Performance evaluation & monitoring of HIL test systems – Online algorithm for arrival- and service-curve estimation

Christoph Funda, Pablo Garcia Marin, Kai-Steffen Hielscher, Reinhard German, WueWoWas'23
ZF Mobility Solutions GmbH & Friedrich-Alexander University Erlangen-Nuremberg

Introduction

Big Picture

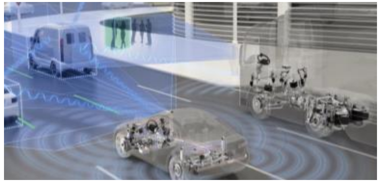
- Development of autonomous transit systems (**ATS**) requires **massive testing** with a solid test strategy compliant to ISO26262 and ISO21448
- Strong demand for **reliable** and **high-performant** Hardware-in-the-Loop (**HIL**) test systems for ATS validation

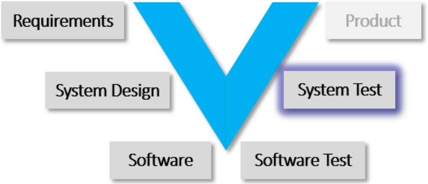
Motivation

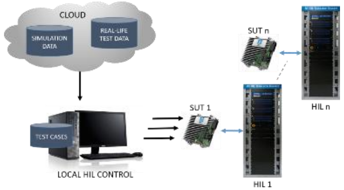
- Early identification of **service performance** & **bottlenecks** in test system architecture with potential impact on test quality
- Design **pre-buffer time** and **buffer sizes** to ensure high Quality of Service (QoS) of HIL test system for ATS serial validation
- **Time** and **cost** reduction in HIL development & operation

Proposed Solution:

- **Framework** of performance evaluation methods for HIL testsystems (based on simulation, measurements, analytics)
- Monitoring of HIL test systems to verify strict real-time & QoS requirements for ATS validation

