Speeding up TCP's loss recovery

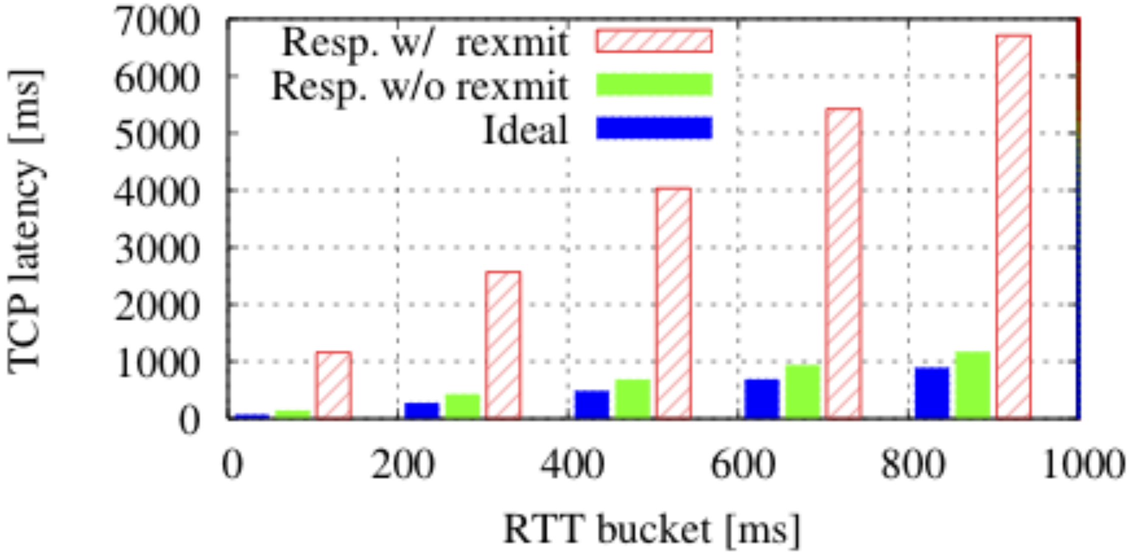
Tail loss probe (TLP)
TCP with forward error correction (TCP-FEC)

Nandita Dukkipati
nanditad@google.com

Linux Plumbers Conference, August 2012.

Motivation

Lossy responses last 10 times longer than lossless ones.



6.1% responses and 10% TCP connections experience losses.

30% losses recovered by TCP's fast recovery, 70% by timeouts.

Our contributions

- PRR: make fast recovery even faster. (Linux 3.2-rc1)
- TLP: convert timeouts to fast recoveries.
- FEC: 0-RTT loss recovery.

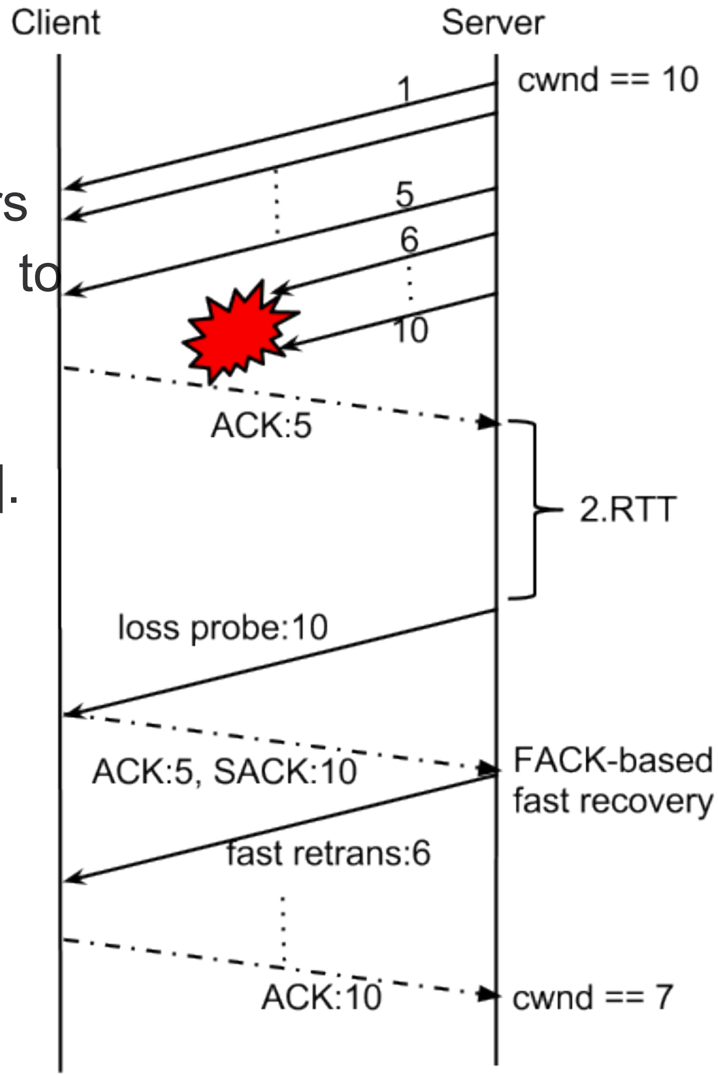
Tail loss probe (TLP)

