Study on EPICS Communication over Long Distance

L. Lobes[1], R. Lange[2], O. Semenov[3], D. Stepanov[2]

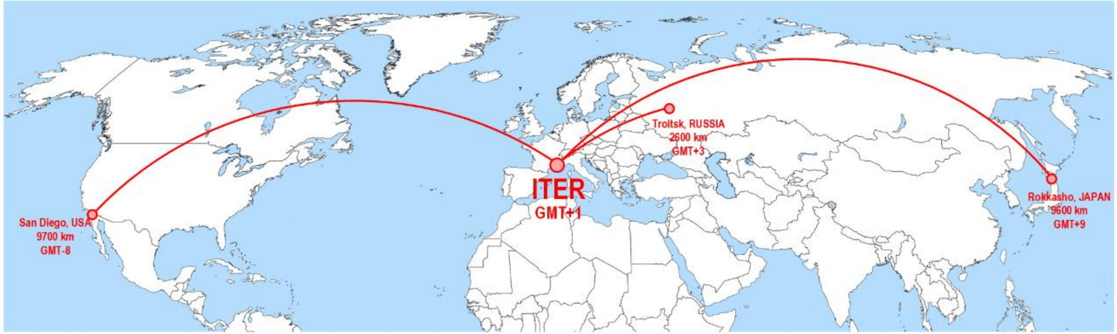
[1] Tomsk Polytechnic University, Lenin av. 30, 634050 Tomsk, Russia

[2] ITER Organization, Route de Vinon-sur-Verdon, CS 90 046, 13067 St. Paul Lez Durance Cedex, France

[3] Institution "Project Center ITER", Kurchatov sq. 1, Building 3, 123182 Moscow, Russia

Disclaimer: The views and opinions expressed herein do not necessarily reflect those of the ITER Organization

General test algorithm

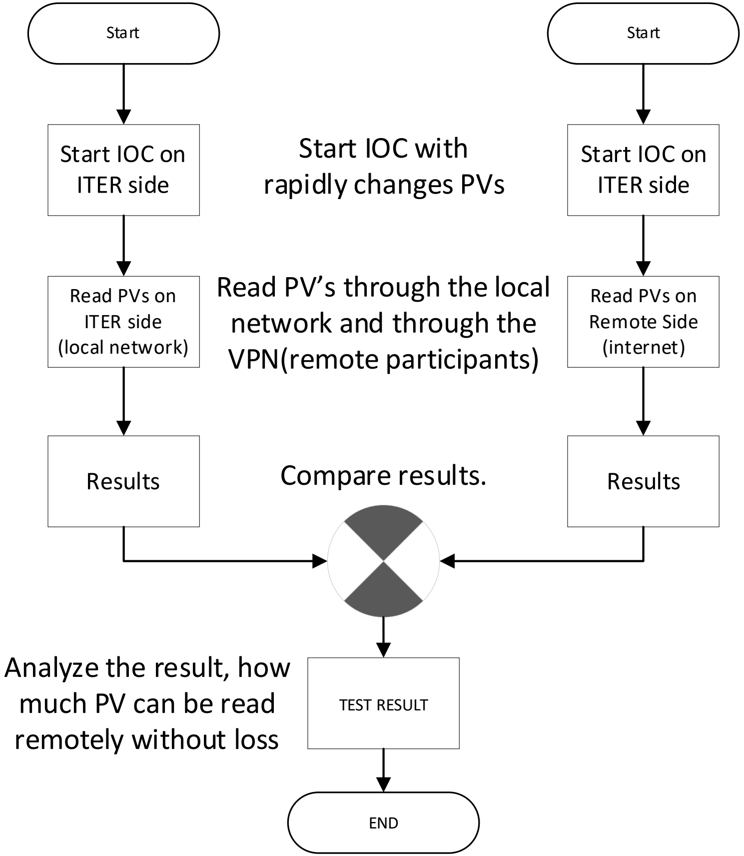


Already connected Remote participants:
- ❖ Troitsk, RUSSIA, IPSec VPN, ping around 69 ms
- ❖ San Diego, USA, IPSec VPN, ping around 160 ms
- ❖ Rokkasho, JAPAN, L2 VPN, ping around 250 ms

All remote participants will be watchers only, with no control possibility

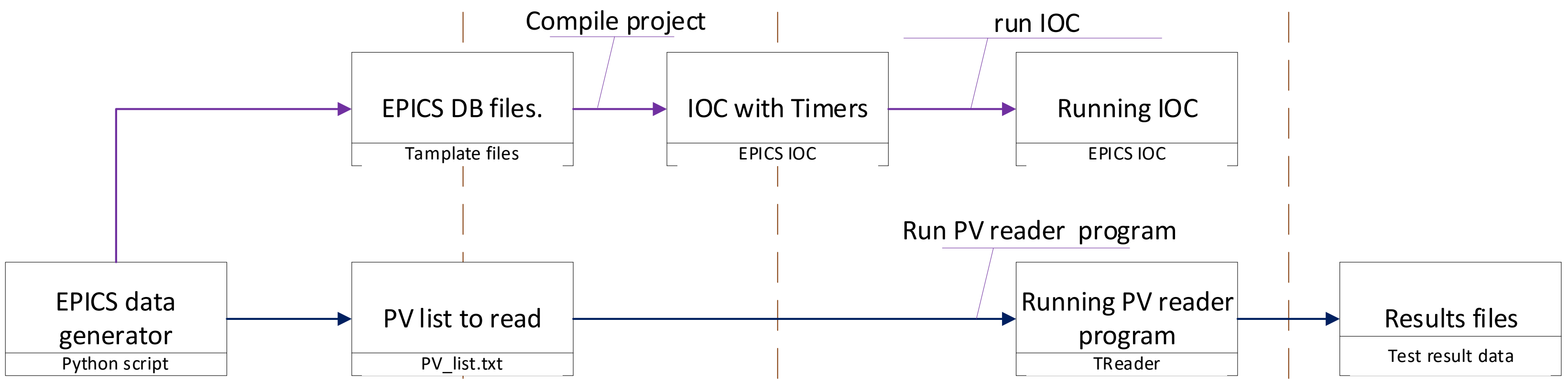
---

Mainly tests results presented in this presentation was obtained with Troitsk, RUSSIA and USA San Diego.
Japan in plans in near future.



Program execution sequence

STEP 1 | STEP 2 | STEP 3 | STEP 4



STEP 1: set up the generator. The generator generates files for EPICS and also generates a list of the same PVs in a text file(PV_list.txt) for using by Treader program

STEP 2: Compile simple maven project which contain only one IOC.

STEP 3: Start the IOC and start the reading program

STEP 4: Data analysis and processing

EPICS DB, Control part

EPICS DB consist of configurable number of counters(Timers).

