Server-based Traffic Scheduling in Ethernet Switches

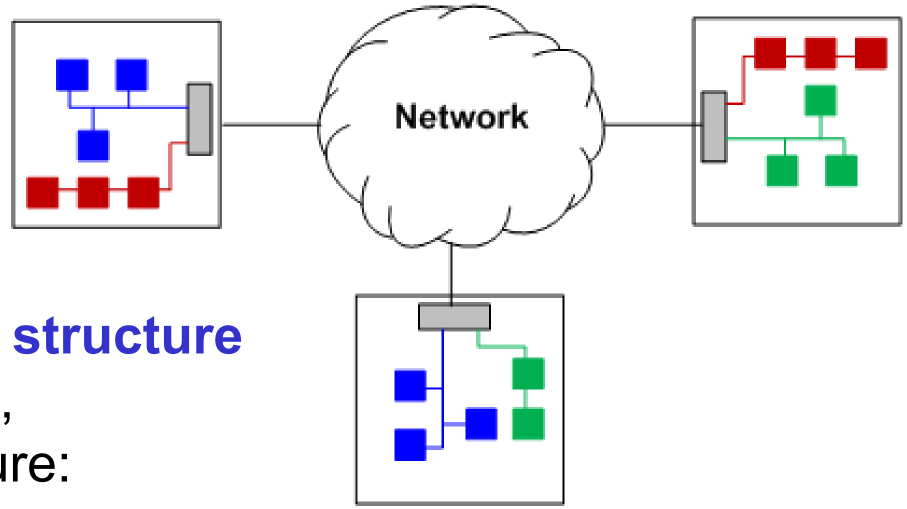
Rui Santos, *rsantos@ua.pt*

Outline

- Motivation
- Server-based traffic scheduling
    - Implementation architectures
    - Online schedulability analysis algorithm
    - Application in FTT-enabled Ethernet Switch
- Conclusions

Motivation

- **Component-oriented design techniques** are increasingly used in the development of Networked Embedded Systems (NES), as a strategy to:
  - Handle the increasing system complexity
  - Allow efficient resource sharing



- From the application of these techniques results a **compositional structure** of subsystems. Its distributed nature, require a communication infrastructure:
  - Flexible, adaptable and deterministic
  - Able to mirror the composability

Motivation

- **Switched Ethernet** and derivative protocols are becoming important technologies in NES due to the:
  - High bandwidth, low cost, hardware availability,…
  - Real-time Ethernet (RTE) protocols provide QoS guarantees

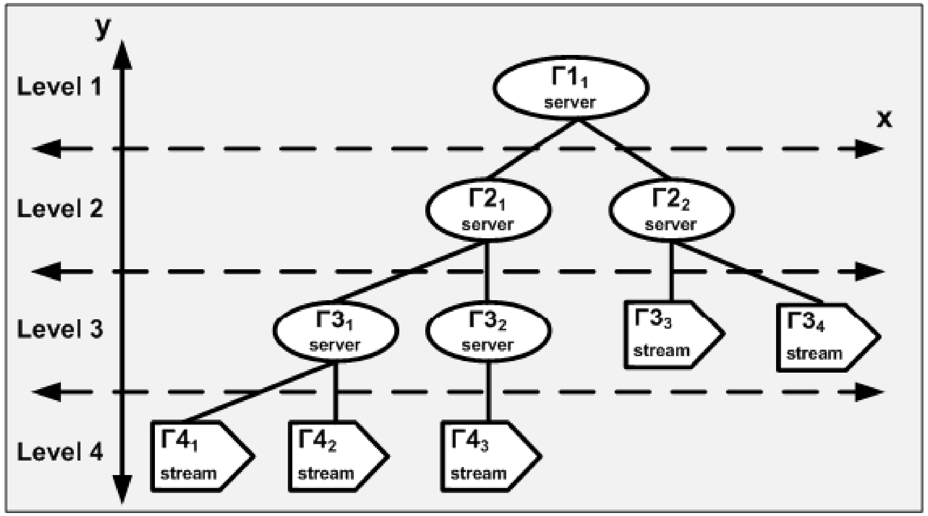
- But, in general these **Real-Time Ethernet** protocols use **static** approaches for the real-time traffic and they have difficulties to support:
  - Heterogeneous nodes, information and requirements
  - Dynamic environments
    - Joining/departing stream and nodes
  - Hierarchical composition of real-time channels

Motivation

- Summing up:
  - The lack of efficient support to hierarchical composition in most of RTE protocols prevents its use in complex NES with component-oriented design techniques or otherwise, it limits the use of these component-oriented techniques in NES.

