TWIST FINAL PRESENTATION

NAVISP-075 BEARER INDEPENDENT SECURE TIME TRANSFER

AGENDA

Time	Section
T0	Introduction
T0 + 5mn	Service-Level requirements
T0 + 10mn	System architecture tradeoffs
T0 + 15mn	Secure time-bounding concepts
T0 + 25mn	Demonstration results
T0 + 40mn	Conclusion

INTRODUCTION

1/22/2026
Not referenced

OUTLINE OF THE PROJECT

/// Complement GNSS Synchronization with security

- Low-cost, opportunistic use of Telecom constellation
- Low-performance secure synchronization
- Main use case → Initial synchronization for OS-NMA Users

/// Consortium

- TO = ESA – Gianluca CAPARRA – gianluca.caparra@esa.int
- Prime contractor = **Thales Alenia Space France**
  - Project Manager = Mounia BELHABIB – mounia.belhabib@thalesaleniaspace.com
  - Product Design Architect = Etienne ROUANET-LABE – etienne.rouanet-labe@thalesaleniaspace.com
- Security analysis / Testbed responsible = **Qascom**
  - Security analysis responsible = Luca CANZIAN – luca.canzian@qascom.it
  - Testbed responsible =
    - Federica ROZZI – federica.rozzi@qascom.it
    - Federico CAPUTO – federico.caputo@qascom.it

ACTIVITIES PERFORMED

/// Use case analysis

- Derive KPIs for secure synchronization system
- Assign KPI values to identified use cases

/// System design

- Identify relevant system architectures
- Design secure synchronization algorithms
- Preliminary performance assessment of the concepts

/// Demonstration phase

- Design Sw testbed
- Consolidated performance assessment of the concepts

1/22/2026
Not referenced

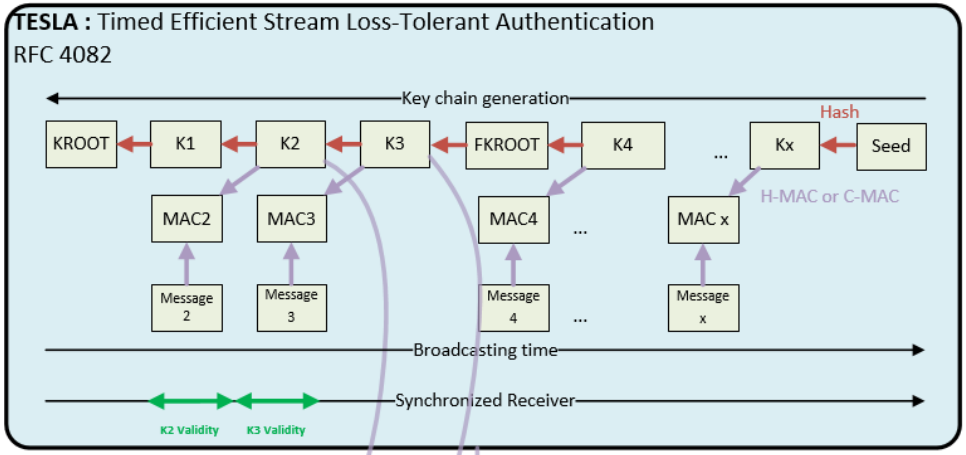
USE CASE STUDY

1/22/2026
Not referenced

MAIN USE CASE – TESLA RECEIVER SYNCHRONIZATION NEED

/// TESLA Keys valid only during a time slot

/// TESLA Keys validity assessed with receiver clock

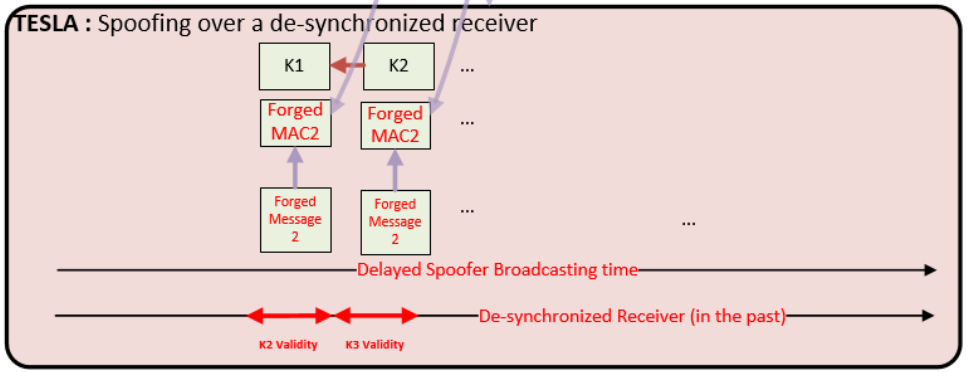


/// Desynchronized receiver may be spoofed if:

- De-synchronized in the past
- De-synchronized of more than half of the broadcasting keys period

/// Spoofer :

- Receive the Keys from the genuine signal
- Forge a Message with a forged MAC from the real Key
- Spoof the keys delayed with forged Message & MAC
- Receiver validates the key with Kroot at forged message reception time



03/04/2023
Not referenced

KPIS FOR SERVICE-LEVEL REQUIREMENTS

/// Demand requirements → ***Needs for telecom resources***

- Coverage area
- User density
- Request period

/// Communication requirements → ***Definition of successful communication***

- Maximum RTT (defined success of a session)
- Session success rate

/// Timing Performance requirements → ***Definition of verifiable synchro. and False / Missed Alarm Rates***

- Time Bound diameter (timing error acceptable for the System)
- PFA / PMD

/// Environment requirements → ***Applicable conditions***

- Masking angle

SUMMARY OF SERVICE-LEVEL REQUIREMENTS

Use Case	Acyclic PRN Acquisition	Waypointing TESLA users	Critical Infrastructure Synchronization
Coverage Area	Global	Global	Continental
Global Demand	10M		
Peak Density	10k/km²		
Session Period	15mn		
Maximum RTT	16s		
Session Success Rate	99.5%		
Time Bound Alarm Limit	10ms	0.5s to 2.5s for future systems 15s for current systems	100ns / 1µs / 10µs / 100µs / 1ms / 50ms depending on use case
Time Bound Risk	1e-7/h	1e-9/h	1e-3/h for Financial app. 1e-7/h for Power Grids
Nominal Timing Accuracy	100µs		
Granularity	1ms		
Availability	99.99%		
Elevation Mask Angle	30°		
Maximum User Velocity	600m/s	600m/s	Static

SYSTEM TRADEOFFS

HOW TO USE A TELECOM CONSTELLATION FOR SECURE TIME ?