Main control record:

```
record(mbbo, "MSC:TP") {
    field(DESC, "Counters period controll record")
    field(OUT,  "MSC:T-0.SCAN")
    field(ZRST, "off")          field(ZRVL, "0")
    field(FVST, "2 second")     field(FVVL, "5")
    field(SXST, "1 second")     field(SXVL, "6")
    field(SVST, ".5 second")    field(SVVL, "7")
    field(EIST, ".2 second")    field(EIVL, "8")
    field(NIST, ".1 second")    field(NIVL, "9")
```

Reset timers value record:

```
record(bo, "MSC:RESET") {
    field(DESC, "reset timers")
    field(VAL,  "0")
    field(ZNAM, "GO")
    field(ONAM, "RESET")
}
```

EPICS DB: Timers part

```
record(calc, "MSC:T-0"){
    field(CALC, "B?0:val+1")
    field(SCAN, "5 second")
    field(FLNK, "MSC:T-1")
    field(INPB, "MSC:RESET")
}

record(calc, "MSC:T-1"){
    field(CALC, "B?0:val+1")
    field(SCAN, "Passive")
    field(FLNK, "MSC:T-2")
    field(INPB, "MSC:RESET")
}
```

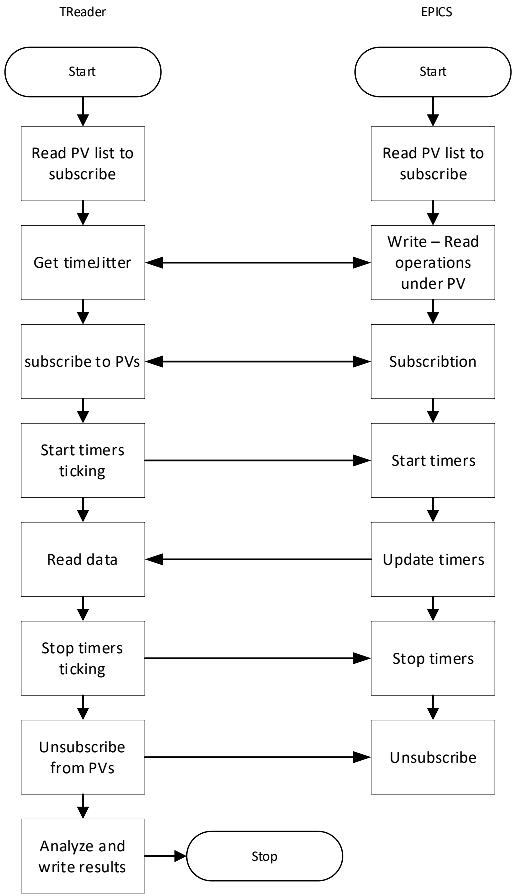
Up to 4000 records

Every 4000 PV:

```
record(calc, "MSC:T-4000"){
    field(CALC, "B?0:val+1")
    field(SCAN, "Passive")
    field(FLNK, "MSC:T-4001.PROC CA")
    field(INPB, "MSC:RESET")
}

record(calc, "MSC:T-4001"){
    field(CALC, "B?0:val+1")
    field(SCAN, "Passive")
    field(FLNK, "MSC:T-4002")
    field(INPB, "MSC:RESET")
}
```

Timers reading program(TReader)



TReader is control:

- Timers update frequency
- Reading time
- Number of timers to subscribe
- Thread quantity

Only writing to RAM while reading data, no other tasks occur here

The program analyzes data and calculates several metrics

Test Metrics

TReader measure several metrics during the test:

- PV value order metric
- Connection time – time between star connection and connection complete.
- Latency of update MAX – maximum time between the same event on the EPICS server and on the client
- Latency of update AVG – average time between the same event on the EPICS server and on the client
- Latency of update MIN – minimum time between the same event on the EPICS server and on the client
- Server – Server update time MAX AVG MIN – time between update the same PV on EPICS server side
- Client – Client update time MAX AVG MIN – time between update the same PV on client side

PV value order metric

The IOC timers is incrementing by 1 in each update, if there is a difference in neighborhood points on client side read data it is marked as an error.

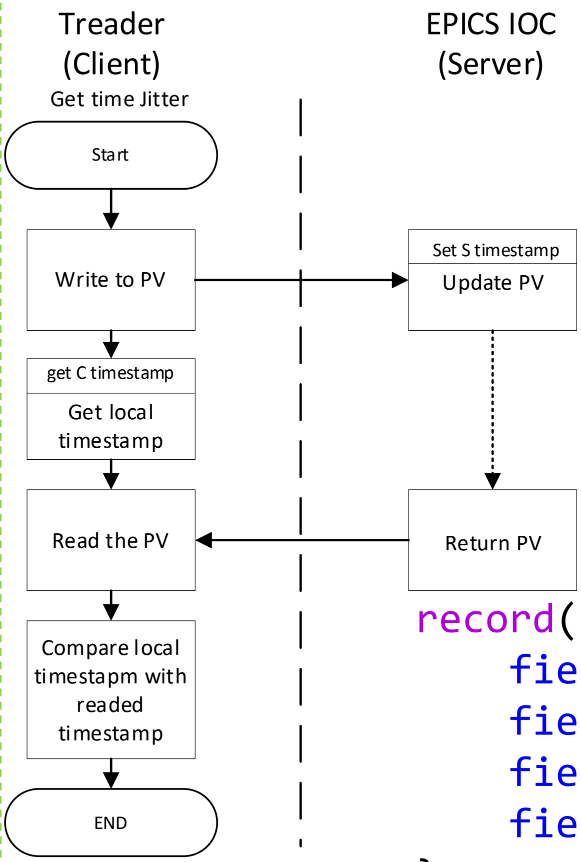
	Server side PVs			Client readed PVs		
(time) t	T1	T2	T3	T1	T2	T3
	1	1	1	1	1	1
	2	2	2	2	2	2
	3	3	3	3	3	3
	4	4	4	4	4	4
	5	5	5	6	5	5
	6	6	6	7	6	6
	7	7	7	8	7	7
	8	8	8	9	8	8
	9	9	9	10	9	9
	10	10	10	11	10	10
	11	11	11	12	14	11
	12	12	12	13	15	12
	13	13	13	14	16	13
	14	14	14	15	17	14
	15	15	15	16	18	15
	⋮	⋮	⋮			

ERROR Event
Difference between neighborhood elements bigger than 1

Channel latency compensation algorithm

Channel latency from ITER to RF RPC in used VPN channel is 69 ms.

