traffic types

1	Isochronous
2	Cyclic
3	Alarms and events
4	Configuration and diagnostics
5	Network control
6	Best effort
7	Video
8	Audio/voice

OPC UA Publish @ 100us | Qbv Window 1

Intel’s Sample Code @ say 2ms (In future it will be another OPC UA Publish in parallel)

OPC UA Client Server

Config & Diagnostics (NETCONF)

PTP

Best Effort (flood ping, secure Client/Server)

Video (Infotainment/Streaming) (VideoLAN)

Audio/Voice (Peer to peer) (SIP)

our goal

Test two Linux systems running OPC UA TSN under

- Realistic functional load (all traffic types & realistic CPU load)
- Periodic robustness test loads

Our definition of test loads:

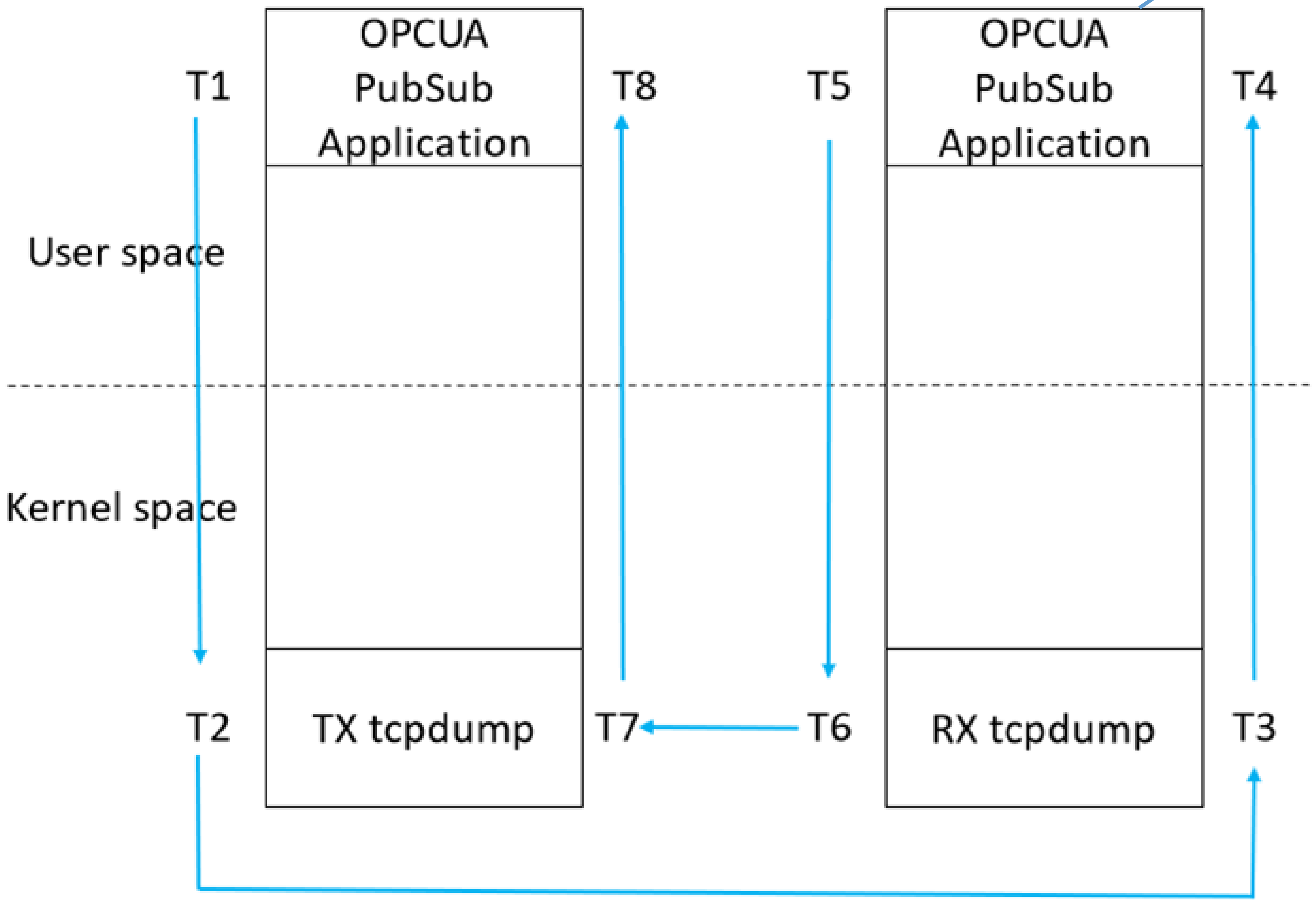
- No Load – no network load other than PTP and no CPU load other than cyclictest to establish max latencies
- Functional Load – Data in 8 identified traffic types + reasonable CPU load (say 30 to 40%)
- Robustness Load – Max functional data in 8 identified traffic types + above average CPU load (say 80%)
- Knock-out Load – Load system using tools like stress-ng and flood pings, etc to crash the system and identify precautions to be taken in case of an automation system – for e.g. IO status

Future scope:

- Introduce switches, more nodes and more network topologies
- Introduce more hardware platforms

24x7x365 test setup

Oscilloscope connected to parallel ports of each system



T1 – UDP packet publish timestamp

T2 – tcpdump outgoing packet timestamp *(not taken into account because it is not reliable)*

T3 – tcpdump incoming packet timestamp

T4 – UDP packet subscribe timestamp

T5 – UDP loopback packet publish timestamp

T6 – tcpdump looped-back outgoing packet timestamp *(not taken into account because it is not reliable)*

