SCTE•ISBE CABLE-TEC EXPO® VIRTUAL EXPERIENCE » OCTOBER 12-15 2020

2020 Fall Technical Forum SCTE·ISBE • NCTA • CABLELABS®

Building the RPHY Upstream Scheduler with YANG

A Technical Paper prepared for SCTE•ISBE by

Tong Liu, PhD
Principal Engineer
Cisco Systems Inc
300 Beaver Brook Road, BOXBOROUGH, MA 01719
978-936-1217
tonliu@cisco.com

John T Chapman
CTO Cable Access and Cisco Fellow
Cisco Systems Inc
170 W Tasman Dr, San Joe, CA 92677
408-526-7651
jchapman@cisco.com

Table of Contents

List of Figures

List of Tables

1. Introduction

The remote upstream (US) scheduler is a low latency Remote PHY solution proposed in [1] to maintain low latency for DOCSIS regardless of the CIN distance. Moving the US scheduler from the CCAP core to the remote PHY device (RPD) creates a new management interface between the remote US scheduler at the RPD and the upper MAC layer clients at the CCAP core. This entails a robust and standard application programming interface (API) for easy configuration / management of multiple remote US schedulers at scale and enabling interoperability between different CCAP core and RPD venders.

Towards this goal, we propose a YANG based API to manage the remote US scheduler. YANG is an API contract language widely used in the world of networking and is now being introduced into the world of DOCSIS. A specification written in YANG is referred as a "YANG module", and a set of YANG modules are collectively called a "YANG model". A YANG model characterizes the behavior of a network function with data hosted by the server that a client can manipulate and observe using standardized operations. Once the YANG model is published, both client and server can have faith that the other knows the syntax and semantics behind the modeled data.

In this paper, we describe how we used YANG to define the remote US scheduler service contract. Our YANG model has several modules. The first is the US scheduler itself that provides the scheduling services such as best effort, rtPS, nrtPS, UGS and PGS. The second component is the MAP Builder that that allocates bandwidth across the US RF channels. The remote upstream YANG model is intended to be included in the CableLabs' standard modules and become an integral part of the standard DOCSIS YANG ecosystem.

The rest of the paper is organized as follows. Section 2 explains why the remote US scheduler may help maintain low-latency for DOCSIS in R-PHY deployments. Section 3 shows how to separate the real-time scheduling functions from the CCAP core while keeping DOCSIS control plane intact. Section 4 examines the reasoning for using YANG data model-driven management for the remote US scheduler. Section 5 explains the fundamental modeling principles to establish the remote US scheduler YANG model. Section 6 presents the remote US scheduler YANG model structure and the key components. Section 7 discusses how to add the YANG based remote US scheduler to existing RPDs that are managed by the legacy Generic Control Protocol (GCP). Section 8 describes how to integrate the remote US scheduler into the upcoming DOCSIS YANG ecosystem. Finally, section 9 concludes the paper and highlights the takeaways.

2. Why Remote US Scheduler

In the R-PHY architecture today, the US scheduler is centralized at the CCAP core, leaving only the PHY elements at the RPD. This design choice fits most of the R-PHY deployment cases today where the CIN distance is less than 100 miles and the MAPs run at 2-millisecond interval. Under these operation conditions, CIN delay has no impact on the upstream scheduling latency based on the analysis in [1], and R-PHY system is equivalent to the I-CCAP (Integrated Converged Cable Access Platform) in terms of the latency performance, but with much better RF capacity.

As the distributed access architecture (DAA) transformation deepens, there are scenarios where the CIN is stretched beyond the 100-mile mark, for reasons such as hub-side consolidation that relocates a CCAP core to the central headend or a regional data center. Meanwhile, driven by the new low latency applications, such as cloud gaming and mobile xHaul, the DOCSIS upstream request-grant (REQ-GNT) protocol is tightened to a shorter MAP interval at 1 millisecond [2]. Under these new operation conditions, the CIN becomes contributing factor to the upstream scheduling latency. By relocating the

upstream scheduler to the RPD, the CIN delay factor can be effectively remoted from the REQ-GNT loop, therefore achieving latency improvement.

Figure 1 shows the centralized vs. remote US scheduler deployment scenarios and the traverse paths of the REQ-GNT(MAP) messages.