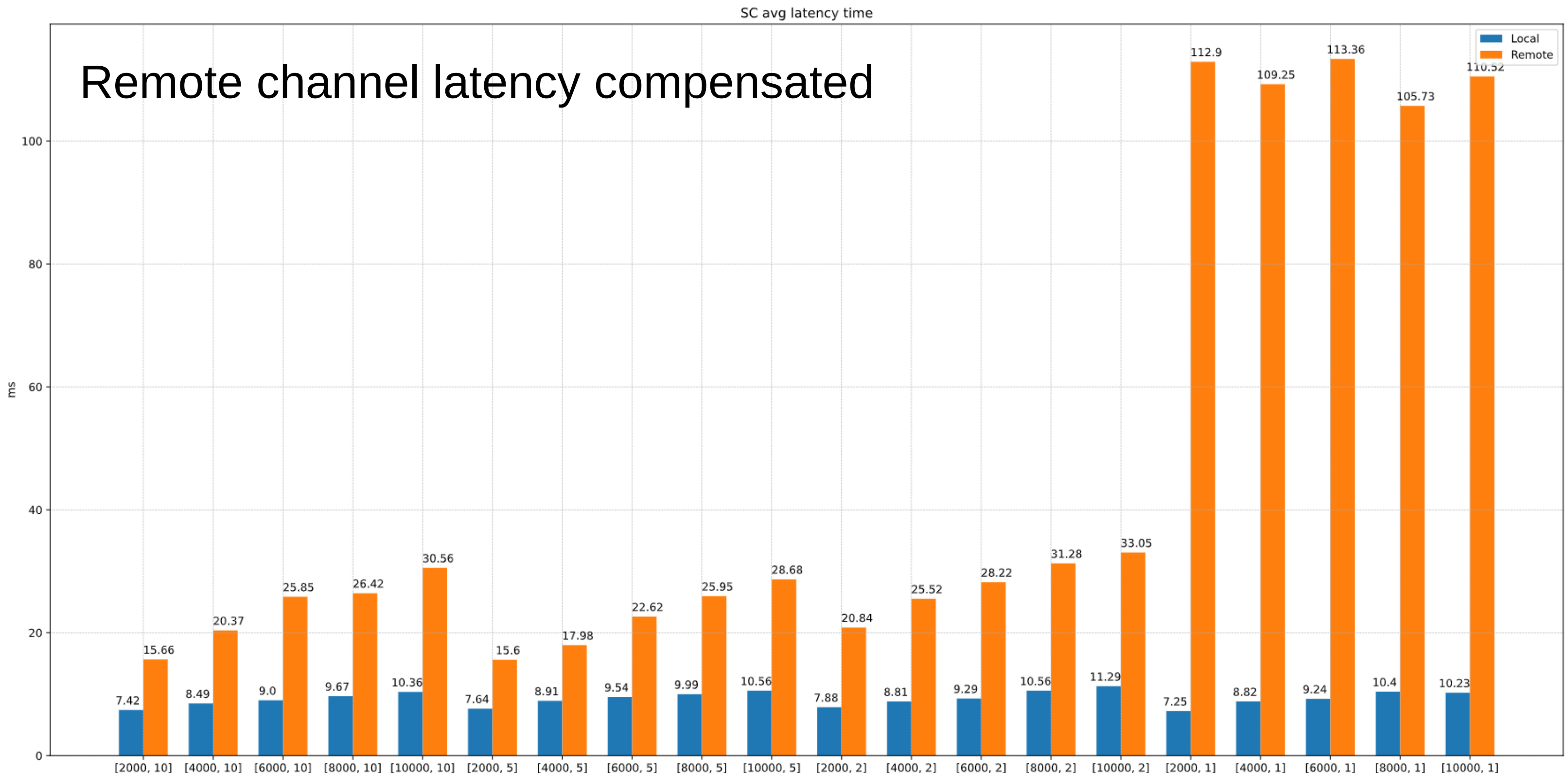
Latency measured by compare EPICS time stamp is around 34 ms.




```
record(bi, "MSC:JITTER") {
    field(DESC, "time jitter")
    field(VAL,  "0")
    field(ZNAM, "0")
    field(ONAM, "1")
}
```

Average Server-Client time latency, Results (1 stream)

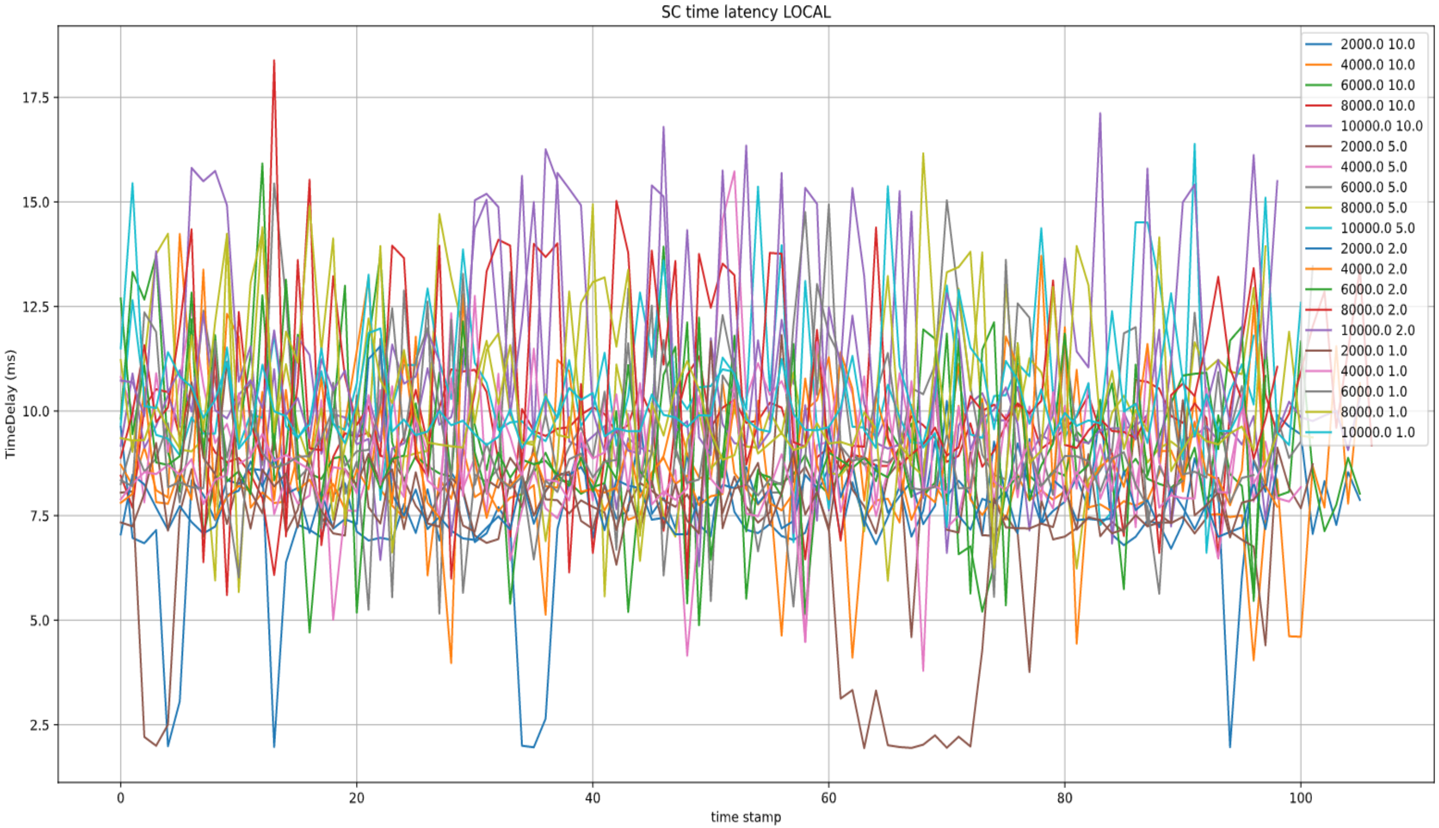
Time in ms.