T7 – tcpdump looped-back incoming packet timestamp

T8 – UDP loopback packet subscriber timestamp

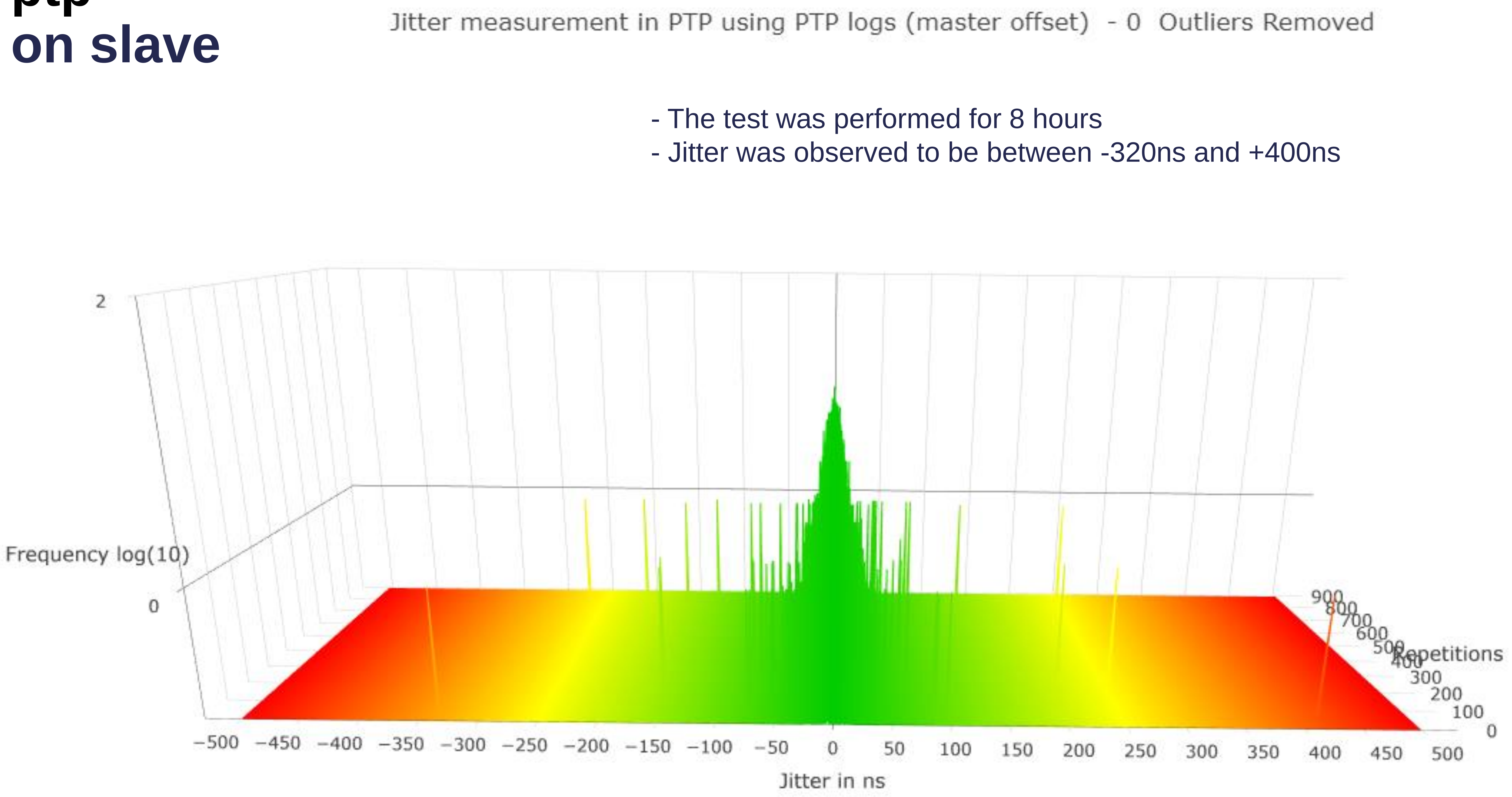
4 traces to cross-check PTP
4 traces from Application on top of OPC UA Stack
2 TCP Dumps only capturing received packets

traces collected from test setup

- Oscilloscope in persistent mode
  - Helps us visually ensure that PTP & PHC2SYS are working in nano-sec accuracy
  - Helps us visually ensure that time-sync accuracy in user space (is about ± 20us today)
  - Helps us cross check the T4-T1 visually
- PTP
  - ptp4l logs on both systems (will not be available in master)
  - phc2sys logs on both systems
- Qbv
  - T3 (only Rx) – 8 individual graphs + 1 graph showing all traffic
  - T7 (only Rx) – 8 individual graphs + 1 graph showing all traffic
- Kernel & User Space
  - Long term jitter and max latency for each software/hardware interrupt (high precision timers, Rx, etc).
  - Log memory & CPU utilization
- Jitter on Rx
  - Long term 3d plots & gaussian plots on T3 & T7
- Latency on Rx
  - Long term 3d plots & gaussian plots on T4-T1 (end to end one way)
  - T3-T1 (needed until XDP - eXpress Data Path comes in place)
  - T4-T3 (XDP)
  - T8-T1 (end to end round trip)

ptp
on slave

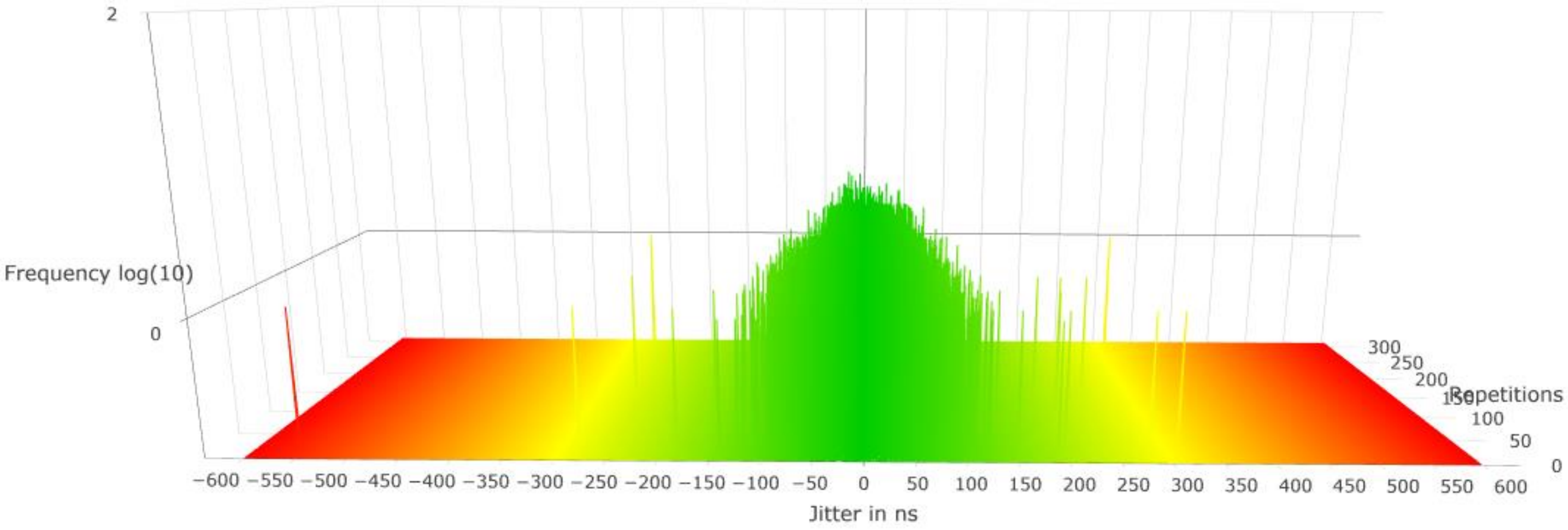
- The test was performed for 8 hours
- Jitter was observed to be between -320ns and +400ns



phc2sys on master

- The test was performed for 8 hours
- Jitter was observed to be between –1.1us and 1us

