instituto de engenharia electrónica e telemática de aveiro

universidade de aveiro

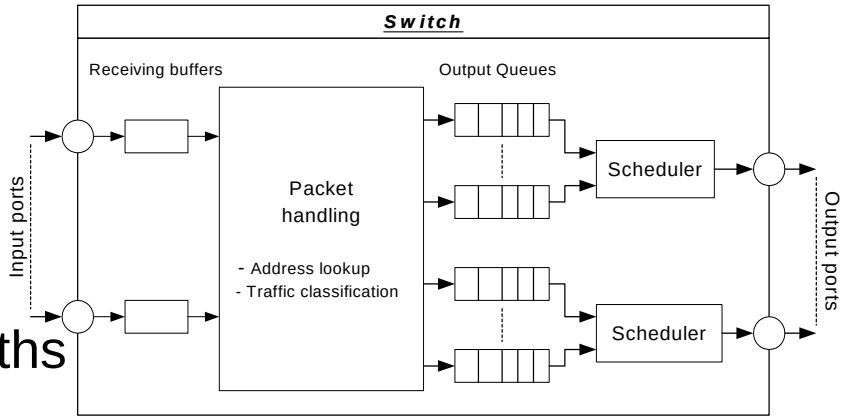
# Enhanced Ethernet Switching Technology for Adaptive Hard Real-Time Applications

Rui Santos
(rsantos@ua.pt)

7 / 05 / 2009

# Problem

- **Switched Ethernet became common in real-time communications**
  - Some interesting properties
    - Large bandwidth
    - Cheap network controllers
    - Micro-segmentation
      - Collisions are eliminated
    - Multiple parallel forwarding paths
    - High availability




- **But there are still limitations**
  - FIFO queues
  - Limited number of priorities
  - Memory overflows

# Solutions

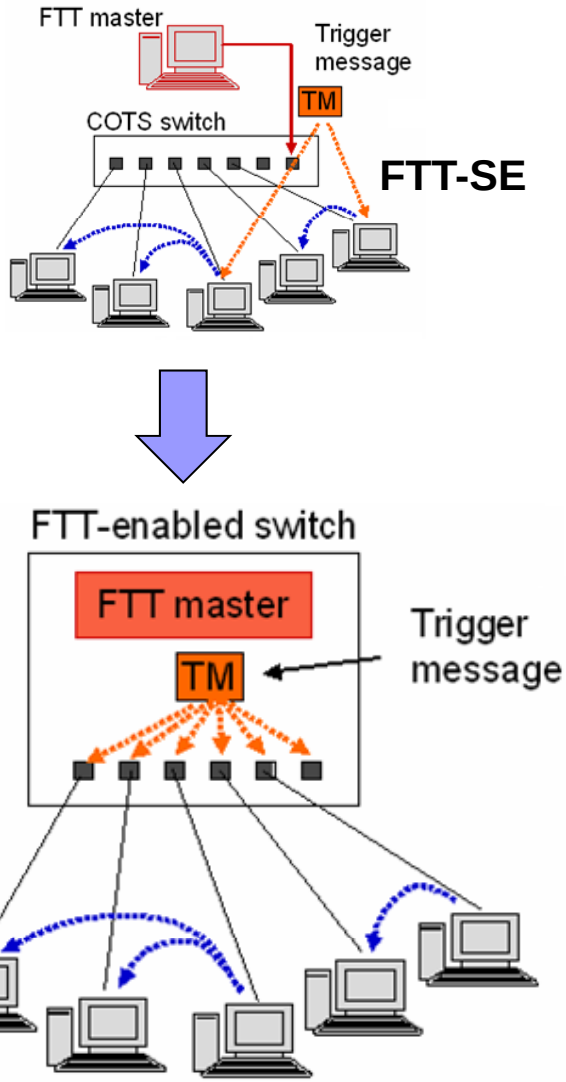
- **Commercial Off-The-Shelf Ethernet switches**
  - Limiting the generated traffic by the application design
  - Traffic shaping
  - Master-Slave protocols (FTT-SE, …)
- **Customized Ethernet Switches**
  - TTEthernet
  - Profinet-IRT
  - FTT-Enabled Switch (HaRTES) - our solution

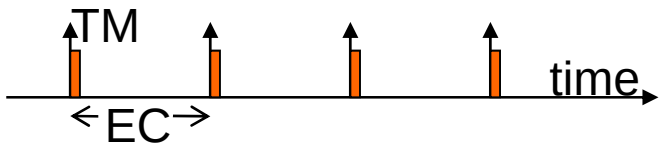
# TTEthernet & Profinet-IRT Limitations

- Require a **static pre-defined configuration** for the real-time traffic
- **On-line admission control** is **not** generally **available**
- **Miss on-line adaptation** to the communication requirements and quality-of-service policies

# Protocol Mechanism

- **Based on Flexible Time-Triggered Paradigm**
- **Master-slave transmission control technique**
- **Communication occurs in fixed slots (Elementary Cycles – ECs)**
  - ECs are organized in RT and NRT windows
  - Supports synchronous, asynchronous and non real-time traffic, with strict temporal isolation
- **The ECs start with a Trigger Message (TM) sent by the Master (switch)**
  - TM contains the schedule for each EC