[PVs, Freq(Hz)]

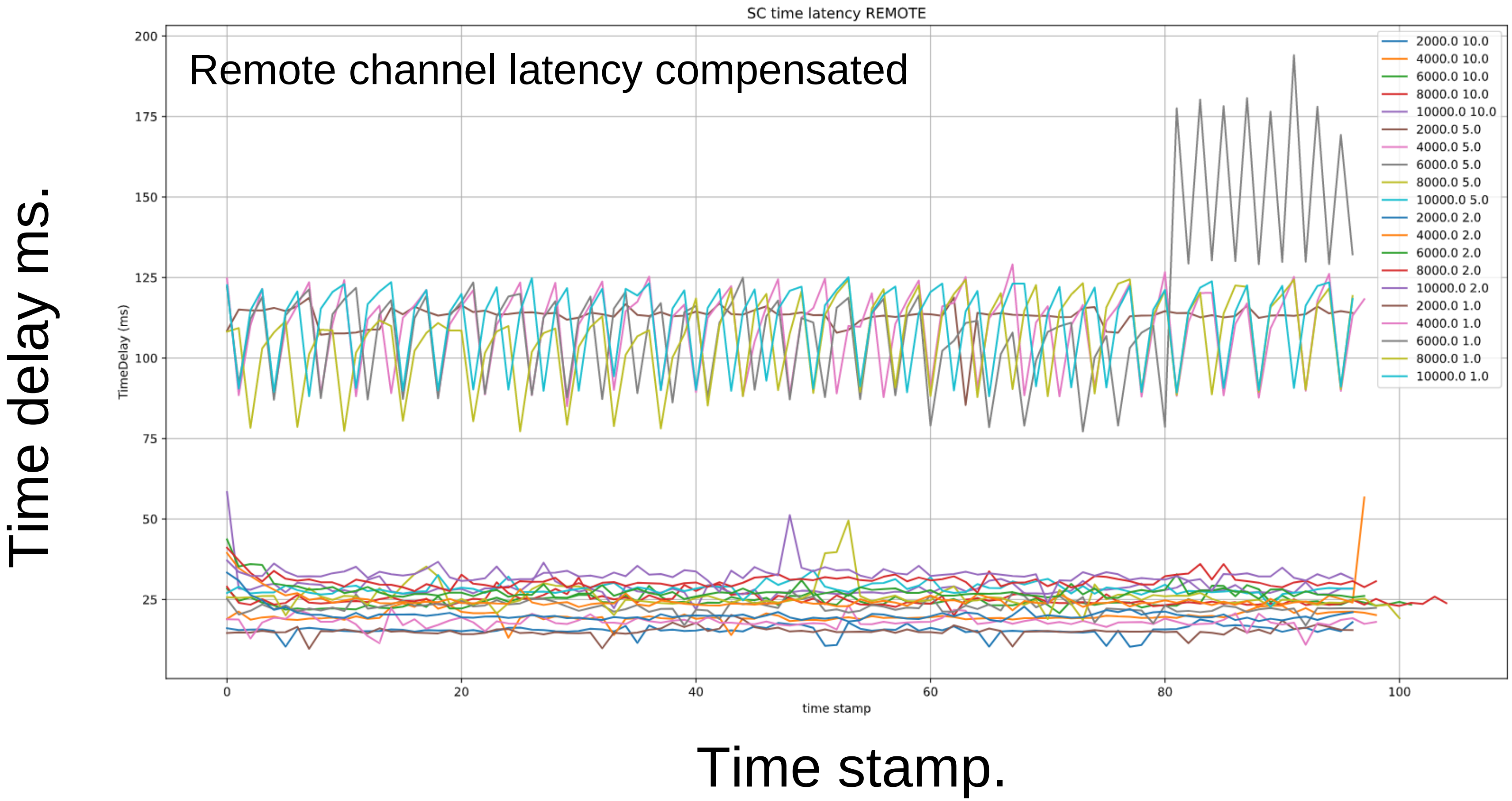
Average Server-Client time latency LOCAL

Time delay ms.



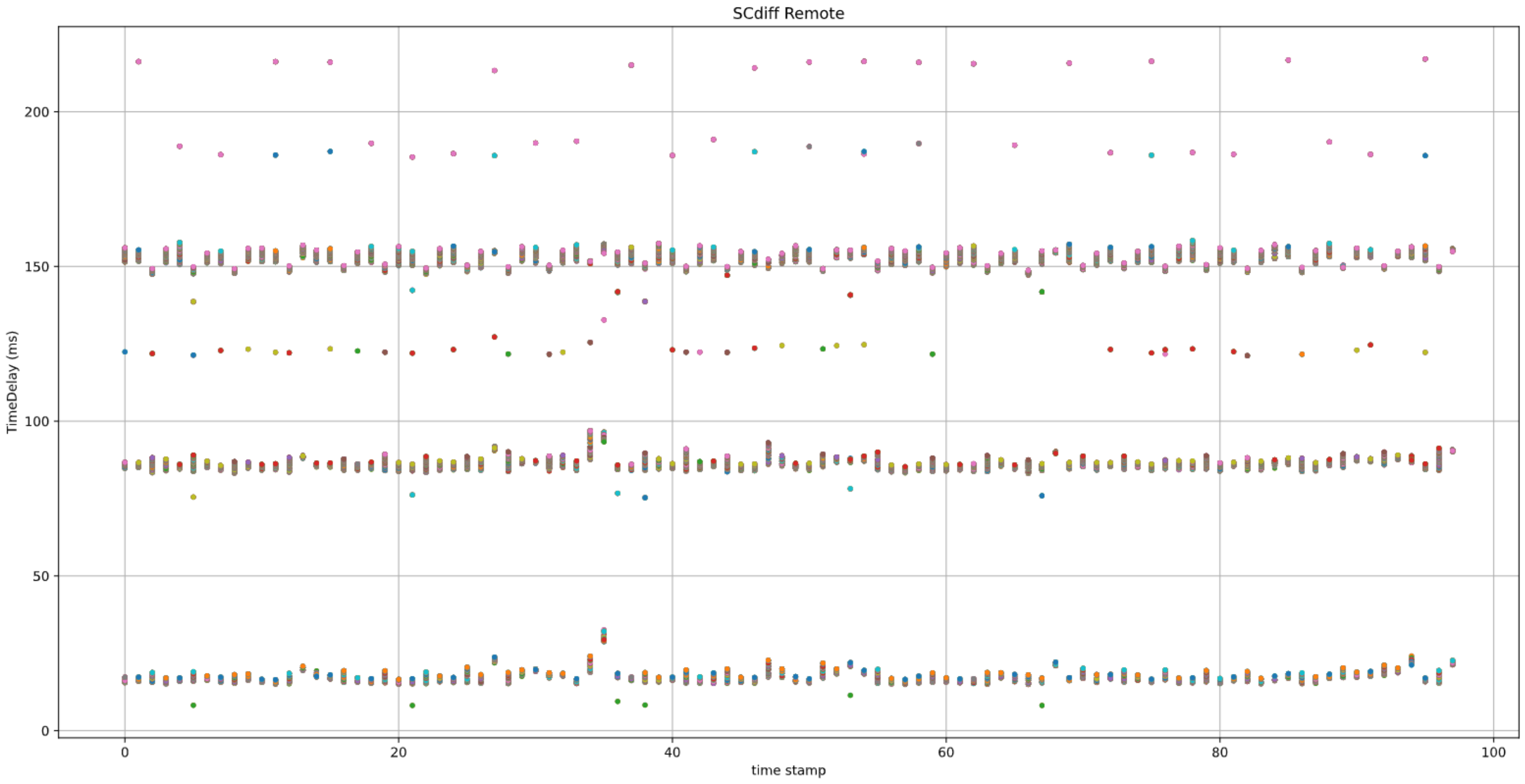
Time stamp.