/// Option 1 – Use Telecom constellation as a Tunnel – ‘‘Over-The-Top’’

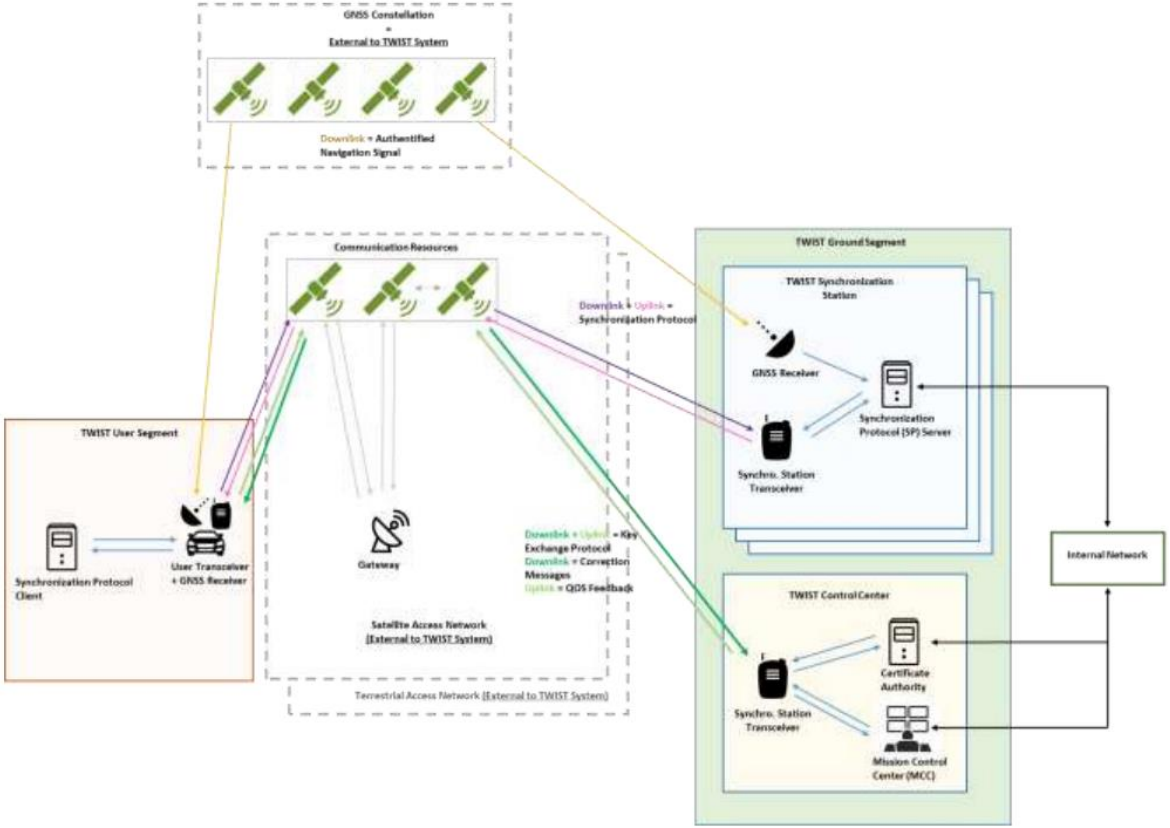
- Servers / Clients = Connected through telecom Network
- Diameter of secure bound = depends on Latency distribution in network

/// Option 2 – Integrate secure synchronization service in Telecom constellation

- Service hosted by telecommunication operator
- Internal information to reduce RTT uncertainty
- Performance depends on RTT uncertainty
  - Timestamping onboard satellites → > Perfo
  - Timestamping at ground → < Perfo

