

Ubiquiti

Adding, Configuring and Troubleshooting DSCP VoIP QoS in EdgeOS

In our FreePBX installs, we use the following upstream shaper based on DSCP markings. Modify the bandwidth and DSCP marking values as necessary for the site. Note that shaping traffic will disable offload on **that direction** of traffic. You can also use this shaper for UniFi environments by using [config.gateway.json](#).

```
set traffic-policy shaper Upstream description "Upstream QoS policy"
set traffic-policy shaper Upstream bandwidth 100mbit
set traffic-policy shaper Upstream class 10 description "RTP"
set traffic-policy shaper Upstream class 10 bandwidth 10%
set traffic-policy shaper Upstream class 10 burst 15k
set traffic-policy shaper Upstream class 10 ceiling 100%
set traffic-policy shaper Upstream class 10 match VOIP-RTP ip dscp 46
set traffic-policy shaper Upstream class 10 priority 0
set traffic-policy shaper Upstream class 10 queue-type fair-queue
set traffic-policy shaper Upstream class 20 description "SIP"
set traffic-policy shaper Upstream class 20 bandwidth 10%
set traffic-policy shaper Upstream class 20 burst 15k
set traffic-policy shaper Upstream class 20 ceiling 100%
set traffic-policy shaper Upstream class 20 match VOIP-SIP ip dscp 24
set traffic-policy shaper Upstream class 20 priority 1
set traffic-policy shaper Upstream class 20 queue-type fair-queue
set traffic-policy shaper Upstream default bandwidth 90%
set traffic-policy shaper Upstream default ceiling 100%
```

Apply the policy to the internet interface.

```
set interfaces ethernet ethX traffic-policy out Upstream
```

In our usual deployments we don't shape downstream traffic, as it's typically counterproductive in an asynchronous or high bandwidth environment. However in some cases it's helpful, for instance with low bandwidth circuits. Use the script below to implement QoS on the downstream.

```
set traffic-policy shaper Downstream description "Downstream QoS policy"
set traffic-policy shaper Downstream bandwidth 100mbit
set traffic-policy shaper Downstream class 10 description "RTP"
set traffic-policy shaper Downstream class 10 bandwidth 10%
set traffic-policy shaper Downstream class 10 burst 15k
set traffic-policy shaper Downstream class 10 ceiling 100%
set traffic-policy shaper Downstream class 10 match VOIP-RTP ip dscp 46
set traffic-policy shaper Downstream class 10 priority 0
set traffic-policy shaper Downstream class 10 queue-type fair-queue
set traffic-policy shaper Downstream class 20 description "SIP"
set traffic-policy shaper Downstream class 20 bandwidth 10%
set traffic-policy shaper Downstream class 20 burst 15k
```

Ubiquiti

```
set traffic-policy shaper Downstream class 20 ceiling 100%
set traffic-policy shaper Downstream class 20 match VOIP-SIP ip dscp 24
set traffic-policy shaper Downstream class 20 priority 1
set traffic-policy shaper Downstream class 20 queue-type fair-queue
set traffic-policy shaper Downstream default bandwidth 90%
set traffic-policy shaper Downstream default ceiling 100%
```

Create an input interface, assign the shaper to the input and redirect the downstream to the input interface.

```
set interfaces ifb0 description "Downstream Input for QoS"
set interfaces ifb0 traffic-policy out Downstream
set interfaces ethernet ethX redirect ifb0
```

Additionally the **smart queue** feature works well, another alternative is to set it up on the Upstream interface. Here are optimizations for various bandwidth levels:

```
# 100mbps up
set traffic-control smart-queue QoS upload fq-quantum 1514
set traffic-control smart-queue QoS upload htb-quantum 8000
set traffic-control smart-queue QoS upload target 5ms
```

```
# 70mbps up
set traffic-control smart-queue QoS upload fq-quantum 1514
set traffic-control smart-queue QoS upload htb-quantum 5000
set traffic-control smart-queue QoS upload target 5ms
```

```
# 40mbps or less
set traffic-control smart-queue QoS upload fq-quantum 300
set traffic-control smart-queue QoS upload htb-quantum 1500
set traffic-control smart-queue QoS upload target 5ms
```

```
# 3mbps or less
set traffic-control smart-queue QoS upload fq-quantum 300
set traffic-control smart-queue QoS upload htb-quantum 1500
set traffic-control smart-queue QoS upload target 15ms
```

To troubleshoot your QoS implementation you can use the following commands:

```
sudo tc -s qdisc ls dev ethX
sudo tc -s class show dev ethX
```

DSCP reference chart:

Ubiquiti

Application	CoS=IPP	AF	DSCP	ToS	ToS HEX	DP
Best Effort	0	0	0	0	0	
Scavenger	1	CS1	8	32	20	
Bulk Data	1	AF11	10	40	28	Low
	1	AF12	12	48	30	Medium
	1	AF13	14	56	38	High
Network Mgmt.	2	CS2	16	64	40	
Transaction Data	2	AF21	18	72	48	Low
	2	AF22	20	80	50	Medium
	2	AF23	22	88	58	High
Call Signaling	3	CS3	24	96	60	
Mission-Critical	3	AF31	26	104	68	Low
Streaming Video	3	AF32	28	112	70	Medium
	3	AF33	30	120	78	High
	4	CS4	32	128	80	
Interactive Video	4	AF41	34	136	88	Low
	4	AF42	36	144	90	Medium
	4	AF43	38	152	98	High
	5	CS5	40	160	A0	
Voice	5	EF	46	184	B8	
Routing	6	CS6	48	192	C0	
	7	CS7	56	224	E0	

Unique solution ID: #1061

Author: n/a

Last update: 2021-02-02 09:34