Supporting Multi-Hop Communications with HaRTES Ethernet Switches

Mohammad Ashjaei[1], Paulo Pedreiras[2], Moris Behnam[1], Luis Almeida[3], Thomas Nolte[1]
[1] MRTC/Mälardalen University, Västerås, Sweden
[2] DETI/IT/University of Aveiro, Aveiro, Portugal
[3] IT/DEEC/University of Porto, Portugal

***Abstract*—In this paper we identify the challenges of multi-hop communication when using micro-segmented switched-Ethernet with HaRTES switches. These switches provide dynamic virtual channels that can be composed hierarchically and provide bounded latency together with temporal isolation. Herein we propose two different solutions regarding the traffic forwarding in multi-switch architectures, while maintaining the unique properties of the single HaRTES switch case. In the first approach, the traffic is buffered and scheduled sequentially in each hop. In the second solution the traffic is scheduled once and forwarded immediately through multiple switches without buffering. In this paper we present a brief comparison of both approaches and we report on the on-going work towards effective support to real-time communications in dynamic and complex Cyber-Physical Systems.**

I. Introduction and motivation

Cyber-Physical Systems (CPS) are nowadays pervasive, being present in countless aspects of everyday life, such as medical devices, industrial process control and automotive systems, to name a few. The complexity of CPS has grown fast and reached the stage in which conventional technologies and development methodologies reveal limitations [1]. Focusing specifically on the data distribution infrastructure, CPS pose new requirements that are not efficiently handled by existing communication protocols. For example, many CPS incorporate traffic with diverse activation patterns (event- and time-triggered) and timeliness constraints (e.g. hard, soft and firm), that must be handled in a dynamic way, providing on-the-fly reconfiguration support without service disruption.

The inefficient support of existing communication protocols to CPS led to the development of the new Hard Real-Time Ethernet Switches (HaRTES) [2] [3]. Specifically, these switches aim at supporting: 1) heterogeneous traffic classes with temporal isolation; 2) partitioning and virtualization mechanisms; 3) hierarchical multi-level server composition; 4) dynamic adaptation and reconfiguration of message streams with temporal guarantees.

II. Problem Definition

We define the HaRTES architecture as a micro-segmented network using HaRTES switches. We also consider two traffic types, synchronous and asynchronous. The synchronous traffic is scheduled by the switch, following a master-slave paradigm, in a FTT fashion [4]. Asynchronous traffic is transmitted autonomously by the source-nodes and immediately forwarded by the switch through a hierarchy of servers, in order to enforce isolation and to guarantee minimum QoS levels.

The switch organizes the communication in fixed duration time slots, designated Elementary Cycles (ECs). ECs are composed by two disjoint windows, one carrying the synchronous traffic and the other one the asynchronous traffic. Each EC starts with a Trigger Message (TM), transmitted by the switch, which contains the EC-schedule, i.e., the identification of which synchronous messages shall be transmitted in that EC. Synchronous and asynchronous streams can be real-time, being confined to virtual communication channels, following a multicast-based publisher-subscriber model. Each channel has a set of real-time attributes that are enforced by the HaRTES switch. Such attributes include deadline, minimum inter-arrival time/period and priority.

Until this moment, only single switch HaRTES architectures have been considered. In this work we report on the on-going efforts to build multi-switch HaRTES architectures without jeopardizing the unique dynamic, real-time and traffic separation capabilities of the HaRTES switch. Such a problem has already been addressed in the scope of the FTT-SE protocol, which operates over COTS switches with a software layer in the end nodes that controls all traffic submitted to the network [5]. However, the extended traffic control capabilities of HaRTES open the way to more efficient solutions. Thus, we herein propose two possible architectures with multiple HaRTES switches, that allow forming complex topologies, while preserving the timeliness guarantees of the real-time traffic.

III. Proposed Solutions

In order to build architectures with multiple HaRTES switches, we propose connecting them in a tree topology (Fig. 1). In this architecture we define two types of messages. The messages transmitted between nodes connected to the same switch are called *local*, otherwise they are called *global*. Moreover, we define the connections between nodes and a switch as *local-links*, whereas the connections among the switches are defined as *inter-links*. Finally, the HaRTES switch on the top of the tree is called the *root switch*.

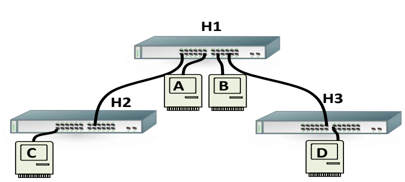

Fig. 1. The Multi-Hop HaRTES Network

A. Distributed Global Scheduling

In this solution, the scheduling of global traffic is carried out in a distributed fashion by all the switches involved. Basically, each switch schedules its hop without distinction of global or local messages. The global messages received by a switch are buffered and scheduled for the next hop in following ECs. The step-wise scheduling of the global messages continues until the last switch, where the destination node is connected to.