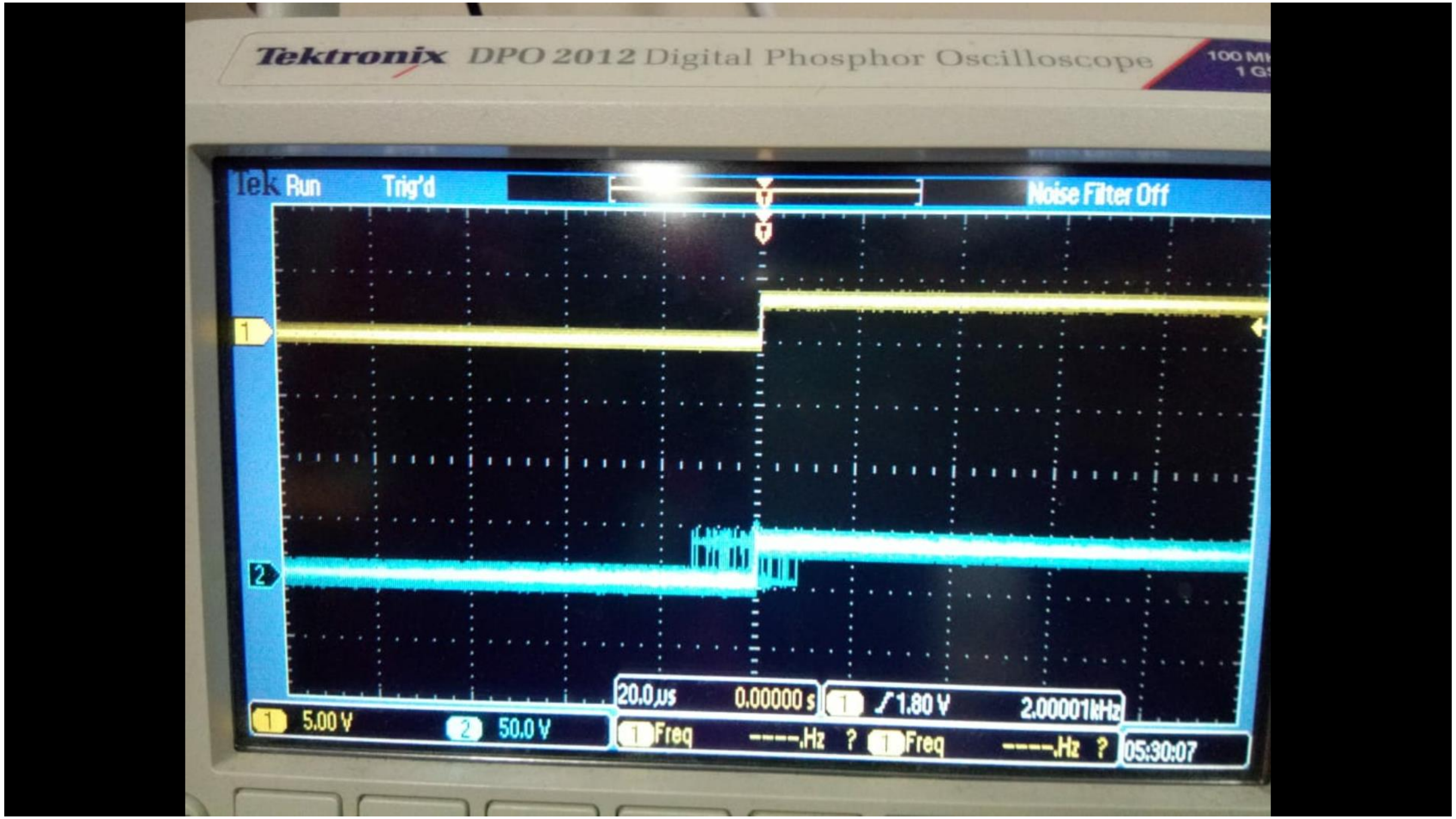
phc2sys on slave



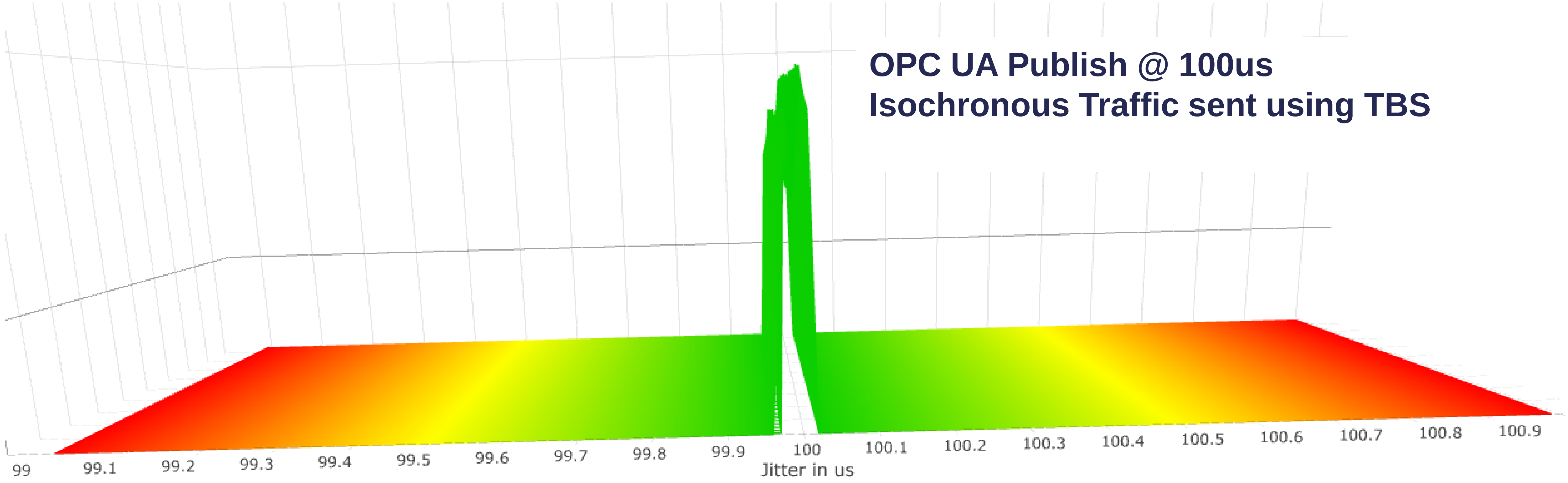
- The test was performed for 4 hours
- Jitter was observed to be between -560ns and +300ns