Solution

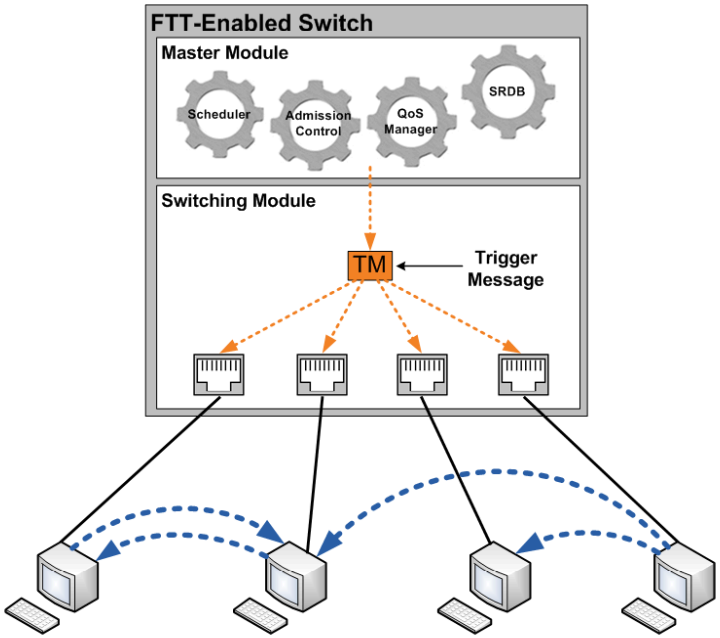
- We propose a **flexible multi-level hierarchical server-based scheduling framework** for switched Ethernet that allows:
  - To divide, subdivide and so on the network bandwidth in a hierarchical way, creating virtual channels
    - Composability in the time domain
  - Real-time guarantees and differentiated QoS (using admission control)
  - Reconfiguration and adaptation of channels for varying flows
  - Assures temporal isolation among channels



Architectures

Servers Implementation

- We have been presented and discuss two different **server implementation** options for the FTT-enabled Ethernet switch:
  - in **software**
    - Servers scheduled within in the Master Module
  - in **hardware**
    - Server scheduling in the Switching Module



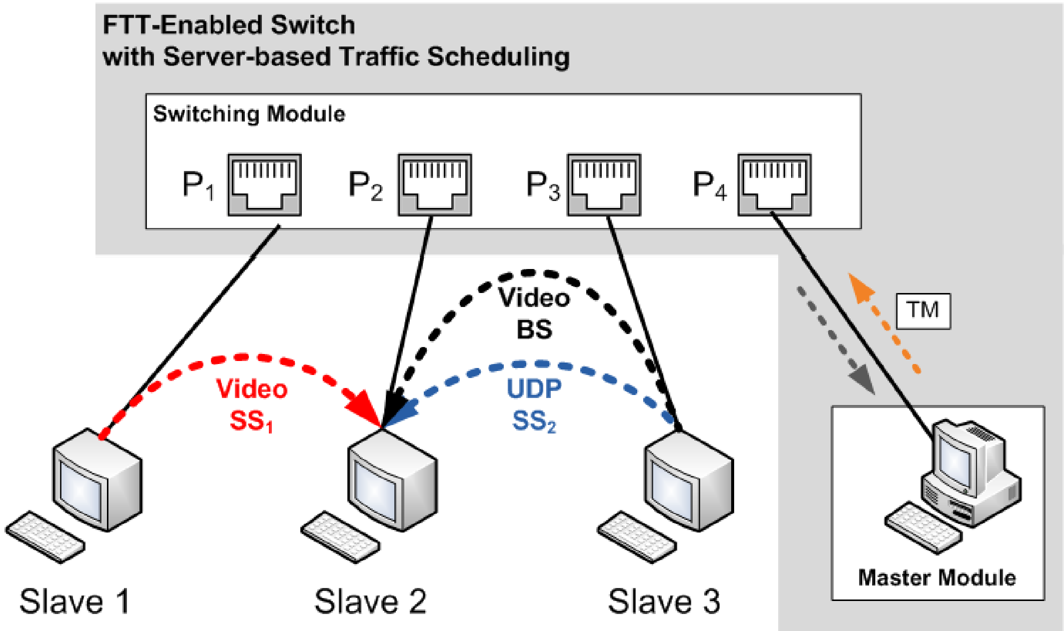
- These options are compared in terms of **flexibility** and **responsiveness**

Comparison

- **Servers implemented in software – Master Module**
  - Higher flexibility
    - Arbitrary number and type of servers (servers are SW entities)
    - Scheduling complexity can be large
  - The server latency is relatively large and depends on the EC duration
- **Server implemented in Hardware – Switching Module**
  - Higher reactivity
  - Lower flexibility
  - Complex server scheduling methods can require a significant amount of hardware resources

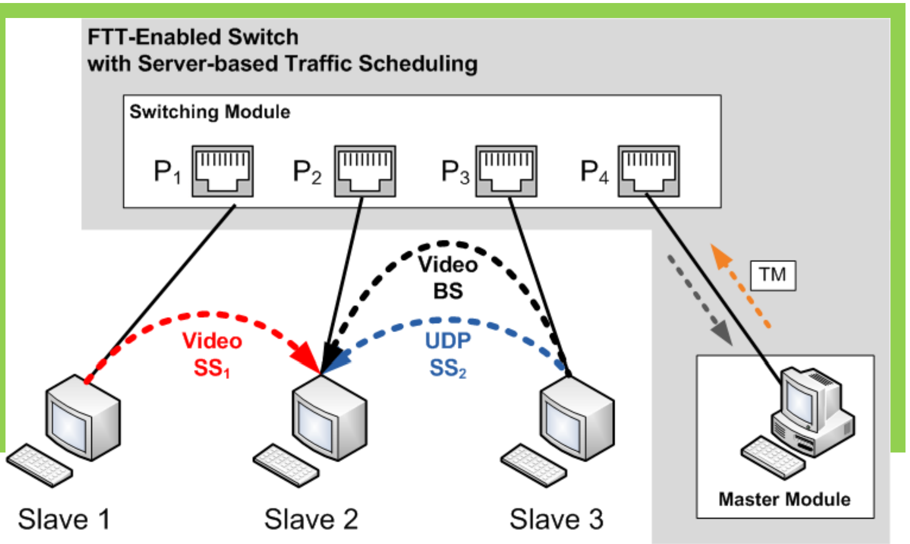
Experimental Validation (correctness of the servers)

