Explicit Flow Measurements

draft-mdt-ippm-explicit-flow-measurements

July 28, 2021, IPPM WG – IETF 111

M. Cociglio - Telecom Italia (TIM)

A. Ferrieux - Orange Lab

G. Fioccola - Huawei Technologies

I. Lubashev - Akamai Technologies

F. Bulgarella - Telecom Italia (TIM)

I. Hamchaoui - Orange Labs

M. Nilo - Telecom Italia (TIM)

R. Sisto - Politecnico di Torino

D. Tikhonov - LiteSpeed Technologies

Explicit Flow Measurements (EFM)

- Explicit Flow Measurement techniques employ few marking bits, inside the header of each packet, for loss and delay measurement (protocol independent and valuable for encrypted header protocols: e.g. QUIC)

- EPM metrics described in this draft:
  - RTT: Delay bit (D-bit) (with «the hidden RTT» option: D^-bit)
  - Round Trip Packet Loss: Spin bit (S-bit) + roundTrip loss bit (T-bit)
  - One Way Packet Loss, 2 options:
    1) sQuare bit (Q-bit) + Loss event bit (L-bit)
    2) sQuare bit (Q-bit) + Reflection square bit (R-bit)

IETF Hackathon and Implementations

- Some of the methodologies are already included in ongoing experiments and implementations:
- “QUIC Measurements” project during the last IETF 111
- EFM Implementations in production network reported by the contributors:
  - *Telecom Italia-TIM Implementation => android mobile phones probe.*
  - *Ericsson implementation => core network probes.*
  - *Orange-Akamai implementation => Akamai production CDNs and core network probes.*
  - *Aachen University implementation: ANRW paper (Packet Loss measurements: L, Q, R, T bits).*
  - *Huawei is working on the topic.*

Draft Updates

- **Q-bit and R-bit improved burst loss resiliency. Now there is the correct detection of burst losses up to two Q blocks (before up to one Q block).**
- **New option in the D-bit implementation: D^-bit (hidden Delay bit).**

The Delay bit

The Delay bit is a single bit RTT measurement (like the Spin bit).

How Delay bit works

The marking bit is the Delay bit (D-bit) and a packet with D-bit=1 is a Delay bit Sample (DbS).

- The idea is to have a single marked packet, the DBS, generated by the Client, that bounces between Client and Server, using the production traffic.
- When the Client doesn't detect the DbS for more then a specific time, Tmax (**greater than maximum RTT: e.g. 1000 ms.**), the DbS is declared lost; so the Client regenerates the DbS.
- Client and Server don't reflect the DbS if the reflection time is more than **1 ms.** (the **application delay threshold:** «E»).
- Observer:
  - It knows Tmax
  - Every two consecutive DbS it calculates RTT:

    «Valid RTT» Rule => *RTT< Tmax* (in practice we use RTT< 90% Tmax)

The Delay bit «Hidden RTT» version (D^-bit)

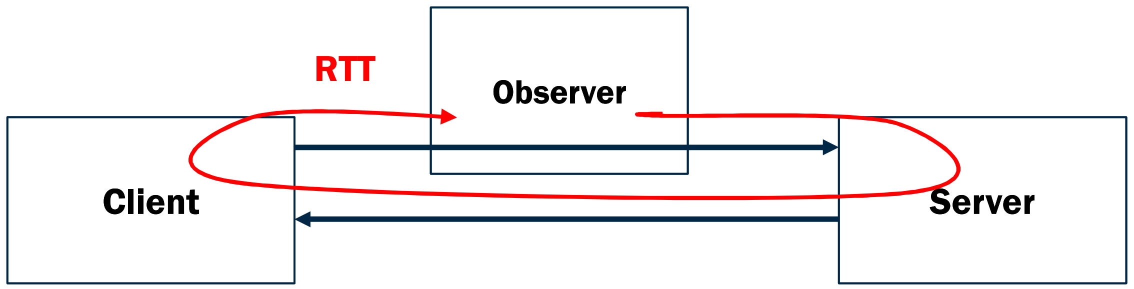
Spin Bit is standardized (QUIC) but often non implemented for privacy reasons, because the RTT value could approximate the Client/Server distance.

There is a simple way to hide the RTT and to allow only Observer/Server RTT measurements or Inter domain RTT measurements.

The idea is to slighty modify the Delay bit implementation adding a fixed amount of millisecond to the RTT measurements.

In practice the Client does not reflect immediatly the delay sample but wait for an Additional Delay before reflecting the marking.

