**QOE RATE PLAN MANAGEMENT**

**USING REST API & PYTHON**

**Cambium Networks QoE**

Graphical user interface, website

Description automatically generated

Contents

[Introduction 3](#_Toc114760817)

[Functionality 3](#_Toc114760818)

[Configuration 3](#_Toc114760819)

[Usage 5](#_Toc114760820)

[Rate Policy Management 6](#_Toc114760821)

[Add or Update Rate Policy 6](#_Toc114760822)

[Delete Rate Policy 6](#_Toc114760823)

[Retrieve Rate Policy 7](#_Toc114760824)

[Assign (Update) Subscriber’s Rate Policy 8](#_Toc114760825)

[Retrieve Subscriber’s Rate Policy 8](#_Toc114760826)

[Delete Subscriber’s Rate Policy 9](#_Toc114760827)

[Retrieve Subscriber Metrics 10](#_Toc114760828)

# Introduction

This tool allows the QoE user to manage subscriber rate plans from any system using REST APIs. QoE REST API documentation is available at Cambium Networks support site (<https://support.cambiumnetworks.com/files/qoe_qoe/>).

The script is developed by Cambium Networks to illustrate how to use REST APIs to manage subscribers rate plans. The script can be updated by end users to meet their system configuration and network model.

# Functionality

The tool provides the following functionalities:

1. Add Rate Policy (Service Plan)
2. Update Rate Policy (Service Plan)
3. Delete Rate Policy (Service Plan)
4. Retrieve Rate Policy (Service Plan)
5. Assign Subscriber to Rate Policy
6. Update Subscriber’s Rate Policy
7. Delete Subscriber’s Rate Policy
8. Retrieve Subscriber’s Rate Policy
9. Retrieve Subscriber’s Metrics

# Configuration

There are 4 configuration parameters that must be changed in the script located at the top section with the “To Do” title:

* QoE\_MNG\_IP = "10.0.0.100"
* QoE\_REST\_PORT = "3443"
* QoE\_REST\_USER = "qoe-rest-user"
* QoE\_REST\_PASSWORD = "qoe-rest-passwd"
* The QoE\_MNG\_IP is the management IP address of the QoE server.
* The QoE\_REST\_PORT is the REST API Port number configure don the QoE.
* The QoE\_REST\_USER is the REST API Users username configure don the QoE.
* The QoE\_REST\_PASSWORD is the REST API Users password configure don the QoE.

The following image shows an example of how-to setup a REST APU User on the on the QoE

Graphical user interface, application

Description automatically generated

To add a REST API User click on the  and select Add User

Graphical user interface, application

Description automatically generated

The following windows will pop up:

Graphical user interface, text, application, Teams

Description automatically generated

Enter the user and password and click on OK. Update the python script QoE\_REST\_USER and QoE\_REST\_PASSWORD with the configured user and password.

Python

Python version 3.10 or above

Install the argpars and optparse modules. (pip install argpars .. pip install optparse)

# Usage

The script has a built is help option to display usage:

**C:\python.exe QoE-REST-RatePlansManagement.py -h**

**usage: QoE-REST-RatePlansManagement.py [-h] [-p POLICYNAME] [-pi POLICYID] [-dl DOWNLINKRATE] [-ul UPLINKRATE] [-acm ACM] [-s SUBSCRIBER] [-si SUBSCRIBERID] [-m METRIC] [-mi METRIC\_INTERVAL] [-mp METRIC\_PERIOD] action**

**mandatory arguments:**

**action Action to be performed: addPolicy|getPolicy|deletePolicy|setSubRatePolicy|getSubRatePolicy|deleteSubRatePolicy|getSubMetrics**

