[Linux FAQ / Howtos](#)

tc: Linux HTTP Outgoing Traffic Shaping (Port 80 Traffic Shaping)

by Vivek Gite on April 13, 2010 · [2 comments](#)

I've 10Mbps server port dedicated to our small business server. The server also act as a backup DNS server and I'd like to slow down outbound traffic on port 80. How do I limit bandwidth allocation to http service 5Mbps (burst to 8Mbps) at peak times so that DNS and other service will not go down due to heavy activity under Linux operating systems?
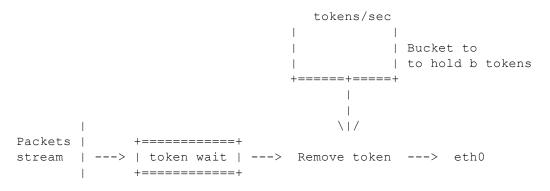
You need use the tc command which can slow down traffic for given port and services on servers and it is called traffic shaping:

1. When traffic is shaped, its rate of transmission is under control, in other words you apply some sort of bandwidth allocation for each port or or so called Linux services. Shaping occurs on egress.
2. You can only apply traffic shaping to **outgoing or forwarding** traffic i.e. you do not have any control for incoming traffic to server. However, tc can do policing controls for arriving traffic. Policing thus occurs on ingress. This FAQ only deals with traffic shaping.

Token Bucket (TB)

[A token bucket](#) is nothing but a common algorithm used to control the amount of data that is injected into a network, allowing for bursts of data to be sent. It is used for network traffic shaping or rate limiting. With token bucket you can define the maximum rate of traffic allowed on an interface at a given moment in time.

```
                              tokens/sec
                         |           |
                         |           | Bucket to
                         |           | to hold b tokens
                         +======+=====+
                              |
                              |
         |                   \|/
Packets  |      +===========+
stream   | ---> | token wait | --->  Remove token  --->  eth0
         |      +===========+
```

1. The TB filter puts tokens into the bucket at a certain rate.

2. Each token is permission for the source to send a specific number of bits into the network.
3. Bucket can hold b tokens as per shaping rules.
4. Kernel can send packet if you've a token else traffic need to wait.

How Do I Use tc command?

> ⚠️ **WARNING!** These examples requires good understanding of TCP/IP and other networking concepts. All new user should try out examples in test environment.

tc command is by default installed on my Linux distributions. To list existing rules, enter:

```
# tc -s qdisc ls dev eth0
```

Sample outputs:

```
qdisc pfifo_fast 0: root bands 3 priomap  1 2 2 2 1 2 0 0 1 1 1 1 1 1 1 1
 Sent 2732108 bytes 10732 pkt (dropped 0, overlimits 0 requeues 0)
 rate 0bit 0pps backlog 0b 0p requeues 0
```

Your First Traffic Shaping Rule

First, send ping request to cyberciti.biz from your Local Linux workstation and note down ping time, enter:

```
# ping cyberciti.biz
```

Sample outputs:

```
PING cyberciti.biz (74.86.48.99) 56(84) bytes of data.
64 bytes from txvip1.simplyguide.org (74.86.48.99): icmp_seq=1 ttl=47 time=304 ms
64 bytes from txvip1.simplyguide.org (74.86.48.99): icmp_seq=2 ttl=47 time=304 ms
64 bytes from txvip1.simplyguide.org (74.86.48.99): icmp_seq=3 ttl=47 time=304 ms
64 bytes from txvip1.simplyguide.org (74.86.48.99): icmp_seq=4 ttl=47 time=304 ms
64 bytes from txvip1.simplyguide.org (74.86.48.99): icmp_seq=5 ttl=47 time=304 ms
64 bytes from txvip1.simplyguide.org (74.86.48.99): icmp_seq=6 ttl=47 time=304 ms
```

Type the following tc command to slow down traffic by 200 ms:

```
# tc qdisc add dev eth0 root netem delay 200ms
```

Now, send ping requests again:

```
# ping cyberciti.biz
```

Sample outputs:

```
PING cyberciti.biz (74.86.48.99) 56(84) bytes of data.
64 bytes from txvip1.simplyguide.org (74.86.48.99): icmp_seq=1 ttl=47 time=505 ms
64 bytes from txvip1.simplyguide.org (74.86.48.99): icmp_seq=2 ttl=47 time=505 ms
64 bytes from txvip1.simplyguide.org (74.86.48.99): icmp_seq=3 ttl=47 time=505 ms
64 bytes from txvip1.simplyguide.org (74.86.48.99): icmp_seq=4 ttl=47 time=505 ms
64 bytes from txvip1.simplyguide.org (74.86.48.99): icmp_seq=5 ttl=47 time=505 ms
64 bytes from txvip1.simplyguide.org (74.86.48.99): icmp_seq=6 ttl=47 time=505 ms
64 bytes from txvip1.simplyguide.org (74.86.48.99): icmp_seq=7 ttl=47 time=505 ms
64 bytes from txvip1.simplyguide.org (74.86.48.99): icmp_seq=8 ttl=47 time=505 ms
^C
--- cyberciti.biz ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7006ms
rtt min/avg/max/mdev = 504.464/505.303/506.308/0.949 ms
```

To list current rules, enter:

```
# tc -s qdisc ls dev eth0
```

Sample outputs:

```
qdisc netem 8001: root limit 1000 delay 200.0ms
```

```
 Sent 175545 bytes 540 pkt (dropped 0, overlimits 0 requeues 0)
 rate 0bit 0pps backlog 0b 0p requeues 0
```

To delete all rules, enter:

```
# tc qdisc del dev eth0 root
# tc -s qdisc ls dev eth0
```

TBF Example

To attach a TBF with a sustained maximum rate of 1mbit/s, a peakrate of 2.0mbit/s, a 10kilobyte buffer, with a pre-bucket queue size limit calculated so the TBF causes at most 70ms of latency, with perfect peakrate behavior, enter:

```
# tc qdisc add dev eth0 root tbf rate 1mbit burst 10kb latency 70ms peakrate 2mbit
minburst 1540
```

HTB - Hierarchy Token Bucket

To control the use of the outbound bandwidth on a given link use HTB:

1. **rate** - You can set the allowed bandwidth.
2. **ceil** - You can set burst bandwidth allowed when buckets are present.
3. **prio** - You can set priority for additional bandwidth. So classes with lower prios are offered the bandwidth first. For example, you can give lower prio for DNS traffic and higher for HTTP downloads.
4. **iptables and tc**: You need to use iptables and tc as follows to control outbound HTTP traffic.

