Progress®
Kemp® LoadMaster®

Feature Description Rate Limiting

Generated on: 10 April 2025

Copyright

Visit the following page online to see Progress Software Corporation's current Product Documentation Copyright Notice/Trademark Legend: Product Documentation Copyright Notice & Trademarks | Progress

Table of Contents

Chapter 1: Introduction 4

Chapter 2: Limiting QoS UI Options 6

Global Limits 7

Limiter Options 7

Client Limiting 8

Per-Virtual Service Limiting 13

Client Limit Statistics 14

Virtual Service Limiting Statistics 14

Chapter 3: Rate Limiting Log Information 15

Chapter 4: RESTful API Details 17

Maximum Client Concurrent Connection Limit 18

URL-Based Limiting Rules 21

Statistics 22

1

Introduction

In firmware version 7.2.52, a new **QoS/Limiting** feature was introduced. The terms Quality of Service (QoS) and limiting are used interchangeably. Throughout the remainder of this document, this feature is referred to as limiting. This is a system-level limit/rate controller. It tracks ingress activity. The purpose of the limiting feature is to protect the machine as a whole. Rate limiting can guard against certain types of attacks, for example Distributed Denial of Service (DDoS) or brute-force password-guessing attacks. You can also use rate limiting to protect servers from being overwhelmed by too many requests at once.

An example scenario may be that a machine becomes resource-saturated, for example, 100% CPU utilization at 1,000 Connections Per Second (CPS) and 10,000 Requests Per Second (RPS). You may never want a machine to saturate. With the limiting feature in the LoadMaster, you can apply a system-level controller to cap or curtail levels of ingress traffic to the LoadMaster (for example, 800 CPS and 8,000 RPS).

You can configure:

- Max connections (the maximum number of established connections)
- Connections Per Second (CPS) rate
- Requests Per Second (RPS) rate
- Bandwidth limits

Note: In firmware version 7.2.53, the rate limiting and Quality of Service (QoS) capabilities have been enhanced to support bandwidth limiting at three levels.

- **Global:** Across all clients accessing any Virtual Service.-
- **Client:** For a single IP address or subnet accessing any Virtual Service.

- **Virtual Service:** For any client accessing a specific Virtual Service or SubVS.

The global and client limits are available in the User Interface (UI) on the **System Configuration > QoS/ Limiting** page, at the bottom of **Global Limits** and at the bottom of the **Client Limiting** section. A bandwidth limit set at the global level overrides one set at either the client or Virtual Service/SubVS levels.

You can also set bandwidth limits at the Virtual Service and SubVS levels using a new control at the bottom of the Virtual Service and SubVS **QoS/Limiting** section. A bandwidth limit set at the Virtual Service level overrides one set at the SubVS level. Similarly, a bandwidth limit set at the global level overrides one set at the client, Virtual Service, or SubVS level.

A log is generated every five seconds (this is configurable and is off by default) to include the following information:

- Current active connections
- Current CPS
- Current RPS
- Current CPS being rate-controlled (that is, the number being rejected)
- Current RPS being rate controlled (that is, the number being rejected)

2

Limiting QoS UI Options

Limiting UI Options

This section describes the limiting options available in the LoadMaster User Interface (UI). To access the limiting screen in the UI, go to **System Configuration > QoS/Limiting**.

Related Links

For details, see the following topics:

- Global Limits
- Limiter Options
- Client Limiting
- Per-Virtual Service Limiting
- Client Limit Statistics
- Virtual Service Limiting Statistics

Global Limits

Global Limits

Global Limits

In the **Global Limits** section, you can configure the following options:

- **Maximum Concurrent Connections:** Limit the maximum number of simultaneous connections (combined total of TCP and UDP connections) allowed to the LoadMaster. Setting the limit to 0 disables this option. Valid values are 0 - 100000000.

Note: The maximum values are based on the hardware or Virtual LoadMaster that is in use and may vary per model.

- **Global Connections/s Limit:** Limit the maximum number of connection attempts (per second). Setting the limit to 0 disables this option. Valid values are 0 - 1000000.
- **Global HTTP Requests/s Limit:** Limit the maximum number of HTTP request attempts (per second). This has no effect on non-HTTP traffic. Setting the limit to 0 disables this option. Valid values are 0 - 1000000.