Client-Server RTT:



End-to-End Round-Trip Time => RTT = Ts(DbS_2) – Ts(DbS_1) – AD

Ts: Timestamp

DbS: Delay bit Sample

AD: Additional Delay

AD: the Additional Delay

Main features:

- It’s a system parameter of the Client
- The value is equal for all applications and sessions

Options:

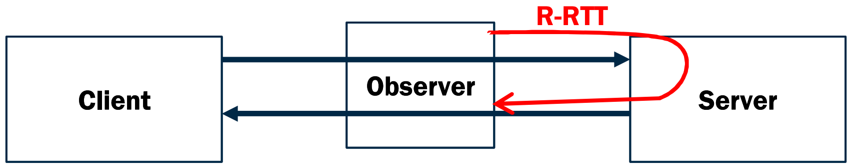
- AD value is set when a Client is switched on
- Random choice in a range of values
- Could change when the IP address change

Suggestions:

- The AD choice could be implementation dependent (not standardized)
- An Observer does not know if it is a D-bit or a D^-bit implementation.

2 direction Observer: right RTT

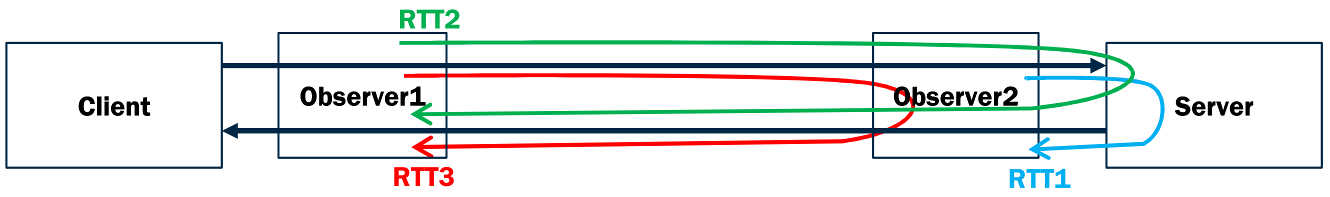
- Right RTT (RTT Observer-Server):



Max Error = +E (e.g. E=1 ms)

2-Point measurement: intra-domain RTT

- Observer1-Observer2 RTT:



Observer1-Observer2 RTT: RTT2 - RTT1 = RTT3

Max Error = ±E (E-0=+E; 0-E=-E) when measured with different marked packets

Max Error = 0 (E-E=0) when measured with the same marked packets

“Delay Bits” Summary

	Bits	Unidirectional Observer	Bidirectional Observer	# of Measurements	Impairment resiliency
S: Spin Bit	1	RTT	x2 Half-RTT	Very High	Low
D: Delay bit	1	RTT	x2 Half-RTT	Medium °	High
D^: Hidden Delay bit	1	RTT^	x2 Left Half-RTT^ Right Half-RTT	High ~	High
SD: Spin bit + Delay bit *	2	RTT	x2 Half-RTT	Very High	High

° It depends on the “application delay” threshold (e.g. E=1 ms.), causing DbS discarded, and on DbS losses. But many of these missing measurements are “errored” measurements.

~ The “application delay” threshold (e.g. E=1 ms.) is only on the Server (see previous note).

* Both algorithms work independently; an observer could use approximate spin bit measures when delay bit ones aren't available.

X2 Same metric for both directions.

^ Masked metric (real value can be calculated only by those who know the Additional Delay).

“Loss Bits” Summary

Method	Bits	Unidirectional Observer	Bidirectional Observer	Proto	Measurement Fidelity	Measurement Delay
T round Trip loss bit	1+spin	Round Trip	Round Trip Half-RT x2	*	Rate by sampling $\frac{1}{3}$ to $\frac{1}{3*ppa}$ packets over 2 RTT	~6 RTT
Q sQuare bit	1	Upstream	Upstream x2	*	Rate over N packets (e.g. N=64)	N packets (e.g. B-64)
L Loss event bit	1	End-to-End	End-to-End x2	#	Loss shape and rate	Min: RTT Max: RTO
QL sQuare + Loss event bits	2	Upstream Downstream End-to-End	Upstream x2 Downstream x2 End-to-End x2	#	→ see Q → see Q|L → see L	→ see Q → see L → see L
QR sQuare + Reflection square bit	2	Upstream “3/4 RT” Opp. Dir. E2E	Upstream x2 “3/4 RT” x2 End-to-End x2 Downstream x2 Half-RT x2	*	Rate over $N * ppa$ packets (e.g. N=64)	Upstream: see Q Others: $N * ppa$ pkts (e.g. N=64)

*	All protocols	x2	Metric in both directions	ppa	Packets-per-Ack
#	Protocols with loss detection (w/ or w/o pure ACK loss detection)			Q|L	See Q if Upstream loss is significant; L otherwise

3bit Explicit Flow Measurements

If there are only 3 bits for EFM (e.g. QUIC):

- Option 1:
  - RTT (S-bit + D-bit)
  - RT Packet Loss (T-bit)
- Option 2a:
  - RTT (S-bit)
  - OneWay P. Loss (Q-bit + L-bit)
- Option 2b:
  - RTT (D-bit or D^-bit)
  - OneWay P.Loss (Q-bit + L-bit)
- Option 3a:
  - RTT (S-bit)
  - OneWay P. Loss (Q-bit + R-bit)
- Option 3c:
  - RTT (D-bit or D^-bit)
  - OneWay P.Loss (Q-bit + R-bit)

