# A Demonstration of Hierarchical Scheduling over Ethernet


Rui Santos, Paulo Pedreiras
*IEETA / University of Aveiro*
*Aveiro, Portugal*
{*rsantos, pbrp*}*@ua.pt*

Moris Behnam, Thomas Nolte
*MRTC / Mälardalen University*
*Västerås, Sweden*
{*moris.behnam, thomas.nolte*}*@mdh.se*

Luis Almeida
*University of Porto*
*Porto, Portugal*
*lda@fe.up.pt*


## I. Introduction

The complexity of Distributed Embedded Systems (DES) has been growing steeply, due to increases both in size and functionality, and is becoming a major development concern. This situation is pushing for paradigm changes in DES design methodologies towards higher composability and flexibility. Component-oriented design technologies, in particular supported by server-based scheduling, seem to be good candidates to provide the needed properties.

In response to such trend, we developed a multi-level hierarchical server-based architecture for Ethernet switches that provides composability and supports online adaptation and reconfiguration [1] [2]. This framework exhibits the following features: 1) a hierarchical architecture that supports server composition and virtual real-time channels, providing temporal isolation (composability in the time domain); 2) analytical tools for guaranteed real-time behavior; 3) simple interface to adapt and reconfigure servers during runtime.

## II. Hierarchical server-based traffic scheduling

The hierarchical server-based traffic scheduling framework has been presented as a way to provide composability of real-time channels that handle communications in each output port of an Ethernet switch. This framework can be organized as a hierarchical structure of entities, represented as an inverted tree (Figure 1). Each branch of the tree represents a server that handles a portion of the total bandwidth assigned by the parent server.

The hierarchical server framework herein presented supports both server adaptation (the server attributes such as capacity and period may be adapted online) and hierarchy reconfiguration (servers may be added, moved to different branches or even removed). Each change request is subject to a schedulability analysis algorithm, which rejects every request that may compromise the system timeliness, therefore continued real-time behavior is inherently guaranteed.

## III. Application Platform

A prototype implementation of the hierarchical server-based traffic scheduling framework has been deployed within the FTT-enabled Ethernet switch to manage the asynchronous traffic. This is an instantiation of the FTT (Flexible Time-Triggered) communication paradigm in a customized Ethernet switch. According to this paradigm, the communication is organized as an infinite sequence of fixed duration time slots denoted Elementary Cycles that, in turn, comprise two windows, synchronous and asynchronous. The former window is used for isochronous communication (time-triggered traffic) and is managed by a master implemented inside the switch. The asynchronous window is used for asynchronous communication (sporadic and aperiodic traffic) and is managed by the hierarchical server-based traffic scheduling herein described. Each asynchronous message or set of asynchronous messages is associated with a specific server and they are autonomously sent by the end nodes. Upon reception of an asynchronous message, the switch logic inspects the message to identify and queue it in the corresponding server buffer. Then, the messages will be scheduled by all the servers of the hierarchy in the path to the top, before being transmitted on the channel.

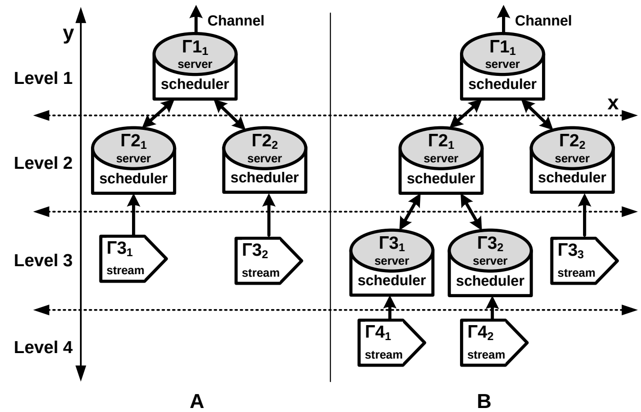


Figure 1. Example of server hierarchies.

Naturally, the FTT-enabled Ethernet switch also has a limited set of physical resources. Namely, each server instantiation requires some logic to handle the budget accounting, memory to hold the server attributes and buffer memory to hold messages. Therefore, the maximum number of servers that can be instantiated in a switch is limited. However, due to the design techniques employed, this is the only restriction that applies to the switch. In particular, the hierarchical server structure is fully configurable.