time synced
user space applications

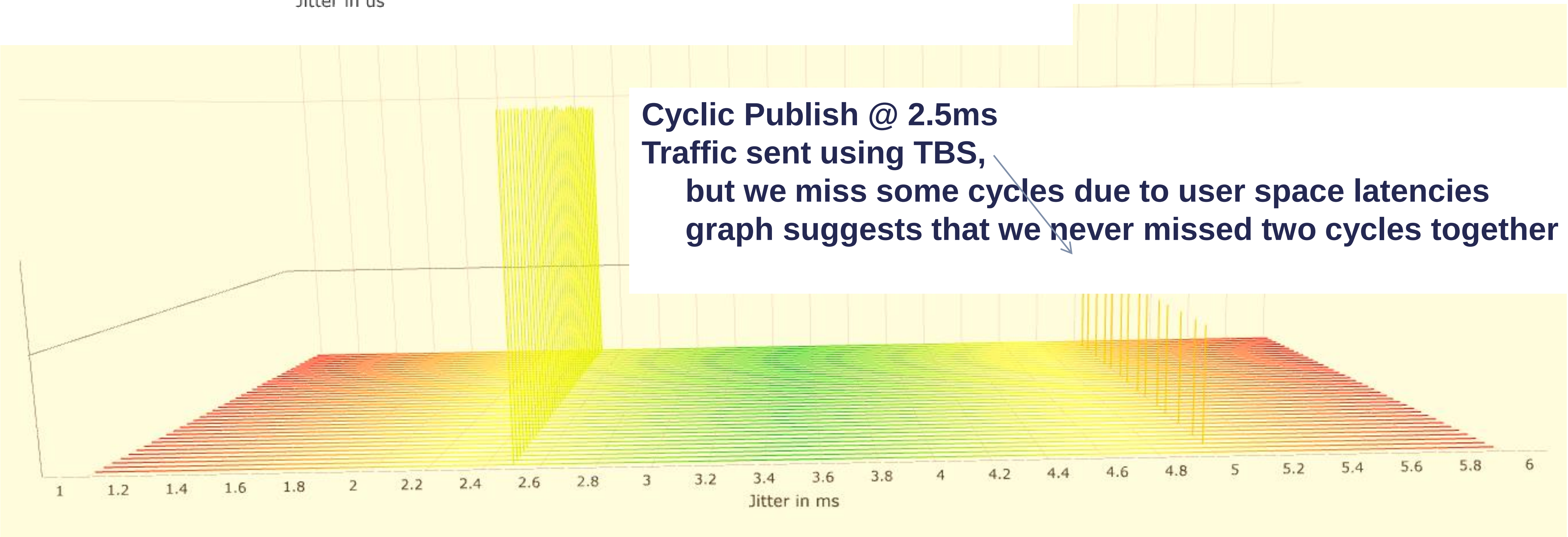
First priority: Achieve stable ±20us
Second priority: Reduce from ±20us
(PTP time sync is in itself in ±1us range)