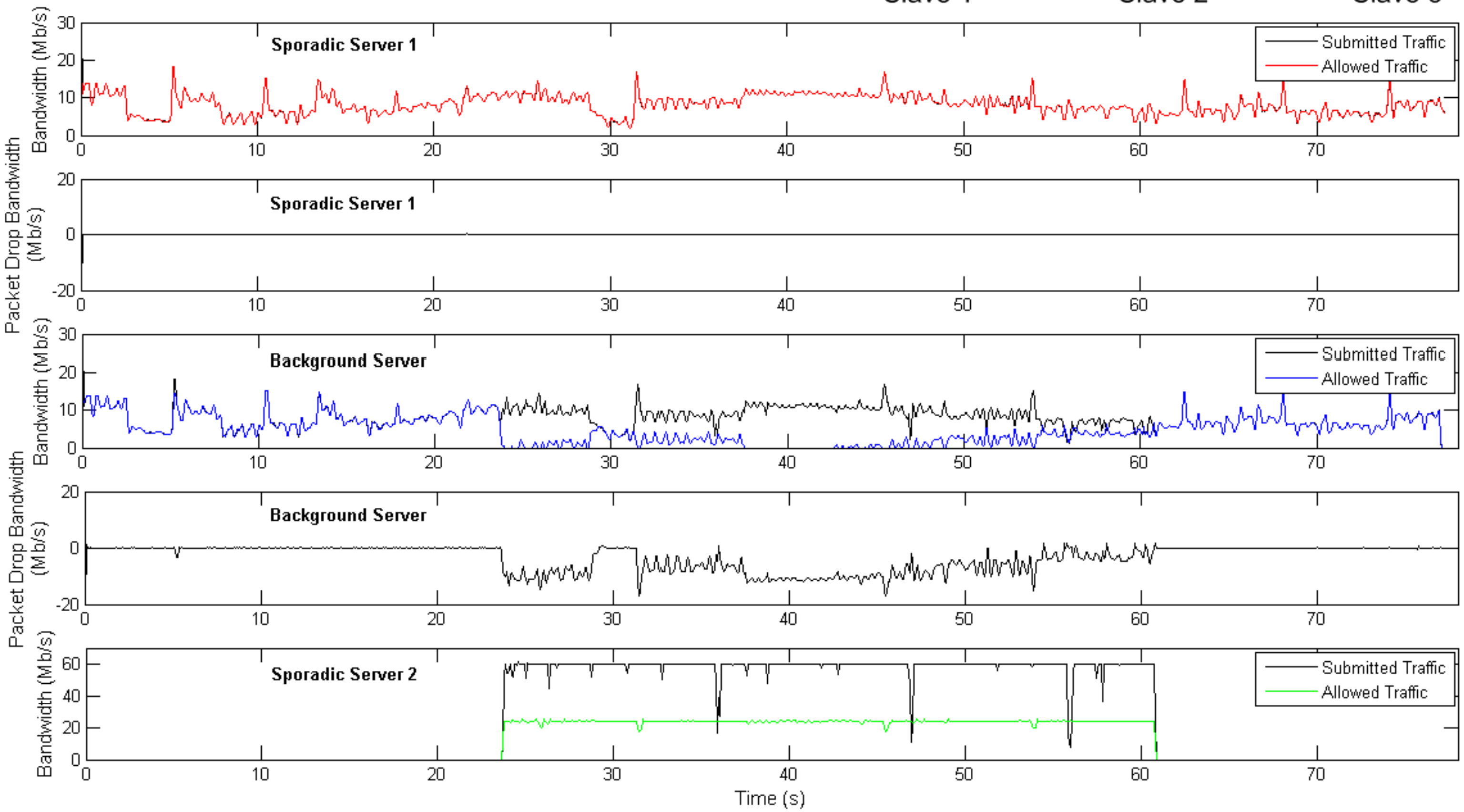
- Elementary Cycle = 1ms; Asynchronous Window = 42%
- SS1, SS2 – sporadic servers with C=3200B and T=1ms
- BS – background server uses the remaining bandwidth



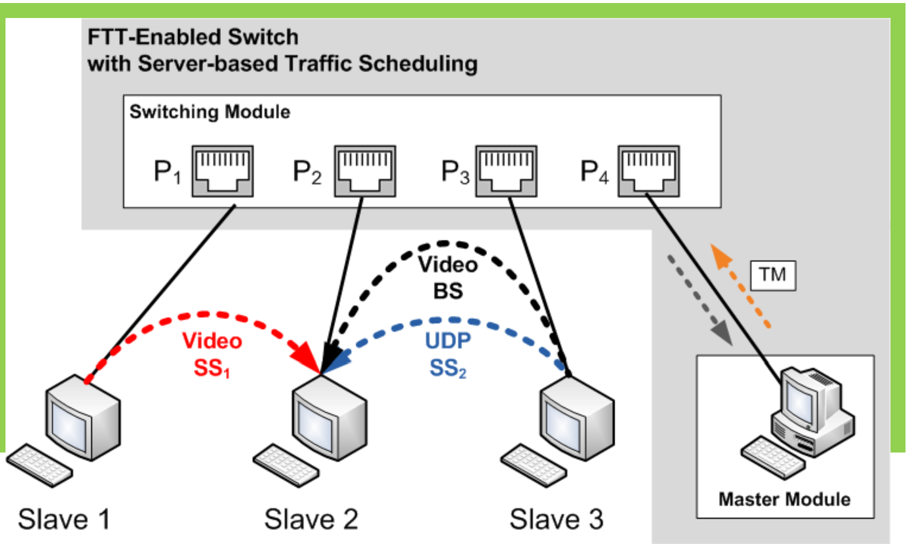
- Video sent by SS1 and BS
  - Peak load = 21.9Mbps
- UDP traffic by SS2
  - Average load = 99.9 Mbps