Average Server-Client time latency REMOTE



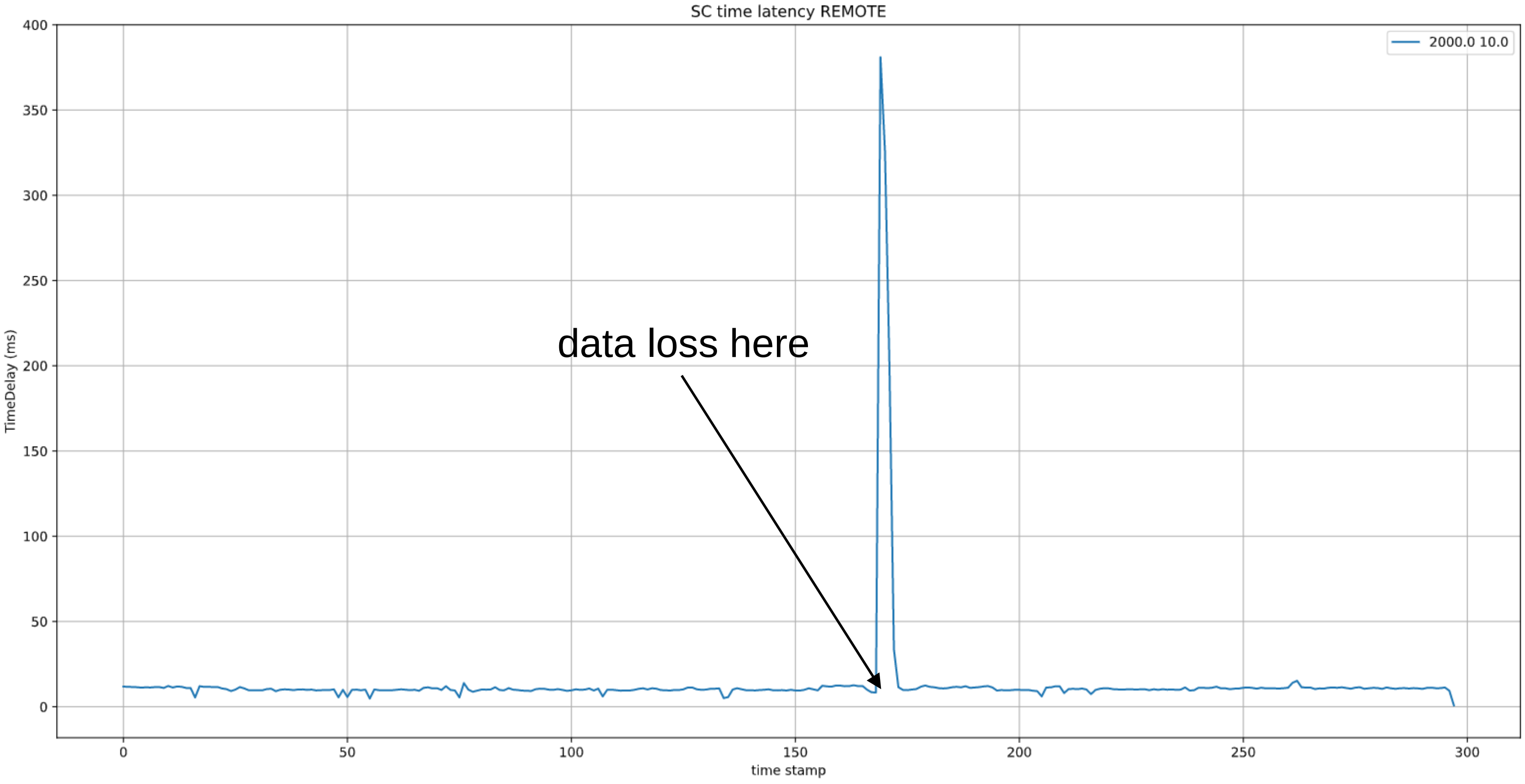
[10000 PV 1Hz] Server Client time latency REMOTE

Time delay ms.



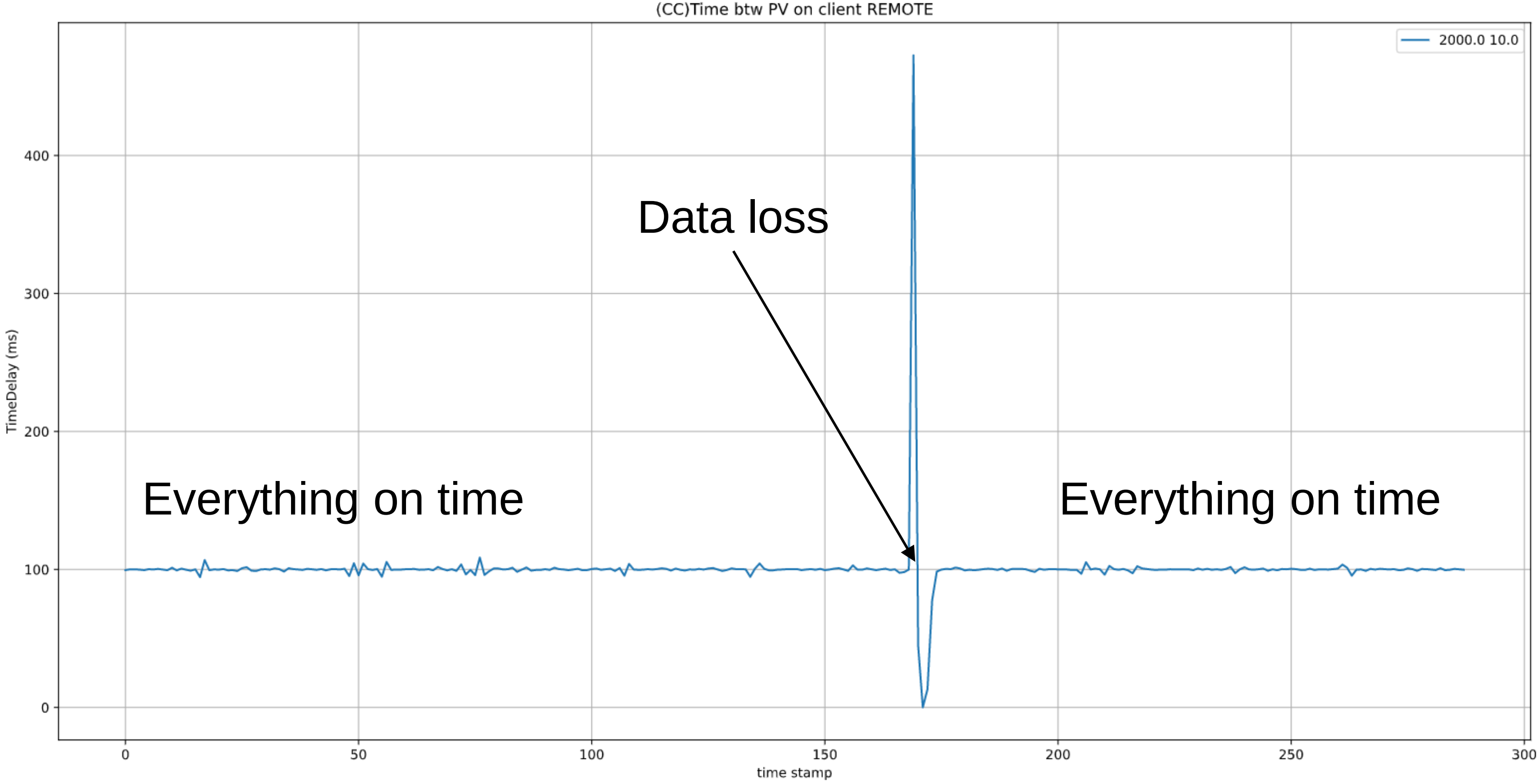
Time stamp.