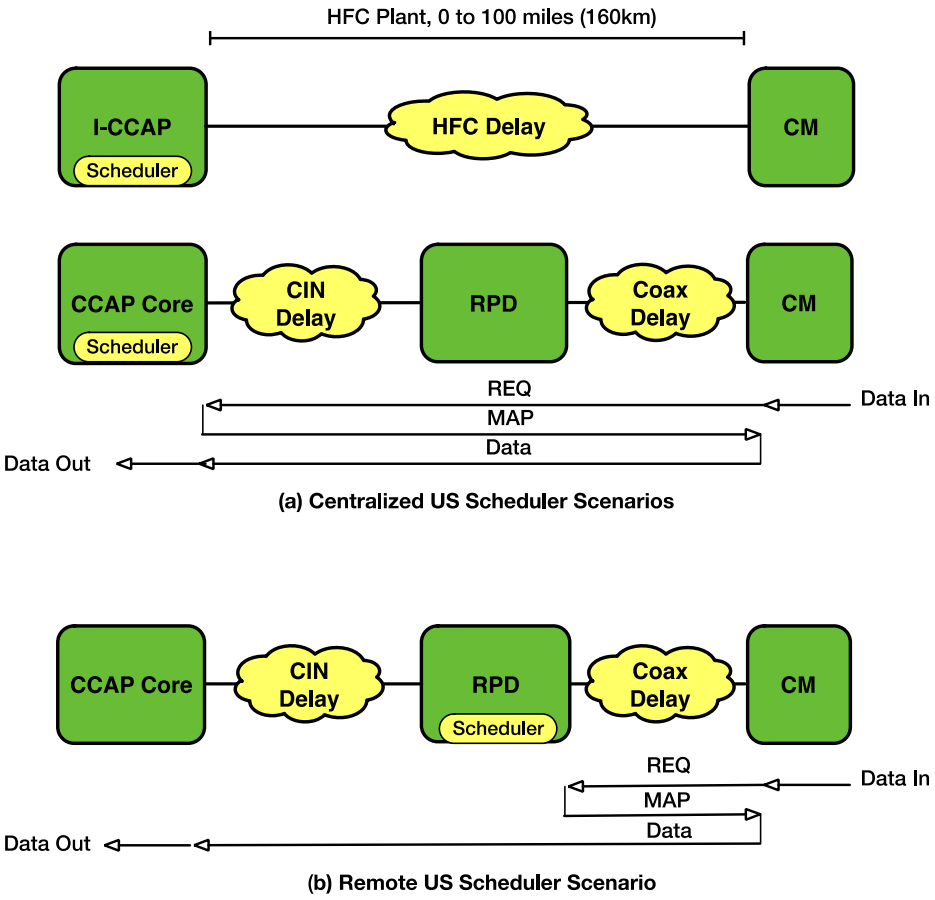

Figure 1 – Centralized vs. Remote Upstream Scehduler Deployment Scenarios

It should be noted that a centralized scheduler for Remote PHY will provide equivalent performance to a classic I-CCAP system as they both have the US scheduler in the same location. That location is the hub site that is 100 miles or less from the CM. The request and MAP messages are given high priority on the CIN so that any CIN queuing and latency will not add to the request grant delay.

Thus, the remote US scheduler is an operational option for when performance is desired that is better than a classic I-CCAP or if the CIN needs to be significantly longer than the 100 miles.

3. How to Spin off the Remote US Scheduler

In the R-PHY architecture today, the US scheduler is an internal component located at the CCAP Core, as shown in Figure 2.

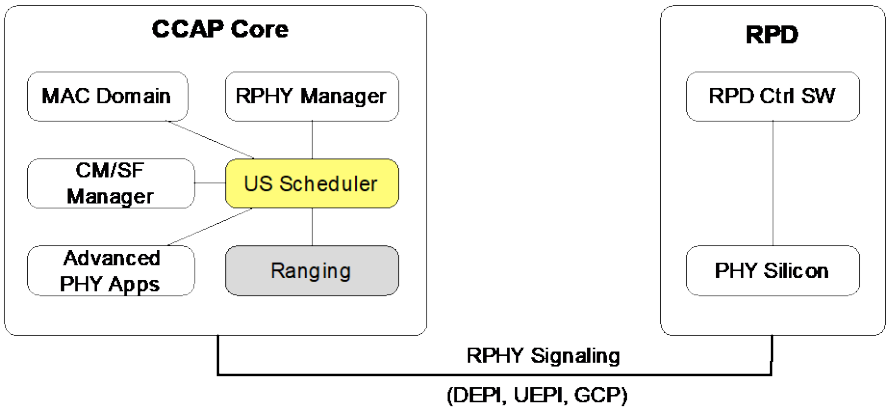

Figure 2 – Centralized Upstream Scheduler in R-PHY Today

The US Scheduler interacts with various clients in the DOCSIS control plane using vendor specific interfaces, as listed in Table 1.

Table 1 – US Scheduler Clients in DOCSIS Control Plane

Clients	US Scheduler Northbound Interface
MAC Domain	Notifies per MAC Domain upstream channel configurations including the corresponding primary capable downstream channels for carrying the MAP messages.
Cable Modem (CM) Manager	Notifies the CM operation state.
Service Flow (SF) Manager	Notifies the SF QoS Parameters, SF SID / SID cluster assignment.
Ranging	Requests for ranging transmission opportunities in MAP, including initial ranging, maintenance ranging and probing in OFDMA.
Advanced PHY Applications	Requests for transmission opportunities for advanced PHY features, such as the data transmission opportunity for OFDMA US data profile (OUDP) test, or a silent transmission opportunity to capture per channel noise floor.
Remote PHY Manager	Notifies the DEPI and UEPI session / pseudo wires to transport the scheduler signaling messages, such as MAP and bandwidth request pseudo wires.

Figure 3 shows a remote US scheduler architecture that separates out the real-time scheduling functions while keeping the northbound client interface intact. This is achieved by partitioning the US scheduler into a control component (US scheduler manager) in the CCAP core and a real-time component (US scheduler engine) in the RPD. The US scheduler manager interfaces the northbound control plane clients and manages the southbound US scheduler engines. The US scheduler engine fulfils the real-time scheduling services at the RPD.

In between the US scheduler manager and the US scheduler engine, we propose that a YANG data model-driven interface is used to formally describe the managed data source at the remote US scheduler engine. The US scheduler manager may simultaneously manage multiple US scheduler engines. To meet the timing and scaling requirements, a stream-lined YANG model-driven network protocol, such as gNMI, may be used to transport the API calls. gNMI stands for the gRPC Network Management Interface, and gRPC is known as Google remote procedure call.