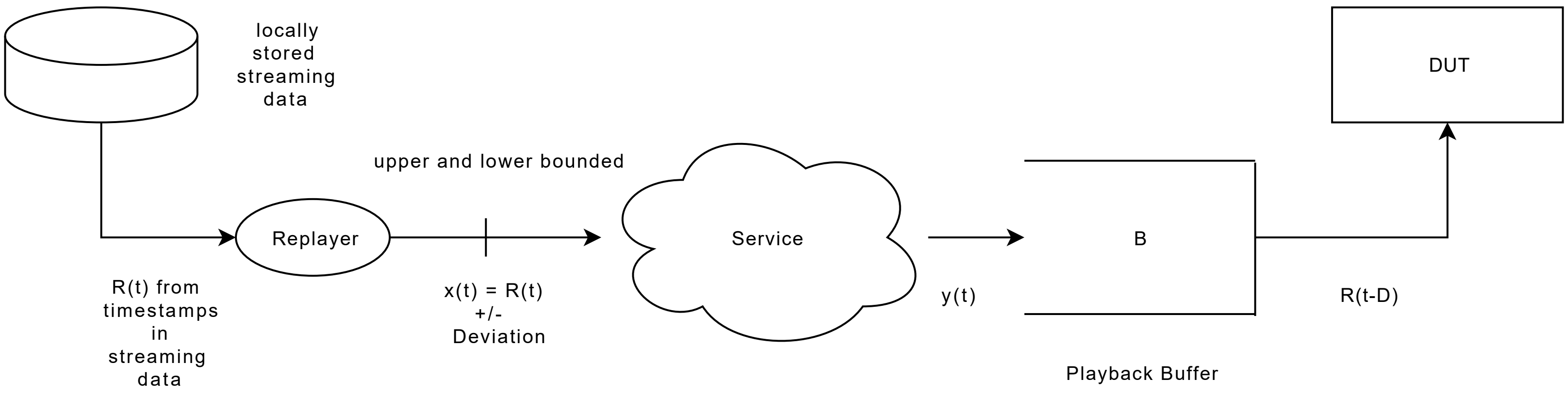





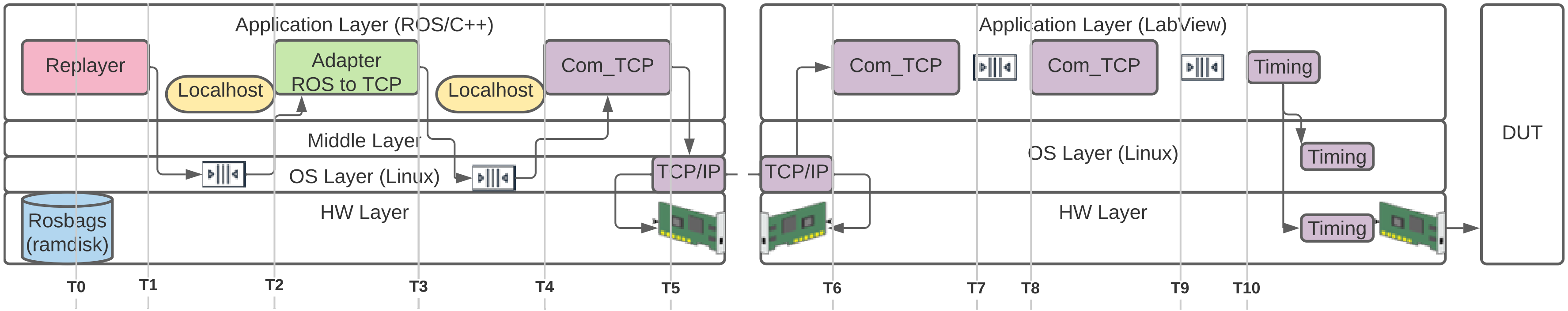
© ZF Friedrichshafen AG

Conceptual model of HIL streaming system



© ZF Friedrichshafen AG

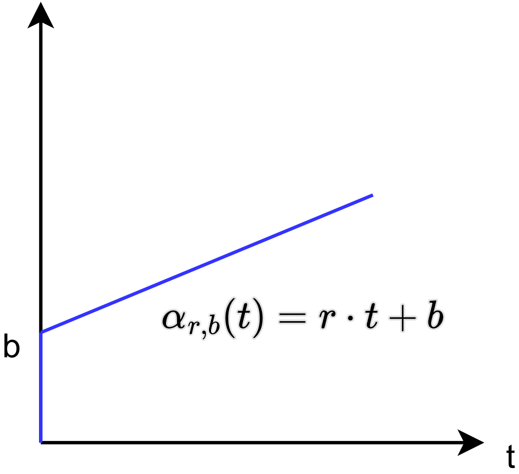
SW instrumentation for timestamp logging at HIL

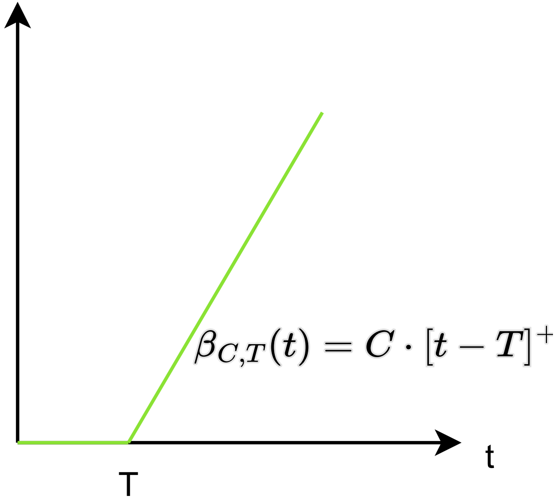


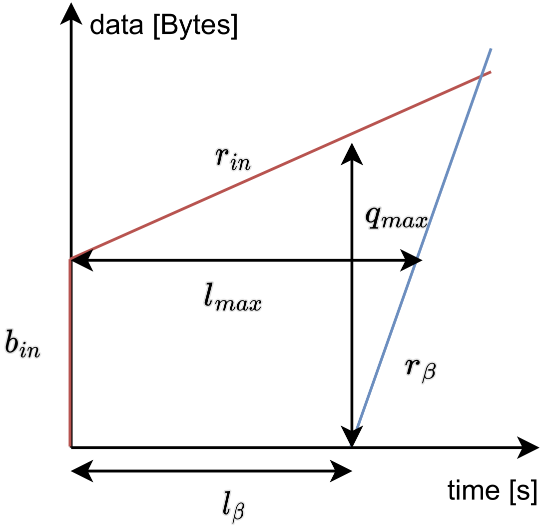
© ZF Friedrichshafen AG

Network Calculus basics

- Mathematical system theoretical tool
- provides bounds for flows: maximum delay, maximum backlog.
- uses min-plus algebra: addition --> minimum and multiplication --> addition.
- Arrival curves
- Service curves