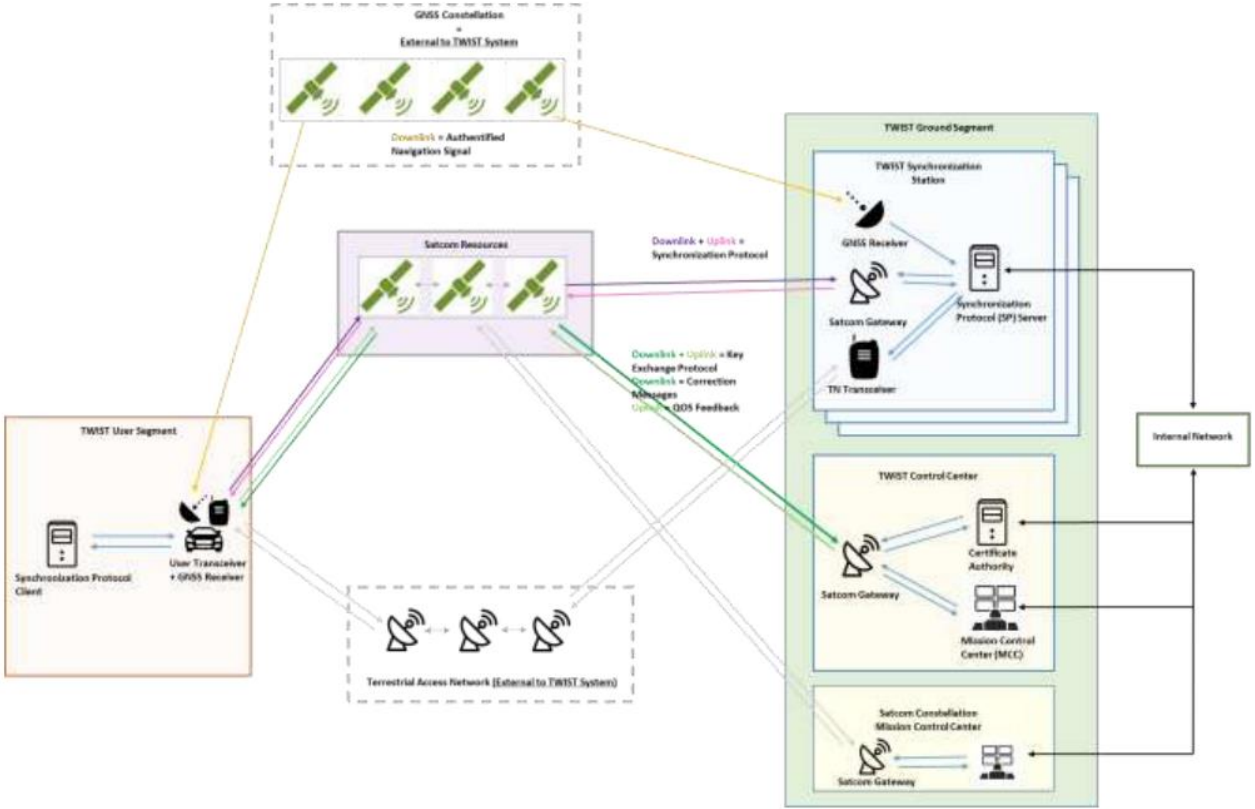
1/22/2026
Not referenced

OTT ARCHITECTURE



1/22/2026

INTEGRATED ARCHITECTURE



PRELIMINARY PERFORMANCE ASSESSMENT

/// Performance of Over-The-Top architecture

- Depends on network RTT distributions

/// Performance of Integrated architecture

- Satellite timestamping
  - Uncertainties = User-Satellite propagation time + Timestamping error

$$d = 4.\sigma_{TS,server} + \Delta RTT_{min-max}$$

- Ground timestamping
  - Uncertainties = User-Satellite propagation time + ISL hops + Processing time

$$d = 4.\sqrt{\sigma^2_{UL-ISL} + \sigma^2_{ISL-DL} + 2 * n_{hop} * \sigma^2_{ISL-ISL} + \sigma^2_{TS,server}} + \Delta RTT_{min-max}$$

/// Theoretical Performance

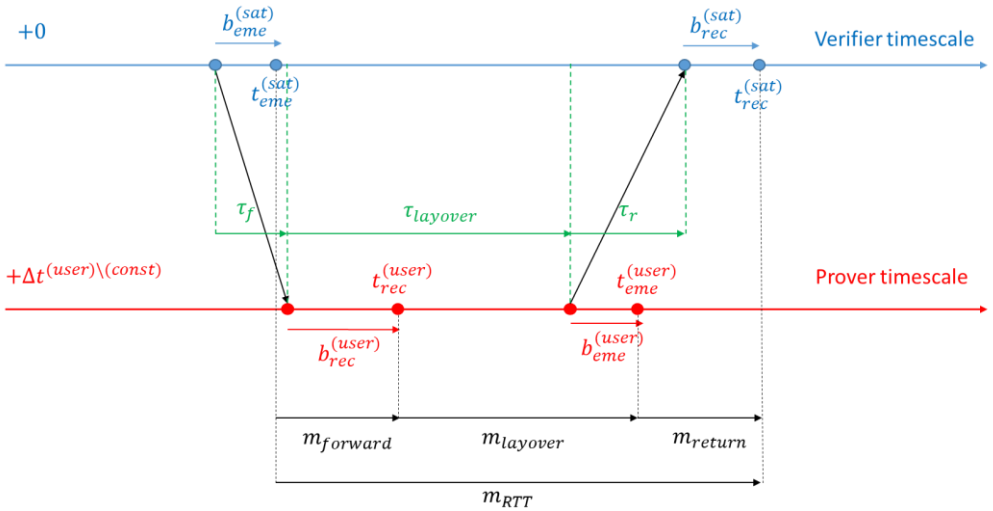
	OTT	Integrated GT BF	Integrated GT SF	Integrated ST BF	Integrated ST SF
Starlink	350 ms / 2M	62.8 ms / 2M	60.0 ms / 2M	20.8 ms / 2M	19.0 ms / 2M
OneWeb	700ms / 2M *(current OS-NMA only)*	54.0 ms / 2M	52.5 ms / 2M	18.5 ms / 2M	17.0 ms / 2M
Iridium	1 s / 2M *(current OS-NMA only)*	58.7 ms / 2M	54.0 ms / 2M	29.4 ms / 2M	25.1 ms / 2M

1/22/2026
Not referenced

TIME BOUNDING ALGORITHMS

PRINCIPLE OF SECURE SYNCHRONIZATION

/// Two-way protocol ensuring RTT security



/// Synchronization as ½(mForward-mReturn)

22/01/2026
Not referenced

PRINCIPLE OF TIME BOUNDING ALGORITHMS

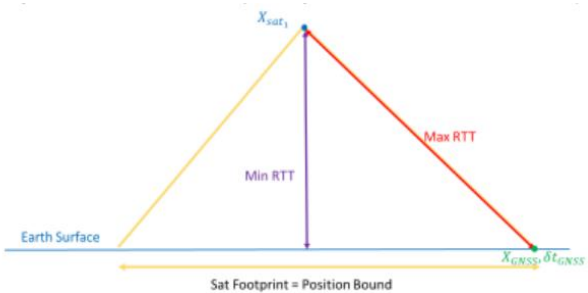
/// Distance-Bounding protocol

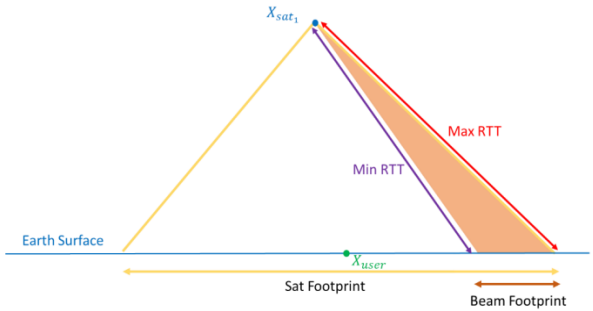
- Alea contained in request
- Signed response

➔ Protection against anticipation attacks

/// Ensure to bound the synchronization of all users within the Satellite / Beam footprint

