
  - Only send TLP if PTO < RTO - SRTT

Corner case: sender with 1 packet in flight

Won't react to the single drop repaired by TLP

Solution 1: make one packet like N>1 packets

- Retransmits only the last byte
- What if the sender only send 1 byte?

Solution 2: react to the later DUPACKs by spurious TLP

- Complex to get right

Solution 3: don't do TLP in this case

Solution 3: just ignore it

Google

ER, TLP, RTO-restart, F-RTO

	ER	TLP	RTO-restart	F-RTO
Scenarios	#dupacks < dupthresh	Tail drops	Tail drops	Timeout
Idea	Smaller dupthresh	Send last or new packet before RTO	offsetting timeout by sndbuf q delay	send new data on timeout
Pros	2RTT recovery time	3RTT recovery time	Shorter timeout	Avoid spurious timeout setting cwnd to 1
Implementation Complexity	Small - medium	Medium	Small?	Large
need SACK	no	yes. FACK.	no	no
Status	Linux default	Linux?	?	Linux default, FreeBSD

Google

WG adoption

- Work in progress
  - Experiment with different PTOs and probes
    - A parity packet (FEC)
  - Upstream to Linux
  - A research paper
  - Merge ER, F-RTO, TLP together?

- Enough interests for WG adoption?

Detecting TLP repaired losses

- Problem: congestion control not invoked if TLP repairs loss **and** the only loss is last segment.

- Approach 1: Count DUPACKs for TLP
  - TLP episode: N consecutive TLP segments for same tail loss.
  - End of TLP episode: ACK above SND.NXT.
  - No loss: sender receives N TLP dupacks before episode ends.
  - Loss: sender recvs <N TLP dupacks.

- Approach 2: Restrict TLP retransmission to 1-byte.

- We are experimenting both

Google

Relating TLP to RTO Restart draft

- TLP and RTO Restart are philosophically not coherent.
- View-point of TLP
  - Try fast recovery as far as possible, use RTO as last resort.
  - Push RTO farther away to be always able to schedule a TLP.
  - A spurious probe is less risky than a spurious RTO.
- View-point of RTO Restart
  - Make RTOs more "tight" while being RFC-compliant.
- Difference in scope
  - RTO Restart used when #outstanding segments <= 3.
  - TLP used only for SACK enabled connections.