RAP Extensions for the Hybrid Configuration Model

Lukas Osswald*, Steffen Lindner*, Lukas Wüsteney†, and Michael Menth*
*University of Tuebingen, Chair of Communication Networks, 72076 Tuebingen, Germany
†Hirschmann Automation and Control GmbH, CTO Office, 72654 Neckartenzlingen, Germany
Email: {lukas.osswald,steffen.lindner, menth}@uni-tuebingen.de, lukas.wuesteney@belden.com

***Abstract*—Modern applications in industrial automation rely on a deterministic network service, i.e., low latency, high reliability, and network convergence. Therefore, the IEEE 802.1 TSN Task Group introduces Time-Sensitive Networking (TSN). Besides mechanisms for traffic shaping, time synchronization, and reliability, TSN introduces three different configuration models for resource reservation: the fully distributed, the fully centralized, and the centralized network/distributed user model. Furthermore, IEEE P802.1Qdd specifies the Resource Allocation Protocol (RAP) to enable resource reservation for TSN streams in the fully distributed model. In this paper we give an introduction to RAP, and propose extensions to RAP for use in the hybrid configuration model. Additionally, we implement a prototype which is published under an open-source license.**

***Index Terms*—TSN, Resource Allocation Protocol (RAP), 802.1Qdd, centralized network/distributed user model, CUC**

I. INTRODUCTION

Many modern industrial applications, e.g., automation, require an ultra-low latency, deterministic network service. Typically, networks provide such a high quality of service (QoS) by reserving bandwidths for flows using resource reservation protocols. Audio/Video Bridging (AVB) is a standard for realtime communication in Ethernets. It is further developed under the name Time-Sensitive Networking (TSN) to meet even stricter time constraints. The IEEE 802.1 TSN Task Group (TG) defines concepts and protocols for resource management, time synchronization, bounded latency and delay variation, avoidance of congestion-based packet loss, as well as for reliability. The objectives are deterministic services for unidirectional unicast and multicast streams which carry realtime data. Thus, TSN supports the transport of multiple protocols for realtime applications over the same link, and facilitates integration of IT and Operational Technology (OT) networks.

AVB introduced the Stream Reservation Protocol (SRP) for admission control of streams. It is a hop-by-hop reservation protocol with local resource management carried out by every node, and supports a distributed configuration model. IEEE Std 802.1Qcc [7] extends SRP to support additional TSN features. Additionally, two new central entities for centralized network management are introduced. An application-specific Centralized User Configuration (CUC) receives flow requests including requirements from applications, communicates them in a uniform way to the Centralized Network Configuration (CNC) which is responsible for resource management and configures the switches to treat the flows with the requested QoS. In addition, IEEE Std 802.1Qcc introduces the centralized network/distributed user model, we further refer to as the hybrid configuration model. It leverages the CNC and end station convey stream requirements with a distributed resource reservation protocol to the network. The first bride, connected to an end stations, directly forwards the request to the CNC instead of performing distributed signalling. TSN offers new mechanisms to provide QoS. The Resource Allocation Protocol (RAP) is being defined by the TSN TG in IEEE P802.1Qdd to support these new mechanisms. RAP is a hop-by-hop protocol for dynamic resource reservation based on the Link-local Registration Protocol (LRP) for transport purposes.

The contribution of this paper is manifold. We survey RAP and LRP, and point out the improvements to SRP. As IEEE Std 802.1Qcc introduces centralized resource We analyse RAP's data model and propose extensions for RAP to be able to reserve scheduled streams in a centrally managed TSN network. Therefore, we describe a network based on the centralized network/distributed user model which includes a CNC, a new central component, called RAP-CUC, and leverages RAP for resource reservation of streams. Then, we provide a detailed architecture for a general CUC component, whose core is independent of a user-specific protocol and CNC implementation. Following that, we derive a specific CUC for RAP as a user-specific protocol.

II. RESOURCE RESERVATION AND QOS MECHANISMS

In this section we give an overview of resource reservation and quality of service (QoS) mechanisms in realtime networks, in particular in Audio/Video Bridging (AVB) and Time-Sensitive Networking (TSN) Ethernet. Further, we give an introduction to the resource reservation protocols of AVB.

A. *Resource Reservation in Realtime Networks*

Realtime streams have QoS requirements, e.g., bounded delay and delay variation, and minimum throughput. Bridges apply special mechanisms to guarantee the QoS for such streams despite of other traffic load. For instance, they limit the amount of realtime traffic to avoid overload in the network. This is performed per stream and/or per aggregate with the

This work has been supported by the German Federal Ministry of Education and Research (BMBF) under support code 16KIS1161 (Collaborative Project KITOS). The authors alone are responsible for the content of the paper.

©26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2021), Semptember 2021, Västerås, Sweden

help of shapers or policers. Another option is to assign dedicated transmission slots to frames, which is also known as scheduled traffic.

In both cases, admission control is performed. That is, realtime streams are explicitly admitted. Their properties, e.g., transmission rate and burst size, are considered for bookkeeping. Additional streams are admitted only if the remaining transmission capacity suffices. Otherwise, admission requests are declined to protect the QoS of already admitted streams. For this purpose, static and dynamic resource reservation exist which we review in the following.