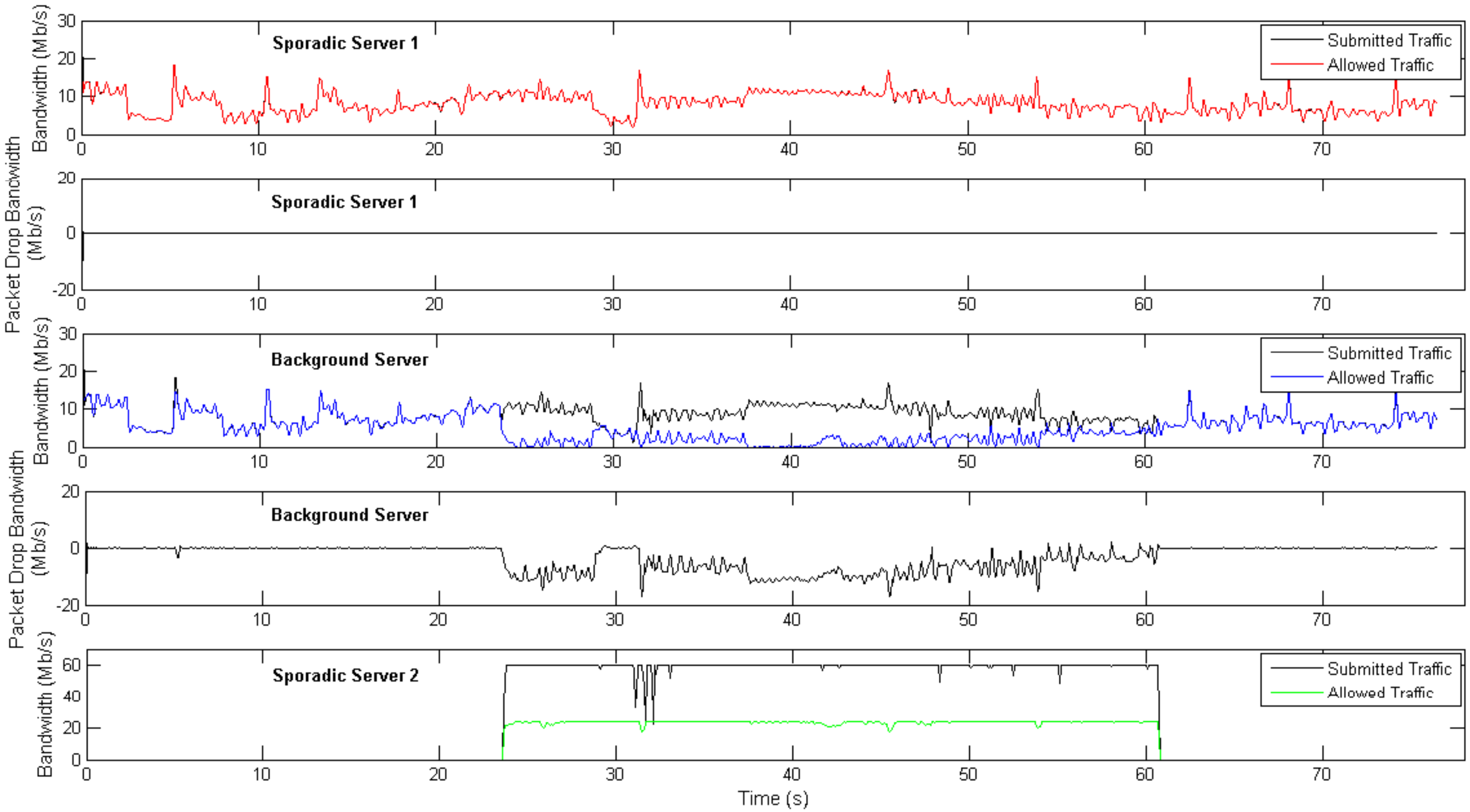
Experimental Results (correctness of the servers- software implementation)