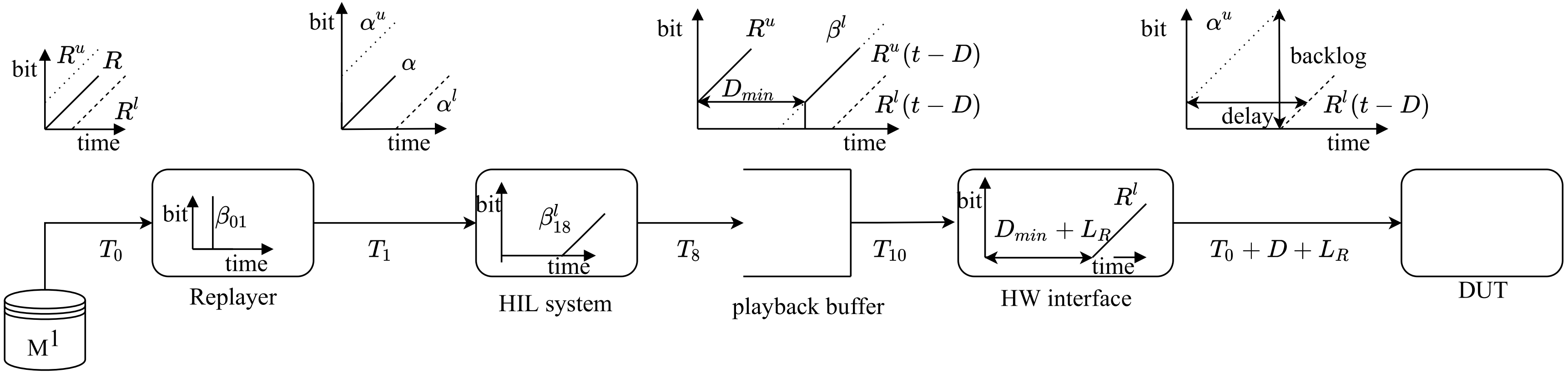






Internal

Application of NC for streaming sytems to HIL simulator



[4] C. Funda, T. Konheiser, T. Herpel, R. German, and K.-S. Hielscher, "An industrial case study for performance evaluation of hardware-in-the-loop simulators with a combination of network calculus and discrete-event simulation," in 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), 2022, pp. 1–7.

© ZF Friedrichshafen AG

Problem description and research questions

Problem description

- Timestamp logging is resource intensive
- Low Scalability
- Not usable for monitoring during operation
- Timestamps used to generate arrival- and service-curves
- More efficient and effective to generate curves directly?

Benefit for the HIL-system:

- Performance monitoring of SW/HW services during operation
- Using for:
  - Debugging & error root-cause analysis
  - HIL design & optimization (pre-buffer time, buffer dimensioning)
  - Bottleneck identification

Research Questions

- How can we monitor the performance of our HIL streaming system during operation with the following requirements:
  - Producing less logging data
  - Low usage of CPU & RAM
- How accurate are the bounds by online generated arrival and service-curves compared to offline methods based on timestamps?
- How performant are the arrival- and service-curve estimation methods implemented as online algorithms compared to state-of-the-art timestamp logging?

© ZF Friedrichshafen AG

Algorithm requirements and performance rating

TABLE I
COMPARISON OF METHODS FOR ESTIMATING SERVICE CURVES

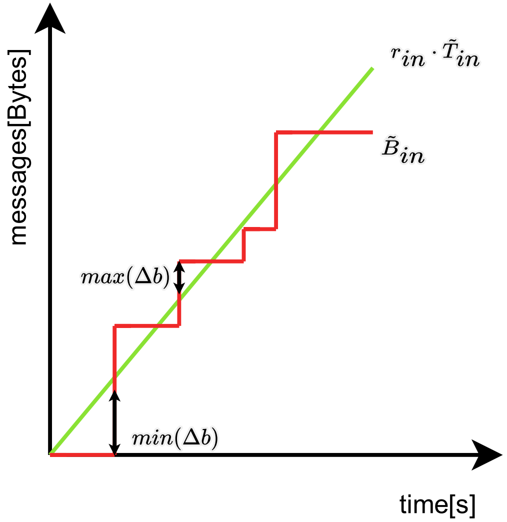
Method	Computational effort	Memory usage	Scalability	accuracy of bounds
Alcuri et al. [2]	High	High	Low	High
Funda et al.- WCET [1] inspired by Helm et al. lowest service approach [3]	Low	Low	High	Low
Funda et al.- MCET [1] inspired by Helm et al. mean service approach [3]	Medium	Medium	Medium	Medium
Funda et al.- BCET [1] inspired by Liebeherr et al. Sliding-Window approach [6]	High	High	Low	Medium
Online Algorithm Requirements	Low	Low	High	Medium-High
timestamp logging	Medium	High	Low	High

TABLE II
COMPARISON OF METHODS FOR ESTIMATING ARRIVAL CURVES

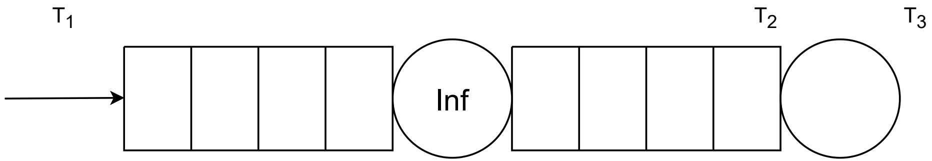
Method	Computational effort	Memory usage	Scalability	Precission of bounds
Bouilliard [5]	Medium	High	Medium	High
Funda et al. [4]	Low	High	Medium	Medium
Online Algorithm Requirements	Low	Low	High	Medium-High
timestamp logging	Medium	High	Low	High

[1] C. Funda, P. Marin Garcia, R. German, and K.-S. Hielscher, “Arrival and service curve measurement-based estimation methods to analyze and design soft real-time streaming systems with network calculus,” 2023, ICECCME’23, unpublished.

