

# Hierarchical Server-based Traffic Scheduling in Ethernet Switches

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

Distributed Embedded Real-Time Systems (DERTS) are ever more complex, consisting of an increasing number of nodes with more functionality and handling more data. This scenario is pushing for paradigm changes in the methodologies to design complex DERTS. Namely, component-oriented design methodologies, exhibiting the composability property are considered an effective means to tackle the growing complexity of DERTS. Server-oriented architectures have proven to be an effective means to enable controlled resource sharing, providing composability among application components as well as between the application software and the hardware platform on which they execute.

Real-Time Ethernet-based protocols (RTE), such as PROFINET, EtherCAT, ETHERNET-Powerlink and TTEthernet, are emerging as the de facto communication technologies for wireline-based DERTS, taking advantage of the appealing attributes of the underlying Ethernet protocol (e.g. large bandwidth, cheap silicon), while removing or reducing the sources of non-determinism arising from its MAC protocol and/or from the standard switched architectures. However, currently available RTE protocols do not allow efficient server-based scheduling policies. Even when available, network partitions are typically static, as in TDMA-based approaches. Moreover, the respect for network partitions is frequently delegated to the end nodes that must execute a specific software layer, preventing the integration of legacy systems and other general purpose systems. Finally, even in the cases in which such a layer can be effectively deployed, the proper operation of the system requires the compliance of all system components to achieve a correct temporal behavior.

To overcome these limitations, to manage the bandwidth in a switched Ethernet network, a hierarchical server-based traffic scheduling framework has been developed by the authors [1] [2]. This framework exhibits the following features: 1) A hierarchical architecture that supports server composition, 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 (create, remove and update attributes).

The purpose of demonstrating our research prototype (in-

cluding hardware) is to illustrate the potential of hierarchical server-based traffic scheduling on Ethernet switches. In particular we will emphasize composability in the time domain, predictability and reconfigurability of real-time channels.

## II. HIERARCHICAL SERVER-BASED TRAFFIC SCHEDULING

The hierarchical server-based traffic scheduling framework has been presented as a way to provide a 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. Therefore, each server (parent) can be connected to several servers (children) that share the bandwidth assigned to the parent. This procedure is recursive, and thus arbitrary server compositions can be designed. The actual data flows are placed at the end of the branches, i.e., the leaves, being connected to the corresponding servers.

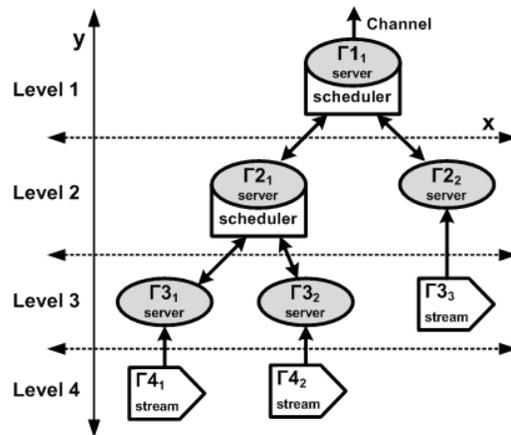


Figure 1. Server Hierarchy.

Many DERTS exhibit requirements that may vary over time. To cope with those evolving requirements efficiently, the hierarchical server framework herein presented supports both server adaptation, i.e., the server attributes such as capacity and period may be adapted online, and hierarchy reconfiguration, i.e. servers may be added, moved to different

branches or even removed. The adaptation and reconfiguration services are accessed via reconfiguration request messages issued by the nodes to the switch. 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 on 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 synchronous communication (time-triggered traffic) and is managed by a master implemented inside the switch (out of the scope of this demonstration). 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 server of the hierarchy in the path to the top, before being transmitted on the channel.

Like any ordinary switch, the FTT-enabled Ethernet switch 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 the server structure depicted in Figure 1 is created to handle the data flows represented in Figure 2. With this setup the following demonstrations will be made:

- **Temporal isolation.** The bus utilization of one of the streams  $\Gamma_{4_1}$ ,  $\Gamma_{4_2}$ ,  $\Gamma_{3_3}$  is monitored while the traffic generation pattern of the other two streams is varied. It is expectable to observe that the link utilization of the monitored stream does not change despite the variation of network load, even when some of the other sources

try to use more resources than what was negotiated (e.g. an babbling idiot scenario);

- **Timeliness.** The response time of one of the streams  $\Gamma_{4_1}$ ,  $\Gamma_{4_2}$ ,  $\Gamma_{3_3}$  is monitored while the traffic generation pattern of the other two streams is varied. The maximum response time of the monitored stream should be verified independently of the load pattern of the other streams;
- **Reconfigurability.** The link utilization of one of the streams  $\Gamma_{4_1}$ ,  $\Gamma_{4_2}$ ,  $\Gamma_{3_3}$  is monitored while the associated server attributes are changed, e.g. the budget is increased. The observed traffic pattern should evolve according to the new attributes, whenever the change request is found feasible and thus accepted by the system.

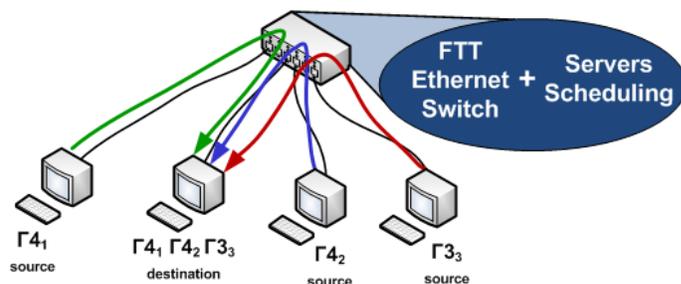


Figure 2. Setup.

### 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 will be validated, namely its composability, reconfigurability and predictability, showing that hierarchical server-based traffic scheduling can be effectively commissioned in distributed embedded real-time applications. <sup>1</sup>

### 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.

<sup>1</sup>This work was partially supported by the iLAND project, call 2008-1 of the EU ARTEMIS JU Programme, by the European Community through the ICT NoE 214373 ArtistDesign and by the Portuguese Government through the FCT project HaRTES - PTDC/EEA-ACR/73307/2006 and Ph.D. grant - SFRH/BD/32814/2006.