1) Static Resource Reservation: With static resource reservation, the realtime streams and their properties are known prior to computation of configuration. Only a supportable amount of traffic is admitted and configured to obtain preferential treatment by the network. Other streams cannot demand this level of QoS at runtime.

2) Dynamic Resource Reservation: Dynamic resource reservation leverages network protocols to signal admission requests, configure QoS mechanisms in switches, and inform the requesting entity about the result. A resource reservation protocol conveys the properties of streams such that the network can take admission decisions. Such a resource reservation protocol can be classified as distributed or agent-based.

Distributed resource reservation leverages a resource reservation protocol which signals stream requirements along the path of the relevant stream. In case of success, it applies configurations to the bridges along the path of the admitted stream.

Agent-based resource reservation utilizes a centralized controller with a global view. End stations communicate stream properties and QoS requirements to the centralized controller which computes configuration data and applies them to bridges and hosts along the path of the stream.

In AVB networks, dynamic, distributed resource reservation is used. In TSN networks, both dynamic and static resource reservation are supported.

B. Resource Reservation and Traffic Shaping in AVB

In AVB [3], senders and receivers of a stream are denoted as Talkers and Listeners. Subsequently, we give a short introduction to resource reservation and traffic shaping in AVB.

1) Resource Reservation: The Stream Reservation Protocol (SRP) supports dynamic, distributed resource reservation in AVB networks. SRP leverages three protocols for resource reservation: the Multiple VLAN Registration Protocol (MVRP), the Multiple MAC Registration Protocol (MMRP), and the Multiple Stream Registration Protocol (MSRP). These three protocols use the Multiple Registration Protocol (MRP) as transport layer. MRP transmits the data provided by application protocols through the network where it is persistently stored in each hop [6].

VLANs are used to limit the scope of streams within the network. End stations may join a VLAN using MVRP. With MMRP a station subscribes to traffic from specific multicast or unicast MAC addresses. As a result, forwarding rules are configured in the briges along the path from the Talker to the Listener within the corresponding VLAN. MSRP is used to reserve resources for these streams.

IEEE Std 802.1Qcc extends the capability of SRP for TSN by introducing a central control elements for network management and users control. We discuss the configuration models of TSN in Section III.

2) Traffic Shaping: AVB introduces two traffic classes: Class A and Class B. A maximum delay of 2 ms and 50 ms is guaranteed for the traffic of the respective traffic classes over up to seven hops [3]. To achieve that, the traffic of both classes is policed with a Credit-Based Shaper (CBS) [2] which is a token bucket based algorithm to limit the burst size and bandwidth of traffic aggregates.

C. Selected QoS Functions in TSN

The TSN standards specify multiple mechanisms to guarantee QoS for realtime streams. Some of them are adopted from AVB, others are new [8], [18]. In the following, we describe two TSN-specifc mechanisms that are relevant in the context of this paper.

1) Time-Aware Shaper (TAS): To achieve ultra-low latency and delay variation for applications requiring hard realtime, TSN supports scheduled traffic. The Time-Aware Shaper (TAS) [4] leverages a TDMA paradigm. All bridges are synchronized in time and forward traffic according to a global schedule without queuing delay. If end stations are not synchronized, their traffic may be buffered at the access bridge. Optionally, Talkers which are synchronized to the network may be included in the schedule.

TAS is optimally applicable with static and dynamic, agent-based resource reservation. With agent-based resource reservation, time-aware Talkers communicate the earliest and latest possible transmit time of each stream to the network. After schedule synthesis, the network notifies the Talkers about the precise start of transmit in an interval for each stream.

2) Frame Replication and Elimination for Reliability (FRER): IEEE Std 802.1CB introduces Frame Replication and Elimination for Reliability (FRER) [5] to TSN which enables seamless redundancy over multiple paths for a stream. Frames are replicated at a bridge and sequence numbers are attached. The duplicated frames are sent along disjoint paths to another bridge which eliminates duplicate frames with the help of the sequence numbers. If one path fails, the traffic still reaches the destination over the working path.

III. CONFIGURATION MODELS FOR RESOURCE RESERVATION IN TSN

This section presents the configuration models for TSN specified in IEEE Std 802.1Qcc [7]. First, we give an overview of the User/Network Interface (UNI). Afterwards, we examine the configuration models, i.e., the fully centralized, the fully distributed, and the centralized network/distributed user model, further referred to as the hybrid configuration model.

A. User/Network Interface (UNI)

IEEE Std 802.1Qcc defines a User/Network Interface (UNI) leveraging YANG as a modeling language. The UNI is a bidirectional interface that is used to communicate QoS requirements and stream properties of end stations, and propagate the admission control status of a stream from the network to the end stations. For that, it consists of four YANG groupings that are further specified below.