Assume a network depicted in Figure 1 and consider m_1 to be transmitted from node C to node D. H2 schedules m_1 to be transmitted to H1. H1 then schedules m_1 to be transmitted to H3 with an adequate offset so that it always occurs after the reception of m_1 from H2. Finally, H3 schedules m_1 for transmission to node D, again using an adequate offset.

The definition of consistent offsets in the transmission of synchronous messages across multiple switches requires global synchronization. The HaRTES architecture guarantees a contention-free transmission of the TM, which is broadcasted to all the nodes connected to a switch, including other switches down the tree topology. Therefore, the TM can be used as a precise time mark. In this proposal, all switches synchronize with their parent switch whenever receiving a TM. The exception is the root switch, which behaves as a time master.

The asynchronous traffic is forwarded immediately by the switches, within the asynchronous windows, only. During the synchronous windows, or when the capacity of associated servers is exhausted, such traffic is suspended and queued.

B. Synchronous Global Scheduling

In this solution, the global messages are scheduled globally by the root switch. The global scheduling in each EC is encoded in a particular message called Global Trigger Message (GTM) which is broadcasted to the other switches during the synchronous window, i.e., the GTM acts as a global synchronous message. The GTM is received by the other switches, it is decoded and the ID's of scheduled global messages are inserted into the local TM at the source switches, only, to trigger the global messages. The switches in the route of the global messages forward them immediately in the same EC in which they are scheduled. This way, the global messages are propagated through the network in just one EC.

The synchronous window is divided into two sub-windows, local and global. This partitioning is carried out just for the local-links. The inter-links are not divided since there is no local traffic transmitted through them. The EC windows allocation is depicted in Figure 2 for different links. Moreover, the assigned global sub-window in each link can vary depending on the global load, which can increase the efficiency in using the links bandwidth.

Since the global messages take longer to propagate through the network due to additional forwarding delays, to guarantee their transmission within the synchronous window we enforce the global traffic to be transmitted by the nodes earlier than the local traffic. This can be done by assigning higher priorities to the global messages or by sequencing the IDs in the TM according to the desired order and forcing end-nodes to follow this sequence.

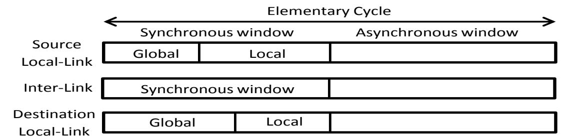

Fig. 2. The EC Partitioning in Different Links

The asynchronous traffic is handled similarly to the previous solution, with immediate forwarding.

Time synchronization among the switches is also required for this solution. The synchronization can be carried out similarly to the previous solution using the TM.

This approach bears some resemblances with the solution for FTT-SE presented in [5]. However, contrarily to the FTT-SE scenario, in this case it is possible to have differentiated windows allocation, i.e., each link may have an individual global window, with a duration computed according to the traffic that effectively crosses that link. This feature is expected to allow a significant performance improvement. In fact, note that the local scheduling can be carried out after the reception of the GTM. Thus, the local scheduler can deduce the effective time taken by the already scheduled global messages (remember they are scheduled 1 EC ahead) and use the remainder of the synchronous window for local traffic.

IV. Conclusion and Future Work

In this paper we identified the challenges that arise from connecting multiple HaRTES switches together. This will allow benefiting from the enhanced dynamic, timing and isolation features of the HaRTES switches, to cater for the requirements of complex CPS. We proposed two solutions to handle traffic forwarding, one is based on buffering the traffic in each hop and the other is by transmitting the global messages in a synchronized manner. The former requires several ECs to transmit the global traffic hop-by-hop, whereas in the latter we scheduled and transmit the global traffic in one EC. Also, in the former solution we need to dedicate a smaller window for the message transmission, while we need a bigger global synchronous window in the second solution to be able to send the traffic through different switches. The on-going work aims at developing the timing analysis for both solutions.

Acknowledgment

This work is supported by the Swedish Foundation for Strategic Research via MRTC at Mälardalen University. Also, it is partially supported by the Portuguese Government through FCT grant Serv-CPS PTDC/EEA-AUT/122362/2010, FCT grants SENODS CMU-PT/SAI/0045/2009 and FEDER (COMPETE program).

References

[1] E. Lee, "Cyber physical systems: Design challenges," in *11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC'08)*, 2008, pp. 363–369.

[2] R. Santos, M. Behnam, T. Nolte, P. Pedreiras, and L. Almeida, "Multi-level hierarchical scheduling in ethernet switches," in *Proceedings of the Int. Conference on Embedded Software (EMSOFT'11)*, Oct. 2011.

[3] "Serv-cps project, available at http://serv-cps.av.it.pt/."

[4] "Ftt fashion, available at http://www.fe.up.pt/ftt."

[5] M. Ashjaei, M. Behnam, L. Almeida, and T. Nolte, "Performance analysis of master-slave multi-hop switched ethernet networks," in *8th IEEE Int. Symposium on Industrial Embedded Systems (SIES' 2013)*, June 2013.