Draft next steps

- **Explicit Flow Measurements are gaining interest for encrypted transport protocols:**
  - already discussed in TSVWG and QUIC WG;
  - implementation at IETF Hackathon;
  - thread on the IPPM mailing list.
  - **Packet Loss bits paper**
- **WG adoption requested**
- **Welcome questions and comments.**

Thank you

User Devices Explicit Monitoring

draft-cnbf-ippm-user-devices-explicit-monitoring

July 28, 2021, IPPM WG – IETF 111

M. Cociglio - Telecom Italia (TIM)

F. Bulgarella - Telecom Italia (TIM)

M. Nilo - Telecom Italia (TIM)

G. Fioccola - Huawei Technologies

Proposal: EFM Probes on user devices

- **The draft proposes to put the Explicit Flow Measurements probe also on the user device (e.g. mobile phones, PCs).**
- **"User device EFM rules":**
  1. **The device owner decides whether to mark his traffic.**
  2. **The device owner decides whether to share his performance data.**
- **Strenghts:**
  1. **Scalability**. On the user device there are few connections to monitor.
  2. **More precise measurements**. Client application delay can be measured.
  3. **Both directions monitoring.**
  4. **Network monitoring equipment savings**. Network probes can monitor only impaired connections through "**user device and network probes coordination**". *It's possible to set alarm thresholds on the user device (and to signal to network probes to monitor only the sessions with impairments, in order to segment the performance measurements and to locate the faults). In this case network probes, also embedded into network nodes, need to monitor only a limited number of connections.*

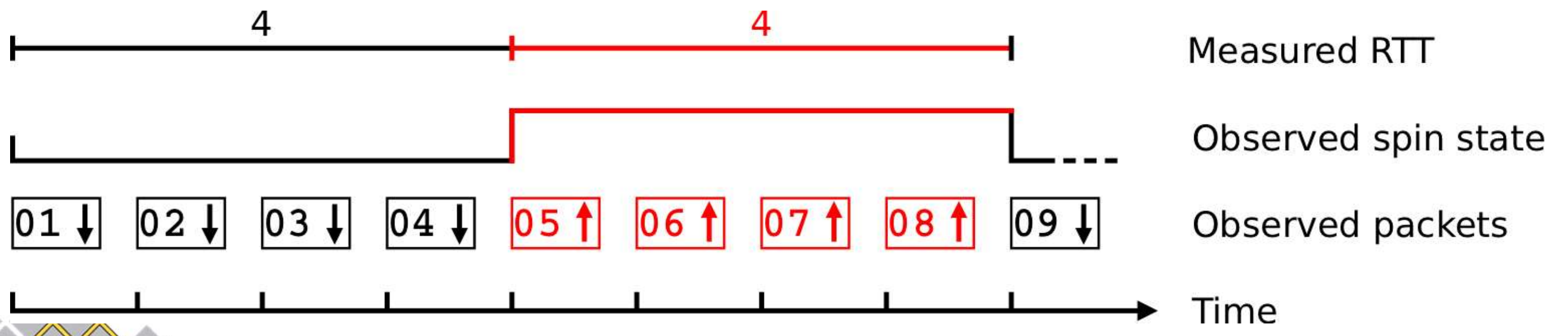
More precise measurements

- Spin bit and Delay bit: the observer-server RTT component measured on the user device is equivalent to the RTT, but without including the client-side application delay and therefore more precise (Right RTT).
- *D-bit and D^-bit give the same measurements (Right RTT)*
- sQuare bit: would measure the End-to-End loss rate in the download direction instead of upstream loss rate.
- Loss event bit: would measure, as before, the End-to-End loss rate in both directions. Moreover, in the upload direction, the signal would be "clean" since it is captured at the origin and therefore not affected by losses.
- Reflection square bit: would measure the RT loss rate instead of three-quarters connection loss rate.

Appendix: support slides

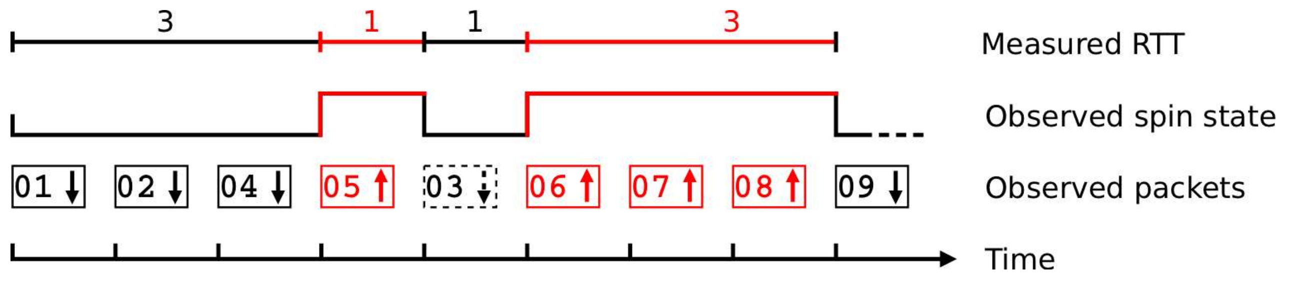
Round Trip Time: Spin bit

- Spin bit for RTT measurement was the first case of Explicit PM.
- It's implemented, optionally, 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.