# FTT-E. Switch (HaRTES) Properties

- **Traffic scheduling and management**

Old – Global traffic coordination in a common timeline
  - Master synchronizes all nodes

Old – Supports online admission control and dynamic QoS management

Old – Allows arbitrary traffic scheduling policies

New – Reduction in the switching latency jitter

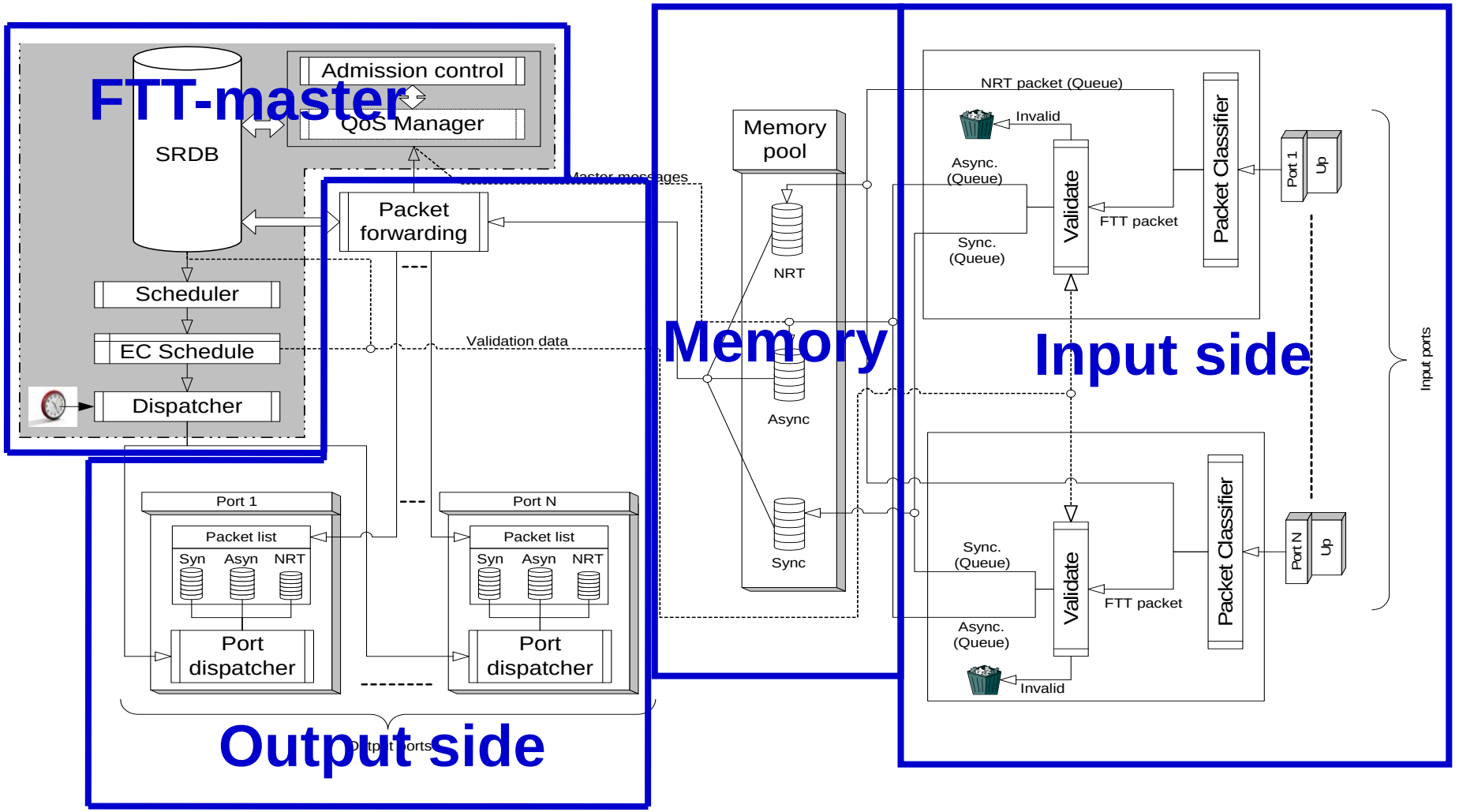
- **Traffic classification, confinement and policing**

New – Seamless integration of standard non-FTT-compliant nodes without jeopardizing the real-time services

