

Traffic Shaping Precision Test

Sven Ubik

16.5.2001

1 Introduction

The goal of this test was to verify the precision of traffic shaping as implemented by Cisco routers used in our CESNET2 network and to check the influence of traffic shaping on the router load. The precision indicates how much we can rely on the shape of the processed traffic being compliant to the traffic shaping specification. Precision is an important parameter in large-scale diffserv-based networks with many aggregation points where incorrectly shaped traffic can cause congestions within the diffserv domain. The work was done as a part of the TF-NGN¹ project Support of Delay and Jitter Requirements².

Traffic shaping is specified by two parameters: CIR (Committed Access Rate) and Bc (Burst Committed). The shaped traffic can be sent at the maximum rate of CIR measured over any interval (called a shaping interval) equal to or greater than Bc/CIR. Within the shaping interval, the instantaneous output rate can be higher, possibly up to the output link rate. The input rate can be up to the input link rate (if it is not limited by another traffic conditioning mechanism) at any time. Packets are buffered until they can be sent in compliance with the traffic shaping specification provided that there is enough buffer space. Packets that arrived when the buffer is full are discarded. The buffer should accommodate at least the number of packets corresponding to Bc bits at any possible packet size. Cisco extends the basic traffic shaping specification by adding the Be (Extended Burst) parameter, which allows to sent out up to Be bits during the shaping interval under certain conditions. We disabled this functionality by setting Be=0.

The test was performed on a Fast Ethernet interface of the Cisco 7500 router. We used this particular type of router because it is commonly used in our network and because it provides sufficient computing power for advanced traffic processing such as traffic shaping. We also performed some parts of the test on the Cisco 2600 router with very similar results (except for performance characteristics).

¹<http://www.terena.nl/task-forces/tf-ngn>

²<http://www.cnaf.infn.it/ferrari/tfngn>

2 Measurement configuration

The measurement configuration is shown in Fig. 1. The Cisco 7500 router was equipped with an RSP8 processor, two Fast Ethernet ports on a FEIP2-DSW-2TX card and two Fast Ethernet modules on a VIP4-80 card. We used the RUDE³ program to generate an input stream of a specified rate and shape on WS1 and a companion program CRUDE to capture an output stream on WS2. Both workstations were running Debian Linux. RUDE/CRUDE are similar in operation to mgen/drec [mgen] utilities, but they are much more precise and can run at higher speeds. For each received packet, CRUDE records stream identifier, sequence number, departure and arrival times and packet length in a binary or text log file.

System clocks of both PCs were precisely synchronized using a PPS signal distributed from a GPS receiver to PC serial ports. Nanokernel [nanokernel] was used to obtain the system clock state with finer granularity. We developed a utility which takes as input the text log file created by CRUDE and produces as output data and command files for gnuplot [gnuplot] tool, which can be then used to plot graphs indicating lost packets, throughput, delay, jitter, delay distribution and jitter distribution. We presented a discussion of various factors affecting precision of time synchronization in the proposed system and a description of the utility for computing QoS characteristics in [qofis2001].

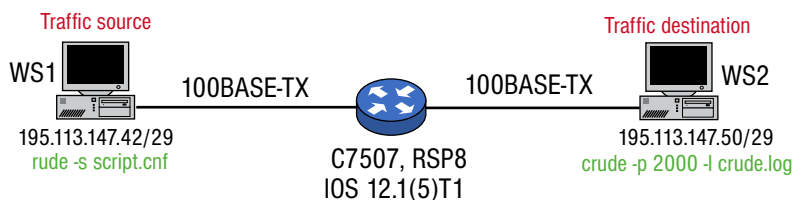


Figure 1: Test configuration

3 Router configuration

The principal part of the router software configuration is shown below:

```
class-map match-all class1
  match access-group 100
!
policy-map ds
  class class1
```

³<http://www.atm.tut.fi/rude/>

```

    shape average 5000000 500000 0
!
interface FastEthernet6/1/0
    service-policy output ds
!
access-list 100 permit ip host 195.113.147.42 any