- Measured RTT > True RTT
- Minimum RTT < True RTT





/// Account for SCER attacks in Time Bound computation

SECURE SYNCHRONIZATION IN OVER-THE-TOP SERVICE

/// Based on assessed distribution of nominal network RTT

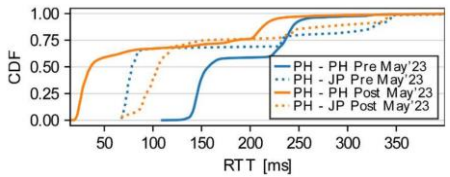
- Same principle – difference between measured RTT and minimum RTT

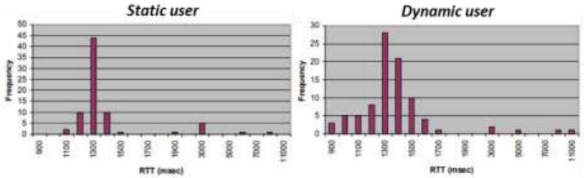
/// Attainable Performance – Time Bound Diameter computed by User

$$d = \widehat{RTT}_{obs} - q_{\widehat{RTT}_{obs}}\left(\frac{\alpha}{2}\right) + 2T_S$$

/// Numerical application – operational constellations

- Starlink
- Iridium





➔ ***< 0.5s of Time Bound diameter***

DEMONSTRATION RESULTS

TESTBED 1: NTP VS. NTS

/// **NTP (Network Time Protocol)** A long-standing protocol used to synchronize clocks over IP networks. It's widely deployed, lightweight, and extremely mature.

/// **NTS (Network Time Security)** A modern security extension to NTP designed to protect time synchronization from tampering, spoofing, and man-in-the-middle attacks.

In the project we evaluated both NTP and NTS: NTP provides basic time synchronization, while NTS adds modern cryptographic security to protect against spoofing and tampering. The main difference lies in NTS's secure key exchange and message authentication, which make it suitable for environments where trust and integrity are critical.

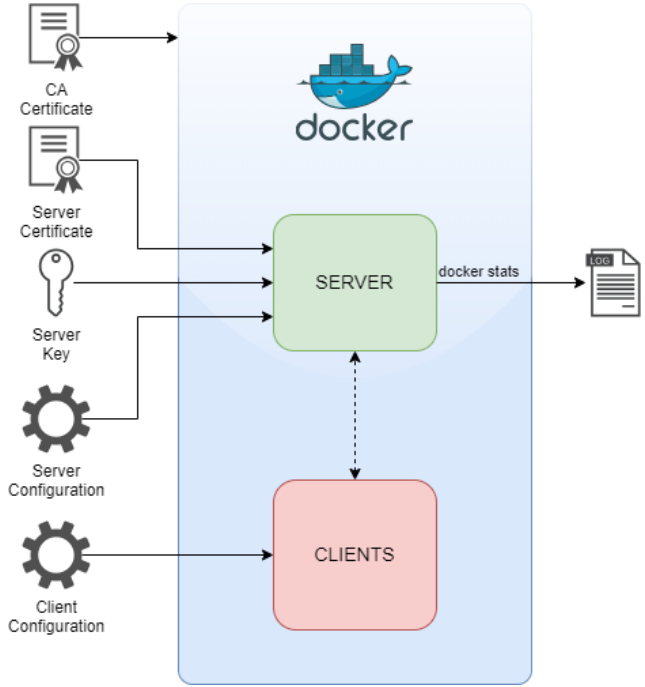
TESTBED 1: OVERVIEW

Key functions:

/// Create an NTS server and a configurable number of NTS clients

/// Monitor server's performance requirements

Launching a variable number of requests allows to assess the way in which the performance requirements grow and to scale to the scenarios of interest.



1/22/2026
Not referenced

TESTBED 1: TESTED CONFIGURATIONS

/// **Number of clients**: simulate different loads.

/// **Protocol**: enables/disables authentication, to assess the impact of the secure protocol.

/// **Delay between clients**: either let clients as fast as allowed, or after a delay for more realistic scenario.

Parameter	Values to Test
Number of clients	[50, 100, 150, 200, 250]
Protocol	[NTS, NTP]
Delay between clients	[0, 1] s

TESTBED 1: OUTPUTS

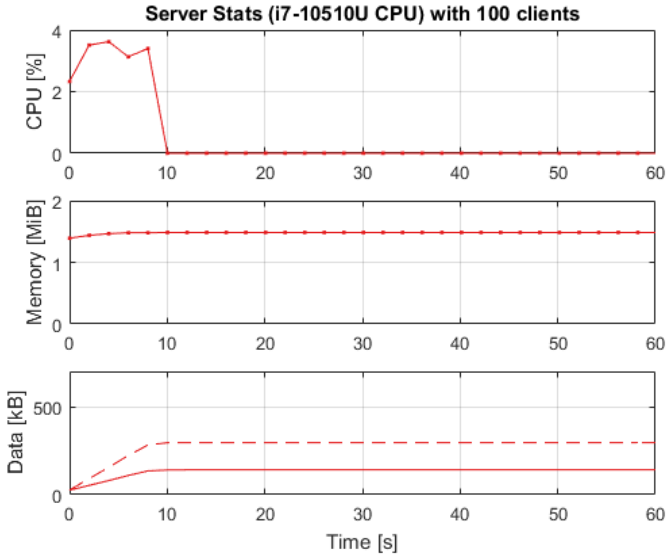
/// **CPU %**, **Memory** and **Data Exchanged** vs **Time Elapsed**

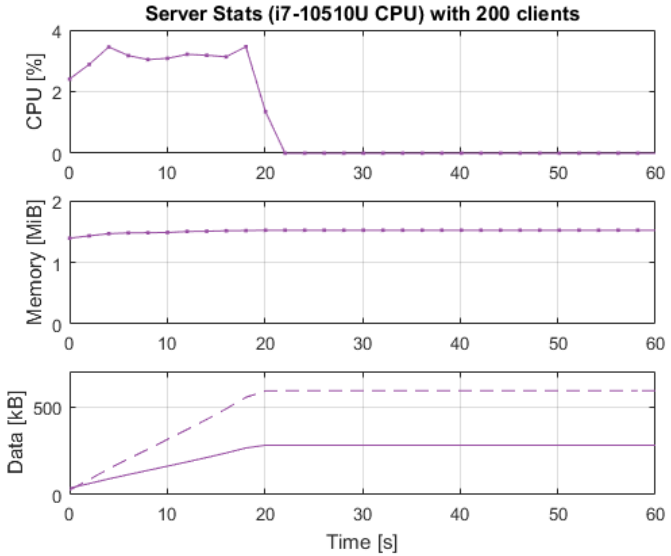
/// **CPU % Integral** and **Memory Max** vs **Number of Clients**

Where "CPU% Integral" is the integral of the CPU% in time, to allow comparison for different loads.