Note: The **Global Limits** take precedence over the other limits configured. For example, if you set the **Client Concurrent Connection Limit** to **5000** but the global **Maximum Concurrent Connections** limit is set to **50**, then 50 is the limit that is enforced.

Note: If the total number of connections from all clients exceed the global limit, they will be dropped.

- **Global Bandwidth Limit:** The global bandwidth limit. Setting the limit to **0** disables bandwidth limiting. Units are in kilobits/second. The minimum value is 16 kilobits/second (2 kilobytes/second). The maximum is 99999999 (which is just under 100 Gbit) but most LoadMasters have a bandwidth limit set in the license and the license bandwidth limit will be enforced if the value specified in the **Global Bandwidth Limit** field is greater than that. When calculating bandwidth, data in both directions are tracked and used in the calculation.

Note: You can also configure per-Virtual Service bandwidth limiting. For further details, refer to the Per-Virtual Service Bandwidth Limiting section. The lowest of the three possible limits (global, client, and Virtual Service) that is reached first is enforced. Note that the global limit is for all Virtual Services, the Virtual Service limit is for the current Virtual Service with multiple clients, and the client limit is for a single client.

Limiter Options

Limiter Options

Limiter Options

In the **Limiter Options** section, you can configure the following options:

- **Error Responses:** By default, the LoadMaster simply drops any connections when the RPS limit is

reached. The system can send a 429 or 503 HTTP error response instead (followed by a close) if you select the appropriate option in this drop-down list.

- **Fail on RS/Sub-VS Rate Limiting:** If rate limiting is activated for a Real Server (RS) or a SubVS, the LoadMaster normally tries to select a different RS/SubVS to use for the connection. Enabling this check box forces the request to fail if the RS that was selected (for example, by persistence) was rate limited. An error response is sent back if one is selected in the **Error Responses** drop-down list.
- **Generate Limiter Statistics:** Enabling this option generates a global summary syslog message every five seconds containing the current state of the limiting subsystem.

Note: This option is disabled by default. Depending upon your client limiting configuration, this can generate a lot of log messages which could be resource intensive.

- **Client Message Repeat Delay:** Set the minimum time after a client is no longer limited before a new message is generated. If a client generates a message and continues to be blocked for continuously hitting the limit, no new message is generated. Only if the client goes quiet for the delay period will a new message be generated. Valid values range from 10 - 86400 seconds. The default value is 60 seconds.

Client Limiting

Client Limiting

Refer to the sections below for details on the client limiting options.

Maximum Client Concurrent Connection Limit

In the **Maximum Client Concurrent Connection Limit** section, you must configure the global **Client Concurrent Connection Limit** before you get options to configure concurrent connection limits for particular addresses or networks. The **Client Concurrent Connection Limit** limits the default maximum number of concurrent connection attempts (per second) from a specific host. Setting the limit to 0 disables this option. Valid values range from 0 - 1000000.

When you set a **Client Concurrent Connection Limit**, each client has this limit unless you have a specific entry for that client. If there is a specific limit entry for a client, the client-specific limit is applied. The options allow you to specify addresses or networks with particular limits for the concurrent connection attempts (per second) from that specific host/network. If you specify a subnet, all clients in the subnet get the same limit.

Note: The global **Maximum Concurrent Connection** value takes precedence over the client concurrent connection limits. If you try to set a client concurrent connection limit to a value greater than what is currently configured as the Maximum Concurrent Connections limit, you will get an error message.

Note: If you attempt to set a new specific concurrent connection limit for a particular address or network that has a limit that is greater than the **Client Concurrent Connection Limit**, you will get a warning message and will be asked if you want to continue and confirm the change.

Client Connections/sec Limit

In the **Client Connections/sec Limit** section, you must configure the global **Client Connection Limit** before you get options to configure the CPS limits for particular addresses or networks. The **Client Connection Limit** limits the default maximum number of connection attempts (per second) from a specific host. Setting the limit to 0 disables this option. Valid values range from 0 - 1000000.

When you set a **Client Connection Limit**, each client has this limit unless you have a specific entry for that client. If there is a specific limit entry for a client, the client-specific limit is applied. The options allow you to specify addresses or networks with particular limits for connection attempts (per second) from that specific host/network. If you specify a subnet, all clients in the subnet get the same limit. When there are multiple subnets, the lower limit applies.