```

4 Throughput measurement

We wanted to make sure that the processor could route the testing traffic streams so that the results were not distorted by processor saturation. We also wanted to see how traffic shaping increased the processor load. We used a set of constant-rate streams, each consisting of packets of a different length. The test was performed twice. First, the stream was sent between two ports of the FEIP2-DSW-2TX card. Second, the stream was sent from a port of the FEIP2-DSW-2TX card to a port of the VIP4-80 card. The measured throughput for packets of different lengths and corresponding processor load without traffic shaping is summarized in the following tables:

FEIP2-DSW-2TX -> FEIP2-DSW-2TX		
Packet size	Throughput	FEIP2-DSW-2TX load
1500 bytes	97.3 Mb/s	6 %
1024 bytes	96.2 Mb/s	59 %
512 bytes	92.3 Mb/s	88 %
256 bytes	86.3 Mb/s	99 %
128 bytes	66.0 Mb/s	99 %
64 bytes	32.9 Mb/s	99 %

Table 1: Throughput of forwarding to a port on FEIP2-DSW-2TX

FEIP2-DSW-2TX -> VIP4-80			
Packet size	Throughput	FEIP2-DSW-2TX load	VIP4-80 load
1500 bytes	97.3 Mb/s	35 %	7 %
1024 bytes	96.2 Mb/s	24 %	11 %
512 bytes	83.9 Mb/s	79 %	21 %
256 bytes	15.2 Mb/s	99 %	34 %
128 bytes	2.2 Mb/s	99 %	45 %
64 bytes	8.2 Mb/s	99 %	42 %

Table 2: Throughput of forwarding to a port on VIP4-80

4.1 Observations:

- Load of the RSP8 processor was negligible (indicated as 0%)
- VIP4-80 processor is more powerful than FEIP2-DSW-2TX
- Although the VIP4-80 processor is more powerful, the throughput decreases more rapidly with decreasing packet size when sending packets from a port of the FEIP2-DSW-2TX processor to a port of the VIP4-80 processor than when sending packets between the ports of the FEIP2-DSW-2TX processor
- For some reason, when sending from a port of the FEIP2-DSW-2TX processor to a port of the VIP4-80 processor, the throughput is lower and the VIP4-80 processor load is higher for 128-byte packets than for 64-byte packets
- We did not measure the throughput for sending packets from a port of the VIP4-80 processor to a port of the FEIP2-DSW-2TX processor and between two ports of the VIP4-80 processor, these tests should be done for more comprehensive overview

5 Input streams for traffic shaping

We used three different input streams shown in Figures 2 to 4: a constant-rate stream 5Mb/s (equal to CIR), a bursty stream 5Mb/s generated as a sequence of bursts with a steady rate of 10Mb/s for 0.1s followed by gaps with no packets sent for 0.1s and a bursty stream 5Mb/s generated as a sequence of bursts with a steady rate of 20Mb/s for 0.1s followed by gaps with no packets sent for 0.3s. For clarity, only the first 3 seconds of streams 2 and 3 are shown in Figures 3 and 4, respectively.

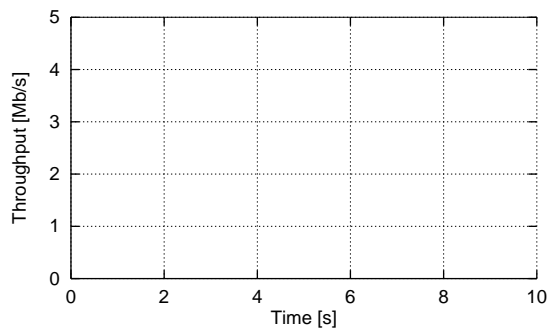


Figure 2: Input stream 1 - constant rate 5Mb/s

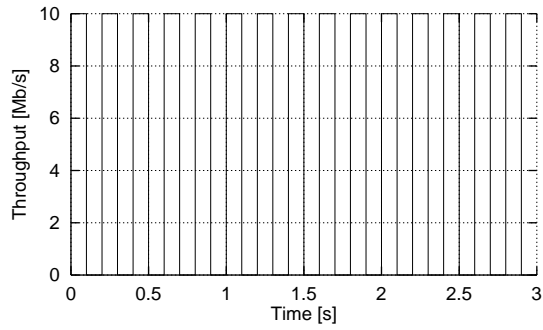


Figure 3: Input stream 2 - bursty 5Mb/s (10Mb/s for 0.1s, a gap for 0.1s)

