802.1DD RAP – Reasons for Current Architecture

Martin Mittelberger (Siemens AG)

03/2025

v01

During the development of numerous 802.1Qdd drafts, the TSN TG discussed and designed the architecture which ended in 802.1Qdd d0.9 which was used as a basis for 802.1DD d1.0.

This contribution provides a short summary about the reasons why RAP choose a slightly different architecture while still using the main MSRP principles.

Requirements of IA devices

Industrial Automation and other devices require higher amounts of data for attributes to be synchronized through all or a part of network.

Industrial Automation devices require the support of proxied systems to operate network functionality on behalf of controlled systems.

802.1CS (LRP) was developed to fulfill these requirements (802.1CS CSD reads: “This project differs from existing and ongoing 802.1Q mechanisms because 802.1Q-2014 MRP, formerly 802.1Qak, does not fulfill scalability and performance requirements needed by some use cases.”)

Therefore, RAP was designed to use LRP instead of MRP what resulted in a different architecture of RAP systems compared to MSRP systems.

Different architecture

MSRP uses an architecture where applications become part of the exchange protocol MRP, thus MRP defines an application framework.

RAP on the other hand uses a layered architecture with applications on top of LRP.

Different data exchange protocols

LRP

a) exchanging data in data records

b) keeps consistency on a per data record level

- consistency ensured by periodic database checksum Exchange
- acknowledge of reception through checksum Exchange

c) fast protocol: immediate change notification on adding or removing a data record

d) no explicit data record limit (implicit by datatype of data record number)

e) support for virtualization (using LRP-DS-TCP Information Exchange can be redirected from physical device to any proxy device or vice versa)

f) applications are completely agnostic of the actual data exchange protocol. Right now, ECP and TCP are supported by LRP

g) hides consistency mechanisms from upper layer applications (in our case RAP)

h) support up to 1MB database per portal

i) LRP defines a per-system (Bridge / end station) service interface, only Portal is per port per application, ECP is per port, TCP is per system.

j) LRP handles only records and is unaware of semantics of application data

k) RAP participant is responsible for portal connection and attribute registration and declaration, while RAP Propagator handles attribute processing, propagation and resource allocation

l) no support for shared media (not required for resource reservation anyway)

MRP

a) Exchange data in MRP PDUs containing application attributes directly

b) consistency handled indirectly by periodic retransmission of all data

c) periodic retransmission is used by the applications as means of propagating/depropagating attributes (indirectly by timeout)

d) slow protocol, maximum number of PDUs per cycle may not be exceeded

e) amount of data limited by maximum number of PDUs

f) link local data Exchange only

Extended attribute propagation

The RAP Propagator is, by its logical function, not that different from what the MSRP (MAP) does.

Yet RAP requires some extensions which contradict the definitions about MAP as well as the MSRP Extensions to MAP.

RAP Propagator is not only responsible for attribute propagation but also attribute processing and bridge configurations as required for resouce allocation.

In MAP this is somehow true too yet it's not basic functionality of MAP but only of the MSRP MAP extensions, which on their own are not a real fit to RAP Propagator either (is going to be explained in 3).

RAP attributes contain more parameters to support various TSN shapers and per-hop latency calculation.

To be discussed

Maybe "re-"using some names (e.g. Participant, RAP "Propagator") of M(S)RP wasn't the best decision considering the number of differences.