New – Asynchronous traffic is autonomously triggered by the nodes

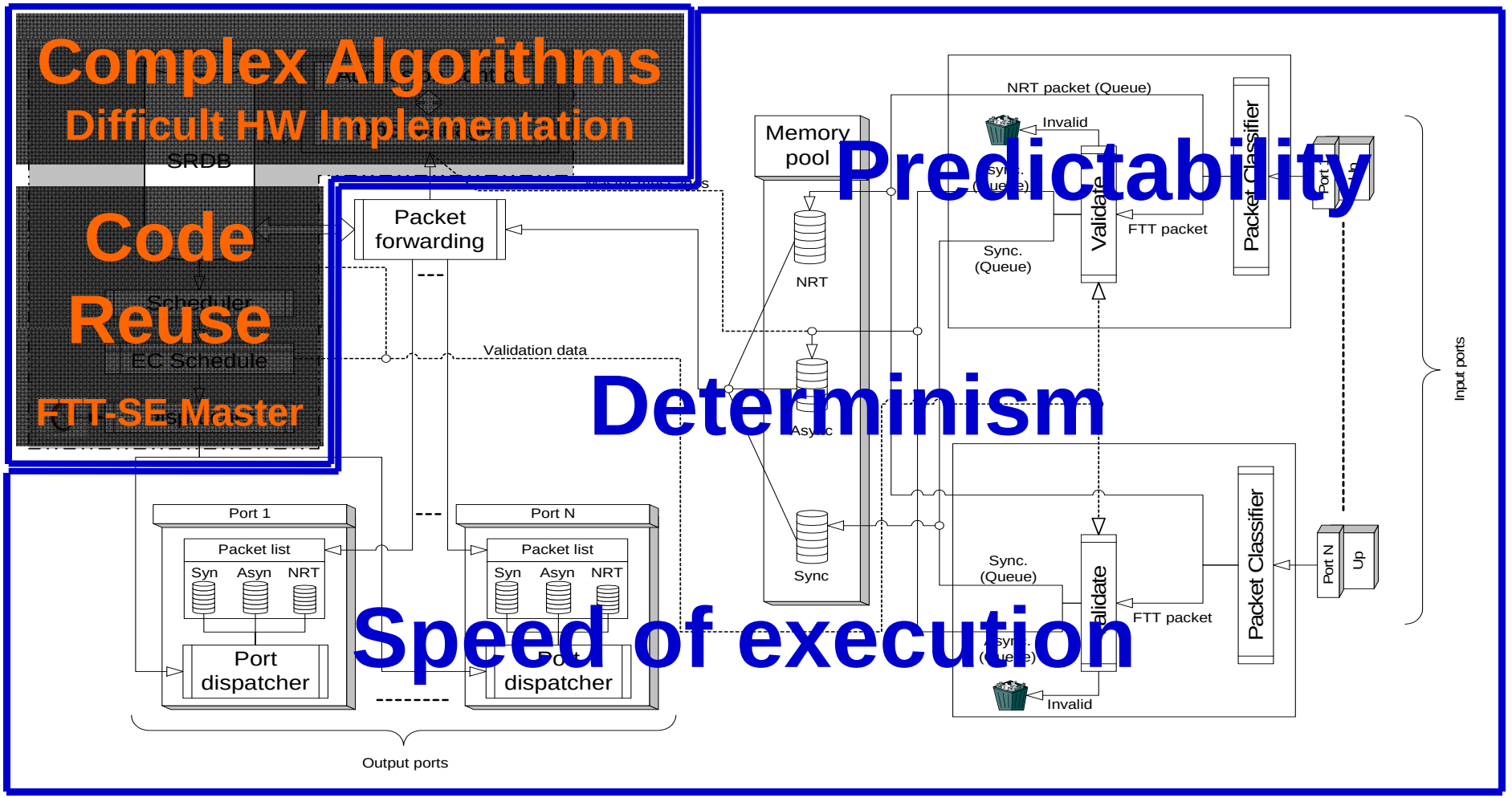
New – Non-conforming transmissions can be readily identified and blocked at the switch input ports, thus not interfering with the rest of the system