Note: The **Global Connections/s Limit** value takes precedence over the client CPS limits. If you try to set a client CPS limit to a value greater than what is currently configured as the **Global Connections/s Limit**, you will get an error message.

Note: If you attempt to set a new specific CPS limit for a particular address or network that has a limit that is greater than the **Client Connection Limit**, you will get a warning message and will be asked if you want to continue and confirm the change.

Client Requests/sec Limit

In the **Client RPS Limit** section, you must configure the global **Client HTTP Request Limit** before you get options to configure the RPS limits for particular addresses or networks. The **Client HTTP Request Limit** limits the default maximum number of HTTP request attempts (per second) from a specific host. This has no effect on non-HTTP traffic. Setting the limit to 0 disables this option. Valid values range from 0 - 1000000.

When you set a **Client HTTP Request Limit**, each client has this limit unless you have a specific entry for that client. If there is a specific limit entry for a client, the client-specific limit is applied. The options allow you to specify addresses or networks with particular limits for HTTP request attempts (per second) from that specific host/network. If you specify a subnet, all clients in the subnet get the same limit. When there are multiple subnets, the lower limit applies.

Note: The **Global HTTP Requests/s Limit** value takes precedence over the client RPS limits. If you try to set a client RPS limit to a value greater than what is currently configured as the **Global HTTP Requests/s Limit**, you will get an error message.

Note: If you attempt to set a new specific RPS limit for a particular address or network that has a limit that is greater than the **Client HTTP Request Limit**, you will get a warning message and will be asked if you want to continue and confirm the change.

Client Bandwidth Limit

In the **Client Bandwidth Limit** section, you must configure the global **Client Bandwidth Limit** before you get options to configure the bandwidth limits for particular addresses or networks. The **Client Bandwidth Limit** limits the default maximum number of bandwidth attempts (per second) from a specific host. Setting the limit to **0** disables this option. Units are in kilobits/second. The minimum value is 16 kilobits/second (2

kilobytes/second). The maximum is the value configured in the **Global Bandwidth Limit**. When calculating bandwidth, data in both directions is tracked and used. This means both the client and server-side data is tracked and used as part of the calculation.

When you set a **Client Bandwidth Limit**, each client has this limit unless you have a specific entry for that client. If there is a specific limit entry for a client, the client-specific limit is applied. The options allow you to specify addresses or networks with particular limits for bandwidth used by that specific host/network If you specify a subnet, all clients in the subnet get the same limit. When there are multiple subnets, the lower limit applies.

Note: The **Global Bandwidth Limit** value takes precedence over the client bandwidth limits. If you try to set a client bandwidth limit to a value greater than what is currently configured as the **Global Bandwidth Limit**, you will get an error message.

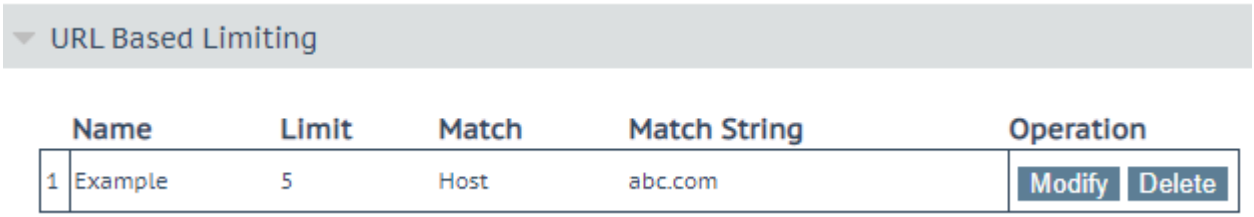
Note: If you attempt to set a new specific bandwidth limit for a particular address or network that has a limit that is greater than the **Client Bandwidth Limit**, you will get a warning message and will be asked if you want to continue and confirm the change.

Note: You can also configure per-Virtual Service bandwidth limiting. For further details, refer to the Per-Virtual Service Bandwidth Limiting section. The lowest of the three possible limits (global, client, and Virtual Service) that is reached first is enforced. Note that the global limit is for all Virtual Services, the Virtual Service limit is for the current Virtual Service with multiple clients, and the client limit is for a single client.