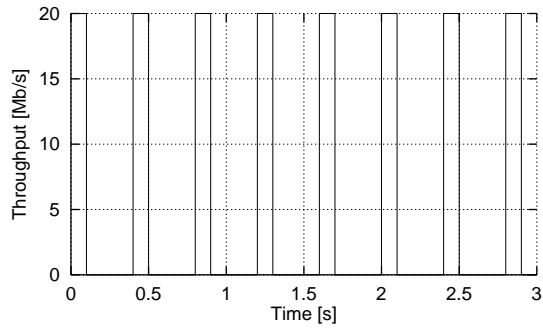


Figure 4: Input stream 3 - bursty 5Mb/s (20Mb/s for 0.1s, a gap for 0.3s)

We generated each of the three streams twice - in 1500-byte packets and in 256-byte packets. Therefore, there were actually six different streams.

6 Measurement results

We set the traffic shaping parameters as follows: CIR=5Mb/s, Bc=500kb or 20kb and Be=0. The first alternative of Bc=500kb corresponds to the shaping interval of 100ms, the second alternative of Bc=20kb corresponds to the shaping interval of 4ms. This is the smallest shaping interval allowed on this type of router and in this IOS version. We found that other Cisco routers and IOS versions allow different minimum shaping intervals, generally longer than 4ms.

The RUDE/CRUDE package includes a script that can compute throughput at the UDP payload level with time granularity of 100ms, 10ms or 1ms. The first choice is optimal for Bc=500kb. However, it is not possible to use this script to compute throughput with 4ms granularity, which is needed to verify correctness of traffic shaping for Bc=20kb. With our utility, it is possible to measure all QoS characteristics with arbitrary time granularity.

Some of the throughput graphs show fluctuations caused by a low number of packets sent during the shaping interval. In the worst case of 1500-byte packets and the shaping interval of 4ms, theoretically 1.644 packets should be sent per shaping interval. When we compute throughput over a certain period of time (given by time granularity), we consider all packets whose last bit was received within that period, even though the first and last packet could be in part received in the previous or following period, respectively. Moreover, it seems unlikely that the router can schedule individual packets in ideal times so that no more than Bc bits are sent during the shaping interval. At best, it would probably continue sending packets in some rate until Bc bits is sent in the shaping interval.

Measured throughput, delay and delay variation characteristics are shown in the following figures. 1500-byte streams are depicted in solid lines, 256-byte streams are depicted in dashed lines. In some cases there are two separate graphs for 1500-byte and 256-byte streams, respectively.

6.1 Comments on individual figures and observations:

- Fig. 5: Constant input stream is passed through as constant output stream.
- Fig. 6: Measuring throughput in finer granularity than the set shaping interval reveals that the constant input stream actually produces some fluctuations at the output (these fluctuations are really present in the stream, there are not result of a low number of packets within a shaping interval as described before). However, these fluctuations are allowed