Online algorithm for arrival curve estimation



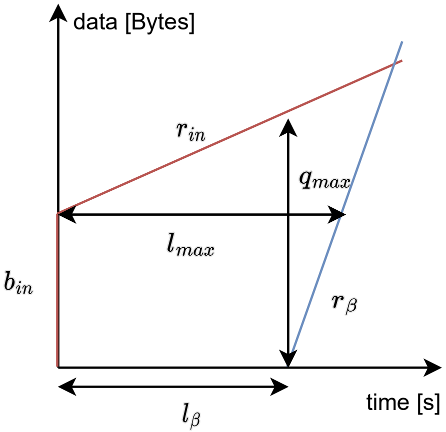
$$b_{in} = max\{\Delta b, 0\} - min\{\Delta b, 0\}$$



- **Mean input-rate (r_{in}):**
  - Mean input-rate r_{in} computed from T0 packets
- **Burst parameter (b_{in}):**
  - Compare r_{in} * T1 to the counted message curve b(T1)
  - Measure min(Δb) and update b_{min} if smaller than prev. value
  - Reset b_{max} to 0 if b_{min} is smaller than prev. value
  - Measure max(Δb) and update b_{max} if higher than prev. value
  - Calculate b_{in} as $|b_{min}| + |b_{max}|$ and update max. value of b_{in} if higher than prev. value

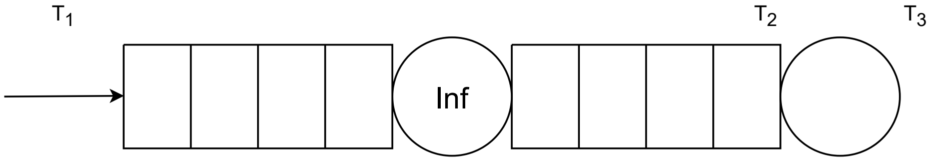
© ZF Friedrichshafen AG

Online algorithm for arrival & service curve estimation



$$l_\beta = \frac{max\{q_{max} - b_{in}, 0\}}{r_{in}}$$

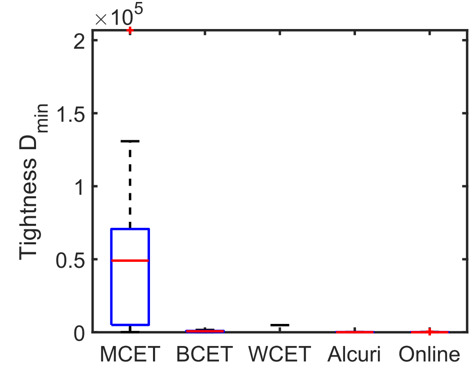
$$r_\beta = \frac{b_{in}}{max\{l_{max} - l_\beta, 0\}}$$



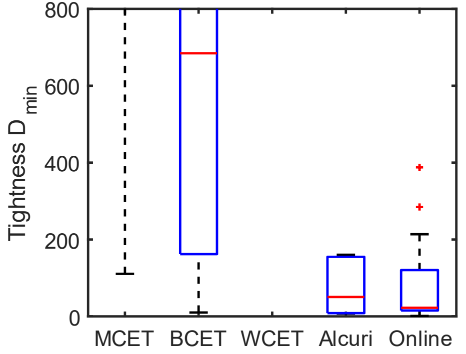
- **Maximum delay (l_{max}):**
  - Time between T1 and T3
  - Iteratively check and save the maximum value
- **Maximum queue length (q_{max}):**
  - Difference between incoming msgs at T1 and outgoing msgs at T3
  - Iteratively check and save the maximum value

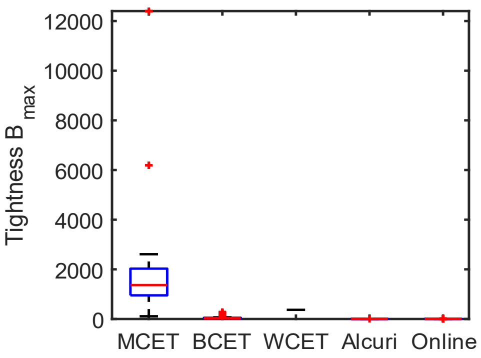
© ZF Friedrichshafen AG

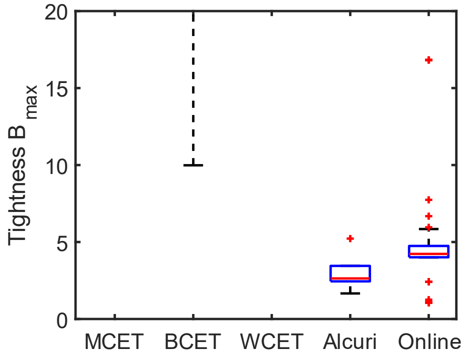
Tightness of bounds in 53 datasets from the HIL



$$Tigthness = \frac{bound}{\max\_value}$$







Conclusion

Accuracy

- Online algorithm and Alcuris algorithm similar tight bounds
- Alcuri algorithm = definition of a strict service curve → Reference algorithm for online-algorithm

Performance

- Memory consumption of online algorithm is assumed very less compared to timestamp logging
  - 4 variables instead of 2 timestamps per message
- RAM utilization depends on:
  - Waiting times
  - Start of streaming to HIL (pre-buffer phase) and start of streaming to DUT (start of test)
  - Processing performance of the Online-Algorithm
- Assume low memory and CPU performance compared to timestamp-logging
- Verification still open (Algorithm implementation in LabVIEW is work-in-progress)

© ZF Friedrichshafen AG

Thank you for your attention!