URL Based Limiting

The **URL Based Limiting** is based on options in a HTTP request. A request consists of a URL, Method, and request headers. **Host** and **User-Agent** are request headers. The LoadMaster URL-based limiting rules inspect based on what is selected in the **Match** drop-down list (**Request URL**, **Host**, **User Agent**, **Method**, **!Request URL**, **!Host**, **!User Agent**, or **!Method**. If the limit is hit the LoadMaster sends a response code (as set in the **Error Responses** drop-down list in the **Limiter Options** section).

Note: The values with an exclamation mark (**!**) before them matches the inverse, for example, not a specific request or not a specific user agent.



The above screenshot shows a simple example. If a request comes into the LoadMaster with a host header of **abc.com** this rule gets triggered and if the requests per second is greater than the limit set on the rule, the LoadMaster limits the request and sends out the response depending on what is selected in the **Error Responses** drop-down list.

Here is a further breakdown of this example:

- A rule exists for a host **abc.com** with a **Limit** of **5** RPS

- The **Error Responses** drop-down list is set to **Send 429 Too Many Requests**
- Requests or traffic with the specified host header is hitting the RPS limit (sending 10 RPS)
- For the requests breaching the limit, a **429 Rate Limited Rate Limit exceeded** response is sent

The request should have the **host** header as **abc.com**, for example:

```
GET /a.html HTTP/1.1

Accept: */*

Accept-Encoding: gzip, deflate

Connection: keep-alive

host: abc.com
```

Here is the example response when the limit is not hit:

```
HTTP/1.1 200 OK

Date: Tue, 08 Sep 2020 15:15:33 GMT

Server: Apache/2.4.7 (Ubuntu)

Last-Modified: Fri, 15 Feb 2019 07:40:17 GMT

ETag: "1d-581e9e2e8d033"

Accept-Ranges: bytes

Content-Length: 29

Accept: */*

User-Agent: qa-agen

Accept-Encoding: gzip, deflate

Connection: keep-alive, Keep-Alive

host: abc.com

Via: 1.1 172.16.178.55:80

X-Forwarded-For: 172.16.128.217

X-Forwarded-For-Port: 54385
```

```
MyHeader1: D=138 t=1599578133851755

Keep-Alive: timeout=150, max=100

Content-Type: text/html
```

Here is the example response when the RPS limit is hit:

```
HTTP/1.1 429 Rate Limited

Date: Tue, 08 Sep 2020 15:15:33 GMT

X-Frame-Options: SAMEORIGIN

X-XSS-Protection: 1; mode=block

X-Content-Type-Options: nosniff

Connection: close

Content-Length: 84

Content-Type: text/html

<html><head><title>429 Rate Limited</title></head><body>Rate limit
exceeded</body>
```

The **User-Agent** header works similarly to the **Host** example provided above.

In the **URL Based Limiting** section, you can configure the following options for a specific URL-based limiting rule:

- **Name:** The name of the new request limit. This must be unique, alpha-numeric (underscores are also allowed) and it must not start with a number.
- **Limit:** Limit the number of attempts (per second) to a specific request/URL. Valid values range from 0 - 1000000. Setting the value to 0 disables the rule (but does not delete it). This can be useful when testing, for example, if a rule has a limit of 0 it does not incur an performance impact on the system.
- **Match:** The request field/URL to match. This drop-down list contains the following values:
  - Request URL
  - Host
  - User Agent
  - !Request URL
  - !Host
  - !User Agent
- **Match String:** The pattern (regular expression) to use to match the request field/URL.

When processing HTTP traffic (non-HTTP traffic is not affected), the URL is matched against the set of rules that contain regular expressions. Each rule has a limit associated with it. If the number of requests per second exceeds the specified limit, the request is blocked and the connection is closed (an error response can be sent if an appropriate selection is made in the **Error Responses** drop-down list).

If a specific request could match more than one rule, the limit is applied to the first rule that matches in the list. You can change the order of the rules using the **Move** option.

You can also modify or delete any existing rules.

Per-Virtual Service Limiting

Per-Virtual Service Limiting

In addition to being able to configure global and client-based limits, you can also configure limits at the Virtual Service and SubVS level. To access the settings, expand the **QoS/Limiting** section on the Virtual Service or SubVS modify screen (**Virtual Services > View/Modify Services > Modify**).