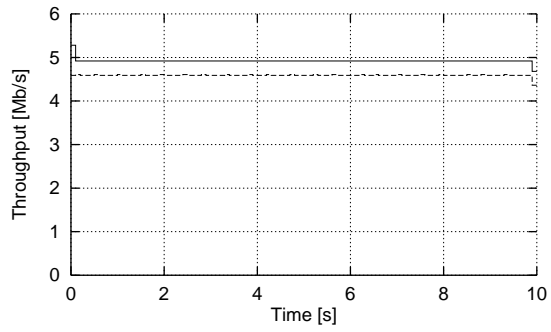


Figure 5: Throughput - input stream 1, Bc=500kb, 100ms granularity

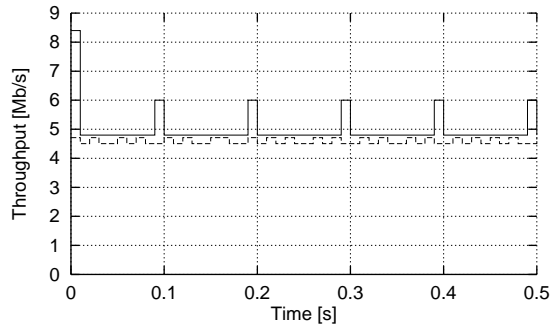


Figure 6: Throughput - input stream 1, Bc=500kb, 10ms granularity

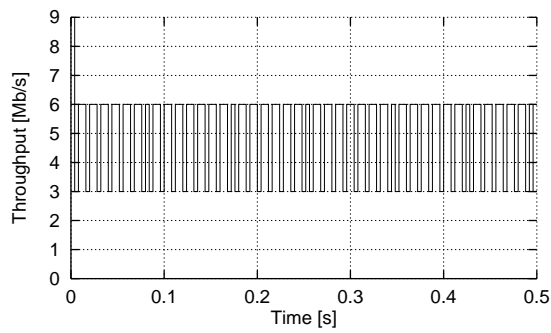


Figure 7: Throughput - input stream 1 (1500-byte), Bc=20kb, 4ms granularity

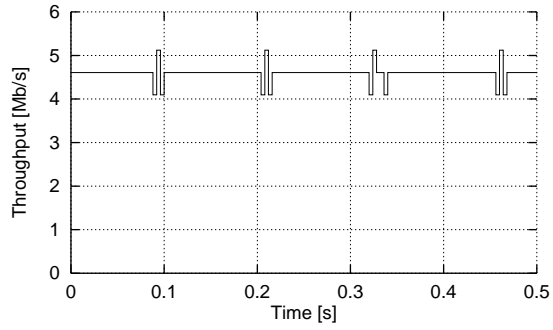


Figure 8: Throughput - input stream 1 (256-byte), Bc=20kb, 4ms granularity

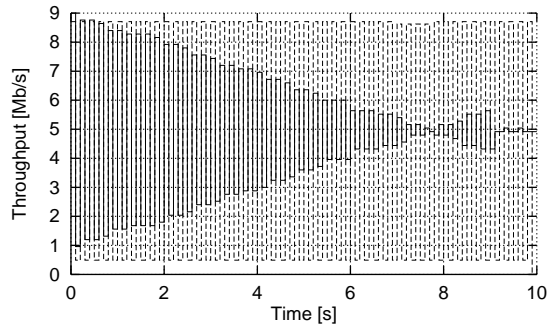


Figure 9: Throughput - input stream 2, Bc=500kb, 100ms granularity

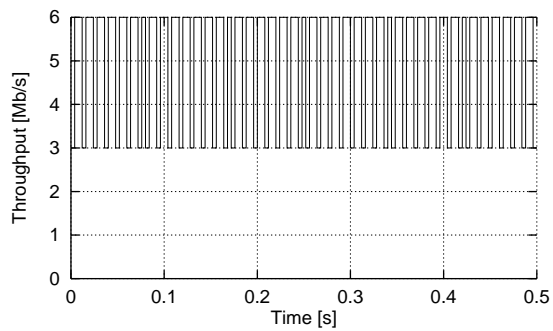


Figure 10: Throughput - input stream 2 (1500-byte), Bc=20kb, 4ms granularity

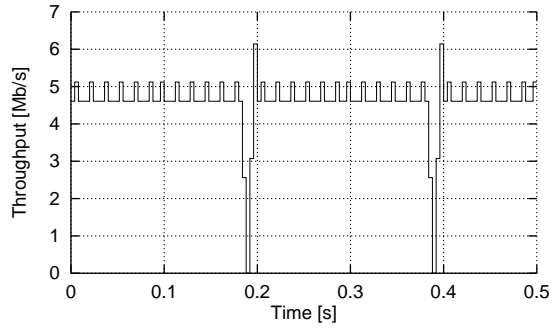


Figure 11: Throughput - input stream 2 (256-byte), Bc=20kb, 4ms granularity

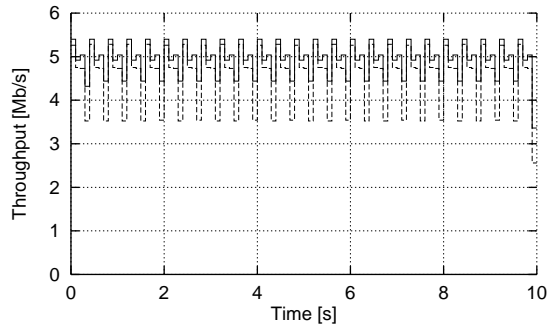