Discussion!

Questions?

References

[1] C. Funda, P. Marin Garcia, R. German, and K.-S. Hielscher, “Arrival and service curve measurement-based estimation methods to analyze and design soft real-time streaming systems with network calculus,” 2023, ICECCME’23, unpublished.

[2] L. Alcuri, G. Barbera, and G. D’Acquisto, “Service curve estimation by measurement: An input output analysis of a softswitch model,” in Quality of Service in Multiservice IP Networks, M. Ajmone Marsan, G. Bianchi, M. Listanti, and M. Meo, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 49–60.

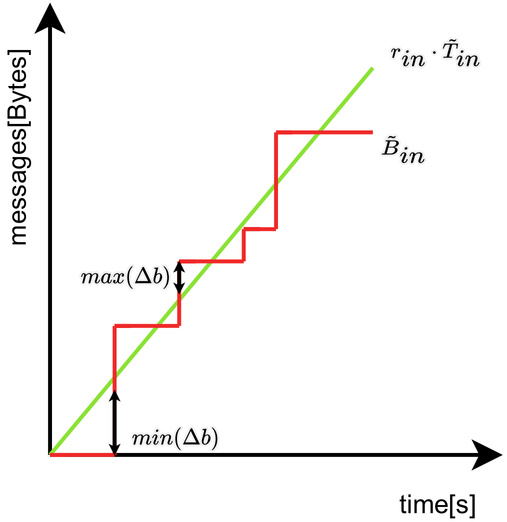
[3] M. Helm, H. Stubbe, D. Scholz, B. Jaeger, S. Gallenmüller, N. Deric, E. Goshi, H. Harkous, Z. Zhou, W. Kellerer, and G. Carle, “Application of network calculus models on programmable device behavior,” in 2021 33rd International Teletraffic Congress (ITC-33), Avignon, France, Aug. 2021, pp. 1–9. [Online]. Available: https://gitlab2.informatik.uni-wuerzburg.de/itc-conference/itcconference-public/-/raw/master/itc33/hel21ITC33.pdf?inline=true

[4] C. Funda, T. Konheiser, T. Herpel, R. German, and K.-S. Hielscher, “An industrial case study for performance evaluation of hardware-in-the-loop simulators with a combination of network calculus and discrete-event simulation,” in 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), 2022, pp. 1–7.

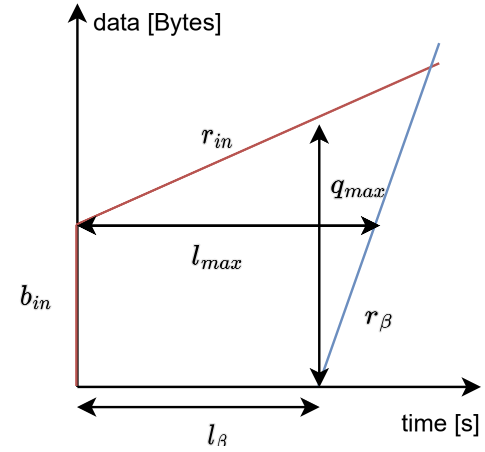
[5] A. Bouillard, “Algorithms and efficiency of Network calculus,” Habilitation `a diriger des recherches, Ecole Normale Sup´erieure (Paris), Apr. 2014. [Online]. Available: https://hal.inria.fr/tel-01107384

© ZF Friedrichshafen AG

Online algorithm for arrival & service curve estimation



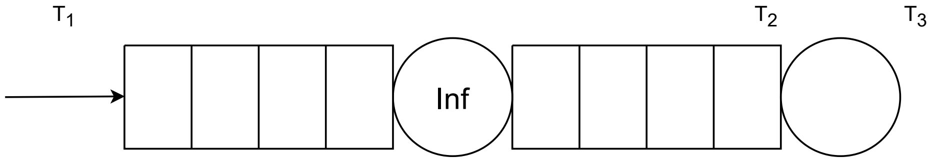
$$b_{in} = max\{\Delta b, 0\} - min\{\Delta b, 0\}$$



