Evaluation of APODIS Plasma Disruption Predictor within the ITER Real-Time Framework

G. Arranz (german.arranz@alumnos.upm.es), *C. González, M. Ruiz, J. Vega, S. Dormido-Canto, E. Barrera, J. Nieto, A. Carpeño, and JET EFDA Contributors*

25th IEEE Real Time Conference
La Biodola - Isola d'Elba, Italy

May 25th–29th, 2026

UNIVERSIDAD POLITÉCNICA DE MADRID

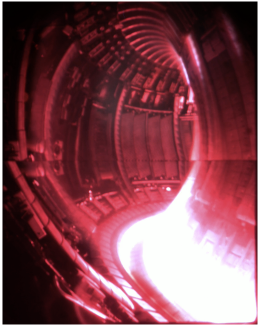
Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas

This work was supported by Spanish Ministerio de Ciencia e Innovación/Agencia Estatal de Investigación MCIN/AEI/10.13039/501100011031/FEDER project PID2022-137680OB-C31, MCIN/AEI/10.13039/501100011032/FEDER project PID2022-137680OB-C32, MCIN/AEI/10.13039/501100011033/FEDER project PID2022-137680OB-C33.

Outline

1. Introduction
   - Previous work for JET
   - ITER RTF and CODAC
   - Work Objectives
2. Implementation
3. Results
4. Conclusions

Introduction: Previous work developed for JET experiment



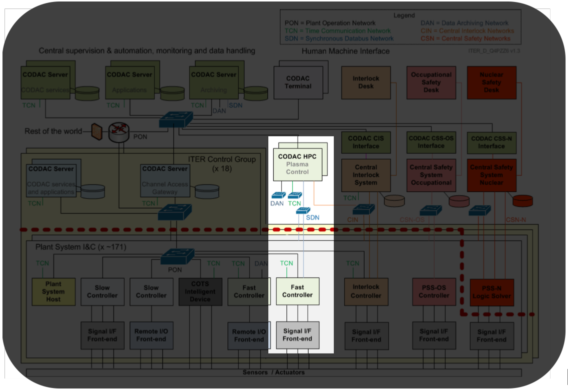
[1]

- Previous experience from CIEMAT/UNED in the JET fusion experiment device (Cullham), developing disruption predictors using Machine Learning techniques.
- Successful development and deployment of APODIS predictor implemented with Support Vector Machine techniques using MARTe Real-Time environment.

References

J. Vega, "Disruption prediction with artificial intelligence techniques in tokamak plasmas" 10.1038/s41567-022-01602-2
J. Vega, "Results of the JET real-time disruption predictor in the ITER-like wall campaigns" 10.1016/j.fusengdes.2013.03.003
J.M. López, "Implementation of the disruption predictor APODIS in JET's real-time network using the MARTe framework" 10.1109/TNS.2014.2309254

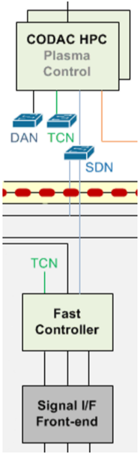
Introduction: ITER CODAC and Synchronous Databus Network



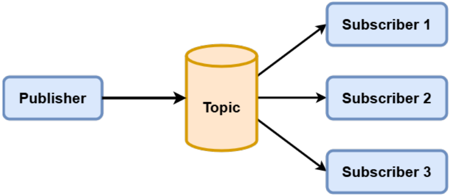
[2]

- Control Data and Communication (CODAC) is ITER's central control system for plant-wide I&C integration.
- Within the CODAC system, the Plasma Control System exists to ensure plasma stability.
- The loss of confinement due to disruptions takes place in times O(1ms); therefore, it is necessary to develop real-time systems to predict and mitigate them.