Example: HTTP Outbound Traffic Shaping

First , delete existing rules for eth1:

```
# /sbin/tc qdisc del dev eth1 root
```

Turn on queuing discipline, enter:

```
# /sbin/tc qdisc add dev eth1 root handle 1:0 htb default 10
```

Define a class with limitations i.e. set the allowed bandwidth to 512 Kilobytes and burst bandwidth to 640 Kilobytes for port 80:

```
# /sbin/tc class add dev eth1 parent 1:0 classid 1:10 htb rate 512kbps ceil 640kbps prio 0
```

Please note that port 80 is NOT defined anywhere in above class. You will use iptables mangle rule as follows:

```
# /sbin/iptables -A OUTPUT -t mangle -p tcp --sport 80 -j MARK --set-mark 10
```

To save your iptables rules, enter (RHEL specific command):

```
# /sbin/service iptables save
```

Finally, assign it to appropriate qdisc:

```
# tc filter add dev eth1 parent 1:0 prio 0 protocol ip handle 10 fw flowid 1:10
```

Here is another example for port 80 and 22:

```
/sbin/tc qdisc add dev eth0 root handle 1: htb
/sbin/tc class add dev eth0 parent 1: classid 1:1 htb rate 1024kbps
/sbin/tc class add dev eth0 parent 1:1 classid 1:5 htb rate 512kbps ceil 640kbps prio 1
/sbin/tc class add dev eth0 parent 1:1 classid 1:6 htb rate 100kbps ceil 160kbps prio 0
/sbin/tc filter add dev eth0 parent 1:0 prio 1 protocol ip handle 5 fw flowid 1:5
/sbin/tc filter add dev eth0 parent 1:0 prio 0 protocol ip handle 6 fw flowid 1:6
/sbin/iptables -A OUTPUT -t mangle -p tcp --sport 80 -j MARK --set-mark 5
/sbin/iptables -A OUTPUT -t mangle -p tcp --sport 22 -j MARK --set-mark 6
```

How Do I Monitor And Test Speed On Sever?

Use the following tools

```
# /sbin/tc -s -d class show dev eth0
# /sbin/iptables -t mangle -n -v -L
# iptraf
# watch /sbin/tc -s -d class show dev eth0
```

To test download speed use lftp or wget command line tools.

References:

- Read tc, tc-tbf and tc-htb command man pages.
- Linux Advanced Routing & Traffic Control

Featured Articles:

- 20 Linux System Monitoring Tools Every SysAdmin Should Know
- 20 Linux Server Hardening Security Tips
- My 10 UNIX Command Line Mistakes
- The Novice Guide To Buying A Linux Laptop   new
- 10 Greatest Open Source Software Of 2009
- Top 5 Email Client For Linux, Mac OS X, and Windows Users
- Top 20 OpenSSH Server Best Security Practices
- Top 10 Open Source Web-Based Project Management Software
- Top 5 Linux Video Editor Software

{ 2 comments… read them below or add one }

1 Catalin(ux) M. BOIE April 15, 2010

> For nice ncurses display of classes, you can use lqdump from lqkit (I am the author).
> RPM, tgz, source: http://kernel.embedromix.ro/us/
> Screenshot: http://kernel.embedromix.ro/us/lqkit/ss/lqdump_ss1.png
>
> Reply

2 Guy Egozy November 4, 2010

> For simplicity sake you may also want to check out trickle. I've successfully used it in bash scripts for upload

capping but it should also work for services.

Leave a Comment

[   ] Name *

[   ] E-mail *

[   ] Website

You can use these HTML tags and attributes for your code and commands: <strong> <em>   <u> <blockquote> <pre> <a href="" title="">

☐ Notify me of followup comments via e-mail

[   ]

Tagged as: Arch Linux Shape Traffic, bandwidth allocation, CentOS Linux Shape Traffic, Debian Linux Shape Traffic, Fedora Linux Shape Traffic, Linux shape port 80 traffic, Linux tc throttle http, Linux tc throttle port 80 traffic, maximum rate, OpenSuse Linux Shape Traffic, Redhat Linux Shape Traffic, RHEL Linux Shape Traffic, Slackware Linux Shape Traffic, Suse Linux Shape Traffic, tc command, token bucket, Ubuntu Linux Shape Traffic

Previous post: Linux: Set OR Change The Library Path

Next post: UNIX / Linux: PGP TarBall File Signature Keys Verification

Sign up for our daily email newsletter:

[Enter your email address] [   ]

- ## Related FAQs

    - [tcpdump: Monitor ALL eth1 Traffic Except My Own SSH Session](#)
    - [How To Run Linux Web Server / Service on Private IP Network](#)
    - [Host a domain without CPanel or Plesk Control Panel](#)
    - [FreeBSD dnstop: Monitor and Display DNS Server Traffic On Your Network](#)
    - [How do I view my Linux or UNIX server bandwidth usage?](#)