TESTBED 1: RESULTS

/// Resources utilization over time. NTS protocol, no delay.

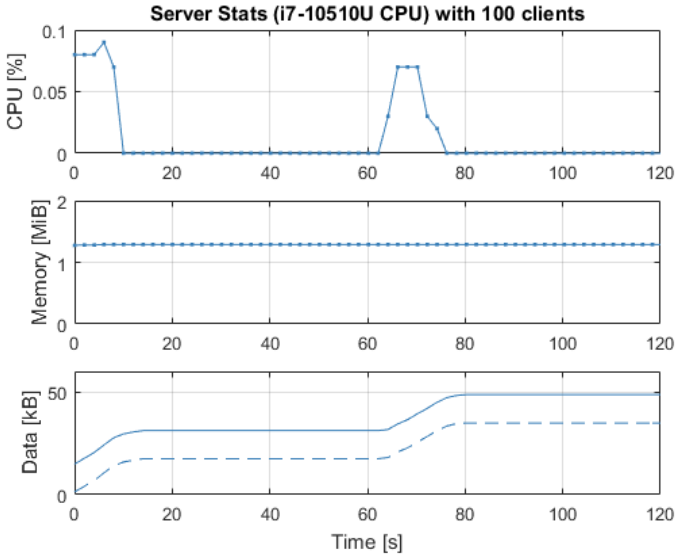




- Load spread over time.
- Memory footprint doesn't change.
- Data sent and received scales linearly.

TESTBED 1: RESULTS

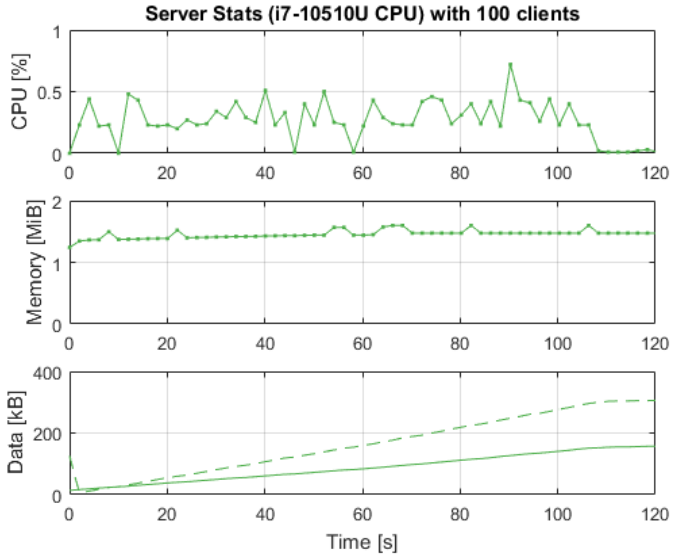
/// Resources utilization over time. NTS (left) vs NTP (right), no delay.



- Initial exchange of encryption keys and cookies. At $t = 0$ there is difference in CPU load while at $t = 60$ is the same.
- Memory usage is very slightly higher for NTS.
- The data exchange exhibits a sharp rise for NTS the first time, second jump is comparable.

TESTBED 1: RESULTS

/// Resources utilization over time. NTS protocol, 0s (left) vs 1s (right) delay.

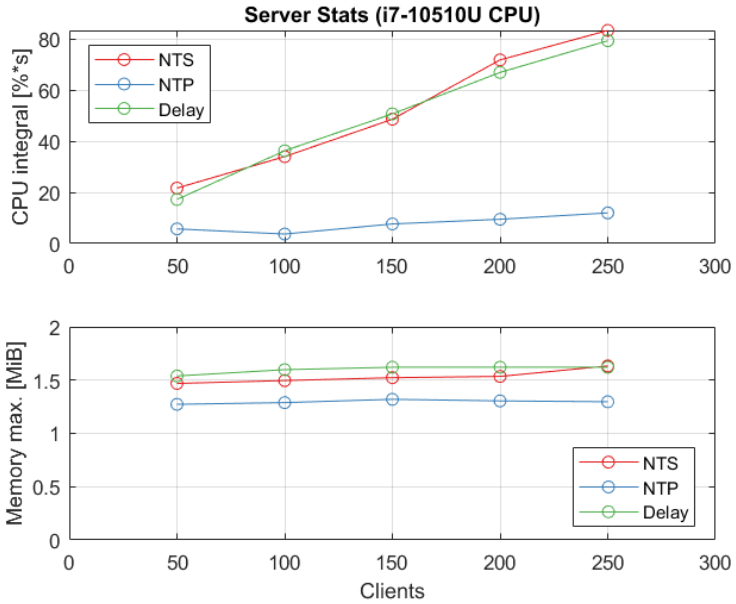


- Spread of CPU load over time (same total effort).
- Memory usage stays the same.
- Amount of data exchanged catches up to the same value.

1/22/2026
Not referenced

TESTBED 1: RESULTS

/// Performance metrics per number of clients.



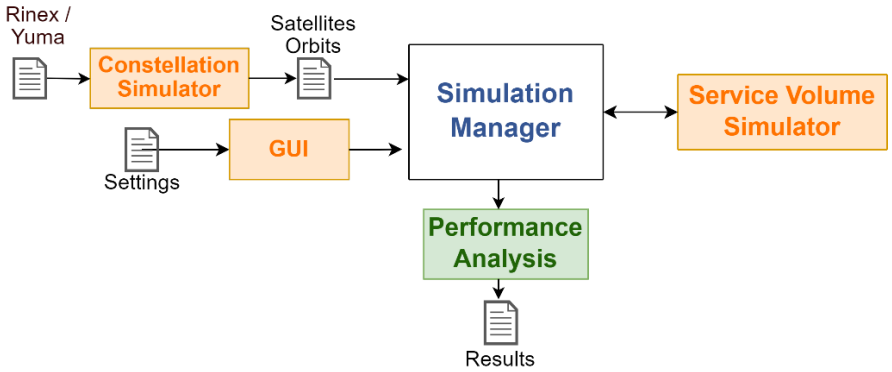
- The CPU integral (total work) differs between NTS and NTP, showing a linear trend.
- Memory stays the same per client. Stateless design.