Problem: timeouts are expensive for short transfers

- Timeout recovery is 10-100x longer compared to fast recovery
- Tail losses are the majority
  [A L *] pattern more common than [A L * S * L].

TLP key idea: convert timeouts to fast recovery

- Retransmit last segment in 2.RTT to trigger SACK information and invoke fast recovery



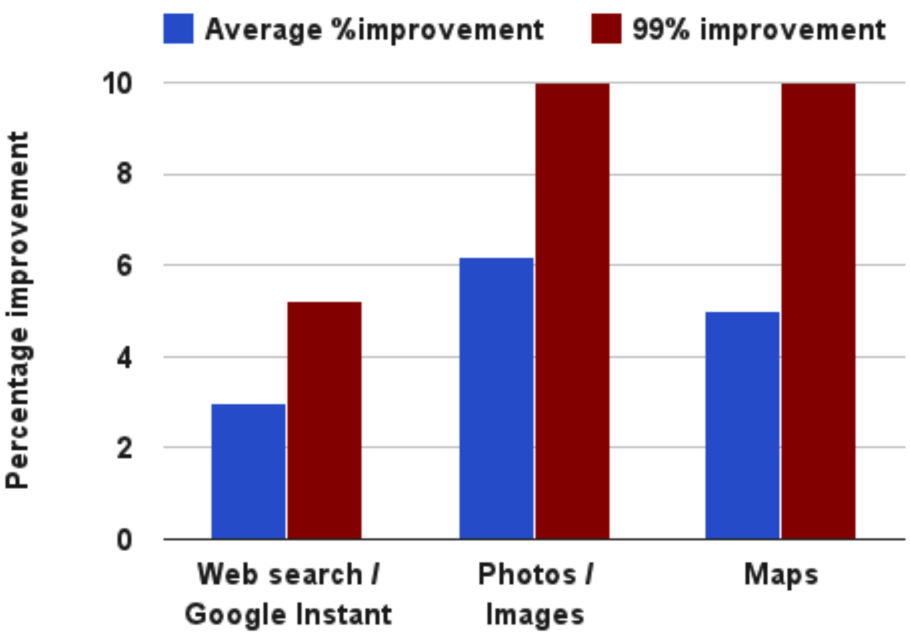
TLP example

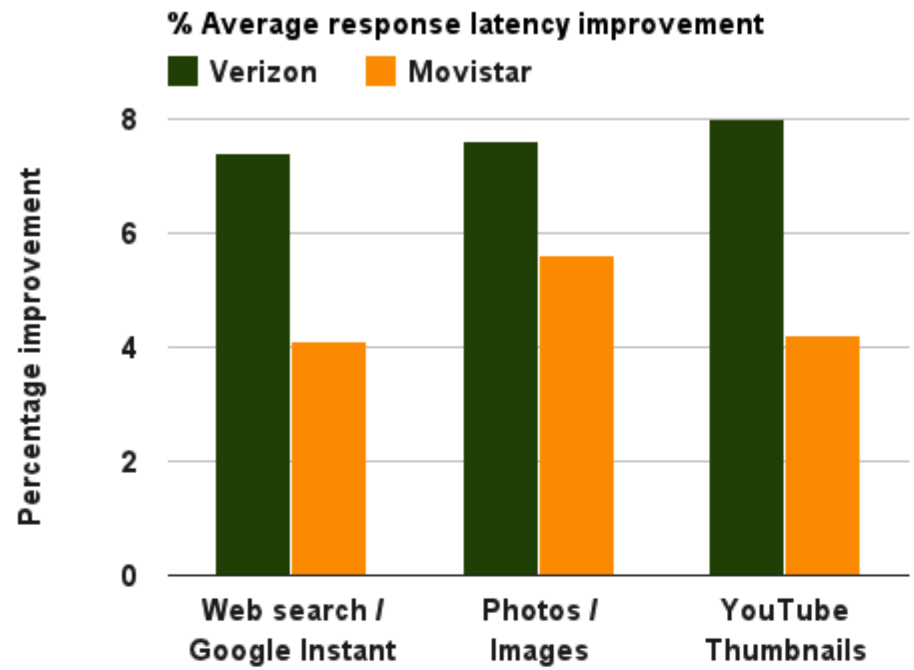
TLP pseudo code

```
Probe timeout (PTO): timer event indicating that an ACK is overdue.

Schedule probe on transmission of new data in Open state:
    -> Either cwnd limited or application limited.
    -> RTO is farther than PTO.
    -> FlightSize > 1: schedule PTO in max(2*SRTT, 10ms).
    -> FlightSize == 1: PTO is max(2*SRTT, 1.5*SRTT + WCDelAckT)

When probe timer fires:
 (a) If a new previously unsent segment exists:
     -> Transmit new segment.
     -> FlightSize += SMSS. cwnd remains unchanged.

 (b) If no new segment exists:
     -> Retransmit the last segment.

 (c) Reschedule PTO.

ACK processing:
    -> Cancel any existing PTO.
    -> Reschedule PTO relative to time at which the ACK is received
```

TLP experiments results

- 2-way experiment over 10 days: Linux baseline versus TLP.
- 6% avg. reduction in HTTP response latency for image search.
- 10% reduction in RTO retransmissions.
- 0.6% probe overhead.





Mobile only

Google

TLP properties

- Property 1: Unifies recovery regardless of loss position.
- Property 2: fast recovery of any N-degree tail loss for any sized transaction.

loss position	scoreboard after TLP ACKed	mechanism	outcome
A A A L	A A A A	TLP loss detection	All repaired
A A L L	A A L S	Early retransmit	All repaired
A L L L	A L L S	Early retransmit	All repaired
L L L L	L L L S	FACK fast recovery	All repaired
>=5 L	...L S	FACK fast recovery	All repaired

