

HotRFC – IETF 106 Singapore

Explicit measurements: Round Trip Packet Loss

draft-cfb-tsvwg-spinbit-new-measurements

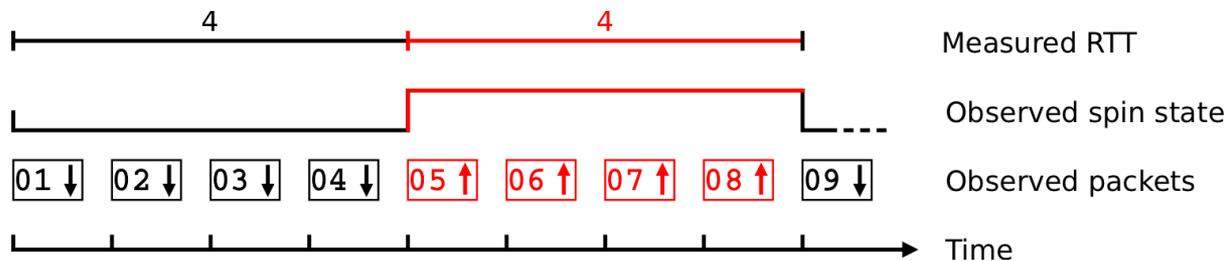
Singapore, November 2019, IETF 106 – TSVWG

Mauro COCIGLIO (Telecom Italia - TIM)
Fabio BULGARELLA (Telecom Italia - TIM)
Giuseppe FIOCCOLA (Huawei)
Riccardo SISTO (Politecnico di Torino)



Explicit in-band measurements: Spin Bit

- ▶ Spin bit for RTT measurement was the first case of explicit in-band measurement.
- ▶ It's implemented in QUIC protocol (<https://www.ietfjournal.org/enabling-internet-measurement-with-the-quic-spin-bit/>)
- ▶ The spinbit idea is to create a square wave signal on the data flow, using a bit, whose length is equal to RTT.
- ▶ An observer in the middle (wherever is located) can measure the end-to-end RTT only watching the spinbit.



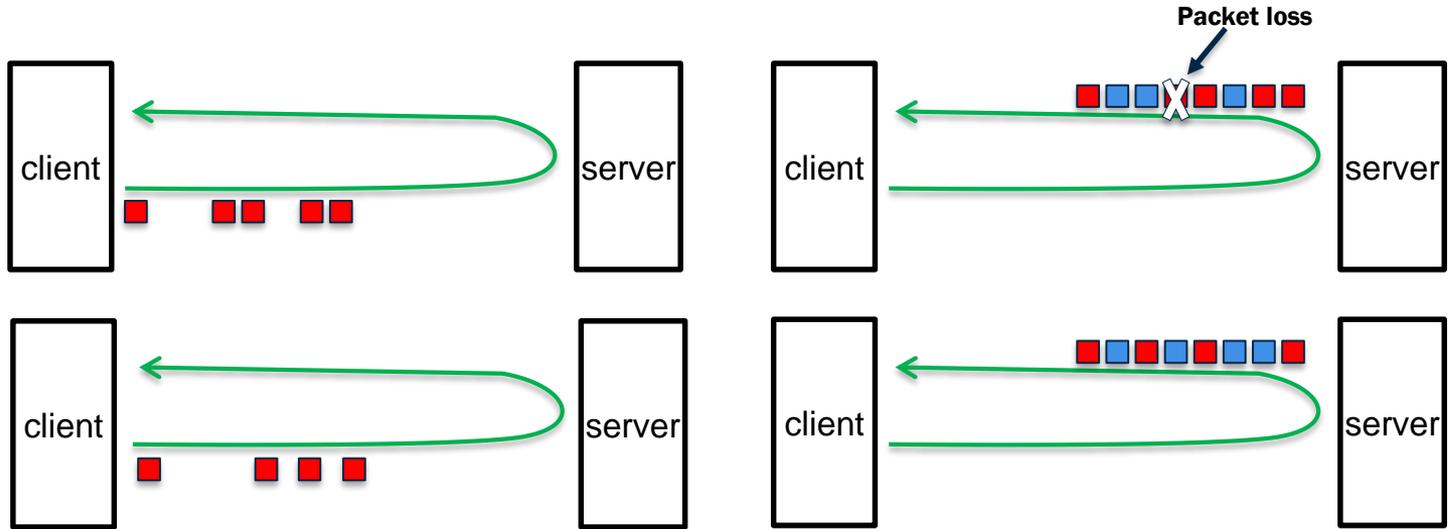
RoundTrip Packet Loss (PL bit)

- ▶ **A new performance metric, the RoundTrip Packet Loss**
 - ▶ Measured on production traffic between Client and Server.
- ▶ **How it works:**
 - ▶ The Client marks «a train» of production packets (using the PL bit) and these marked packets «bounces» between Client and Server to complete 2 rounds.
 - ▶ Client and Server «reflects» marked packets by marking production packets flowing in the opposite direction.
 - ▶ An Observer counts the marked packets during the 2 rounds and compares numbers to find losses.
- ▶ **The main issue:** Upload and Download usually have different packet rates.
 - ▶ QUESTION: How many packets to mark to avoid marked packets congestion on the slowest traffic direction?
 - ▶ ANSWER: the number of packets that transit, in the marking period, on the slowest direction (it's implemented using a token system).