1) 802.1Qcc Group-Talker: The group-talker is used to convey the Talker related stream properties, QoS requirements, and TSN capabilities for a stream to the network. It contains fields for specifying traffic characteristics of the stream, i.e., interval, maximum frame size and maximum amount of frames per interval. Time-aware Talkers additionally communicate the earliest and the latest possible transmission start time within an interval. Further, QoS properties like maximum latency and delay variation can be included as user-to-network requirements. A stream rank is included which is used by bridges to determine streams to be dropped in an oversubscription scenario. Additionally, the Talker discloses the TSN capabilities of its interface and specifies the characteristics of a stream's frame such that the network is able to associate it with its stream.

2) 802.1Qcc Group-Listener: The group-listener is intended as an admission control request for Listeners to indicate participation in a stream. It comprises QoS requirements of a Listener and its TSN capabilities.

3) 802.1Qcc Group-Status-Stream: The group-status-stream defines information about the admission control status of streams. The included information originates from the network. The grouping includes a status code for the Talkers and the Listeners each. In case of a failure, additional failure information specifies the cause and identifies the device by MAC address and interface name.

4) 802.1Qcc Group-Status-Talker-Listener: The group-status-talker-listener comprises status information and configuration data for one Talker or Listener. It originates from the network as a result of admission control procedure. It includes the worst-case latency a frame of a stream can experience along its path. Additionally, configuration data is provided to the end station, e.g., the point in time a time-aware Talker has to start transmission, or the VLAN ID and priority used for stream identification.

B. Fully Centralized Configuration Model

Figure 1 illustrates the fully centralized configuration model. The fully centralized model introduces a central network management controller, i.e., Centralized Network Configuration, and one or more controllers for user management, i.e., Centralized User Configuration.

(1) End stations signal admission control requests to a Centralized User Configuration (CUC) via a user-specific protocol. A commonly used user-specific protocol is OPC-UA client-server, a protocol developed by the OPC Foundation [1].

(2) The CUC collects stream requirements of all users participating in the same stream.

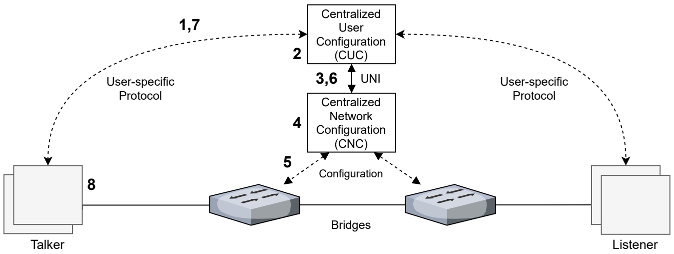


Figure 1: The fully centralized configuration model is composed of end stations, bridges, one or more CUCs, and a single CNC. The CNC-CUC interface is defined by the User/Network Interface (UNI)

(3) When all users provide sufficient information, the CUC initiates the resource reservation process via the UNI with the Centralized Network Configuration (CNC).

(4) A single CNC provides central network management. It takes admission control decisions based on the users' requirements obtained from all CUCs. The CNC computes configurations for bridges and end stations.

(5) It configures the bridges with network management protocols like SNMP [10], RESTCONF [9], or NETCONF [13].

(6) The CNC communicates the computed configuration for end stations and admission control status to the CUC.

(7) The CUC propagates both to the respective end stations.

(8) End stations reconfigure their interfaces accordingly and can start data transmission. Afterwards, the CNC and CUC can dynamically react on events, like node/link failures.

IEEE P802.1Qdj [16] further enhances the definition of the interface between CNC-CUC. Therefore, it will provide a fully functional YANG model for the centralized configuration model in TSN. This model will based on the four YANG groupings of the current UNI and may add extensions. As a consequence, the implementation of a CNC-CUC interface can be based on a YANG based protocol, e.g. RESTCONF or NETCONF.

The fully centralized model introduces network management with a global view to TSN networks. This enables computation of globally optimized schedules for TAS, eliminating queuing delay [12]. This makes the fully centralized model suitable for environments that require precise timing of packets and complex planning.

C. Fully Distributed Configuration Model

Figure 2 illustrates the fully distributed configuration model. The fully distributed configuration model performs admission control decisions with a hop-by-hop resource reservation protocol. As a consequence, the UNI is located between all participating devices. SRP may be used for that purpose. However, the Resource Allocation Protocol (RAP) is currently defined to provide support for novel TSN mechanisms. End stations signal stream requirements to the network. The bridges take admission control decisions based on local information

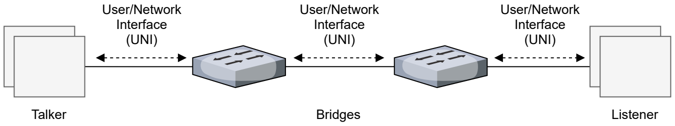


Figure 2: The fully distributed configuration model is composed of end stations and bridges and was originally defined for AVB networks.

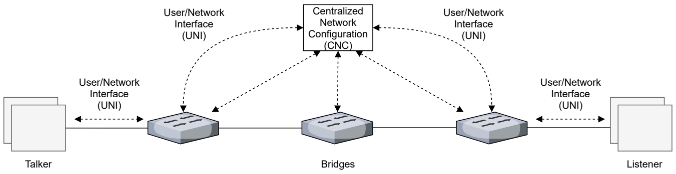


Figure 3: The centralized network/distributed user model is composed of end stations, bridges and a single CNC.

only. Therefore, computation of a globally optimized schedule for TAS is not optimally possible.

