TCP Instant Recovery: Incorporating Forward Error Correction in TCP

draft-flach-tcpm-fec-00

T. Flach, N. Dukkipati, Y. Cheng, B. Raghavan

TCPM WG, IETF 87, 30 Jul 2013

Introduction

- Recover lost segments instantly without requiring retransmissions
- Motivation
  - Web transfers are short and finish within a few round-trip times (RTT)
  - Detecting packet loss can take multiple RTTs, plus at least one RTT for recovery
- Goals
  - Reduce tail latency
  - Scale loss recovery with bandwidth
  - Trade bandwidth for latency

Google

Design Rationale for TCP Instant Recovery (TCP-IR)

- No major TCP implementation supported redundant transmissions at time of design
- Design choices made based on measurements and observations of Internet loss patterns:
  - If a flow experiences loss, often only one or two consecutive packets are lost
  - Loss often happens at the tail of a burst
- We can deal with these loss patterns, even when using a simple coding scheme like XOR
- Need to be able to cope with middlebox interference

TCP-IR: The Sender Side

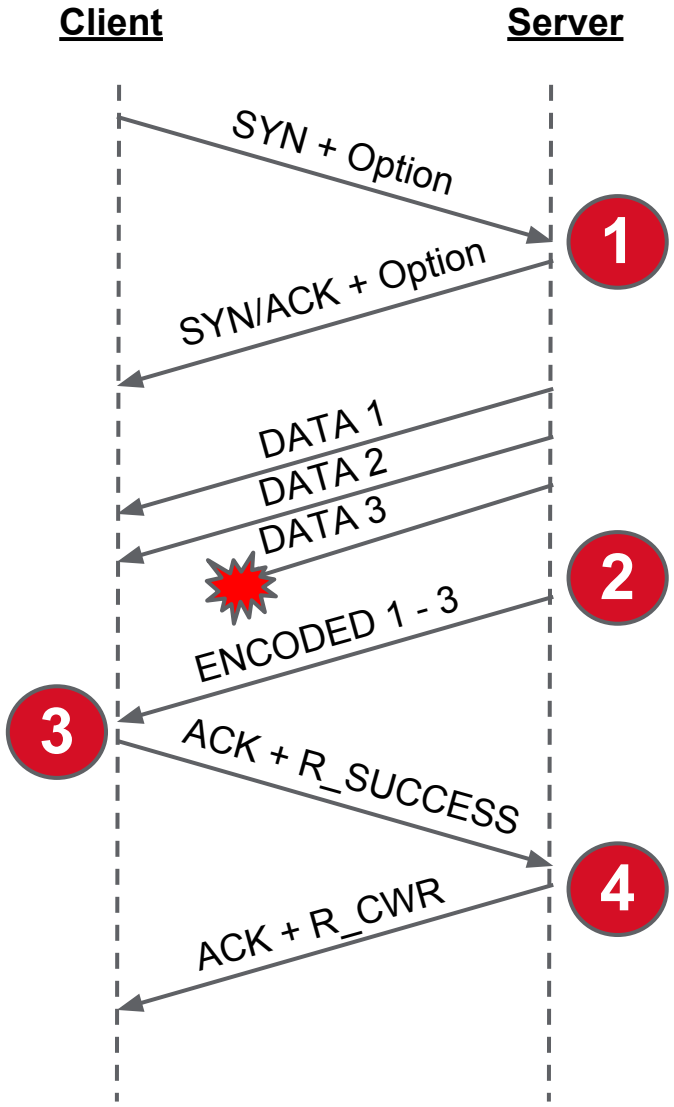
- Newly transmitted segments are encoded in a MSS-length segment
- Encoding done along MSS byte boundaries → guarantees that a receiver can instantly recover any single packet loss
- TCP-IR packet header fields:
  - Uses same sequence number as first byte it encodes
  - TCP-IR option stores encoding length, and has the ENCODED flag set to signal that the packet carries encoded payload
- No reliability is provided for TCP-IR segments
- *All packets need to carry the TCP-IR option to ensure that encoded packets can always be distinguished from regular packets*