Loss data cases, Server-client time latency chart



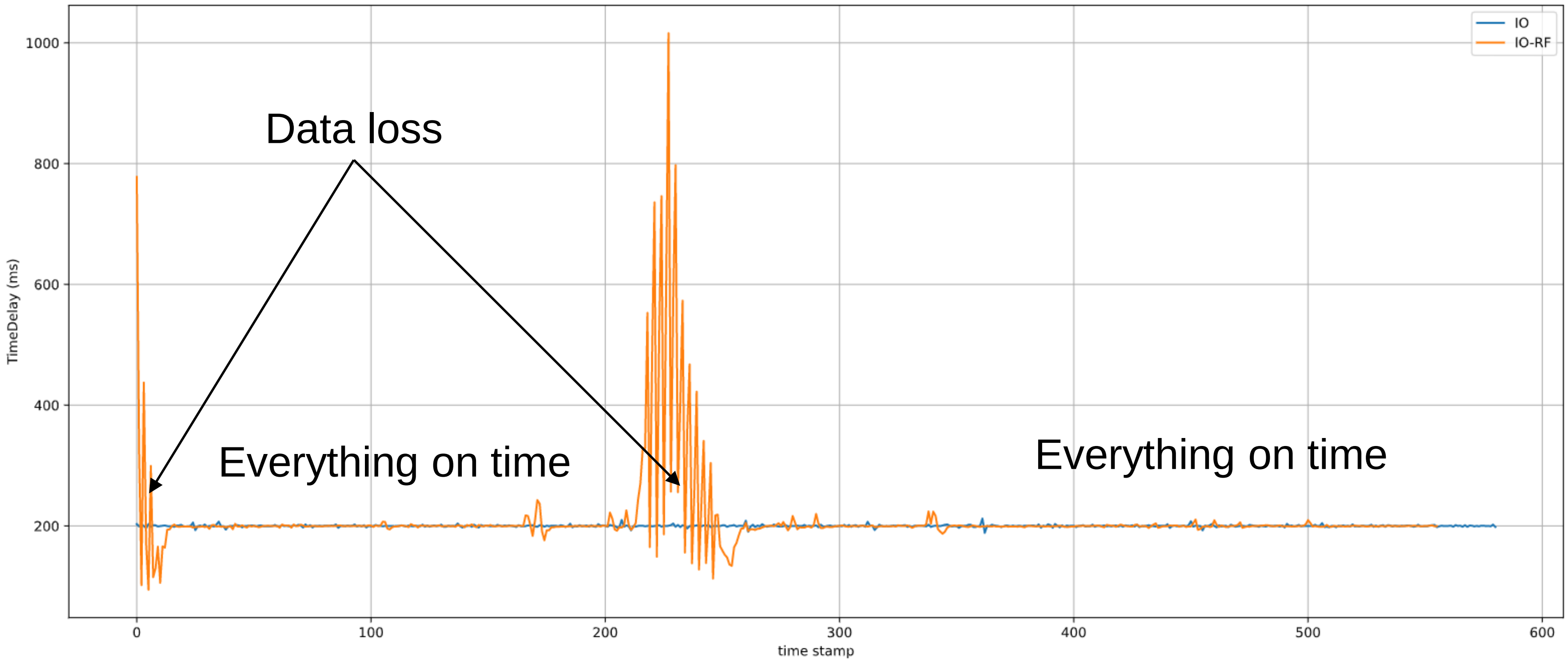
Loss data cases, client-client update time chart

The same place as on the previous slide



Loss data cases, client-client update time chart

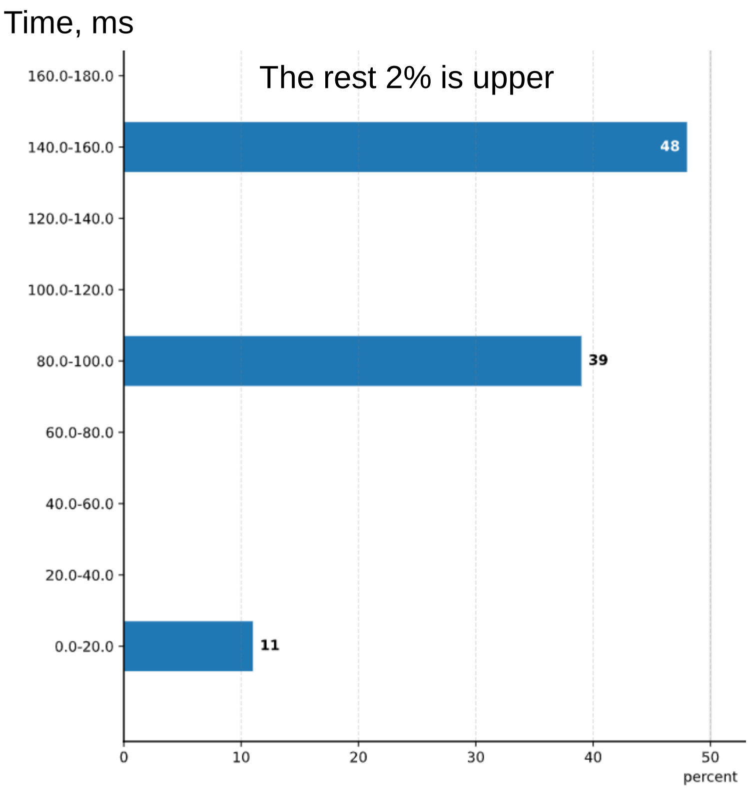
A more serious case [7000 PV, 5Hz]