The fields you can configure are as follows:

- **Connections per second:** Set the maximum connections per second of this Virtual Service. The maximum is 1000000. Setting this value to 0 removes any limit.
- **HTTP Requests per second:** Set the maximum HTTP requests per second of this Virtual Service. The maximum is 1000000. Setting this value to 0 removes any limit.

  Note: For Virtual Services created on 443 port, the **HTTP requests per second** option is available only when the **SSL Acceleration** option is enabled.

- **Concurrent Connections:** Set the maximum concurrent connections for this Virtual Service. The maximum is 100000000. Setting this value to 0 removes any limit.
- **Bandwidth Limit (Kilobits/sec):** Set the maximum bandwidth of this Virtual Service. The value is in Kilobits/second. The minimum value is 16. The maximum value is 99999999. Setting this value to 0 removes any bandwidth limit. This limits all traffic going through the Virtual Service. If a bandwidth limit is set for a Virtual Service, then it is forced to be a Layer7 (L7) service.

Note: Setting the limit to **0** removes the limit for that particular rule. However, the rule remains on the LoadMaster as disabled and the LoadMaster ignores the rule. If you want, you can delete a rule by clicking **Delete**. However, if you want to use the same rule in the future you can set the limit to **0** to disable it temporarily and when required you can change the value.

You cannot set the Virtual Service or SubVS limits to higher limits than the global limits set in **System Configuration > QoS/Limiting**.

SubVSs have two additional limit fields in the **Basic Properties** section of the SubVS modify screen (**Virtual Services > View/Modify Services > Modify > SubVSs > Modify**):

- **SubVS Limit:** The maximum number of connections that can be sent to this SubVS before it is taken out

of rotation from the main Virtual Service. The maximum limit is 1000000.

- **SubVS Rate Limit:** The maximum number of connections per second that can be sent to this SubVS before it is taken out of rotation from the main Virtual Service. The maximum limit is 1000000.

Note: The lowest of the possible limits (global, client, Virtual Service, and SubVS) that is reached first is enforced. Note that the global limit is for all Virtual Services, the Virtual Service limit is for the current Virtual Service with multiple clients, and the client limit can be applicable for all clients, multiple clients, or for a single client - depending on the configuration. Client limits are enforced regardless of what Virtual Service is in use.

Client Limit Statistics

Client Limit Statistics

You can view statistics relating to the client limits by going to **Statistics > Real Time Statistics > Client Limits**. Statistics are updated every 10 seconds. The **Client Limits** button is only displayed if there is at least one client limit enabled in the **System Configuration > QoS/Limiting** screen. Statistics are only generated if the **Generate Limiter Statistics** check box is enabled in **System Configuration > QoS/Limiting**. There are buttons on the right of the **Client Limits** statistics screen where you can select different pages for **Connections/s**, **HTTP Requests/s**, **Total Connections**, and **Bandwidth Usage**.

Note: These buttons are only displayed if the corresponding client limits are set in **System Configuration > QoS/Limiting**.

The top 10 clients are displayed for the **Last 30 seconds**, **Last 5 minutes**, and **Last 30 minutes**. There are separate columns to show the number of **Ok** and **Blocked** connections. Based on these insights, you can configure specific rate controls for specific client IP addresses.

Virtual Service Limiting Statistics

Virtual Service Limiting Statistics

You can see limiting statistics for a Virtual Service by going to **Statistics > Real Time Statistics > Virtual Services** and clicking the **IP address:port** link for the relevant Virtual Service. The following limiting statistics are available:

- Conns/Sec Blocked
- Req/Sec Blocked
- MaxConns Blocked

3

Rate Limiting Log Information

Logging is off by default. To enable logging, select the **Generate Limiter Statistics** check box in **System Configuration > QoS/Limiting > Limiter Options**. When this is enabled, a log is generated every five seconds. The following information is recorded in the logs:

- **curconns:** The total number of active connections on the system (at this specific moment in time).
- **totconns:** The total number of attempted connections (of all time). If you reset the statistics this value is cleared.
- **totreqs:** The total number of successful requests (of all time).
- **totrulereq:** The total number of requests that have matched a rule (of all time).
- **totcblocked:** The total number of connections that have been blocked (of all time). This is client connections (no HTTP processing).
- **totrblocked:** The total number of HTTP requests that have been blocked (of all time), which were not blocked by the client connection blocking.
- **totruleblock:** The total number of URL rules that have been blocked (of all time), which were not blocked by the client connection blocking and not blocked by HTTP request blocking.
- **totmaxcblocked:** Total cumulative number of client requests blocked due to the maximum number of connections being exceeded.

