Limiters

https://docs.netgate.com/pfsense/en/latest/trafficshaper/limiters.html

Limiters are an alternate method of traffic shaping. Limiters use dummynet(4) to enact bandwidth limits and perform other prioritization tasks, and they do not rely on ALTQ. Limiters are currently the only way to achieve per-IP address or per-network bandwidth rate limiting using pfSense® software. Limiters are also used internally by Captive Portal for per-user bandwidth limits.

Limiters are managed at **Firewall > Traffic Shaper** on the **Limiters** tab.

Like HFSC and CBQ, Limiters may be nested with queues inside other queues. Root-level limiters (Also called Pipes), may have bandwidth limits and delays, while child limiters (Also called queues), may have priorities (Also called weights). Bandwidth limits can be optionally masked by either the source or destination IP address, so that the limits can be applied on a per-IP address or network basis instead of as a general group.

Limiters are nearly always used in pairs: One for incoming traffic and one for outgoing traffic.

According to its man page the dummynet(4) system was originally designed as a means to test TCP congestion control and it grew up from there. Due to this purpose, a unique feature of limiters is that they can be used to induce artificial packet loss and delay into network traffic. That is primarily used in troubleshooting and testing (or being evil and playing a prank on someone), and not often found in production.

Uses for Limiters

The primary use for limiters is to apply bandwidth limits for users or specific protocols, e.g. “Maximum of 1Mbit/s for SMTP”, or “Joe’s PC only can use 5Mbit/s”. Limiters can apply a per-IP address or per-network limit, such as “All Users in 192.168.50.0/24 can use a maximum of 3Mbit/s each” or “The guest network and public network can use 1Mbit/s for each segment”.

Limiters are the only type of shaper available in pfSense software which is capable of oversubscription in this manner. The ALTQ shaper requires all child queues to sum up to no more than the speed of the parent queue, but masked limiters allow a set limit to as many IP addresses as can be funneled through the limiter by firewall rules.

Conceptually, consider a limiter as a bucket of bandwidth. All traffic flowing through an unmasked limiter draws bandwidth from the same bucket. Masking a limiter effectively sets up multiple buckets of the same size, one per masked group. Whether that is a single host or an entire network

depends on the mask value.

Limiters can also allow for reserved bandwidth by limiting everything *except* a specific protocol which can then consume all remaining bandwidth. In this type of setup on a 10Mbit/s link the firewall would pass traffic from, for example, a SIP server with no limiter. Then the firewall would use a pass rule for all other traffic with a limit of 8Mbit/s. This would let the SIP server use all of the bandwidth it wanted, but it would always have a minimum of 2Mbit/s to itself.

Limiters can also help with issues such as Bufferbloat by controlling the delay of certain packets, using the CoDel algorithm similar to the one available in ALTQ (CoDel Active Queue Management).

See also

- Configuring CoDel Limiters for Bufferbloat

How Limiters Work

Limiters, like ALTQ, hold traffic to a certain point by dropping or delaying packets to achieve a specific line rate. Usually taking advantage of built-in mechanisms from protocols that detect the loss and back off to a sustainable speed.

In situations where packets are queued under the same parent pipe, the firewall considers their weights when ordering the packets before it sends them. Unlike priorities in CBQ and PRIQ, the weight of a queue in a limiter will never starve it for bandwidth.

Limiters and IPv6

Limiters work with IPv6, though it requires separate IPv4 and IPv6 rules to apply limiters properly.

Limitations

Limiter pipes do not have a concept of borrowing bandwidth from other pipes. A limit is always a hard upper limit.

Limiters use `dummynet` pipes, so there will be additional (though small) overhead from the extra packet processing involved.

Limiters cannot effectively guarantee a minimum bandwidth amount for a pipe or queue, only a maximum.

Child queues cannot have bandwidth values, so a pipe cannot be split into smaller pipes by queues. Child queues can only use weights to prioritize packets inside a pipe.

The overhead from delaying and queuing packets can cause increased mbuf usage. For more information on increasing the amount of available mbufs, see Hardware Tuning and Troubleshooting.

Limiters and Multi-WAN

When using limiters with Multi-WAN, limits for non-default gateways must be applied using floating rules set for the *out* direction and configured with the appropriate gateway.

Creating Limiters

Limiters are managed under **Firewall > Traffic Shaper** on the **Limiters** tab.

To create a new root-level limiter (pipe), click fa-plus **New Limiter**.

To create a child limiter (queue), click an existing limiter under which it can be created, and click fa-plus **Add New Queue**.

Tip

In nearly all cases, limiters exist in pairs at the same level (e.g. two pipes, or two queues): One for inbound traffic and one for outbound traffic. When creating new limiters or queues, create one for each direction.

Enable

Check the box to enable this limiter. If the limiter is disabled, it will not be available for use by firewall rules.

Name

This defines the name of the limiter, as it will appear for selection on firewall rules.

The name must be alphanumeric, and may also include `-` and `_`.

Tip

When choosing a name, avoid using `In` and `Out` since the same limiter, if used on both WAN and LAN, would be used in the *In* direction on one interface and the *Out* direction on another. The best practice is to use `Down` or `Download` and `Up` or `Upload`.

Bandwidth (Pipes)

This section defines a bandwidth value for the pipe, or multiple bandwidths if schedules are involved. This option does not appear when editing a child limiter (queue).