# FTT-E. Switch (HaRTES) Architecture



# FTT-E. Switch (HaRTES)

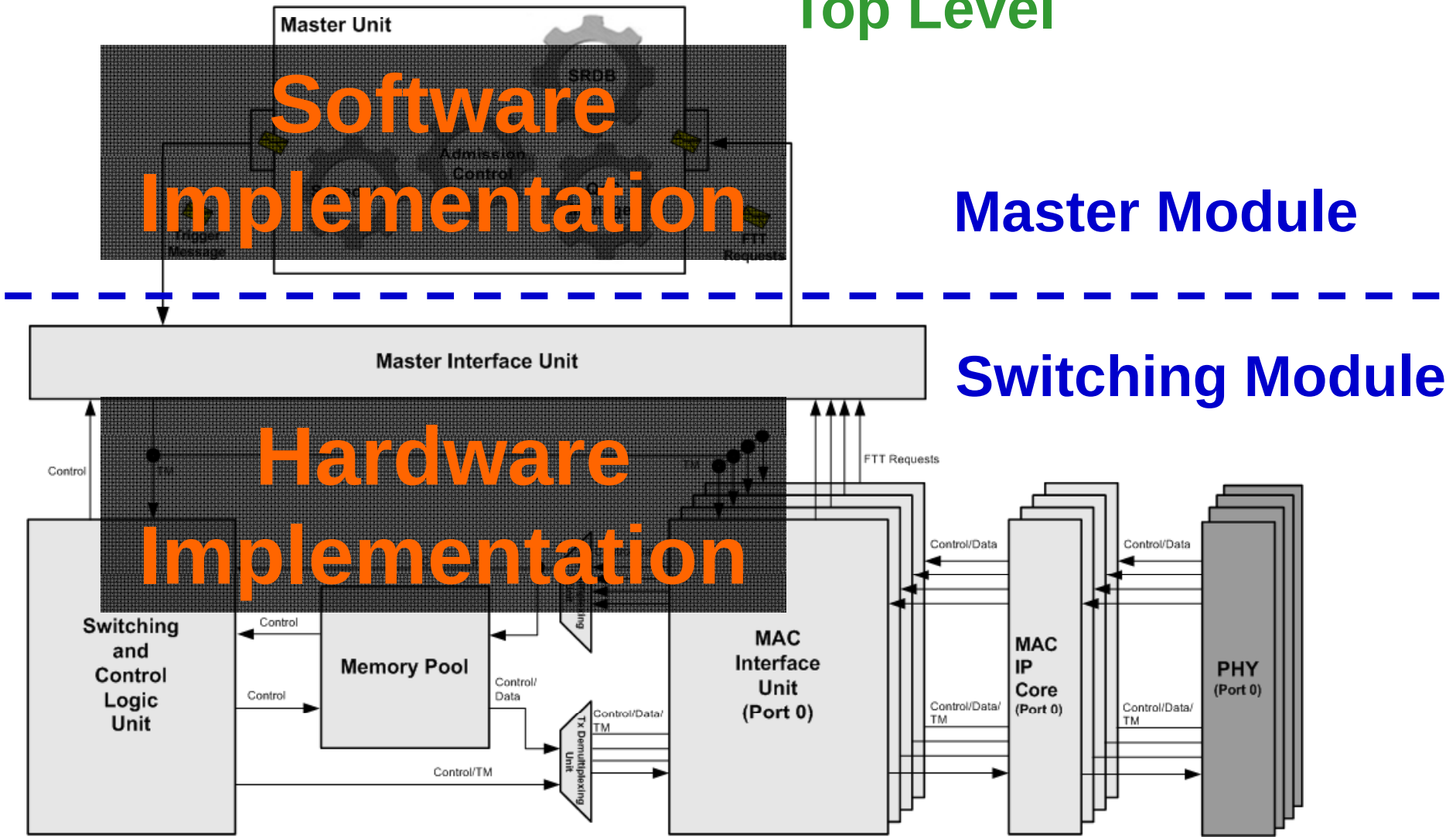
## Architecture



# FTT-E. Switch (HaRTES)

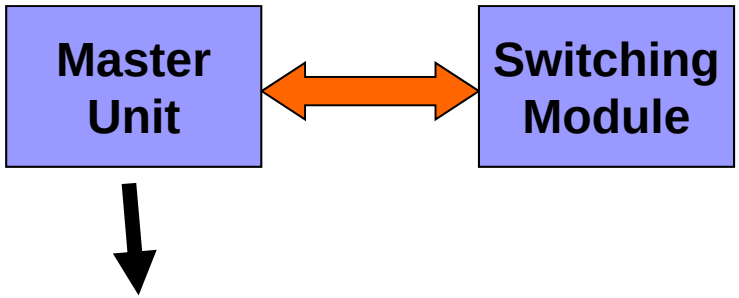
## Implementation

### Top Level



# Implementation

## Master Unit & Switching module integration



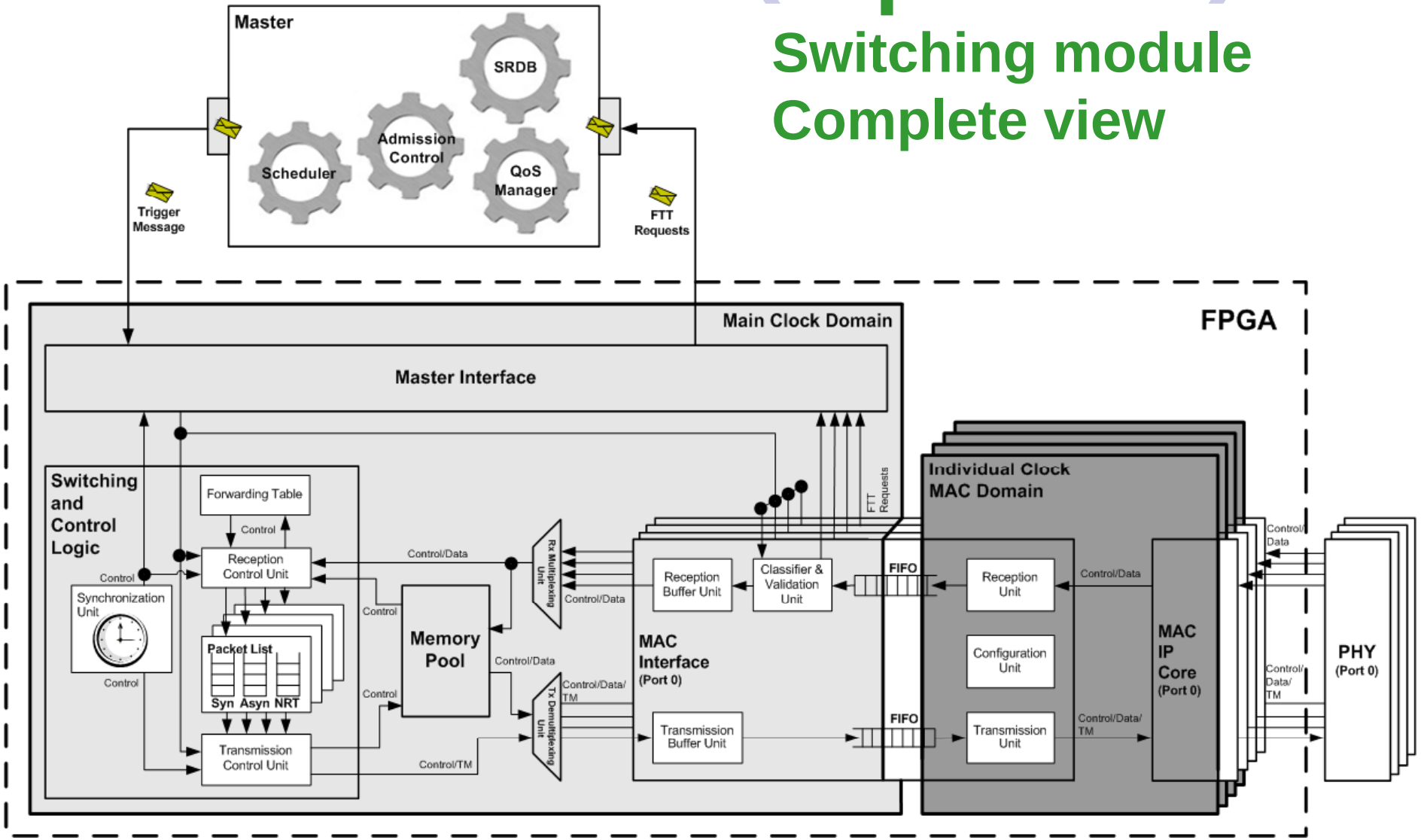

- **Utilization of an FPGA embedded processor (Synthetizable or Hardwired)**
  - More FPGA resources required
- **Independent CPU – communication with the FPGA is carried out by conventional interface (Ethernet, USB PCI, …)**
  - More expensive
  - More free space in the FPGA
  - Sharing the FTT-Master between FTT-SE version and FTT-Enabled Switch