Note: All counters are 64-bit

When the bandwidth limit is exceeded, logs similar to the ones below are generated:

```
2020-11-03T10:08:05+00:00 lb100 kernel: L7: Bandwidth limit exceeded in the
last 10 seconds

2020-11-03T10:08:35+00:00 lb100 kernel: L7: Bandwidth limit exceeded in the
last 60 seconds
```

4

RESTful API Details

This section contains details about the limiting API commands and parameters. You can retrieve or configure each of these parameters using the **get** or **set** RESTful API commands.

Here is an example of a **get** command to retrieve the **MaxConnsLimit** parameter value:

```
/access/get?param=MaxConnsLimit
```

Here is an example of a set command to set the **MaxConnsLimit** to **80000**:

```
/access/set?param=MaxConnsLimit&value=80000
```

The following limiting parameters can be retrieved or configured using the **get** or **set** commands:

- **MaxConnsLimit:** The maximum number of simultaneous connections (TCP and UDP).
- **MaxCPSLimit:** The global connection limit (per second).
- **MaxRPSLimit:** The global request limit (per second).
- **MaxBandwidthLimit:** The global bandwidth limit (kilobits per second)
- **SendRateLimitError:** This parameter accepts the following values:
  - 0 - no error response (the connection is simply dropped)
  - 1 - **Send 429 Too Many Requests** error response
  - 2 - **Send 503 Service Unavailable** error response
- **RateLimitFail:** Fail on rate limit. This parameter accepts the following values:
  - 0 - disabled (the LoadMaster attempts to select a different RS or SubVS to use for the connection)

  - 1 - enabled (forces an error)
- **LimitLogging:** Generate a summary log entry every 5 seconds. This parameter accepts the following values:
  - 0 - disabled
  - 1 - enabled
- **ClientRepeatDelay:** Set the minimum time after a client is no longer limited before a new message is generated. If a client generates a message and continues to be blocked for continuously hitting the limit, no new message is generated. Only if the client goes quiet for the delay period will a new message be generated. Valid values range from 10 - 86400 seconds.
- **ClientMaxConnsLimit:** This limits the default maximum number of concurrent connection attempts (per second) from a specific host. Setting the limit to 0 disables this option. Valid values range from 0 - 1000000.
- **ClientCPSLimit:** The global client connection limit.
- **ClientRPSLimit:** The global client request limit.
- **ClientMaxBandwidthLimit:** The global client maximum bandwidth limit.

For further details on the RESTful API in general, refer to the RESTful API documentation.

Related Links

For details, see the following topics:

- Maximum Client Concurrent Connection Limit
- URL-Based Limiting Rules
- Statistics

Maximum Client Concurrent Connection Limit

Maximum Client Concurrent Connection Limit

To list the existing client concurrent connection limits, run the **clientmaxclimitlist** command, for example:

```
/access/clientmaxclimitlist
```

To add a new client concurrent connection limit, run the **clientmaxclimitadd** command, for example:

```
/access/clientmaxclimitadd?l7addr=<Address>&l7limit=<Limit>
```

To delete an existing client concurrent connection limit, run the **clientmaxclimitdel** command, for example:

```
/access/clientmaxclimitdel?l7addr=<Address>
```

Client CPS Limit

To list the existing Connections Per Second (CPS) limits, run the **clientcpslimitlist** command, for example:

```
/access/clientcpslimitlist
```

To add a new CPS limit, run the **clientcpslimitadd** command, for example:

```
/access/clientcpslimitadd?l7addr=<Address>&l7limit=<Limit>
```

To delete an existing CPS limit, run the **clientcpslimitdel** command, for example:

```
/access/clientcpslimitdel?l7addr=<Address>
```

Legacy Client CPS Limit Commands

Before the rate limiting functionality was improved in LoadMaster version 7.2.52, you could run the following commands relating to client CPS limits.

Note: Although these commands still work - we recommend using the new commands outlined in the **Client CPS Limit** section above.

Client limiting can be used to limit the default maximum number of connection attempts (per second) from a specified host. The limit can be set using the **set** parameter, for example:

```
https://<DeviceIPAddress>/access/set?param=l7limitinput&value=25
```

Setting the limit to zero disables the option.

