A Test To Allow TCP Senders to Identify Receiver Non-Compliance

Toby Moncaster†, **Bob Briscoe***, Arnaud Jacquet*

† University of Cambridge * BT

draft-moncaster-tcpm-rcv-cheat-03

R/TE

Reducing Internet Transport Latency

Bob Briscoe's work was part-funded by the European Community under its Seventh Framework Programme through the Reducing Internet Transport Latency (RITE) project ICT-317700

Background and why come to IETF?

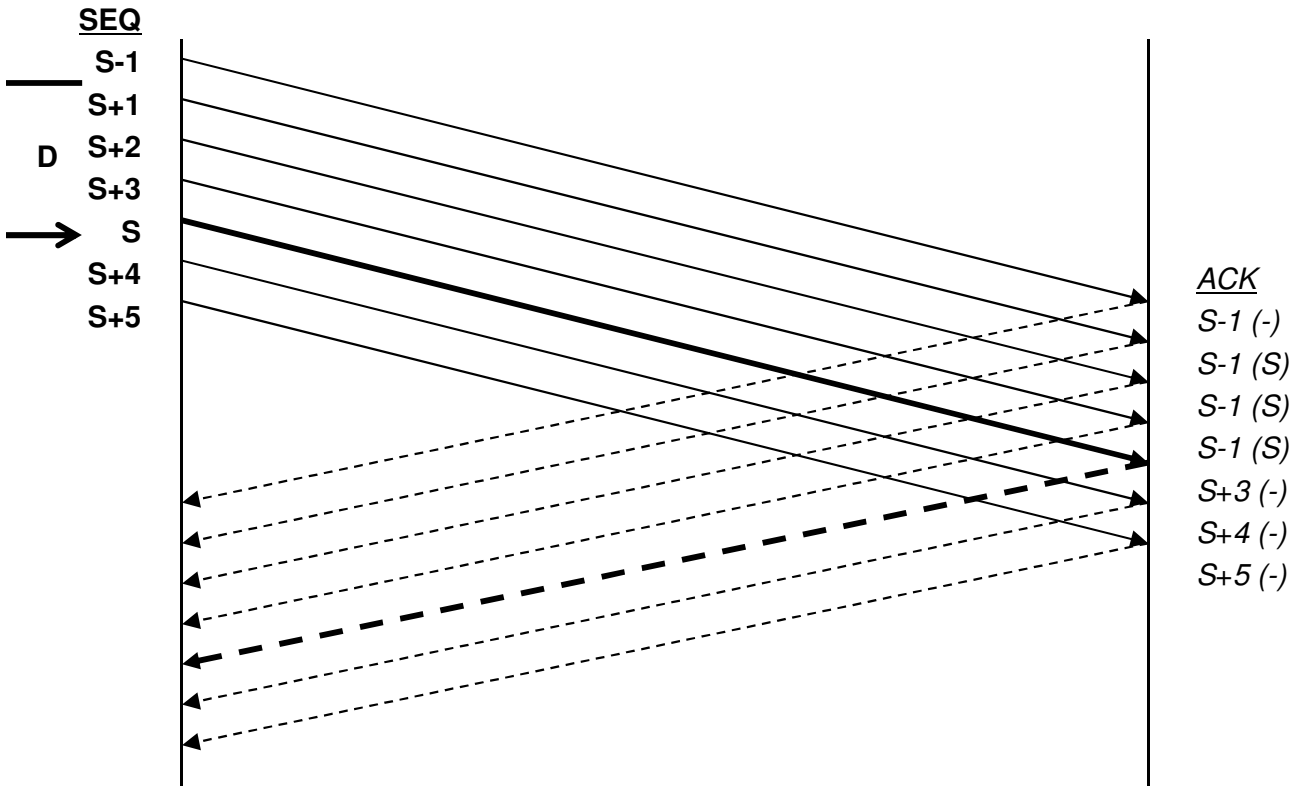
- TCP senders rely on accurate congestion feedback from receivers
- A strategising loss-insensitive, e.g. streaming, receiver can:
  - Send **optimistic acks**: fool a sender into increasing its rate
  - **Conceal** lost data segments: to avoid a congestion response
- These disrupt sharing of sender's own resources & fool sender into using excess network resources by concealing congestion

Solution is a simple test – draft-moncaster-tcpm-rcv-cheat

- Originally written in 2007
  - parked for 6 years due to lack of interest/time
- New draft (-03) published July 2014
  - Want to get community oversight of the proposed tests
  - Motivated by new interest in this area from large content providers
  - Also could free-up ECT(1) codepoint assigned to ECN Nonce expt
  - Have chosen to go to experimental track

Proposed solution

- Based on Sherwood's Randomly Skipped Segments, but add initial stage to reduce impact on receivers [1]
- Key goal was to leverage existing TCP behaviour
  1. Delay a segment by a small amount and check the responses