# FTT-E. Switch (HaRTES)

## Implementation

### Switching module

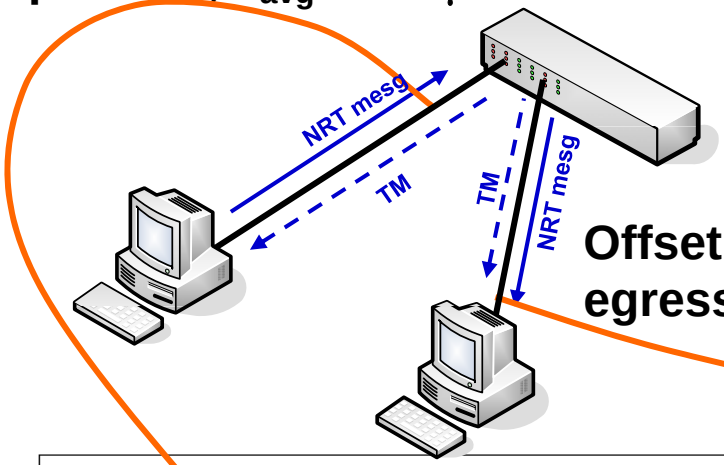
### Complete view

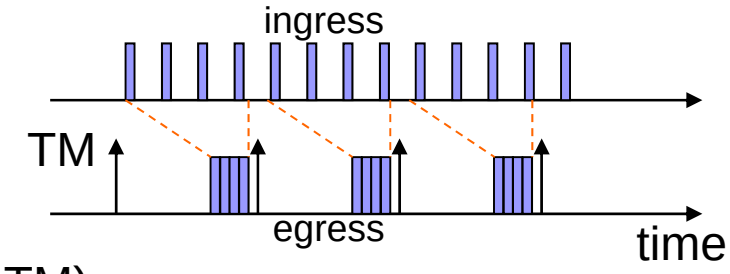


# Experimental Results

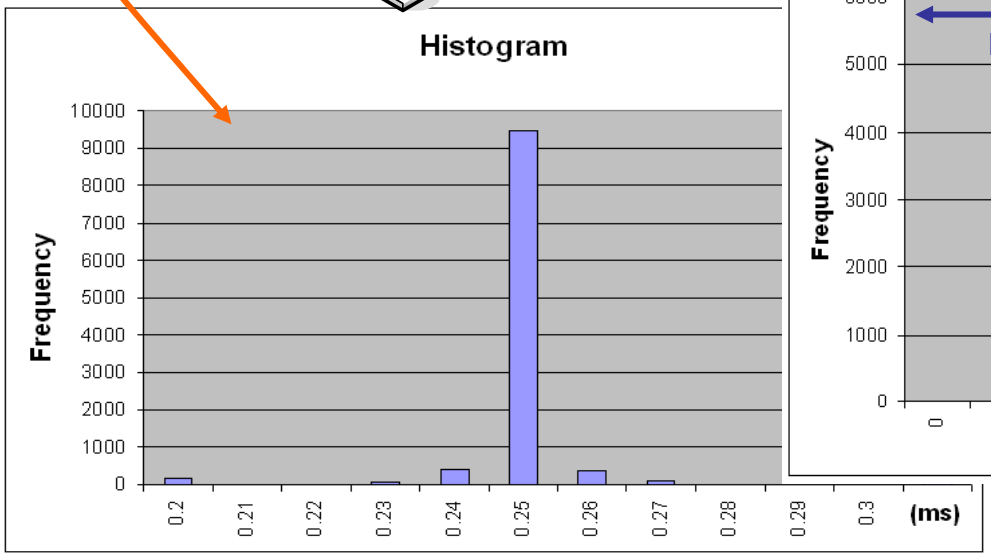
## Confinement of NRT Traffic

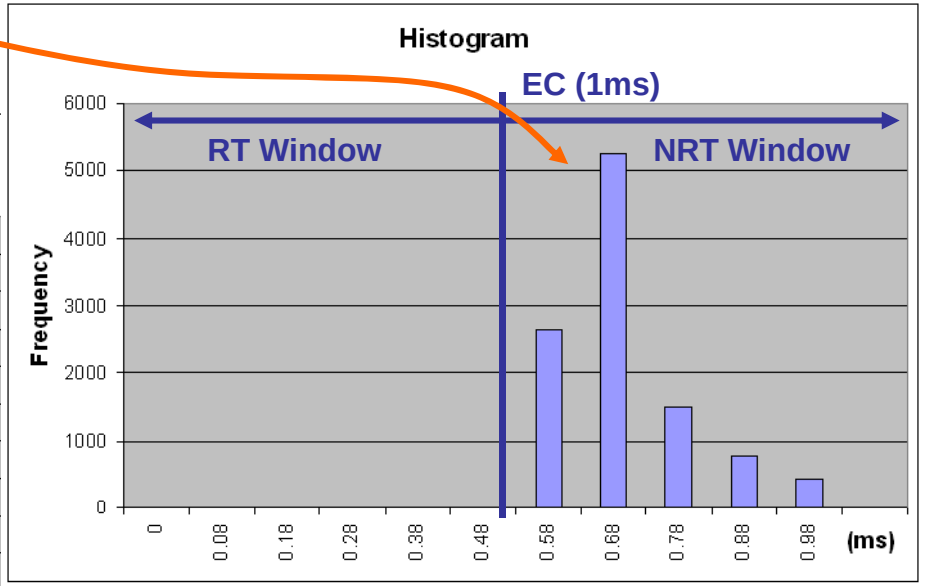
Submitted traffic
**1kB packets, $T_{avg}$ = 250µs**