Spin bit limitations

- Packet loss will tend to cause wrong estimates of RTT due to period width changes.
- Reordering of a spin edge will cause drastic underestimates of RTT since it will cause multiple edges to be observed per RTT. So we need an extra instrument to correctly recognize periods, eluding overlapping.



- “Holes” in the traffic flow can introduce delay in the edge reflection.

We introduce the Delay bit to overcome all these limitations (but the cost could be less measurements if we use only the D-bit):

- One marked packet is not involved in out of sequence (no false periods).
- Deterministic application delay (the «traffic hole»): max 1 ms.
- Edge packet loss can’t cause wrong estimate but only a measurement loss.

D-bit Tmax: the DbS expiration time (Tmax > maxRTT)

Tmax «a priori» (Tmax_p)

The simplest case, Tmax is known «a priori» by Client and Observer:

Proposed value Tmax_p = 1000 ms

Tmax = Tmax_p.

Tmax calculated (Tmax_c)

Tmax is calculated, by Client and Observer, using a measured RTT (RTTm):

Proposed formula Tmax_c=(2*RTTm+100 ms).

Tmax = Tmax_c,

«Measured» RTT (RTTm)

«Measured» RTT has to be known by Client and Observer.

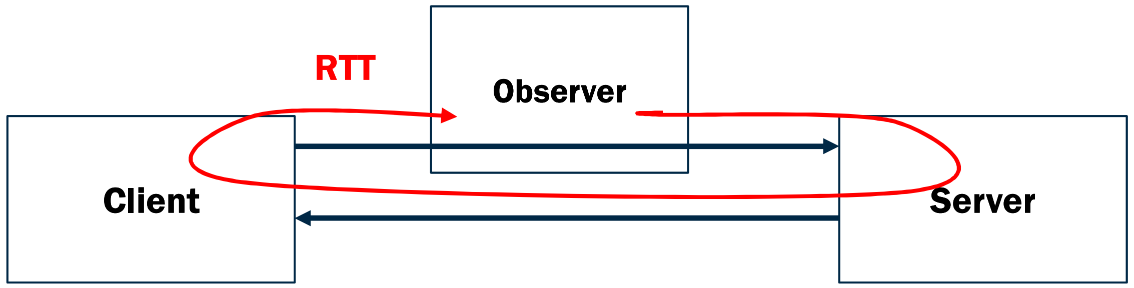
E.g.:

1. 3-way handshake
2. 1° RTT measured with Tmax=1000 ms. (and than Tmax=Tmax_c)
3. Spin bit (if present)

IETF®

D-bit: 1 direction Observer, RTT

- Client–Server RTT:



End-to-End Round-Trip Time => RTT = Ts(DbS_2) – Ts(DbS_1)

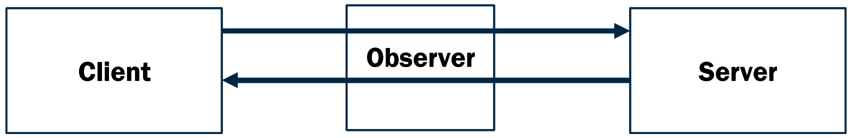
Ts: Timestamp

DbS: Delay bit Sample

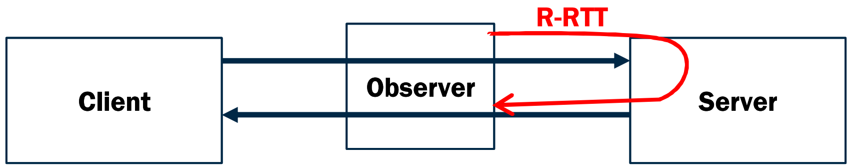
Max Error = +2E (e.g. E=1 ms)

E: application delay threshold

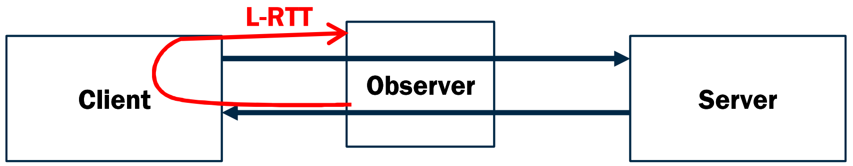
D-bit: 2 direction Observer, Half RTT



- Right RTT (RTT Observer-Server):