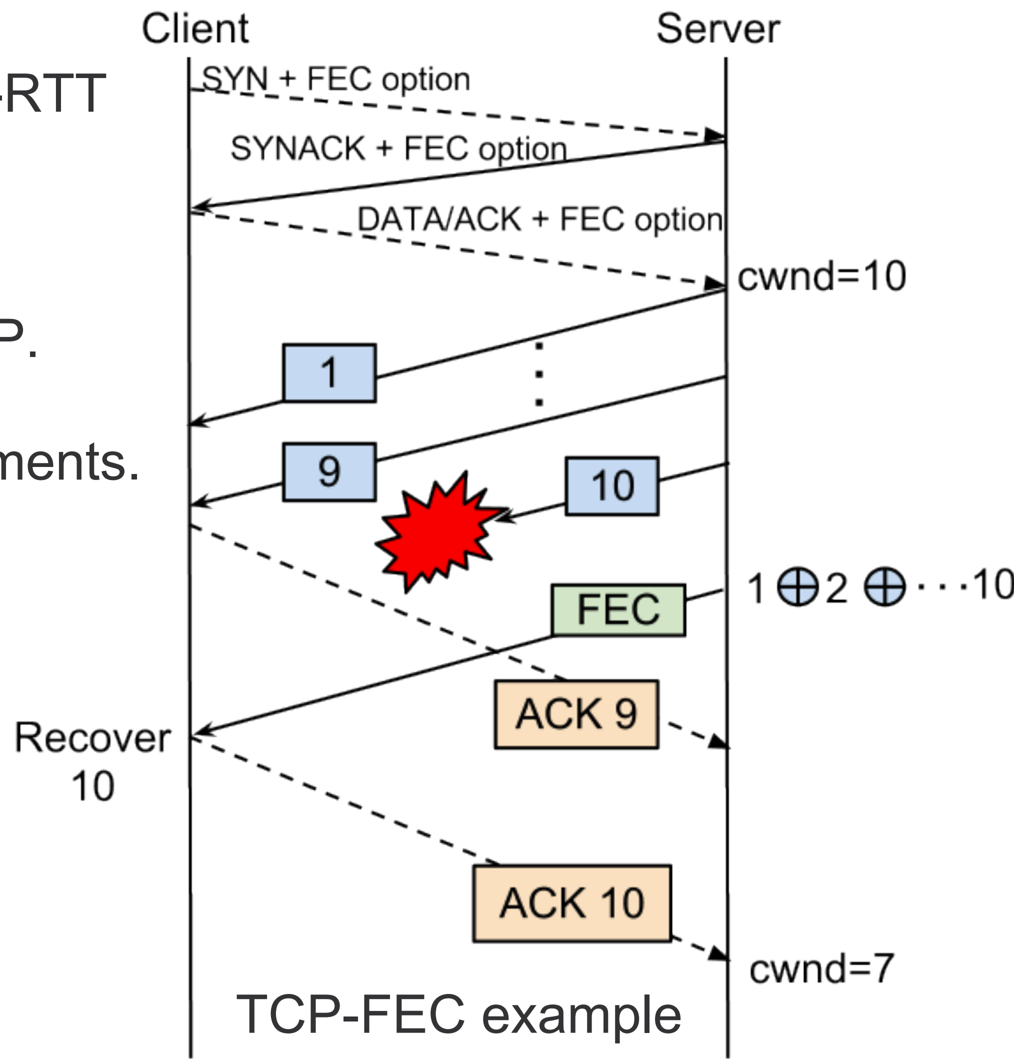
Detecting repaired losses: basic algorithm

- Problem: congestion control not invoked if TLP repairs loss **and** the only loss is last segment.
- Basic idea
  - TLP episode: N consecutive TLP segments for same tail loss.
  - End of TLP episode: ACK above SND.NXT.
  - Expect to receive N TLP dupacks before episode ends
- Algorithm is conservative: cwnd reduction can occur with no loss.
  - Delayed ACK timer.
  - ACK loss.

TCP with forward error correction (TCP-FEC)