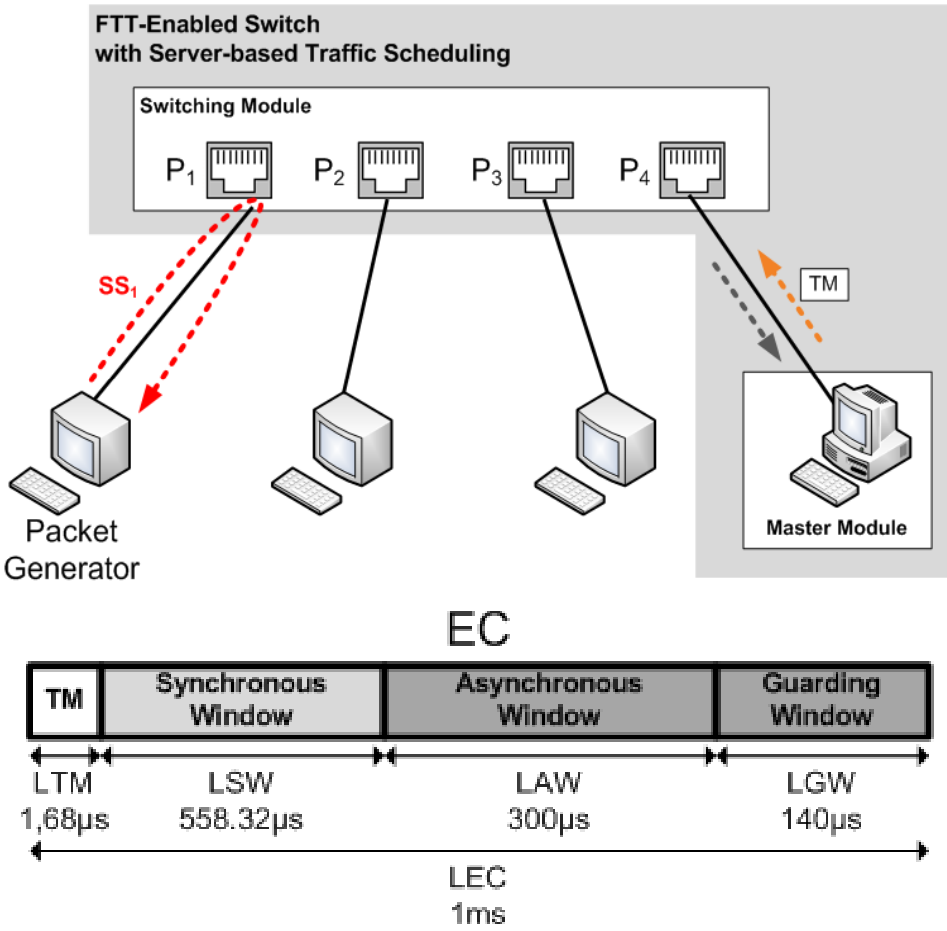


Experimental Results (correctness of the servers- hardware implementation)

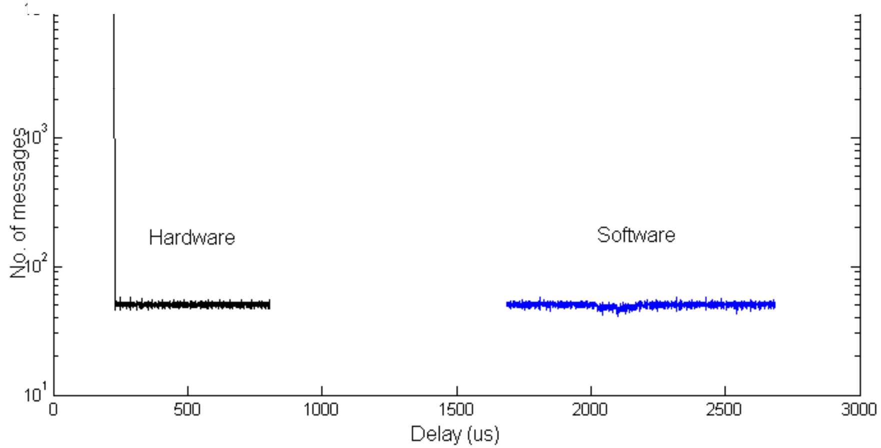




Experimental Results (assess the reactivity)



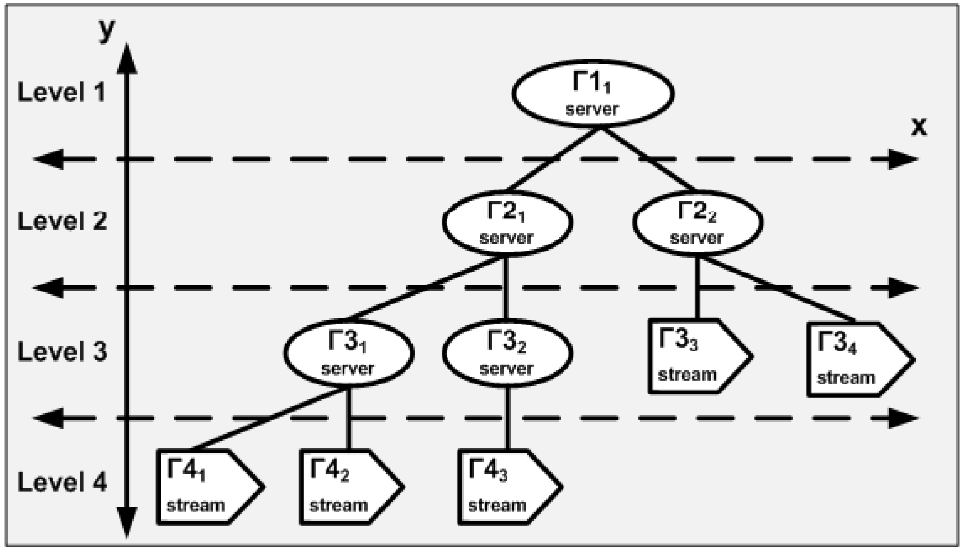
- Packets period = 1 EC + delta
  - To rotate the phase with respect to the EC
- **Round trip delay** measured at the generator node
  - Results match estimates



Online Schedulability Analysis

Model Definition

- Servers: $Srv_{y_x} = (C_{y_x}, T_{y_x}, Mmax_{y_x}, Mmin_{y_x}, P_{y_x}, RT_{y_x})$
- Streams: $AS_{y_x} = (C_{y_x}, Tmit_{y_x}, Mmax_{y_x}, Mmin_{y_x}, P_{y_x}, RT_{y_x})$