D. Hybrid Configuration Model

Figure 3 illustrates the hybrid configuration model. End stations communicate admission control requests via a distributed resource reservation protocol. The edge bridges ensure that admission control requests are directly forwarded to the CNC, and not hop-by-hop as in the fully distributed model. The CNC then takes admission control decisions and computes configurations as in the fully centralized model. As a consequence, the UNI is located between end stations, the edge bridge, and the CNC.

The hybrid model avoids the use of multiple, application specific CUCs by relying on a single protocol for resource reservation. This protocol can provide admission control as a service to multiple applications. This can reduce implementation complexity of end stations and the network management components. Still the same level of QoS as in the fully centralized model can be achieved.

Currently, SRP is a candidate for resource reservation in the hybrid configuration model but does not support all TSN features. RAP mitigates this issue, but does not yet address the hybrid configuration model in its current draft. A CNC for the hybrid model must provide a RAP-capable interface. As IEEE P802.1Qdj emerges, future CNCs will provide a RESTCONF/NETCONF based interface. Therefore, we propose an approach to implement the hybrid configuration model using an extended RAP, a novel RAP-CUC, and a RESTCONF based CNC in Section V.

IV. OVERVIEW OF LRP AND RAP FOR TSN NETWORKS

We first give an overview of the Link-Local Registration Protocol (LRP) [14], as well as, its proxy models, and introduce the current state of RAP [11].

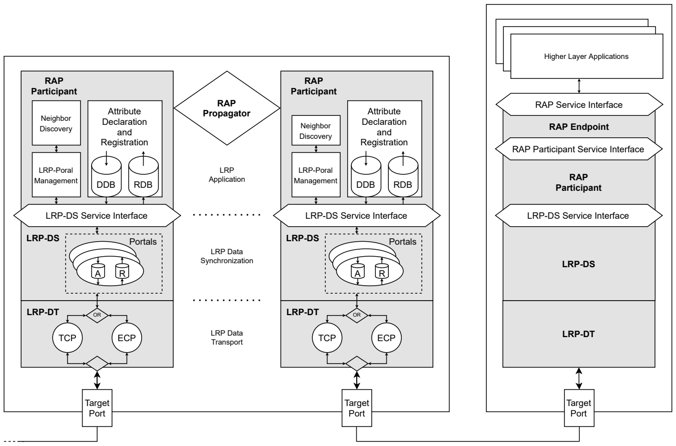


Figure 4: On the left side is a two port, RAP-capable bridge and on the right side is a RAP-capable end station.

A. Link-Local Registration Protocol (LRP)

LRP has been specified in IEEE P802.1CS [14] as transport protocol for TSN. We explain its architecture and signalling, system types and proxy models.

1) Architecture and Signalling: LRP is intended to transport LRP Data Units (LRPDU) hop-by-hop and to store this data persistently. LRP is similar to MRP but addresses scalability issues and adds new proxy mechanisms. LRP can be efficiently used to distribute databases of up to 1MB between communication peers, resolving scalability issues of MRP which was optimized for databases of up to 1500 Bytes [14].

Figure 4 illustrates a LRP bridge and end station with RAP as LRP application. The LRP protocol stack consists of three layers: LRP application, LRP Database Synchronization (LRP-DS), and LRP Database Transport (LRP-DT).

A LRP application implements application specific behavior. It can take forwarding decisions and configure bridge hardware based on received data. One or more LRP applications leverage the data synchronization service provided by LRP-DS.

LRP-DS establishes connections and controls data synchronization with LRP Portal instances. LRP Portals contain two databases for transmission and receipt of data: the Applicant Database and Registrar Database. The records of the Applicant Database of one system are copied to the Registrar Database of the peer. The connection managed by a Portal is bound to a single LRP application and to a physical port of a device, called target port. The target port is uniquely characterized by the chassis identifier and port identifier of a system. It is not required to reside on the same host as the Portal itself. For Portal creation, LRP-DS supports several methods, i.e., manual or protocol assisted configuration. Link Layer Discovery Protocol (LLDP) can be used for protocol assisted Portal creation. Therefore, the target port associated with a Portal advertises a list of applications and corresponding address information. Based on that, the two LRP devices establish a connection between their applications. As an alternative LRP introduces a special handshake for connection establishment, initiated by sending an Exploratory Hello LRP Data Unit (LRPDU).

LRP-DT transmits and receives LRPDU via Transmission Control Protocol (TCP) [15] or the Edge Control Protocol (ECP) [6]. ECP is a simple transport mechanism implementing flow control for the local link using a stop-and-wait automatic repeat request paradigm [17].

2) System Types: IEEE P802.1CS defines three system types.

i) A LRP system is native, when application, Portal and target port are physically located in the system. Native systems provide computational power for the applications, data storage capabilities for the Portals and a local target port.

ii) A proxy system implements an application and Portal instance. Additionally, it leverages the remote target port of a controlled system, e.g., for Portal creation via LLDP. The proxy systems can reside at a remote location like the edge of the network or in the cloud.

iii) A controlled system only provides a physical port as a remote target port for a proxy system, and does not have to implement LRP itself. A proxy system and a controlled system must be used in combination to form a functional LRP system.