jitter graphs
that show tbs

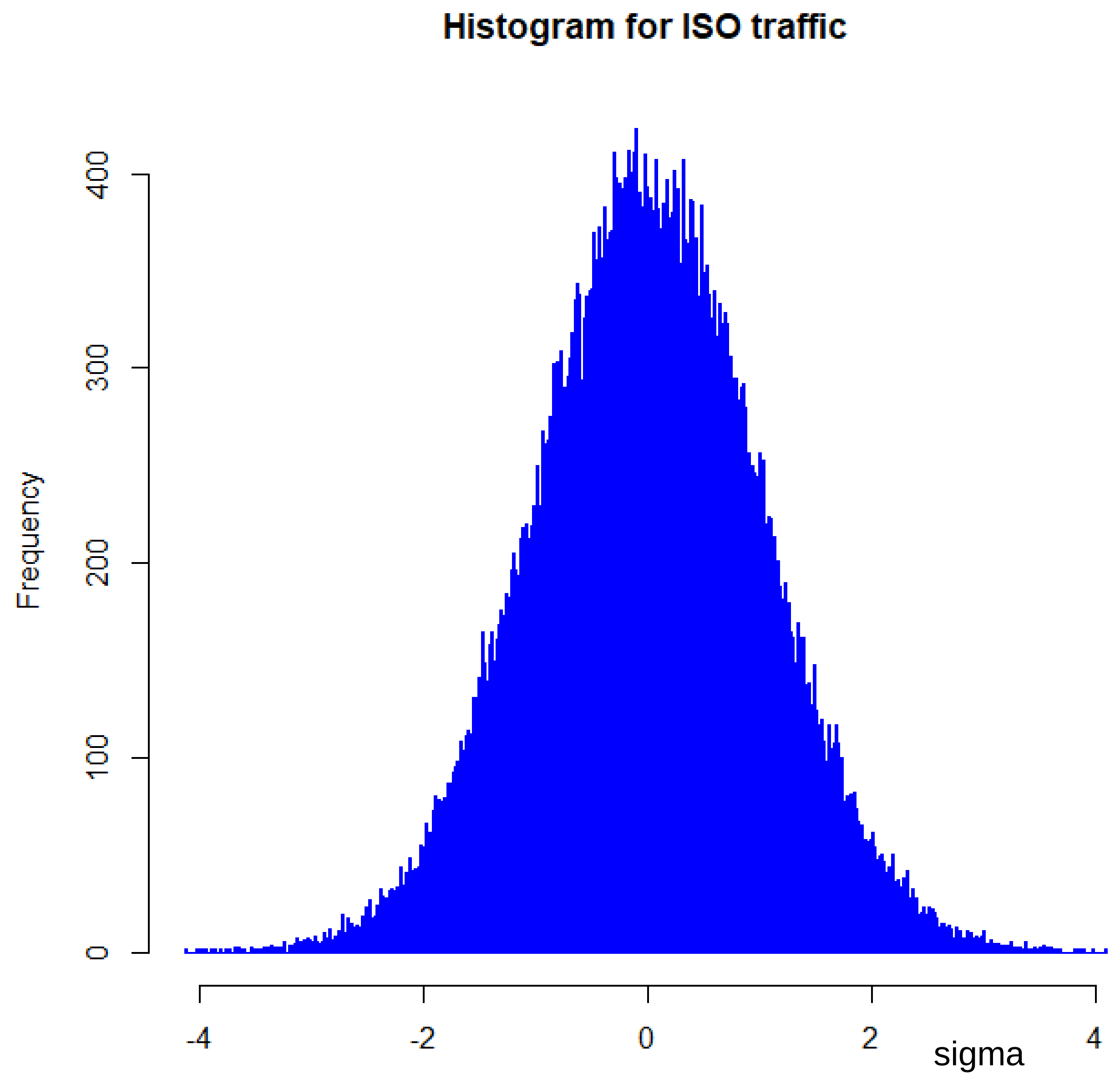


OPC UA Publish @ 100us
Isochronous Traffic sent using TBS

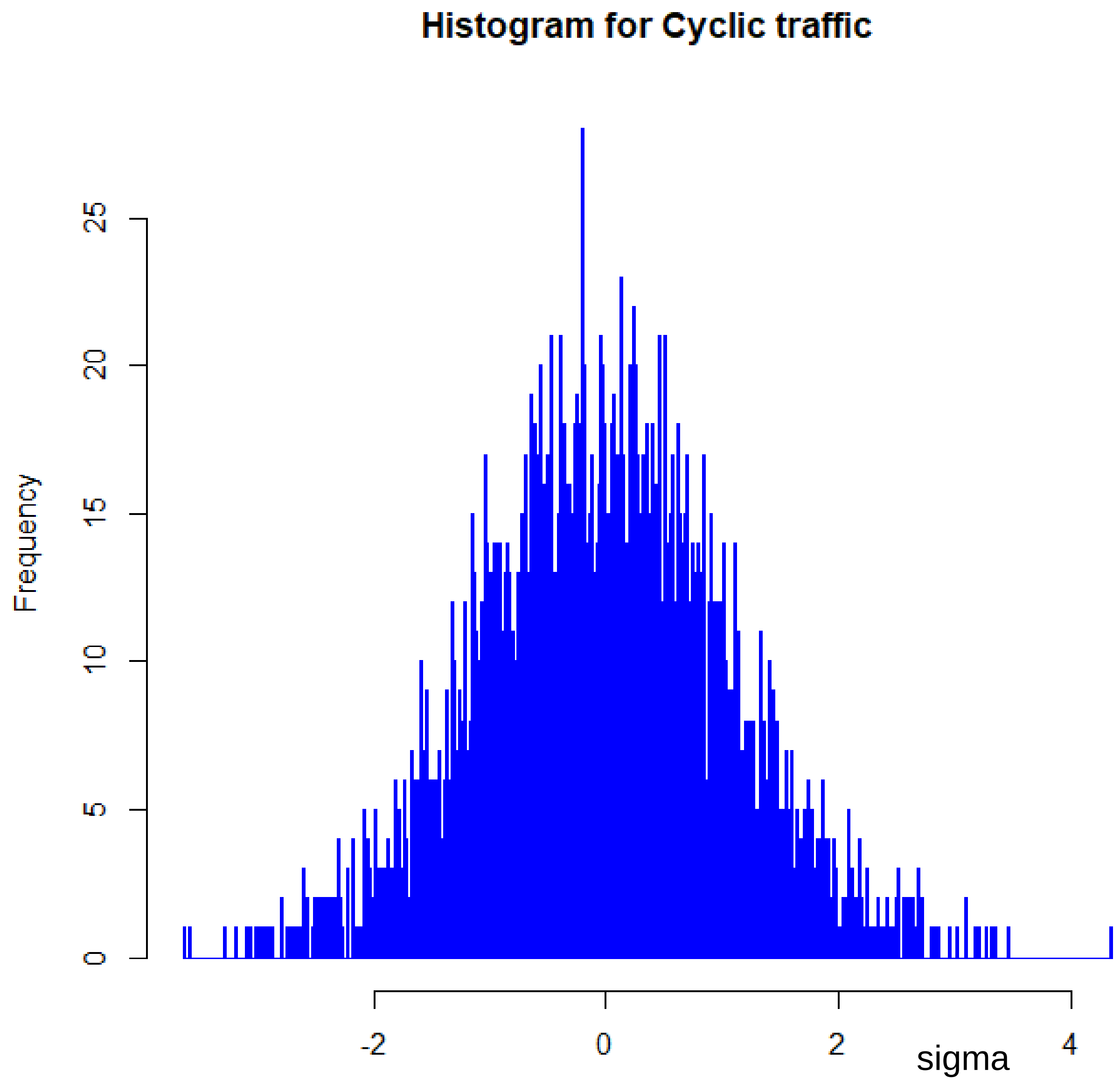


Cyclic Publish @ 2.5ms
Traffic sent using TBS,
- **but we miss some cycles due to user space latencies**
- **graph suggests that we never missed two cycles together**

standard normal distribution
jitter in isochronous and cyclic publish

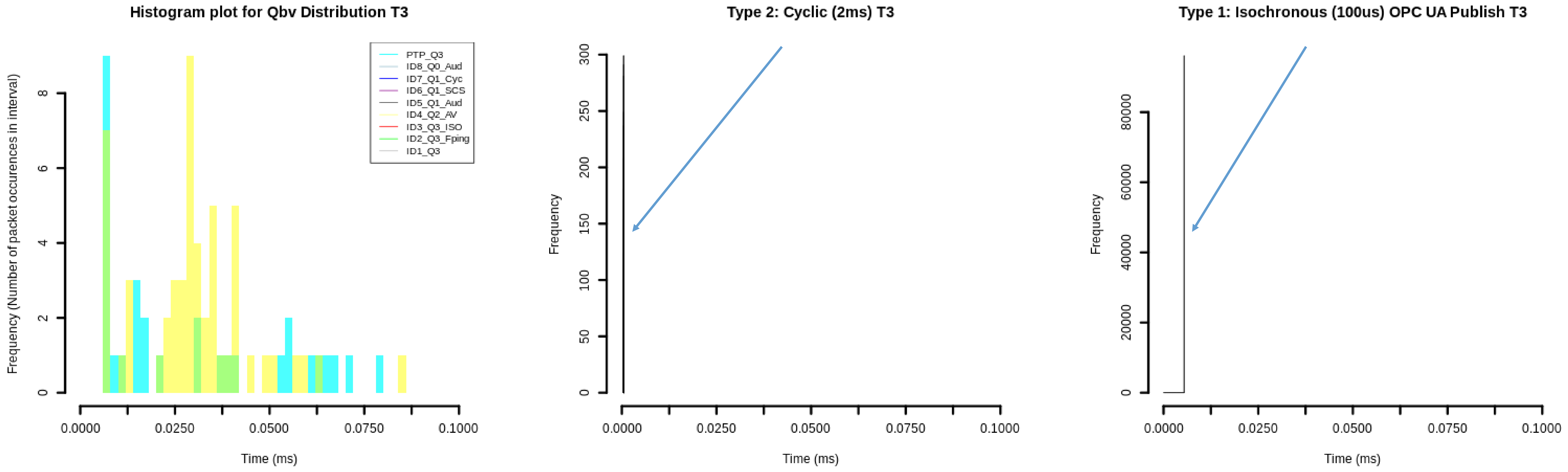


mean - 100000 (ns)
sd - 155 (ns)



mean - 2500000(ns)
sd - 133(ns)