- Goal: reduce tail latency via 0-RTT loss recovery.

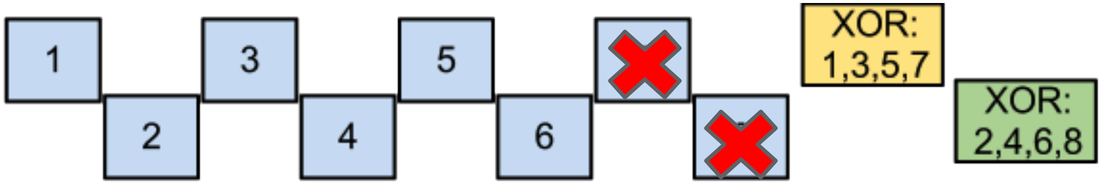
- Key design aspects
  - FEC is integrated with TCP.
  - Encoding scheme.
  - Signaling of encoded segments.
  - Congestion response.
  - Middlebox considerations.



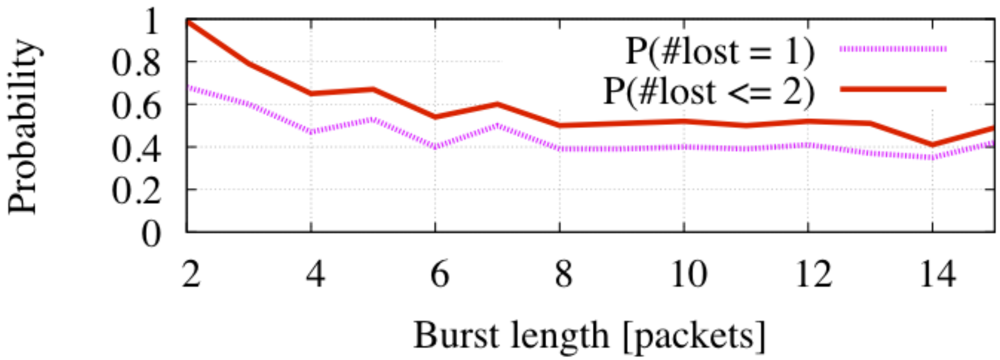
TCP-FEC example

FEC encoding approach

- Simple XOR based checksum encoding.
- Data encoded in blocks of MSS size bytes.
  - Robustness against repacketizations and variable length pkts.
- Interleaved XOR supports recovery of back-to-back losses.