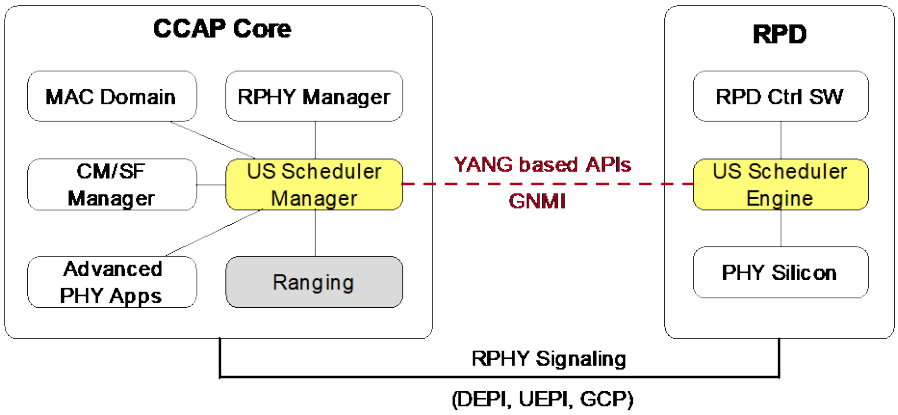

Figure 3 – Remote Upstream Scehduler Architecture

4. Why YANG Model-Driven API for the Remote US Schduler

4.1. A Formal Defination of Service Contract

YANG is a full, formal contract language with rich syntax and semantics. A YANG model is the specification written in YANG about the managed data objects that a client can manipulate and observe using standardized operations. The YANG model-driven management is about building the YANG data and using the model via tooling to generate code to access the YANG objects on the managed devices.

As displayed in Figure 4, which covers the data model-driven management components, once the YANG models are specified and implemented, a particular encoding (XML, JSON, protobuf etc.) can be selected along with a particular remote procedure call (RPC) protocol (NETCONF, RESTCONF, or gNMI/gRPC) for transport. Based on the encoding and protocol selections, code or API can then be generated via proper applications / tooling (for example, in Python, C++, Go or any other language) [3].

To enable remote US scheduling, a formal service contract is needed between the US scheduler manager and the US scheduler engine. The YANG model-driven API is a natural choice to achieve this.

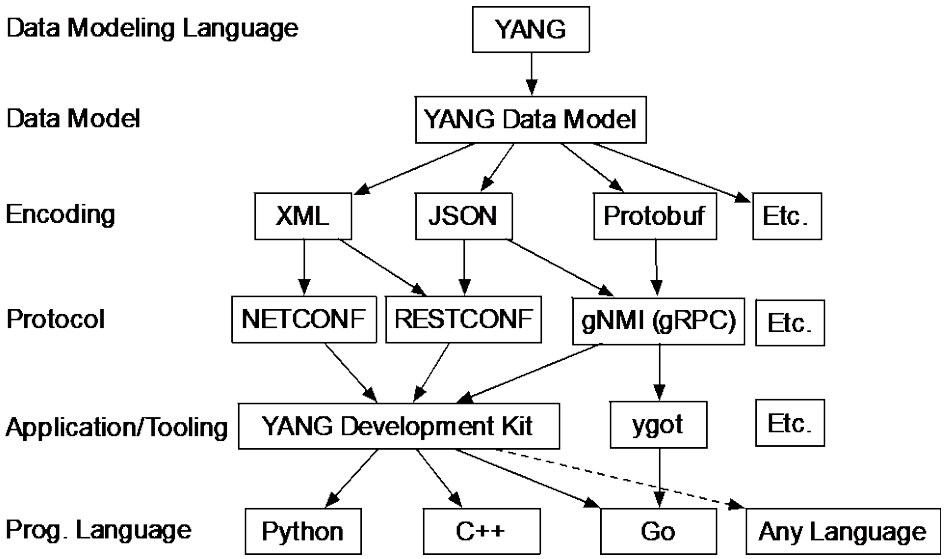

Figure 4 – YANG Data Model-Driven Management Components

4.2. Mature and Well Adoped

YANG was originally created at IETF in 2007 to describe standard data models for network automation via NETCONF. Since then, YANG data model-driven management has become a well-established trend in the network industry. YANG data models are produced by many parties including standard organizations, consortiums, forums and open source projects (such as OpenConfig and Broadband Forum), and network equipment vendors.

YANG data modeling has also been adopted in cable. There are several efforts involving YANG data modeling driven by CableLabs, including the ICCAP YANG modeling, as part of CCAP (Data-Over-Cable Service Interface Specifications) OSSI (Operations Support System Interface Specification) effort, the Flexible MAC Architecture (FMA) YANG modeling, and a recent initiative to build a common YANG ecosystem across different network elements (NEs), including CCAP Core, RPD, RMD and ICCAP.

4.3. Interoperation Made Easy

YANG solves the multi-vendor interoperability issue by using the data model as the core definition of the configuration and operational data and renders them to the management interface in a protocol specific construct. This approach is different from the traditional information model, which only models the managed objects at a conceptual level independent from the implementations. The YANG data model is intended for implementation. The YANG based API is bound to specific message encoding and data transport protocols that can be directly used by the client and server code. So as long as the underlying YANG modules are same, the client and server will have the same view of the API, avoiding the typical misinterpretation issues found at the interop.

4.4. Sufficiecnt Performance at Scale

The YANG based data model-driven management scheme supports a number of design options that can be used to achieve low-latency at scale. These include:

Asynchronous and Parallel RPCs

By using asynchronous and parallel RPCs, a client can avoid the serialization delay and manage multiple devices or data objects at the same time. YANG natively supports asynchronous RPCs using the concept of intended configuration and applied configuration. This allows a client to acquire the RPC result asynchronously by either polling the applied configuration or subscribing to the applied configuration update, without being blocked by the RPC results. Additionally, the client and server can use the YANG data tree to identify the interdependencies among the data objects and enable parallel processing of the data objects are isolated from each other.