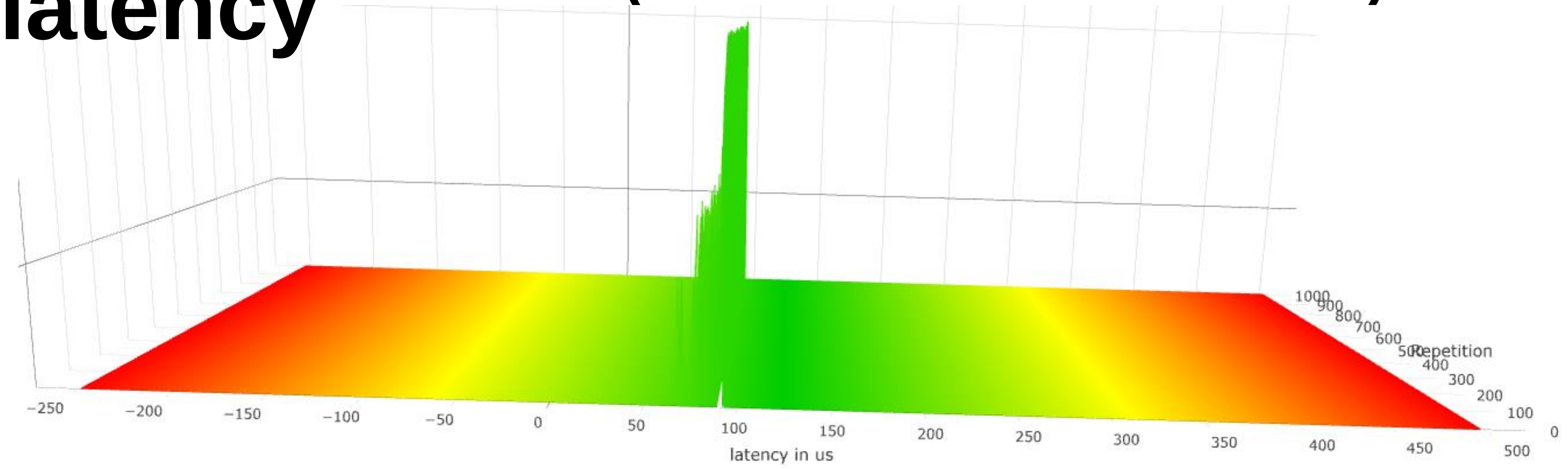
Qbv
graph



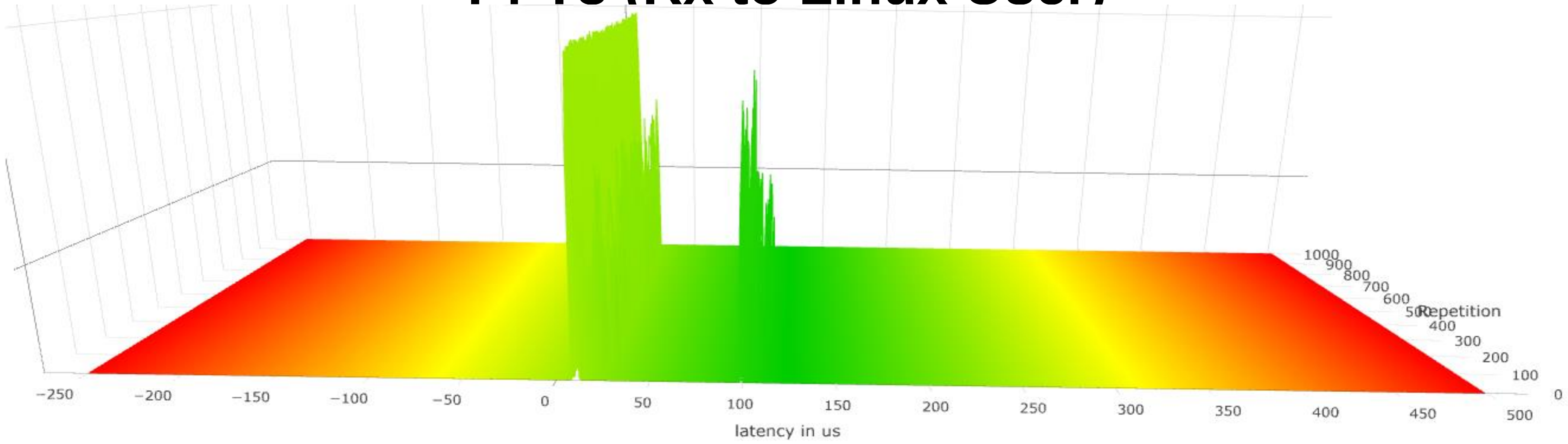
Number of frames: 100000

latency

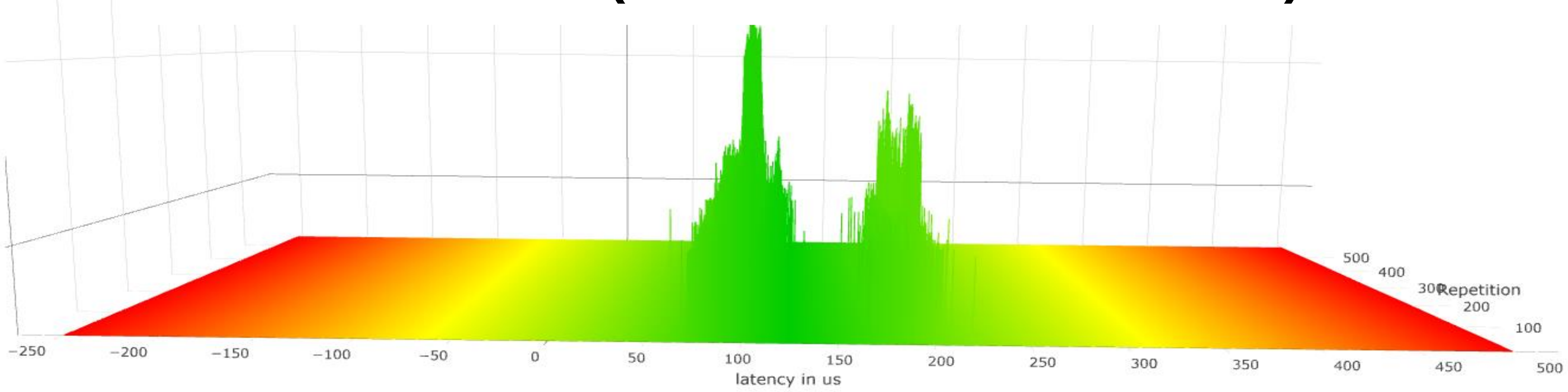
T3-T1 (excludes Rx to Linux User)