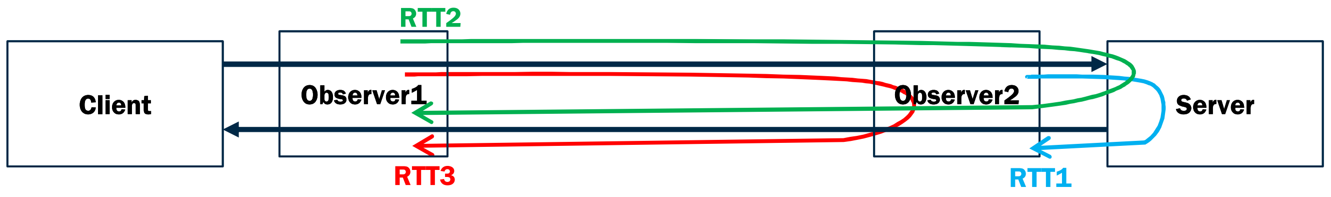
- Left RTT (RTT Observer-Client):



Max Error = +E (e.g. E=1 ms)

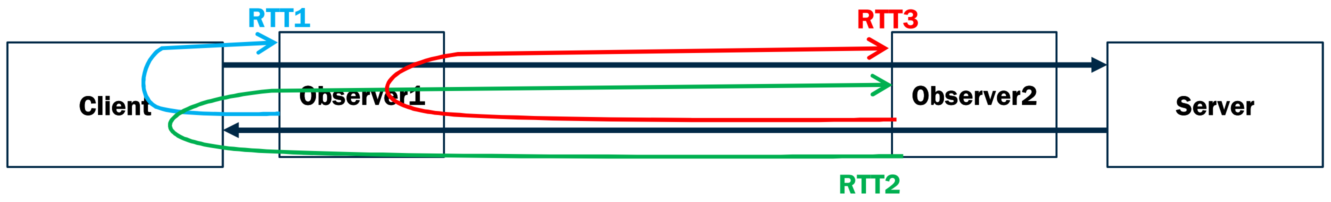
D-bit: 2-Point, intra-domain RTT

- **Observer1-Observer2 RTT:**



Observer1-Observer2 RTT: RTT2 - RTT1 = RTT3

- **Observer2-Observer1 RTT:**



Observer2-Observer1 RTT: RTT2 - RTT1 = RTT3

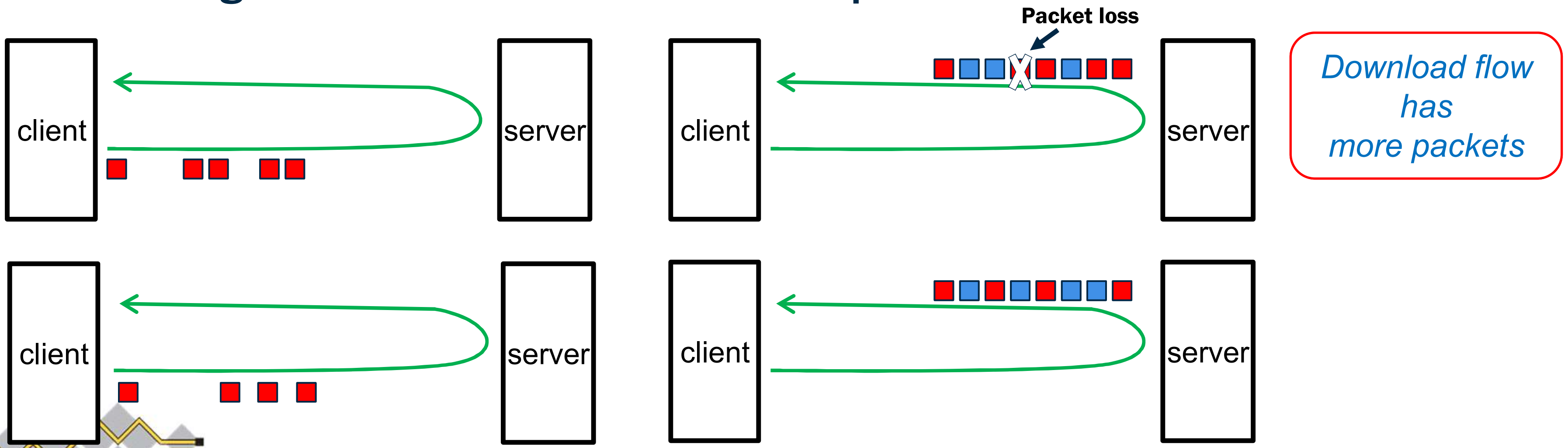
Max Error = ±E (E-0=+E; 0-E=-E) when measured with different marked packets