Figure 12: Throughput - input stream 3, Bc=500kb, 100ms granularity

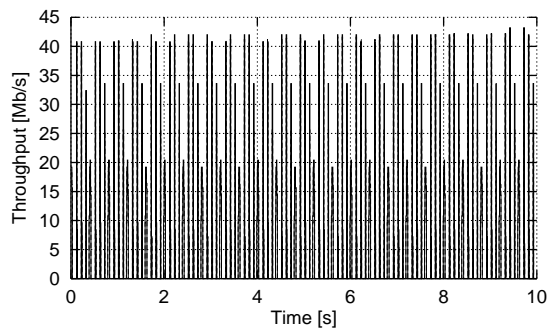


Figure 13: Throughput - input stream 3, Bc=500kb, 10ms granularity

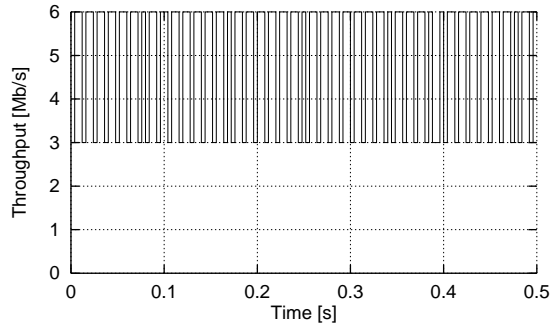


Figure 14: Throughput - input stream 3 (1500-byte), Bc=20kb, 4ms granularity

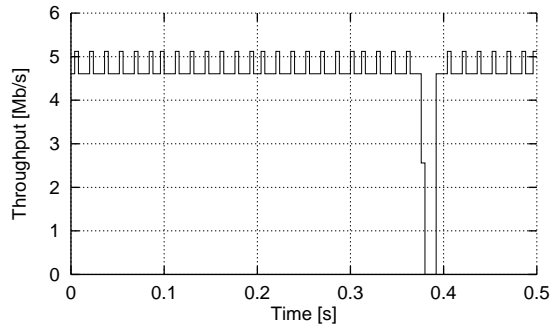


Figure 15: Throughput - input stream 3 (256-byte), Bc=20kb, 4ms granularity

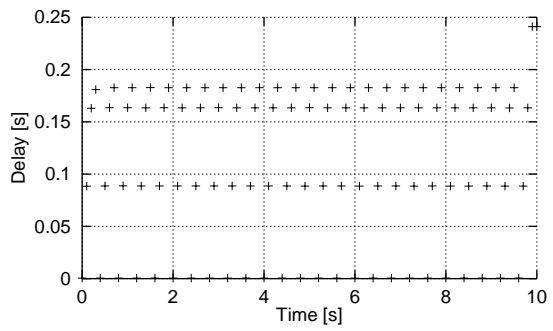


Figure 16: Delay - input stream 3, Bc=500kb, 100ms granularity

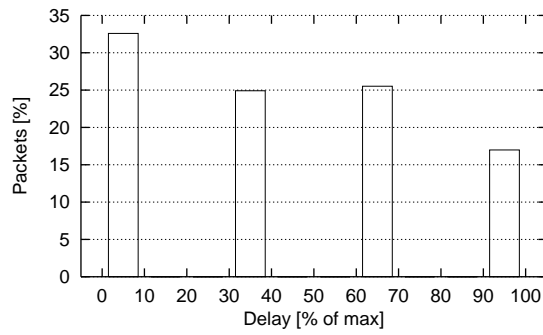


Figure 17: Delay distribution - input stream 3 (1500-byte), Bc=500kb, 100ms granularity

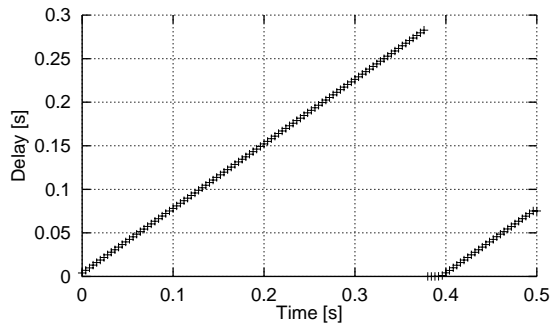


Figure 18: Delay - input stream 3 (256-byte), Bc=20kb, 4ms granularity

within the shaping interval as long as no more than Bc bits are set during the shaping interval.