Introduction: ITER CODAC and Synchronous Databus Network



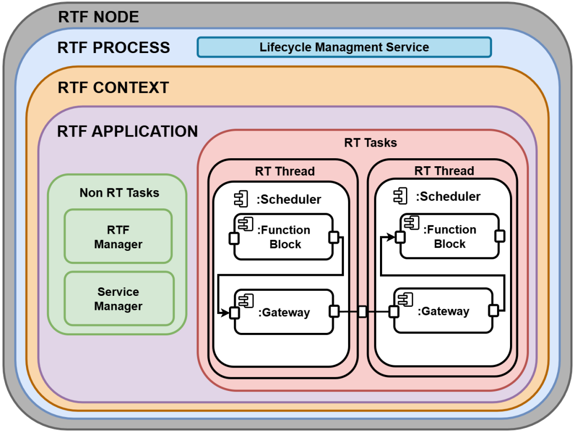
- The Synchronous Databus Network is used for real-time data exchange. It is based on UDP/IPv4 multicast over 10 Gbps Ethernet links.



- The different machines comprising the CODAC/PCS architecture are based on the CODAC Core System (CCS) distribution, which is based on Red Hat Enterprise Linux (RHEL). It also supports the use of the Linux `PREEMPT_RT` kernel patch.

Introduction: The ITER Real-Time Framework

- Platform-independent, modular, **distributed** framework designed for real-time applications.
- It differentiates between real-time and non-real-time tasks.
- Services provide site-specific facilities (profiling, networking, etc...).
- Function blocks are the principal building blocks for RT applications. Each function block accepts inputs and produces outputs whenever it is processed.



Introduction: Objectives

- Implement the **JET real-time APODIS** disruption predictor using the ITER Real Time Framework.
- **Evaluate its performance** in terms of **latency** and **determinism** within an emulated **CODAC environment** using the **ITER CODAC Core System** software distribution and the **ITER SDN network**.
- Provide a detailed description of both the configuration of the RTF application and the **low-latency optimizations** conducted to improve the performance of the developed application.
- Prove the suitability of the **RTF** for real-time applications within the CODAC architecture.