Long-lived RPC sessions

Certain YANG based protocols, such as NETCONF and gNMI (gRPC), support long-lived RPC sessions to minimize session setup overhead. gNMI (gRPC), which uses HTTP/2, further allows multiple RPCs to be multiplexed onto one TCP connection. By doing so, it supports parallel RPCs without increasing the number of TCP connections.

Binary Encoding

gNMI (gRPC) supports Protobuf, an efficient binary message format, that improves message encoding/decoding processing time on both client and server. Based on Google's gRPC performance benchmark test, 700k unary RPC calls per second is achievable [5].

As to the remote US scheduler interface, the most stringent timing constraints come from two basic requirements. These are the dynamic service flow setup for voice, and the need for continuous ranging.

Let's first examine the dynamic service flow setup case as shown in Figure 5. After receiving the DSA-REQ for a voice call, the scheduler manager in the CCAP core needs to check with the US scheduler engine in the RPD to make sure the required quality path can be established. In this case, the YANG based API transaction delay contributes to the overall service flow setup time, which needs to be one second or less to meet the post-dial delay requirement [4].

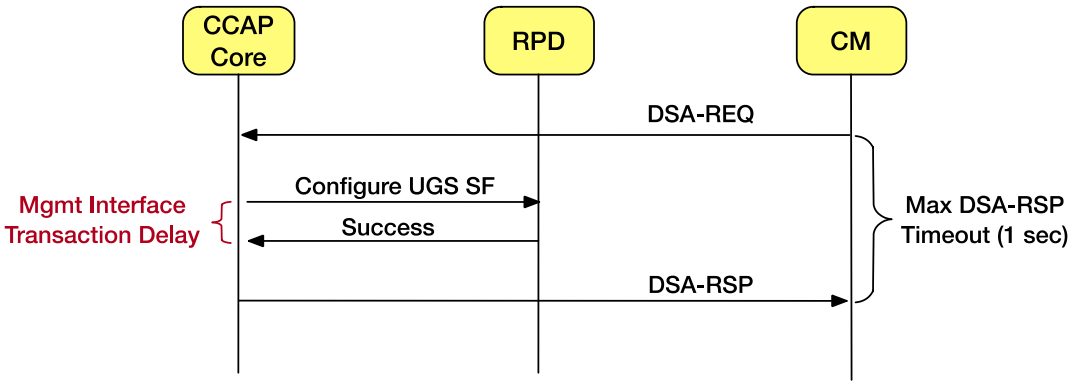

Figure 5 – Adding a UGS SF in a CM Initiated DSA Transaction

Now let's see about ranging as shown in Figure 6. After receiving the RNG-REQ from the CM, the ranging module in the CCAP core may decide to continuously range the CM by requesting a subsequent ranging opportunity in less than a second if it detects the CM deviates from the optimum timing/power/frequency settings.

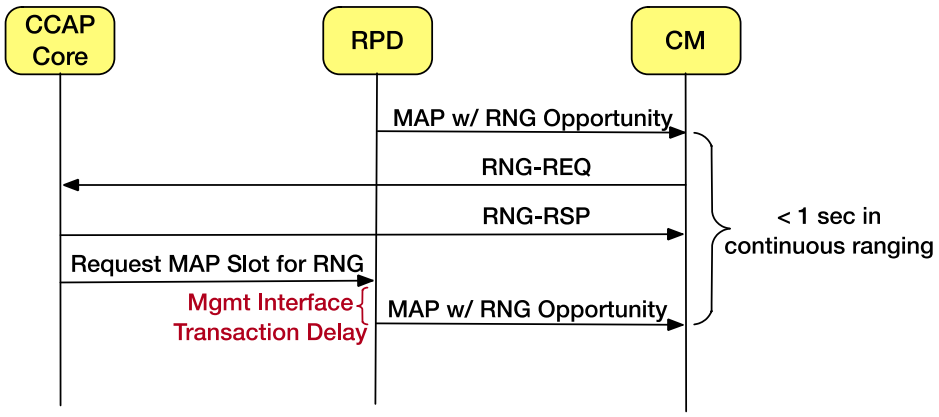

Figure 6 – Adding a RNG-REQ Opportunity in MAP

For both cases, the overall DOCSIS transaction delay is capped at one second or less. To meet this requirement, we can assume tighter budget of 100-millisecond delay budget for the YANG based remote US scheduler management interface, taking into consideration of the typical RPD CPU speed, CIN delay, and RPC transaction time.

4.5. Easy to Maintain

One advantage for using YANG based management interface is the rich tooling and applications for processing YANG modules and their metadata. YANG applications validate YANG modules, generate APIs and provide language binding. YANG metadata allows the client to pre-validate the instance data, confirm the module support in the network elements, and check for potential non-backward-compatible changes that could have been introduced between versions. YANG also has a huge library contributed by various organizations that promotes reusability.

YANG model driven telemetry (MDT) provides a new way to maintain service quality by streaming data continuously from the managed devices using a push model to give client near real-time access to the operational statistics.

5. Remote Upstream Scheduler Behavior Model

The goal is to place a remote upstream scheduling engine into an RPD that works in tandem with a remote upstream scheduling manager in the CMTS Core. Aside from the specific functionality of this system, there are some high-level objectives that have to be met to turn this into a product. Some of these at least are:

- It has to work
- It has to scale
- It has to interoperate
- It has to be maintainable

The following discussion is aimed at hitting all of these goals.

One option would just to let different manufacturers load their US schedulers from their cloud CMTS into someone else's RPD. In theory that could work, and may well be a product option. But then there is software from two manufacturers in one platform which is challenging from a built and test viewpoint.

Instead, this paper proposes building an US scheduler from one manufacturer that will interoperate with another manufacturer.

This paper proposes two fundamental principles for constructing a remote scheduler.