$$l_\beta = \frac{max\{q_{max} - b_{in}, 0\}}{r_{in}}$$

$$r_\beta = \frac{b_{in}}{max\{l_{max} - l_\beta, 0\}}$$

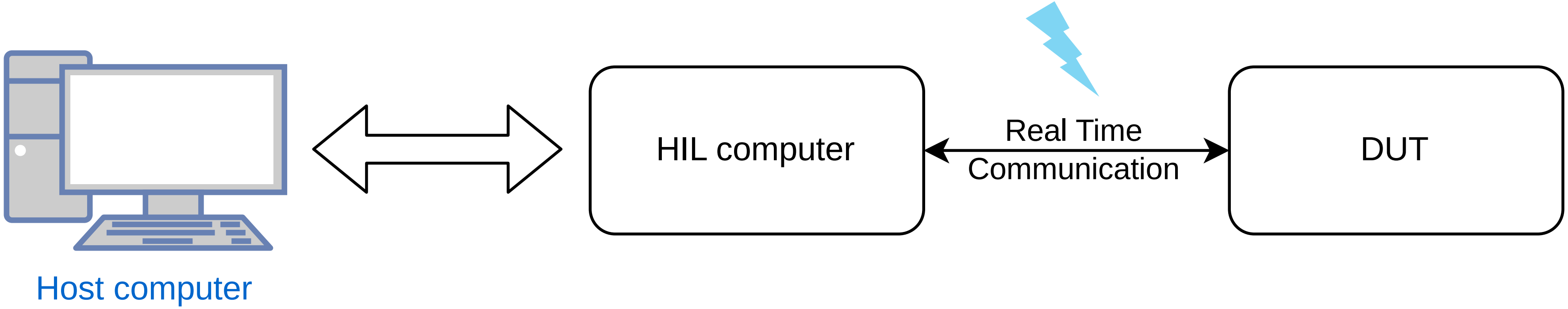
Iterative online algorithm



During operation, following three parameters are iteratively computed:

- Maximum delay (l_max):
  - Time between T1 and T3
  - Iteratively check and save the maximum value
- Maximum queue length (q_max):
  - Difference between incoming msgs at T1 and outgoing msgs at T3
  - Iteratively check and save the maximum value
- Burst parameter (b_in):
  - Mean input-rate computed from T0 packets
  - Compare to the counted message curve (db) at T1
  - Calculate negative deviation between db and mean-rate estimation and update b_min if smaller than prev. value
  - Reset b_max to 0 if b_min is smaller than b_min(old)
  - Calculate positive deviation of db from mean-rate estimation and update b_max if higher than prev. value
  - Calculate b_in as abs of b_min + b_max and update max. value if higher than prev. value

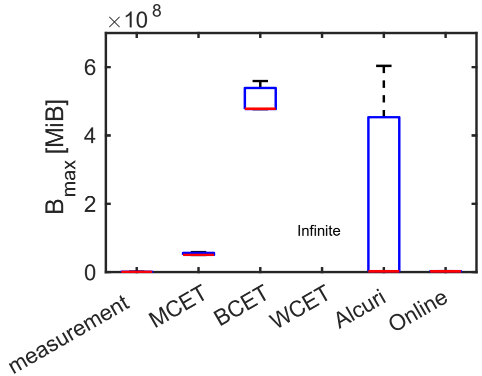
Conceptual model of HIL test system and QoS requirements



- Quality of Service (QoS) requirements to the HIL test system:
  - **High timing accuracy** and **precision** at the HIL interface to the Device under Test (DUT)
  - **High data-integrity** (No data-loss or data-corruption)

- What if error on HIL + error on DUT → true test results?:
  - System design of HIL to meet QoS requirements
  - Performance evaluation of HIL to verify QoS requirements & detailed design of HIL
  - Monitoring of QoS during HIL operation

Performance evaluation – accuracy comparison of bounds



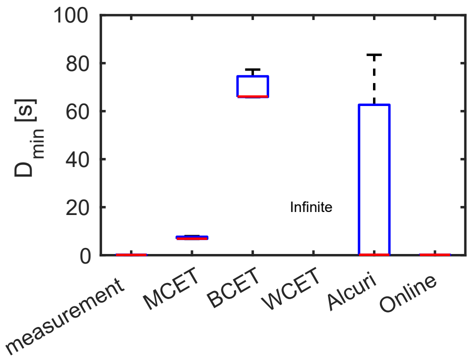


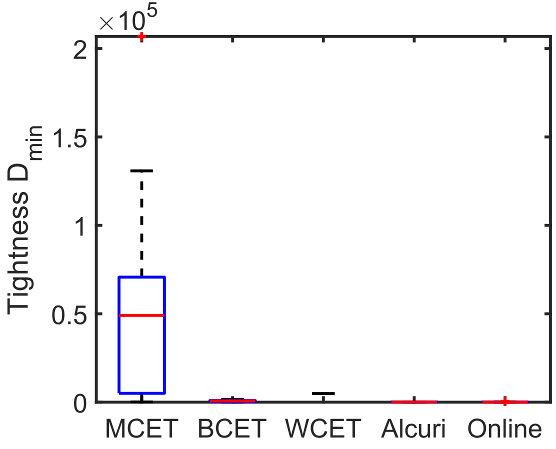
TABLE III
TIGHTNESS OF NC METHODS (USING MAX. VALUES)