TESTBED 2: DEMONSTRATOR OVERVIEW

/// Goal

- Assess the performances of the protocol for a target user in nominal scenarios

/// Demonstrator Characteristics



- **GUI**
- **Constellation Simulator**
  - Generate the satellite positions from Rinex/ephemeris data
- **Service Volume Simulator**
  - Two-way communication
    - User uplink request
    - Satellite downlink response
  - Time Bound Estimation
- **Performance Analysis Tool**
  - **System Availability** A_{sys}: percentage of times the algorithm can be applied according to visibility conditions
  - **Target Availability** A_{target}: percentage of time the secure time bound is inside the configured Time Bound Alarm Limit
  - Average value of the time **Time Bound diameter** TB_{avg}

TESTBED 2: SIMULATED PROTOCOL PHASES

/// Calibration phase

- Run at the beginning of the simulation, before synchronization
- Exploits multiple two-way exchanges between user and satellite to calibrate biases
- Calibration done for each user-satellite link

/// Synchronization phase

- Two Synchronization Options
  - Single RTT: Synchronization with 1 RTT (i.e. 1 satellite)
  - Multi RTT: Synchronization with multiple RTTs (i.e. different satellites at multiple time instants)
- User position
  - Known user position: correct RTT by the estimated path delay
  - Unknown user position
    - Assume to known the satellite footprint, which is split into N regions having the same dimension
    - Assume to know in which region the user is located

1/22/2026
Not referenced

TESTBED 2: TUNED PARAMETERS

/// Use case parameters

Parameter	Symbol	Values to Test	Unit of Measure
Time Bound Risk Requirement	TBR_{req}	[1e-7, 1e-9]	/
Time Bound Alarm Limit	t_{alarm}^{req}	[1e-6, 1e-3, 10e-3, 0.5]	[s]
Number of measurements	n_{RTT}	[1, 2, 5, 7, 10]	/
Timestamping error	Δt	[0.1e-6, 1e-6, 10e-6, 1e-3, 10e-3]	[s]
User Position	x_{user}	[known, unknown]	/
Satellite constellation	S_{const}	Iridium, Globalstar	/
Number of regions	N_{reg}	[5, 10]	/

TESTBED 2: TUNED PARAMETERS

/// Use case parameters

/// Protocol parameters

Parameter	Symbol	Values to Test	Unit of Measure
Time Bound Risk Requirement	TBR_{req}	[1e-7, 1e-9]	/
Time Bound Alarm Limit	t_{alarm}^{req}	[1e-6, 1e-3, 10e-3, 0.5]	[s]
Number of measurements	n_{RTT}	[1, 2, 5, 7, 10]	/
Timestamping error	Δt	[0.1e-6, 1e-6, 10e-6, 1e-3, 10e-3]	[s]
User Position	x_{user}	[known, unknown]	/
Satellite constellation	S_{const}	Iridium, Globalstar	/
Number of regions	N_{reg}	[5, 10]	/

TESTBED 2: TUNED PARAMETERS

/// Use case parameters

/// Protocol parameters

/// Visibility parameters

Parameter	Symbol	Values to Test	Unit of Measure
Time Bound Risk Requirement	TBR_{req}	[1e-7, 1e-9]	/
Time Bound Alarm Limit	t_{alarm}^{req}	[1e-6, 1e-3, 10e-3, 0.5]	[s]
Number of measurements	n_{RTT}	[1, 2, 5, 7, 10]	/
Timestamping error	Δt	[0.1e-6, 1e-6, 10e-6, 1e-3, 10e-3]	[s]
User Position	x_{user}	[known, unknown]	/
Satellite constellation	S_{const}	Iridium, Globalstar	/
Number of regions	N_{reg}	[5, 10]	/

TESTBED 2: TUNED PARAMETERS

/// Use case parameters

/// Protocol parameters

/// Visibility parameters

/// Simulation duration of 72h, protocol rate of 15min

Parameter	Symbol	Values to Test	Unit of Measure
Time Bound Risk Requirement	TBR_{req}	[1e-7, 1e-9]	/
Time Bound Alarm Limit	t_{alarm}^{req}	[1e-6, 1e-3, 10e-3, 0.5]	[s]
Number of measurements	n_{RTT}	[1, 2, 5, 7, 10]	/
Timestamping error	Δt	[0.1e-6, 1e-6, 10e-6, 1e-3, 10e-3]	[s]
User Position	x_{user}	[known, unknown]	/
Satellite constellation	S_{const}	Iridium, Globalstar	/
Number of regions	N_{reg}	[5, 10]	/

TESTBED 2: RESULTS BASELINE SCENARIO

/// Baseline scenario

- Single RTT, known user position, use case: $t_{alarm}^{req} = 10\text{ms}$; $TBR_{req} = 1e^{-7}/\text{h}$
- $TB_{avg} \simeq 2\text{ms}$ for Iridium, $TB_{avg} \simeq 0.22\text{ms}$ for Globalstar
  - Different order of magnitude due to the different symbol time ($T_S = 1\text{ms}$ for Iridium, $T_S \simeq 0.1\text{ms}$ for Globalstar)

ID	S_{const}	n_{RTT}	TBR_{req}	t_{alarm}^{req} [s]	Δt [s]	x_{user}	N_{reg}	A_{sys} [%]	A_{target} [%]	TB_{avg}[s]
1	Iridium	1	1e-7	10e-3	1e-6	know n	/	100	100	2.0159e-03
2	Globalstar	1	1e-7	10e-3	1e-6	know n	/	100	100	2.2389e-04
3	Iridium	1	1e-7	10e-3	0.1e-6	know n	/	100	100	2.0016e-03
4	Globalstar	1	1e-7	10e-3	0.1e-6	know n	/	100	100	2.0990e-04
5	Iridium	1	1e-7	10e-3	10e-6	know n	/	100	100	2.1571e-03
6	Globalstar	1	1e-7	10e-3	10e-6	know n	/	100	100	3.6716e-04
7	Iridium	1	1e-7	10e-3	1e-3	know n	/	100	0.35	1.7788e-02
8	Globalstar	1	1e-7	10e-3	1e-3	know n	/	100	0.70	1.5705e-02
9	Iridium	1	1e-7	10e-3	10e-3	know n	/	100	0	1.5725e-01
10	Globalstar	1	1e-7	10e-3	10e-3	know n	/	100	0	1.5515e-01