1. Create a well-defined behavior model
2. Use a well-defined API on that behavior model

In the construction of this behavior model and API, a very conscience decision was made to take the larger schedular and break it down into smaller independent schedulers that each had very specific functionality. There would then be a rule set on how the different schedulers interacted with each other. Finally, there would be another behavior model for the MAP Builder.

It's time for an example. Let's look at the behavior model for the unsolicited grant scheduler (UGS).

UGS Service Flow

- Service Flow ID (SFID)
- MAC domain name
- MAC domain upstream channel
- Service Identifier (SID)
- UGS QoS Parameters
  - request-policy
  - unsolicited grant size (bytes)
  - nominal grant interval (ms)
  - tolerated grant jitter (ms)
  - grants per Interval
- Grant statistics
  - cumulative grant size
  - grant rate

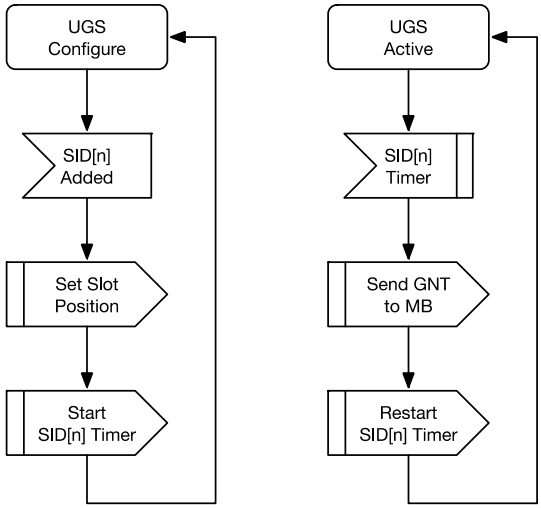

Figure 7 – UGS Behavior Model

A behavior model and a basic API is shown in Figure 7. This API will ultimately be represented in YANG but is shown here in variable form.

The behavior model has two states, configure and active. The configure state is for adding and removing service flows. The active state is run-time that generates the grants when needed. The model is simple. A service flow/SID is added into a scheduling wheel. For example, if a grant for voice-over-IP needs to be sent every 20 ms and a MAP message is every 2 ms, then there are 10 entries in the scheduling wheel. At run time, an event timer expires and a grant is scheduled.

Now let's look at the API. There are three basic elements. First, there is the logical associations. This is done with a service flow ID, a mac domain name, an upstream channel identifier and a SID. Second, is what action to take. This is described by the QoS parameters. The action is then with a request policy of UGS, to prohibit any request opportunities, instead, directly provide an unsolicited grant size (say 100 bytes) every nominal grant interval (say 20 ms) with a tolerated jitter (say 2 ms) with a grants per interval (say 1). Third, it is important to always measure what happened. In this example, the grant statistics include the cumulative grant size and the grant rate.

So, the API connects the two sides together, an action is requested, and a measurement is made to see how the action played out. That API describes a behavior model. That's it. This methodology can be repeated for the other schedulers and for the MAP builder.

In Figure 8, we see the actual implementation of the final model in the YANG tree and UML.

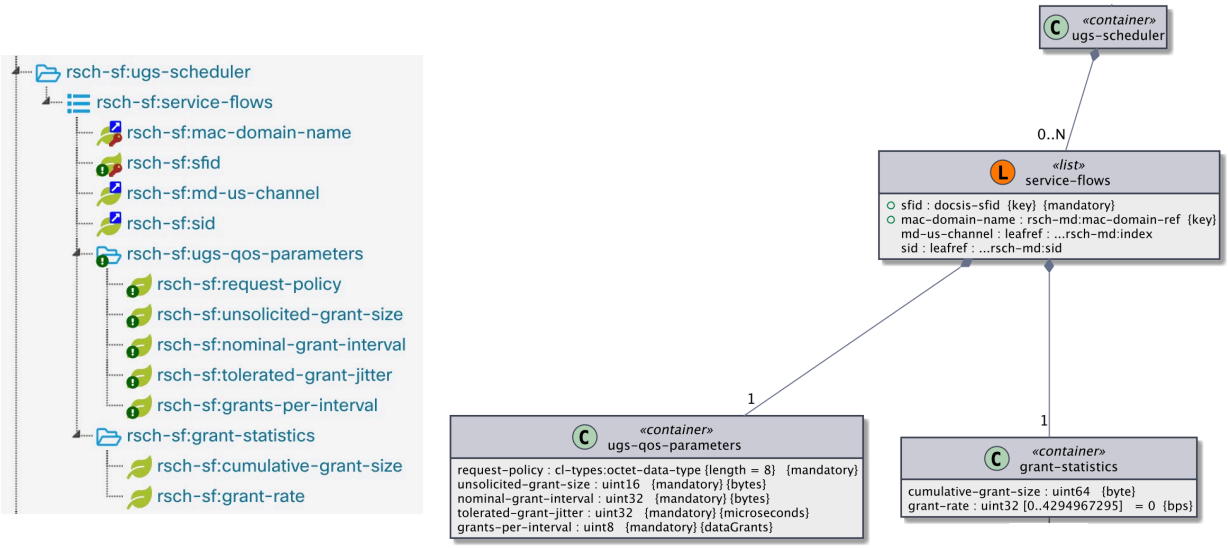

Figure 8 – UGS Diagram and UML Model

To go back to our original objectives, the model has to work. That is achieved through simplicity and modularization of the model. The model has to scale. That is achieved by setting realistic performance and scaling goals. The model has to interoperate. That is achieved by using YANG as the API and well-defined behavior models. And the model has to be maintainable. That is achieved again with YANG and the MDT features that allow monitoring and measuring of what has happened.