Max Error = 0 (E-E=0) when measured with the same marked packets

I E T F®

round Trip packet loss: T-bit

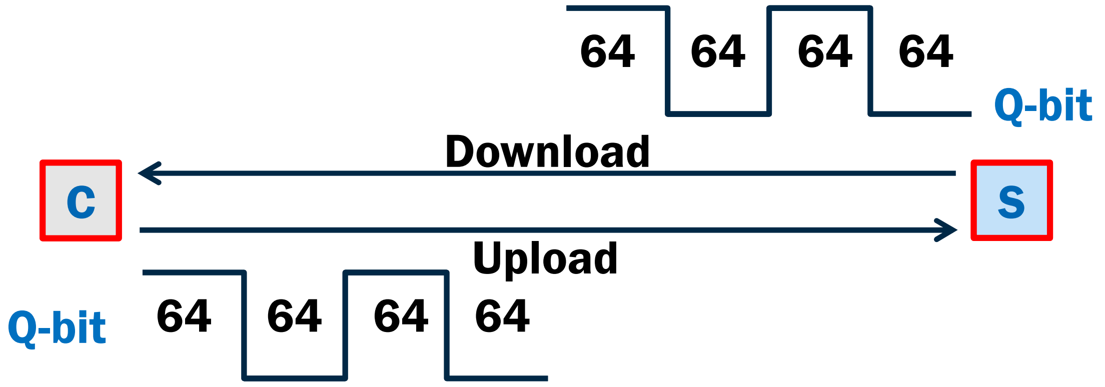
- The Client generate a «train» of market packets (using the T-bit)
- The Server «reflects» these packets (marking production packets flowing in the opposite direction). The Server inserts some not marked packets if download flow has more packets than upload flow.
- The Client reflects the marked packets.
- The Server again reflects the marked packets (two complete Client-Server rounds, so an intermediate Observer can see the «train» twice and compare the marked packets number to measure the RT Packet Loss).
- The Client generate a new train of market packets and so on.

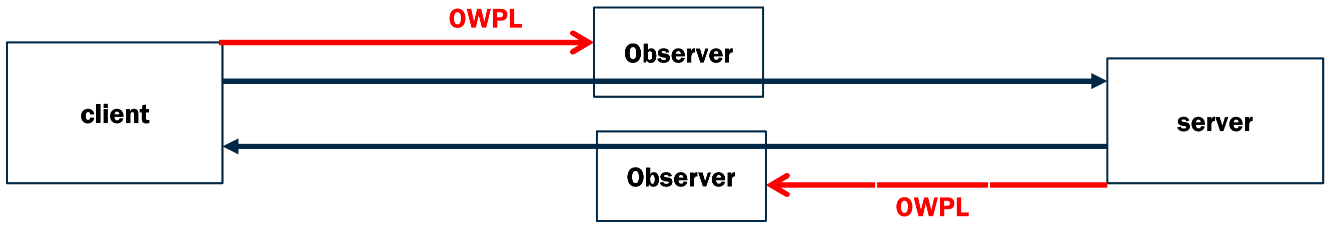


Marked packets: red,
Not Market packets: blue

One-Way Packet Loss: sQuare bit (Q-bit)

- The Q-bit (firstly described in draft-ferrieuxhamchaoui-quic-lossbits) creates square waves of a known length (e.g. 64 packets) as defined in the Alternate Marking RFC 8321

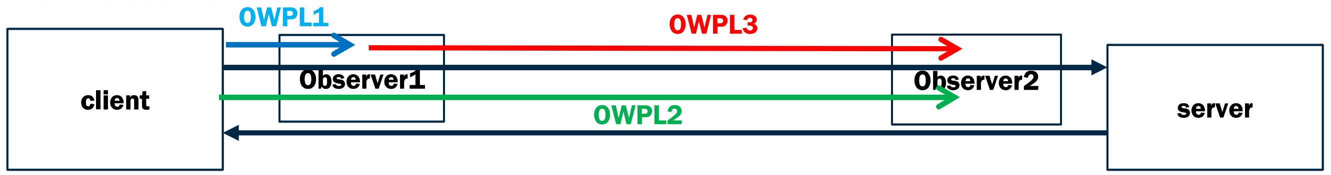




OWPL: One Way Packet Loss

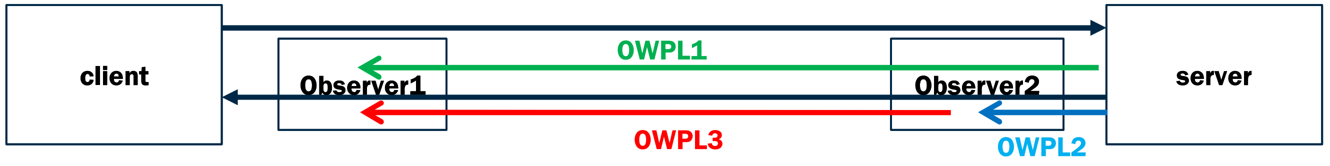
2Point Interdomain Packet Loss (Q-bit)

- Observer2-Observer1 OWPL:



Observer2-Observer1 One-Way: OWPL2 – OWPL1 = OWPL3

- Observer1-Observer2 OWPL:



Observer1-Observer2 One-Way: OWPL1 – OWPL2 = OWPL3

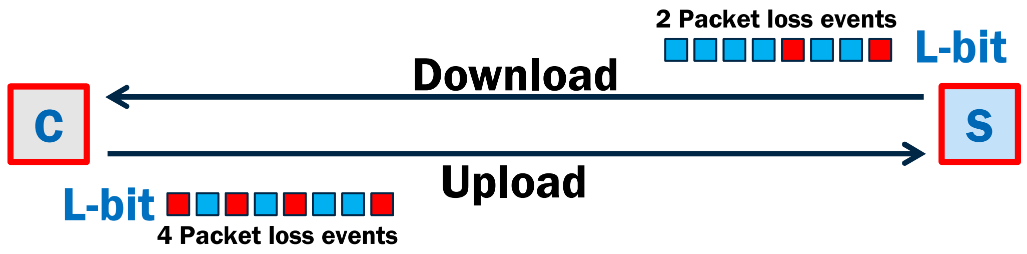
OWPL: One Way Packet Loss

One-Way Packet Loss: Q-bit+L-bit (Loss event bit)

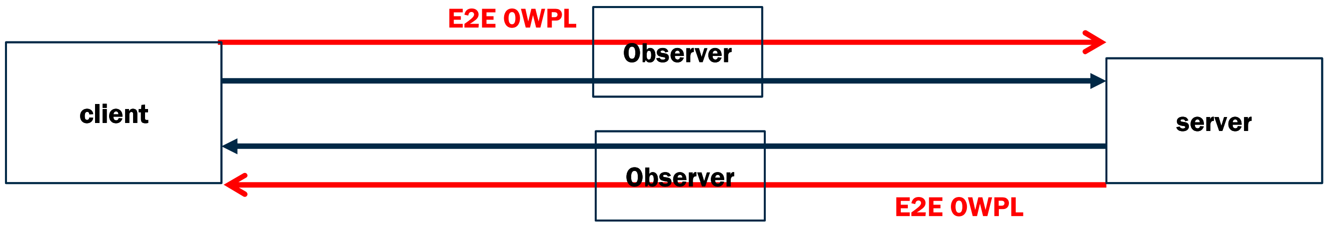
This method uses 2 bits: the sQuare bit (Q-bit) and Loss event bit (L-bit).

- The L-bit (firstly described in [draft-ferrieuxhamchaoui-quic-lossbits]()) marks a packet each time the protocol detect a loss packet event.

L-bit Marked packets: red,
L-bit Not Market packets: blue



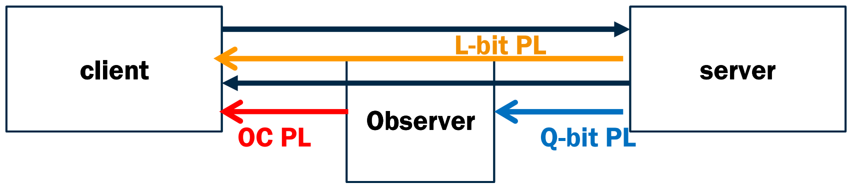
- L-bit measurement:



OWPL: One Way Packet Loss

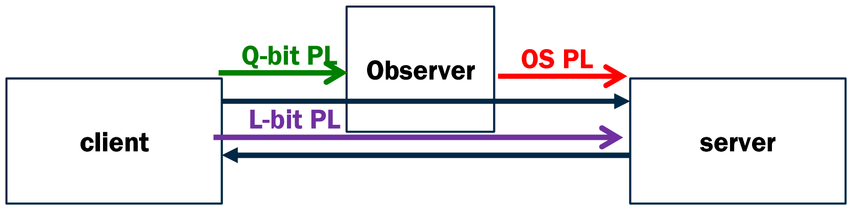
One direction Observer (Q-bit + L-bit):

- **Download Observer:**



Observer-Client PL = L-bit PL Down – Q-bit PL Down

- **Upload Observer:**



Observer-Server PL= L-bit PL Up – Q-bit PL Up

OW Packet Loss: Q-bit+R-bit (Reflection square bit)

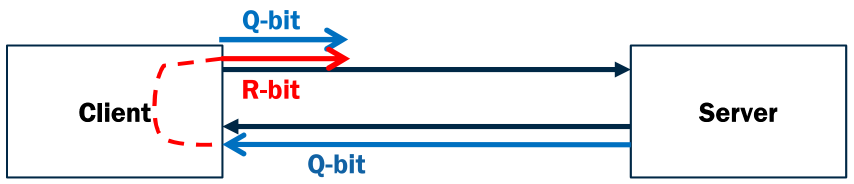
This method uses 2 bits: the sQuare bit (Q-bit) and Reflection square bit (R-bit).

The idea is to reflect the Q-bit in the opposite direction using the R-bit.

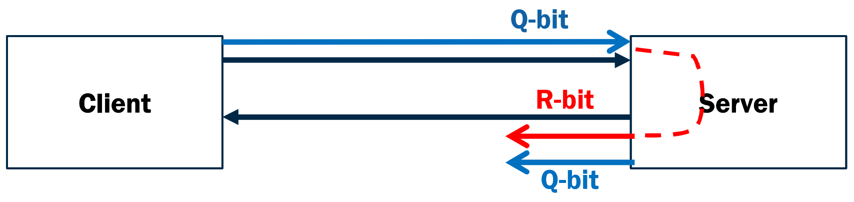
The sizes of the transmitted R-bit blocks are the "average sizes" of the received Q-bit blocks.