Roundtrip Packet Loss: how it works (1)

- ▶ The Client generate a train of market packets (using the Packet Loss bit)
- ▶ The Server «reflects» these packets (marking production packets flowing in the opposite direction). The Server inserts some not marked packets if download flow has more packets than upload flow.
- ▶ The Client reflects the marked packets.
- ▶ The Server again reflects the marked packets
- ▶ The Client generate a new train of market packets and so on.

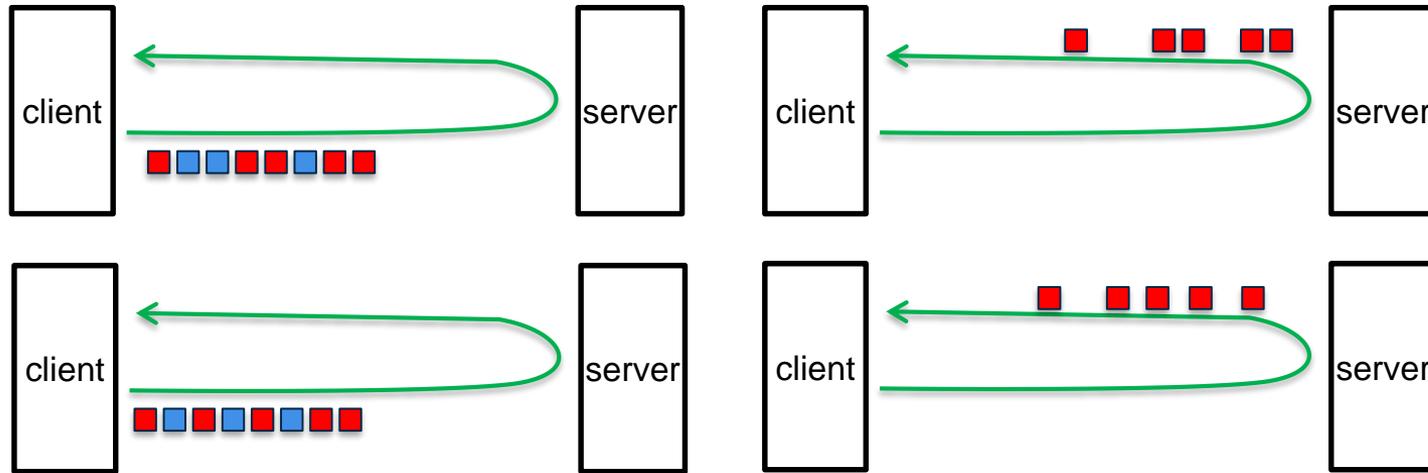
Download flow has more packets



Marked packets: red,
Not Market packets: blue

Roundtrip Packet Loss: how it works (2)

- ▶ When upload flow has more packets than download flow we use a token system to maintain the same marked packets rate on both directions (upload and download):