6. YANG Model for Remote Upstream Scheduler

The remote US Scheduler YANG model includes a base module that describes the remote US scheduler framework and several submodules that specifies the component level attributes. The overall remote US scheduler YANG model is intended for the following uses cases:

- Configure the remote US scheduler, e.g.,
  - Configure US SF scheduling type and QoS parameters
  - Configure US MAP attributes
- Invoke US scheduling actions, e.g.,
  - Request an US ranging opportunities in MAP
- Collect the remote US scheduling telemetry data, e.g.,
  - Acquire per SF requesting rate and granting rate

6.1. Base remote-us-scheduler YANG Module

The base module, titled "remote-us-scheduler" defines the remote US scheduler structure that holds together three functional submodules, namely us-scheduler-domain, us-qos-scheduler and map-builder, as shown in Figure 9. Each of the submodules can be defined separately and augmented into the base module. This arrangement decomposes a complex scheduler into smaller and simpler submodules that can be independently modeled and managed.

```
module: remote-us-scheduler
  +--rw remote-us-scheduler
     +--rw us-scheduler-domain
     +--rw us-qos-scheduler
     +--rw map-builder
```

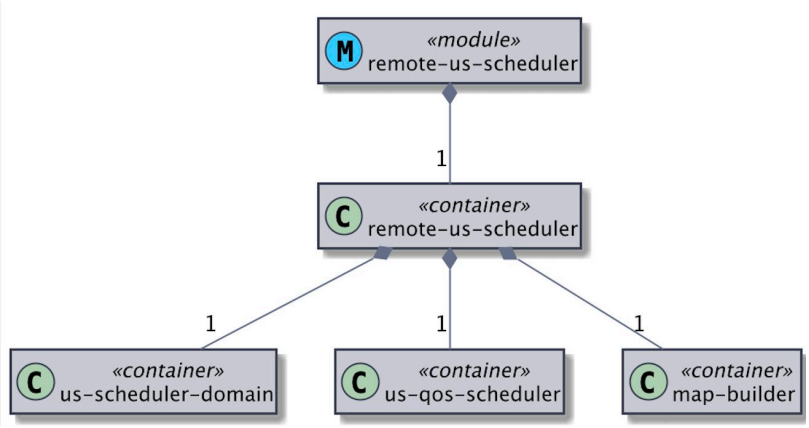

Figure 9 – remote-us-scheduler base module tree diagram and UML

6.2. us-scheduler-domain YANG Module

This module defines the MAC domains served by the remote US scheduler. Each MAC domain contains the parameters required for US scheduling, including common QoS policies, US channels and SID assignments and the list of primary capable DS channels for carrying MAP messages.

Figure 10 shows the top-levels of the us-scheduler-domain YANG module tree diagram, omitting the lower level leaf entries for simplicity. The us-scheduler-domain module is augmented to the us-scheduler-domain container in the remote-us-scheduler base module.

```
module: us-scheduler-domain
  augment /rsch:remote-us-scheduler/rsch:us-scheduler-domain:
    +--rw us-scheduler-domain* [mac-domain-name]
       +--rw mac-domain-name     string
       +--rw md-us-channel* [index]
       |  +--rw index                  uint8
       |  +--rw us-rf-chan-cfg
       |  |     ...
       |  +--rw ds-rf-chan-cfg* [port-number channel-index]
       |  |     ...
       |  +--ro md-us-channel-state
       |  |     ...
       |  +--rw sid-cfg* [sid]
       |        ...
       +--rw md-us-qos-policy* [type]
          +--rw type    identityref
          +--rw (parameters)?
```

Figure 10 – us-scheduler-domain module top-three level tree diagram

6.3. us-qos-scheduler YANG Module

This module defines the US QoS scheduler of different scheduling types, including best effort, rtPS, nrtPS, UGS, PGS and aggregated service flows (ASF). Each scheduler type represents a specific scheduler behavior as discussed in Section 5.

Figure 11 shows the top-level tree diagram of the us-qos-scheduler YANG module, omitting the lower level leaf entries for simplicity. Figure 12 shows a detailed tree-diagram for the best-effort-scheduler. The us-qos-scheduler module is augmented to the us-qos-scheduler container in the remote-us-scheduler base module.

```
module: us-qos-scheduler
  augment /rsch:remote-us-scheduler/rsch:us-qos-scheduler:
    +--rw best-effort-scheduler
    |     ...
    +--rw rtps-scheduler
    |     ...
    +--rw nrtps-scheduler
    |     ...
    +--rw ugs-scheduler
    |     ...
    +--rw ugs-ad-scheduler
    |     ...
    +--rw pgs-scheduler
    |     ...
    +--rw lld-asf-scheduler
    |     ...
    +--rw dhqos-asf-scheduler
          ...
```

Figure 11 – us-qos-scheduler module top level tree diagram

```
module: us-qos-scheduler
  augment /rsch:remote-us-scheduler/rsch:us-qos-scheduler:
    +--rw best-effort-scheduler
    |  +--rw service-flows* [mac-domain-name sfid]
    |     +--rw sfid                          docsis-sfid
    |     +--rw mtc-mode-enabled?             boolean
    |     +--rw mac-domain-name               rsch-md:mac-domain-ref
    |     +--rw (sid-channel-assignment)?
    |     |  +--:(non-mtc)
    |     |  |     ...
    |     |  +--:(mtc)
    |     |        ...
    |     +--rw best-effort-qos-parameters
    |     |  +--rw request-policy?           cl-types:octet-data-type
    |     |  +--rw priority?                 uint8
    |     |  +--rw data-rate-unit-setting?   cl-fma-qos:data-rate-type
    |     |  +--rw max-traffic-rate?         uint32
    |     |  +--rw max-traffic-burst?        uint32
    |     |  +--rw min-reserved-rate?        uint32
    |     |  +--rw min-reserved-packet?      uint16
    |     |  +--rw max-concatenated-burst?   uint16
    |     |  +--rw peak-traffic-rate?        uint32
    |     +--ro request-grant-statistics
    |        +--ro cumulative-request-size?   uint64
    |        +--ro request-rate?              uint32
    |        +--ro cumulative-grant-size?     uint64
    |        +--ro grant-rate?                uint32
```

Figure 12 – best-effort-scheduler tree diagram

6.1. Map-builder YANG Module

This module defines the MAP builder for the remote US scheduling. It contains MAP builder configuration and operational data, as well as RPC calls as shown in Figure 13.