TESTBED 2: TUNE TIMESTAMPING ERROR

- A higher timestamping error increases the TB width and vice versa
- Results
  - Iridium: minimum $TB_{avg} \simeq 2$ms for $\Delta t = 0.1$µs, maximum $TB_{avg} \simeq 0.157$s for $\Delta t = 10$ms
  - Globalstar: minimum $TB_{avg} \simeq 0.2$ms for $\Delta t = 0.1$µs, maximum $TB_{avg} \simeq 0.155$s for $\Delta t = 10$ms
  - When $\Delta t \leq T_S$, TB is driven by the symbol period → difference between Iridium and Globalstar results
  - When $\Delta t \geq T_S$, results are driven by the timestamping error → results similar for the two constellations
- Target availability $A_{target} < 100\%$ for Δt at ms level

ID	S_{const}	n_{RTT}	TBR_{req}	t_{alarm}^{req} [s]	Δt [s]	x_{user}	N_{reg}	A_{sys} [%]	A_{target} [%]	TB_{avg}[s]
1	Iridium	1	1e-7	10e-3	1e-6	know n	/	100	100	2.0159e-03
2	Globalstar	1	1e-7	10e-3	1e-6	know n	/	100	100	2.2389e-04
3	Iridium	1	1e-7	10e-3	0.1e-6	know n	/	100	100	2.0016e-03
4	Globalstar	1	1e-7	10e-3	0.1e-6	know n	/	100	100	2.0990e-04
5	Iridium	1	1e-7	10e-3	10e-6	know n	/	100	100	2.1571e-03
6	Globalstar	1	1e-7	10e-3	10e-6	know n	/	100	100	3.6716e-04
7	Iridium	1	1e-7	10e-3	1e-3	know n	/	100	0.35	1.7788e-02
8	Globalstar	1	1e-7	10e-3	1e-3	know n	/	100	0.70	1.5705e-02
9	Iridium	1	1e-7	10e-3	10e-3	know n	/	100	0	1.5725e-01
10	Globalstar	1	1e-7	10e-3	10e-3	know n	/	100	0	1.5515e-01

TESTBED 2: UNKNOWN USER POSITION

/// Unknown user position

- Assumption: knowledge of the region where the user is located
- Tuned the number of regions (5/10) with elevation mask $\theta_{mask}^{el} = 10°$

/// Results

- RTT cannot be compensated by the estimated delay
- Exploit the minimum delay of the footprint region where the user is located

ID	S_{const}	n_{RTT}	TBR_{req}	t_{alarm}^{req} [s]	Δt [s]	x_{user}	N_{reg}	A_{sys} [%]	A_{target} [%]	TB_{avg}[s]
11	Iridium	1	1e-7	10e-3	1e-6	unknow n	5	100	100	2.9738e-03
12	Globalstar	1	1e-7	10e-3	1e-6	unknow n	5	100	100	1.1984e-03
13	Iridium	1	1e-7	10e-3	1e-6	unknow n	10	100	100	2.5737e-03
14	Globalstar	1	1e-7	10e-3	1e-6	unknow n	10	100	100	7.1735e-04
15	Iridium	1	1e-7	10e-3	1e-6	unknow n	5	100	100	2.5945e-03
16	Globalstar	1	1e-7	10e-3	1e-6	unknow n	5	100	100	8.0180e-04
17	Iridium	1	1e-7	10e-3	1e-6	unknow n	10	100	100	2.5052e-03
18	Globalstar	1	1e-7	10e-3	1e-6	unknow n	10	100	100	7.1345e-04

TESTBED 2: TUNE #RTT MEASUREMENTS

/// Number of measurements $n_{RTT} > 1$ weights the timestamping errors and slightly reduces the TB

/// Iridium

- the TB remains around from 2ms with 1 RTT (ID #1) or 10 RTT (ID #31) with $\Delta t = 1\mu s$
- the TB moves from 18ms (ID #7) to 7ms (ID #26) when using 10 RTTs and $\Delta t = 1ms$
- $A_{target} \simeq 0.35\%$ with $\Delta t = 1ms$, $n_{RTT} = 1$ →TB is tents of ms, not comparable with $t_{alarm}^{req} = 10ms$
- $A_{target} = 100\%$ with $\Delta t = 1ms$, $n_{RTT} = 10$

ID	S_{const}	n_{RTT}	TBR_{req}	t_{alarm}^{req} [s]	Δt [s]	x_{user}	N_{reg}	A_{sys} [%]	A_{target} [%]	TB_{avg}[s]
1	Iridium	1	1e-7	10e-3	1e-6	know n	/	100	100	2.0159e-03
7	Iridium	1	1e-7	10e-3	1e-3	know n	/	100	0.35	1.7788e-02
19	Iridium	2	1e-7	10e-3	1e-6	know n	/	100	100	2.0111e-03
20	Iridium	5	1e-7	10e-3	1e-6	know n	/	100	100	2.0073e-03
21	Iridium	7	1e-7	10e-3	1e-6	know n	/	100	100	2.0060e-03
22	Iridium	10	1e-7	10e-3	1e-6	know n	/	100	100	2.0050e-03
23	Iridium	2	1e-7	10e-3	1e-3	know n	/	100	2.82	1.3101e-02
24	Iridium	5	1e-7	10e-3	1e-3	know n	/	100	77.89	9.1999e-03
25	Iridium	7	1e-7	10e-3	1e-3	know n	/	100	93.66	8.1706e-03
26	Iridium	10	1e-7	10e-3	1e-3	know n	/	100	100	7.0656e-03