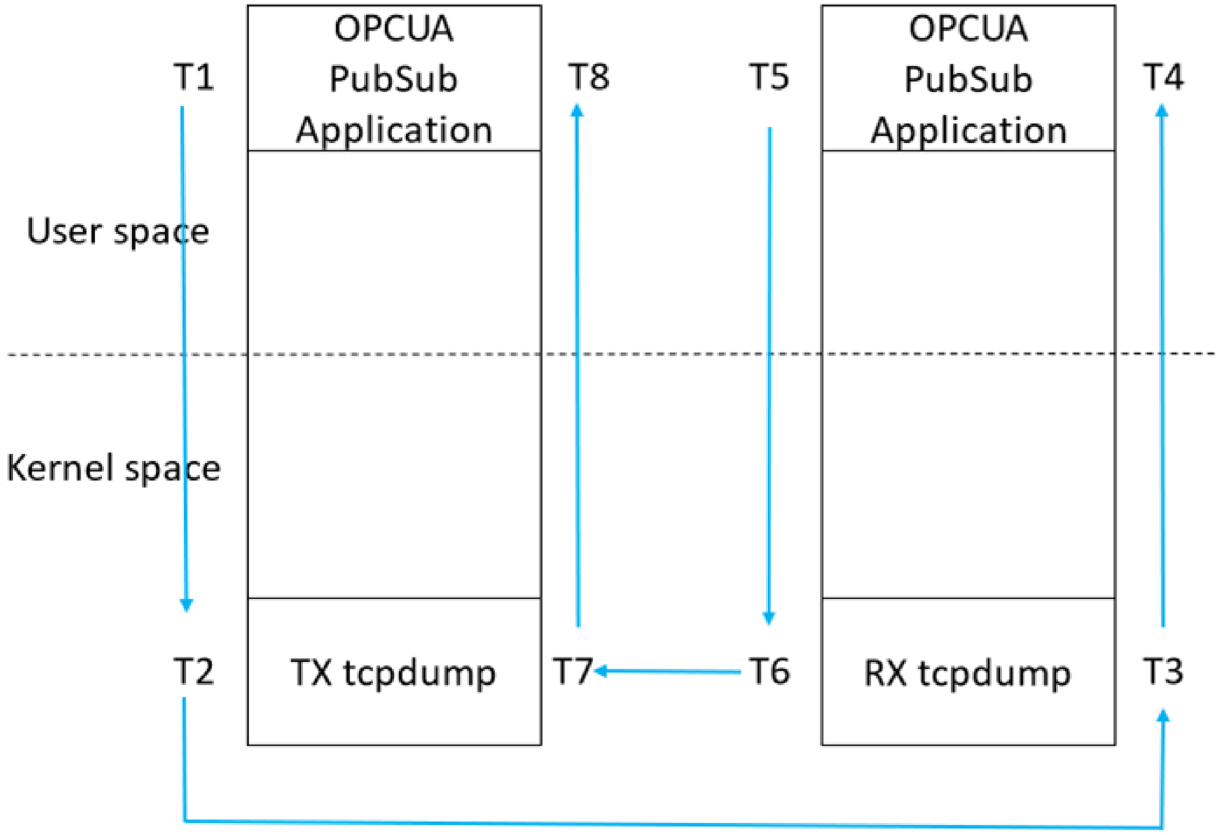


T4-T3 (Rx to Linux User)



T4-T1 (includes Rx to Linux User)