A number of addresses or networks can be specified to be limited. To add an address, run the command below:

```
https://<DeviceIPAddress>/access/
afeclientlimitadd?l7addr=<L7Address>&l7limit=<L7Limit>
```

To delete an address, run the command below:

```
https://<DeviceIPAddress>/access/afeclientlimitdel?l7addr=<L7Address>
```

To list the addresses and their limits, run the command below:

```
https://<DeviceIPAddress>/access/afeclientlimitlist?
```

Client RPS Limit

To list the existing RPS limits, run the **clientrpslimitlist** command, for example:

```
/access/clientrpslimitlist
```

To add a new RPS limit, run the **clientrpslimitadd** command, for example:

```
/access/clientrpslimitadd?l7addr=<Address>&l7limit=<Limit>
```

To delete an existing RPS limit, run the **clientrpslimitdel** command, for example:

```
/access/clientrpslimitdel?l7addr=<Address>
```

Client Bandwidth Limit

To list the existing client bandwidth limits, run the **clientbandwidthlimitlist** command, for example:

```
/access/clientbandwidthlimitlist
```

To add a new client bandwidth limit, run the **clientbandwidthlimitadd** command, for example:

```
/access/clientbandwidthlimitadd?l7addr=<Address>&l7limit=<Limit>
```

Note: The global client maximum bandwidth limit (**ClientMaxBandwidthLimit**) must be set before you add a specific client bandwidth limit.

To delete an existing client bandwidth limit, run the **clientbandwidthlimitdel** command, for example:

```
/access/clientbandwidthlimitdel?l7addr=<Address>
```

Per-Virtual Service Limits

To add a new Virtual Service with limits, run the **addvs** command, for example:

```
/access/addvs?vs=<VirtualServiceAddress>&port=<VirtualServicePort>&prot=<tcp/
udp>&ConnsPerSecLimit=<Limit>&RequestsPerSecLimit=<Limit>&MaxConnsLimit=<Limit>
&bandwidth=<Limit>
```

To modify the limits for an existing Virtual Service, run the **modvs** command, for example:

```
/access/modvs?vs=<VirtualServiceAddress>&port=<VirtualServicePort>&prot=<tcp/
udp>&ConnsPerSecLimit=<Limit>&RequestsPerSecLimit=<Limit>&MaxConnsLimit=<Limit>
&bandwidth=<Limit>
```

To modify the limits for an existing SubVS, run the **modvs** command, for example:

```
/access/
modvs?vs=<SubVSIndex>&ConnsPerSecLimit=<Limit>&RequestsPerSecLimit=<Limit>&MaxC
onnsLimit=<Limit>&bandwidth=<Limit>
```

You can retrieve the SubVS **Index** by running the **listvs** command or by checking the **Id** at the top of the SubVS modify screen.

URL-Based Limiting Rules

URL-Based Limiting Rules

You can list the existing URL-based limiting rules by running the **listlimitrules** command, for example:

```
/access/listlimitrules
```

You can add a new URL-based limiting rule by running the **addlimitrule** command, for example:

```
/access/addlimitrule?name=ExampleRule&pattern=/test/a.html&limit=5&match=0
```

You can modify an existing URL-based limiting rule by running the **modlimitrule** command, for example:

```
/access/modlimitrule?name=ExampleRule&pattern=/test/b.html&limit=5&match=0
```

Valid values for the **match** parameter are as follows:

- 0 - Request
- 1 - Host
- 2 - User Agent
- 64 - !Request
- 65 - !Host
- 66 - !UserAgent

Note: The values with an exclamation mark (**!**) before them matches the inverse, for example, not a specific request or not a specific user agent.

You can delete an existing URL-based limiting rule by running the **dellimitrule** command, for example:

```
/access/dellimitrule?name=ExampleRule
```

You can move the position of an existing URL-based limiting rule by running the **movelimitrule** command, for example:

```
/access/movelimitrule?name=ExampleRule3&position=1
```

Setting the **position** parameter to a value larger than the size of the list will move the rule to the end of the list.

Statistics

Statistics

Statistics

Global and Virtual Service statistic information (including some related to limiting) is available in the **stats** RESTful API command:

```
/access/stats
```