- Loss stats
  - 40% losses (10-pkt burst) are single packet losses
  - Probability of atmost 2 packets lost > 0.5



Signaling FEC information

- Basic approach
  - Reuse SEQ number: FEC packet carries sequence# of first encoded byte.
  - New TCP option distinguishes FEC packet from original SEQ.

- Example FEC option

Option kind	Option length	FEC option magic	Flags	Encoding range

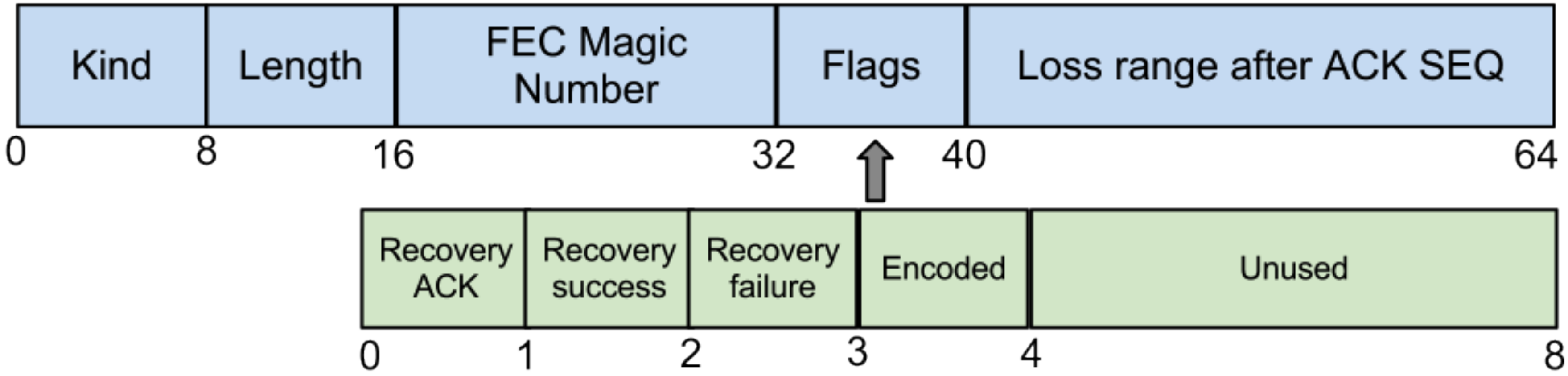
0 8 16 32 40 64

SEQ=10000; MSS=1460; Encoding range=14600
XOR range 10000 - 24600

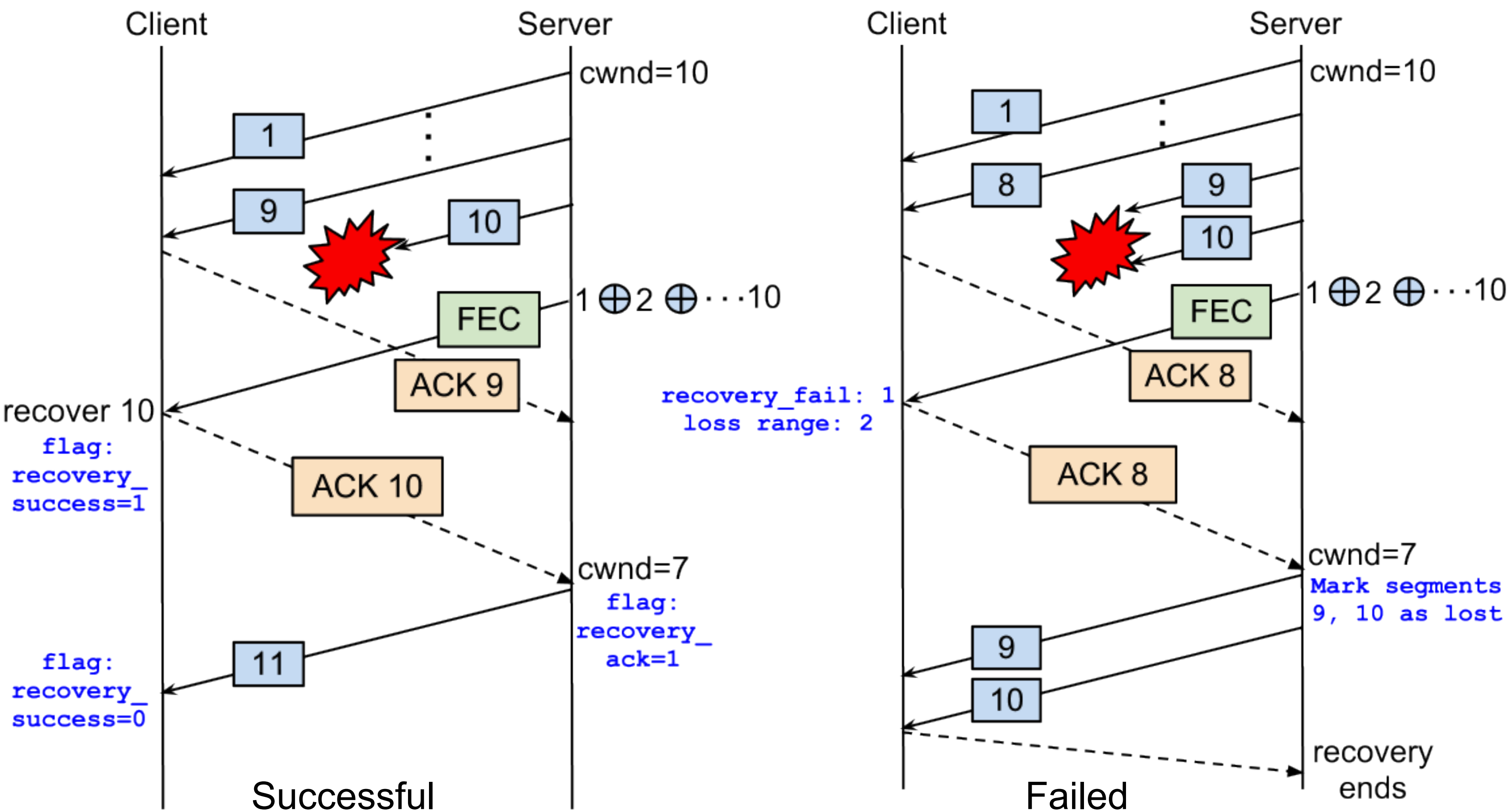
- Three flavors of options:
  - FEC negotiation in SYN/ACK handshake.
  - FEC option in FEC packets.
  - FEC option in every DATA packet.

FEC Acknowledgements

- Successful recovery
  - Treated similar to a successful fast retransmit.
  - Sender reduces congestion window like in fast recovery.
  - Loss recovery notification similar to explicit congestion notification (ECN).
- Failed recovery
  - Key: FEC packet has information on range of transmitted sequence.
  - Sender is notified of the sequence range that is lost.
  - Sender triggers fast recovery.



Examples of successful and failed recoveries



Middleboxes and alternative designs

Middlebox issue	Solution
Rewrite ISN; preserve unknown options.	Relative sequence numbers.
Removal of new TCP options.	Negotiate option in handshake; Enable option in every packet carrying data.
Rewrite ACK number to match state of middlebox.	Retransmit recovered data; suppress DSACK block in ACK.
Resegmentation (split, coalesce).	Segments with options are OK.
Buffering OOO segments.	None - no worse than today.
Normalization: rewrite payloads for previously seen sequence ranges.	(potential: Checksum FEC payload

Reference: [Is it still possible to extend TCP?]

Alternative designs

- No reuse of SEQ numbers: FEC and original have different SEQ.
- Receiver and sender maintain running checksum.

Google

What's next?

- FEC prototype ~1500 LoC.
- Experiments with FEC.
    - Impact on Web page download time.
    - FEC performance in mobile networks.
- Pursue IETF standardization.
- FEC should eventually replace TLP.
- Near term
    - TLP to net-dev.
    - TLP Internet Draft: http://tools.ietf.org/html/draft-dukkipati-tcpm-tcp-loss-probe-00