This idea allows to have continuous alternate marked packet blocks in both directions.

The Client generates the Q-bit signal and reflects the received Q-bit signal using the R-bit signal:

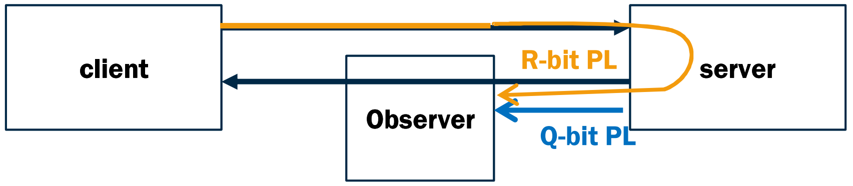


The Server does the same in the opposite direction:

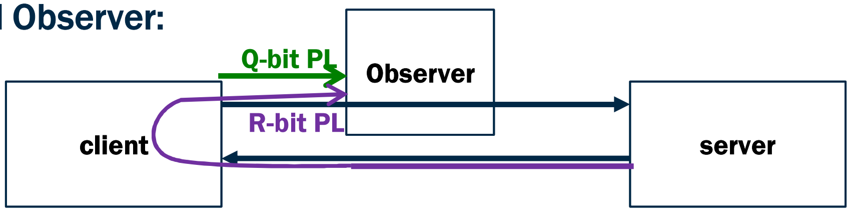


One direction Observer (Q-bit + R-bit):

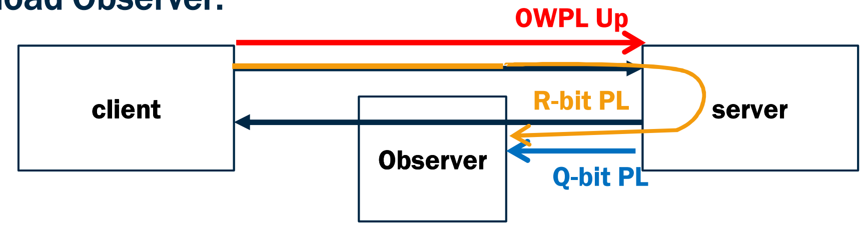
- Download Observer:



- Upload Observer:

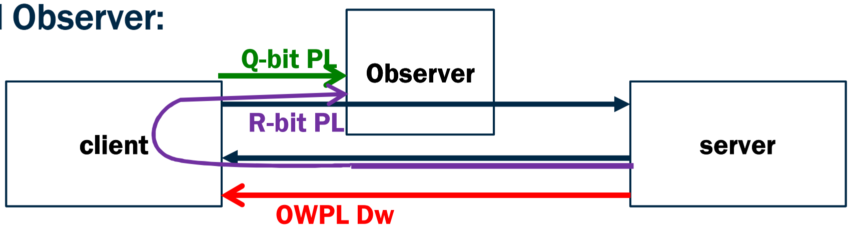


- Download Observer:



OWPL Up = R-bit PL Dw – Q-bit PL Dw

- Upload Observer:

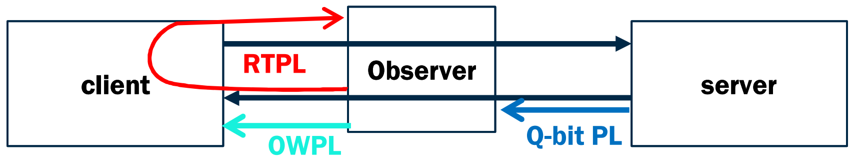


OWPL Dw = R-bit PL Up – Q-bit PL Up

OWPL: One Way Packet Loss

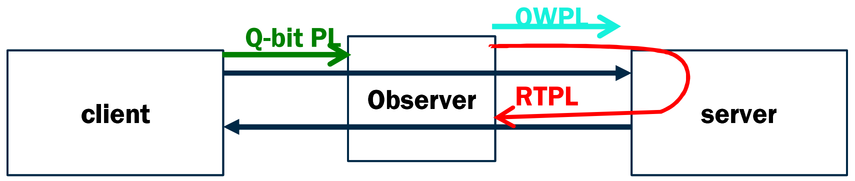
Two direction Observer (Q-bit + R-bit): :

- Observer-Client RTPL and OWPL:



RTPL = R-bit PL Up – Q-bit PL Dw ⇨ RTPL – Q-bit PL Up = OWPL

- Observer-Server RTPL and OWPL:



RTPL = R-bit PL Dw – Q-bit PL Up ⇨ RTPL – Q-bit PL Dw = OWPL

RTPL: Round Trip Packet Loss
OWPL: One Way Packet Loss