Such a composite system only supports TCP for LRP-DT, and LLDP or manual configuration for Portal creation. The proxy system provides a list of application services and address information to the controlled system via network management. The controlled system advertises the provided information on its local target port via LLDP to its peer. With the advertised address and application information a peer-to-peer, TCP connection is established. This connection can be used to exchange application data between the peer's and the proxy system's application. The controlled system is not necessarily involved in data transport after the connection is established.

3) Proxy Models: IEEE P802.1CS proposes four proxy models using proxy/controlled systems for either relay systems, end systems, or both.

i) The "full native system" model, consists of a native bridge and a native end system. Connections can be established manually, with LLDP, and Exploratory Hello LRPDUs. LRP-DT can be based on ECP or TCP in that scenario. A use case for the "full native system model" is implementing the UNI for the fully distributed configuration model with RAP as LRP application.

ii) The "proxied relay systems" model is illustrated in Figure 5. It is composed of controlled relay systems providing target ports for a proxy system. This proxy model can be leveraged to implement the hybrid configuration model of TSN where the proxy system is part of the CNC.

iii) The "proxied end systems" model includes native relay systems, controlled end systems and an end systems' proxy. A use case can be incorporating legacy or simple end systems into a TSN network. The end systems' proxy handles, e.g., admission control on behalf of the end systems with the network.

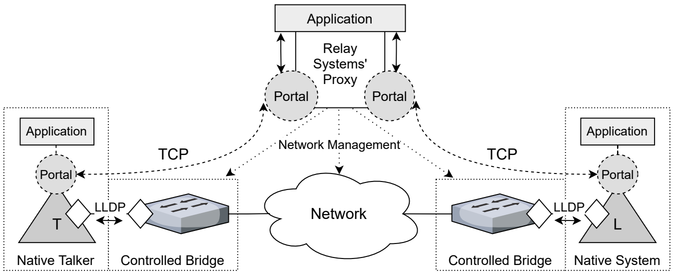


Figure 5: Hybrid configuration model implemented with LRP: the proxy system's application and address information is announced by the target port of the controlled bridges via LLDP. End stations discover the application of the proxy and establish a connection. The resulting TCP connection is used for data exchange between the LRP applications of the end stations and the proxy system's application.

iv) The "end systems' proxy and relay systems' proxy" model combines the two preceding proxy models. A use case for the model is the fully centralized configuration model of TSN. The end systems' proxy is the CUC and the relay systems' proxy is the CNC. The proxies exchange application data directly via TCP. The connection is set up by the controlled end stations and bridges physically connected. Both advertise address information of their proxy systems via LLDP to create Portals.

B. Resource Allocation Protocol (RAP)

RAP is a protocol for dynamic resource reservation for unicast and multicast streams. It is specified in IEEE P802.1Qdd Draft 0.4 [11] as a successor for SRP. RAP advances SRP to provide support for the recent evolution in TSN standardization, e.g., FRER. We illustrate a RAP-capable bridge and end station in Figure 4.

1) Domain Establishment with RAP: For resource reservation, all devices along the path of a stream must be members of the same RAP domain. A RAP domain comprises the set of neighbouring RAP devices which support a priority for a traffic class. This priority characterizes one of eight Resource Allocation classes (RA classes), along with a RA Class Template (RCT). The RCT describes a set of TSN mechanisms to be applied to streams of the class.

For domain establishment, each RAP capable device announces its RA classes link locally to its neighbours. Devices identify domain boundaries based on the priority values of the RA classes. The RCT is not evaluated for domain establishment, thus the use of different traffic shaping mechanisms along a path is possible.

2) RAP Attributes: For resource reservation RAP end stations exchange structured data, so called attributes. RAP defines three attributes, encoded as a Type-Length-Value (TLV).

i) As discussed in the previous section, all RAP end stations declare RA class attributes for domain establishment link locally.

ii) The Talker Announce Attribute (TAA) is sent by Talkers to its Listeners for conveying stream identification information and traffic specification. For traffic specification, RAP offers a TLV for token bucket based shaping, including minimum/maximum frame size, committed information rate, and committed burst size. An alternative is the MSRP traffic specification, including maximum number of frame size and maximum number of frames per interval. Additionally, the network uses the TAA to compute the wort-case latency of a path and give status information to the Listener.

iii) The Listener Attach Attribute (LAA) is declared by a Listener for communicating the interest in participating in a stream to the network. The network uses it to convey admission control status to the Talker. The status carried by LAAs can be: Listener Ready, Failed, or Partial Failed.

All attributes can be extended with organizationally specific TLVs for adding custom features to RAP.

3) Resource Reservation Process: A resource reservation process with RAP is initiated by a control application of an end station which requires QoS guarantees for its application data streams. As a result, Talkers declare TAAs and Listeners declare LAAs to request resources for streams from the network.