C_{y_x} - Maximum transmission time

$Mmax_{y_x}$ - Transmission time - largest packet

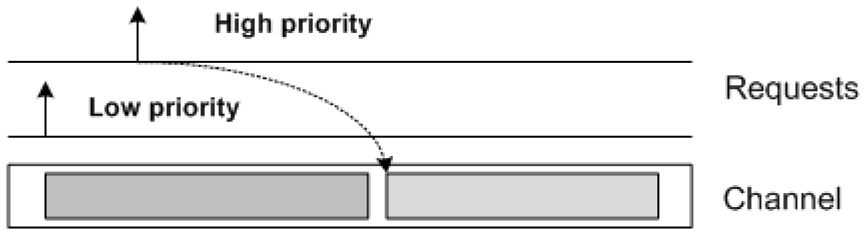
$Mmin_{y_x}$ - Transmission time - smallest packet

P_{y_x} - Parent component (server)

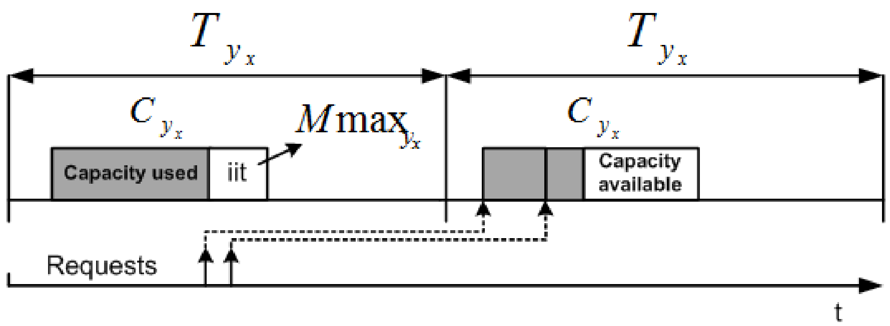
RT_{y_x} - Response time of each component

Model Definition

- The application of this model to a switched Ethernet (FTT) architecture has to assume some restrictions:
  - The preemption of packets that are in transmission process is not allowed



  - Overruns are not allowed in a server component, in order to guarantee strict temporal isolation among servers. Therefore idle time (iit) is inserted in the end of each server