**options:**

**-h, --help show this help message and exit**

**-p POLICYNAME, --policyName POLICYNAME**

**Policy Name, no spaces or special characters**

**-pi POLICYID, --policyId POLICYID**

**Policy Identifier, no spaces or special characters**

**-dl DOWNLINKRATE, --downlinkRate DOWNLINKRATE**

**Downlink rate in kbps**

**-ul UPLINKRATE, --uplinkRate UPLINKRATE**

**Uplink rate in kbps**

**-acm ACM, --acm ACM Automatic Congestion Management (ACM) enabled: true/false**

**-s SUBSCRIBER, --subscriber SUBSCRIBER**

**subscriber IP address, IPv4 only**

**-si SUBSCRIBERID, --subscriberId SUBSCRIBERID**

**subscriber Id**

**-m METRIC, --metric METRIC**

**subscriber metric to retrieve: bandwidth|flows|latency|retransmission|volume**

**-mi METRIC\_INTERVAL, --metric\_interval METRIC\_INTERVAL**

**subscriber metric time interval in minutes (default 60 minutes)**

**-mp METRIC\_PERIOD, --metric\_period METRIC\_PERIOD**

**subscriber metric period in hours (default: 24 hours). The maximum query period is 3 months**

# Rate Policy Management

## Add or Update Rate Policy

*Usage:*

*python.exe QoE-REST-RatePlansManagement.py addPolicy --policyName [policyname] --policyId [policyID] --downlinkRate [dl\_rate\_kbps] --uplinkRate [ul\_rate\_kbps] --acm [true/false]*

The following example adds (or updates) a rate policy with:

* Name: 100MDL\_20MUL
* Policy ID: 100MDL\_Policy
* DL Rate: 100 Mbps
* UL Rate: 20 Mbps
* ACM (Automatic Congestion Management feature): true (enabled)

**C:\python.exe QoE-REST-RatePlansManagement.py addPolicy -p 100MDL\_20MUL -pi 100MDL\_Policy -dl 100000 -ul 20000 -acm true**

## Delete Rate Policy

*Usage:*

*python.exe QoE-REST-RatePlansManagement.py deletePolicy --policyName [policyname]*

The following example deletes a rate policy named 100MDL\_20MUL

**C:\python.exe QoE-REST-RatePlansManagement.py deletePolicy -p 100MDL\_20MUL**

The following example deletes ALL rate policies with Policy ID 100MDL\_Policy

**C:\python.exe QoE-REST-RatePlansManagement.py deletePolicy -pi 100MDL\_Policy**

## Retrieve Rate Policy

*Usage:*

*python.exe QoE-REST-RatePlansManagement.py getPolicy [--policyName [policyname]]*

The following example retrieves the rate policy named 100MDL\_20MUL

**C:\python.exe QoE-REST-RatePlansManagement.py getPolicy -p 100MDL\_20MUL**

The output is in JSON format:

{

"policyName": "100MDL\_20MUL",

"policyId": "100MDL\_Policy",

"rateLimitDownlink": {

"rate": 100000,

"congestionMgmt": true

},

"rateLimitUplink": {

"rate": 20000

}

}

The following example retrieves ALL rate policies

**C:\python.exe QoE-REST-RatePlansManagement.py getPolicy**

The output is in JSON format:

{

"items": [

{

"policyName": "100MDL\_20MUL",

"policyId": "100MDL\_Policy",

"rateLimitDownlink": {

"rate": 100000,

"congestionMgmt": true

},

"rateLimitUplink": {

"rate": 20000

}

},

"policyName": "100MDL\_30MUL",

"policyId": "100MDL\_Policy",

"rateLimitDownlink": {

"rate": 100000,

"congestionMgmt": true

},

"rateLimitUplink": {

"rate": 30000

}

},

{

"policyName": "100MDL\_50MUL",

"policyId": "100MDL\_Policy",

"rateLimitDownlink": {

"rate": 100000,

"congestionMgmt": true

},

"rateLimitUplink": {

"rate": 50000

}

}

]

}

## Assign (Update) Subscriber’s Rate Policy

*Usage:*

*python.exe QoE-REST-RatePlansManagement.py setSubRatePolicy --subscriber [IPv4] --subscriberId [ID] --policyName [policyname]*

The following example assigns subscriber 10.0.0.133 to rate policy named 100MDL\_20MUL