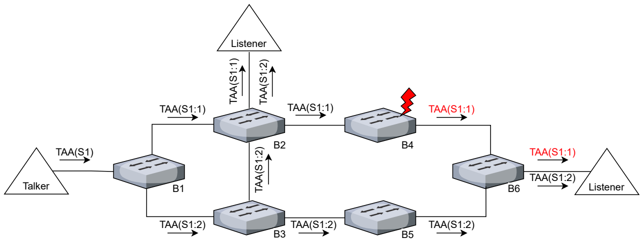
Figure 6(a) illustrates the signaling of Talker for reserving resources for a multicast stream via two disjoint paths. The Talker declares a TAA which is propagated by the bridges in direction of all Listeners along the path of the stream. On receipt of a TAA, bridges evaluate if sufficient resources for applying the requested QoS level are available. When resources are available, the attribute is forwarded in direction of the Listeners. In case of an error, failure information is attached to the TAA before forwarding. For computing worst-case path latency, bridges add their maximum forwarding latency to the accumulated latency field in the TAA.

Figure 6(b) illustrates the signaling of the Listeners. They declare a LAA to signal participation in a stream each. After a TAA and LAA of the same stream is registered on the ports of a bridge, resources for the stream are finally reserved and underlying QoS mechanisms are configured. When multiple LAAs for the same stream are received by a bridge, it merges the status information of all received LAAs before forwarding in direction to the Talker.

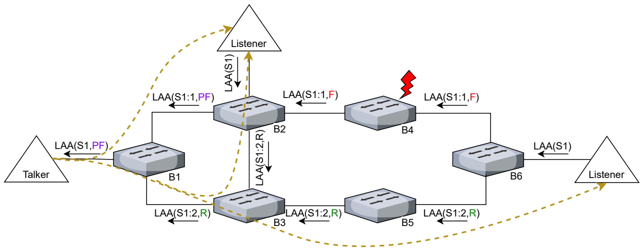
After a successful resource reservation, RAP notifies the control application which initiated resource reservation such that it can start deterministic data transmission.

V. Analysis And Extension of RAP for Hybrid Configuration Model

In this section we develop a concept for using RAP for admission control in the hybrid configuration model. First, we explain the problem statement for our scenario. We conclude that a RAP-CUC is needed and propose a general architecture for a CUC. We show the compatibility of RAP to the TSN UNI. Finally, we describe the design and implementation of a RAP-CUC.



(a) A Talker declares a TAA to signal stream requirements to the network. Based on that bridges pre-reserve resources. Bridge B4 is overbooked, therefore attaches failure information to the TAA to notify Listeners.



(b) Listeners declare a LAA to participate in a stream. Bridges merge status information of multiple LAAs for the same stream and forward one LAA in direction of the Talker. Additionally, resources are reserved and QoS functions are configured. The dashed arrow shows the path of the partially reserved realtime stream.

Figure 6: RAP signalling for reserving resources for an 1+1 protected, multicast stream.

A. Problem Statement

RAP standardization currently focuses on the fully distributed configuration model and mostly token bucket based traffic shaping. The hybrid configuration model requires a distributed signalling protocol, like RAP, and a CNC. Our goal is to enable resource reservation for globally scheduled realtime streams using RAP. We want to enable resource reservation for time-aware end stations in a TSN network following the hybrid configuration model. In our scenario the CNC provides a RESTCONF-based CUC-CNC interface.

In hybrid configuration model end stations' RAP requests are directly forwarded to the CNC by edge bridges. Since future CNCs will have a YANG based interface, we introduce an additional component, as a gateway between end stations and the CNC, to transform RAP-based requests to a format understandable by the CNC. Additionally, this component manages state of ongoing resource reservations. We state that such a component is equal to the CUC component known from fully centralized configuration model.

B. Life-Cycle of a TSN Stream

We assume that the life-cycle of a stream is managed by the CUC which keeps track of the state of streams and reacts to different events originating from CNC or end stations. We define the following states: *new*, *pending*, *deployed*, *withdrawn*, *error*.

The initial state of a stream is *new*, when either the requirements of a Talker or Listener is registered but not both.

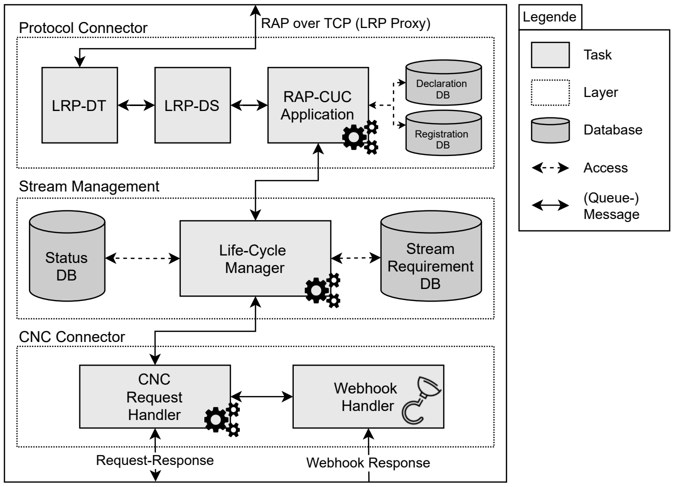


Figure 7: The proposed CUC consists of a three layer architecture: Protocol Connector, Stream Management, CNC Connector. The Protocol Connector is specific for RAP.