The MAP building configuration parameters reflect channel level MAP attributes such as ranging and data back off window, broadcast IM and idle slot settings. The MAP builder RPCs are used to invoke dynamic MAP actions, such as building a station maintenance (SM) ranging opportunity or request for an OFDMA upstream data profile (OUDP) test slot in MAP. The input and output parameters allow the client to specify the action requirement and retrieve data coming out of the action. Unsolicited notifications can also be added to report the MAP builder operational status. Alternatively, the client can subscribe the operational data using telemetry.


```
module: map-builder
  augment /rsch:remote-us-scheduler/rsch:map-builder:
    +--rw map-builder* [mac-domain-name md-us-channel]
       +--rw mac-domain-name           rsch-md:mac-domain-ref
       +--rw md-us-channel             -> /rsch:remote-us-schedule
       +--rw map-channel-enable        enumeration
       +--rw ranging-backoff-start     uint8
       +--rw ranging-backoff-end       uint8
       +--rw transmit-backoff-start    uint8
       +--rw transmit-backoff-end      uint8
       +--rw broadcast-im-cfg
       |     ...
       +--rw padding-slot-cfg
       |     ...
       +--ro map-channel-state
             ...

  rpcs:
    +---x oudp-test-slot-request
    |  +---w input
    |  |     ...
    |  +--ro output
    |        ...
    +---x unicast-im-request
    |  +---w input
    |  |     ...
    |  +--ro output
    |        ...
    +---x sm-request
    |  +---w input
    |  |     ...
    |  +--ro output
    |        ...
    +---x probe-request
    |  +---w input
    |  |     ...
    |  +--ro output
    |        ...
    +---x idle-slot-request
       +---w input
       |     ...
       +--ro output
             ...
```

Figure 13 – map-builder top level tree diagram

7. Coexistence with Existing RPD Management Interface

In R-PHY today, the management interface runs over the Generic Control Protocol (GCP) using the Type-Length-Value (TLV) tuples to carry the signaling messages defined in the R-PHY specification [6]. Since the remote US scheduler is a new functional component, running GCP as it is today does not prevent using the YANG based API for the remote US scheduler, as the two management interfaces do not manipulate the same data objects.

Figure 14 shows the initialization sequence with the YANG based management interface coexisting with the legacy GCP based RPD management interface. Once the RPD is assigned with an IP address and paired with the CCAP core, the core can initiate the YANG capability discovery process. For an RPD that declares the remote US scheduler YANG support, the core can program the scheduler engine once the RPD is initialized. If there are any configuration dependencies on the GCP managed objects, for example the RF configuration, the US scheduler engine at the RPD can hold off the intended configuration and only apply it when the GCP dependencies are resolved.

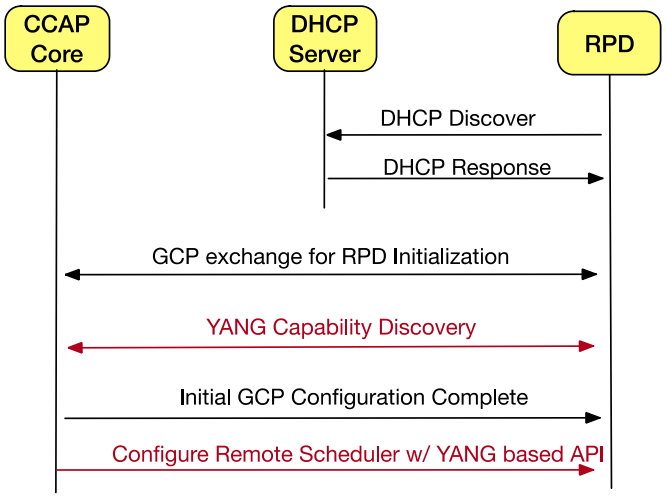

Figure 14 – Coexistence with Existing RPD Mangment Interface

With this hybrid management model, the remote US scheduler can be deployed on existing RPDs without waiting for the full-blown RPHY YANGification.

8. Remote US Scheduler in DOCSIS YANG Echosystem

In the DOCSIS YANG ecosystem, different CCAP network elements (NEs) can be organized into classification hierarchies as shown in Figure 15, where the rpd-ne (representing RPD NE) is one type of the CCAP NE that inherits the ccap-phy-ne attributes. The remote US scheduler YANG module can be incorporated into the CCAP NE hierarchy by augmenting the rpd-ne base module as shown in Figure 16. The “when” and “if-feature” tags allow the remote US scheduler module to be only present if the CCAP-NE is the RPD type and supports remote US scheduling.