Bandwidth

The numerical part of the bandwidth for the pipe, e.g. `3` or `500`.

Bw Type

The units for the **Bandwidth** field, such as *Mbit/s*, *Kbit/s*, or *Bit/s*.

Schedule

If the firewall has schedules defined (Time Based Rules), the firewall offers them in this list. When schedules are in use by the firewall, the limiter can have a bandwidth value for each potential schedule. Define these by clicking fa-plus **Add Schedule** to add another bandwidth definition.

If a limiter contains multiple bandwidth specifications, they must each use a different schedule. For example if the firewall has a “Work Day” schedule, then it must also have an “Off Hours” schedule that contains all of the time not included in “Work Day” for the second bandwidth specification.

Mask

This drop-down list controls how the limiter will mask addresses in the pipe or queue.

None

When set to *none*, the limiter does not perform any masking. The pipe bandwidth will be applied to all traffic as a whole.

Source / Destination address

When a limiter is set for *Source Address* or *Destination Address*, the pipe bandwidth limit will be applied on a per-IP address basis or a subnet basis, depending on the masking bits, using the direction chosen in the masking.

In general, a limiter should mask the **Source Address** on **Upload** (In) limiters for LAN-type interfaces, and **Destination Address** on **Download** (Out) limiters on LAN-type interfaces. Similar to swapping the directionality of the limiters when applying to LAN and WAN, masking is swapped as well, so the same masked limiter set for **In** on LAN should be used for **Out** on WAN.

Mask Bits

There are separate boxes to control the address masking for IPv4 and IPv6. For IPv4 a value of *32* for **IPv4 mask bits** sets up a per-IPv4 address limit, which is the most common usage. For a per-IPv6-address limit, use *128* as the **IPv6 mask bits** value.

To create per-subnet or similar masks, enter the subnet bits in the appropriate field for either IPv4 or IPv6 mask bits, such as *24* to limit IPv4 in groups of /24 subnets.

Description

An optional bit of text to explain the purpose for this Limiter.

Advanced Options

Additional options that vary when editing a pipe or a queue.

Delay (Pipes)

The **Delay** option is only found on limiter pipes. It introduces an artificial delay (latency), specified in milliseconds, into the transmission of any packets in the limiter pipe. This is typically left blank so that packets are transmitted as fast as possible by the firewall. This can be used to simulate high-latency connections such as satellite uplinks for lab testing.

Weight (Queues)

The **Weight** option is only found on child limiters (queues). This value can range from 1 to 100. Higher values give more precedence to packets in a given queue. Unlike PRIQ and CBQ priorities, a lowly-weighted queue is not in danger of being starved of bandwidth by the firewall.

Packet loss rate

Another method of artificially degrading traffic. The **Packet Loss Rate** can be configured to drop a certain fraction of packets that enter the limiter. The value is expressed as a decimal representation of a percentage, so `0.01` is 1%, or one packet out of a hundred dropped. This field is typically left empty so every packet is delivered by the firewall.

Queue Size

Sets the size of the queue, specified in queue slots, used for handling queuing delay. Left blank, it defaults to `50` slots, which is the recommended value. Slow speed links may need a lower queue size to operate efficiently. High speed links may need more slots.

Tip

In cases where there are several limiters or limiters with large **Queue Size** values, a **System Tunable** may need set to increase the value of `net.inet.ip.dummynet.pipe_slot_limit` above the total number of configured queue lots among all pipes and queues.

Bucket Size

The **Bucket Size**, also specified in slots, sets the size of the hash table used for queue storage. The default value is `64`. It must be a numeric value between `16` and `65536`, inclusive. This value is typically left blank.

See also

For more information about these values, consult the ipfw(8) man page, in the section titled “Traffic Shaper (Dummynet) Configuration”. Though current versions of pfSense software utilize dummynet through `pf` instead of `ipfw`, the configuration options are the same.

Assigning and Using Limiters

Limiters are assigned using firewall rules via the **In/Out Pipe** selectors under **Advanced Options**. Any potential matching criteria that a firewall rule supports can assign traffic to a limiter.

The most important thing to remember when assigning a limiter to a rule is that the **In** and **Out** fields are designated **from the perspective of the firewall itself**.

For example, in a firewall configuration with a single LAN and single WAN, inbound traffic on a LAN interface is leaving toward the Internet, i.e. *uploaded* data. Outbound traffic on the LAN interface is going toward the client PC, i.e. *downloaded* data. On the WAN interface the directionality is reversed; Inbound traffic is coming from the Internet to the client (download), and outbound traffic is going from the client to the Internet (upload).

In most cases, a firewall rule will have both an **In** limiter and **Out** limiter, but only the **In** limiter is required by the firewall to limit traffic in a single direction.

Limiters may be applied on normal interface rules, or on floating rules. On floating in the *out* direction, the In/Out selections are flipped conceptually.

Checking Limiter Usage

Information about active limiters may be found under **Diagnostics > Limiter Info**. Here, each limiter and child queue is shown in text format.

The set bandwidth and parameters for each limiter are displayed by the page, along with the current traffic level moving inside the limiter. In the case of masked limiters, the firewall displays the bandwidth of each IP address or masked group.

Revision #2
Created 23 December 2023 15:30:37 by ColtM
Updated 7 August 2024 23:24:39 by ColtM