**C:\python.exe QoE-REST-RatePlansManagement.py setSubRatePolicy -s 10.0.0.133 -si John-Brown -p 100MDL\_20MUL**

The following example changes subscriber 10.0.0.133 rate policy to 100MDL\_50MUL

**C:\python.exe QoE-REST-RatePlansManagement.py setSubRatePolicy -s 10.0.0.133 -si John-Brown -p 100MDL\_50MUL**

## Retrieve Subscriber’s Rate Policy

*Usage:*

*python.exe QoE-REST-RatePlansManagement.py getSubRatePolicy --subscriber [IPv4]*

The following example retrieves subscriber 10.0.0.133 rate policy

**C:\python.exe QoE-REST-RatePlansManagement.py getSubRatePolicy -s 10.0.0.133**

The output is in JSON format:

{

"subscriberIp": "10.0.0.133",

"subscriberId": "John-Brown",

"policyRate": "100MDL\_50MUL"

}

The following example retrieves the rate policy of ALL subscribers configured through REST:

**C:\python.exe QoE-REST-RatePlansManagement.py getSubRatePolicy**

The output is in JSON format:

{

"items": [

{

"subscriberIp": "10.0.0.133",

"subscriberId": "John-Brown",

"policyRate": "100MDL\_50MUL"

},

{

"subscriberIp": "10.0.0.100",

"subscriberId": "Mike-Brown",

"policyRate": "100MDL\_20MUL"

},

{

"subscriberIp": "10.0.0.90",

"subscriberId": "Al-Harris",

"policyRate": "100MDL\_20MUL"

}

]

}

## Delete Subscriber’s Rate Policy

*Usage:*

*python.exe QoE-REST-RatePlansManagement.py deleteSubRatePolicy --subscriber [IPv4] | --subscriberId [ID]*

The following example deletes the rate policy of subscriber 10.0.0.100 using the IP address

**C:\python.exe QoE-REST-RatePlansManagement.py deleteSubRatePolicy -s 10.0.0.100**

The following example deletes the rate policy of subscriber 10.0.0.90 using thesubscriber ID

**C:\python.exe QoE-REST-RatePlansManagement.py deleteSubRatePolicy -si Al-Harris**

## Retrieve Subscriber Metrics

*Usage:*

*python.exe QoE-REST-RatePlansManagement.py getSubMetrics --subscriber [IPv4] --metric [bandwidth|flows|latency|retransmission|volume] --metric\_interval [interval-minutes] --metric\_period [period-hours]*

The following example retrieves the volume metrics for subscriber 10.0.0.133 in the last 2 hours with interval 5 minutes:

**C:\python.exe QoE-REST-RatePlansManagement.py getSubMetrics -s 10.0.0.133 -m volume -mp 2 -mi 5**

The output is in JSON format:

{

"subscriberIp": "10.0.0.133",

"timestamp": [

1663878000,

1663878300,

1663878600,

1663878900,

1663879200,

1663879500,

1663879800,

1663880100,

1663880400,

1663880700,

1663881000,

1663881300,

1663881600,

1663881900,

1663882200,

1663882500,

1663882800,

1663883100,

1663883400,

1663883700,

1663884000,

1663884300,

1663884600,

1663884900,

1663885200

],

"dataDownlink": [

0.9675,

1.5055,

1.399,

1.4171,

2.832,

1.0049,

5.0151,

1.1529,

6.4514,

65.889,

0.6082,

0.9383,

0.9498,

9.7952,

1.2841,

0.6781,

124.3488,

1.27,

0.9199,

2.028,

22.6179,

0.7469,

4.8027,

2.9811,

1.1995

],

"dataUplink": [

0.6642,

0.8723,

1.0162,

1.3307,

1.155,

1.024,

1.1205,

1.9935,

1.6216,

1.6804,

0.6573,

0.8235,

1.2274,

1.3263,

0.8746,

0.753,

2.0302,

0.7017,

0.9183,

2.4477,

1.24,

0.8684,

1.3673,

1.1814,

0.8274

]

}

The entries show Dl Volume and UL Volume in MB for each timestamp entry.