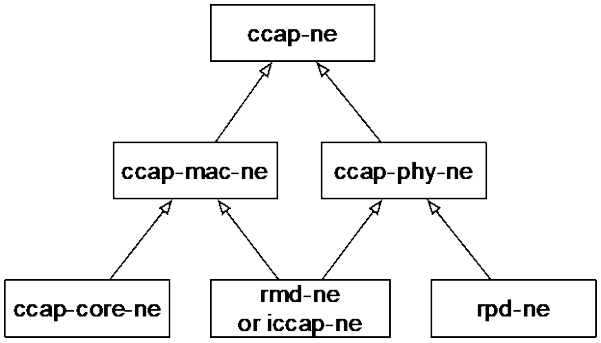

Figure 15 – CCAP Newwork Element Types and the Subclassing Hierarchy

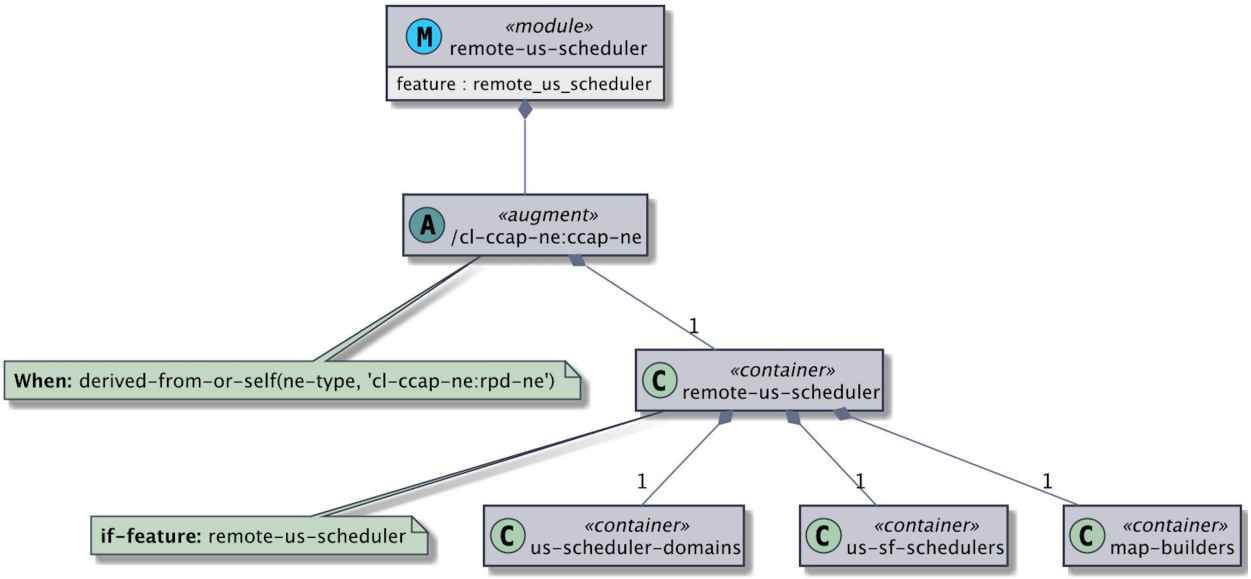

Figure 16 – Remote US Scheduler Module in DOCSIS YANG Echosystem

9. Conclusion

The remote US scheduler is an operational option for R-PHY when the desired latency performance needs to be better than a classic I-CCAP or if the CIN needs to be significantly longer than the 100 miles. The remote US scheduler design has two functional parts, the US scheduler manager that lives in the CCAP core and the US scheduler engine that lives in the RPD. The US scheduler manager northbound is internal to the CCAP core, while the southbound is a data model driven interface that allows the US scheduler manager to communicate with one or more US scheduler engines to fulfill the US scheduling services.

This paper proposes to use a YANG data model-driven interface between the US scheduler manager and the US scheduler engines. The YANG data-model driven management techniques are mature and well adopted in the network industry, with rich tooling and applications that can auto-generate code from the data model; hence facilitating multi-vendor interoperability. The YANG data model can be paired with high performance message encoding and RPC options (such as protobuf and gRPC) to meet the timing and scaling requirement for managing remote US schedulers.

To construct the remote US scheduler YANG data model, this paper presents a modeling method that involves decomposing a complex service function into smaller independently manageable subservices, creating a well-defined behavior model for each subservice, and defining the API that completely and precisely characterizes the behavior.

The YANG model-driven management interface for remote US scheduler can co-exist with existing GCP based RPD management interface, reducing time-to-market for the remote US scheduler feature while R-PHY is transitioning to YANG data-model driven management. The remote US schedule YANG model is an integral part of the DOCSIS YANG ecosystem, it is planned to be contributed to CableLabs as a feature augmented to the R-PHY YANG model.

Abbreviations

API	application programming interface
CM	cable modem
CCAP	converged cable access platform
CIN	converged interconnect network
DS	downstream
DSA	dynamic service addition
FMA	flexible MAC architecture
GCP	generic control protocol
gNMI	gRPC network management interface
GNT	DOCSIS bandwidth grant
gRPC	Google RPC
MAP	DOCSIS bandwidth request
NE	network element
RPC	remote Procedure Call
RPD	remote PHY device
RPD-NE	RPD Network Element
YANG	yet another next gen (data modeling language)

Bibliography & References

[1] Tong Liu, John Chapman, "R-PHY with Remote Upstream Scheduler", *2019 SCTE Expo Technical Forum Proceedings*, Oct, 2019.

[2] "CM-SP-MULPIv3.1-I18-190422: MAC and Upper Layer Protocols Interface Specification", CableLabs, 2019

[3] Joe Clarke, Jan Lindblad, Benoit Claise: *Network Programmability with YANG* Addison-Wesley Professional Book

[4] ITU-T: Series J: Cable Networks and Transmission of Television, Sound Programm and Other Multimedia Signals

[5] https://grpc.io/docs/guides/benchmarking/

[6] "CM-SP-R-PHY-I14-200323: DOCSIS Remote PHY Specification", CableLabs, 2020