Outline

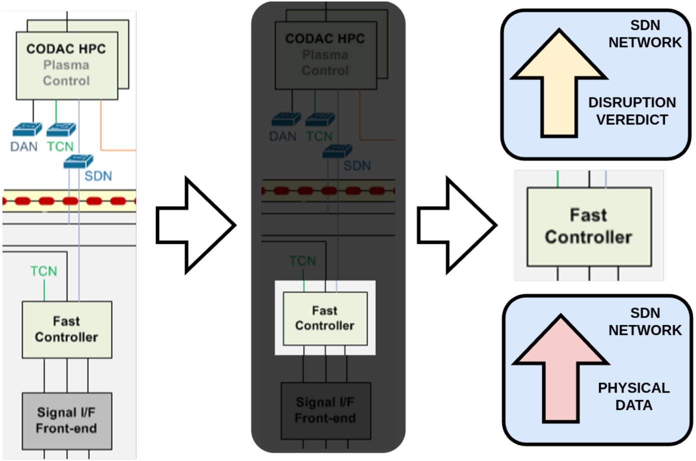
1 Introduction

2 Implementation
- RTF APODIS Application Architecture
- Performance tuning

3 Results

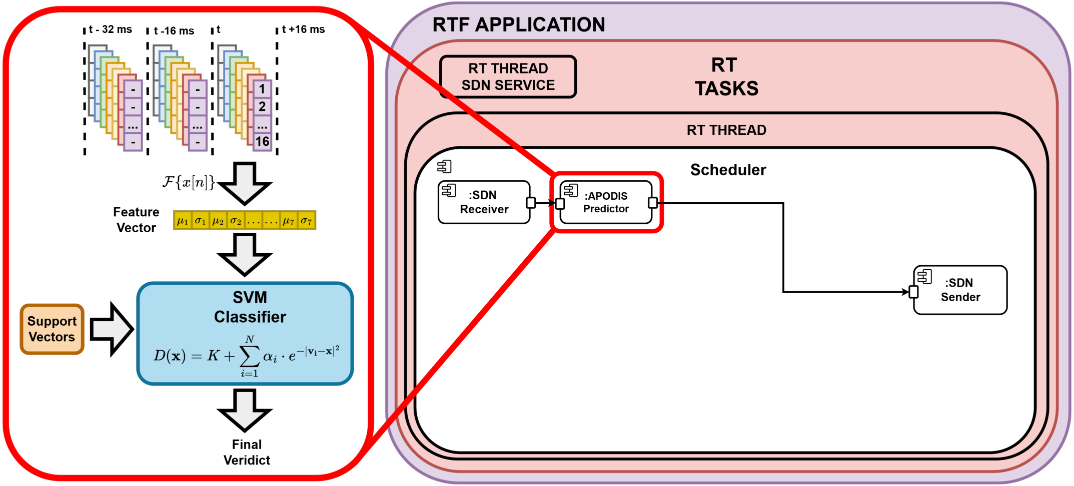
4 Conclusions

Implementation: RTF APODIS Application Architecture

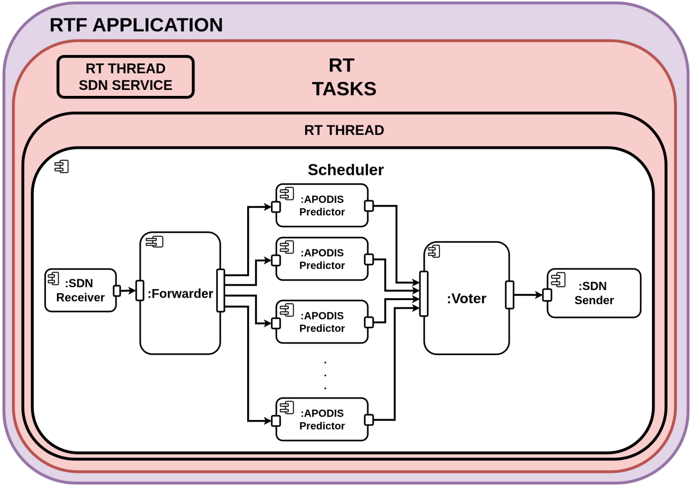


- Designed for deployment on ITER fast controllers.
- The application receives physical input data and sends the discharge classification over the SDN network.

Implementation: RTF APODIS Application Architecture

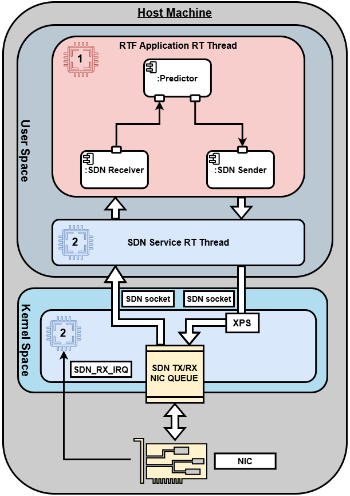


Implementation: RTF APODIS Application Architecture



Implementation: Performance tuning

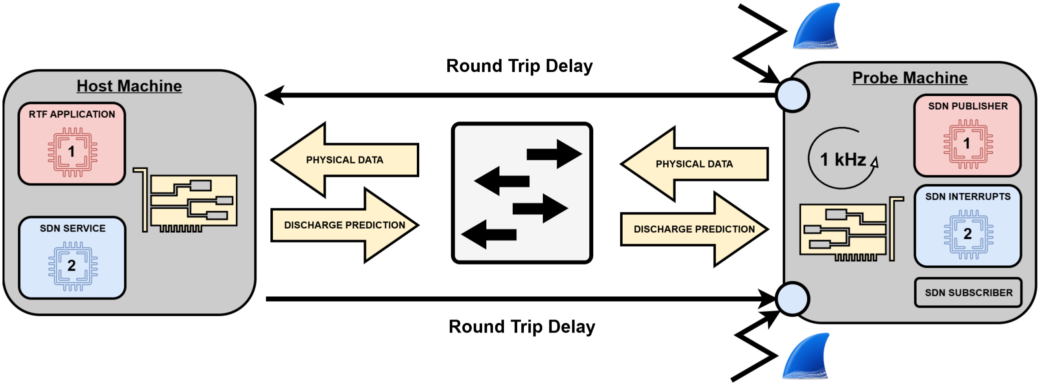
- **RTF Threads:** `SCHED_FIFO` (priority 80)
- **NIC Optimization:**
  - Unified TX/RX queue with a single SDN IRQ.
  - Pin SDN RX IRQ and service thread to the same core.
  - XPS to map SDN CPU to the queue.
- **Kernel Tuning**:
  - Ban isolated cores and SDN IRQs from `irqbalance`.
  - Enable `nohz_full` on isolated cores.
  - Move `kworkers` to non-isolated cores.
- **CPUs States:**
  - Disable C-states and frequency scaling.