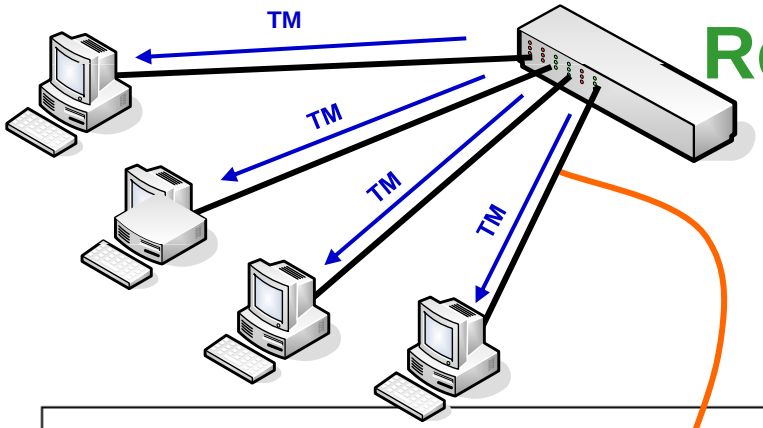

**Offset** at the **switch egress** (relative to the TM)

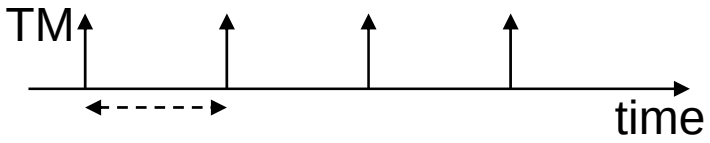


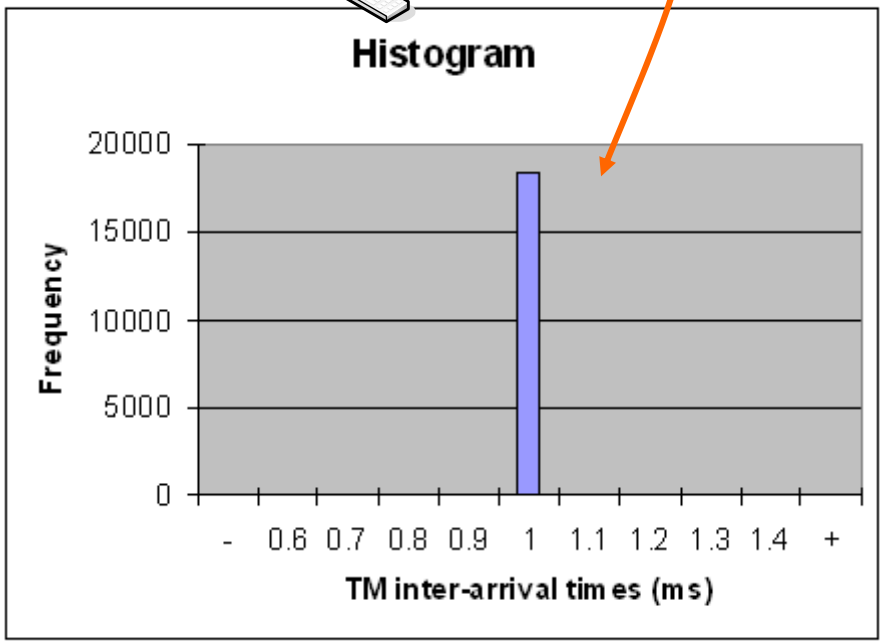



# Experimental Results

## Regularity of the TM




- Measures:
  - $T\_TM_{avg}$ = 1,000ms
  - $T\_TM_{max}$ = 1,0003ms
  - $T\_TM_{min}$ = 0,99998ms
  - STD_TM = 138ns

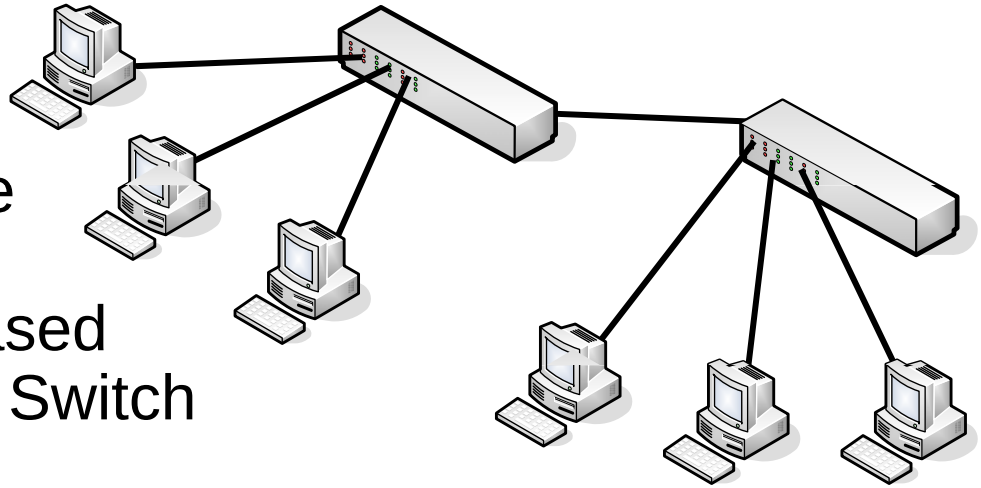
Jitter purely from the switch

# FTT-E. Switch (HaRTES) Current Status

- **HaRTES/B**
  - Basic switching — Executed
  - Capability to separate different traffic classes — Partially executed
  - On-line scheduling — Not yet executed
- **HaRTES/S**
  - Error detection — Partially executed
  - Traffic policing
- **HaRTES/Q**
  - Dynamic QoS management capabilities — Not yet executed

# Multiple Switches

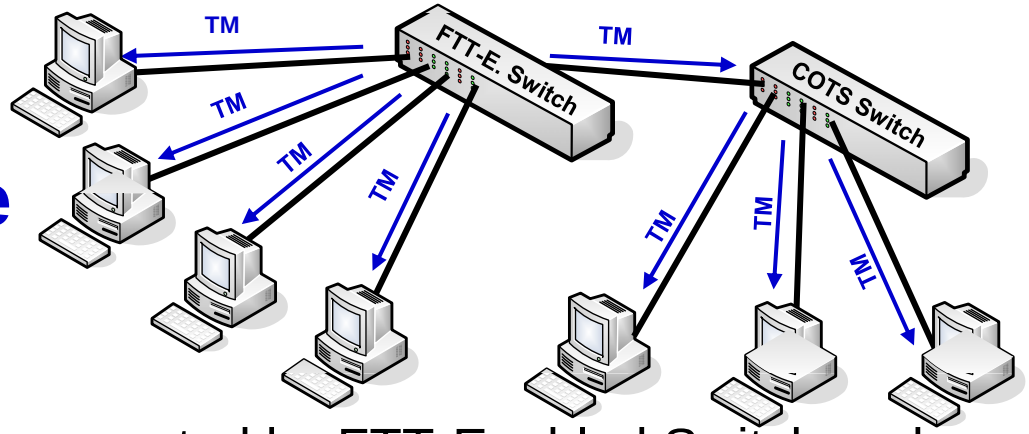
- **Problem**
  - How to create a network with multiple switches, where the communication is based on the FTT-Enabled Switch (HaRTES)?



