Rate Limiting

Feature Description

UPDATED: 12 October 2020

Copyright Notices

Table of Contents

1 Introduction

In LoadMaster 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 limitingto 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

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 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**.

2.1 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.

  > 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.

  > 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.

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

2.2 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.
- **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.

2.3 Client Limiting

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

2.3.1 Maximum Client Concurrent Connection Limit

In the **Maximum Client Concurrent Connection Limit** section, you can configure the **Client Concurrent Connection Limit**. 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.

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.

> 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.

> 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.

2.3.2 Client CPS Limit

In the **Client CPS Limit** section, you can configure the **Client Connection Limit**. This 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.

> 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.

> 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.

2.3.3 Client RPS Limit

In the **Client RPS Limit** section, you can configure the **Client HTTP Request Limit**. This 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.

> 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.

> 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.

2.3.4 URL Based Limiting

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**

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

2.4 Client Limit Statistics

You can view statistics relating to the client limits by going to **Statistics > Real Time Statistics > Client Limits**. The **Client Limits** button is only displayed if there is at least one client limit enabled in the **QoS/Limiting** screen. There are three buttons on the right where you can select different pages for **Connections/s**, **HTTP Requests/s**, and **Total Connections**. 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.

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.

All counters are 64-bit

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 **TotalConnectionLimit** parameter value:

```
/access/get?param=TotalConnectionLimit
```

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

```
/access/set?param=TotalConnectionLimit&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).
- **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.

The global statistic information is also available in the **stats** RESTful API command:

`/access/stats`

For further details on the RESTful API in general, refer to the Long Term Support (LTS) RESTful API Interface Description.

For PowerShell help, run the **Get-Help** command for the relevant commands.

4.1 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>`

4.2 Client CPS Limit

To list the existing 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>`

4.3 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>

4.4 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

> 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.

Last Updated Date

This document was last updated on 12 October 2020.