Outline

1. Introduction
2. Implementation
3. Results
   - Experimental Setup
   - FBs' Individual Execution Times
   - Network Latency
4. Conclusions

Results: Experimental Setup



- Test Data (JET C23): 200 iterations of 8 discharges (72,896 points) resulting in 14,579,200 points and 911,200 predictions, over a 4-hour experimental time frame.

Results: FBs' Individual Execution Times

Function Block	Mean (μs)	Std. Dev. (μs)	Range (μs)	
			Min	Max
Receiver	4.15	0.15	3.56	40.07
Buffering	0.12	0.02	0.09	0.98
DFT	3.74	0.12	3.42	13.92
Overall	0.34	0.88	0.09	13.92
Sender	7.50	0.32	5.96	34.10
Voter	0.14	0.01	0.10	3.89
Forwarder	3.19	0.03	3.01	6.25

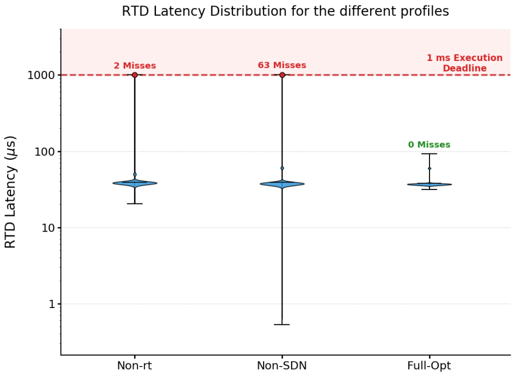
- A profiling service is provided by the RTF to characterize the execution times of the different FBs.
- All FBs implementations achieve real-time performance: microsecond-scale execution times with highly deterministic behavior.

Results: Network Latency

- To demonstrate its suitability in a real-time environment, it is mandatory to measure the global execution time of the whole application.
- Three different profiles to test the application and its different optimizations:
  1. **Non-RT:** No optimizations have been applied to the host machine.
  2. **Non-SDN:** This profile implements all the optimizations only related to the RTF application while disregarding the ones in charge of optimizing the SDN communication.
  3. **Full-opt:** All the optimization techniques have been applied.

Results: Network Latency

- **Execution Performance:**
  - Average execution time $\mu \approx 40\ \mu s$
  - Std dev remains stable $\sigma \approx 6\ \mu s$
- **Optimizations Impact:**
  - Full-Opt profile successfully eliminates all execution misses $> 1ms$.
  - Successfully caps maximum execution times
  - Maximum execution time $= 91.82\ \mu s$



Outline

1 Introduction

2 Implementation

3 Results

4 Conclusions

Conclusions

- A real-time disruption predictor based on the **APODIS algorithm** was **implemented and validated within the ITER RTF**.
- Provide a detailed description of the **optimizations** carried out to improve **latency and determinism**.
- Achieved average latency of $\sim$ **40 µs**, standard deviation of $\sim$ **6 µs**, and a maximum execution time of **91.82 µs** over a **4-hour** experimental time frame, well **under the 1 ms** target.
- Demonstrated the RTF as a **feasible platform** for **real-time plasma control** applications within **ITER's CODAC environment**.

References

[1] EUROfusion, *Fusion pulse*, https://euro-fusion.org/eurofusion-news/dte3record/, [Accessed: May 20, 2026], 2024.

[2] ITER Organization, *Codac architecture*, https://www.iter.org/machine/supporting-systems/codac/architecture, [Accessed: April 20, 2026].

Evaluation of APODIS Plasma Disruption Predictor within the ITER Real-Time Framework

<u>G. Arranz</u> (german.arranz@alumnos.upm.es), *C. González, M. Ruiz, J. Vega, S. Dormido-Canto, E. Barrera, J. Nieto, A. Carpeño, and JET EFDA Contributors*

25th IEEE Real Time Conference
La Biodola - Isola d'Elba, Italy

May 25th–29th, 2026

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.