Software-Defined Real-Time Mesh Networking: Protocol and Experimentation Method

Florian Greff, Ye-Qiong Song, Arnaud Samama, Laurent Ciarletta

To cite this version:

Florian Greff, Ye-Qiong Song, Arnaud Samama, Laurent Ciarletta. Software-Defined Real-Time Mesh Networking: Protocol and Experimentation Method. RESCOM’17, Jun 2017, Le Croisic, France. hal-01542911v2

HAL Id: hal-01542911
https://inria.hal.science/hal-01542911v2
Submitted on 23 Jun 2017
Nowadays, in embedded real-time systems, there is an evolution of needs in terms of computing power and communication between applications. This is due to the changing qualities of sensors, whose data processing needs are increasing (for example in radar applications), and the emergence of new application categories such as multi-sensors. Interaction between applications and sensors is increasing, while features tend to be spread over several computing units. Communication architectures are becoming more complex.

Mesh networking of the components of such systems would reconcile their real-time needs with the new application needs [1]. The plurality of communication paths should result in increased flexibility, resilience, scalability and load balancing characteristics. From the networking point of view, this implies the ability to dynamically allocate network resources with respect to the needs of running applications. We define this kind of platform as a Software-Defined Real-time Network (SDRN).

Link virtualization
- Each link is divided into sub-channels
- Sub-channels are allotted to flows
- Credit-Based Weighted Round Robin scheduling

Admission control and pathfinding
- From high-level requests: \( \text{req}_f = \{ T_f, S_f, d_{lj}, src_f, dest_f \} \)
- Take an optimization criterion into account (e.g. load balancing)

Fault tolerance: hybrid routing
- In nominal mode, routing is source-based
- In recovery mode, routing is hop-by-hop
- Sub-channels are reserved for flows in recovery mode

Ongoing work
- In-silicon implementation
- Using the SDN paradigm for node configuration


Second Contribution: ERICA, a Framework for Conducting Hybrid Experiments

ERICA helps building hybrid experimentation stacks in order to assess new network mechanisms [2]. Adaptation mechanisms allow to build a custom experimentation stack and to run the same application layer on top of different kind of networks (either real or simulated), thus maximizing relevance, portability and reusability.

Contact
Florian GREFF, PhD candidate
Thales Research & Technology
LORIA, University of Lorraine
florian.greff@loria.fr

References