The stream reaches the *pending* state, when Listener and Talker requirements for a stream are received by the CUC. As a reaction, the CUC initiates a resource reservation process with the CNC.

When the CNC indicates a successful resource reservation, the state advances to *deployed*.

Otherwise, the *error* state is reached, e.g., due to insufficient resources. Additionally, the CNC can convey an error state to the CUC, even after the stream has been successfully deployed, e.g., on recognition of a link failure.

The state is *withdrawn* when either the Talker or all Listeners cancel their intent to participate in the stream. As a consequence, the CUC withdraws reserved resource from the CNC to free network resources.

C. Design of a General CUC Architecture

The CUC fulfills three tasks: communicate with end stations, manage the life-cycle of streams, and request resources for streams from the CNC. With regard to these tasks, we propose a three layer architecture consisting of a Protocol Connector, Stream Management, and CNC Connector. An example for RAP as user-specific protocol is depicted in Figure 7.

1) Protocol Connector Layer: The Protocol Connector communicates with end stations over a user-specific protocol. Thus, the implementation depends on the user-specific protocol and its signalling. The Protocol Connector extracts data relevant for admission control from user-specific protocol packets, transforms this data to group-talker or group-listener structures, and hands it to the Stream Management Layer (SML). In addition, it generates messages, as the user-specific protocol defines, to notify end stations about stream status changes and sends configuration data on behalf of the SML. We describe an implementation of a Protocol Connector based on RAP in Section V-E.

2) Stream Management Layer: The SML manages the whole life-cycle of a stream as described in Section V-B. Therefore, it stores stream requirements of end stations from Protocol Connector, manages stream reservation and withdrawal with the CNC Connector, and triggers notification of end stations. For that, it keeps track of end stations requirements in the Stream Requirement Database (SRDB) and stream reservation status in the Status Database (SDB). The SRDB stores the stream requirements per end station, either as IEEE Std 802.1Qcc group-talker or group-listener. The SDB contains a record for each stream currently under management. Each record contains information about the state of reservation process, a list of participating end stations, and the group-status-stream and group-status-talker-listener returned from CNC Connector on successful reservation.

SML is independent of the user-specific protocol and the CNC's implementation. Thus, the SML implements a general model for life-cycle management of streams in TSN.

3) CNC Connector Layer: The CNC Connector conveys stream requirements to the CNC and invokes remote procedure calls. It uses the groupings of IEEE Std 802.1Qcc, communicated by the SML, to initiate stream reservation/withdrawal process. For that, it communicates stream requirements, initiates computation and deployment of network configuration. The signalling performed by the CNC Connector is specific to the implementation of the CNC's interface.

Since schedule synthesis is a complex problem, it can take some time for larger networks [12]. A simple blocking, request-response paradigm or polling for the result is using resources inefficiently.

Therefore, we introduce the Webhook Handler for subscription-based result propagation. The Webhook Handler implements a REST-based application interface for providing callback addresses to the CNC. For each request, the CNC Connector obtains an Uniform Resource Identifier (URI) from the Webhook Handler. This URI will be sent along the request whose response is to be subscribed. The CNC transmits the computational result to the specified URI. Thus, a high number of open connections and blocking can be avoided.

D. Compatibility of RAP with TSN UNI

We analysed if RAP can be used for reserving resources for scheduled streams originating from time-aware Talkers. Therefore, we compared the minimum set of information, defined by the UNI, needed to take admission control decisions, with the attributes and signalling of RAP. Analysis has shown that RAP currently does not provide sufficient information to the CNC to take admission control decisions for the scheduled streams with time-aware end stations.

RAP lacks the possibility to communicate the interval length and the earliest/latest transmit time to the CNC. Additionally, choosing a custom latency bound of a stream for schedule synthesis is not possible using RAP. Furthermore, the resulting end station configuration, like the transmission start time of a Talker, can not be communicated to the end stations.

We propose to consider including the interval time, the earliest/latest transmit time, and the maximum latency as optional fields in the MSRP traffic specification. We also propose to allow attachment of interface configuration to the LAA and TAA attributes. These gaps can be immediately resolved by integrating the proposed extensions as organizationally defined TLVs in LAA/TAA.

E. Design of a RAP-specific Protocol Connector

We introduce a Protocol Connector which performs RAP specific signalling with the end stations. For connecting the end stations to the RAP-CUC, we leverage the "proxied relay systems" model of LRP as illustrated in Figure 5.

In that scenario the Protocol Connector for RAP is the LRP proxy system and the end stations are native LRP systems. The target port of the controlled edge bridge announces RAP as an application and corresponding address information to the end stations via LLDP. As an alternative, Portals can be manually configured. The advertised information is used by the applications to establish a TCP connection. This connection from the end stations to the Protocol Connector of the RAP-CUC enables exchange of RAP attributes for resource reservation.