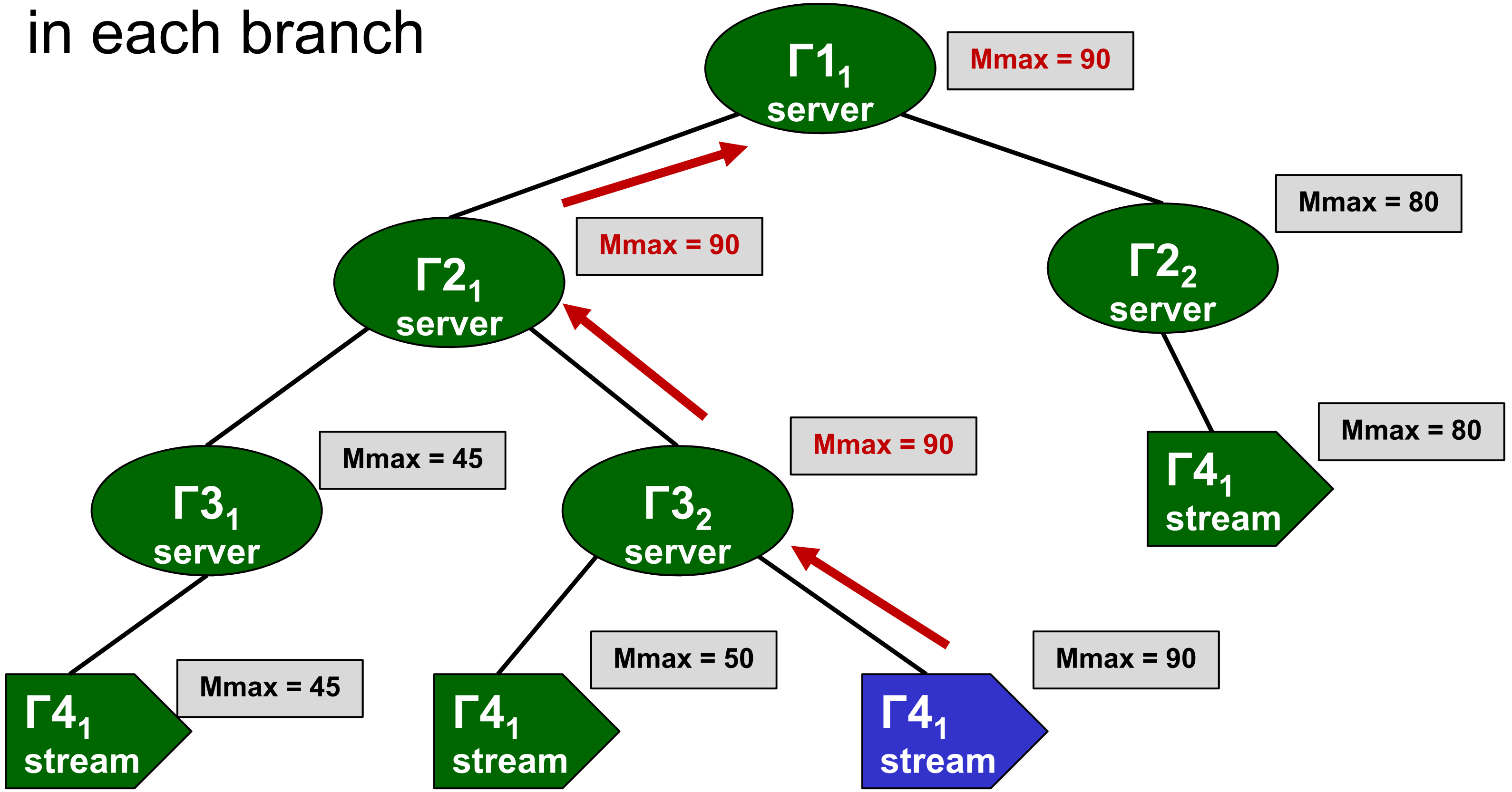
Schedulability Analysis Algorithm

- The **schedulability analysis algorithm** is used to guarantee that online changes in the hierarchy do not affect the real-time guarantees of running streams.

- The proposed schedulability analysis is based on **response time** of the messages inside the servers and it is performed in two phases:
  - **First phase** – verifies the impact of the requested change in the maximum iit in each branch
  - **Second phase** – verifies the schedulability analysis of each component from the top to the bottom of the hierarchy

Schedulability Analysis Algorithm
(first phase)

- From the bottom to the top, this phase aims to find the *Mmax* (maximum iit) *and Mmin* packet transmission time in each branch



Schedulability Analysis Algorithm
(second phase)

- From the top to the bottom, this phase verifies the schedulability of each component and consequently the schedulability of all system
  - For that, a schedulability analysis under FPS is used

$$\forall \Gamma_{y_x} \exists t : 0 < t \leq T_{y_x}, rbf_{y_x}(t) \leq sbf_{P_{y_x}}(t)$$

$$RT_{y_x} = \text{shortest } t^* : rbf_{y_x}(t^*) = sbf_{P_{y_x}}(t^*)$$

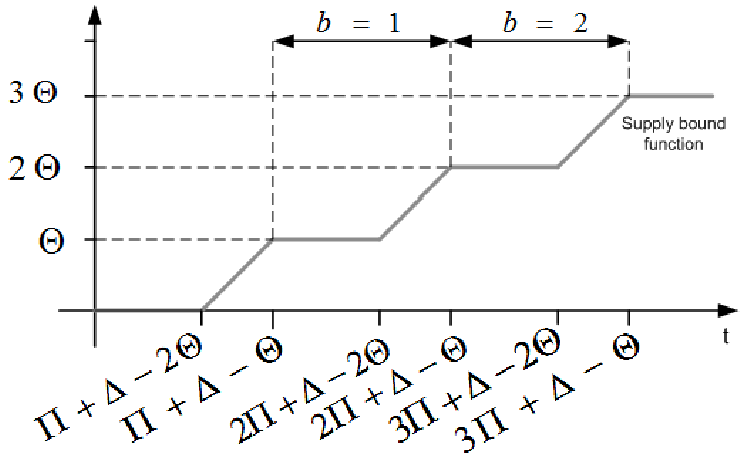
  - ***rbf(t)*** quantifies the maximum load submitted to the parent component *P*
  - ***sbf(t)*** computes the minimum bandwidth supply provided by the parent component to its children

Schedulability Analysis Algorithm
(second phase)

- Supply Bound Function
  - Used the *Explicit Deadline Periodic (EDP)* resource model [A. Easwaran et al, RTSS 2007] that generalizes the periodic resource model for compositional analysis of HSFs
    - Π – Period - **(period of the server component)**
    - Θ – Units of the resource - **(capacity of the server component - Mmax)**
    - Δ - Time units during, which the resource units are available - **(response time of the parent server component)**

$$sbf_{\Gamma_{y_x}} = \begin{cases} b\Theta_{y_x} + \max\{0, t - a - b\Pi_{y_x}\}, & t \geq \Delta_{y_x} - \Theta_{y_x} \\ 0, & \text{otherwise} \end{cases}$$

$$a = \Pi_{y_x} + \Delta_{y_x} - 2\Theta_{y_x} \quad \text{and}$$
$$b = \left\lfloor \left(t - \left(\Delta_{y_x} - \Theta_{y_x}\right)\right) / \Pi_{y_x} \right\rfloor$$



Schedulability Analysis Algorithm
(second phase)