## IV. Demonstration

Due to its practical relevance, the demonstration is focused on the composability in the time domain, predictability and reconfigurability of real-time channels. For that, we propose two demos based on the setup presented in Figure 2.

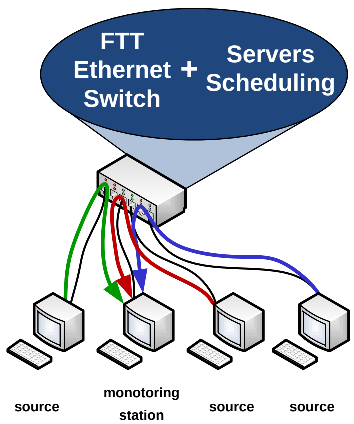


Figure 2. Demo setup.

### A. Demo 1

The first demonstration allows us to show the strict periodicity of the FTT-enabled switch isochronous services and the traffic isolation between isochronous services in conjunction with asynchronous traffic. Finally, we also show the temporal guarantees provided by the multi-level hierarchical server-based architecture.

For that, two video streams (from IP cameras, MPEG320x240, 30fps) are transmitted to the monitoring station, using the FTT asynchronous window through the server hierarchy presented in Figure 1A. Moreover, one periodic data source is transmitted using the synchronous window (i.e. an isochronous channel) to the monitoring station which shows the fps, bandwidth and jitter for each traffic source.

This demonstration will show the constant level of the bandwidth, latency and jitter of the isochronous traffic, i.e., it is not sensitive to the other traffic (Figure 3). Moreover, the operation of both cameras is completely independent, i.e, structural changes on one of the images do not impact on the other traffic sources, since they are managed by the corresponding servers.

### B. Demo 2

The second demonstration shows the dynamic reconfiguration of the servers and the traffic isolation among the asynchronous streams. For that, three video streams (from IP cameras, MPEG320x240, 30fps) are transmitted to the monitoring station, using the FTT asynchronous window

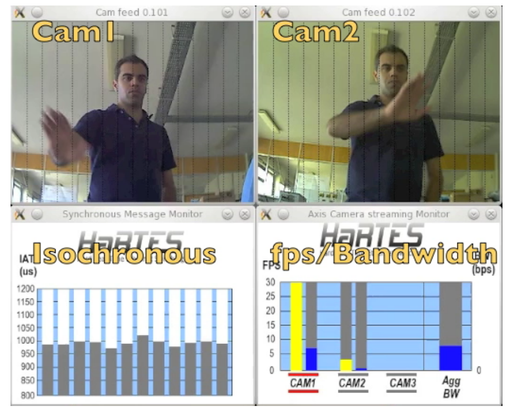


Figure 3. Monitoring station.

through the server hierarchy presented in Figure 1B. During this demonstration the QoS given to each camera (video stream) can be updated, either directly by a human console operator or automatically by the application software, in reaction to changes in the environment .

This demonstration will show the real-time dynamic reconfigurability capabilities of the switch, since changes are made online without any impact on the streams not affected by the reconfiguration. On the other hand, it will be seen that traffic sources are not allowed to use more bandwidth than the one granted by the servers.

## V. Conclusions

Component-based design is a powerful design paradigm to manage the growing complexity of embedded applications, providing functionality by independent sub-components in a composable way. For distributed environments, a communication framework with composability enforcement and reconfigurability is required. In this demonstration, the hierarchical server-based traffic scheduling framework is validated, namely its composability, reconfigurability and predictability, showing that hierarchical server-based traffic scheduling can be effectively commissioned in distributed embedded realtime applications.[1]

## References

[1] R. Santos, A. Vieira, R. Marau, P. Pedreiras, A. Oliveira, L. Almeida, and T. Nolte, "Implementing Server-based Communications within Ethernet Switches," in *Proceedings of 2nd International Workshop on Compositional Theory and Technology for Real-Time Embedded Systems*, December 2009.

[2] R. Santos, M. Behnam, T. Nolte, P. Pedreiras, and L. Almeida, "Schedulability Analysis for Multi-level Hierarchical Server Composition in Ethernet Switches," in *Proceedings of the 9th International Workshop on Real-Time Networks*, July 2010.

[1]This work was partially supported by the iLAND project (EU ARTEMIS JU Prog.), by the European Community through the ICT NoE 214373 Artist-Design and by the Portuguese Government through the FCT project HaRTES - PTDC/EEA-ACR/73307/2006 and Ph.D. grant - SFRH/BD/32814/2006.