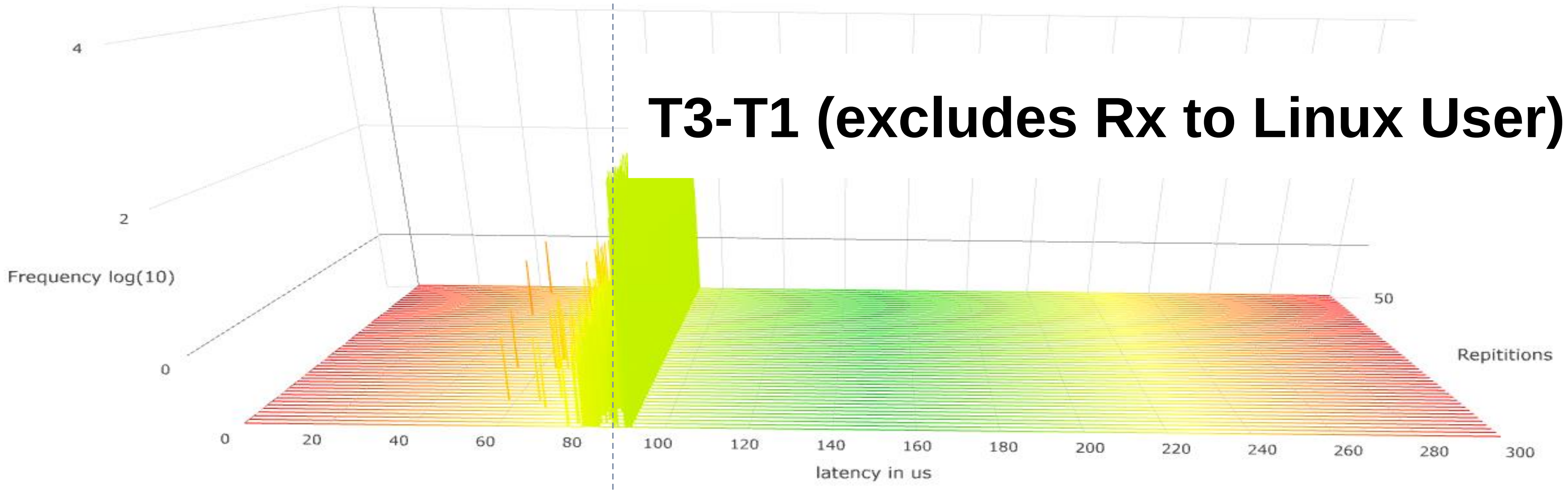


T3 – T1 :
Max latency: 92.16us
Job number: 299

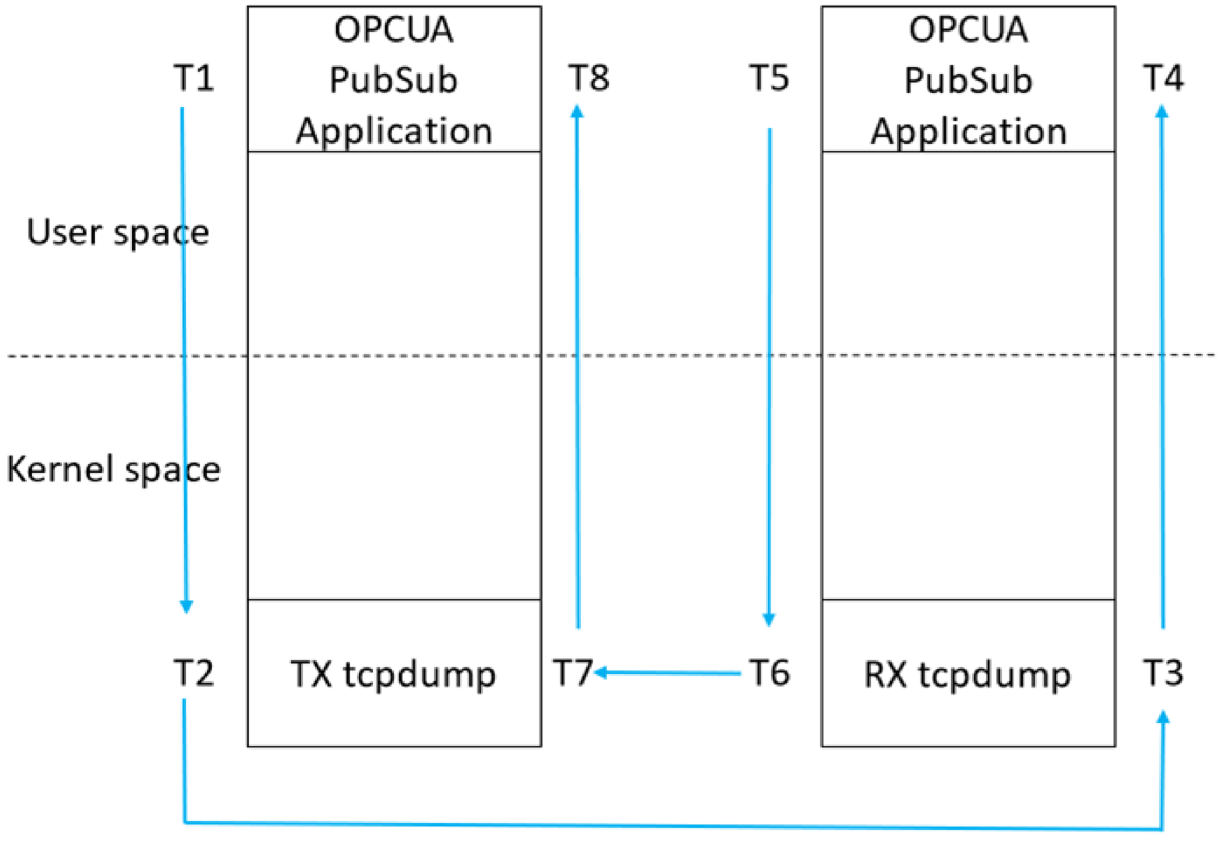
T4 – T3 :
Max latency: 114.4us
Job number: 296

T4 – T1 :
Max latency: 222.20us
Job number: 290

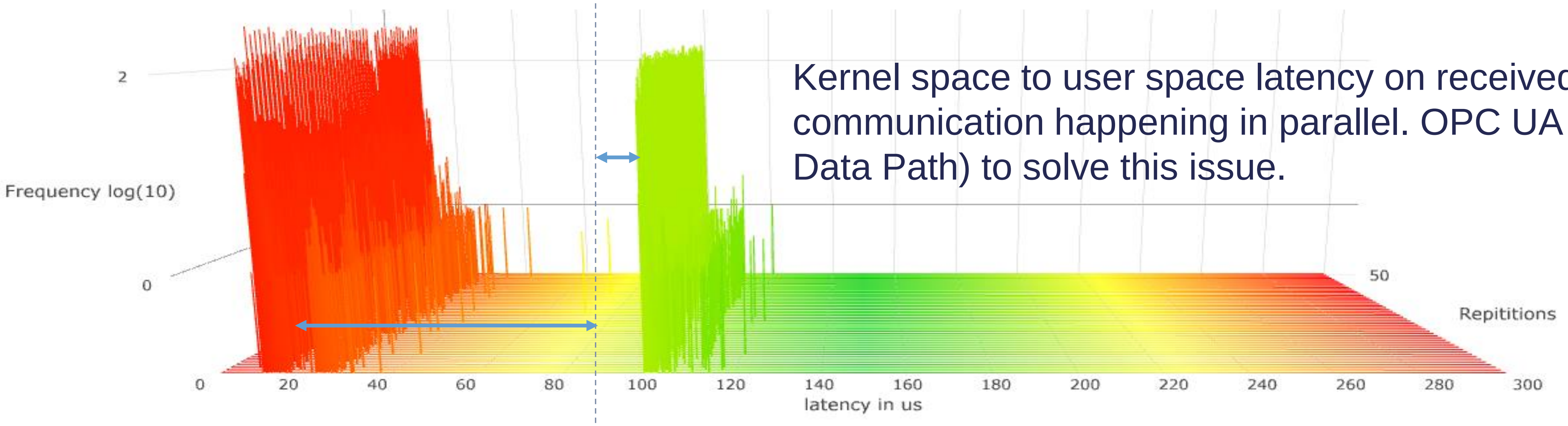
OPC UA
Subscribe latency



T3-T1 (excludes Rx to Linux User)



T4-T3 (Rx to Linux User)



Kernel space to user space latency on received packet is quite high when we also have other communication happening in parallel. OPC UA Subscribe to be modified to use xDP (Express Data Path) to solve this issue.

All traces shown are taken using currently available Open Source implementations of OPC UA TSN

Global Open Source Eco-system

We are pioneering the creation of global
OPC UA TSN open source eco-system
to reduce entry barrier for customers and
to reduce time-to-market for products

OPC UA

German Government stated Industry 4.0 initiative
in 2014 and recommended OPC UA as the only
standard for communication system in 2015

TSN

IEEE standards for Ethernet got their first major
revision in 40 years, becoming time aware to
enable new use-cases that aren’t possible yet

open62541 is maintained by

Commercial support partners of open62541

Hardware and project participants of first project phase

Project participants of first project phase

thought leadership in industry 4.0

Open Source Eco-system for OPC UA TSN

Embedded World, Nuremberg 2018

IIC TSN Testbed 2018

Hannover Messe 2018