- Requested Bound Function

$$rbf_{y_x}(t) = C_{y_x} + IH_{y_x}(t) + IL_{y_x}$$

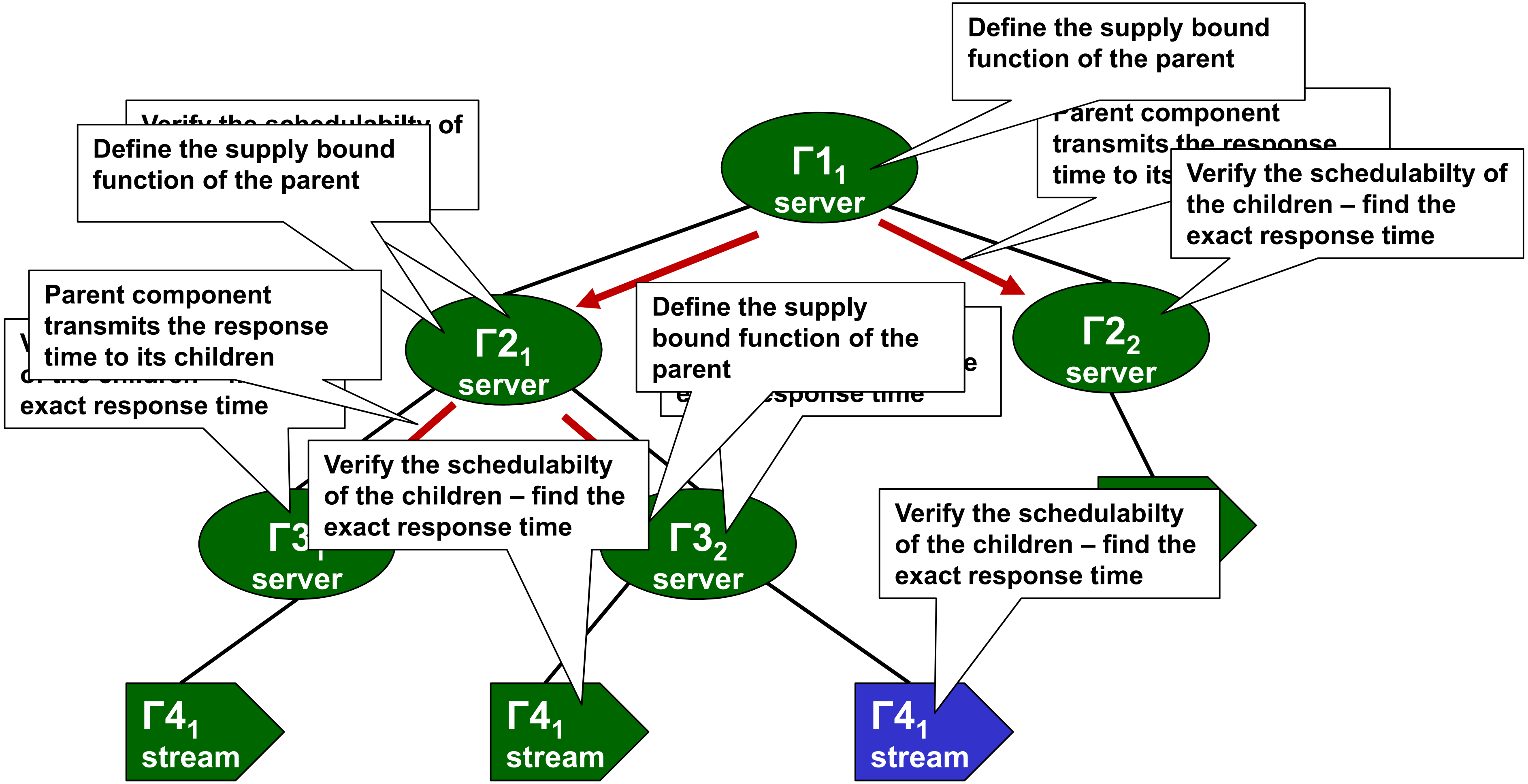
$$IH_{y_x}(t) = \sum_{\Gamma_{y_j} \in hp_{y_x}} \left\lceil \frac{t}{T_{y_j}} \right\rceil \times C_{y_j}$$

Interference from the components with higher priority in the same level and branch

$$IL_{y_x} = \max_{\Gamma_{y_j} \in lp_{y_x}} M\max_{y_j}$$

Blocking from components with lower priority in same level and branch

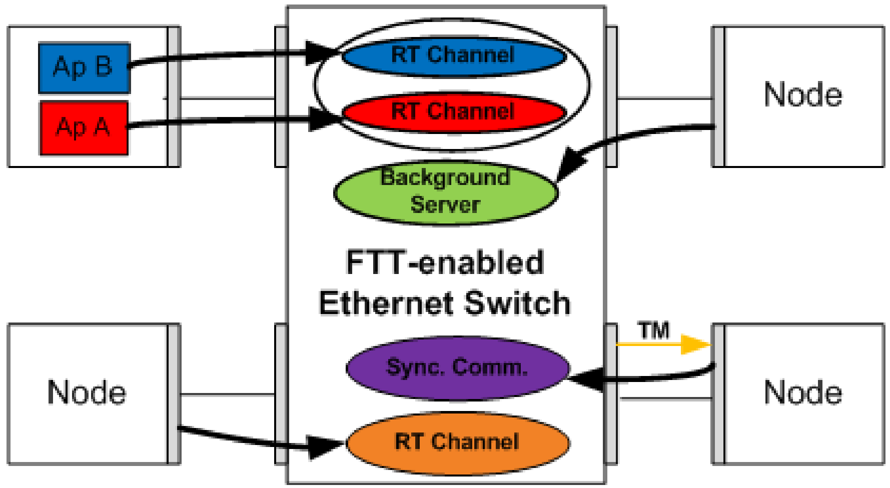
Schedulability Analysis Algorithm
(second phase)



Application

Application in FTT-enabled Ethernet Switch

- The integration with the FTT-enabled Ethernet switch allows:
  - Efficient management of the asynchronous window
  - Highly configurable RT channels (different BW and latency)
  - RT channels can be allocated to nodes or applications inside the nodes
  - RT channels can be created by third party nodes
  - All ports have a background server by default



Conclusions and Future Work

- **Component-based design** is a powerful design paradigm to address the growing complexity of embedded applications and networks
- We are addressing the **multi-level hierarchical server scheduling** within Ethernet switches, namely FTT-enabled Ethernet switch
- A **schedulability analysis** that allows verifying whether a given composition of server is schedulable have been developed.
- As future work we propose to evaluate, test and compare this approach with others similar

Thank You