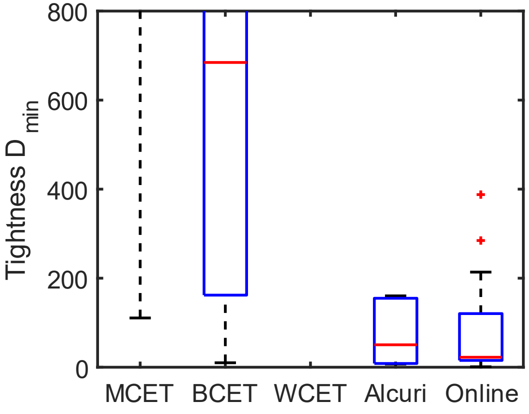
Method	D_{min}[s]	B_{max}[MiB]	D_{min} Tightness	B_{max} Tightness
Measurement	0.15	390256	-	-
Funda- MCET	7.95	$5.84x10^7$	51.51	41.99
Funda- BCET	77.33	$5.58x10^8$	501.14	402.39
Funda- WCET	Inf	Inf	Inf	Inf
Alcuri et al.	83.47	$6.04x10^8$	540.99	434.32
Funda - Online Algorithm	0.19	2382992	1.26	1.71

- 1 Data-set from timestamp logging of the HIL system
- Online-Algorithm has tightest bounds
- Tightness = bound/maximum measured value
- Tightness <1 → underestimation (very bad)
- Tightness >1 → overestimation (not so much bad)
- Tightness should be sligthly over 1

© ZF Friedrichshafen AG

Tigthness of 53 datasets

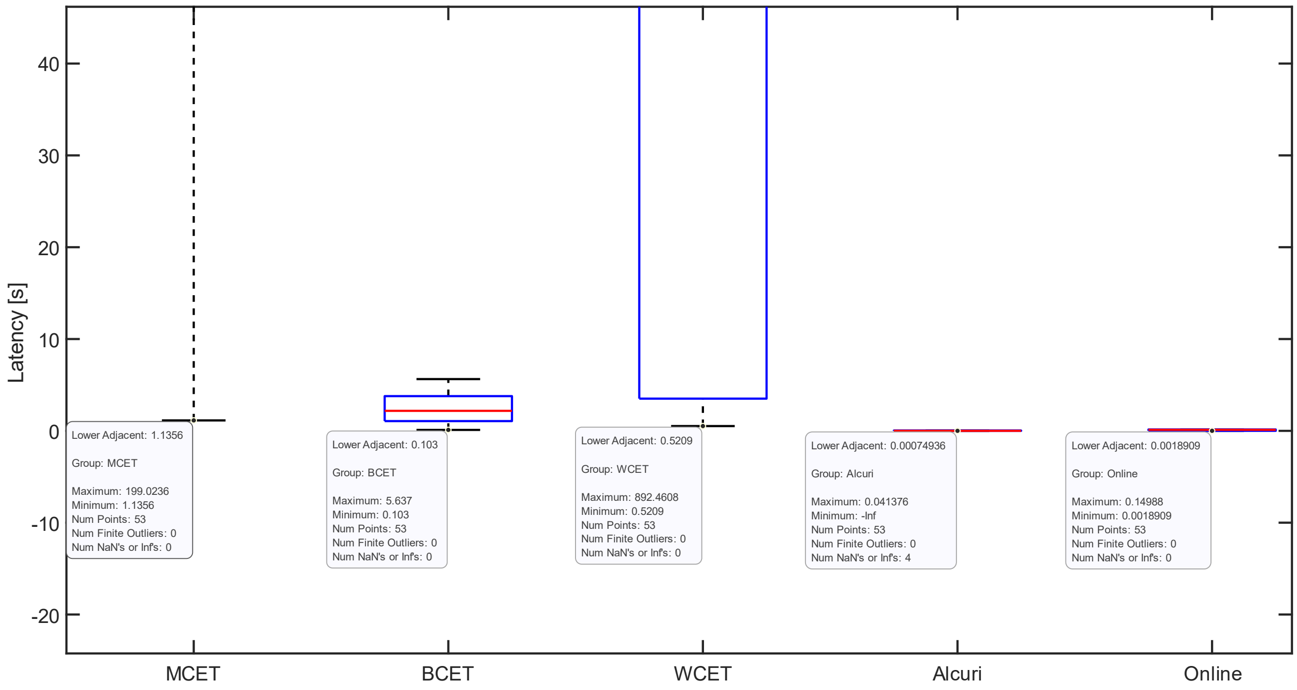




method	min(Tightness(D_min))	mean(Tightness(D_min))	max(Tightness(D_min))	std(Tightness(D_min))	min(Tightness(B_max))	mean(Tightness(B_max))	max(Tightness(B_max))	std(Tightness(B_max))
Measurement	1	1	1	0	1	1	1	0
Funda - MCET	110,6651154	41067,20358	206829,5315	42998,70307	109,9423077	2497,958061	12400,98547	3180,873735
Funda - BCET	10,03199535	621,8443061	1599,803205	512,304615	9,983526563	66,81064626	284,7407734	78,45510356
Funda - WCET	4897,745807	17639,28655	Inf	21775,3684	369,2403011	688,177208	Inf	276,305202
Alcuri et al.	7,132805937	35,28414322	Inf	36,58877393	1,660238281	2,561089503	Inf	0,541337821
Funda - Online Algorithm	1,038685282	66,54588196	387,772681	78,41703609	1,050996049	5,055594253	16,82482578	3,613084655