- **Solutions**
  - **Network with one FTT-Enabled Switch and multiple COTS switches**
  - **Network with multiple FTT-Enabled Switches**

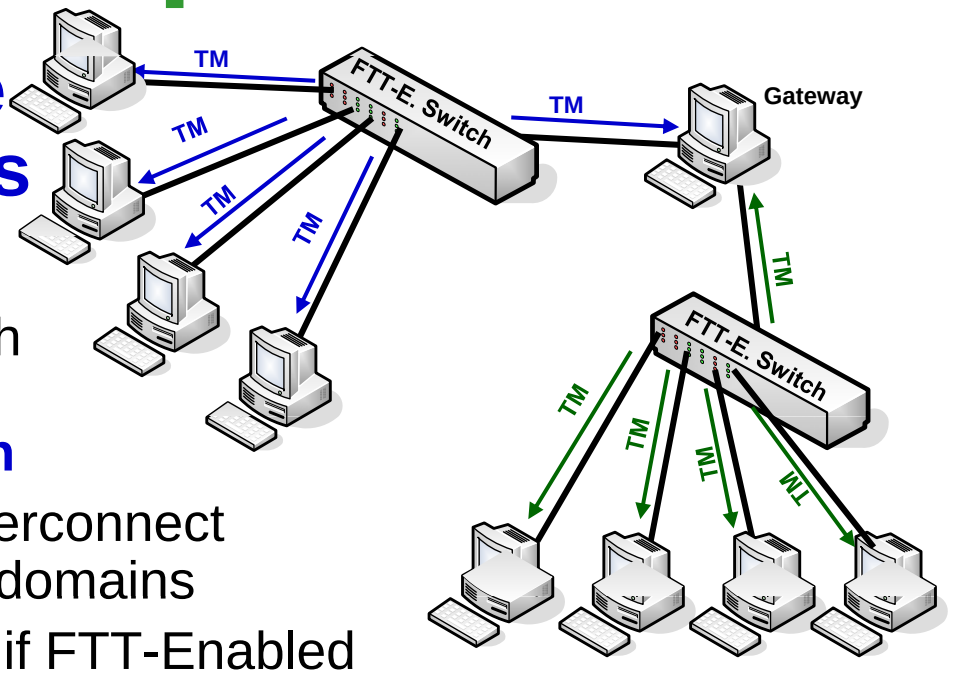
# Multiple Switches

- **Network with one FTT-Enabled Switch and multiple COTS switches**
  - **Properties**
    - Trigger Messages are generated by FTT-Enabled Switch and disseminated by the others switches
  - **Advantages and Disadvantages**
    - ✔ Solution compatible with common networks
    - ✔ COTS switches are cheaper
    - ⊘ COTS switches don’t perform traffic policing
    - ⊘ The Trigger Message latency can generate problems of synchronization



# Multiple Switches

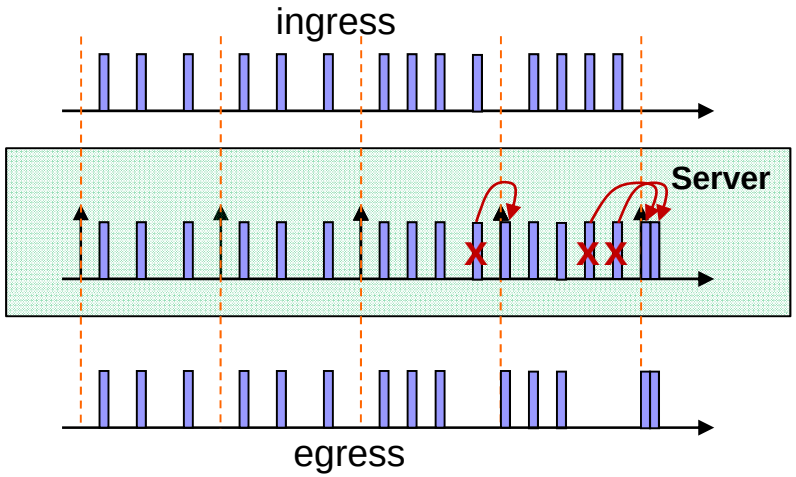
- **Network with multiple FTT-Enabled Switches**
  - **Properties**
    - Each FTT-Enabled Switch creates its own **synchronization domain**
    - It needs a **gateway** to interconnect different synchronization domains
    - Gateway can be avoided if FTT-Enabled Switches are slaves to each other
  - **Advantages and Disadvantages**
    - ✔ Whole network is covered by the traffic policing
    - ⛔ It needs a gateway
    - ⛔ More expensive



# Server / FTT-E Switch

# Another protocol

- **Motivation**
  - Synchronous (TT) Real-Time Ethernet protocols have difficulties in efficient handling messages streams that are asynchronous
    - Video streams, Alarms, Sensors, …
- **Solution**
  - We propose to integrate CPU based server policy in the FTT-Enabled Switch
    - Polling Server, Deferrable Server, Sporadic Server, …
  - Providing **reconfigurability** and **adaptability**
    - Online creation, deletion and adaptation of servers




- **Advantages**
  - ✔ Full control over streams of messages, no matter the arrival patterns
  - ✔ Unnecessary to send trigger message to the slaves.

# Conclusions

- The growing availability of FPGAs, associated tools and communication IP cores opens the way to build customizable devices with properties that are tuned to specific application domains
- We propose an enhanced Ethernet switch that brings substantial improvements in timeliness, integrity and operational flexibility:
  - Isolation of traffic classes
  - Integration of standard Ethernet nodes
  - Transmission of the Trigger Message with high precision
- The proposed hardware/software partition allows reusing the FFT-SE Master with minimal adaptations

# Future Work

- Finish the propose work on the project
- Integrate multiple switches
- Adapt the enhanced switch to allow integration in architectures with multiple synchronization domains
- Replicate the Master
- Integrate CPU based server policy