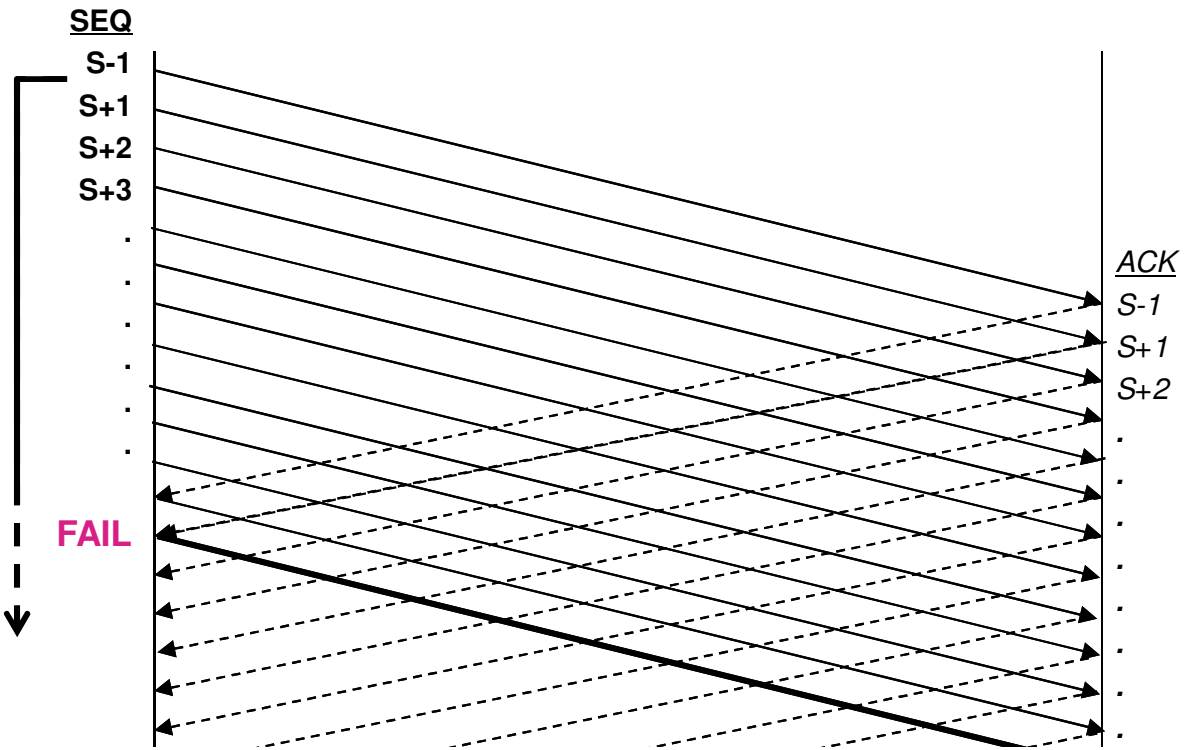
- select segment S & displacement D *(2 < D < K-2, K = current window)**
- Segment S transmitted after segment S+D
- Receiver should generate D dup acks

** SACK allows longer gap but shorter reduces harm to receiver*

[1] K. Gao student project 2004: http://www.cs.cmu.edu/~kgao/course/15744/network.pdf

Proposed solution

- Based on Sherwood's Randomly Skipped Segments, but we add initial stage to reduce impact on receivers
- Key goal was to leverage existing TCP behaviour
  1. Delay a segment by a small amount and check the responses
  2. Delay a segment until you have seen 3 dup-ACKs



- Hold back segment S* Transmit segment S after seeing 3 Dup ACKs for S-1
- If you see any ACK for a segment >S then receiver has failed

** perform pessimistic cong. response if no SACK*

Aims of the experiment

- Primary aims of experiment:
  - Establish whether this two-stage testing strategy works
  - Establish whether they cause any harm to other flows, and minimal harm to compliant receivers
- Secondary aim:
  - see if any receivers use these strategies
- Hope to get content providers to run these tests
  - Already identified one possible provider
  - Would love to hear from others
- Experiment would run for one year from adoption

Summary

- Strategising receivers skew a sender's allocation of its own resources
- Receivers can fool the sender into giving excess network resources to the receiver, risking v high congestion
- This 2-stage test allows senders to identify such receivers
  - ♦ It causes minimal problems for honest receivers
  - ♦ It encourages compliant behaviour:
    - The only way to avoid the first test carries a delay penalty
    - Second test can't be avoided
- Only requires modification of sender, so easy to implement

Next steps

- Get draft adopted by WG
- Considering extension to cover concealing ECN marks:
  - Easy to do if using AccECN ‘cos an introduced CE mark won’t conceal any real congestion for a whole round-trip
  - Harder with RFC3168 ECN, but possible (send a CE mark followed soon after by setting CWR)
  - Minimal impact on any actual congestion signal
- Large content provider interested in deploying this
  - Easier to convince it is safe with expert tcpm review
  - Implementation will be open source (initially FreeBSD)

Questions?

Existing solutions

- Randomly skip segments – hold back a segment and transmit it once a duplicate acknowledgement is received
- Delayed Segments – from Kun Gao, CMU student.
- A transport layer nonce – add a specific nonce to the transport layer which has to be echoed back to the sender
- The ECN nonce* –receiver has to guess correct nonce sum (NS) [RFC3540] for any lost segment or optimistic ack

**experimental and only for ECN-capable transports*

	Skip Seg	Delay seg	ECN nonce	TCP nonce
Doesn't interfere with CC	✓	✓	✓	✓
Utilise existing features	✓	✓	x	x
Receiver plays passive role	✓	✓	x	x
No negotiable TCP options	x	✓	x	x
Receiver unaware of testing	✓	✓	n/a	n/a
Able to prove non-compliance?	✓	x	✓	✓
Doesn't harm compliant rcvr	x	✓	✓	✓