TESTBED 2: TUNE #RTT MEASUREMENTS

/// $n_{RTT} > 1$ weights the timestamping errors and slightly reduces the TB

- Globalstar
  - the TB moves from 0.223ms (ID #2) to 0.213ms (ID #30) when using 10 RTTs and $\Delta t = 1\mu s$
  - the TB moves from 16ms (ID #8) to 5ms (ID #34) when using 10 RTTs and $\Delta t = 1ms$
  - $A_{target} \simeq 0.7\%$ with $\Delta t = 1ms$, $n_{RTT} = 1$
  - $A_{target} = 100\%$ with $\Delta t = 1ms$, $n_{RTT} = 7/10$

ID	S_{const}	n_{RTT}	TBR_{req}	t_{alarm}^{req} [s]	Δt [s]	x_{user}	N_{reg}	A_{sys} [%]	A_{target} [%]	TB_{avg}[s]
2	Globalstar	1	1e-7	10e-3	1e-6	know n	/	100	100	2.2389e-04
8	Globalstar	1	1e-7	10e-3	1e-3	know n	/	100	0.70	1.5705e-02
27	Globalstar	2	1e-7	10e-3	1e-6	know n	/	100	100	2.1944e-04
28	Globalstar	5	1e-7	10e-3	1e-6	know n	/	100	100	2.1520e-04
29	Globalstar	7	1e-7	10e-3	1e-6	know n	/	100	100	2.1436e-04
30	Globalstar	10	1e-7	10e-3	1e-6	know n	/	100	100	2.1333e-04
31	Globalstar	2	1e-7	10e-3	1e-3	know n	/	100	20.77	1.1244e-02
32	Globalstar	5	1e-7	10e-3	1e-3	know n	/	100	98.24	7.4585e-03
32	Globalstar	7	1e-7	10e-3	1e-3	know n	/	100	100	6.1960e-03
34	Globalstar	10	1e-7	10e-3	1e-3	know n	/	100	100	5.1565e-03

TESTBED 2: SIMULATION RESULTS

/// Tune use cases

Waypointing TESLA users

- $t_{alarm}^{req} = 0.5$s and $TBR_{req} = 1e^{-9}$/h
- Looser time requirement → all cases satisfy the constraint, even with $\Delta t = 1$ms

ID	S_{const}	n_{RTT}	TBR_{req}	t_{alarm}^{req} [s]	Δt [s]	x_{user}	N_{reg}	A_{sys} [%]	A_{target} [%]	TB_{avg}[s]
35	Iridium	1	1e-9	0.5	1e-6	know n	/	100	100	2.0182e-03
36	Globalstar	1	1e-9	0.5	1e-6	know n	/	100	100	2.2658e-04
37	Iridium	1	1e-9	0.5	1e-3	know n	/	100	100	1.9806e-02
38	Globalstar	1	1e-9	0.5	1e-3	know n	/	100	100	1.8334e-02
39	Iridium	5	1e-9	0.5	1e-6	know n	/	100	100	2.0089e-03
40	Globalstar	5	1e-9	0.5	1e-6	know n	/	100	100	2.1718e-04
41	Iridium	5	1e-9	0.5	1e-3	know n	/	100	100	1.0783e-02
42	Globalstar	5	1e-9	0.5	1e-3	know n	/	100	100	9.1187e-03
43	Iridium	10	1e-9	0.5	1e-6	know n	/	100	100	2.0065e-03
44	Globalstar	10	1e-9	0.5	1e-6	know n	/	100	100	2.1484e-04
45	Iridium	10	1e-9	0.5	1e-3	know n	/	100	100	8.4871e-03
46	Globalstar	10	1e-9	0.5	1e-3	know n	/	100	100	6.7912e-03

TESTBED 2: SIMULATION RESULTS

/// Tune use cases

Power grids

- $t_{alarm}^{req} = 1$ms and $TBR_{req} = 1e^{-7}$/h
- or $t_{alarm}^{req} = 1$µs and $TBR_{req} = 1e^{-7}$/h
- Results in both scenarios are the same, the only difference is t_{alarm}^{req}
  - t_{alarm}^{req} is too tight in the second case → availability is always null

ID	S_{const}	n_{RTT}	TBR_{req}	t_{alarm}^{req} [s]	Δt [s]	x_{user}	N_{reg}	A_{sys} [%]	A_{target} [%]	TB_{avg}[s]
47	Iridium	1	1e-7	1e-3	1e-6	know n	/	100	0	2.0156e-03
48	Globalstar	1	1e-7	1e-3	1e-6	know n	/	100	100	2.2382e-04
49	Iridium	1	1e-7	1e-3	1e-3	know n	/	100	0	1.7911e-02
50	Globalstar	1	1e-7	1e-3	1e-3	know n	/	100	0	1.5993e-02
51	Iridium	5	1e-7	1e-3	1e-6	know n	/	100	0	2.0073e-03
52	Globalstar	5	1e-7	1e-3	1e-6	know n	/	100	100	2.1556e-04
53	Iridium	5	1e-7	1e-3	1e-3	know n	/	100	0	9.1737e-03
54	Globalstar	5	1e-7	1e-3	1e-3	know n	/	100	0	7.4283e-03
55	Iridium	10	1e-7	1e-3	1e-6	know n	/	100	0	2.0049e-03
56	Globalstar	10	1e-7	1e-3	1e-6	know n	/	100	100	2.1329e-04
57	Iridium	10	1e-7	1e-3	1e-3	know n	/	100	0	7.0511e-03
58	Globalstar	10	1e-7	1e-3	1e-3	know n	/	100	0	5.1949e-03

CONCLUSION

1/22/2026
Not referenced

SUMMARY

/// Opportunistic use of Telecom constellation = Low-cost Low-Performance secure synchronization

- Over-The-Top
  - Cost ++ – Subscribe to sat telecom network
  - Performance -
    - Time Bound Diameter < 0.5s
    - Varying Performance – Constellation updates / Region-dependent Performance
  - Feasibility ++
- Integrated
  - Cost - – for Telecom constellation operator
  - Performance -
    - Time Bound Diameter < 50ms (Use of Software Timestamping)
  - Feasibility – depends on telecom constellation operator services

/// Dedicated Two-Way Ranging system = High-Performance, better use case coverage, higher cost

- Hw timestamping at satellite → ~ns of timestamping accuracy
- Short symbol waveform → Robustness to SCER attacks ~µs of Time Bound inflation
- Overvall Performance → sub-microsecond of Time Bound Diameter