- Fig. 7, Fig. 8: With the minimum shaping interval of 4ms, a low number of packets per shaping interval causes high fluctuations for large packets.
- Fig. 9: 1500-byte stream experiences some transitional effect during the first 9.5 seconds. After that the stream is perfectly shaped. However, the 256-byte stream is not shaped correctly.
- Fig. 10: Traffic shaping works, fluctuations are caused by the low number of packets per shaping interval.
- Fig. 11: Traffic shaping works very good, small fluctuations are caused by the low number of packets per shaping interval, large drops every 0.2s are probably caused by using input stream with slightly lower long-term bandwidth than the set CIR=5Mb/s.
- Fig. 12: Traffic shaping works for both 1500-byte and 256-byte streams. Considering relatively large shaping interval (and thus a large number of packets per shaping interval and lower requirements on the precision of scheduling in the router), resulting fluctuations are relatively large, but probably acceptable for practical use.
- Fig. 13: Fluctuations within a shaping period are allowed, but the peaks in this case are rather high.
- Fig. 14: Similar case as shown in Fig. 6, but fluctuations within shaping intervals are quite large. Considering a very fine shaping interval, resulting fluctuations are acceptable.
- Fig. 15: Traffic shaping works very good, similar case as in Fig. 11.
- Fig. 16: Average delay per shaping interval circulates between a very low delay and several higher values. The reason is that the router passes some of the packets from each input burst to the output almost immediately (with packet processing delay) until Bc bits are sent out. The remaining packets of the input burst must be delayed and sent during the following shaping intervals to keep the maximum of Bc bits sent out in a shaping interval.
- Fig. 17: Delay distribution of packets that passed through a router exhibits four significant peaks. The first column corresponds to packets sent during the same shaping period in which they arrived. The other three columns correspond to packets that were delayed and sent out during one of the next three shaping periods.

- Fig. 18: With fine shaping interval and small packets, delay linearly increases within the period of input bursts as packets are sent in small quantities to produce a smoothly shaped stream.

6.2 Influence of traffic shaping on the processor load

The following table shows the load of both processors when routing the input stream 1 (steady 5Mb/s) without and with traffic shaping:

Bc	Packet size	FEIP2-DSW-2TX load	VIP4-80 load
no shaping	1500 bytes	1 %	0 %
no shaping	256 bytes	9 %	2 %
500 kb	1500 bytes	2 %	1 %
500 kb	256 bytes	16 %	7 %
20 kb	1500 bytes	3 %	29 %
20 kb	256 bytes	14 %	99 %

Table 3: Influence of traffic shaping on processor load

6.3 Observations:

- Load of the FEIP2-DSW-2TX processor increases more as a result of sending shorter packets while the load of the VIP4-80 processor increases more as a result of finer traffic shaping. This is not surprising, because traffic shaping is performed by the output interface.
- With the smallest possible shaping interval of 4ms and short packets, the VIP4-80 processor becomes overloaded.

7 Conclusion

We found that with the hardware and software equipment used, traffic shaping is working, in most cases, with precision sufficient for practical purposes - passing bursty traffic through a limited bandwidth channel, when introducing delay and jitter is acceptable and preventing congestion, which could be caused by aggregation within diffserv domain. However, there are certain issues, which should be considered. First, with input stream 2 traffic shaping did not work for Bc=500kb. The reason is not clear and requires further investigation. Second, with a very small shaping interval and small packets, the VIP4-80 is highly loaded. Finally, while the router correctly maintains the limit of Bc bits which can be sent out during the shaping period, high peaks within the shaping period can occur. These short peaks, however, should not cause congestion resulting from aggregation.

References

- [nanokernel] D.L. Mills, P.-H. Kamp: *The Nanokernel*, Proc. of Precision Time and Time Interval (PTTI) Applications and Planning Meeting, Reston, VA, November 2000.
- [qofis2001] Sven Ubik, Vladimír Smotlacha, Sampo Saaristo, Juha Laine: *Low-Cost Precise QoS Measurement Tool*, submitted to QofIS, the 2nd International Workshop on Quality of Future Internet Services.
- [gnuplot] Thomas Williams, Colin Kelley, et al.: *Gnuplot: An Interactive Plotting Program*
- [mgen] Brian Adamson, Sean Gallavan: *The Multi Generator (MGEN) Toolset*, <http://manimac.itd.nrl.navy.mil/MGEN>⁴

⁴<http://manimac.itd.nrl.navy.mil/MGEN>