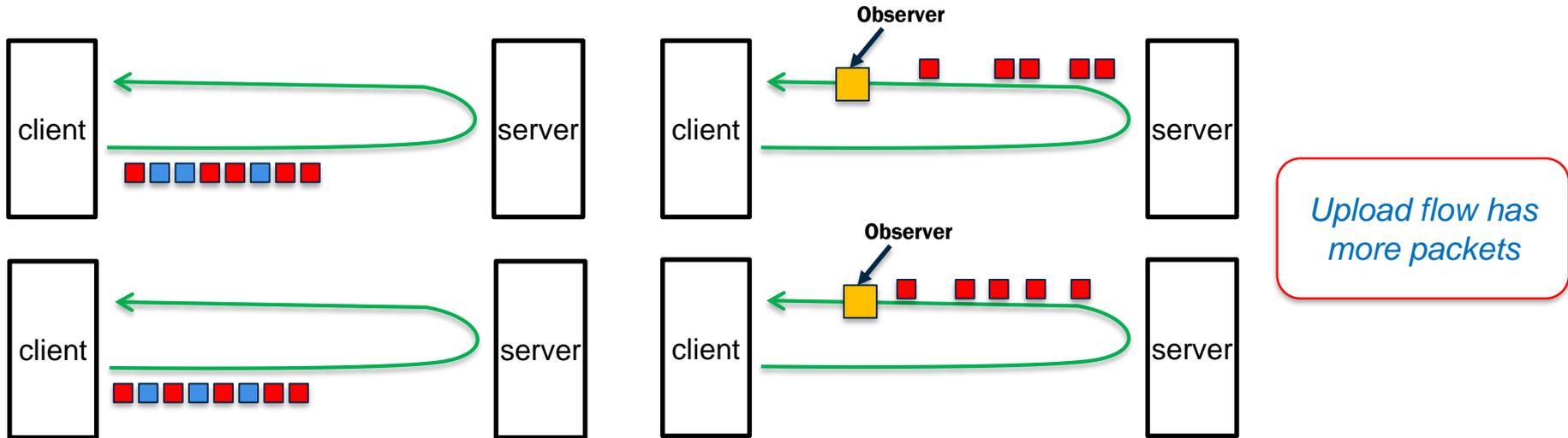


Upload flow has more packets

Marked packets: red, Not Marked packets: blue

Roundtrip Packet Loss: the Observer

- ▶ The Observer in the middle (upstream or downstream) sees the packet train twice and so it calculates the «observer roundtrip packet loss» that, statistically, will be equal to the «end-to-end roundtrip packet loss».



Marked packets: **red**, Not Marked packets: **blue**

How to know more

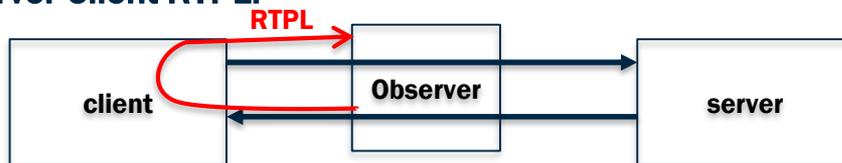
- ▶ **Hackathon project:** “QUIC Measurements & SpinDump”
- ▶ **Draft:** <https://tools.ietf.org/html/draft-cfb-tsvwg-spinbit-new-measurements-00>
- ▶ **TSVWG meeting:** Thursday (10:00-12:00, Canning) - draft presentation
- ▶ **Hackdemo Happy Hour:** Monday (18:10-19:40, Moor/Morrison) - SpinDump demo
- ▶ **TSVWG mailing list:** tsvwg@ietf.org
- ▶ **Mail to:**
 - ▶ mauro.cociglio@telecomitalia.it
 - ▶ fabio.bulgarella@guest.telecomitalia.it
 - ▶ giuseppe.fioccola@huawei.com
 - ▶ riccardo.sisto@polito.it

Thank you

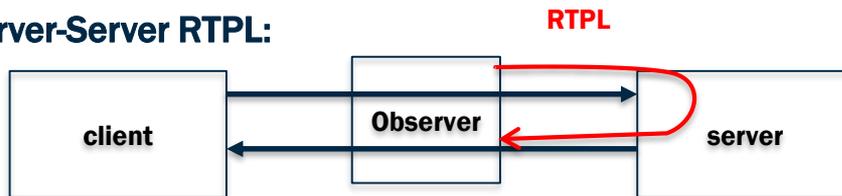
2Point RoundTrip Packet Loss properties (1 observer)



▶ **Observer-Client RTPL:**

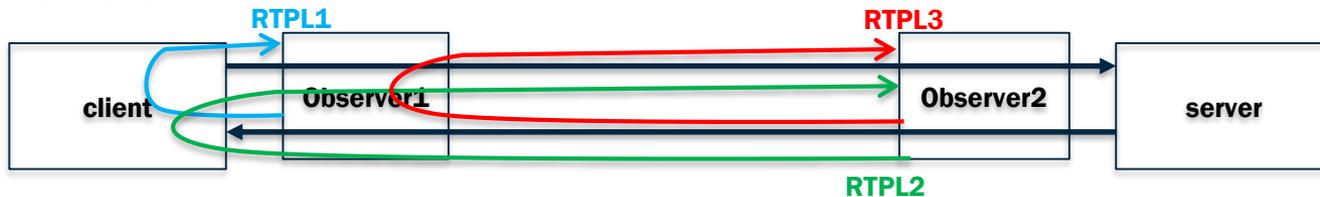


▶ **Observer-Server RTPL:**



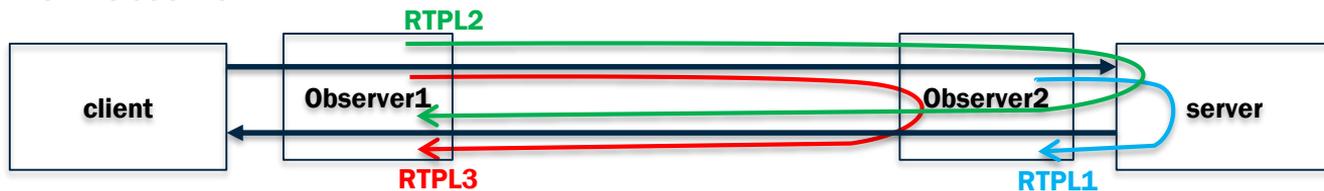
2Point RoundTrip Packet Loss properties (2 observers)

- ▶ Observer2-Observer1 RTPL:



Observer2-Observer1 Round-Trip: $RTPL2 - RTPL1 = RTPL3$

- ▶ Observer1-Observer2 RTPL:



Observer1-Observer2 Round-Trip: $RTPL2 - RTPL1 = RTPL3$