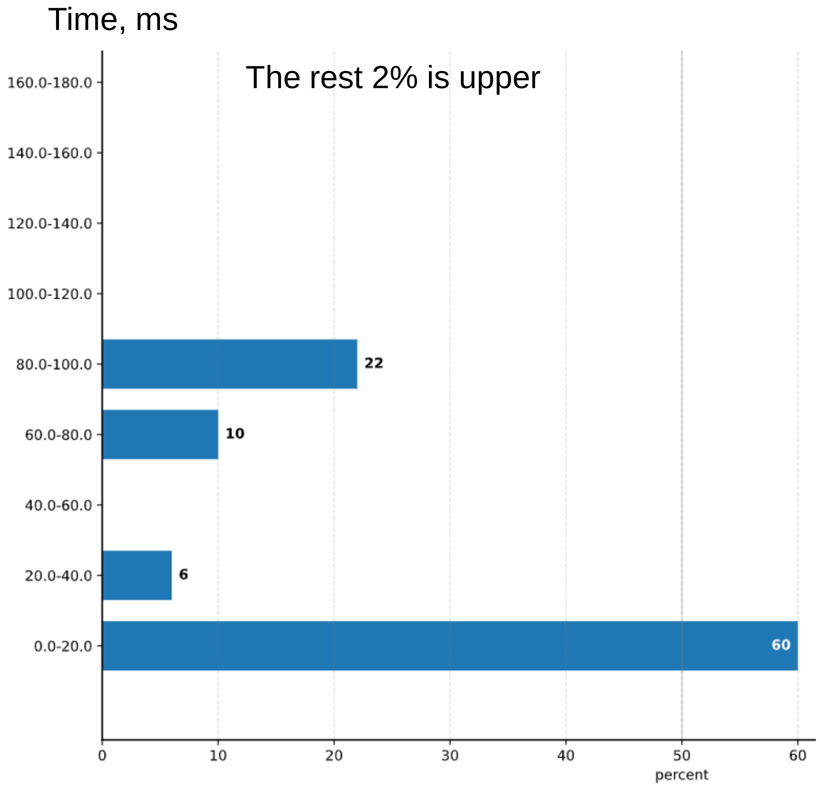
about 10 seconds of not good user experience in the center and few seconds in the beginning.

Multithreading effect

8000 PVs, 1 Hz



1 Thread



32 Thread

Long term test

Treader-long is control:

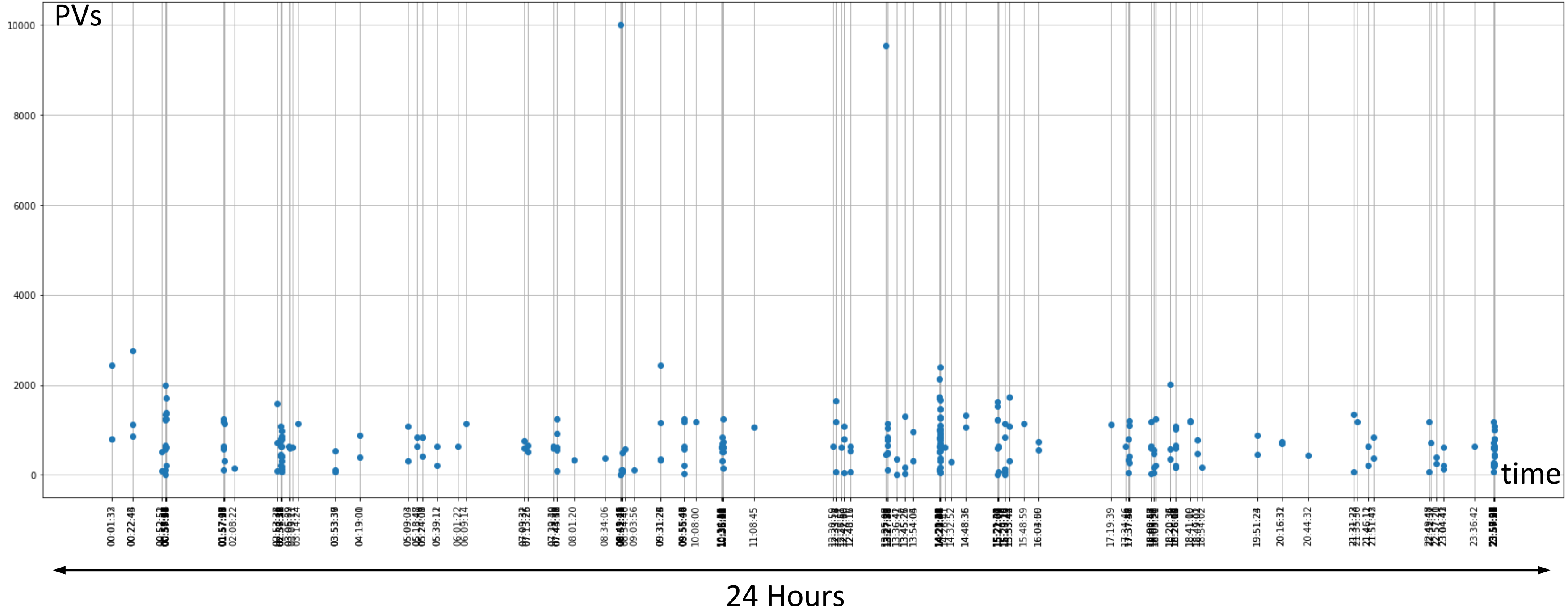
- ~~Timers update frequency~~
- Reading time
- Number of timers to subscribe
- Thread quantity