The Protocol Connector includes an implementation of LRP-DT, LRP-DS and a newly introduced RAP-CUC application component. The RAP-CUC application extracts information relevant for admission control from the received TAA and LAA attributes. The information is passed to the SML for managing state of stream reservation. Additionally, withdrawal of attributes by end stations is conveyed, too. On behalf of SML, the RAP-CUC application updates the fields of the registered RAP attributes and controls forwarding for conveying admission control results to the end stations.

VI. IMPLEMENTATION OF A RAP-SPECIFIC PROTOCOL CONNECTOR

We release an implementation of the described RAP-CUC [19]. The prototype follows the proposed general CUC architecture and consists of: the RAP specific Protocol Connector, SML, and a partial implementation of the CNC Connector.

The Protocol Connector uses manual configuration for LRP Portal creation. Since Portal creation results in a TCP connection from the end stations to the RAP-CUC, we omit implementation of LRP-DS and LRP-DT and directly establish the aforementioned TCP connection. This approach is sufficient for evaluating RAP as a resource reservation protocol for hybrid configuration model.

We successfully tested our implementation against a proprietary CNC which provides a RESTCONF-based interface. We can not disclose implementation detail about the interface of the CNC, therefore only a rudimentary CNC Connector is provided.

The published CNC Connector can be used as a basis for developing an other CNC Connectors for a specific CNC. Furthermore, the RAP specific Protocol Connector can be replaced by a custom Protocol Connector for connecting end stations via other user-specific protocols.

VII. CONCLUSION

We gave an introduction to LRP and RAP as a future resource reservation protocol of TSN. We found out RAP currently lacks the ability to reserve resources for time-aware scheduled streams in the hybrid configuration model. To resolve the issues, we proposed an extension to RAP and an approach for immediate remedy. We describe a general architecture for CUC components and a specific implementation of a RAP-CUC. The implementation of the RAP-CUC is published under an open-source license [19]. Future work comprises further investigation of use-cases for LRP proxy models.

REFERENCES

[1] Home page - opc foundation. https://opcfoundation.org/. (Accessed on 04/14/2021).

[2] IEEE Standard for Local and Metropolitan Area Network–Virtual Bridged Local Area Networks – Amendment 12: Forwarding and Queueing Enhancements for Time-Sensitive Streams. *IEEE Std 802.1Qav-2009*, 2009.

[3] IEEE Standard for Local and Metropolitan Area Network–Audio Video Bridging (AVB) Systems. *IEEE Std 802.1BA-2011*, 2011.

[4] IEEE Standard for Local and Metropolitan Area Network–Bridges and Bridged Networks – Amendment 25: Enhancements for Scheduled Traffic. *IEEE Std 802.1Qbv-2015*, 2015.

[5] IEEE Standard for Local and Metropolitan Area Network–Frame Replication and Elimination for Reliability. *IEEE Std 802.1CB-2017*, 2017.

[6] IEEE Standard for Local and Metropolitan Area Network–Bridges and Bridged Networks. *IEEE Std 802.1Q-2018*, 2018.

[7] IEEE Standard for Local and Metropolitan Area Network–Bridges and Bridged Networks – Amendment: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements. *IEEE Std 802.1Qcc-2018*, 2018.

[8] L. Lo Bello and W. Steiner. A Perspective on IEEE Time-Sensitive Networking for Industrial Communication and Automation Systems. *Proceedings of the IEEE*, 2019.

[9] A. Bierman, M. Bjorklund, and K. Watsen. RFC8040: RESTCONF Protocol, 2017.

[10] J. D. Case, M. Fedor, M. L. Schoffstall, and J. Davin. RFC1157: Simple Network Management Protocol (SNMP), 1990.

[11] Feng Chen. IEEE Draft Standard for Local and Metropolitan Area Network–Bridges and Bridged Networks – Amendment: Resource Allocation Protocol. *IEEE Std P802.1Qdd Draft 0.4*, 2020.

[12] Frank Dürr and Naresh Ganesh Nayak. No-wait packet scheduling for IEEE time-sensitive networks (TSN). In *Proceedings of the 24th International Conference on Real-Time Networks and Systems - RTNS '16*. ACM Press, 2016.

[13] R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman. RFC6241: Network Configuration Protocol (NETCONF), 2017.

[14] Norman Finn. IEEE Draft Standard for Local and Metropolitan Area Network–Link-local Registration Protocol. *IEEE Std P802.1CS Draft 3.1*, 2020.

[15] Information Sciences Institute. RFC793: Transmission Control Protocol, September 1981.

[16] Stephan Kehrer. IEEE Draft Standard for Local and Metropolitan Area Network–Bridges and Bridged Networks – Amendment: Configuration Enhancements for Time-Sensitive Networking. *IEEE Std P802.1Qdj*.

[17] James F. Kurose and Keith W. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Pearson, 6 edition, 2012.

[18] J. L. Messenger. Time-Sensitive Networking: An Introduction. *IEEE Communications Standards Magazine*, 2018.

[19] Lukas Osswald. uni-tue-kn/rap-cuc. https://github.com/uni-tue-kn/rap-cuc. (Accessed on 05/05/2021).