TCP-IR: The Receiver Side

- TCP-IR packets are detected by checking the ENCODED flag in the TCP-IR option
- If a packet in the encoding range is lost, it is recovered using the TCP-IR packet and the other encoded packets buffered by the receiver
- Successful and failed recoveries are signaled to the sender:
  - Congestion control is enforced (similar to ECN)
  - Triggering retransmission of lost packets

Protocol Overview

1. Negotiation during initial handshake
2. Delayed transmission of encoded packets
3. Signaling of successful and failed recoveries
4. Congestion window reduction upon successful recovery



Performance Results

- Prototype experiments in local testbed using netem, dummynet, and Web page replay to emulate file transfers and web page downloads
- Short transfer latency reduced by 28% in the 90th percentile
- Web page downloads take 15% less time in the 90th percentile
- Performance loss for:
  - Long transfers (existing mechanisms are suitable enough to deal with loss)
  - Loss-free transfers (wasting part of the congestion window to transmitting redundant segments)

See: "Reducing Web Latency: the Virtue of Gentle Aggression" (SIGCOMM 2013)

Discussion

- Applicability of this scheme across the Internet
  - Mice vs. elephant flows
- Accounting for redundant segments by the congestion window
- Middlebox issues we need to work around?
  - Selectively stripping TCP options
  - Rewriting ACK numbers (due to sequence holes)
  - Rewriting payloads (e.g. FTP headers)
  - Packet coalescing / splitting
  - ...

Pointers

- "Reducing Web Latency: the Virtue of Gentle Aggression" (SIGCOMM '13, to appear):
  http://research.google.com/pubs/pub41217.html

- Prototype implementation:
  Will be published soon at code.google.com

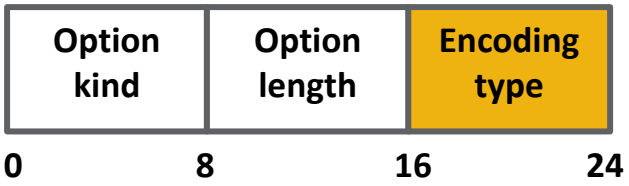
- This presentation (feel free to add comments):
  http://goo.gl/4hgFOV

Additional Slides

Negotiation

- Exchange of TCP-IR option during initial handshake
- TCP-IR is only enabled, if:
  - SYN carries the short TCP-IR option specifying the requested encoding type
  - SYN/ACK responds with same TCP-IR option value
  - Every future packet carries the TCP-IR option

- Option specification:

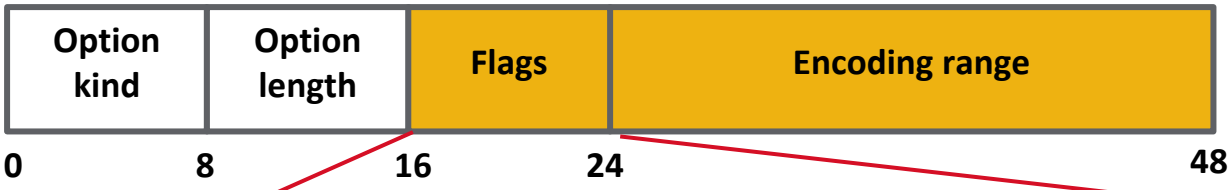


Prototype currently supports regular and interleaved XOR encoding.