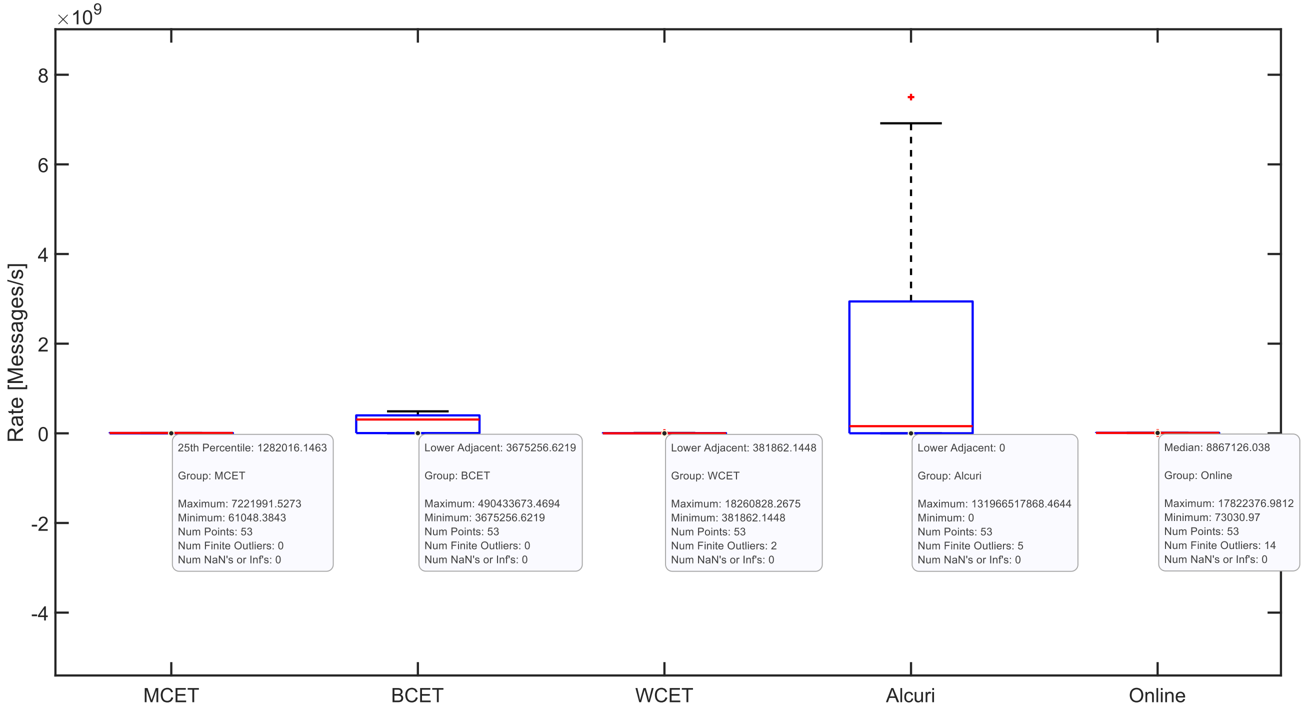
© ZF Friedrichshafen AG

Server Latency



© ZF Friedrichshafen AG

Server Rate


×10^9
8
6
4
2
0
-2
-4
Rate [Messages/s]
25th Percentile: 1282016.1463
Group: MCET
Maximum: 7221991.5273
Minimum: 61048.3843
Num Points: 53
Num Finite Outliers: 0
Num NaN's or Inf's: 0
Lower Adjacent: 3675256.6219
Group: BCET
Maximum: 490433673.4694
Minimum: 3675256.6219
Num Points: 53
Num Finite Outliers: 0
Num NaN's or Inf's: 0
Lower Adjacent: 381862.1448
Group: WCET
Maximum: 18260828.2675
Minimum: 381862.1448
Num Points: 53
Num Finite Outliers: 2
Num NaN's or Inf's: 0
Lower Adjacent: 0
Group: Alcuri
Maximum: 131966517868.4644
Minimum: 0
Num Points: 53
Num Finite Outliers: 5
Num NaN's or Inf's: 0
Median: 8867126.038
Group: Online
Maximum: 17822376.9812
Minimum: 73030.97
Num Points: 53
Num Finite Outliers: 14
Num NaN's or Inf's: 0
MCET
BCET
WCET
Alcuri
Online