writing to DISK when error event is occur

tracked **only** PV value order metric

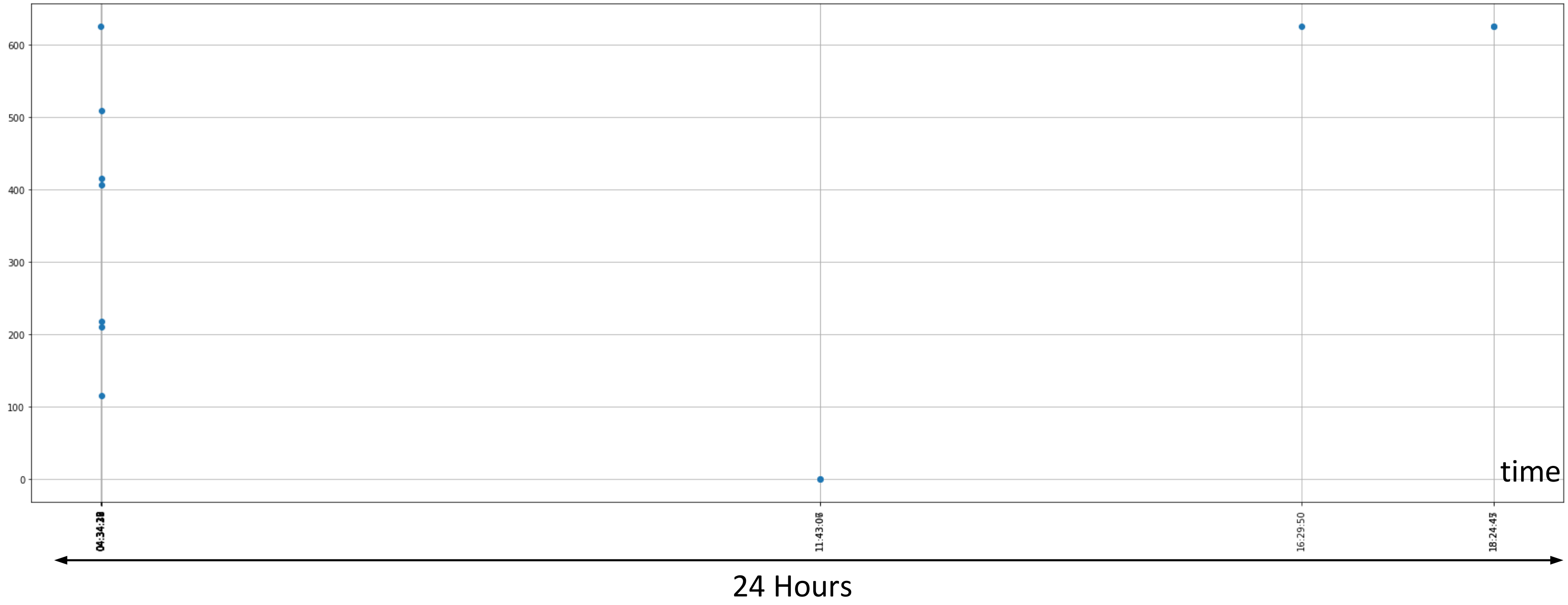
Long-term EPICS reading test

24 Hours EPICS test result, 10 000 PVs 10Hz update rate, 16 reading streams. IO-RF

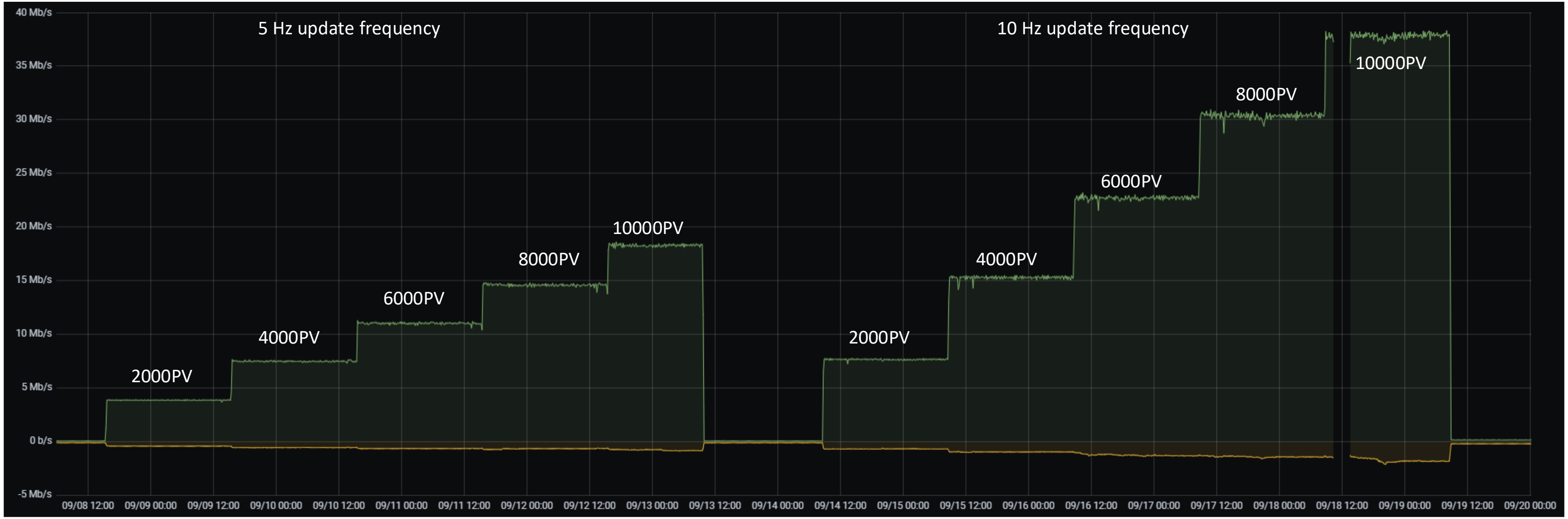


Long-term EPICS reading test

24 Hours EPICS test result, 10 000 PVs 10Hz update rate, 16 reading streams. IO-US



VPN channel load during the testing

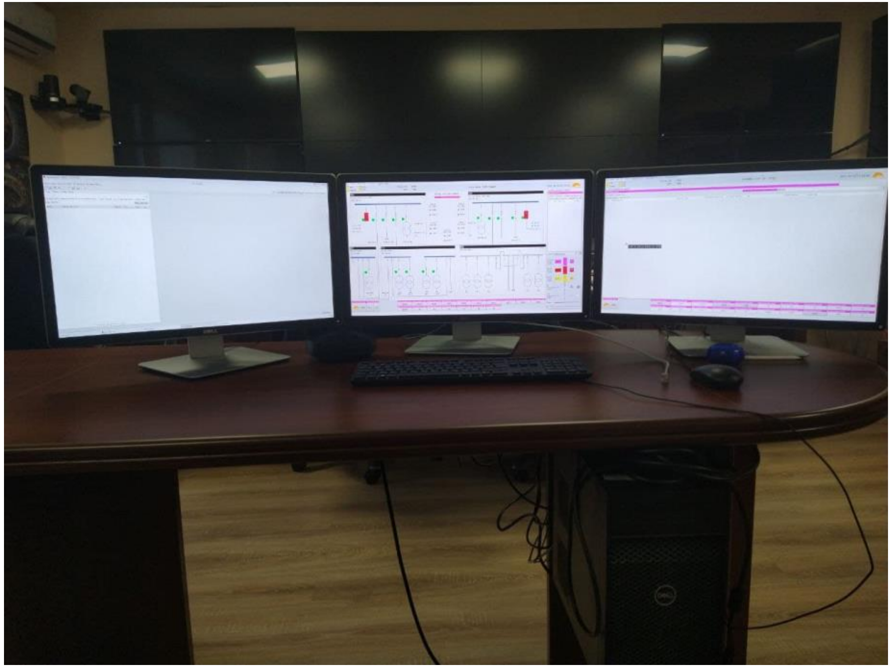


Summary results of long term experiments

	Maximum PV without major errors events.	
Frequency, Hz	IO->RF (less than 5 events per day)	IO->US (less than 5 events per day)
10	2 000	10 000
5	6 000	10 000
2	10 000	10 000
1	10 000	10 000

Practical use case

The practical application of the EPICS over long distances is the organization of remote terminals. As part of work on remote participation in ITER, a remote terminal was organized in RF RPC, technically absolutely identical to those that are installed at ITER site. Tests have shown that the user experience is quite identical to what we have on the ITER side, of course, with the remark that the remote operator can only observe.



Summary

Summary result:

- ❖ During testing, no problems were noticed with the number of PV less than 2000.
- ❖ Since remote centers cannot control any processes, they can only observe, short-term data loss does not affect much to user experience.
- ❖ Multithreading gives a good performance boost in terms of latency update time.
- ❖ Cases of data loss do not depend on the test parameters, and from time to time they occur, which makes it possible to assume that something is happening in the channel.

- ➢ The latency of updates on the side of the remote participant, adjusted to the static network delay due to distance, was demonstrated to be comparable to the latency of local execution. This suggests that EPICS over long distance is quite usable for the purpose of ITER remotes participation tasks.