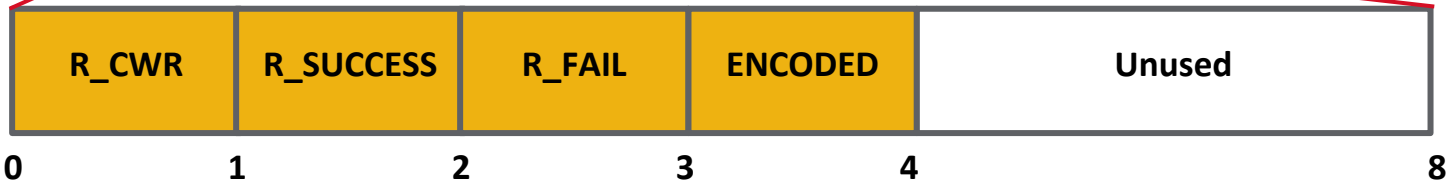
Encoding and sending data

- TCP-IR packet(s) generated after data in the write queue is transmitted
  - All previously unencoded data is encoded in a block of MSS bytes
  - Each TCP-IR packet uses the sequence number of the first encoded byte
  - TCP-IR option signals the encoding range (allows to calculate the sequence number of the last encoded byte) and carries an ENCODED flag
- Transmission delayed to reduce drop probability

- Option specification:

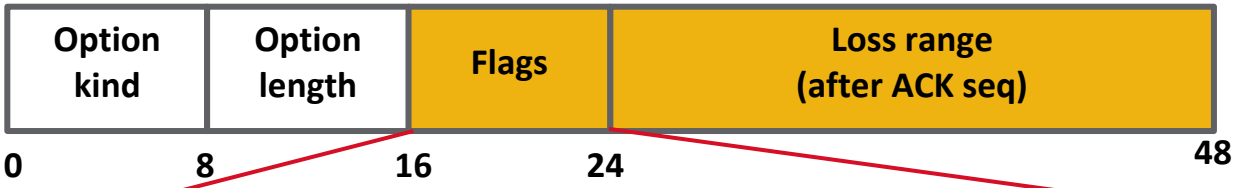


- Flags:

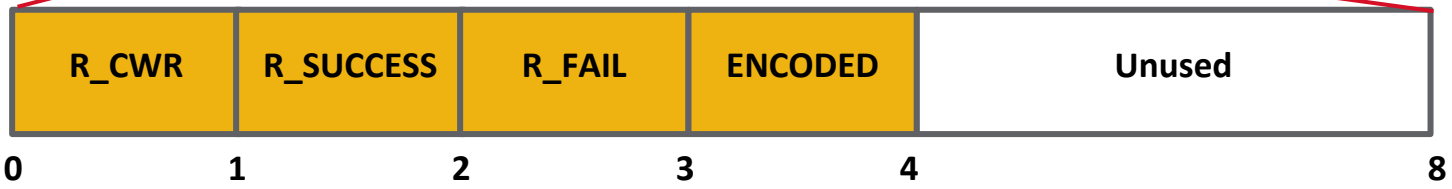


Acknowledgements

- Successful recovery
  - Treated similar to a successful fast retransmit
  - Sender should reduce congestion window and continue in avoidance mode
  - Notification similar to explicit congestion notification (ECN)
- Failed recovery
  - TCP-IR packet already carries information about transmitted sequences
  - Notification of the sender which sequence ranges were lost
  - Sender can mark the corresponding buffers
- Option specification:



- Flags:



Google

Interaction with Middleboxes

Issue	(Possible) Solution
Sequence number translation	Relative sequence numbers
Removal of non-standard TCP options	Negotiate option usage and enable option for every packet carrying data
Buffering of out-of-order packets	None (Fast recovery + TCP-IR without impact)
Rewriting acknowledgements number to match state of middlebox	Retransmit recovered data and suppress DSACK block in the acknowledgement
Rewriting payloads for previously seen sequence ranges	Checksum TCP-IR payload
Packet coalescing / splitting	Signal sequence range with encoded payload

Google

Instant Recovery (IR) on different layers

- Application layer
  - Applications can selectively protect important parts of data
  - If reliable transport layer protocol is used, redundant packets are recovered
  - Does not know which packets are prone to losses
- Transport layer
  - Has necessary data to configure and tune IR (e.g. data about packets with higher loss probability, congestion window size, loss rate, RTT, ...)
  - Additional protocol complexity (e.g. single sequence number space, packet tampering by middleboxes, ...)
- Network layer
  - Can provide protection across multiple higher-layer connections
  - IR data can be transmitted out-of-band
  - Requires additional buffering of higher-layer packets